CodeCrunch

CS1010E 16/17S1 Graph Plot (Question)

Tags & Categories

Tags:

Categories:

Related Tutorials

Task Content

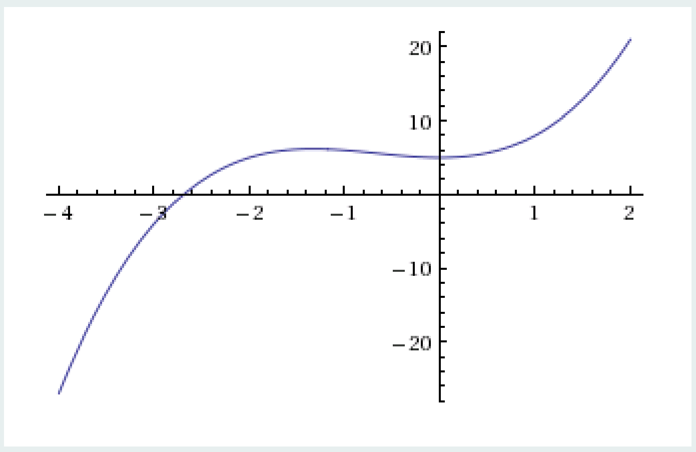
Graph Plot

Topic Coverage

- Assignment and expressions
- Nested control statements
- Functions and procedures

Problem Description

Given the graph of a polynomial `p(x) = `x^3` + 2`x^2` + 5` shown below,



`p(x)` can be plotted on a `20 x 60` rectangular grid within say, the domain interval `x = [-4, 2]` as shown.

```
                                       |                   x
                                       |                 xx
                                       |                x
                                       |              xx
                                       |            xx
                                       |        xxxx
                   xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
               xxxx                    |
-------------xx------------------------+--------------------
           xx                          |
         xx                            |
        x                              |
       x                               |
      x                                |
    xx                                 |
   x                                   |
                                       |
  x                                    |
 x                                     |
x                                      |
```

To generate a plot on a rectangular grid of a fixed width and height, grid locations need to be mapped to discrete points on the graph, and vice-versa. A high-level algorithm to generate the plot is given below.

```
Go through each of the grid locations (r, c),
    Find the x value associated with c
    Compute y = p(x)
    Find r' associated with the value y
    Mark on the location (r, c) if r = r'
```

As the above relies heavily on associating graph points to grid locations, and associating grid locations to graph points, it is advisable to have these functionalities "abstracted" as functions, say `gridToGraph` and `graphToGrid`.

For illustration purposes, the `20 x 60` plot of the same polynomial within the domain `x = [-3, -2]` is shown below,

```
                                                        xxxx
                                                  xxxxxx
                                              xxxx
                                         xxxxx
                                     xxxx
                                  xxx
                              xxxx
                           xxx
                        xxx
                      xx
                   xxx
----------------xxx-----------------------------------------
              xx
            xx
          xx
       xxx
     xx
   xx
 xx
x
```

and that plotted within the domain `x = [-4, 0]` is shown below.

```
                               xxxxxxxxxxxxxxxxxxxxxx      |
                          xxxxx                      xxxxxxx
                       xxx                                 |
                    xxx                                    |
------------------xx---------------------------------------+
                xx                                         |
              xx                                           |
             x                                             |
           xx                                              |
          x                                                |
         x                                                 |
        x                                                  |
      xx                                                   |
     x                                                     |
    x                                                      |
                                                           |
   x                                                       |
  x                                                        |
 x                                                         |
x                                                          |
```

Notice that the range of the polynomial function (i.e. the spread of possible p(x) values) is adjusted such that all possible `p(x)` values evaluated using the domain of x values can be shown on the plot. Specifically, each column has exactly one x marked on the plot.

Task

Write a program that asks the user to enter the integer coefficients (c_3, c_2, c_1, c_0) for a polynomial of degree 3:

$\mathbf{c_3x^3 + c_2x^2 + c_1x + c_0}$.

It then asks for the interval $[x_l, x_h]$, which are real numbers.

The program then computes the range $[y_l, y_h]$ for which the graph appears, and plots the polynomial within the two intervals using a `20 x 60` rectangular grid. In addition, show the x-axis and/or y-axis if it appears within the plot.

Take note of the following:

- Assume that the user enters a polynomial function of at least degree 1, i.e. a linear polynomial.
- Use the **double** data types for real numbers.
- You may use the `pow` and `rint` C Math library functions.

This task is divided into several levels. Read through all the levels (from first to last, then from last to first) to see how the different levels are related. **You may start from any level.**

- **Deadline: Submit your work to CodeCrunch by Thursday, 13 October, 23:59:59.**

Level 1

Name your program `plot1.c`

Write a program that reads in four integer coefficients (c_3, c_2, c_1, c_0) followed by two floating-point values x_l and x_h representing the endpoints of the domain interval $[x_l, x_h]$. Output the values of the coefficients and interval on separate lines.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
1 2 0 5
-4.0 2.0
1 2 0 5
xl = -4.000000; xh = 2.000000
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < plot.in | diff - plot1.out
```

To proceed to the next level (say level 2), copy your program by typing the Unix command

```
cp plot1.c plot2.c
```

Level 2

Name your program `plot2.c`

Write a program that reads in four integer coefficients (c_3, c_2, c_1, c_0) followed by two floating-point values x_l and x_h representing the endpoints of the domain interval $[x_l, x_h]$. Output the values of the coefficients and interval on separate lines.

To plot on a rectangular grid consisting of `60` columns, each of the column value must fall at fixed intervals along the domain of x. Specifically, the first column value is x_l, the last column value is x_h, and the interval length between any neighboring column values x_i and x_{i-1} must be the same.

Output all sixty domain values on separate lines.

You are encouraged to define the following function:

`double gridToGraph(int i, double low, double high, int n);`

Using the mapping of n equidistant points along the real value interval `[low, high]`, return the real-value associated with the i^{th} point.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
1 2 0 5
-4.0 2.0
1 2 0 5
xl = -4.000000; xh = 2.000000
-4.000000
-3.898305
-3.796610
-3.694915
-3.593220
-3.491525
```

```
-3.389831
-3.288136
-3.186441
-3.084746
-2.983051
-2.881356
-2.779661
-2.677966
-2.576271
-2.474576
-2.372881
-2.271186
-2.169492
-2.067797
-1.966102
-1.864407
-1.762712
-1.661017
-1.559322
-1.457627
-1.355932
-1.254237
-1.152542
-1.050847
-0.949153
-0.847458
-0.745763
-0.644068
-0.542373
-0.440678
-0.338983
-0.237288
-0.135593
-0.033898
0.067797
0.169492
0.271186
0.372881
0.474576
0.576271
0.677966
0.779661
0.881356
0.983051
1.084746
1.186441
1.288136
1.389831
1.491525
1.593220
1.694915
1.796610
1.898305
2.000000
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < plot.in | diff - plot2.out
```

To proceed to the next level (say level 3), copy your program by typing the Unix command

```
cp plot2.c plot3.c
```

Level 3

Name your program `plot3.c`

Write a program that reads in four integer coefficients (c_3, c_2, c_1, c_0) followed by two floating-point values x_l and x_h representing the endpoints of the domain interval $[x_l, x_h]$. Output the values of the coefficients and interval on separate lines.

To plot on a rectangular grid consisting of `60` columns, each of the column value must fall at fixed intervals along the domain of x. In addition, if the y-axis (i.e. `x = 0`) is within the domain $[x_l, x_h]$, then this value corresponds to a certain column number within the plot.

You are encouraged to define the following functions:

`double gridToGraph(int i, double low, double high, int n);`

Using the mapping of n equidistant points along the real value interval `[low, high]`, return the real-value associated with the `i`th point.

`int graphToGrid(double x, double low, double high, int n);`

Using the mapping of n equidistant points along the real value interval `[low, high]`, return `i` for which the `i`th point is closest to the real-value x. Use the `rint` C Math library function to perform integer rounding.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
1 2 0 5
-4.0 2.0
1 2 0 5
xl = -4.000000; xh = 2.000000
y-axis is located at column 40
column 40 is x = -0.033898
```

```
$ ./a.out
1 2 0 5
-3.0 -2.0
1 2 0 5
xl = -3.000000; xh = -2.000000
y-axis is outside the plot
```

```
$ ./a.out
1 2 0 5
-4.0 0.0
```

```
1 2 0 5
xl = -4.000000; xh = 0.000000
y-axis is located at column 60
column 60 is x = 0.000000
```

```
$ ./a.out
1 2 0 5
0.0 2.0
1 2 0 5
xl = 0.000000; xh = 2.000000
y-axis is located at column 1
column 1 is x = 0.000000
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < plot.in | diff - plot3.out
```

To proceed to the next level (say level 4), copy your program by typing the Unix command

```
cp plot3.c plot4.c
```

Level 4

Name your program `plot4.c`

Write a program that reads in four integer coefficients (c_3, c_2, c_1, c_0) followed by two floating-point values x_l and x_h representing the endpoints of the domain interval $[x_l, x_h]$.

To plot on a rectangular grid consisting of `60` columns, each of the column value must fall at fixed intervals along the domain of `x`. Go through the `x` values associated with each of the `60` columns to find the minimum and maximum `p(x)` values. This will be used as the range of `p(x)` values for the plot.

You are encouraged to define the following functions:

`double gridToGraph(int i, double low, double high, int n);`
: Using the mapping of `n` equidistant points along the real value interval `[low, high]`, return the real-value associated with the `i`th point.

`int graphToGrid(double x, double low, double high, int n);`
: Using the mapping of `n` equidistant points along the real value interval `[low, high]`, return `i` for which the `i`th point is closest to the real-value `x`. Use the `rint` C Math library function to perform integer rounding.

`double polynomial(double x, int c3, int c2, int c1, int c0);`
: Evaluates and returns the value of the polynomial $c_3x^3 + c_2x^2 + c_1x + c_0$

`void findMinMax(int c3, int c2, int c1, int c0, double *yl, double *yh, double xl, double xh);`
: Finds the minimum y_l and maximum y_h values upon evaluating the polynomial $c_3x^3 + c_2x^2 + c_1x + c_0$ over `60` equidistant points along the domain $[x_l, x_h]$.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
1 2 0 5
-4.0 2.0
Domain: [-4.000000, 2.000000] with y-axis located at column 40
Range: [-27.000000, 21.000000] with x-axis located at row 12
```

```
$ ./a.out
1 2 0 5
-3.0 -2.0
Domain: [-3.000000, -2.000000] with y-axis outside the plot
Range: [-4.000000, 5.000000] with x-axis located at row 9
```

```
$ ./a.out
1 2 0 5
-4.0 0.0
Domain: [-4.000000, 0.000000] with y-axis located at column 60
Range: [-27.000000, 6.184152] with x-axis located at row 16
```

```
$ ./a.out
1 2 0 5
0.0 2.0
Domain: [0.000000, 2.000000] with y-axis located at column 1
Range: [5.000000, 21.000000] with x-axis outside the plot
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < plot.in | diff - plot4.out
```

To proceed to the next level (say level 5), copy your program by typing the Unix command

```
cp plot4.c plot5.c
```

Level 5

Name your program `plot5.c`

Write a program that reads in four integer coefficients (c_3, c_2, c_1, c_0) followed by two floating-point values x_l and x_h representing the endpoints of the domain interval $[x_l, x_h]$. Output the `20 x 60` plot of the function within the domain interval. Show the x-axis and or y-axis if it appears in the plot.

Hint:

- For each column, there will be exactly one associated row in which to mark the plot; the vice-versa is however not true.
- Following convention, rows are output from last row to first.

You are encouraged to define the following functions:

`double gridToGraph(int i, double low, double high, int n);`

Using the mapping of n equidistant points along the real value interval `[low, high]`, return the real-value associated with the `i`th point.

`int graphToGrid(double x, double low, double high, int n);`

Using the mapping of n equidistant points along the real value interval `[low, high]`, return `i` for which the `i`th point is closest to the real-value `x`. Use the `rint` C Math library function to perform integer rounding.

`double polynomial(double x, int c3, int c2, int c1, int c0);`

Evaluates and returns the value of the polynomial $c_3x^3 + c_2x^2 + c_1x + c_0$

`void findMinMax(int c3, int c2, int c1, int c0, double *yl, double *yh, double xl, double xh);`

Finds the minimum y_l and maximum y_h values upon evaluating the polynomial $c_3x^3 + c_2x^2 + c_1x + c_0$ over `60` equidistant points along the domain $[x_l, x_h]$.

`void plotGraph(int c3, int c2, int c1, int c0, double xl, double xh)`

Plots the graph of the polynomial $c_3x^3 + c_2x^2 + c_1x + c_0$ over the domain $[x_l, x_h]$.

The following is a sample run of the program. User input is underlined. Ensure that the last line of output is followed by a newline character.

```
$ ./a.out
1 2 0 5
-4.0 2.0
Domain: [-4.000000,2.000000]; Range: [-27.000000,21.000000]
                                       |                   x
                                       |                 xx
                                       |                x
                                       |              xx
                                       |            xx
                                       |        xxxx
                   xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
               xxxx                    |
-------------xx------------------------+--------------------
           xx                          |
         xx                            |
        x                              |
       x                               |
      x                                |
    xx                                 |
   x                                   |
                                       |
  x                                    |
 x                                     |
x                                      |
```

```
$ ./a.out
1 2 0 5
-3.0 -2.0
Domain: [-3.000000,-2.000000]; Range: [-4.000000,5.000000]
                                                        xxxx
                                                  xxxxxx
                                              xxxx
                                         xxxxx
                                     xxxx
                                  xxx
                              xxxx
                           xxx
                        xxx
                      xx
                   xxx
----------------xxx-----------------------------------------
              xx
            xx
          xx
       xxx
     xx
   xx
 xx
x
```

```
$ ./a.out
1 2 0 5
-4.0 0.0
Domain: [-4.000000,0.000000]; Range: [-27.000000,6.184152]
                               xxxxxxxxxxxxxxxxxxxxxx      |
                          xxxxx                      xxxxxxx
                       xxx                                 |
                    xxx                                    |
------------------xx---------------------------------------+
                xx                                         |
              xx                                           |
             x                                             |
           xx                                              |
          x                                                |
         x                                                 |
        x                                                  |
      xx                                                   |
     x                                                     |
    x                                                      |
                                                           |
   x                                                       |
  x                                                        |
 x                                                         |
x                                                          |
```

```
$ ./a.out
1 2 0 5
0.0 2.0
Domain: [0.000000,2.000000]; Range: [5.000000,21.000000]
|                                                          x
|                                                         x
|                                                       xx
|                                                      x
|                                                    xx
|                                                   x
|                                                 xx
|                                                x
|                                              xx
|                                            xx
|                                          xx
|                                        xx
|                                      xx
|                                   xxx
|                                xxx
|                             xxx
|                         xxxx
|                    xxxxx
|            xxxxxxxx
xxxxxxxxxxxxx
```

```
$ ./a.out
0 1 0 -5
-5.1 5.0
Domain: [-5.100000,5.000000]; Range: [-4.998733,21.010000]
x                             |
 x                            |                            x
  x                           |                           x
                              |                          x
   x                          |                         x
    x                         |                        x
     x                        |                       x
      x                       |                      x
       x                      |                     x
        x                     |                    x
         x                    |                   x
          xx                  |                  x
            x                 |                 x
             x                |               xx
              xx              |              x
----------------xx------------+------------xx---------------
                  x           |          xx
                   xxx        |        xx
                      xxx     |    xxxx
                         xxxxxxxxxx
```

```
$ ./a.out
0 0 -1 0
-5.0 5.1
Domain: [-5.000000,5.100000]; Range: [-5.100000,5.000000]
xx                           |
  xxx                        |
     xxx                     |
        xxx                  |
           xxx               |
              xxxx           |
                  xxx        |
                     xxx     |
                        xxx  |
---------------------------xxx------------------------------
                             |xxx
                             |   xxx
                             |      xxx
                             |         xxx
                             |            xxxx
                             |                xxx
                             |                   xxx
                             |                      xxx
                             |                         xxx
                             |                            xx
```

Click here to submit to CodeCrunch.

Check the correctness of the output by typing the following Unix command

```
./a.out < plot.in | diff - plot5.out
```

Submission (Course)

Select course: CS1010E (2016/2017 Sem 1) - Programming Methodology ▼

Your Files:

SUBMIT (only .java, .c, .cpp and .h extensions allowed)

To submit multiple files, click on the Browse button, then select one or more files. The selected file(s) will be added to the upload queue. You can repeat this step to add more files. Check that you have all the files needed for your submission. Then click on the Submit button to upload your submission.