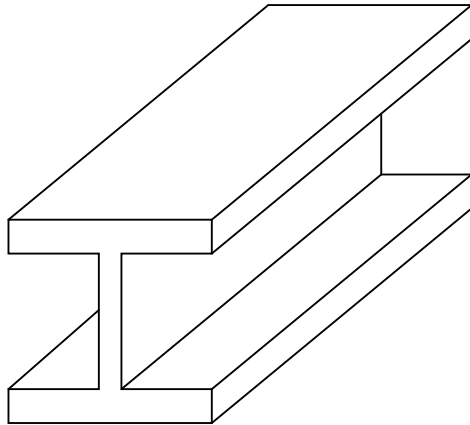


CEE384

# Interpolation

*Interpolation of Discrete Datasets*



ARIZONA STATE UNIVERSITY

*Brian Chevalier*

Updated: October 21, 2019

## Contents

|     |                                |   |
|-----|--------------------------------|---|
| 1   | Linear Interpolation           | 1 |
| 2   | Quadratic Interpolation        | 1 |
| 3   | Quadratic Splines              | 1 |
| 3.1 | Continuous Functions . . . . . | 2 |
| 3.2 | Smooth Functions . . . . .     | 2 |
| 4   | Coding Quadratic Splines       | 4 |

## 1 Linear Interpolation

Linear interpolation is estimating the value between two data points given an input,  $x$ . The output  $y$  is given by solving for the point along a line between the two given data points in the data set. In figure 1, two data points are given  $(x_0, y_0)$  and  $(x_1, y_1)$ .

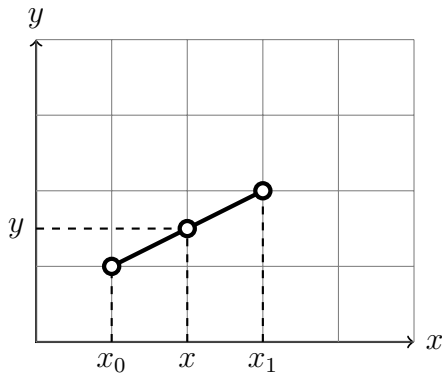


Figure 1: Example dataset plot with splines

From the figure, similar triangles are used to relate the three points.

$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0} \quad (1)$$

Next, the output coordinate can be solved for:

## 2 Quadratic Interpolation

## 3 Quadratic Splines

A *quadratic spline* is a set of piecewise quadratic functions which intersect all points in a dataset. There are three conditions that mathematically govern these functions. The first condition is that the functions must be *continuous*. The second condition is that the functions must be *smooth*, meaning that the derivatives are continuous.

There are  $3n$  unknowns, where  $n$  is the number of data points in the dataset. This is because there is an  $a_i$ ,  $b_i$ , and  $c_i$ , for each  $i$  data point. We will look at a particular example with 4 points (shown in table 1), and three connecting splines.

Table 1: Example dataset for quadratic spline

| $x$ | $x_0$    | $x_1$    | $x_2$    | $x_3$    |
|-----|----------|----------|----------|----------|
| $y$ | $f(x_0)$ | $f(x_1)$ | $f(x_2)$ | $f(x_3)$ |

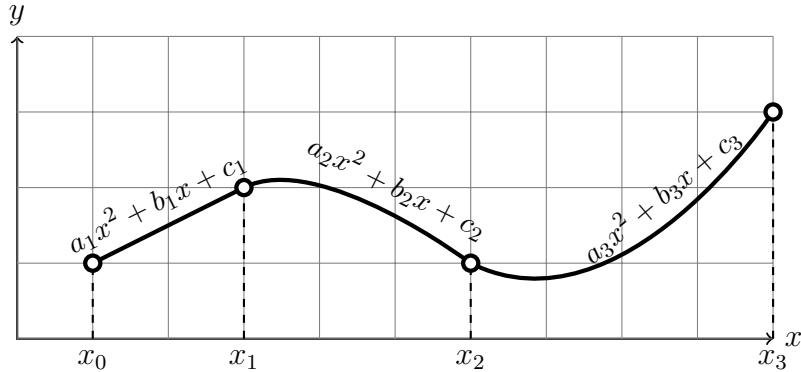


Figure 2: Example dataset plot with splines

### 3.1 Continuous Functions

The spline equations should be continuous and go through all the points in the dataset. This means that the

$$f(x) = \begin{cases} a_1x^2 + b_1x + c_1 & x_0 < x < x_1 \\ a_2x^2 + b_2x + c_2 & x_1 < x < x_2 \\ a_3x^2 + b_3x + c_3 & x_2 < x < x_3 \end{cases} \quad (3)$$

The first spline equation must be equal at the start of the interval,  $x_1$ , to the value of dataset  $f(x_0)$ . As an equation this is:

$$f(x_0) = a_1x_0^2 + b_1x_0 + c_1 \quad (4)$$

At the point,  $x_1$ , the first spline equation must be equal to the output value from the dataset,  $f(x_1)$ . The second equation must *also* be equal to this. Therefore this generates two additional equations:

$$f(x_1) = a_1x_1^2 + b_1x_1 + c_1 \quad (5)$$

$$f(x_1) = a_2x_1^2 + b_2x_1 + c_2 \quad (6)$$

At the second data point,  $x_2$ , the second and third spline must be equal to  $f(x_2)$ . This generates two additional equations.

$$f(x_2) = a_2x_2^2 + b_2x_2 + c_2 \quad (7)$$

$$f(x_2) = a_3x_2^2 + b_3x_2 + c_3 \quad (8)$$

Finally, the last spline must go through the last point, generating one more continuity equation:

$$f(x_3) = a_3x_3^2 + b_3x_3 + c_3 \quad (9)$$

This results in 6 equations total leaving 3 more unknowns that must be found. To generalize this pattern: for every  $n$  points the continuity condition generates  $2(n - 1)$  equations.

### 3.2 Smooth Functions

Next, take the derivative of the three spline equations:

$$f'(x) = \begin{cases} 2a_1x + b_1 \\ 2a_2x + b_2 \\ 2a_3x + b_3 \end{cases} \quad (10)$$

We know that the derivatives of the interior points must be continuous. We take the first two derivatives, evaluate them at the point  $x_1$ , and set them equal to each other. The same is done for spline 2 and 3. This results in two additional equations (11, 12).

$$2a_1x_1 + b_1 = 2a_2x_1 + b_2 \quad (11)$$

$$2a_2x_2 + b_2 = 2a_3x_2 + b_3 \quad (12)$$

Rewriting the equations to move all the variables to the left side:

$$2a_1x_1 + b_1 - 2a_2x_1 - b_2 = 0 \quad (13)$$

$$2a_2x_2 + b_2 - 2a_3x_2 - b_3 = 0 \quad (14)$$

To generalize this pattern, the smooth function condition produces  $(n - 2)$  equations. We now have 9 unknowns but only 8 equations. The last equation is the decision that the first spline will be linear. This is written in equation 15

$$a_1 = 0 \quad (15)$$

$$\begin{array}{cccccccccc} & a_1 & b_1 & c_1 & a_2 & b_2 & c_2 & a_3 & b_3 & c_3 \\ f_1 & x_0^2 & x_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ f_2 & x_1^2 & x_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ f_3 & 0 & 0 & 0 & x_1^2 & x_1 & 0 & 0 & 0 & 0 \\ f_4 & 0 & 0 & 0 & x_2^2 & x_2 & 0 & 0 & 0 & 0 \\ f_5 & 0 & 0 & 0 & 0 & 0 & 0 & x_2^2 & x_2 & 0 \\ f_6 & 0 & 0 & 0 & 0 & 0 & 0 & x_3^2 & x_3 & 0 \end{array} \quad (16)$$

## 4 Coding Quadratic Splines

### Pseudocode

1. Define dataset or take dataset as input if programmed as a stand alone function
2. Initialize  $\mathbf{A}$  as a matrix of zeros  $(3n, 3n)$
3. Initialize  $\mathbf{b}$  as a matrix of zeros  $(3n, 1)$
4. Loop through number of points and use continuous function information
  - (a) Set values for first equation
5. Loop through number of points and use continuous derivatives information
  - (a) Use interior points
6. Solve the system of equations for the constants
7. Plot and output values
  - (a) Plot the piecewise quadratic function *only* in the intervals where they are valid.
  - (b) Scatter plot points
  - (c) Plot query point, if any