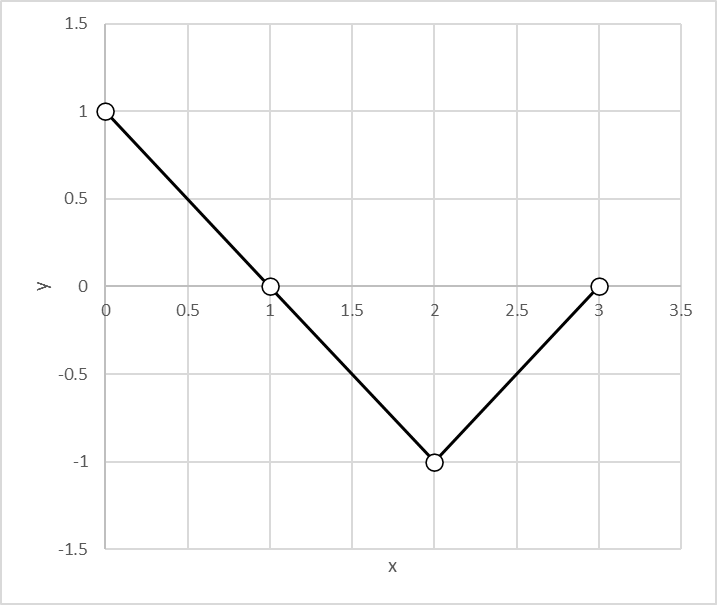
Text

Description automatically generated with medium confidence

Text

Description automatically generated with medium confidence



q0

q1

q2

Here is a plot of the data with just straight lines connecting the data points. We can observe that at x=2, the straight lines do not satisfy the condition of matching slope, nor do the end points match the given slopes k0 and k3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| j | xj | fj | kj | *g*'’(*xj*) |
| 0 | 0 | 1 | 0 | ? |
| 1 | 1 | 0 | ? | ? |
| 2 | 2 | -1 | ? | ? |
| 3 | 3 | 0 | -6 | ? |

For the cubic spline, *g(x)*, we want a cubic function composed of cubic polynomials *q0*, *q1*, *q2* that *i*) match the given data at the left and right limits of the x range for any interval and *ii*) has continuous first and second derivatives across all cubic functions or known slope and curvature at the end points.

Interpolation, **condition *i*)**:

and for j=1,…,n-1

Also

Continuous derivative, **condition *ii*):**

and for j=1,…,n-1

(note: the *j* subscript on *q* indicates which interval and the *j* subscript on *x* indicates which data point. So *qj* is for *x* values between *x­j* and *xj+1*)

Let’s let for j=0,1,…,n-1 be the inverse of the width of the interval.

Since we need the second derivative of our interpolating polynomial to be continuous, we can postulate that the second derivative of the interpolating function g(x) varies linearly across each interval, that is:

**Derivation of *g(x)*:**

For the derivation, I will simplify the symbols with the following substitutions:

∴

We need to integrate *g’’*(*x*) twice to get find *g*(*x*) *for this interval*, namely:

To determine the constants of integration, we apply the boundary conditions that

and

Thus **(1):** and **(2):**

Now **(2)**-**(1)** gives:

Substituting into (1) and solving for *C2*:

Applying these constants in *g(x)* gives:

Collecting terms and simplifying:

Intermediate check at this point (differentiate twice to get back linear variation of g’’):

(*so far, so good*!)

Let’s simplify a little more by remembering D=B-A:

Returning to the old symbols: (and many simplification steps):

Note:

Now, we finally have our interpolation function!

# Interpolating polynomial over segment *qj*:

In terms of Δx’s:

If I differentiate this function once, I get:

Differentiating a second time gives:

For *j*=1 we can evaluate the derivative of the integrating function at the right end (*x*2). Note that the *xj* on the right side of the equation are for the end points of the interval; *x1* and *x2* respectively:

­

, which means our interpolating function matches the second derivative requirement at *x*2.

For the interval *j*=2 between *x2* and *x3*, the derivative at the left side of the interval should match the derivative of the right end of the previous segment.

­

Now we equate the first derivatives at *x*2:

Multiply both sides by 6c1c2:

Dividing both sides by *c1­*:

The above equation was for *agreement* at *x*2 between cubics *q*1 and *q*2.

We can conclude that interior nodes (*j*=1,…,*n*-1) have the formula:

If spacing is equal, then we get:

Our problem has specified slopes at the first and last node. A.k.a.: “clamped” cubic spline.

For the first segment *j*=0, we have *g’*(x0)=k0, thus:

If spacing is equal, no change in this equation.

For the last segment (j=n-2) where n=4 for this case, we have *g’*(*x*3)=*k*3

Or

Finally, to solve our problem we use a matrix formula:

Special case of equal spacing in x:

In this particular problem the spacing is equal to one, so we can simplify

If we don’t specify the slopes at the end points, we have a “natural” cubic spline where the second derivative is specified at the end points (usually zero).

I’ll solve the problem using excel:

A blue square with black numbers and numbers on it

Description automatically generatedA graph of a line

Description automatically generated

A blue square with black numbers

Description automatically generated