Numerical Analysis (10th ed)

R.L. Burden, J. D. Faires, A. M. Burden

Chapter 3

Interpolation and Polynomial Approximation

Chapter 3.1: Interpolation and the Lagrange Polynomial*

The important ideas In this section are:

- How can we want to find a polynomial that agrees with (interpolates) a given function at more than one point and remains as close to the given function as we want?
- How can we calculate a bound for the error involved in approximating a function by the interpolating polynomial?

The first thought might be to use a Taylor polynomial. However, although a Taylor polynomial agrees with the function near a specific point, it does not agree with the function closely over a specified interval. The Lagrange interpolating polynomial was developed specifically for this purpose.

Suppose $x_0, x_1, \ldots, x_n$ are n+1 distinct numbers and f is a function whose values are given at these numbers. Theorem 3.2 guarantees the existence of a unique polynomial $P(x)$ of degree at most n with

$f(x_k) = P(x_k)$, for each $k = 0,1,\ldots,n$

given by

$$P(x) = f(x_0)L_{n,0}(x) + \ldots + f(x_n)L_{n,n}(x) = \sum_{k=0}^{n} f(x_k)L_{n,k}(x)$$

where, for each $k = 0,1,\ldots,n$,

$$L_{n,k}(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)(x_k-x_1)\cdots(x_k-x_{k-1})(x-x_{k+1})\cdots(x_k-x_n)} = \prod_{i=0,\, i\neq k}^{n} \frac{(x-x_i)}{(x_k-x_i)}$$

Thus, a Lagrange interpolating polynomial of degree one that agrees with f at $(x_0, f(x_0))$, $(x_1, f(x_1))$ would be given by:

$$P(x) = \frac{(x-x_1)}{(x_0-x_1)}f(x_0) + \frac{(x-x_0)}{(x_1-x_0)}f(x_1)$$

and a Lagrange interpolating polynomial of degree two that agrees with f at $(x_0, f(x_0))$, $(x_1, f(x_1))$, and $(x_2, f(x_2))$ would be given by:

$$P(x) = \frac{(x - x_1)\cdot(x - x_2)}{(x_0 - x_1)\cdot(x_0 - x_2)} f(x_0) + \frac{(x - x_0)\cdot(x - x_2)}{(x_1 - x_0)\cdot(x_1 - x_2)} f(x_1) + \frac{(x - x_0)\cdot(x - x_1)}{(x_2 - x_0)\cdot(x_2 - x_1)} f(x_2)$$

.

Example 4 from the text shows how the error formula given by

$\frac{f^{n+1}(\xi)}{(n+1)!}(x - x_0)^{n+1}$ can be used to determine a bound on the error when using Lagrange interpolation.

Let's look at the following example where $f(x) = \tan(x)$. The computer algebra system MAPLE was used to generate the polynomials of degree 1, 2, and the graphs.

$restart:$
$with(plots):$
$f := x \rightarrow \tan(x):$
$x_0 := 0:$
$x_1 := 0.6:$
$x_2 := 0.9:$
$$L1 := \frac{(x - x_1)}{(x_0 - x_1)} f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)} f(x_1):$$
$P1 := x \rightarrow 1.140228014\, x$

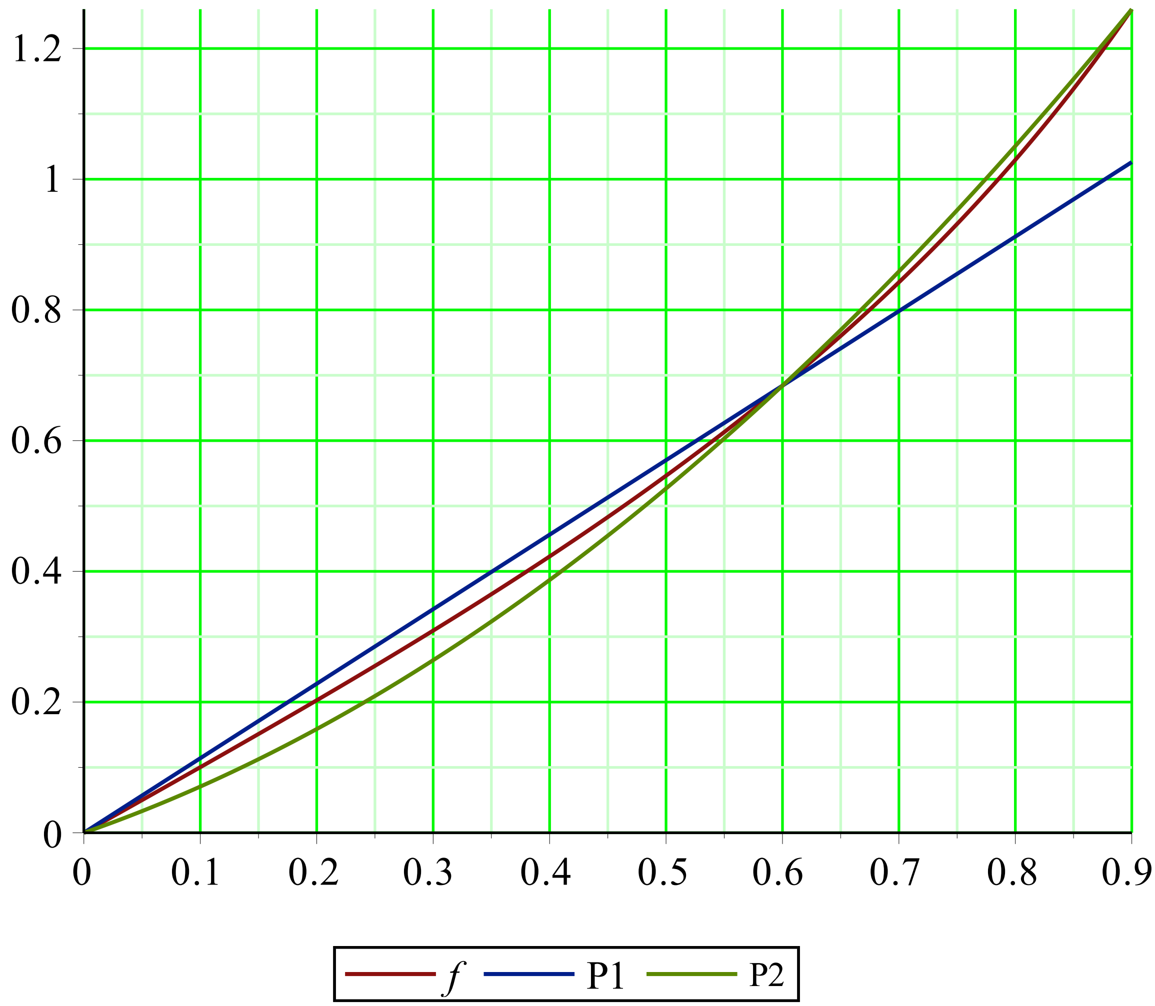
$$x \rightarrow 1.140228014\, x \tag{1.1}$$

$$L2 := \frac{(x - x_1)\cdot(x - x_2)}{(x_0 - x_1)\cdot(x_0 - x_2)} f(x_0) + \frac{(x - x_0)\cdot(x - x_2)}{(x_1 - x_0)\cdot(x_1 - x_2)} f(x_1) + \frac{(x - x_0)\cdot(x - x_1)}{(x_2 - x_0)\cdot(x_2 - x_1)} f(x_2):$$

$P2 := x \rightarrow 0.866492613\, x^2 + 0.620332446\, x$

$$x \rightarrow 0.866492613\, x^2 + 0.620332446\, x \tag{1.2}$$

$plot([f, P1, P2], 0..0.9, axis = [gridlines = [colour = green, majorlines = 2]])$



Notice that *P1* agrees with the values of f at x_0 and x_1 while *P2* agrees with the values of f at x_0, x_1, and x_2.

$f(0) = 0$
$P1(0) = 0.$
$P2(0) = 0.$
$f(0.6) = 0.6841368083$
$P1(0.6) = 0.6841368084$
$P2(0.6) = 0.6841368083$
$f(0.9) = 1.260158218$
$P1(0.9) = 1.026205213$
$P2(0.9) = 1.260158218$

The error bound for $P2(x)$ can be easily determined by maximizing $\left|\frac{f^3(\xi)}{(2+1)!}\right|$ on [0,0.9]. We see that the largest value obtained on [0,0.9] is

$$M := maximize\left(\frac{\left|2\sec^4(x) + 4\sec(x)\cdot\tan(x)\right|}{3!}, x = 0..0.9\right) = 3.584078513 .$$

Thus, the bound is

$M \cdot (x_2 - x_0) = 3.225670662$

Chapter 3.2: Data Approximation and Neville's Method*

In the last section we found that the Lagrange polynomials gave us an explicit formula for the approximation of a function on a given interval. You will notice that occasionally the function we are trying to approximate is given in these sections, however, in reality we quite often only have a finite set of data available to us and do not know the function *f*. In this situation we may not need to know the explicit formula for the approximating function, but we would like to be able to determine function values at certain *x* values of interest. Neville's method is used for this purpose.

Rather than using the Lagrange formula to find the approximating polynomial as we did in Section 3.1, for $f(x) = \tan(x)$, we use the formula for P(x) to calculate P(x*) where x* is some given value of x. From our work in the previous section we know that $f(x^*) \approx P(x^*)$. Thus, we can compute any approximation at a given value of x in this manner. For example, suppose we are given the following data: $(0, 0)$, $(0.6, 0.6841368083)$, and $(0.9, 1.260158218)$ and we want to use the 2nd degree interpolating polynomial to approximate *f*. The computations are in the following table:

$h := x \rightarrow \tan(x)$: The approximation $h(.7) = 0.8422883805$ is shown in the table below:

i	x_i	$f(x_i) \approx P(x_i)$			
0	0	0			
1	0.6	0.6841368083	$\frac{1}{(0.6-0)}\left((0.7-0)\cdot 0.6841368083 - (0.7-0.6)\cdot 0\right) = 0.7981596097$		
2	0.9	1.260158218	$\frac{1}{(0.9-0.6)}\left((0.7-0.6)\cdot 1.260158218 - (0.7-0.9)\cdot 0.6841368083\right) = 0.8761439450$	$\frac{1}{(0.9-0)}\left((0.7-0)\cdot 0.8761439450 - (0.7-0.9)\cdot 0.7981596097\right) = 0.8588140927$	
3	.95	1.398382589	$\frac{1}{(0.95-0.9)}\left((0.7-0.9)\cdot 1.398382589 - (0.7-0.95)\cdot 1.260158218\right)$	$\frac{1}{(0.95-0.6)}\left((0.7-0.6)\cdot 0.7072607340 - (0.7-0.95)\cdot 0.8761439450\right)$	$\frac{1}{(0.95-0)}\left((0.7-0)\cdot 0.8278915989 - (0.7-0.95)\cdot 0.8588140927\right)$

			$= 0.7072607340$	$= 0.8278915989$	) $= 0.8360290973$

As illustrated in this example, iterated interpolation is used to generate successively higher degree polynomial approximations at a specific point, say x*.

Chapter 3.3: Divided Differences*

The key question we want to address in this section is:
• What are other difference formulas for interpolation?

Newton's Divided Difference Method:
In the last section, iterated interpolation is used to generate successively higher degree polynomial approximations at a specific point, say x*. In this section we will use divided differences to successively generate the polynomials.

Given a set of data, we will look at template that should act as a guide in this process.

x_i	$f(x_i) \approx$	**First Divided Difference**	**Second Divided Difference (skip one x between in denominator)**	**Third Divided Difference (skip two xs between in denominator)**	**Fourth Divided Difference (skip three xs between in denominator)**
x_0	$f(x_0)$				
x_1	$f(x_1)$	$a_1 = (f(x_1) - f(x_0)$			
x_2	$f(x_2)$	$a_2 = (f(x_2) - f(x_1)$	$b_1 = \dfrac{a_2 - a_1}{x_2 - x_0}$		
x_3	$f(x_3)$	$a_3 = (f(x_3) - f(x_2)$	$b_2 = \dfrac{a_3 - a_2}{x_3 - x_1}$	$c_1 = 1 / (x_3$	
x_4	$f(x_4)$	$a_4 = (f(x_4) - f(x_3)$	$b_3 = \dfrac{a_4 - a_3}{x_4 - x_2}$	$c_2 = 1 / (x_4$	$d_1 = (c_2 - c_1) / (x_4 - x_0)$

The Newton's divided difference polynomial then becomes

$$P_4(x) = f(x_0) + a_1(x - x_0) + b_1(x - x_0)(x - x_1) + c_1(x - x_0)(x - x_1)(x - x_2) + d_1(x - x_0)(x - x_1)(x - x_2)(x - x_3)$$

Newton's Forward, Backward & Centered Difference Formulas:

The following table shows the difference formulas rooted in the Aitkin's $\triangle^2$ method:

x_i	$f(x_i) \approx$	**First Divided Difference** (same as first column in previous table)	**Second Divided Difference** (same as second column in previous table & also divide by 2)	**Third Divided Difference** (same as third column in previous table & also divide by 3)	**Fourth Divided Difference** (same as fourth column in previous table & also divide by 4)
x_0	$f(x_0)$				
x_1	$f(x_1)$	$a_1 = (f(x_1) - f(x_0))$			
x_2	$f(x_2)$	$a_2 = (f(x_2) - f(x_1))$	$b_1 = (a_2 - a_1) / (2(x_2 - x_0))$		
x_3	$f(x_3)$	$a_3 = (f(x_3) - f(x_2))$	$b_2 = (a_3 - a_2) / (2(x_3 - x_1))$	$c_1 = (b_2 - b_1) / (3(x_3 - x_0))$	
x_4	$f(x_4)$	$a_4 = (f(x_4) - f(x_3))$	$b_3 = (a_4 - a_3) / (2(x_4 - x_2))$	$c_2 = (b_3 - b_2) / (3(x_4 - x_1))$	$d_1 = (c_2 - c_1) / (4(x_4 - x_0))$

When approximating a value when x is close to the end of the tabulated values, we make use of the

earliest data points closest to x.
The **forward difference** polynomial is defined to be:

$$P_4(x) = f(x_0) + s_1 \cdot h \cdot a_1 + s_1 \cdot s_2 \cdot h^2 \cdot b_1 + s_1 \cdot s_2 \cdot s_3 \cdot h^3 \cdot c_1 + s_1 \cdot s_2 \cdot s_3 \cdot s_4 \cdot h^4 \cdot d_1$$

where $h = \dfrac{(x_4 - x_0)}{4}$ and $s_i = \dfrac{(x - x_i)}{h}$ for $i = 0,..., 4$.

When approximating a value when x is close to the end of the tabulated values, we make use of the earliest data points closest to x.
The **backward difference** polynomial is defined to be:

$$P_4(x) = f(x_4) + s_4 \cdot h \cdot a_4 + s_4 \cdot s_3 \cdot h^2 \cdot b_3 + s_4 \cdot s_3 \cdot s_2 \cdot h^3 \cdot c_2 + s_4 \cdot s_3 \cdot s_2 \cdot s_1 \cdot h^4 \cdot d_1$$

where $h = \dfrac{(x_4 - x_0)}{4}$ and $s_i = \dfrac{(x - x_i)}{h}$ for $i = 4,..., 0$ for this example.

x_i	$f(x_i) \approx P(x_i)$	**First Divided Difference** (same as first column in previous table)	**Second Divided Difference** (same as second column in previous table & also divide by 2)	**Third Divided Difference** (same as third column in previous table & also divide by 3)	**Fourth Divided Difference** (same as fourth column in previous table & also divide by 4)
x_0	$f(x_0)$				
x_1	$f(x_1)$	$a_1 = (f(x_1) - f(x_0)) / (x_1 - x_0)$			
x_2	$f(x_2)$	$a_2 = (f(x_2) - f(x_1)) / (x_2 - x_1)$	$b_1 = (a_2 - a_1) / (2(x_2 - x_0))$		
x_3	$f(x_3)$	$a_3 = (f(x_3) - f(x_2)) / (x_3 - x_2)$	$b_2 = (a_3 - a_2) / (2(x_3 - x_1))$	$c_1 = (b_2 - b_1) / (3(x_3 - x_0))$	
x_4	$f(x_4)$	$a_4 = (f(x_4)$	b_3	c_2	d_1

		$-f(x_3)\big)\big/(x_4-x_3)$	$=(a_4-a_3)\big/\big(2(x_4-x_2)\big)$	$=(b_3-b_2)\big/\big(3(x_4-x_1)\big)$	$=(c_2-c_1)\big/\big(4(x_4-x_0)\big)$

When approximating a value when x is close to the middle of the tabulated values, say near x_3, we make use of the earliest data points closest to x.
Stirling's centered difference polynomial is defined to be:

$$P_4(x) = f(x_3) + s_4 \cdot h \cdot a_4 + s_4 \cdot s_3 \cdot h^2 \cdot b_3 + s_4 \cdot s_3 \cdot s_2 \cdot h^3 \cdot c_2 + s_4 \cdot s_3 \cdot s_2 \cdot s_1 \cdot h^4 \cdot d_1$$

where $h = \dfrac{(x_4 - x_0)}{4}$ and $s = \dfrac{(x - x_3)}{h}$ for this example.

Chapter 3.4: Hermite Interpolation*

In this section we look at osculating polynomials that agree with values of f The text provides a detailed example of the Hermite
interpolating polynomial, so we provide a visual illustration of the graph of ff and the Hermite polynomial for the example in the text.

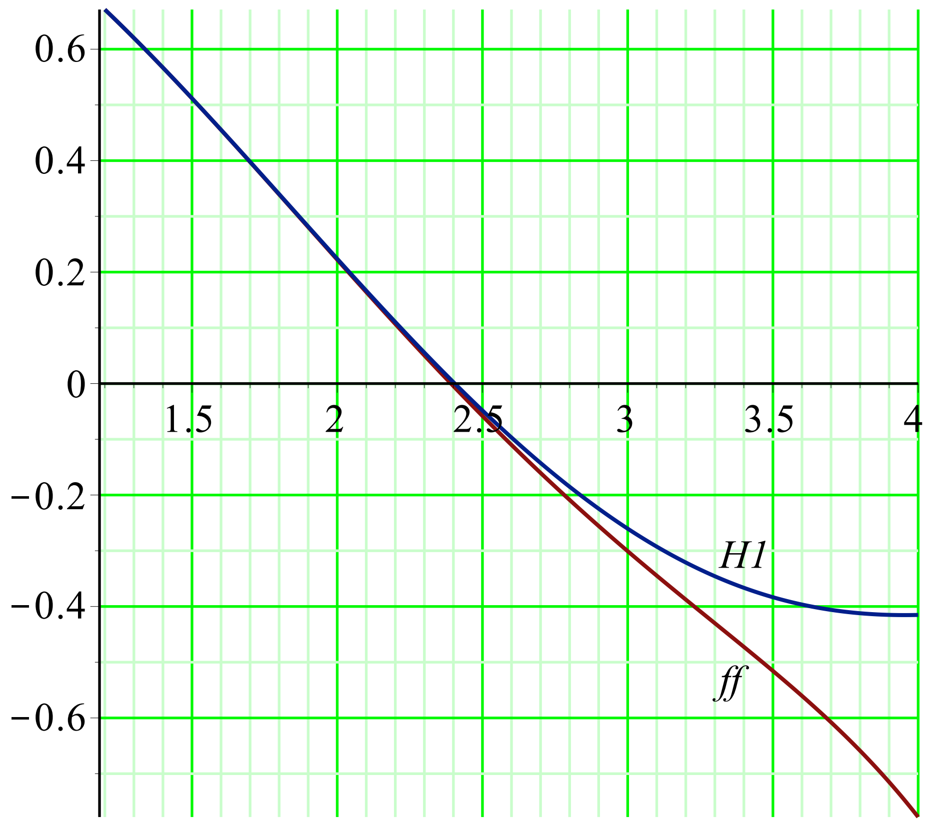
restart :

$$ff := x \to 1 - \frac{1}{4}x^2 + \frac{1}{64}x^4 - \frac{1}{2304}x^6 :$$

$$ff1 = 0.6200860 \cdot (10 \cdot x - 12) \cdot \left(\frac{50}{9}x^2 - \frac{175}{9}x + \frac{152}{9}\right)^2 + 0.4554022 \cdot \left(-\frac{100}{9}x^2 + \frac{320}{9}x - \frac{247}{9}\right)^2 + 0.2818186 \cdot 10 \cdot (2 - x) \cdot \left(\frac{50}{9}x^2 - \frac{145}{9}x + \frac{104}{9}\right)^2 - 0.5220232 \cdot (x - 1.3) \cdot \left(\frac{50}{9}x^2 - \frac{175}{9}x + \frac{152}{9}\right)^2 - 0.5698959 \cdot (x - 1.6) \cdot \left(-\frac{100}{9}x^2 + \frac{320}{9}x - \frac{247}{9}\right)^2 - 0.5811571 \cdot (x - 1.9) \cdot \left(\frac{50}{9}x^2 - \frac{145}{9}x + \frac{104}{9}\right)^2 :$$

$$H1 := x \to 1.001944063 - 0.0082292208 \cdot x - 0.2352161732 \cdot x^2 - 0.01455607812 \cdot x^3 + 0.02403178946 \cdot x^4 - 0.002774691277 \cdot x^5 :$$

$plot([ff, H1], 1.2 ..4.0, axis = [gridlines = [colour = green, majorlines = 2]])$



▶ **Chapter 3.5: Cubic Spline Interpolation**

▶ **Chapter 3.6: Parametric Curves**