Hermite-Interpolation (Osculation)

The method of divided differences with repeated arguments

1. The problem of osculation

When the problem of collocation is extended by the requirement that certain given values of derivatives of order 0 up to some higher order k of the model function y must be met at some of the arguments $x_{0,}, x_1, \cdots, x_n$ we end in an interpolation problem called osculation or Hermite interpolation. This kind of interpolation normally is more complicated than collocation. Generally, the interpolating polynomial has a higher degree than collocating polynomials; the degree, generally, equals "*n + the number of additional conditions on derivatives*" at most.

Example 1.1: A typical problem of osculation is the following one (cf. Schaum's Outline of Numerical Analysis, 2nd edition, Problem 10.12, p. 84):

Compute a polynomial p_2 whose graph passes through the points (2,1) and (4,2) (collocation conditions) and that meets the four derivative conditions $p_2'(2) = 1$, $p_2'(4) = 0$, $p_2''(2) = 0$ and $p_2''(4) = 0$.

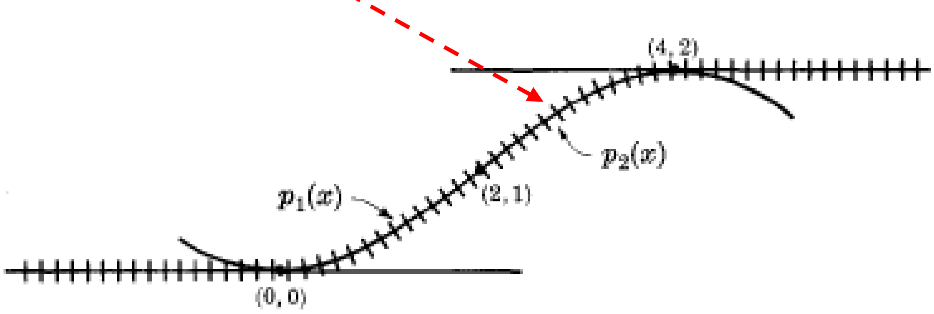

Fig. 10-2

2. The "elegant" solution by divided differences with repeated arguments

There are different methods for solving osculation problems like the naïve method of undetermined coefficients or the Hermite-Lagrange-formula for 1st order derivative conditions. But from the point of view of implementation and comprehension nothing (known to the author) beats the method of divided differences extended to the case where arguments may occur repeatedly.

2.1 The theoretical background

The starting point is the collocation error formula:

If y denotes a model function with n+1 continuous derivatives and p is the collocation polynomial co-allocating with y at the arguments $x_{0,}, x_1, \cdots, x_n$ then the collocation error is representable as follows:

$$y(x) - p(x) = \frac{y^{(n+1)}(\xi)}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n) = \frac{y^{(n+1)}(\xi)}{(n+1)!}\pi_{n+1}(x)$$
$$y(x) = p(x) + \frac{y^{(n+1)}(\xi)}{(n+1)!}\pi_{n+1}(x) \qquad x, \xi \in (\min x_i, \max x_i)_{i=0,1,\ldots,n} \qquad \textbf{(1)}$$

Considering $x = x_{n+1} \neq x_i \quad (i = 0, \cdots, n)$ as the argument of a further measurement the coefficient $\frac{y^{(n+1)}(\xi)}{(n+1)!}$ of the Newton polynomial $\pi_{n+1}(x)$ must equal the divided difference $y(x_0, \ldots, x_n, x) = y(x_0, \ldots, x_n, x_{n+1})$.

$$y(\underbrace{x_0, \ldots, x_n, x_{n+1}}_{(n+2)}) = \frac{y^{(n+1)}(\xi)}{(n+1)!} \qquad x = x_{n+1},\ \xi \in (\min x_i, \max x_i)_{i=0,1,\ldots,n} \tag{2}$$

$$y(\underbrace{x_0, \ldots, x_n}_{(n+1)}) = \frac{y^{(n)}(\xi)}{n!} \qquad \xi \in (\min x_i, \max x_i)_{i=0,1,\ldots,n} \tag{2'}$$

These formulas say that a divided difference can be considered as a derivative. For the case n = 1 this is not a surprise, because $y(x_0, x_1) = \frac{\Delta y_0}{\Delta x_0} = \frac{y(x_1) - y(x_0)}{x_1 - x_0} = y'(\xi) \quad \xi \in (x_0, x_1)$ by the classical mean value theorem of calculus in one variable. Proceeding with this special case and examing what happens when $x_0 = x_1$ (a situation called repetition of arguments) we deduce by a limiting argument that $y(x_0, x_0) = \lim_{x_1 \to x_0} \frac{y(x_1) - y(x_0)}{x_1 - x_0} = y'(x_0)$ by definition of the derivative. Of course, this requires that the derivative exists. By a similar argument applied to (2) we get the following formula for higher order divided differences with repeated arguments:

$$y(\underbrace{x_0, \ldots, x_0}_{(n+1)}) = \lim_{\xi \to x_0} \frac{y^{(n)}(\xi)}{n!} = \frac{y^{(n)}(x_0)}{n!} \tag{3}$$

A divided difference with repeated arguments thus can be interpreted as a derivative. This is the key to solving the osculation problem by divided differences with repeated arguments.

2.2 The method

When given "derivative" values $y^{(j)}(x_i) \quad (j = 0, \ldots, k)$ up to order k have to be met by the osculation polynomial the corresponding argument x_i is repeated k+1 times in the tableau of divided differences and the corresponding divided differences are set equal to $\frac{y^{(j)}(x_i)}{j!}$ according to (3).

The resulting tableau contains repetitions and it is solved by the usual Newton scheme of divided differences. Of course, the Newton basis polynomials have to be modified according to the repetitions of the arguments, but this modification is literally. Since the argument x_i occurs repeatedly (k+1 times) the corresponding part in the modified Newton polynomial is $\cdots \underbrace{(x - x_0)(x - x_0)\cdots(x - x_0)}_{(k+1)} \cdots$ instead of $\cdots (x - x_0) \cdots$. An example computation will illuminate the method and its elegance.

Example 1.2: This continues Example 1.1 from above. For the osculating polynomial p_2 we have the following initial incomplete tableau of divided differences with repetitions:

x	y		
$x_0 = 2$	$y(x_0) = 1$		
$x_0 = 2$	$y(x_0) = 1$	$\frac{y^{(1)}(x_0)}{1!} = 1$	
$x_0 = 2$	$y(x_0) = 1$	$\frac{y^{(1)}(x_0)}{1!} = 1$	$\frac{y^{(2)}(x_0)}{2!} = 0$
$x_1 = 4$	$y(x_1) = 2$	*	*
$x_1 = 4$	$y(x_1) = 2$	$\frac{y^{(1)}(x_1)}{1!} = 0$	*
$x_1 = 4$	$y(x_1) = 2$	$\frac{y^{(1)}(x_1)}{1!} = 0$	$\frac{y^{(2)}(x_1)}{2!} = 0$

This initial tableau is completed by the usual scheme of divided differences yielding the following lower left tableau:

x	y					
2	(1)					
2	1	(1)				
2	1	1	(0)			
4	2	[1/2]	[-1/4]	(-1/8)		
4	2	0	[-1/4]	[0]	(1/16)	
4	2	0	0	[1/8]	[1/16]	(0)

Zero means degree only 4 here

The framed values are divided differences computed by the usual scheme and the values encircled are the values on the main diagonal constituting the coefficients of the modified Newton polynomials. The solution then is:

$$P_2(x) = 1 + 1(x-2) + 0\,(x-2)^2 + \left(-\frac{1}{8}\right)(x-2)^3 + \left(\frac{1}{16}\right)(x-2)^3(x-4)$$

Note that the modified Newton polynomials have to be used:

$$\pi_0 = 1,$$
$$\pi_1 = (x - x_0) = (x-2)$$
$$\pi_2 = (x - x_0)(x - x_0) = (x-2)^2$$
$$\pi_3 = (x - x_0)(x - x_0)(x - x_0) = (x-2)^3$$

$$\pi_4 = (x-x_0)(x-x_0)(x-x_0)(x-x_1) = (x-2)^3(x-4)$$
$$\pi_5 = (x-x_0)(x-x_0)(x-x_0)(x-x_1)(x-x_1) = (x-2)^3(x-4)^2$$
...

3. The osculation error formula

Since the method in section 2 is a generalization of the usual method of divided differences we immediately get a generalization of the collocation error formula (1) for osculating polynomials:

$$y(x) - p(x) = \frac{y^{(d)}(\xi)}{d!}(x-x_0)^{d_0}(x-x_1)^{d_1}\cdots(x-x_n)^{d_n}$$
$$x, \xi \in (\min x_i, \max x_i)_{i=0,1,\ldots,n} \quad \textbf{(4)}$$

Here d denotes the total number of conditions and d_i denotes the number of conditions for the argument $x_i \quad (i = 0,\ldots,n)$. We have $d = d_0 + d_1 + \cdots + d_n$.

Example 1.3: Continuing the example above we get the expression

$\frac{y^{(6)}(\xi)}{6!}(x-2)^3(x-4)^3$ for the osculation error with $x, \xi \in (2;4)$.

4. Missing informations on lower order derivatives

The term Hermite interpolation or osculation refers to the case where given “derivative” values $y^{(j)}(x_i) \quad (j = 0,...,k)$ up to order k have to be met by the osculation polynomial. This means that there are no missing informations about the derivatives between the orders 0 and k for any of the arguments $x_i \quad (i = 0,...,n)$. This guarantees that there will be a unique solution for the interpolation polynomial (with optimal degree). The following example shows that missing information on derivatives can lead to unsolvable interpolation problems.

Example 1.4: There is no polynomial p (of order 2) with $p'(0)$ = 1 and $p(-1)$ = 0 = $p(1)$ since such a polynomial graphs as a parabola and must have a zero derivative in the mean of its zeroes.

If the method of divided differences is tried with a parameter a for $p(0)$ then it results necessarily in a cubic polynomial although there are only three conditions:

x	y			
-1	0			
0	a	a		
0	a	$y'(0)$=1	1-a	
1	0	-a	- a-1	-2/2

Thus $p(x) = a(x+1) + (1-a)(1+x)x - 1(x+1)x^2$ which is necessarily cubic. Working with parameters for unknown values often is the first strategy for solving osculation problems.

The subject of interpolation with missing lower order derivatives is called Birkhoff interpolation (cf. http://en.wikipedia.org/wiki/Birkhoff_interpolation for a survey).