Math 4543: Numerical Methods

Lecture 7 — Newton's Divided Difference Method

Syed Rifat Raiyan
Lecturer
Department of Computer Science & Engineering
Islamic University of Technology, Dhaka, Bangladesh
Email: rifatraiyan@iut-dhaka.edu

Lecture Plan

The agenda for today

- Represent interpolant polynomials using NDD
- Understand the advantage of NDD over the Direct Method
- Generalize the formula for finding the coefficients for n^{th} order interpolant
- Derive the formula for the quadratic NDD interpolant

Newton's Divided Difference Interpolation

What is it?

Newton represented the interpolant polynomial in such a manner so that the ***coefficients*** of the polynomial can be computed using the ***division of some difference values***.

As given in Figure 1, data is given at discrete points such as (x_0, y_0), (x_1, y_1), ..., (x_{n-1}, y_{n-1}), (x_n, y_n).

A continuous function $f(x)$ may be used to represent the $n+1$ data values with $f(x)$ passing through the $n+1$ points.

Then one can find the value of *y* at any other value *x*.

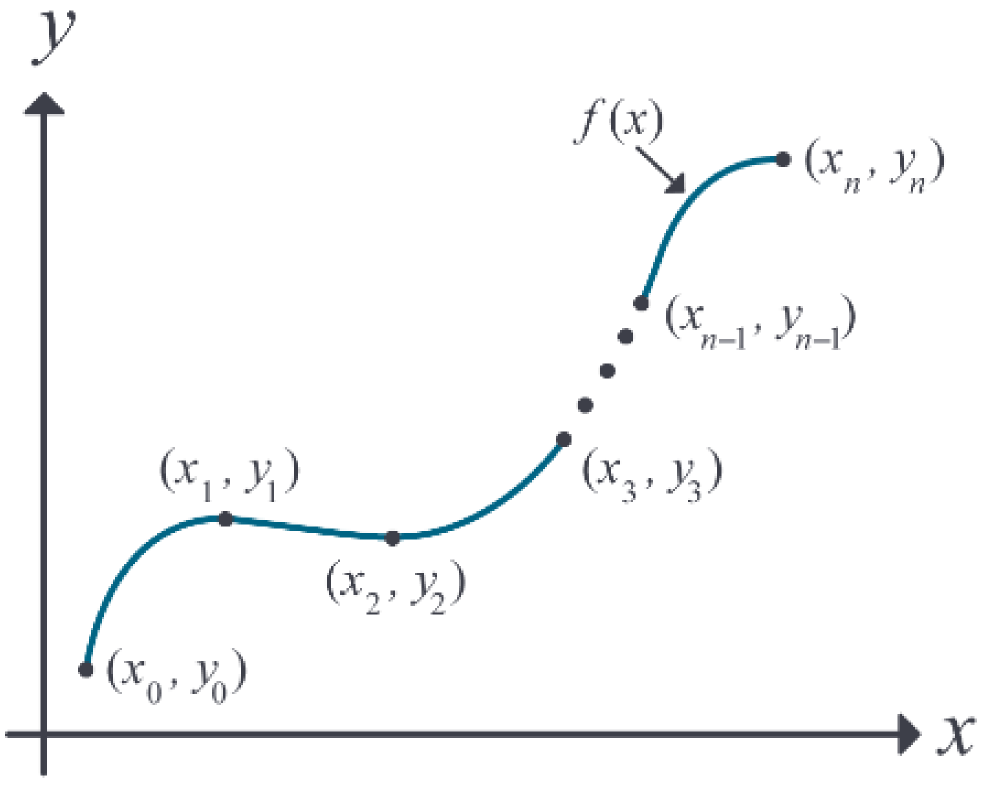

Figure 1. Interpolation of a function given at discrete points

Newton's Divided Difference Interpolation

Linear Interpolation

Given (x_0, y_0) and (x_1, y_1), fit a linear interpolant through the data. Noting $y = f(x)$ and $y_1 = f(x_1)$, assume the linear interpolant $f_1(x)$ is given by (Figure 2)

$$f_1(x) = b_0 + b_1(x - x_0)$$

Since at $x = x_0$,

$$f_1(x_0) = f(x_0) = b_0 + b_1(x_0 - x_0) = b_0$$

and at $x = x_1$,

$$\begin{aligned} f_1(x_1) = f(x_1) &= b_0 + b_1(x_1 - x_0) \\ &= f(x_0) + b_1(x_1 - x_0) \end{aligned}$$

giving

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

So

$$b_0 = f(x_0)$$

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

giving the linear interpolant as

$$f_1(x) = b_0 + b_1(x - x_0)$$

$$f_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

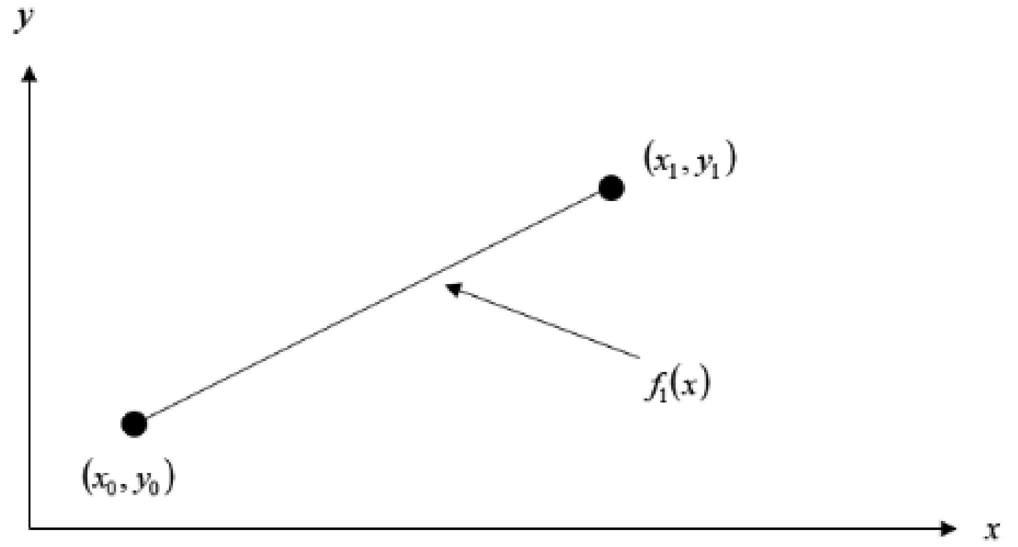

Figure 2 Linear interpolation.

Newton’s Divided Difference Interpolation

Quadratic Interpolation

Given (x_0, y_0), (x_1, y_1), and (x_2, y_2), fit a quadratic interpolant through the data. Noting $y = f(x)$, $y_0 = f(x_0)$, $y_1 = f(x_1)$, and $y_2 = f(x_2)$, assume the quadratic interpolant $f_2(x)$ is given by

$$f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$$

At $x = x_0$,

$$\begin{aligned} f_2(x_0) = f(x_0) &= b_0 + b_1(x_0 - x_0) + b_2(x_0 - x_0)(x_0 - x_1) \\ &= b_0 \end{aligned}$$

$$b_0 = f(x_0)$$

At $x = x_1$

$$f_2(x_1) = f(x_1) = b_0 + b_1(x_1 - x_0) + b_2(x_1 - x_0)(x_1 - x_1)$$

$$f(x_1) = f(x_0) + b_1(x_1 - x_0)$$

giving

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

At $x = x_2$

$$f_2(x_2) = f(x_2) = b_0 + b_1(x_2 - x_0) + b_2(x_2 - x_0)(x_2 - x_1)$$

$$f(x_2) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_2 - x_0) + b_2(x_2 - x_0)(x_2 - x_1)$$

Giving

$$b_2 = \frac{\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

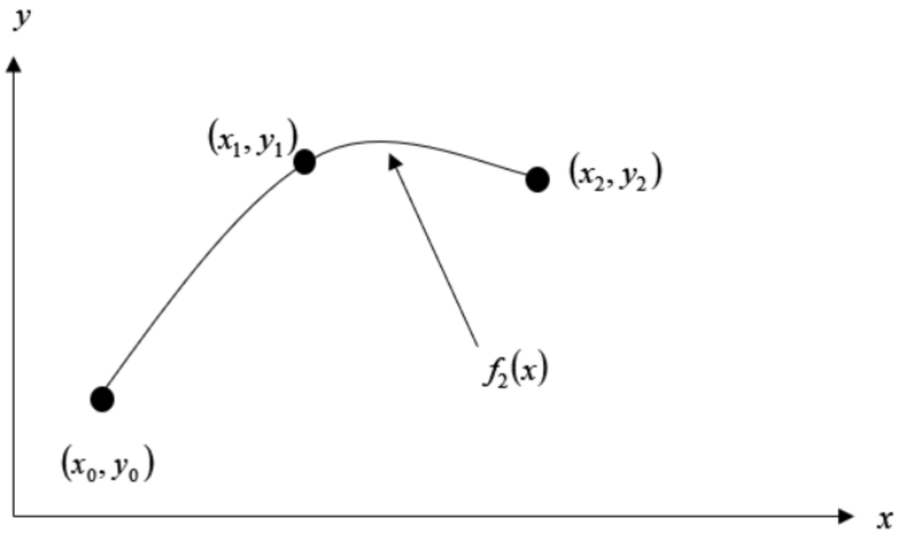

Figure 4 Quadratic interpolation.

Mini Quiz

NDD Interpolant vs Direct Method Interpolant

Do the n^{th} order interpolants obtained using **NDD** and **Direct method** differ?

If so, how do they differ?

$$\boxed{y = a_0 + a_1 x + \ldots\ldots\ldots\ldots\ldots + a_n x^n}$$

Vs.

$$\boxed{f_n(x) = b_0 + b_1(x - x_0) + \ldots + b_n(x - x_0)(x - x_1)\ldots(x - x_{n-1})}$$

Newton's Divided Difference Interpolation

Why do we need it?

The advantages are —

- ✓ Need to solve ***one equation*** to get ***one coefficient***.
- ✓ Overall time complexity to obtain the interpolant is $O(n^2)$.

For the Direct method, we needed to calculate the ***inverse of a matrix*** and simultaneously solve ***all the equations*** to obtain all the coefficients. The time complexities of the algorithms that are used to do this are,

- Naïve Gaussian Elimination — $O\left(n^3 \log\left(\left\|A\right\| + \left\|b\right\|\right)\right)$
- LU Decomposition — $O(n^3)$
- Cramer's Rule — $O((n+1)!)$

Newton's Divided Difference Interpolation

A first-order polynomial example

The upward velocity of a rocket is given as a function of time in Table 1.

Table 1. Velocity as a function of time.

t (s)	$v(t)$ (m/s)
0	0
10	227.04
15	362.78
20	517.35
22.5	602.97
30	901.67

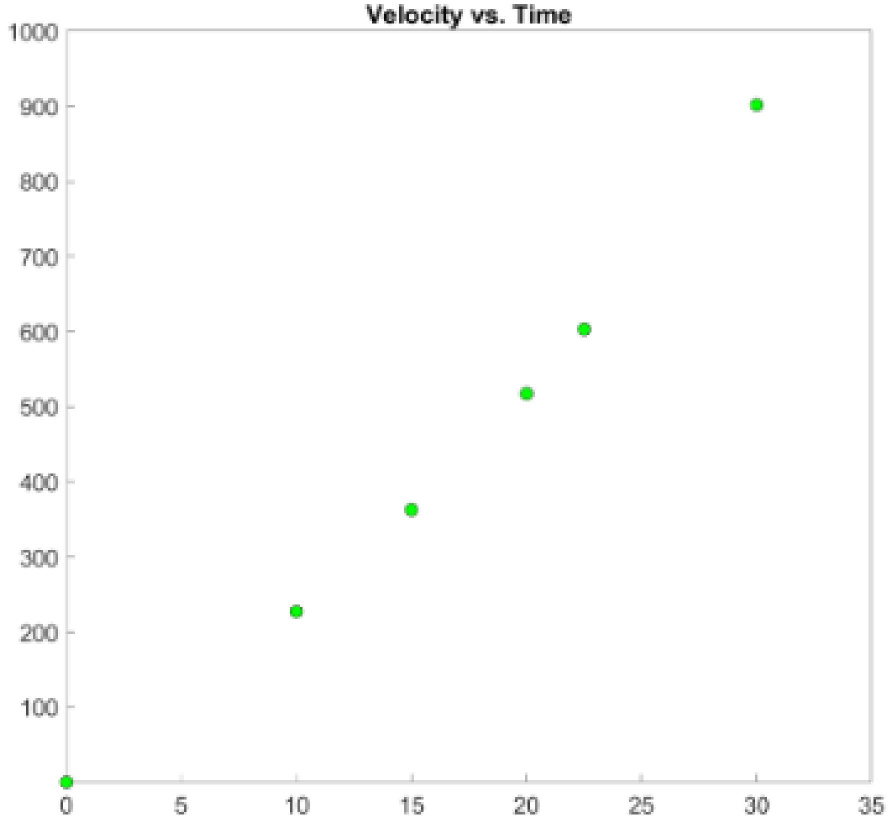

Figure 1. Graph of velocity vs. time data for the rocket example.

Determine the value of the velocity at $t = 16$ seconds using first order polynomial interpolation by Newton's divided difference polynomial method.

Newton's Divided Difference Interpolation

A first-order polynomial example

Solution

For linear interpolation, the velocity is given by

$$v(t) = b_0 + b_1(t - t_0)$$

Since we want to find the velocity at $t = 16$, and we are using a first order polynomial, we need to choose the two data points that are closest to $t = 16$ that also bracket $t = 16$ to evaluate it. The two points are $t = 15$ and $t = 20$.

Then $t_0 = 15,\ v(t_0) = 362.78$

$$t_1 = 20,\ v(t_1) = 517.35$$

$$\begin{aligned} b_0 &= v(t_0) \\ &= 362.78 \end{aligned}$$

$$\begin{aligned} b_1 &= \frac{v(t_1) - v(t_0)}{t_1 - t_0} \\ &= \frac{517.35 - 362.78}{20 - 15} \\ &= 30.914 \end{aligned}$$

$$\begin{aligned} v(t) &= b_0 + b_1(t - t_0) \\ &= 362.78 + 30.914(t - 15),\ 15 \le t \le 20 \end{aligned}$$

At $t = 16$,

$$\begin{aligned} v(16) &= 362.78 + 30.914(16 - 15) \\ &= 393.69 \text{ m/s} \end{aligned}$$

If we expand

$$v(t) = 362.78 + 30.914(t - 15),\ 15 \le t \le 20$$

we get

$$v(t) = -100.93 + 30.914t,\ 15 \le t \le 20$$

and this is the same expression as obtained in the direct method.

Newton's Divided Difference Interpolation

A second-order polynomial example

The upward velocity of a rocket is given as a function of time in Table 1.

Table 1. Velocity as a function of time.

t (s)	$v(t)$ (m/s)
0	0
10	227.04
15	362.78
20	517.35
22.5	602.97
30	901.67

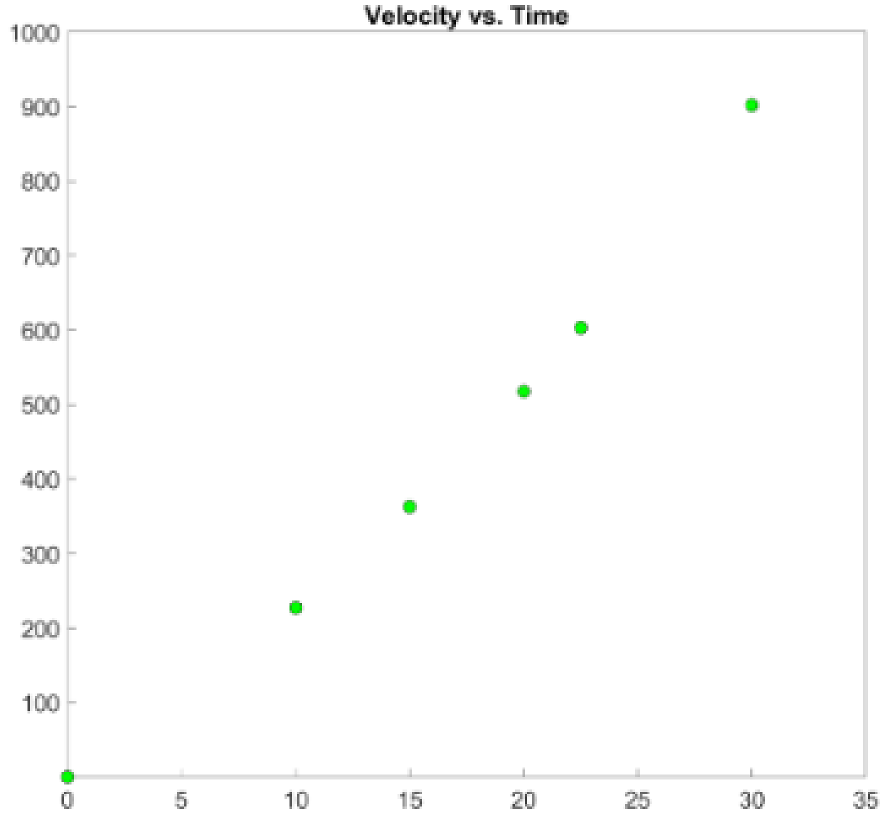

Figure 1. Graph of velocity vs. time data for the rocket example.

Determine the value of the velocity at $t = 16$ seconds using second order polynomial interpolation using Newton's divided difference polynomial method.

Newton's Divided Difference Interpolation

A second-order polynomial example

Solution

For quadratic interpolation, the velocity is given by

$$v(t) = b_0 + b_1(t - t_0) + b_2(t - t_0)(t - t_1)$$

Since we want to find the velocity at $t = 16$, and we are using a second order polynomial, we need to choose the three data points that are closest to $t = 16$ that also bracket $t = 16$ to evaluate it. The three points are $t_0 = 10$, $t_1 = 15$, and $t_2 = 20$.

$$t_0 = 10,\ v(t_0) = 227.04$$

$$t_1 = 15,\ v(t_1) = 362.78$$

$$t_2 = 20,\ v(t_2) = 517.35$$

$$\begin{aligned} b_0 &= v(t_0) \\ &= 227.04 \end{aligned}$$

$$\begin{aligned} b_1 &= \frac{v(t_1) - v(t_0)}{t_1 - t_0} \\ &= \frac{362.78 - 227.04}{15 - 10} \\ &= 27.148 \end{aligned}$$

$$\begin{aligned} b_2 &= \frac{\dfrac{v(t_2) - v(t_1)}{t_2 - t_1} - \dfrac{v(t_1) - v(t_0)}{t_1 - t_0}}{t_2 - t_0} \\ &= \frac{\dfrac{517.35 - 362.78}{20 - 15} - \dfrac{362.78 - 227.04}{15 - 10}}{20 - 10} \\ &= \frac{30.914 - 27.148}{10} \\ &= 0.37660 \end{aligned}$$

Newton's Divided Difference Interpolation

A second-order polynomial example

Hence

$$\begin{aligned} v(t) &= b_0 + b_1(t - t_0) + b_2(t - t_0)(t - t_1) \\ &= 227.04 + 27.148(t - 10) + 0.37660(t - 10)(t - 15),\ 10 \le t \le 20 \end{aligned}$$

At $t = 16$,

$$\begin{aligned} v(16) &= 227.04 + 27.148(16 - 10) + 0.37660(16 - 10)(16 - 15) \\ &= 392.19 \text{ m/s} \end{aligned}$$

If we expand

$$v(t) = 227.04 + 27.148(t - 10) + 0.37660(t - 10)(t - 15),\ 10 \le t \le 20$$

we get

$$v(t) = 12.05 + 17.733t + 0.37660t^2,\ 10 \le t \le 20$$

This is the same expression obtained by the direct method.

Newton's Divided Difference Interpolation

The general form

Note that b_0, b_1, and b_2 are finite divided differences. b_0, b_1, and b_2 are the first, second, and third finite divided differences, respectively. We denote the first divided difference by

$$f[x_0] = f(x_0)$$

the second divided difference by

$$f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

and the third divided difference by

$$f[x_2, x_1, x_0] = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0}$$

where $f[x_0]$, $f[x_1, x_0]$, and $f[x_2, x_1, x_0]$ are called bracketed functions of their variables enclosed in square brackets.

Rewriting,

$$f_2(x) = f[x_0] + f[x_1, x_0](x - x_0) + f[x_2, x_1, x_0](x - x_0)(x - x_1)$$

This leads us to writing the general form of the Newton's divided difference polynomial for $n + 1$ data points,
$(x_0, y_0), (x_1, y_1), \ldots\ldots, (x_{n-1}, y_{n-1}), (x_n, y_n)$, as

$$f_n(x) = b_0 + b_1(x - x_0) + \ldots + b_n(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

Newton's Divided Difference Interpolation

The general form

where

$$b_0 = f[x_0]$$

$$b_1 = f[x_1, x_0]$$

$$b_2 = f[x_2, x_1, x_0]$$

$$\vdots$$

$$b_{n-1} = f[x_{n-1}, x_{n-2}, \ldots, x_0]$$

$$b_n = f[x_n, x_{n-1}, \ldots, x_0]$$

where the definition of the m^{th} divided difference is

$$\begin{aligned} b_m &= f[x_m, \ldots\ldots, x_0] \\ &= \frac{f[x_m, \ldots\ldots, x_1] - f[x_{m-1}, \ldots\ldots, x_0]}{x_m - x_0} \end{aligned}$$

Mini Quiz

Strategy for the implementation of the NDD method

What can be an ***optimal approach*** to write a computer program that calculates all the divided differences? Justify your choice.

Newton's Divided Difference Interpolation

The general form

$$b_0 = f[x_0]$$

$$b_1 = f[x_1, x_0]$$

$$b_2 = f[x_2, x_1, x_0]$$

$$\vdots$$

$$b_{n-1} = f[x_{n-1}, x_{n-2}, \ldots, x_0]$$

$$b_n = f[x_n, x_{n-1}, \ldots, x_0]$$

where the definition of the m^{th} divided difference is

$$b_m = f[x_m, \ldots\ldots, x_0] = \frac{f[x_m, \ldots\ldots, x_1] - f[x_{m-1}, \ldots\ldots, x_0]}{x_m - x_0}$$

From the above definition, it can be seen that the divided differences are calculated recursively.

For an example of a third order polynomial, given (x_0, y_0), (x_1, y_1), (x_2, y_2), and (x_3, y_3),

$$f_3(x) = f[x_0] + f[x_1, x_0](x - x_0) + f[x_2, x_1, x_0](x - x_0)(x - x_1) + f[x_3, x_2, x_1, x_0](x - x_0)(x - x_1)(x - x_2)$$

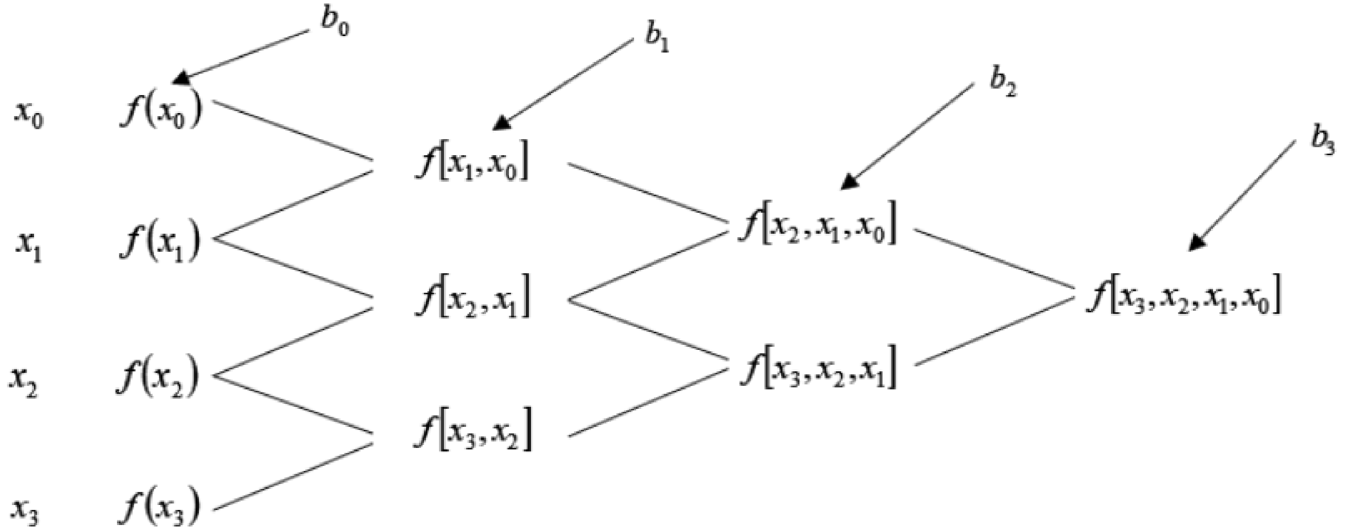

Figure 5 Table of divided differences for a cubic polynomial.

Newton's Divided Difference Interpolation

Deriving the 2nd order NDD polynomial

Given (x_0, y_0), (x_1, y_1), and (x_2, y_2), pass a quadratic interpolant through the data. Noting $y = f(x)$, $y_0 = f(x_0)$, $y_1 = f(x_1)$, and $y_2 = f(x_2)$, assume the quadratic interpolant $f_2(x)$ is given by

$$f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) \qquad (1)$$

At $x = x_2$

$$f_2(x_2) = f(x_2) = b_0 + b_1(x_2 - x_0) + b_2(x_2 - x_0)(x_2 - x_1)$$

$$f(x_2) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_2 - x_0) + b_2(x_2 - x_0)(x_2 - x_1)$$

$$b_2 = \frac{f(x_2) - f(x_0) - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)}$$

$$= \frac{\dfrac{f(x_2) - f(x_0)}{x_2 - x_0} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_1}$$

Newton's Divided Difference Interpolation

Deriving the 2nd order NDD polynomial

But if we want to write this in the form where $(x_2 - x_0)$ is in the denominator so as to express it in the divided difference form of $f[x_2, x_1, x_0]$, we need to do the following manipulations.

Add 0 in the form of $\{-f(x_1) + f(x_1)\}$ to the numerator of equation (4)

$$b_2 = \frac{f(x_2) + \{-f(x_1) + f(x_1)\} - f(x_0) - \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)}$$

Collecting $\{f(x_1) - f(x_0)\}$ terms together

$$b_2 = \frac{f(x_2) - f(x_1) + \{f(x_1) - f(x_0)\}(1 - \frac{x_2 - x_0}{x_1 - x_0})}{(x_2 - x_0)(x_2 - x_1)}$$

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

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

Dividing the numerator and denominator by $(x_2 - x_1)$

$$b_2 = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} + \frac{\{f(x_1) - f(x_0)\}(x_1 - x_2)}{(x_1 - x_0)(x_2 - x_1)}}{x_2 - x_0}$$

$$\boxed{\begin{aligned} &= \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0} \\ &= f[x_2, x_1, x_0] \end{aligned}}$$