Interpolation

Problem: Given a set of measured data, say $n + 1$ pairs,

$$(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)$$

the problem of interpolation is to find a function $f(x)$ such that

$$f(x_i) = y_i, \quad i = 0, 1, ..., n$$

- x_i is called nodes;
- $f(x)$ is said to interpolate the data and is called interpolation function.
- $f(x)$ is said to approximate $g(x)$ if the data are from a function $g(x)$.
- It is called interpolate (or extrapolate) if $f(x)$ gives values within (or outside) $[x_0, x_n]$.

A simple choice for $f(x)$ is a polynomial of degree n:

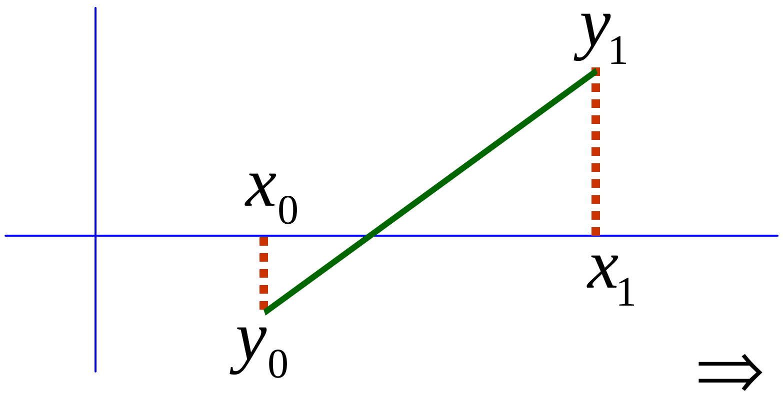
$$f(x) = p_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

The existence and uniqueness have been verified. There is only one polynomial existing for the interpolation.

LAGRANGIAN POLYNOMIALS

(a) Fitting Two Points

Fit the linear polynomial for two given points (x_0, y_0) and (x_1, y_1).



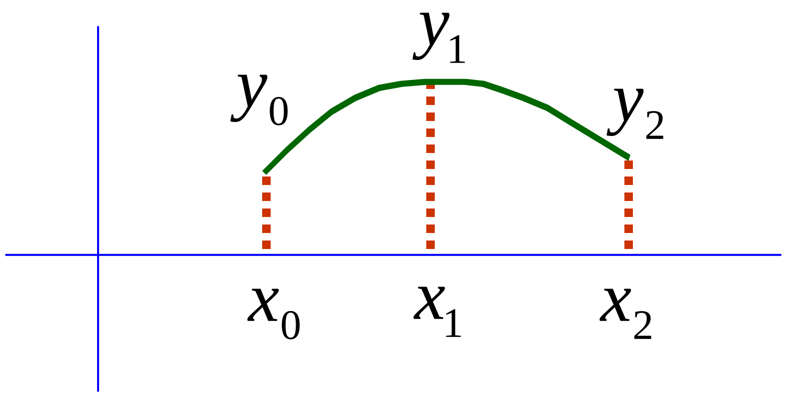
$$p_1(x) = L_0(x) y_0 + L_1(x) y_1$$

$$= \left(\frac{x - x_1}{x_0 - x_1} \right) y_0 + \left(\frac{x - x_0}{x_1 - x_0} \right) y_1$$

$$\Rightarrow \quad p_1(x_0) = y_0, \quad p_1(x_1) = y_1$$

(b) Fitting Three Points

Fit the quadratic polynomial for three given points.



$$\Rightarrow \quad \begin{aligned} p_2(x_0) &= y_0 \\ p_2(x_1) &= y_1 \\ p_2(x_2) &= y_2 \end{aligned}$$

$$p_2(x) = \left(\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}\right)y_0 + \left(\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}\right)y_1 + \left(\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}\right)y_2$$

$$= L_0(x)y_0 + L_1(x)y_1 + L_2(x)y_2$$

$$L_0(x) = \begin{cases} 1, & x = x_0 \\ 0, & x = x_1 \\ 0, & x = x_2 \end{cases} \qquad L_1(x) = \begin{cases} 0, & x = x_0 \\ 1, & x = x_1 \\ 0, & x = x_2 \end{cases} \qquad L_2(x) = \begin{cases} 0, & x = x_0 \\ 0, & x = x_1 \\ 1, & x = x_2 \end{cases}$$

(c) Fitting n+1 Points

Lagrangian polynomial for fitting $n+1$ given points

$$(x_0, y_0), (x_1, y_1), \ldots, \ (x_n, y_n).$$

is given by

$$p_n(x) = \sum_{k=0}^{n} L_k(x) y_k = \sum_{k=0}^{n} \frac{l_k(x)}{l_k(x_k)} y_k$$

where

$$l_k(x) = \frac{1}{x - x_k}(x - x_0)(x - x_1)(x - x_2)\cdots(x - x_n);$$

$$L_k(x) = \begin{cases} 1, & x = x_k \\ 0, & x = x_j, \quad j \neq k \end{cases}$$

NEWTON'S DIVIDED DIFFERENCE METHOD

The following two disadvantages of Lagrangian polynomial method lead us to develop a new method for the interpolation. They are:

(1) it involves more arithmetic operations; and

(2) we essentially need to start over the computation if we desire to add or subtract a point from the set of data.

The Basic Idea of Divided Difference:

Consider the n-th-degree polynomial written in a special way:

$$P_n(x) = a_0 + (x - x_0)a_1 + (x - x_0)(x - x_1)a_2 + \dots + (x - x_0)(x - x_1)\dots(x - x_{n-1})a_n.$$

The key idea is to find $a_0, \dots, a_n$ so that P_n interpolates the given data:

$$(x_0, f_0), (x_1, f_1), \dots, (x_n, f_n).$$

n

Define the first order divided difference between two notes x_i and x_{i+1} as

$$f[x_i, x_{i+1}] = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} = f_i^{[1]} = f[x_{i+1}, x_i]$$

the second order divided difference as

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i} = f_i^{[2]}$$

and the higher order divided differences as

$$f[x_i, x_{i+1}, \cdots x_{i+m}] = \frac{f[x_{i+1}, \cdots, x_{i+m}] - f[x_i, \cdots, x_{i+m-1}]}{x_{i+m} - x_i} = f_i^{[m]}$$

as well as zero-th order divided difference:

$$f[x_i] = f_i = f_i^{[0]}$$

Divided Difference Table:

x_i	f_i	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
x_0	f_0			
		$f_0^{[1]}$		
x_1	f_1		$f_0^{[2]}$	
		$f_1^{[1]}$		$f_0^{[3]}$
x_2	f_2		$f_1^{[2]}$	
		$f_2^{[1]}$		$f_1^{[3]}$
x_3	f_3		$f_2^{[2]}$	
		$f_3^{[1]}$		
x_4	f_4			

Example:

x_i	f_i	1st Order	2nd Order	3rd Order	4th Order
3.2	22.0				
		8.400			
2.7	17.8		2.856		
		2.118		-0.5280	
1.0	14.2		2.012		
		6.342		0.0865	
4.8	38.3		2.263		
		16.750			
5.6	51.7				

Idea: If

$$P_n(x) = a_0 + (x - x_0)a_1 + (x - x_0)(x - x_1)a_2 + \ldots$$
$$+ (x - x_0)(x - x_1)\ldots(x - x_{n-1})a_n.$$

is an interpolation of the given data:

$$(x_0, f_0), (x_1, f_1), \ldots, (x_n, f_n),$$

then we have

$$P_n(x_i) = f_i, \quad i = 0, 1, \cdots, n$$

Thus

$$P_n(x_0) = a_0 + (x_0 - x_0)a_1 + \cdots + (x_0 - x_0)\cdots(x - x_{n-1})a_n = a_0 = f_0 = f_0^{[0]}$$

$$P_n(x_1) = a_0 + (x_1 - x_0)a_1 = f_0 + (x_1 - x_0)a_1 = f_1 \quad \Rightarrow a_1 = \frac{f_1 - f_0}{x_1 - x_0} = f_0^{[1]}$$

In general, we can show that $\boxed{a_k = f_0^{[k]}, \quad k = 1, 2, \cdots, n}$

Thus, given a set of data:

$$(x_0, f_0), (x_1, f_1), \ldots, (x_n, f_n),$$

their n-th degree polynomial interpolation is given by

$$P_n(x) = f_0^{[0]} + (x - x_0) f_0^{[1]} + (x - x_0)(x - x_1) f_0^{[2]} + \cdots + (x - x_0)(x - x_1)\cdots(x - x_{n-1}) f_0^{[n]}.$$

The advantage of the above method is that there is no need to start all over again if their additional pairs of data are added. We simply need to compute additional divided differences.

Since n-th order polynomial interpolation of a given $(n + 1)$ pairs of data is unique, thus the above polynomial and Lagrangian polynomial are exactly the same.

Example: Interpolate the following set of data

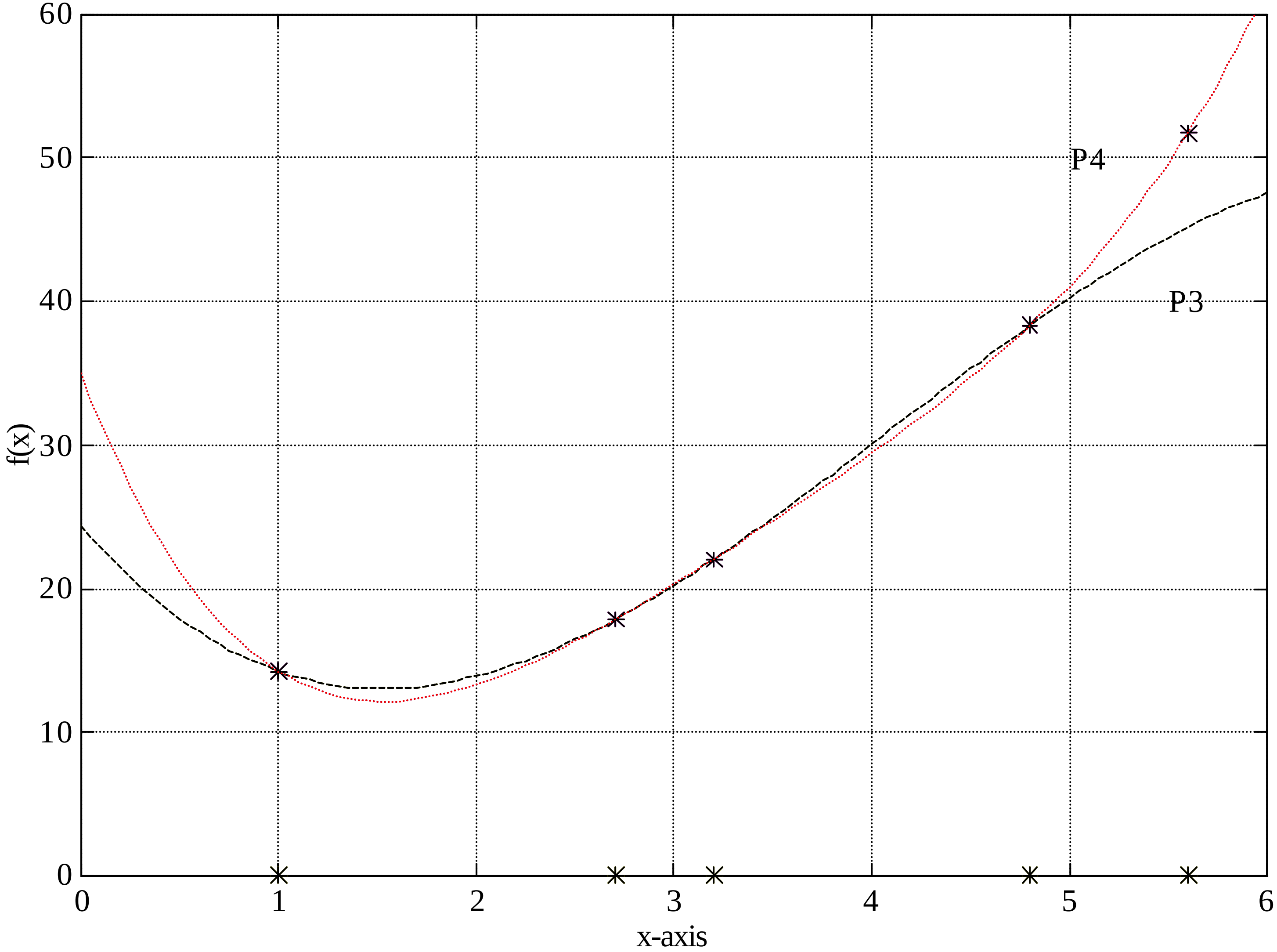
x_i	f_i	1st Order	2nd Order	3rd Order	4th Order
3.2	22.0				
		8.400			
2.7	17.8		2.856		
		2.118		−0.5280	
1.0	14.2		2.012		0.2560
		6.342		0.0865	
4.8	38.3		2.263		
		16.750			
5.6	51.7				

Interpolate from x_0 to x_3:

$$P_3(x) = 22.0 + 8.400(x-3.2) + 2.856(x-3.2)(x-2.7) - 0.528(x-3.2)(x-2.7)(x-1.0)$$

Interpolate from x_0 to x_4:

$$P_4(x) = P_3(x) + 0.256(x-3.2)(x-2.7)(x-1.0)(x-4.8)$$


60
50
40
30
20
10
0
f(x)
0
1
2
3
4
5
6
x-axis
P4
P3

Evenly Spaced Data

The problem of interpolation from tabulated data is considerably simplified if the values of the function are given at evenly spaced intervals of the independent variable.

Difference Table:

Assume that the given set of data is evenly spaced, i.e.,

$$x_{i+1} - x_i = h$$

(a) The first order differences of the functions are defined as:

$$\begin{aligned} \Delta f_0 &= f_1 - f_0 \quad \text{at} \quad x_0 \\ \Delta f_1 &= f_2 - f_1 \quad \text{at} \quad x_1 \\ &\vdots \\ \Delta f_i &= f_{i+1} - f_i \quad \text{at} \quad x_i \end{aligned}$$

(b) The second order differences of the functions are given by:

$$\Delta^2 f_0 = \Delta(\Delta f_0) = \Delta f_1 - \Delta f_0 \qquad \text{at } x_0$$

$$\Delta^2 f_1 = \Delta(\Delta f_1) = \Delta f_2 - \Delta f_1 \qquad \text{at } x_1$$

$$\vdots$$

$$\Delta^2 f_i = \Delta(\Delta f_1) = \Delta f_{i+1} - \Delta f_i \qquad \text{at } x_i$$

(c) The n-th order differences of the functions are given by:

$$\boxed{\Delta^n f_i = \Delta^{n-1} f_{i+1} - \Delta^{n-1} f_i}$$

Newton Forward Method for Evenly Spaced Data:

Given a set of measured data

$$(x_0, f_0), (x_1, f_1), \ldots, \ (x_n, f_n),$$

in which $x_{i+1} - x_i = h$, then the Newton forward interpolation polynomial is

given by

$$P_n(x) = f_0 + \binom{s}{1}\Delta f_0 + \binom{s}{2}\Delta^2 f_0 + \cdots + \binom{s}{n}\Delta^n f_0$$

where

$$s = \frac{x - x_0}{h} \qquad \text{and} \qquad \binom{s}{k} = \frac{s(s-1)\cdots(s-k+1)}{k!}, \quad k = 1, 2, \cdots, n$$

and $\Delta^k f_0$ is the *k*-th order difference of the given data.

Example: Interpolate the following set of data using Newton Forward Method

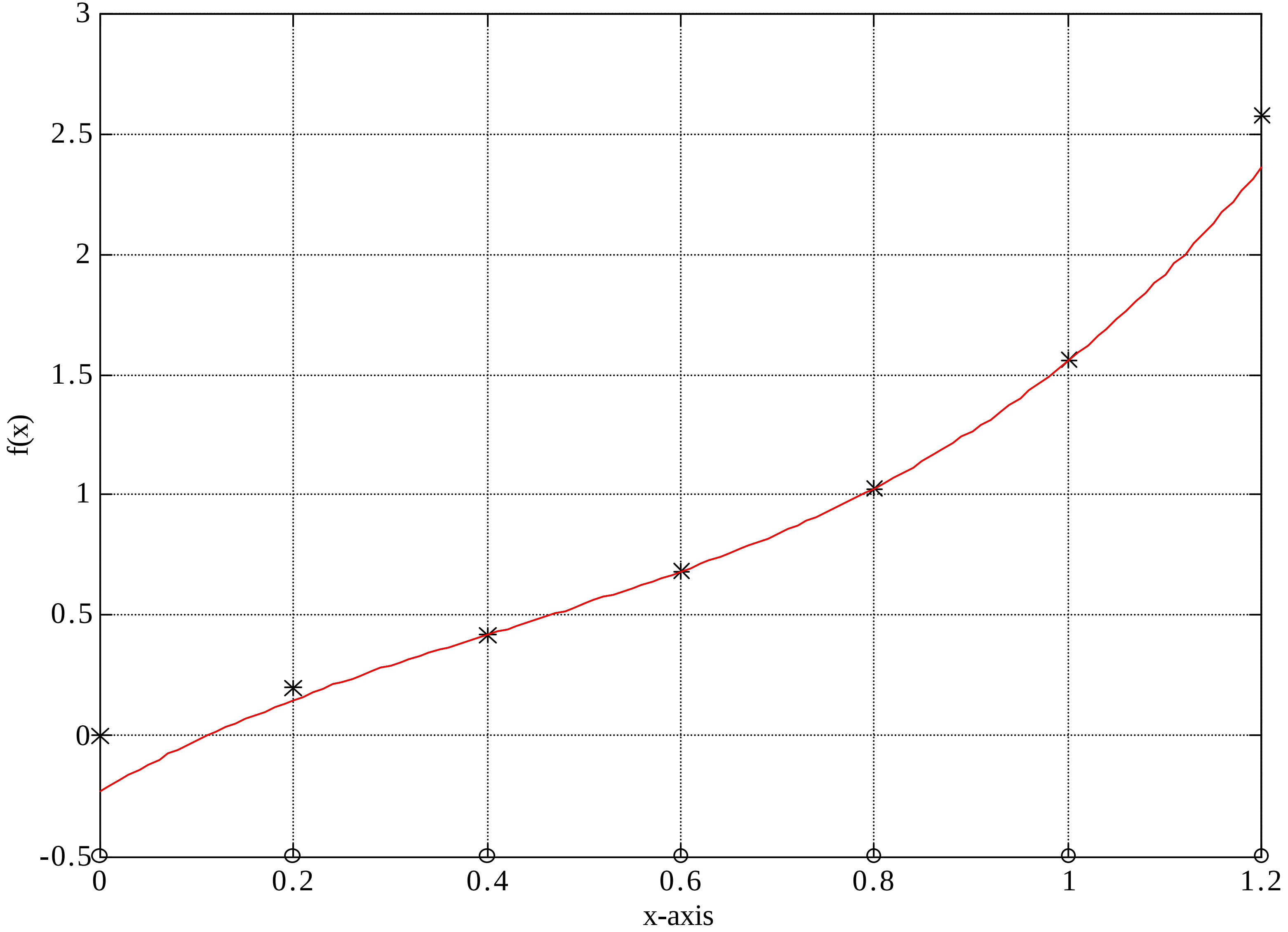
x_i	f_i	1st Order	2nd Order	3rd Order	4th Order
0.0	0.000				
		0.203			
0.2	0.203		0.017		
		0.220		0.024	
0.4	0.423		0.041		0.020
		0.261		0.044	
0.6	0.684		0.085		0.052
		0.346		0.096	
0.8	1.030		0.181		0.211
		0.527		0.307	
1.0	1.557		0.488		
		1.015			
1.2	2.572				

Assume that we only want to interpolate from 0.4 (x_0) to 1.0 (x_3):

$$P_3(x) = 0.423 + 0.261\binom{s}{1} + 0.085\binom{s}{2} + 0.096\binom{s}{3},$$

$$s = \frac{x - 0.4}{0.2} = 5x - 2$$

n


3
2.5
2
1.5
1
0.5
0
-0.5
0
0.2
0.4
0.6
0.8
1
1.2
f(x)
x-axis