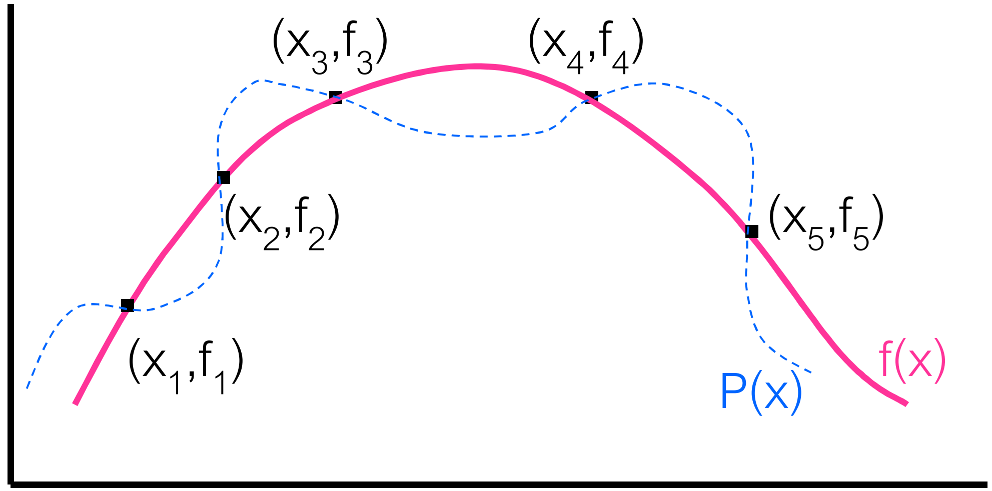
Interpolation



(x3,f3)
(x4,f4)
(x2,f2)
(x5,f5)
(x1,f1)
f(x)
P(x)

Introduction

- The interpolation problem
  - Given values of an unknown function $f(x)$ at values $x = x_0, x_1, \ldots, x_n$, find approximate values of $f(x)$ between these given values
- Polynomial interpolation
  - Find nth-order polynomial $p_n(x)$ that approximates the function $f(x)$ and provides exact agreement at the n node points:

$$p_n(x_0) = f(x_0), \quad p_n(x_1) = f(x_1), \quad \cdots \quad p_n(x_n) = f(x_n)$$

  - The polynomial $p_n(x)$ is unique
  - Interpolation: evaluate $p_n(x)$ for $x_0 <= x <= x_n$
  - Extrapolation: evaluate $p_n(x)$ for $x_0 > x > x_n$

Purposes for Interpolation

- Plotting smooth curve through discrete data points
- Quick and easy evaluation of mathematical function
- Replacing "difficult" function by "easy" one
- "Reading between the lines" of table
- Differentiating or integrating tabular data

Functions for Interpolation

Some families of functions commonly used for interpolation include

- Polynomials
- Piecewise polynomials
- Trigonometric functions
- Exponentials
- Rational functions

We focus on interpolation by polynomials and piecewise polynomials.

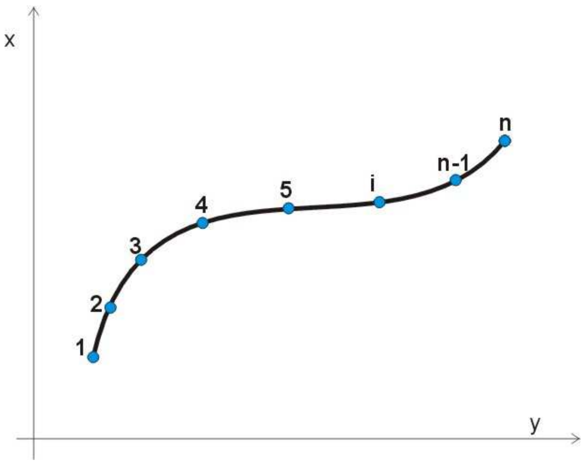
Polynomial Interpolation

$$P_m(x) = a_1 u_1(x) + a_2 u_2(x) + \ldots + a_m u_m(x)$$

$$\mathbf{V}\ \mathbf{A} = \mathbf{Y} \quad (1)$$

Vandermonde matrix

$$\mathbf{V} = \begin{bmatrix} u_1(x_1) & u_2(x_1) & \ldots & u_n(x_1) \\ u_1(x_2) & u_2(x_2) & \ldots & u_n(x_2) \\ \ldots & \ldots & \ldots & \ldots \\ u_1(x_n) & u_2(x_n) & \ldots & u_n(x_n) \end{bmatrix}$$



n=m

$$P_m(x_i) = y_i\,, \qquad i = 1,2,\ldots,m$$

$$\mathbf{A} = \begin{bmatrix} a_1 \\ a_2 \\ \ldots \\ a_n \end{bmatrix}, \qquad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{bmatrix}$$

Extrapolation

$$\mathbf{A} = \mathbf{V}^{-1}\mathbf{Y}$$

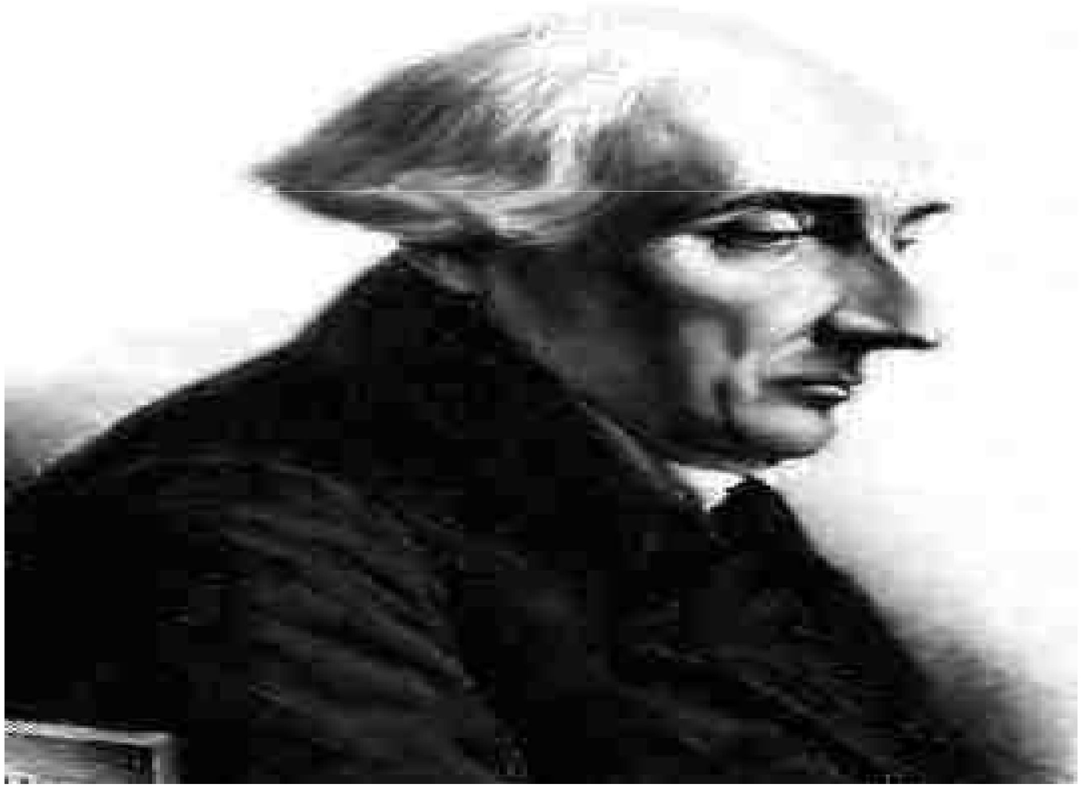
Polynomial Interpolation

Simplest and commonest type of interpolation uses polynomials.

Unique polynomial of degree at most $n - 1$ passes through n data points (t_i, y_i), $i = 1, \ldots, n$, where t_i are distinct.

There are many ways to represent or compute polynomial, but in theory all must give same result.

Lagrange



Lagrange Interpolation

For given set of data points (t_i, y_i), $i = 1, \ldots, n$,

Lagrange basis functions are given by

$$l_j(x) = \frac{(x - x_1)(x - x_2)\ldots(x - x_{j-1})(x - x_{j+1})\ldots(x - x_n)}{(x_j - x_1)(x_j - x_2)\ldots(x_j - x_{j-1})(x_j - x_{j+1})\ldots(x_j - x_n)}$$

$$= \pi_{k=1,k=j}^{n} \frac{(x - x_k)}{(x_j - x_k)}$$

which means that matrix of linear system $Ax = y$ is identity.

Thus, Lagrange polynomial interpolating data points (t_i, y_i) is given by

$$p_{n-1}(t) = y_1 l_1(t) + y_2 l_2(t) + \ldots + y_n l_n(t).$$

Examples: Lagrange Interpolation

1-Use Lagrange interpolation to find interpolating polynomial for three data points (–2, –27), (0, –1), 1, 0).

Lagrange polynomial of degree two interpolating three points

$(t_1, y_1), (t_2, y_2), (t_3, y_3)$ is

$$p_2(t) = y_1 \frac{(t-t_2)(t-t_3)}{(t_1-t_2)(t_1-t_3)} + y_2 \frac{(t-t_1)(t-t_3)}{(t_2-t_1)(t_2-t_3)} + y_3 \frac{(t-t_1)(t-t_2)}{(t_3-t_1)(t_3-t_2)}.$$

For this particular set of data, this becomes

$$p_2(t) = -27 \frac{t(t-1)}{(-2)(-2-1)} + (-1) \frac{(t+2)(t-1)}{(2)(-1)}.$$

2-Use Lagrange interpolation to find interpolating polynomial for

$$f(x) = \sin(x).$$

X_j	f_j	sin f(x)
$X_1 = 0$	Sin(0)	0
$X_2 = 1$	Sin(1)	0.8415
$X_3 = 2$	Sin(2)	0.9093

$$l_1(x) = \frac{(x-1)(x-2)}{(0-1)(0-2)} = \frac{(x-1)(x-2)}{2} \quad , x_1 = 0. \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}$$

$$l_2(x) = \frac{(x-0)(x-2)}{(1-0)(1-2)} = -x(x-2) \quad , x_2 = 1. \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}$$

$$l_3(x) = \frac{(x-0)(x-1)}{(2-0)(2-1)} = \frac{x(x-1)}{2} \quad , x_3 = 2. \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}$$

$$\therefore p_2(x) = \sin(0).\frac{(x-1)(x-2)}{2} - \sin(1).x(x-2) + \sin(2).\frac{x(x-1)}{2}$$

- x = 1.5 **p2(1.5) = 0.9721**
- sin(1.5) = 0.9975

Error in Interpolating a Function

If interpolation points are discrete sample of underlying continuous function, then we may want to know how closely interpolant approximates given function between sample points.

If f is sufficiently smooth function, and p_{n-1} is unique polynomial of degree at most $n - 1$ that interpolates f at n points $t_1, \ldots, t_n$, then

$$f(t) - p_{n-1}(t) = \frac{f^{(n)}(\theta)}{n!}(t - t_1)(t - t_2)\cdots(t - t_n),$$

where θ is some (unknown) point in interval $[t_1, t_n]$.

Since point θ is unknown, this result is not particularly useful unless we have bound on appropriate derivative of f, but it still provides some insight into factors affecting accuracy of polynomial interpolation.

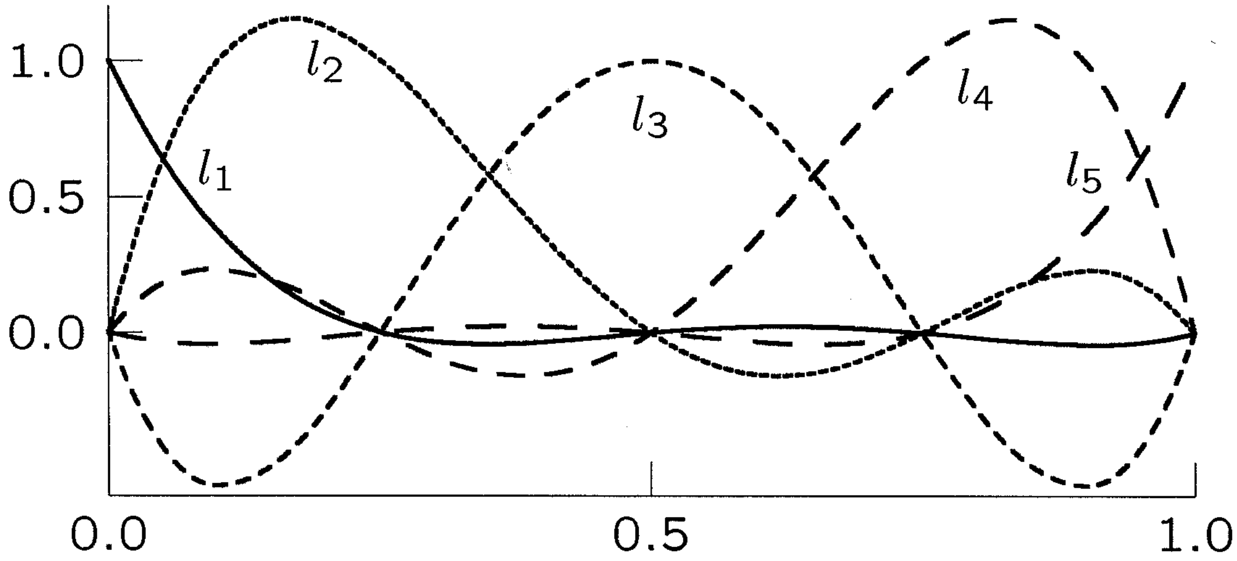
Error in Interpolating a Function

- Take $f(x) = e^x$ on $[0\ .1]$ and consider the error in linear interpolation to $f(x)$ using nodes x_0 and x_1 satisfying $0 \le x_0 \le x_1 \le 1$
- We have: $$e^x - P_1(x) = \frac{(x - x_0)(x - x_1)}{2} e^{\theta}$$
- 1- $$\underset{x_0 \le x \le x_1}{Max} \frac{(x - x_0)(x - x_1)}{2} = \frac{h^2}{8} \quad , h = x_1 - x_0$$
- 2- $$\underset{x_0 \le x \le x_1}{Max} e^{\theta} = e$$
- We conclude $$\left| e^x - P_1(x) \right| = \frac{h^2}{8} e$$

Error in Interpolating a Function

- Take $f(x) = e^x$ on $[0\ .1]$ and consider the error in quadratic interpolation to $f(x)$ using nodes x_0, x_1 and x_2 satisfying $0 \le x_0 \le x_1 \le x_2 \le 1$
- We have: $$e^x - P_2(x) = \frac{(x-x_0)(x-x_1)(x-x_2)}{3!} e^\theta$$
- 1- $$\underset{x_0 \le x \le x_1}{Max} \frac{(x - x_0)(x - x_1)(x - x_2)}{6} = \frac{h^3}{9\sqrt{3}}$$ $$h = x_2 - x_1 = x_1 - x_0$$
- 2- $$\underset{x_0 \le x \le x_1}{Max} e^\theta = e$$
- We conclude $$\left|e^x - P_2(x)\right| = \frac{h^3}{9\sqrt{3}} e = 0.174\, h^3$$

Lagrange Basis Functions



Lagrange interpolant is easy to determine but more expensive to evaluate for given argument, compared with monomial basis representation.

Lagrangian form is also more difficult to differentiate, integrate, etc.

Review and Discussion

- In Lagrange interpolation polynomial, it always go through given points.

  Think with equation below

$$p(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} y_1 + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} y_2 + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} y_3$$

The Lagrange form of polynomial is convenient when the same abscissas may occur in different applications.

- It is less convenient than the Newton form when additional data points may be added to the problem.

Newton

Newton Interpolation Polynomials

- Newton form of the equation of a straight line passing through *two* points (x_1, y_1) and (x_2, y_2) is

$$p(x) = a_1 + a_2(x - x_1)$$

- Newton form of the equation of a parabola passing through *three* points (x_1, y_1), (x_2, y_2), and (x_3, y_3) is

$$p(x) = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2)$$

- the general form of the polynomial passing through *n* points $(x_1, y_1), \ldots, (x_n, y_n)$ is

$$\boxed{p(x) = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2) + \ldots + a_n(x - x_1)\ldots(x - x_{n-1})}$$

Newton Interpolation Polynomials (cont'd)

$$p(x) = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2) + \ldots + a_n(x - x_1)\ldots(x - x_{n-1})$$

- Substituting (x_1, y_1) into $y = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2)$

$$a_1 = y_1$$

- Substituting (x_2, y_2) into $y = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2)$

$$a_2 = \frac{y_2 - y_1}{x_2 - x_1}$$

- Substituting (x_3, y_3) into $y = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2)$

$$a_3 = \frac{\dfrac{y_3 - y_2}{x_3 - x_2} - \dfrac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_1}$$

Newton Divided Difference Interpolation

- Divided difference representation

$$a_n = \frac{f(x_n) - p_{n-1}(x_n)}{(x_n - x_0)(x_n - x_1)\cdots(x_n - x_{n-1})}$$

$$a_1 = f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad a_2 = f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

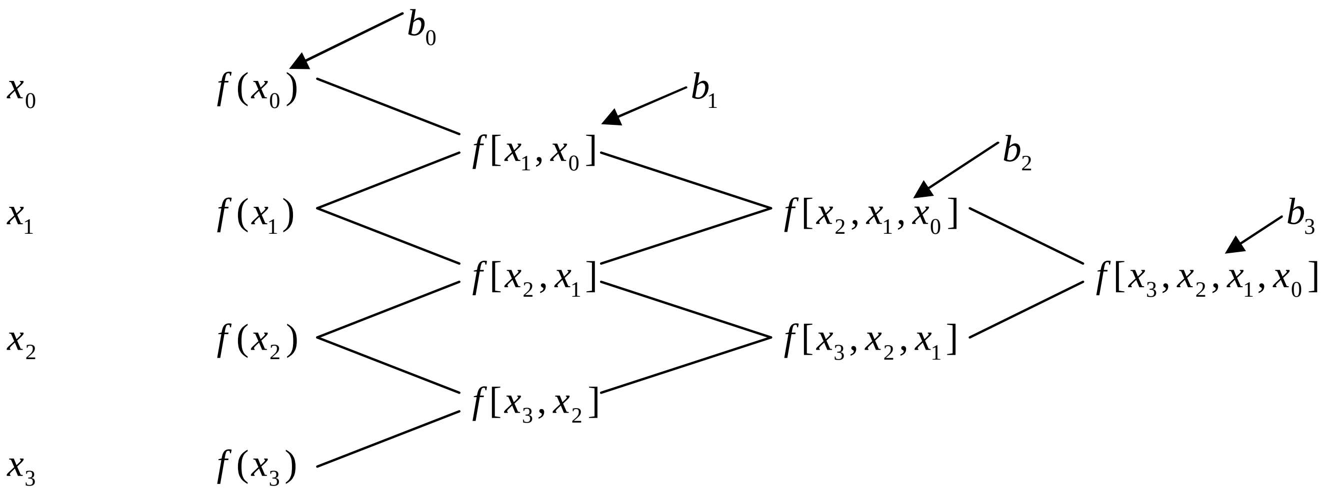
$$a_k = f[x_0, \ldots, x_k] = \frac{f[x_1, \ldots, x_k] - f[x_0, \ldots, x_{k-1}]}{x_k - x_0}$$

$$f(x) \approx f(x_0) + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)\cdots(x - x_{n-1})$$

General form

The third order polynomial, given (x_0, y_0), (x_1, y_1), (x_2, y_2), and (x_3, y_3), is

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



Newton Interpolation

- Passing through the points $(x_1, y_1)=(-2, 4)$, $(x_2, y_2)=(0, 2)$, and $(x_3, y_3)=(2, 8)$.
- The equations is $p(x) = a_1 + a_2(x-(-2)) + a_3(x-(-2))(x-0)$

- Where the coefficients are

$$a_1 = y_1 = 4$$

$$a_2 = \frac{y_2 - y_1}{x_2 - x_1} = \frac{2-4}{0-(-2)} = -1$$

$$a_3 = \frac{\dfrac{y_3 - y_2}{x_3 - x_2} - \dfrac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_1} = \frac{\dfrac{8-2}{2-0} - \dfrac{2-4}{0-(-2)}}{2-(-2)} = 1$$

- thus

$$\boxed{p(x) = 4 - (x+2) + x(x+2) = x^2 + x + 2}$$

Newton Interpolation

- Passing through the points $(x_1, y_1)=(-2, 4)$, $(x_2, y_2)=(0, 2)$, and $(x_3, y_3)=(2, 8)$.

$$p(x) = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2)$$

x_i	y_i	$d_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$	$dd_i = \frac{d_{i+1} - d_i}{x_{i+2} - x_i}$
-2	$\boxed{4}$		
		$\frac{y_2 - y_1}{x_2 - x_1} = \frac{2 - 4}{0 - (-2)} = \boxed{-1}$	
0	2		$\frac{d_2 - d_1}{x_3 - x_1} = \frac{3 - (-1)}{2 - (-2)} = \boxed{1}$
		$\frac{y_3 - y_2}{x_3 - x_2} = \frac{8 - 2}{2 - 0} = 3$	
2	8		

$$\boxed{p(x) = 4 - (x+2) + x(x+2) = x^2 + x + 2}$$

Additional Data Points

- We extend the previous example, adding the points $(x_4, y_4) = (-1, -1)$ and $(x_5, y_5) = (1, 1)$

$$p(x) = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2) + a_4(x - x_1)(x - x_2)(x - x_3)$$
$$+ a_5(x - x_1)(x - x_2)(x - x_3)(x - x_4).$$

- Divided-difference table becomes (with new entries shown in bold)

x_i	y_i	d_i	dd_i	ddd_i	$dddd_i$
-2	$\boxed{4}$				
		$\frac{(2-4)}{(0+2)} = \boxed{-1}$			
0	2		$\frac{(3+1)}{(2+2)} = \boxed{1}$		
		$\frac{(8-2)}{(2-0)} = 3$		$\frac{(0-1)}{(-1+2)} = \boxed{-1}$	
2	8		$\frac{(3-3)}{(-1-0)} = 0$		$\frac{(2+1)}{(1+2)} = \boxed{1}$
		$\frac{(-1-8)}{(-1-2)} = 3$		$\frac{(2-0)}{(1-0)} = 2$	
-1	-1		$\frac{(1-3)}{(1-2)} = 2$		
		$\frac{(1+1)}{(1+1)} = 1$			
1	1				

- Newton interpolation polynomial is $\boxed{p(x) = 4 - (x+2) + x(x+2) - x(x+2)(x-2) + x(x+2)(x-2)(x+1)}$

Interpolating a Function

If interpolation points are discrete sample of underlying continuous function, then we may want to know how closely interpolant approximates given function between sample points.

If f is sufficiently smooth function, and p_{n-1} is unique polynomial of degree at most $n - 1$ that interpolates f at n points $t_1, \ldots, t_n$, then

$$f(t) - p_{n-1}(t) = \frac{f^{(n)}(\theta)}{n!}(t - t_1)(t - t_2)\cdots(t - t_n),$$

where θ is some (unknown) point in interval $[t_1, t_n]$.

Since point θ is unknown, this result is not particularly useful unless we have bound on appropriate derivative of f, but it still provides some insight into factors affecting accuracy of polynomial interpolation.

Example

The upward velocity of a rocket is given as a function of time in Table 1. Find the velocity at t=16 seconds using the Newton Divided Difference method for cubic interpolation.

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

Table 1: Velocity as a function of time

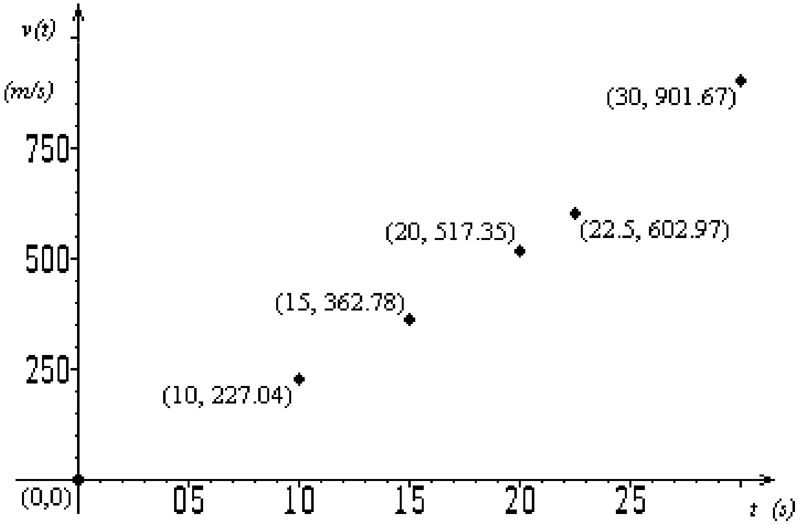


Figure 2: Velocity vs. time data for the rocket example

Example

The velocity profile is chosen as

$$v(t) = b_0 + b_1(t - t_0) + b_2(t - t_0)(t - t_1) + b_3(t - t_0)(t - t_1)(t - t_2)$$

we need to choose four data points that are closest to $t = 16$

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

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

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

$$t_3 = 22.5, \quad v(t_3) = 602.97$$

The values of the constants are found as:

$b_0 = 227.04$; $b_1 = 27.148$; $b_2 = 0.37660$; $b_3 = 5.4347*10^{-3}$

Example

t	b_0	b_1	b_2	b_3
$t_0 = 10$	227.04			
		27.148		
$t_1 = 15,$	362.78		0.37660	
		30.914		$5.4347x10^{-3}$
$t_2 = 20,$	517.35		0.44453	
		34.248		
$t_3 = 22.5,$	602.97			

$b_0 = 227.04$; $b_1 = 27.148$; $b_2 = 0.37660$; $b_3 = 5.4347*10^{-3}$

Example

Hence

$$v(t) = b_0 + b_1(t - t_0) + b_2(t - t_0)(t - t_1) + b_3(t - t_0)(t - t_1)(t - t_2)$$

$$= 227.04 + 27.148(t - 10) + 0.37660(t - 10)(t - 15)$$

$$+ 5.4347 * 10^{-3}(t - 10)(t - 15)(t - 20)$$

At $t = 16$,

$$v(16) = 227.04 + 27.148(16 - 10) + 0.37660(16 - 10)(16 - 15)$$

$$+ 5.4347 * 10^{-3}(16 - 10)(16 - 15)(16 - 20)$$

$$= 392.06 \text{ m/s}$$

The absolute relative approximate error $|\in_a|$ obtained is

$$|\in_a| = \left| \frac{392.06 - 392.19}{392.06} \right| \text{x} 100$$

$$= 0.033427 \ \%$$

High-Degree Polynomial Interpolation

Interpolating polynomials of high degree are expensive to determine and evaluate.

In some bases, coefficients of polynomial may be poorly determined due to ill-conditioning of linear system to be solved.

High-degree polynomial necessarily has lots of "wiggles," which may bear no relation to data to be fit.

Polynomial goes through required data points, but it may oscillate wildly between data points.

Placement of Interpolation Points

Equally spaced interpolation points often yield unsatisfactory results near ends of interval.

If points are bunched near ends of interval, more satisfactory results are likely to be obtained with polynomial interpolation.

One way to accomplish this is to use *Chebyshev* points

$$t_k = -\cos\left(\frac{(2k-1)\pi}{2n}\right), \quad k = 1, \ldots, n,$$

on interval [–1, 1], or suitable transformation of these points to arbitrary interval.

Placement of Points, continued

Chebyshev points are abscissas of points equally spaced around unit circle, but abscissas are bunched near ends of interval [–1, 1].