ME 209
Numerical Methods

6. Interpolation

Asst. Prof. Dr. N. Furkan DOĞAN

Mechanical Engineering Department

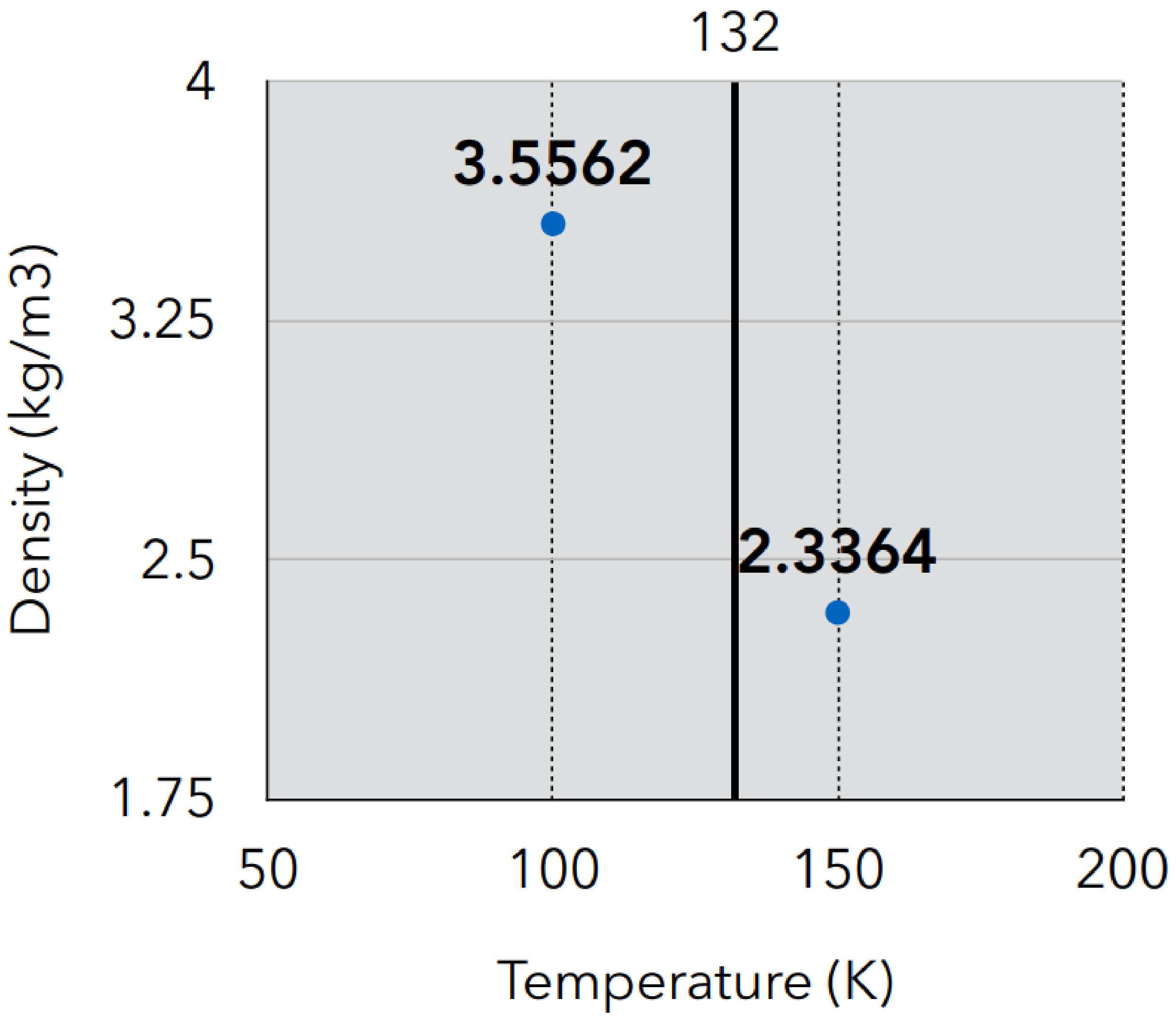
Gaziantep University

2024

- Motivation:
  - Often we have discrete data (tabulated, from experiments, etc) that we need to interpolate.
  - Interpolating functions form the basis for numerical integration and differentiation techniques
    - Used for solving ODEs & PDEs
    - we will cover this later
- Concept:
  - Choose a polynomial function to fit to the data (connect the dots)
  - Solve for the coefficients of the polynomial
  - Evaluate the polynomial wherever you want (interpolation)

T K	ρ kg/m³	λ W/(m K)	μ N s/m²
100	3.5562	0.0093	7.110e-06
150	2.3364	0.0138	1.034e-05
200	1.7458	0.0181	1.325e-05
250	1.3947	0.0223	1.596e-05
300	1.1614	0.0263	1.846e-05
350	0.9950	0.0300	2.082e-05
400	0.8711	0.0338	2.301e-05
450	0.7750	0.0373	2.507e-05
500	0.6864	0.0407	2.701e-05
550	0.6329	0.0439	2.884e-05
600	0.5804	0.0469	3.058e-05
650	0.5356	0.0497	3.225e-05
700	0.4975	0.0524	3.388e-05
750	0.4643	0.0549	3.546e-05
800	0.4354	0.0573	3.698e-05
850	0.4097	0.0596	3.843e-05
900	0.3868	0.0620	3.981e-05
950	0.3666	0.0643	4.113e-05
1000	0.3482	0.0667	4.244e-05

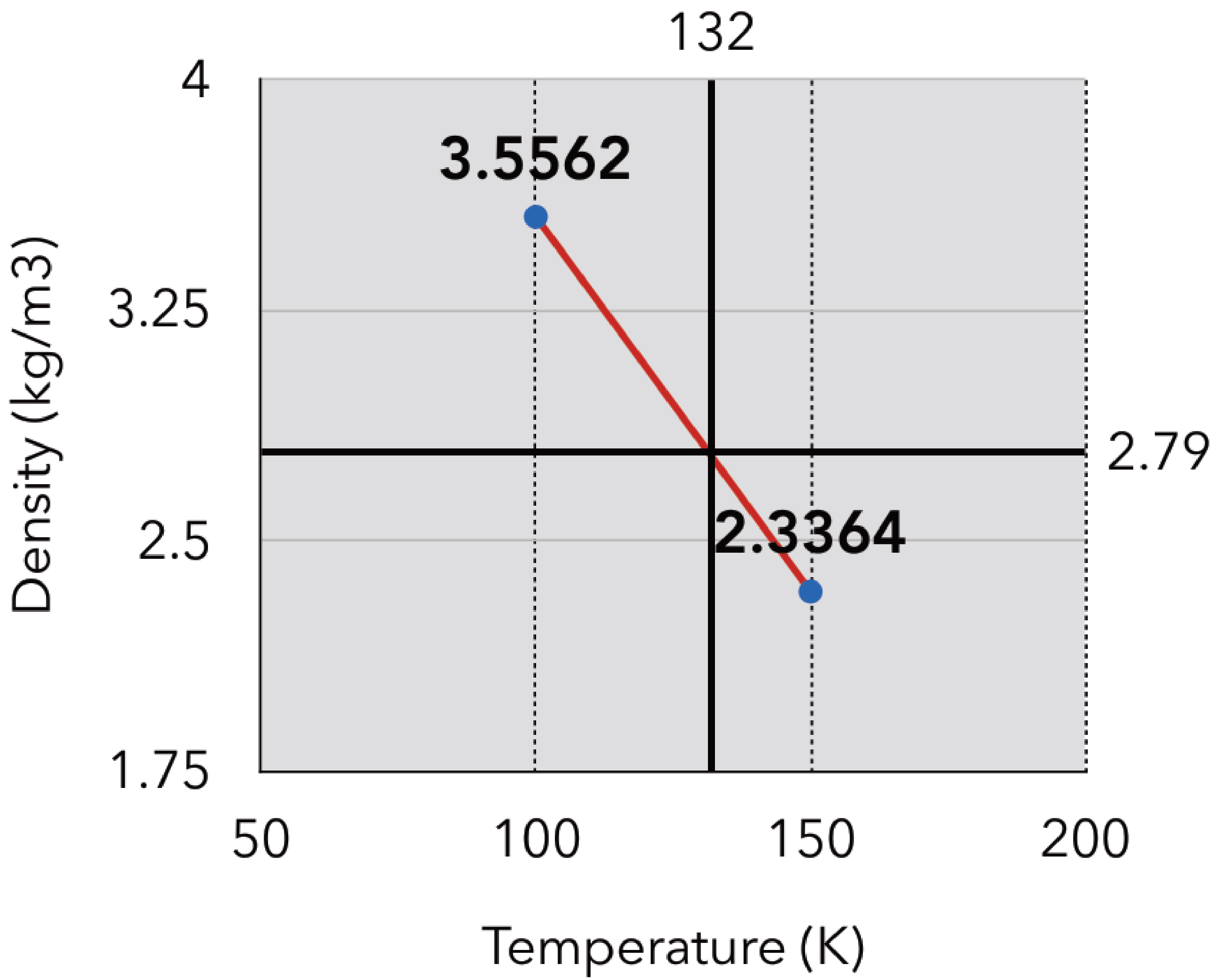
Incropera & DeWitt, *Fundamentals of Heat and Mass Transfer*, 4th ed.



Properties of air at atmospheric pressure

T	ρ
K	kg/m^3
100	3.5562
150	2.3364

find density @ T = 132 K



Connect via straight line

Properties of air at atmospheric pressure

T K	ρ kg/m³
100	3.5562
150	2.3364

find density @ T = 132 K

Find slope: $s = \frac{2.34 - 3.56}{50} = -0.024$

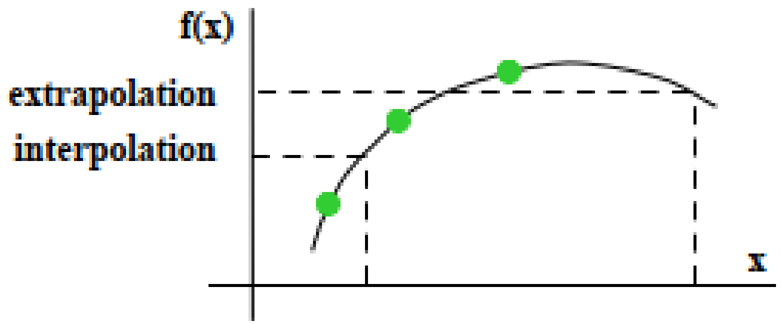
Find value:

$$\rho(132) = 3.56 + s \times (132 - 100) = 2.79$$

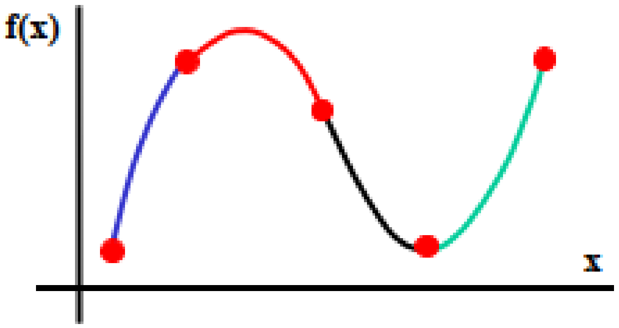
INTRODUCTION

- *Interpolation* is a method of estimating the intermediate values between precise data points.
- The basis of all interpolation algorithms is the fitting of some type of curve or function to a subset of the tabular data.
- Thus, we first fit a function that exactly passes through the given data points and than evaluate intermediate values using this function.

Polynomial Interpolation



Spline Interpolation



6.1 Polynomial Interpolation

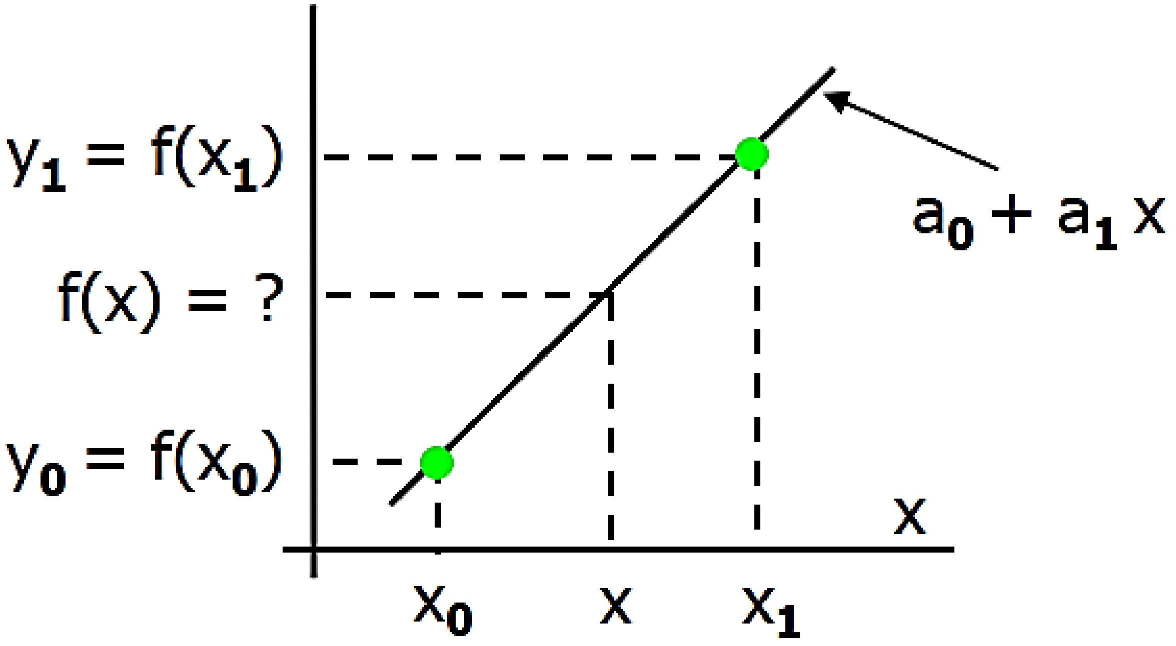
In this method, an nth-order polynomial is used as the interpolation function, $f(x)$:

$$f(x) = b_0 + b_1x + b_2x^2 + b_3x^3 + \ldots + b_nx^n$$

The constants in equation above, b_0, b_1, b_2, b_3, …, b_n, are determined using the measured data points. (x_1, y_1), (x_2, y_2), (x_3, y_3), . . . , (x_{n+1}, y_{n+1}). Here, $y_n=f(x)$

# data points	fit	equation
2 points	straight line	$b_0 + b_1 x$
3 points	quadratic	$b_0 + b_1 x + b_2 x^2$
4 points	cubic	$b_0 + b_1 x + b_2 x^2 + b_3 x^3$
n+1 points	nth order	$b_0 + b_1 x + b_2 x^2 + \ldots + b_n x^n$

Linear Interpolation:



- Given data points: (x_0, y_0) and (x_1, y_1)
- A straight line passes from these two point
- Using similar triangles:

<u>Linear interpolation formula</u>

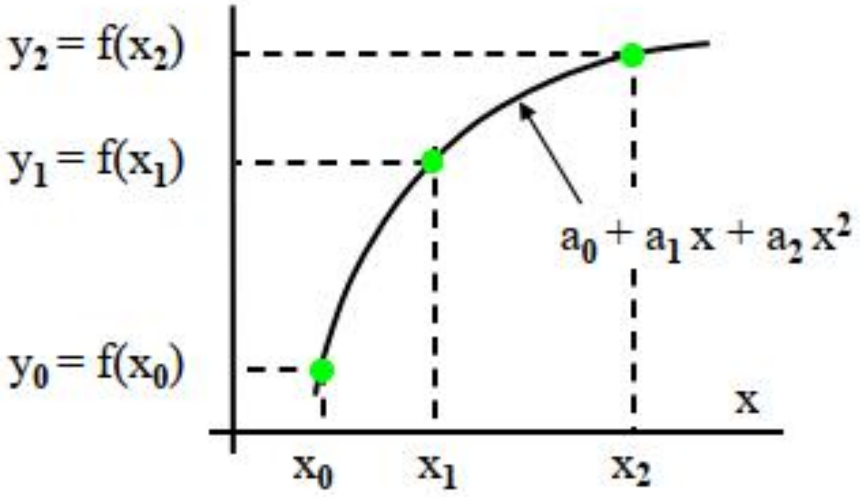
$$\frac{f(x)-f(x_0)}{x-x_0}=\frac{f(x_1)-f(x_0)}{x_1-x_0}$$

$$\boxed{f(x)=f(x_0)+\frac{[f(x_1)-f(x_0)]}{x_1-x_0}x-x_0}$$

or

$$\boxed{f(x)=b_0+b_1(x-x_0)}$$

• Quadratic Interpolation:



- Given: (x_0, y_0) , (x_1, y_1) and (x_2, y_2)
- A parabola passes from these three points.
- Similar to the linear case, the equation of this parabola can be written as

$$\boxed{f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)}$$

Quadratic interpolation formula

- How to find b_0, b_1 and b_2 in terms of given quantities?
  - at $x=x_0$ $\quad f_2(x) = f(x_0) = b_0$ $\quad \rightarrow \quad b_0 = f(x_0)$
  - at $x=x_1$ $\quad f_2(x) = f(x_1) = b_0 + b_1 x_1$ $\quad \rightarrow \quad b_1 = \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}$
  - at $x=x_2$ $\quad f_2(x) = f(x_2) = b_0 + b_1(x_2 - x_0) + b_2 (x_2 - x_0)(x_2 - x_1)$

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

Example 1

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 direct method for linear interpolation.

Table 1 Velocity as a function of time.

$t, (\mathrm{s})$	$v(t), (\mathrm{m/s})$
0	0
10	227.04
15	362.78
20	517.35
22.5	602.97
30	901.67

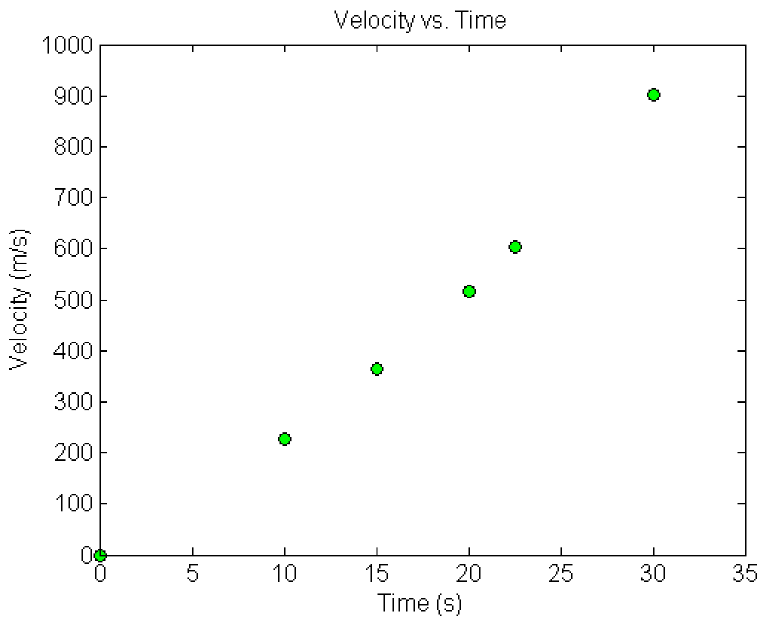


Figure 1 Velocity vs. time data for the rocket example

Solving by Linear Interpolation

$$v(t) = a_0 + a_1 t$$

$$v(15) = a_0 + a_1(15) = 362.78$$

$$v(20) = a_0 + a_1(20) = 517.35$$

Solving the above two equations gives,

$$a_0 = -100.93 \quad a_1 = 30.914$$

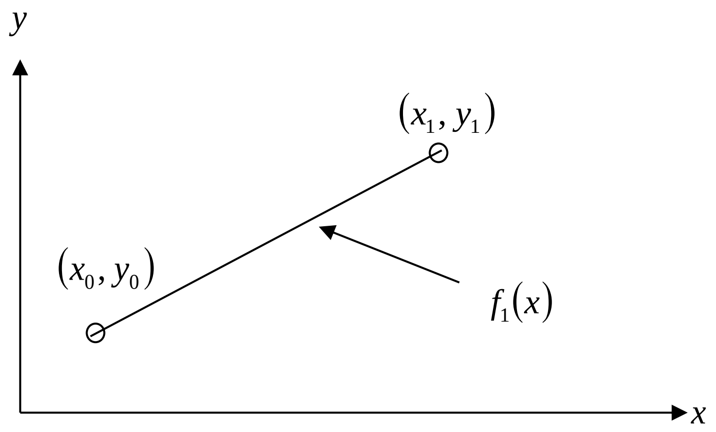


Figure 3 Linear interpolation.

Hence

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

$$v(16) = -100.93 + 30.914(16) = 393.7 \text{ m/s}$$

Solving by Quadratic Interpolation

$$v(t) = a_0 + a_1 t + a_2 t^2$$

$$v(10) = a_0 + a_1(10) + a_2(10)^2 = 227.04$$

$$v(15) = a_0 + a_1(15) + a_2(15)^2 = 362.78$$

$$v(20) = a_0 + a_1(20) + a_2(20)^2 = 517.35$$

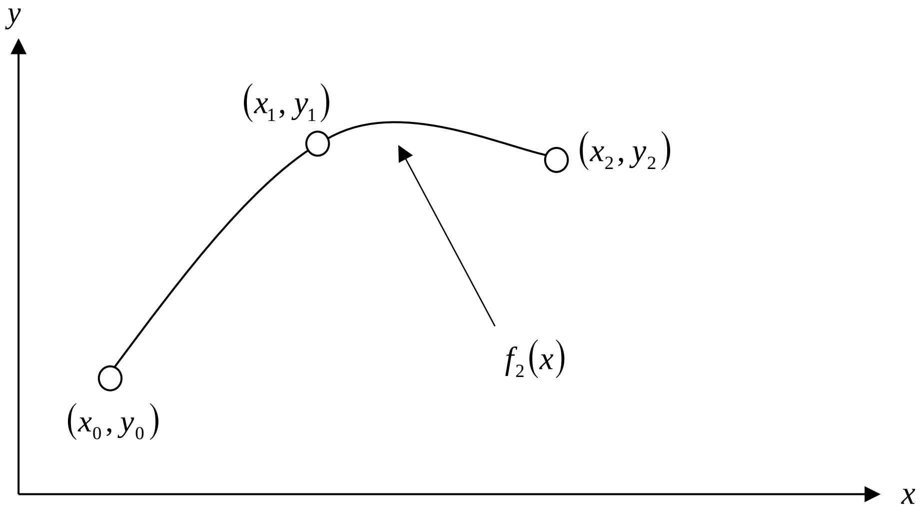


Figure 6 Quadratic interpolation.

Solving the above three equations gives

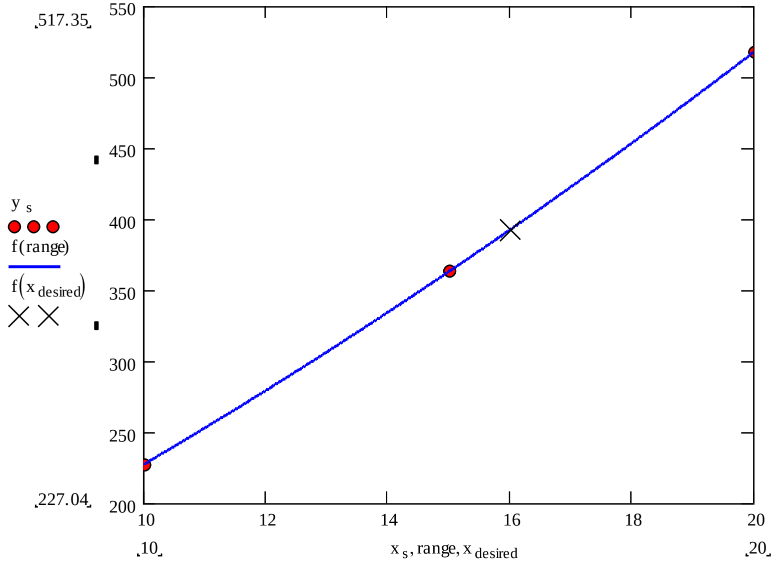
$$a_0 = 12.05 \quad a_1 = 17.733 \quad a_2 = 0.3766$$

Quadratic Interpolation (cont.)

$$v(t) = 12.05 + 17.733t + 0.3766t^2,\ 10 \le t \le 20$$

$$v(16) = 12.05 + 17.733(16) + 0.3766(16)^2$$

$$= 392.19 \text{ m/s}$$



The absolute relative approximate error $|\in_a|$ obtained between the results from the first and second order polynomial is

$$|\in_a| = \left|\frac{392.19 - 393.70}{392.19}\right| \times 100$$

$$= 0.38410\%$$

Solving by Cubic Interpolation

$$v(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

$$v(10) = 227.04 = a_0 + a_1(10) + a_2(10)^2 + a_3(10)^3$$

$$v(15) = 362.78 = a_0 + a_1(15) + a_2(15)^2 + a_3(15)^3$$

$$v(20) = 517.35 = a_0 + a_1(20) + a_2(20)^2 + a_3(20)^3$$

$$v(22.5) = 602.97 = a_0 + a_1(22.5) + a_2(22.5)^2 + a_3(22.5)^3$$

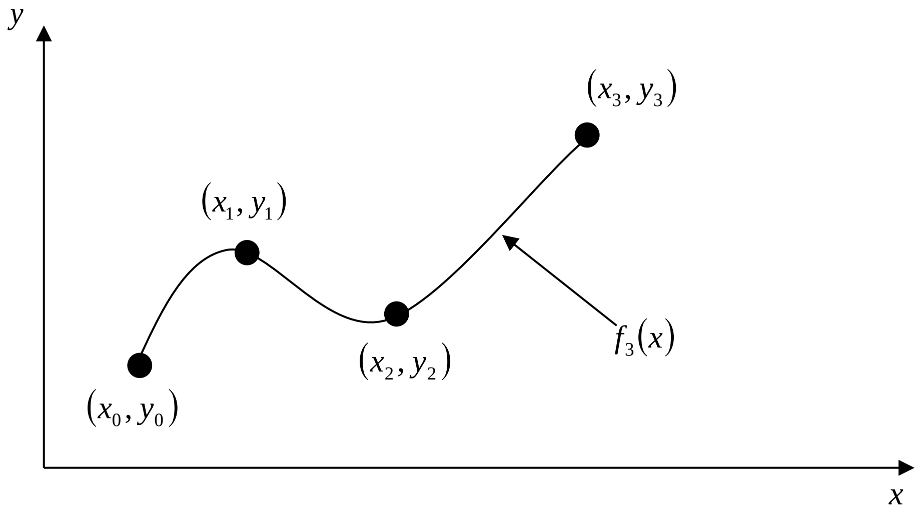


Figure Cubic interpolation.

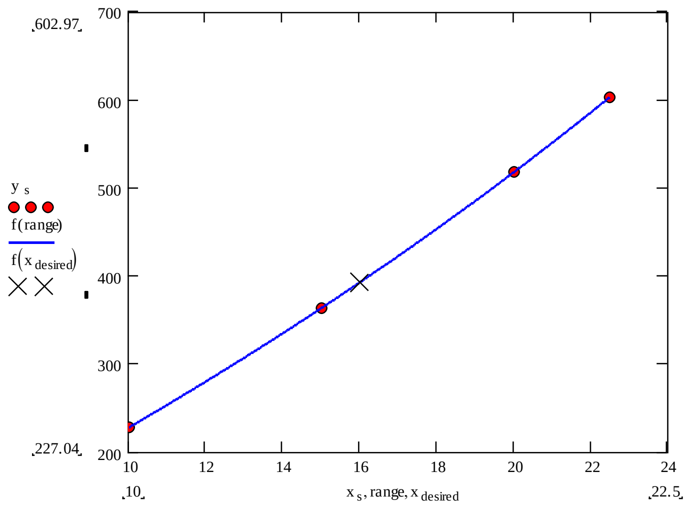
$$a_0 = -4.2540 \qquad a_1 = 21.266 \qquad a_2 = 0.13204 \qquad a_3 = 0.0054347$$

Cubic Interpolation (contd)

$$v(t) = -4.2540 + 21.266t + 0.13204\,t^2 + 0.0054347\,t^3, \quad 10 \le t \le 22.5$$

$$v(16) = -4.2540 + 21.266(16) + 0.13204(16)^2 + 0.0054347(16)^3$$

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



The absolute percentage relative approximate error $|\in_a|$ between second and third order polynomial is

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

$$= 0.033269\%$$

Comparison Table

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

Table 2 Comparison of different orders of the polynomial.

Order of Polynomial	1	2	3
$v(t=16)$ m/s	393.7	392.19	392.06
Absolute Relative Approximate Error	----------	0.38410 %	0.033269 %

Newton's Divided Difference Interpolating Polynomials

- We can generalize the linear and quadratic interpolation formulas for an n^{th} order polynomial passing through n+1 points

$$f_n(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) + \ . \ . \ . + b_n(x - x_0)(x - x_1)\cdots(x - x_{n-1})$$

where the constants are

$$b_0 = f(x_0) \qquad b_1 = f[x_1, x_0] \qquad b_2 = f[x_2, x_1, x_0] \qquad . \ . \ . \qquad b_n = f[x_n, x_{n-1}, . \ . \ ., x_1, x_0]$$

where the bracketed functions are finite divided differences evaluated recursively

$$\mathbf{f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j}} \qquad 1^{st} \text{ finite divided difference}$$

$$\mathbf{f[x_i, x_j, x_k] = \frac{f[x_i, x_j] - f[x_j, x_k]}{x_i - x_k}} \qquad 2^{nd} \text{ finite divided difference}$$

$$\mathbf{f[x_n, x_{n-1}, ..., x_1, x_0] = \frac{f[x_n, x_{n-1}, ..., x_1] - f[x_{n-1}, ..., x_1, x_0]}{x_n - x_0}} \qquad n^{th} \text{ finite divided difference}$$

- There n^{th} order Newton's Divided Difference Interpolating polynomial is

$$\boxed{\begin{aligned} f_n(x) = f(x_0) + (x - x_0) f[x_1, x_0] + (x - x_0)(x - x_1) f[x_2, x_1, x_0] + \ . \ . \ . \\ + (x - x_0)(x - x_1)\cdots(x - x_{n-1}) f[x_n, x_{n-1}, . \ . \ ., x_1, x_0] \end{aligned}}$$

Example 29:

The following logarithmic table is given.

x	f(x)=log(x)
4.0	0.60206
4.5	0.6532125
5.5	0.7403627
6.0	0.7781513

(a) Interpolate log(5) using the points x=4 and x=6

(b) Interpolate log(5) using the points x=4.5 and x=5.5

Note that the exact value is log(5) = 0.69897

(a) Linear interpolation. $f(x) = f(x_0) + (x - x_0)\, f[x_1, x_0]$

$x_0 = 4,\ \ x_1 = 6 \quad \rightarrow \quad f[x_1, x_0] = [f(6) – f(4)] / (6 - 4) = 0.0880046$

$f(5) \approx f(4) + (5 - 4)\, 0.0880046 = 0.690106 \qquad \varepsilon_t = 1.27\ \%$

(b) Again linear interpolation. But this time

$x_0 = 4.5,\ \ x_1 = 5.5 \quad \rightarrow \quad f[x_1, x_0] = [f(5.5) – f(4.5)] / (5.5 - 4.5) = 0.0871502$

$f(5) \approx f(4.5) + (5 – 4.5)\, 0.0871502 = 0.696788 \qquad \varepsilon_t = 0.3\ \%$

Example 29 (cont'd):

x	f(x)=log(x)
4.0	0.6020600
4.5	0.6532125
5.5	0.7403627
6.0	0.7781513

(c) Interpolate log(5) using the points x=4.5, x=5.5 and x=6

(c) Quadratic interpolation.

$x_0 = 4.5,\ x_1 = 5.5\ ,\ x_2 = 6 \rightarrow \quad f[x_1, x_0] = 0.0871502$ (already calculated)

$f[x_2, x_1] = [f(6) – f(5.5)] / (6 – 5.5) = 0.0755772$

$f[x_2, x_1\ , x_0] = \{f[x_2, x_1] - f[x_1, x_0]\} / (6 – 4.5) = -0.0077153$

$f(5) \approx 0.696788 + (5 - 4.5)(5 - 5.5)\ (-0.0077153) = 0.698717 \qquad \varepsilon_t = 0.04\ \%$

- Note that 0.696788 was calculate in part (b).
- Errors decrease when the points used are closer to the interpolated point.
- Errors decrease as the degree of the interpolating polynomial increases.

Finite Divided Difference (FDD) Table

Finite divided differences used in the Newton's Interpolating Polynomials can be presented in a table form. This makes the calculations much simpler.

x	f( )	f [,]	f [, ,]	f [, , ,]
x_0	$f(x_0)$	$f[x_1, x_0]$	$f[x_2, x_1, x_0]$	$f[x_3, x_2, x_1, x_0]$
x_1	$f(x_1)$	$f[x_2, x_1]$	$f[x_3, x_2, x_1]$	
x_2	$f(x_2)$	$f[x_3, x_2]$		
x_3	$f(x_3)$			

Exercise 27: The first two columns of the following table is given. Calculate the missing finite divided differences.

x	f( )	f [,]	f [, ,]	f [, , ,]
4	0.6020600	?	?	?
4.5	0.6532125	?	?	
5.5	0.7403627	?		
6	0.7781513			

• The numbers decrease as we go right in the table. This means that the contribution of higher order terms are less than the lower order terms.

• This is expected. The opposite behavior is an indication of an inappropriate interpolation (see exam questions of Fall 2006).

Example 30:

x	f( )	f[,]	f[, ,]	f[, , ,]
4	0.6020600	0.1023050	-0.0101032	0.001194
4.5	0.6532125	0.0871502	-0.0077153	
5.5	0.7403627	0.0755772		
6	0.7781513			

Use this previously calculated table to interpolate for log(5).

(a) Using points x=4 and x=4.5.

$\log(5) \approx 0.60206 + (5 - 4)\, 0.102305 = 0.704365 \qquad \varepsilon_t = 0.8\ \%$ (this is extrapolation)

(b) Using points x=4.5 and x=5.5.

$\log(5) \approx 0.6532125 + (5 - 4.5)\, 0.0871502 = 0.696788 \qquad \varepsilon_t = 0.3\ \%$

(c) Using points x=4 and x=6.

The entries of the above table can not be used for this interpolation.

(d) Using points x=4.5 , x=5.5 and x=6.

$\log(5) \approx 0.6532125 + (5-4.5)\, 0.0871502 + (5-4.5)(5-5.5)(-0.0077153) = 0.698717 \qquad \varepsilon_t = 0.04\ \%$

(e) Using all four points.

$$\log(5) \approx 0.60206 + (5 - 4)\, 0.102305 + (5 - 4)(5 - 4.5)(-0.0101032) + (5 - 4)(5 - 4.5)(5 - 5.5)(0.001194) = 0.6990149 \qquad \varepsilon_t = 0.006\ \%$$

Exercise 28:

x	f( )
-2	-0.909297
-1	-0.841471
0	0.000000
1	0.841471
3	0.141120
4	-0.756802
6	-0.279415

Create the FDD table for the given data set. Use it to interpolate for f(2).

- For a linear interpolation use the points x=1 and x=3.
- For a quadratic interpolation either use the points x=0, x=1 and x=3 or the points x=1, x=3 and x=4.
- For a third cubic interpolation use the points x=0, x=1, x=3 and x=4.

Important: Always try to put the interpolated point at the center of the points used for the interpolation.

Exercise 29: Complete the following table given for the log function. Do you observe anything strange? Comment.

x	f( )	f[,]	f[, ,]	f[, , ,]	f[, , , ,]	f[, , , , ,]
0.5						
1						
3						
5						
8						
10						

Errors of Newton's DD Interpolating Polynomials

$$f_n(x) = f(x_0) + (x - x_0) f[x_1, x_0] + (x - x_0)(x - x_1) f[x_2, x_1, x_0] + \dots$$
$$+ (x - x_0)(x - x_1) \cdots (x - x_{n-1}) f[x_n, x_{n-1}, \dots, x_1, x_0]$$

- The structure of Newton's Interpolating Polynomials is similar to the Taylor series.
- Remainder (truncation error) for the Taylor series was $R_n = \frac{f^{n+1}(\xi)}{(n+1)!}(x_{i+1} - x_i)^{n+1}$
- Similarly the remainder for the n^{th} order interpolating polynomial is

$$R_n = \frac{f^{n+1}(\xi)}{(n+1)!}(x - x_0)(x - x_1)\dots(x - x_n)$$

where ξ is somewhere in the interval containing the interpolated point x and other data points.

- But usually only the set of data points is given and the function f is not known.
- An alternative formulation uses a finite divided difference to approximate the $(n+1)^{th}$ derivative.

$$R_n \approx f[x, x_n, x_{n-1}, \dots, x_0](x - x_0)(x - x_1)\dots(x - x_n)$$

- But this includes f(x) which is not known.
- Error can be predicted if an additional data point (x_{n+1}) is availbale

$$R_n \approx f[x_{n+1}, x_n, x_{n-1}, \dots, x_0](x - x_0)(x - x_1)\dots(x - x_n)$$

which is nothing but $f_{n+1}(x) - f_n(x)$

Newton's Interpolating Polynomials for Equally Spaced Data

- If the data points are equally spaced and in ascending order, that is,

$$(x_0, y_0)\ ,\quad (x_0 + h, y_1)\ ,\quad (x_0 + 2h, y_1)\ ,\quad . . . . . .\quad ,\quad (x_0 + nh, y_n)$$

finite divided difference simplify.

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

$$f[x_2, x_1, x_0] = \frac{\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0} = \frac{f(x_2) - 2f(x_1) + f(x_0)}{2h^2} = \frac{\Delta f^2(x_0)}{2h^2}$$

$$\text{or in general}\quad f[x_n, x_{n-1}, \ldots, x_0] = \frac{\Delta f^n(x_0)}{n!\, h^n}$$

where $\Delta f^n(x_0)$ is the n^{th} forward difference.

- With this notation Newton's DD Interpolating polynomials simplify to

$$\boxed{f_n(x) = f(x_0) + \Delta f(x_0)\, \alpha + \Delta^2 f(x_0)\, \alpha(\alpha - 1) / 2! + \ . . . + \Delta^n f(x_0)\, \alpha(\alpha - 1) \cdots (\alpha - n + 1) / n! + R_n}$$

where $\alpha = (x - x_0) / h$ and $R_n = f^{(n+1)}(\xi)\, h^{n+1}\, \alpha(\alpha - 1) \cdots (\alpha - n) / (n+1)!$

- This is called the forward Newton-Gregory formula.

Lagrange Interpolating Polynomials

- It is a reformulation of Newton's Interpolating Polynomials.

$$\boxed{f_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i) \quad \text{where} \quad L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}}$$

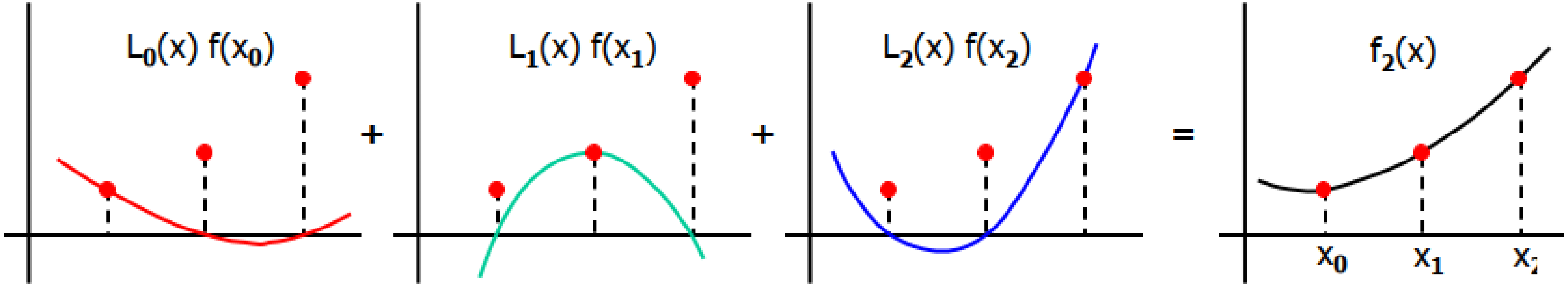
- For n=1 (linear): $f_1(x) = \dfrac{x - x_1}{x_0 - x_1} f(x_0) + \dfrac{x - x_0}{x_1 - x_0} f(x_1)$

- For n=2: $f_2(x) = \dfrac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \dfrac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) + \dfrac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2)$

- To generalize, n^{th} order polynomial is the summation of (n+1) n^{th} order polynomials.

- Each of these n^{th} order polynomials have a value of 1 at one of the data points and have values of 0 at all other data points.

- This is due to the following property of Lagrange functions $L_i(x) = \begin{cases} 1 & \text{at} \quad x = x_i \\ 0 & \text{at} \quad \text{all other data points} \end{cases}$



Example 31:

x	f(x)
1	4.75
2	4.00
3	5.25
5	19.75
6	36.00

Calculate f(4) using Lagrange Interpolating Polynomials

(a) of order 1

(b) of order 2

(c) of order 3

(a) Linear interpolation. Select $x_0 = 3$, $x_1 = 5$

$f_1(x) = L_0(x)\, f(x_0) + L_1(x)\, f(x_1) = (x-5)/(3-5)\ 5.25 + (x-3)/(5-3)\ 19.75$

$f(4) \approx 12.5$

(b) Quadratic interpolation. Select $x_0 = 2$, $x_1 = 3$, $x_1 = 5$

$f_2(x) = L_0(x)\, f(x_0) + L_1(x)\, f(x_1) + L_2(x)\, f(x_2)$

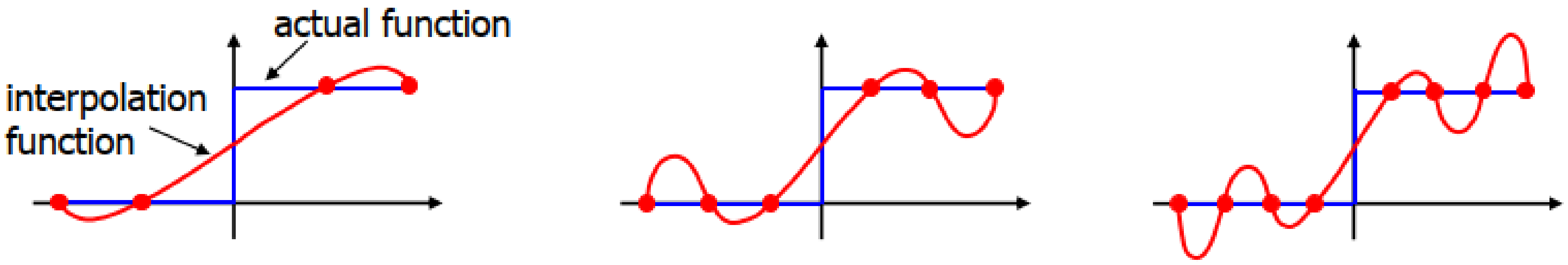
$= (x-3)(x-5)/(2-3)(2-5)\ 4.00 + (x-2)(x-5)/(3-2)(3-5)\ 5.25 + (x-2)(x-3)/(5-2)(5-3)\ 19.75$

$f(4) \approx 10.5$

Exercise 30: Solve part (b) using the last three points. Also solve part (c).

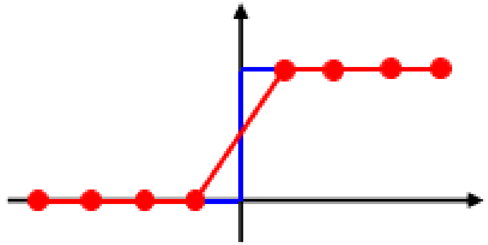
Spline Interpolation

- We learned how to interpolate between n+1 data points using n^{th} order polynomials.
- For high number of data points (typically $n > 6$ or 7), high order polynomials are necessary, but sometimes they suffer from oscillatory behavior.



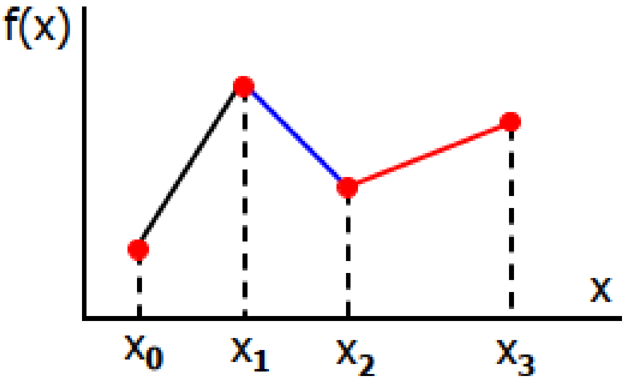
- Instead of using a single high order polynomial that passes through all data points, we can use different lower order polynomials between each data pair.
- These lower order polynomials that pass through only two points are called splines.
- Third order (cubic) splines are the most preferred ones.

first order splines :



Linear Splines:

- Given a set of ordered data points, each two point can be connected using a straight line.



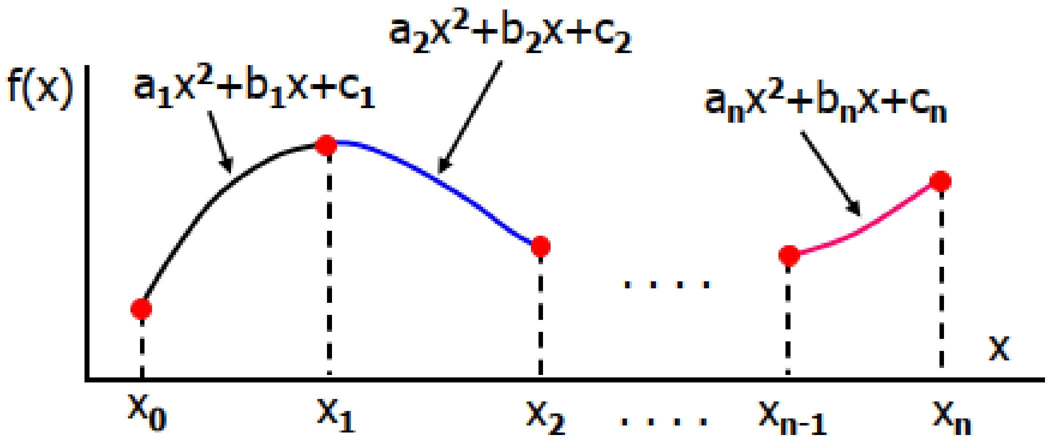
$$f(x) = f(x_0) + m_0(x - x_0) \qquad \text{for } x_0 \le x \le x_1$$
$$f(x) = f(x_1) + m_1(x - x_1) \qquad \text{for } x_1 \le x \le x_2$$
$$f(x) = f(x_2) + m_2(x - x_2) \qquad \text{for } x_2 \le x \le x_3$$

where the slopes are $m_i = [f(x_{i+1}) - f(x_i)] / (x_{i+1} - x_i)$

- Functions are not continuous at the interior points.

Quadratic Splines:

- Every pair of data points are connected using quadratic functions.



- For n+1 data points, there are n splines and 3n unknown constants.
- We need 3n equations to solve for them.

Quadratic Splines (cont'd):

- These 3n equations are
  - The first and last functions must pass through the end points (2 equations).

$$\boxed{\begin{array}{c} a_1\, x_0^2 + b_1\, x_0 + c_1 = f(x_0) \\ a_n\, x_n^2 + b_n\, x_n + c_n = f(x_n) \end{array}}$$

  - The function values must be equal at interior points (2n-2 equations).

$$\boxed{\begin{array}{c} a_{i-1}\, x_{i-1}^2 + b_{i-1}\, x_{i-1} + c_{i-1} = f(x_{i-1}) \\ a_i\, x_{i-1}^2 + b_i\, x_{i-1} + c_i = f(x_{i-1}) \end{array}} \qquad \text{for } i = 2 \text{ to } n$$

  - First derivatives must be equal at the interior points (n-1 equations).

$$\boxed{2\, a_{i-1}\, x_{i-1} + b_{i-1} = 2\, a_i\, x_{i-1} + b_i} \qquad \text{for } i = 1 \text{ to } n$$

  - This makes a total of 3n-1 equations. One more equation is necessary and we need to make an arbitrary choice. Among many possibilities we will use the following,

  - Take the second derivative at the first point to be zero (1 equation).

$\boxed{a_1 = 0}$ i.e. first two points are connected with a straight line.

- Solve this set of 3n linear algebraic equations with any of the methods that we learned.

Cubic Splines:

- For n+1 points, there will be n intervals and for each interval there will be a 3rd order polynomial

$$a_i\, x_i^3 + b_i\, x_i^2 + c_i\, x + d_i \qquad\qquad \text{for } i = 1 \text{ to } n$$

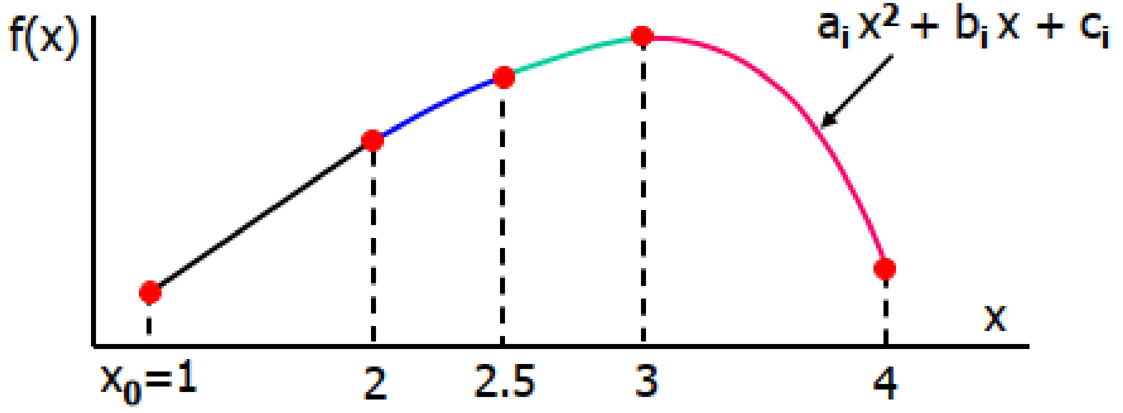
- Totally there are 4n unknowns. They can be solved using the following equations
  - The first and last functions must pass through the end points (2 equations).
  - The function values must be equal at interior points (2n-2 equations).
  - First derivatives must be equal at the interior points (n-1 equations).
  - Second derivatives must be equal at the interior points (n-1 equations).
  - This makes a total of 4n-2 equations. Two extra equations are (other choices are possible)
  - Second derivatives at the end points are zero (2 equations).

- Setting up and solving 4n equations is costly. There is another way of constructing cubic splines that results in only n-1 equations in n-1 unknowns. See pages 502-503 of the book.

Example 32:

x	f(x)
1	1
2	5
2.5	7
3	8
4	2

Develop quadratic splines for these data points and predict f(3.4) and f(2.2)



- There are 5 points and n=4 splines. Totally there are 3n=12 unknowns. Equations are
- End points: $a_1 1^2 + b_1 1 + c_1 = 1$, $a_4 4^2 + b_4 4 + c_4 = 2$
- Interior points: $a_1 2^2 + b_1 2 + c_1 = 5$, $a_2 2^2 + b_2 2 + c_2 = 5$

  $a_2 2.5^2 + b_2 2.5 + c_2 = 7$, $a_3 2.5^2 + b_3 2.5 + c_3 = 7$

  $a_3 3^2 + b_3 3 + c_3 = 8$, $a_4 3^2 + b_4 3 + c_4 = 8$
- Derivatives at the interior points: $2a_1 2 + b_1 = 2a_2 2 + b_2$

  $2a_2 2.5 + b_2 = 2a_3 2.5 + b_3$

  $2a_3 3 + b_3 = 2a_4 3 + b_4$
- Arbitrary choice for the missing equation: $a_1 = 0$

Example 32 (cont'd):

- a_1=0 is already known. Solve for the remaining 11 unknowns.

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 4 & 1 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6.25 & 2.5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6.25 & 2.5 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 & 3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 3 & 1 \\ 1 & 0 & -4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 1 & 0 & -5 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & 1 & 0 & -6 & -1 & 0 \end{bmatrix} \begin{Bmatrix} b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ a_3 \\ b_3 \\ c_3 \\ a_4 \\ b_4 \\ c_4 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 2 \\ 5 \\ 5 \\ 7 \\ 7 \\ 8 \\ 8 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \quad \rightarrow \quad \begin{Bmatrix} b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ a_3 \\ b_3 \\ c_3 \\ a_4 \\ b_4 \\ c_4 \end{Bmatrix} = \begin{Bmatrix} 4 \\ -3 \\ 0 \\ 4 \\ -3 \\ -4 \\ 24 \\ -28 \\ -6 \\ 36 \\ 46 \end{Bmatrix}$$

- Equations for the splines are

  1st spline: $f(x) = 4x - 3$ (Straight line.)

  2nd spline: $f(x) = 4x - 3$ (Same as the 1st. Coincidence)

  3rd spline: $f(x) = -4x^2 + 24x - 28$

  4th spline: $f(x) = -6x^2 + 36x - 46$

- To predict f(3.4) use the 4th spline. $f(3.4) = -6\,(3.4)^2 + 36\,(3.4) - 46 = 7.04$

  To predict f(2.2) use the 2nd spline. $f(2.2) = 4\,(2.2) - 3 = 5.8$