❖ Data points and curve fitting

- Data points are often given for discrete values along a continuum. However, one may require **estimates** at points **between the discrete values**. So, an appropriate curve is to be fitted from the given discrete data points.
- There are **two general approaches for curve fitting** that are distinguished from each other on the basis of the amount of error associated with these data.
  - **First,** where these data exhibit a significant degree of error or "noise," the strategy is to derive a single curve that represents the general trend of these data. Because any individual data point may be incorrect, no effort is made to intersect every point. Rather, the curve is designed to follow the pattern of the points taken as a group. The approach of this nature is called **regression**.
  - **Second,** where these data are known to be very precise, the basic approach is to fit a curve or a series of curves that pass directly through each of the points. Examples are values for various thermodynamic properties of water or gases as a function of temperature. The estimation of values between well-known discrete points is called **interpolation**.

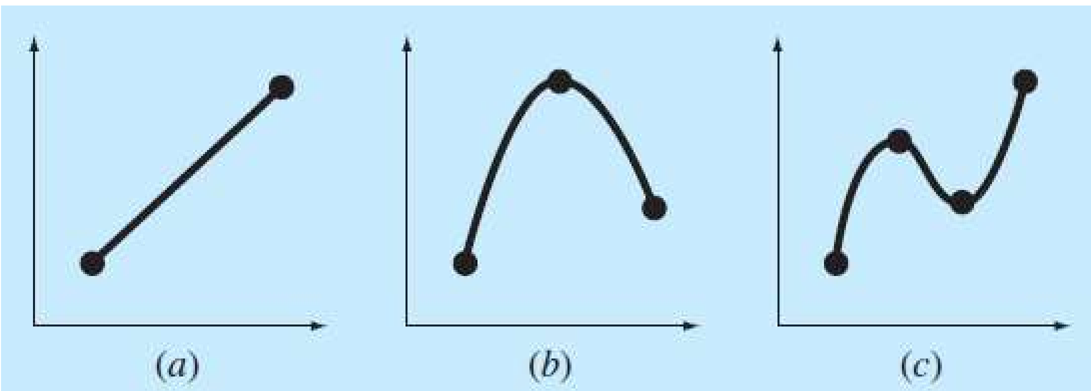
INTERPOLATION

❖ The most common method to estimate intermediate values between precise data points is polynomial interpolation. The general formula for an nth-order polynomial is:

$$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \tag{6.1}$$

❖ For $n + 1$ data points, there is one and only one polynomial of order n that passes through all the points. For example:

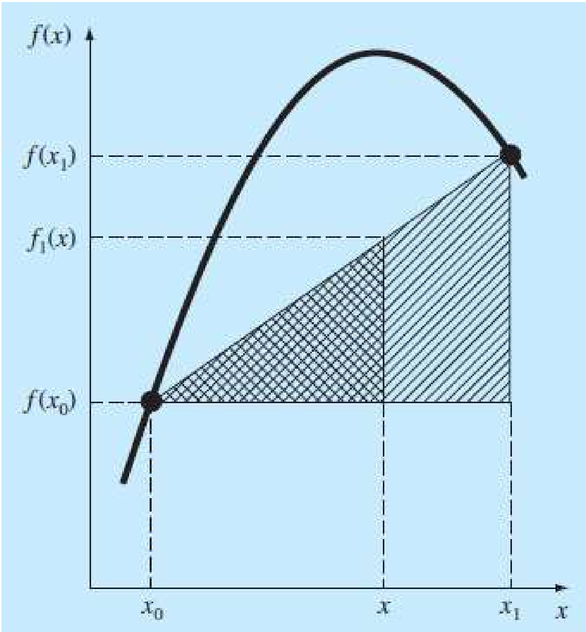
- There is only one straight line (that is, a first-order or linear polynomial) that connects two points as shown in Fig. (a) below.
- There is only one parabola (that is, a second-order or quadratic polynomial) connects a set of three points as shown in Fig. (b) below.
- There is only one cubic or third-order polynomial that connects a set of four points as shown in Fig. (c) below.



NEWTON'S DIFFERENCE FORMULAE

Newton's difference formulas are among the most popular and useful forms for expressing an interpolating polynomial. The first-order, second-order and the general equations are described below.

❖ **First-order (linear) interpolation** is the simplest interpolation technique, where two data points, $[x_0, f(x_0)]$ and $[x_1, f(x_1)]$, are connected by a straight line. The function value at any intermediate point x is then estimated along this line, as illustrated in the side figure. Using similar triangles,



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

which can be rearranged to yield:

$$f_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) \quad (6.2)$$

which is a linear-interpolation formula.

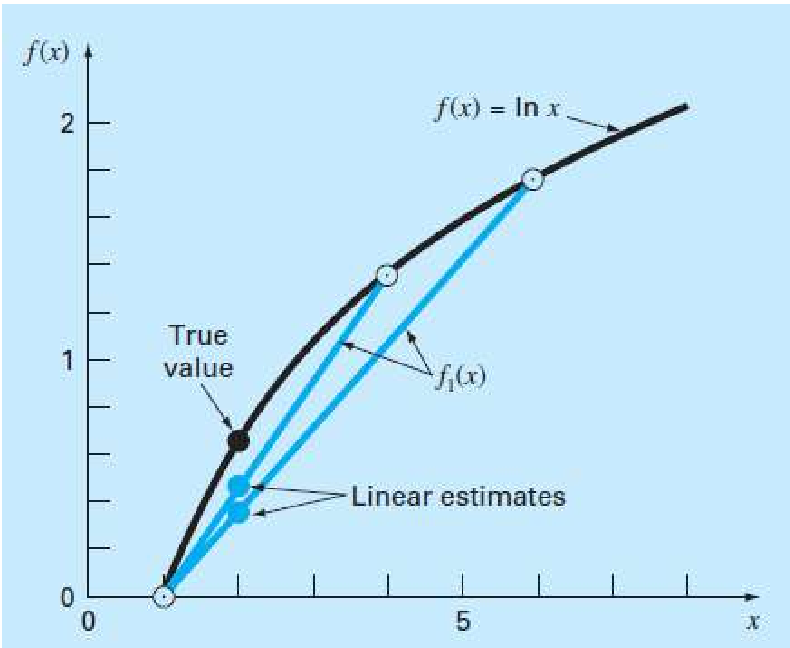
- The notation $f_1(x)$ designates that this is a first-order interpolating polynomial of form:

$$f_1(x) = a_0 + a_1 x$$

- The term $[f(x_1) - f(x_0)]/(x_1 - x_0)$ is a finite-divided-difference approximation of the first derivative.
- In general, the **smaller the interval** between the data points $(x_1 - x_0)$, the **better the approximation**. This is due to the fact that, as the interval decreases, a continuous function will be better approximated by a straight line.

Example 6.1: Estimate the value of the function $f(x) = \ln x$ at $x = 2$ using linear interpolation for given data points at (a) $x_0 = 1$ and $x_1 = 6$, (b)) $x_0 = 1$ and $x_1 = 4$.

Solution:



❖ **Second-order (quadratic) interpolation** introduces curvature into the line connecting the data points, thereby providing a more accurate estimate compared to linear interpolation, which approximates the curve using a straight line.

- **If three data points are available**, this can be accomplished with a second-order polynomial (also called a quadratic polynomial or a parabola). A particularly convenient form for this purpose is:

$$f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) \tag{6.3}$$

By multiplying and collecting the terms of Eq. (6.3), we can get the following form, similar to Eq. (6.1).

$$f_2(x) = a_0 + a_1 x + a_2 x^2$$

where

$$a_0 = b_0 - b_1 x_0 + b_2 x_0 x_1$$
$$a_1 = b_1 - b_2 x_0 - b_2 x_1$$
$$a_2 = b_2$$

- The following procedure can be used to determine the values of the coefficients of Eq. (6.3).
  - For b_0, Eq. (6.3) with $x = x_0$ can be used to compute:

$$b_0 = f(x_0) \tag{6.4}$$

  - Equation (6.4) can be substituted into Eq. (6.3), which can be evaluated at $x = x_1$ for

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \tag{6.5}$$

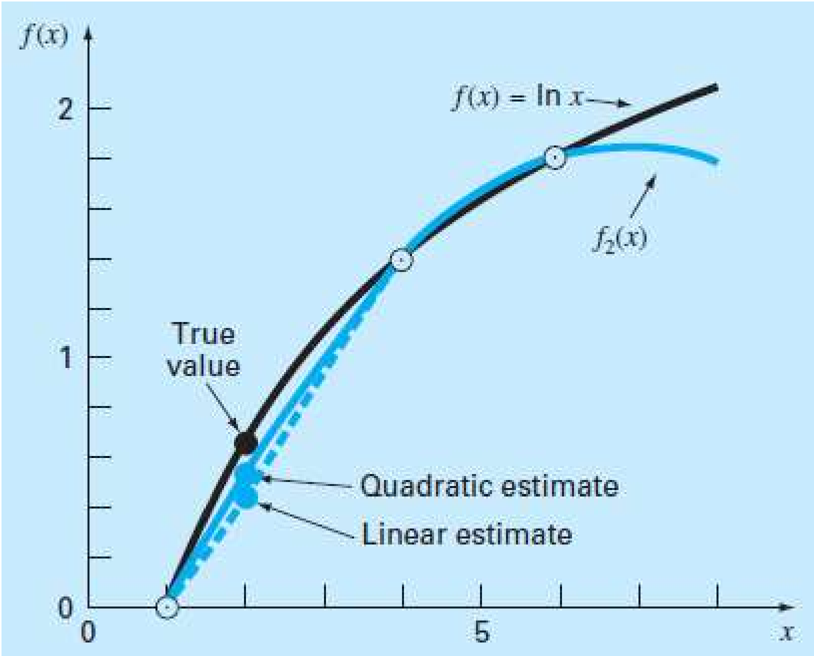
  - Finally, Eqs. (6.4) and (6.5) can be substituted into Eq. (6.3), which can be evaluated at $x = x_2$ and solved (after some algebraic manipulations) for

$$b_2 = \frac{\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0} \tag{6.6}$$

- **Note that:**
  - As was the case with linear interpolation, b_1 still represents the slope of the line connecting points x_0 and x_1. Thus, the first two terms of Eq. (6.3) are equivalent to linear interpolation from x_0 to x_1, as specifi ed previously in Eq. (6.2).
  - The last term, $b_2(x - x_0)(x - x_1)$, introduces the second-order curvature into the formula.
  - Do you notice any similarity between the expression for b_2 in Eq. (6.6) and the forward/backward/central difference approximation for the second derivative discussed in the previous note?

Example 6.2: Estimate the value of the function $f(x) = \ln x$ at $x = 2$ using second-order interpolation and compare its value obtained from linear interpolation. Take three data points at $x_0 = 1, x_1 = 4$ and $x_2 = 6$.

Solution:



❖ **General Form of Newton's Interpolating Polynomials:** The preceding analysis can be generalized to fit an nth-order polynomial to $n + 1$ data points. The nth-order polynomial is:

$$f_n(x) = b_0 + b_1(x - x_0) + \cdots + b_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}) \tag{6.7}$$

- As was done previously with the linear and quadratic interpolations, data points can be used to evaluate the coefficients $b_0, b_1, \cdots, b_n$.
- For an nth-order polynomial, $n + 1$ data points are required: $[x_0,\ f(x_0)]$, $[x_1,\ f(x_1)], \ldots, [x_n,\ f(x_n)]$.
- We use these data points and the following equations to evaluate the coefficients:

$$b_0 = f(x_0) \tag{6.8}$$

$$b_1 = f[x_1, x_0] \tag{6.9}$$

$$b_2 = f[x_2, x_1, x_0] \tag{6.10}$$

$$\vdots$$

$$b_n = f[x_n, x_{n-1}, \cdots, x_1, x_0] \tag{6.11}$$

- In the above equations, the square bracketed, i.e. [ ], function evaluations are **finite divided differences**. For example:
  - **The first finite divided difference is** represented generally as:

$$f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j} \tag{6.12}$$

  - **The second finite divided difference**, which **represents the difference of two first divided differences**, is expressed generally as:

$$f[x_i, x_j, x_k] = \frac{f[x_i, x_j] - f[x_j, x_k]}{x_i - x_k} \tag{6.13}$$

- **The third finite divided difference**, which **represents the difference of two second divided differences**, is expressed generally as:

$$f[x_i, x_j, x_k, x_l] = \frac{f[x_i, x_j, x_k] - f[x_j, x_k, x_l]}{x_i - x_l} \tag{6.13a}$$

$$= \frac{\dfrac{f[x_i, x_j] - f[x_j, x_k]}{x_i - x_k} - \dfrac{f[x_j, x_k] - f[x_k, x_l]}{x_j - x_l}}{x_i - x_l}$$

$$= \frac{\dfrac{\dfrac{f(x_i) - f(x_j)}{x_i - x_j} - \dfrac{f(x_j) - f(x_k)}{x_j - x_k}}{x_i - x_k} - \dfrac{\dfrac{f(x_j) - f(x_k)}{x_j - x_k} - \dfrac{f(x_k) - f(x_l)}{x_k - x_l}}{x_j - x_l}}{x_i - x_l}$$

- Similarly, **the *n*th finite divided difference** is:

$$f[x_n, x_{n-1}, \cdots, x_1, x_0] = \frac{f[x_n, x_{n-1}, \cdots, x_1] - f[x_{n-1}, x_{n-2} \cdots, x_1, x_0]}{x_n - x_0} \tag{6.14}$$

- These differences can be used to evaluate the coefficients in Eqs. (6.8) through (6.11), which can then be substituted into Eq. (6.7) to yield the interpolating polynomial:

$$f_n(x) = f(x_0) + (x - x_0)\, f[x_1, x_0] + (x - x_0)(x - x_1)\, f[x_2, x_1, x_0] + \cdots + (x - x_0)(x - x_1) \cdots (x - x_{n-1})\, f[x_n, x_{n-1}, \cdots, x_0] \tag{6.15}$$

which is called Newton's divided-difference interpolating polynomial.

- **Note that:**
  - It is not necessary that the data points used in Eq. (6.15) be equally spaced.
  - It is not necessary that the abscissa values in Eq. (6.15) be in ascending order.
  - Eqs. (6.12) through (6.14) are recursive, that is, higher-order differences are computed by taking differences of lower-order differences as depicted in below figure. (This property is exploited to develop efficient computer program.)

i	x_i	$f(x_i)$	First	Second	Third
0	x_0	$f(x_0)$	$f[x_1, x_0]$	$f[x_2, x_1, x_0]$	$f[x_3, x_2, x_1, x_0]$
1	x_1	$f(x_1)$	$f[x_2, x_1]$	$f[x_3, x_2, x_1]$	
2	x_2	$f(x_2)$	$f[x_3, x_2]$		
3	x_3	$f(x_3)$			

Example 6.3: In Example 6.2, data points at $x_0 = 1$, $x_1 = 4$ and $x_2 = 6$ were used to estimate ln 2 with a parabola. Now,

(a) Adding a fourth point $x_3 = 5$, estimate ln 2 with a third-order Newton's interpolating polynomial.

(b) Verify the result obtained in (a) by interchanging the values of x_2 and x_3.

ERRORS OF NEWTON'S INTERPOLATING POLYNOMIALS

❖ **Note that:**

- The structure of Eq. (6.15) is similar to the Taylor series expansion in the sense that terms are added sequentially to capture the higher-order behavior of the underlying function.
- These terms are finite divided differences and, thus, represent approximations of the higher-order derivatives.
- Consequently, as with the Taylor series, if the true underlying function is an *n*th-order polynomial, the *n*th-order interpolating polynomial based on $n + 1$ data points will yield exact results.

❖ From Eq. (5.5) in notes 5, the truncation error for Taylor series is expressed as:

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x_{i+1} - x_i)^{n+1}, \qquad x_i < \xi < x_{i+1}$$

where ξ is somewhere in the interval x_i to x_{i+1}.

- Similarly, the error for an *n*th-order interpolating polynomial is expressed as:

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)(x - x_1) \cdots (x - x_n), \qquad x_0 < \xi < x_n \tag{6.16}$$

where ξ is somewhere in the interval containing the unknown and the data.

- To use the above formula, the function in question must be known and differentiable. This is not usually the case.
- Fortunately, an alternative formulation is available that does not require prior knowledge of the function. Rather, it uses a finite divided difference to approximate the $(n + 1)$th derivative,

$$R_n = f[x, x_n, x_{n-1}, \cdots, x_0]\,(x - x_0)(x - x_1) \cdots (x - x_n) \tag{6.17}$$

where $f[x, x_n, x_{n-1}, \cdots, x_0]$ is the $(n + 1)$th finite divided difference.

- Because Eq. (6.17) contains the unknown $f(x)$, it cannot be solved for the error. However, if an additional data point $f(x_{n+1})$ is available, Eq. (6.17) can be used to estimate the error, as in

$$R_n \cong f[x_{n+1}, x_n, x_{n-1}, \cdots, x_0]\,(x - x_0)(x - x_1) \cdots (x - x_n) \tag{6.18}$$

❖ **Error Estimation for Newton's Polynomial:** The error estimate for the *n*th-order polynomial is equivalent to the difference between the $(n + 1)$th order and the *n*th-order prediction. That is,

$$R_n = f_{n+1}(x) - f_n(n) \tag{6.19}$$

- In other words, the increment that is added to the *n*th-order case to create the $(n + 1)$th order case [that is, Eq. (6.18)] is interpreted as an estimate of the *n*th-order error. This can be clearly seen by rearranging Eq. (6.19) to give

$$f_{n+1}(x) = f_n(n) + R_n$$

About forward difference operators $\Delta, \Delta^2, \Delta^3, \ldots$

- The symbol Δ is called **forward difference operator** and $\Delta f(x_0), \Delta f(x_1), \cdots \Delta f(x_{n-1})$ are called **first forward differences**. These are expressed as:

$$\Delta f(x_0) = f(x_1) - f(x_0), \qquad \Delta f(x_1) = f(x_2) - f(x_1),$$
$$\cdots, \qquad \Delta f(x_{n-1}) = f(x_n) - f(x_{n-1})$$

- The differences of first forward differences are called **second forward differences** and are denoted by $\Delta^2 f(x_0), \Delta^2 f(x_1), \cdots$ These are expressed as:

$$\Delta^2 f(x_0) = \Delta f(x_1) - \Delta f(x_0), \qquad \Delta^2 f(x_1) = \Delta f(x_2) - \Delta f(x_1)$$

- Similarly, third forward differences (Δ^3), forth forward differences (Δ^4), … can be developed

INTERPOLATION WITH EQUALLY SPACED DATA

❖ If data are equally spaced and in ascending order, then the independent variable assumes values of

$$x_1 = x_0 + h$$
$$x_2 = x_0 + 2h$$
$$\vdots$$
$$x_n = x_0 + nh$$

where h is the interval, or step size, between these data.

- On this basis, the finite divided differences can be expressed in concise form.
  - For example, **the second forward divided difference** is:

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

or $$f[x_0, x_1, x_2] = \frac{f(x_2) - 2f(x_1) + f(x_0)}{2h^2} = \frac{\Delta^2 f(x_0)}{2!\,h^2} \qquad (6E.1)$$

  - Similarly, **the third forward divided difference** is:

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0} = \frac{\dfrac{\Delta^2 f(x_1)}{2!\,h^2} - \dfrac{\Delta^2 f(x_0)}{2!\,h^2}}{3h}$$

or $$f[x_0, x_1, x_2, x_4] = \frac{\Delta^2 f(x_1) - \Delta^2 f(x_0)}{3!\ h^3} = \frac{\Delta^3 f(x_0)}{3!\ h^3} \qquad (6E.1a)$$

  - or, **in general**,

$$f[x_0, x_1, \cdots, x_n] = \frac{\Delta^n f(x_0)}{n!\,h^2} \qquad (6E.2)$$

❖ Using Eq. (6E.2), we can express Newton's interpolating polynomial [Eq. (6.15)] for the case of equally spaced data as:

$$f_n(x) = f(x_0) + \frac{\Delta f(x_0)}{h}(x - x_0) + \frac{\Delta^2 f(x_0)}{2!\,h^2}(x - x_0)(x - x_0 - h) + \cdots$$
$$+ \frac{\Delta^n f(x_0)}{n!\,h^n}(x - x_0)(x - x_0 - h) \cdots [x - x_0 - (n-1)h] + R_n \quad (6E.3)$$

where the remainder is the same as Eq. (6.16). This equation (6E.3) is known as Newton's formula, or the **Newton-Gregory forward formula**.

❖ Eq. (6E.3) can be **simplified further by defining a new quantity**, α:

$$\alpha = \frac{x - x_0}{h}$$

Therefore, we can write:

$$x - x_0 = \alpha h$$
$$x - x_0 - h = \alpha h - h = h(\alpha - 1)$$
$$\vdots$$
$$x - x_0 - (n-1)h = \alpha h - (n-1)h = h(\alpha - n + 1)$$

which can be substituted into Eq. (6E.3) to give

$$f_n(x) = f(x_0) + \Delta f(x_0)\alpha + \frac{\Delta^2 f(x_0)}{2!}\alpha(\alpha - 1)$$
$$+ \cdots + \frac{\Delta^n f(x_0)}{n!}\alpha(\alpha - 1) \cdots (\alpha - n + 1) + R_n \quad (6E.4)$$

Where

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}h^{n+1}\alpha(\alpha - 1)(\alpha - 2) \cdots (\alpha - n)$$

- **Note:**
  - The above concise notation will have utility in derivation and error analyses of the integration formulas.
  - In addition to the forward formula, backward and central Newton-Gregory formulas are also available.

Pseudocode for Newton's interpolating polynomial

```
SUBROUTINE NewtInt (x, y, n, xi, yint, ea)
  LOCAL fdd(n,n)
  DOFOR i = 0, n
    fdd(i,0) = y(i)
  END DO
  DOFOR j = 1, n
    DOFOR i = 0, n − j
      fdd(i,j) = (fdd(i+1,j−1) − fdd(i,j−1))/(x(i+j) − x(i))
    END DO
  END DO
  xterm = 1
  yint(0) = fdd(0,0)
  DOFOR order = 1, n
    xterm = xterm * (xi − x(order−1))
    yint2 = yint(order−1) + fdd(0,order) * xterm
    ea(order−1) = yint2 − yint(order−1)
    yint(order) = yint2
  END order
END NewtInt
```