Numerical Differentiation

Peng Yu

Tel: 0755 8801 8911

Email: yup6@sustech.edu.cn

Introduction

- Numerical differentiation is important for solving ODE and PDE numerically. For example, in fluid dynamics, solving Navier-Stokes Equations

$$\rho \frac{dV}{dt} = \rho g - \nabla p + \mu \nabla^2 V$$

- In Cartesian coordinates

$$\rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}\right) = f_x - \frac{\partial P}{\partial x} + \mu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2})$$

$$\rho\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z}\right) = f_y - \frac{\partial P}{\partial y} + \mu(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2})$$

$$\rho\left(\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z}\right) = f_z - \frac{\partial P}{\partial z} + \mu(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2})$$

Introduction

- How to calculate derivative by using numerical approximation

Bessel function $J_1(x)$: Eight equally spaced points over [0, 7] are (0, 0.0000), (1, 0.4400), (2, 0.5767), (3, 0.3391), (4,−0.0660), (5,−0.3276), (6,−0.2767), and (7,−0.004).

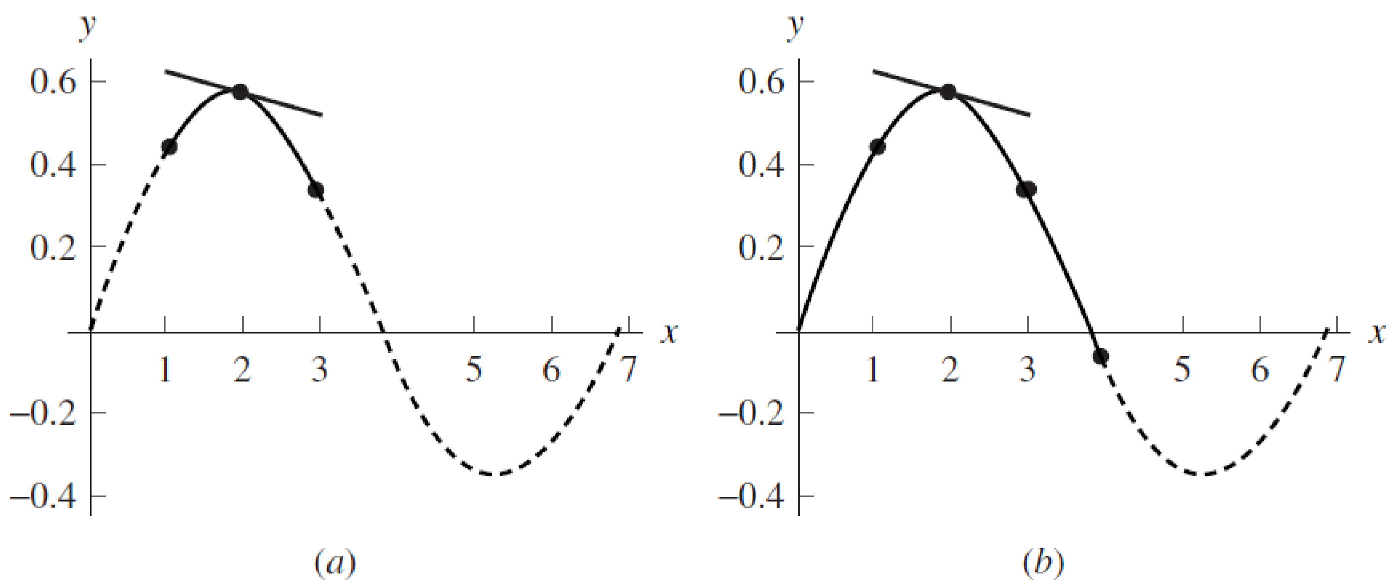

Figure 6.1 (a) The tangent to $p_2(x)$ at $(2, 0.5767)$ with slope $p_2'(2) = -0.0505$. (b) The tangent to $p_4(x)$ at $(2, 0.5767)$ with slope $p_4'(2) = -0.0618$.

Numerical Differentiation

- Approximating the Derivative

- Numerical Differentiation Formulas

Approximating the Derivative

Limit of the Difference Quotient

The numerical process for approximating the derivative of $f(x)$:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \ .$$

The method seems straightforward; choose a sequence $\{h_k\}$ so that $h_k \to 0$ and compute the limit of the sequence

$$D_k = \frac{f(x+h_k) - f(x)}{h_k} \quad \text{for } k = 1, 2, \ldots, n, \ldots$$

- We will only compute a finite number of terms $D_1, D_2, \ldots, D_N$ in the sequence and it appears that we should use D_N for our answer.
- What value h_N should be chosen so that D_N is a good approximation to the derivative?

An example

- Consider the function $f(x) = e^x$ and use the step sizes $h = 1$, $1/2$, and $1/4$ to construct the secant lines between the points $(0, 1)$ and $(h, f(h))$, respectively.

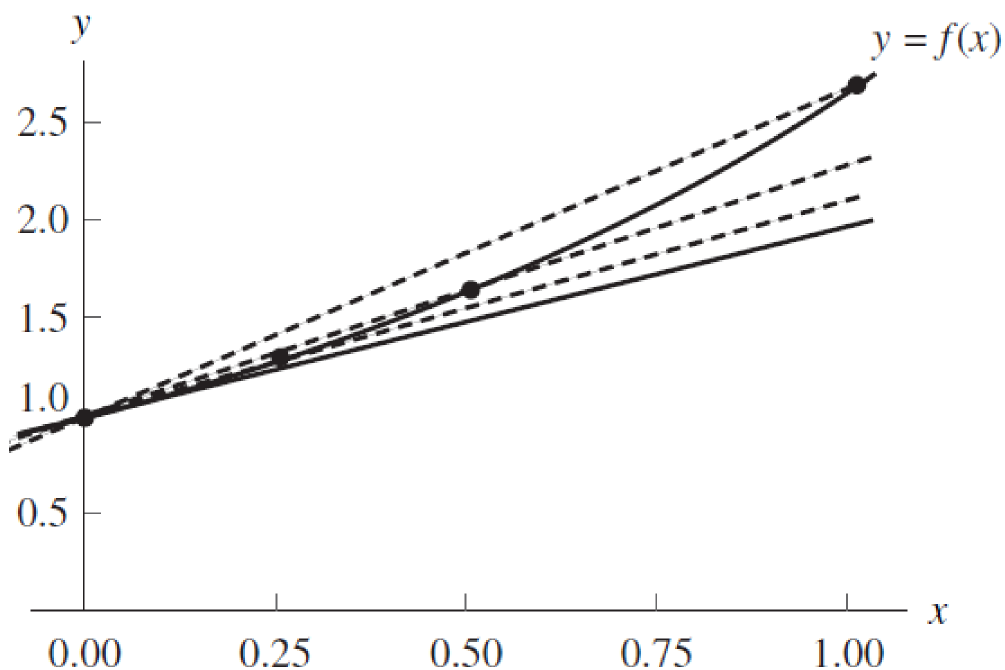

Figure 6.2 Several secant lines for $y = e^x$.

An example

Let $f(x) = e^x$ and $x = 1$. Compute the difference quotients D_k using the step sizes $h_k = 10^{-k}$ for $k = 1, 2, \ldots, 10$. Carry out nine decimal places in all calculations.

Table 6.1 Finding the Difference Quotients $D_k = (e^{1+h_k} - e)/h_k$ e = 2.718281828459045

h_k	$f_k = f(1 + h_k)$	$f_k - e$	$D_k = (f_k - e)/h_k$
$h_1 = 0.1$	3.004166024	0.285884196	2.858841960
$h_2 = 0.01$	2.745601015	0.027319187	2.731918700
$h_3 = 0.001$	2.721001470	0.002719642	2.719642000
$h_4 = 0.0001$	2.718553670	0.000271842	2.718420000
$h_5 = 0.00001$	2.718309011	0.000027183	2.718300000
$h_6 = 10^{-6}$	2.718284547	0.000002719	2.719000000
$h_7 = 10^{-7}$	2.718282100	0.000000272	2.720000000
$h_8 = 10^{-8}$	2.718281856	0.000000028	2.800000000
$h_9 = 10^{-9}$	2.718281831	0.000000003	3.000000000
$h_{10} = 10^{-10}$	2.718281828	0.000000000	0.000000000

- The sequence starts to converge to e, and D_5 is the closest; then the terms move away from e.
- In Program 6.1 it is suggested that terms in the sequence $\{D_k\}$ should be computed until $|D_{N+1} - D_N| \geq |D_N - D_{N-1}|$. This is an attempt to determine the best approximation before the terms start to move away from the limit.
- Here, we have $0.0007 = |D_6 - D_5| > |D_5 - D_4| = 0.00012$; hence D_5 is the answer we choose.

Central-Difference Formulas

Theorem 6.1 (Centered Formula of Order $\boldsymbol{O}(\boldsymbol{h^2})$). Assume that $f \in C^3[a,b]$ and that $x-h, x, x+h \in [a,b]$. Then

$$(3) \qquad f'(x) \approx \frac{f(x+h)-f(x-h)}{2h}.$$

Furthermore, there exists a number $c = c(x) \in [a,b]$ such that

$$(4) \qquad f'(x) = \frac{f(x+h)-f(x-h)}{2h} + E_{\text{trunc}}(f,h),$$

where

$$E_{\text{trunc}}(f,h) = -\frac{h^2 f^{(3)}(c)}{6} = \boldsymbol{O}(h^2).$$

The term $E(f,h)$ is called the ***truncation error.***

Central-Difference Formulas

Proof. Start with the second-degree Taylor expansions $f(x) = P_2(x) + E_2(x)$, about x, for $f(x+h)$ and $f(x-h)$:

(5) $$f(x+h) = f(x) + f'(x)h + \frac{f^{(2)}(x)h^2}{2!} + \frac{f^{(3)}(c_1)h^3}{3!}$$

and

(6) $$f(x-h) = f(x) - f'(x)h + \frac{f^{(2)}(x)h^2}{2!} - \frac{f^{(3)}(c_2)h^3}{3!} \; .$$

After (6) is subtracted from (5), the result is

(7) $$f(x+h) - f(x-h) = 2f'(x)h + \frac{((f^{(3)}(c_1) + f^{(3)}(c_2))h^3}{3!} \; .$$

Since $f^{(3)}(x)$ is continuous, the intermediate value theorem can be used to find a value c so that

(8) $$\frac{f^{(3)}(c_1) + f^{(3)}(c_2)}{2} = f^{(3)}(c).$$

Central-Difference Formulas

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{f^{(3)}(c)h^2}{3!}. \tag{9}$$

The first term on the right side of (9) is the central-difference formula (3), the second term is the truncation error, and the proof is complete.

- Suppose that the value of the third derivative $f^{(3)}(c)$ does not change too rapidly; then the truncation error in (4) goes to zero in the same manner as h^2, which is expressed by using the notation $\boldsymbol{O}(h^2)$.

- In numerical calculations, it is not desirable to choose h too small. For this reason, it is useful to have a formula for approximating $f'(x)$ that has a truncation error term of the order $\boldsymbol{O}(h^4)$.

Higher-order Central-Difference Formulas

Theorem 6.2 (Centered Formula of order $\boldsymbol{O}(\boldsymbol{h}^4)$). Assume that $f \in c^5[a,b]$ and that $x-2h, x-h, x, x+h, x+2h \in [a,b]$. Then

$$f'(x) \approx \frac{-f(x+2h)+8f(x+h)-8f(x-h)+f(x-2h)}{12h}. \tag{10}$$

Furthermore, there exists a number $c = c(x) \in [a,b]$ such that

$$f'(x) = \frac{-f(x+2h)+8f(x+h)-8f(x-h)+f(x-2h)}{12h} + E_{\text{trunc}}(f,h), \tag{11}$$

where

$$E_{\text{trunc}}(f,h) = \frac{h^4 f^{(5)}(c)}{30} = \boldsymbol{O}(h^4).$$

Higher-order Central-Difference Formulas

Proof. One way to derive formula (10) is as follows. Start with the difference between the fourth-degree Taylor expansions $f(x) = P_4(x) + E_4(x)$, about x, of $f(x+h)$ and $f(x-h)$:

$$f(x+h) - f(x-h) = 2f'(x)h + \frac{2f^{(3)}(x)h^3}{3!} + \frac{2f^{(5)}(c_1)h^5}{5!}. \tag{12}$$

Then use the step size $2h$, instead of h, and write down the following approximation:

$$f(x+2h) - f(x-2h) = 4f'(x)h + \frac{16f^{(3)}(x)h^3}{3!} + \frac{64f^{(5)}(c_2)h^5}{5!}. \tag{13}$$

Next multiply the terms in equation (12) by 8 and subtract (13) from it. The terms involving $f^{(3)}(x)$ will be eliminated and we get

$$\begin{aligned} &-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h) \\ &\qquad = 12f'(x)h + \frac{(16f^{(5)}(c_1) - 64f^{(5)}(c_2))h^5}{120}. \end{aligned} \tag{14}$$

Higher-order Central-Difference Formulas

If $f^{(5)}(x)$ has one sign and if its magnitude does not change rapidly, we can find a value c that lies in $[x-2h, x+2h]$ so that

(15) $$16f^{(5)}(c_1) - 64f^{(5)}(c_2) = -48f^{(5)}(c).$$

After (15) is substituted into (14) and the result is solved for $f'(x)$, we obtain

(16) $$f'(x) = \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h} + \frac{f^{(5)}(\mathrm{c})h^4}{30}.$$

- Suppose that $|f^{(5)}(c)|$ is bounded for $c \in [a, b]$; then the truncation error in (11) goes to zero in the same manner as h^4, which is expressed with the notation $\boldsymbol{O}(h^4)$.
- Suppose that $f(x)$ has five continuous derivatives and that $|f^{(3)}(c)|$ and $|f^{(5)}(c)|$ are about the same. Then the truncation error for the fourth-order formula (10) is $\boldsymbol{O}(h^4)$ and will go to zero faster than the truncation error $\boldsymbol{O}(h^2)$ for the second-order formula (3). This permits the use of a larger step size.

An example

Let $f(x) = \cos(x)$.

(a) Use formulas (3) and (10) with step sizes $h = 0.1, 0.01, 0.001,$ and 0.0001, and calculate approximations for $f'(0.8)$. Carry nine decimal places in all the calculations.

(b) Compare with the true value $f'(0.8) = -\sin(0.8)$.

(a) Using formula (3) with $h = 0.01$, we get

$$f'(0.8) \approx \frac{f(0.81) - f(0.79)}{0.02} \approx \frac{0.689498433 - 0.703845316}{0.02} \approx -0.717344150\,.$$

Using formula (10) with $h = 0.01$, we get

$$\begin{aligned} f'(0.8) &\approx \frac{-f(0.82) + 8f(0.81) - 8f(0.79) + f(0.78)}{0.12} \\ &\approx \frac{-.682221207 + 8(0.689498433) - 8(0.703845316) + 0.710913538}{0.12} \\ &\approx -0.717356108. \end{aligned}$$

An example

(b) The error in approximation for formulas (3) and (10) turns out to be −0.000011941 and 0.000000017, respectively. In this example, formula (10) gives a better approximation to $f'(0.8)$ than formula (3) when $h = 0.01$. The error analysis will illuminate this example and show why this happened . The other calculations are summarized in Table 6.2.

Table 6.2 Numerical Differentiation Using Formulas (3) and (10)

Step size	Approximation by formula (3)	Error using formula (3)	Approximation by formula (10)	Error using formula (10)
0.1	−0.716161095	−0.001194996	−0.717353703	−0.000002389
0.01	−0.717344150	−0.000011941	−0.717356108	0.000000017
0.001	−0.717356000	−0.000000091	−0.717356167	0.000000076
0.0001	−0.717360000	−0.000003909	−0.717360833	0.000004742

Error Analysis and Optimum Step Size

An important topic in the study of numerical differentiation is the effect of the computer's round-off error. Let's consider

$$f(x_0 - h) = y_{-1} + e_{-1} \quad \text{and} \quad f(x_0 + h) = y_1 + e_1 \,,$$

Corollary 6.1 (a). Assume that f satisfies the hypotheses of Theorem 6.1 and use the ***computational formula***

(17) $$f'(x_0) \approx \frac{y_1 - y_{-1}}{2h}.$$

The error analysis is explained by the following equations:

(18) $$f'(x_0) = \frac{y_1 - y_{-1}}{2h} + E(f, h),$$

where

(19) $$\begin{aligned} E(f, h) &= E_{round}(f, h) + E_{trunc}(f, h) \\ &= \frac{e_1 - e_{-1}}{2h} - \frac{h^2 f^{(3)}(c)}{6}, \end{aligned}$$

Error Analysis and Optimum Step Size

Corollary 6.1 (b). Assume that f satisfies the hypotheses of Theorem 6.1 and that numerical computations are made. If $|e_{-1}| \le \epsilon$, $|e_1| \le \epsilon$, and $M = \max_{a \le x \le b}\{|f^{(3)}(x)|\}$, then

(20) $$|E(f,h)| \le \frac{\epsilon}{h} + \frac{Mh^2}{6},$$

and the value of h that minimizes the right-hand side of (20) is

(21) $$h = \left(\frac{3\epsilon}{M}\right)^{1/3}.$$

When h is small, the portion of (19) involving $(e_1 - e_{-1})/2h$ can be relatively large.

Error Analysis and Optimum Step Size

In previous example, when h = 0.0001, the round-off errors are:

$$f(0.8001) = 0.696634970 + e_1 \quad \text{where } e_1 \approx -0.0000000003$$
$$f(0.7999) = 0.696778442 + e_{-1} \quad \text{where } e_{-1} \approx 0.0000000005.$$

The truncation error term is

$$\frac{-h^2 f^{(3)}(c)}{6} \approx -(0.0001)^2 \left(\frac{\sin(0.8)}{6}\right) \approx 0.000000001.$$

The error term $E(f,h)$ in (19) can now be estimated:

$$E(f,h) \approx \frac{-0.0000000003 - 0.0000000005}{0.0002} - 0.000000001$$
$$= -0.000004001.$$

Indeed, the computed numerical approximation for the derivative using $h = 0.0001$ is found by the calculation

$$f'(0.8) \approx \frac{f(0.8001) - f(0.7999)}{0.0002} = \frac{0.696634970 - 0.696778442}{0.0002}$$
$$= -0.717360000,$$

Error Analysis and Optimum Step Size

- When $h = 0.0001$, a loss of about four significant digits is evident. The error is -0.000003909 and this is close to the predicted error, -0.000004001.
- When formula (21) is applied to previous Example, we can use the bound $|f^{(3)}(x)| \leq |\sin(x)| \leq 1 = M$ and the value $= 0.5 \times 10^{-9}$ for the magnitude of the roundoff error. The optimal value for h is easily calculated: $h = 0.001144714$.
- The step size $h = 0.001$ was closest to the optimal value.

Error Analysis and Optimum Step Size

Table 6.2 Numerical Differentiation Using Formulas (3) and (10)

Step size	Approximation by formula (3)	Error using formula (3)	Approximation by formula (10)	Error using formula (10)
0.1	−0.716161095	−0.001194996	−0.717353703	−0.000002389
0.01	−0.717344150	−0.000011941	−0.717356108	0.000000017
0.001	−0.717356000	−0.000000091	−0.717356167	0.000000076
0.0001	−0.717360000	−0.000003909	−0.717360833	0.000004742

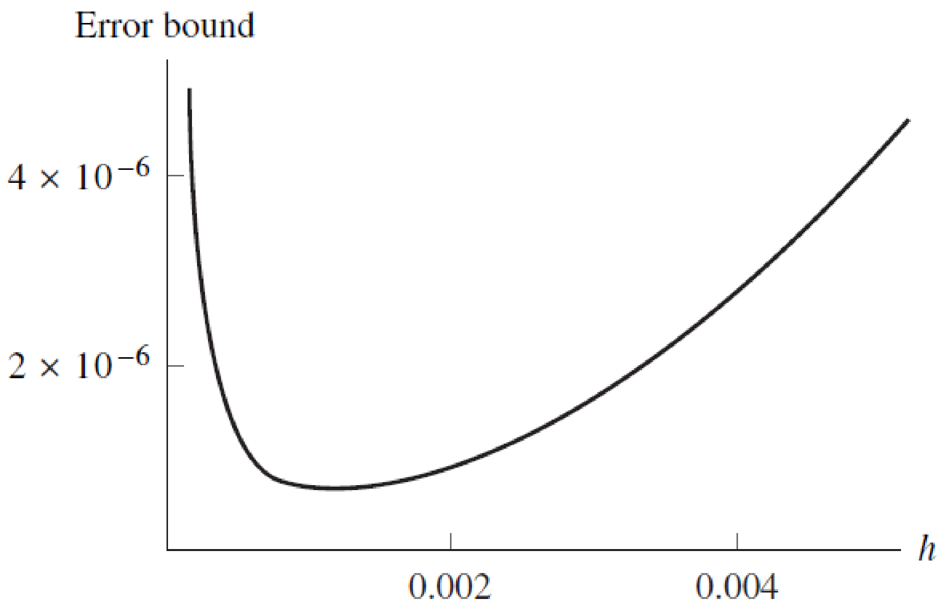

Figure 6.3 Finding the optimal step size $h = 0.001144714$ when formula (21) is applied to $f(x) = \cos(x)$ in Example 6.2.

Error Analysis and Optimum Step Size

An error analysis of formula (10) is similar. Assume that a computer is used to make numerical computations and that $f(x_0 + kh) = y_k + e_k$.

Corollary 6.2(a). Assume that *f* satisfies the hypotheses of Theorem 6.2 and use the ***computational formula***

$$f'(x_0) \approx \frac{-y_2 + 8y_1 - 8y_{-1} + y_{-2}}{12h}. \tag{22}$$

The error analysis is explained by the following equations:

$$f'(x_0) = \frac{-y_2 + 8y_1 - 8y_{-1} + y_{-2}}{12h} + E(f,h), \tag{23}$$

where

$$\begin{aligned} E(f,h) &= E_{round}(f,h) + E_{trunc}(f,h) \\ &= \frac{-e_2 + 8e_1 - 8e_{-1} + e_{-2}}{12h} + \frac{h^4 f^{(5)}(c)}{30}. \end{aligned} \tag{24}$$

Error Analysis and Optimum Step Size

Corollary 6.2(b). Assume that f satisfies the hypotheses of Theorem 6.2 and that numerical computations are made. If $|e_k| \leq \epsilon$ and $M = \max_{a \leq x \leq b}\{|f^{(5)}x|\}$, then

(25) $$|E(f,h)| \leq \frac{3\epsilon}{2h} + \frac{Mh^4}{30}\ .$$

And the value of h that minimizes the right-hand side of (25) is

(26) $$h = \left(\frac{45\epsilon}{4M}\right)^{1/5}\ .$$

- When formula (25) is applied to previous Example, we can use the bound $|f^{(5)}(x)| \leq |\sin(x)| \leq 1 = M$ and the value $= 0.5 \times 10^{-9}$ for the magnitude of the roundoff error. The optimal value for h is calculated: $h =$ 0.022388475.
- The step size $h = 0.01$ was closest to the optimal value.

Error Analysis and Optimum Step Size

Table 6.2 Numerical Differentiation Using Formulas (3) and (10)

Step size	Approximation by formula (3)	Error using formula (3)	Approximation by formula (10)	Error using formula (10)
0.1	−0.716161095	−0.001194996	−0.717353703	−0.000002389
0.01	−0.717344150	−0.000011941	−0.717356108	0.000000017
0.001	−0.717356000	−0.000000091	−0.717356167	0.000000076
0.0001	−0.717360000	−0.000003909	−0.717360833	0.000004742

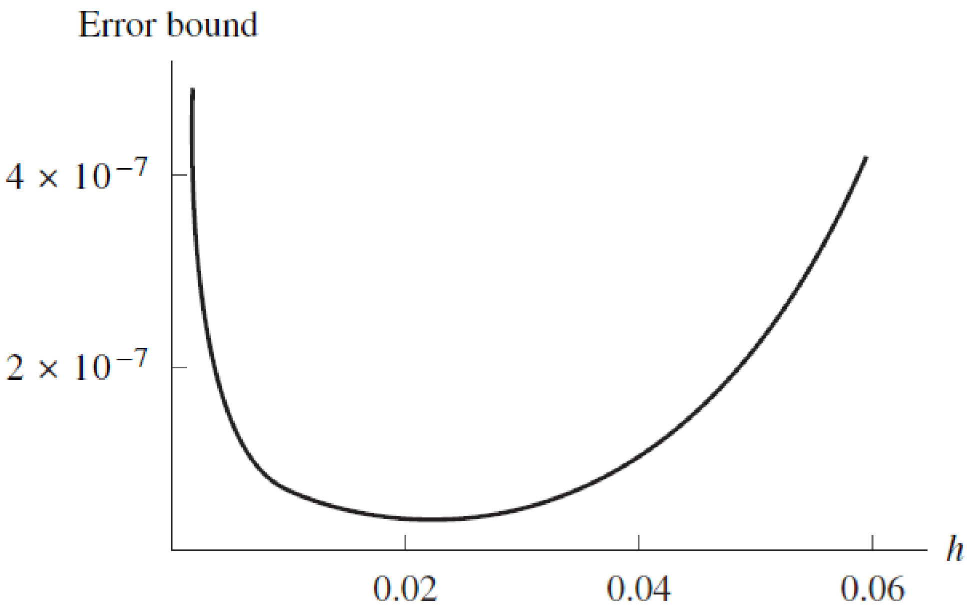

Figure 6.4 Finding the optimal step size $h = 0.022388475$ when formula (26) is applied to $f(x) = \cos(x)$ in Example 6.2.

Differentiation of an interpolation polynomial

- An alternative derivation: derived by differentiation of an interpolation polynomial.

- For example, the Lagrange form of the quadratic polynomial $p_2(x)$ that passes through the three points (0.7, cos(0.7)), (0.8, cos(0.8)), and (0.9, cos(0.9)) is

$$p_2(x) = 38.2421094(x-0.8)(x-0.9) - 69.6706709(x-0.7)(x-0.9)$$
$$+31.0804984(x-0.7)(x-0.8).$$

This polynomial can be expanded to obtain the usual form:

$$p_2(x) = 1.046875165 - 0.159260044x - 0.348063157x^2 .$$

Differentiation of an interpolation polynomial

A similar computation can be used to obtain the quartic polynomial $p_4(x)$ that passes through the points $(0.6, \cos(0.6))$, $(0.7, \cos(0.7))$, (0.8,cos(0.8)), (0.9,cos(0.9)), (1.0,cos(1.0)),

$$p_4(x) = 0.998452927 + 0.0096238391x - 0.523291341x^2$$
$$+0.026521229x^3 + 0.028981100x^4.$$

When these polynomials are differentiated, they produce $p_2'(0.8) = -0.716161095$ and $p_4'(0.8) = -0.717353703$, which agree with the values listed under $h = 0.1$ in Table 6.2. The graphs of $p_2(x)$ and $p_4(x)$ and their tangent lines at $(0.8, \cos(0.8))$ are shown in Figure 6.5(a) and (b), respectively

Differentiation of an interpolation polynomial

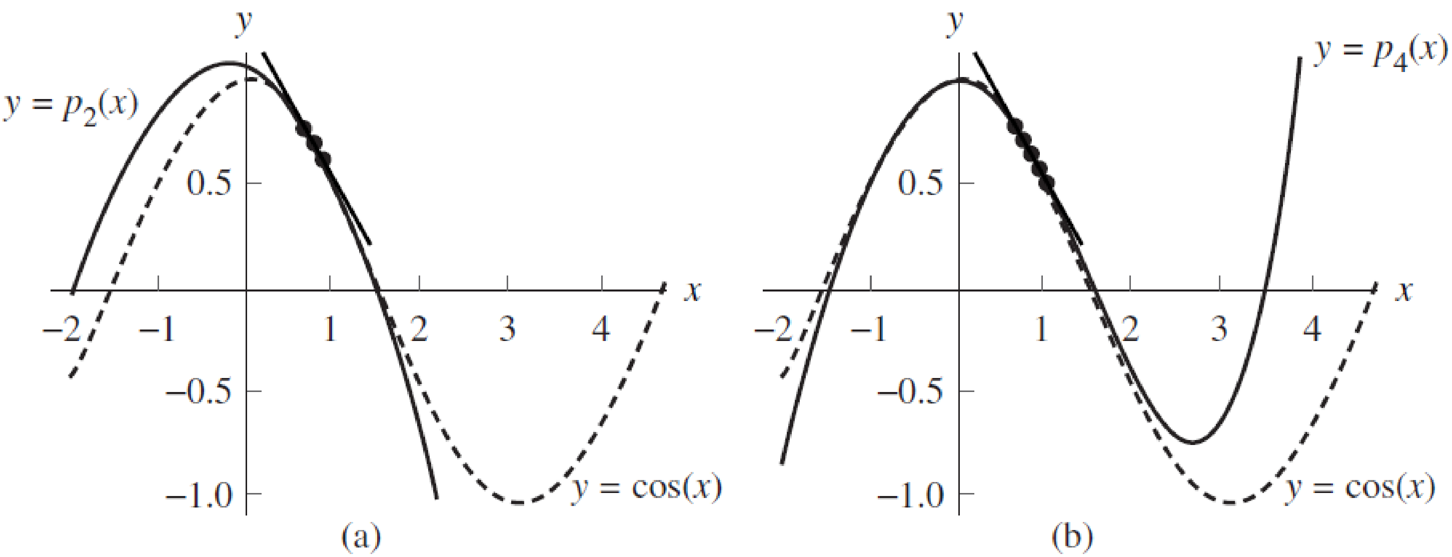

Figure 6.5 (a) The graph of $y = \cos(x)$ and the interpolating polynomial $p_2(x)$ used to estimate $f'(0.8) \approx p_2'(0.8) = -0.716161095$. (b) The graph of $y = \cos(x)$ and the interpolating polynomial $p_4(x)$ used to estimate $f'(0.8) \approx p_4'(0.8) = -0.717353703$.

Richardson's Extrapolation

In this section we emphasize the relationship between formulas (3) and (10). Let $f_k = f(x_k) = f(x_0 + kh)$, and use the notation $D_0(h)$ and $D_0(2h)$ to denote the approximations to $f'(x_0)$ that are obtained from (3) with step sizes h and $2h$, respectively:

(27) $$f'(x_0) \approx D_0(h) + Ch^2$$

and

(28) $$f'(x_0) \approx D_0(2h) + 4Ch^2 \,.$$

If we multiply relation (27) by 4 and subtract relation (28) from this product, then the terms involving C cancel and the result is

(29) $$3f'(x_0) \approx 4D_0(h) - D_0(2h) = \frac{4(f_1 - f_{-1})}{2h} - \frac{f_2 - f_{-2}}{4h}.$$

(30) $$f'(x_0) \approx \frac{4D_0(h) - D_0(2h)}{3} = \frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h}.$$

Richardson's Extrapolation

- The method of obtaining a formula for $f'(x_0)$ of higher order from a formula of lower order is called ***extrapolation***. The proof requires that the error term for (3) can be expanded in a series containing only even powers of h.
- For formula (10), we can apply the same procedure

$$f'(x_0) = \frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h} + \frac{h^4 f^{(5)}(c_1)}{30} \approx D_1(h) + Ch^4 \tag{31}$$

and

$$f'(x_0) = \frac{-f_4 + 8f_2 - 8f_{-2} + f_{-4}}{24h} + \frac{16h^4 f^{(5)}(c_2)}{30} \approx D_1(2h) + 16Ch^4. \tag{32}$$

Suppose that $f^{(5)}(x)$ has one sign and does not change too rapidly; then the assumption that $f^{(5)}(c_1) \approx f^{(5)}(c_2)$ can be used to eliminate the terms involving h^4 in (31) and (32), and the result is

$$f'(x_0) \approx \frac{16D_1(h) - D_1(2h)}{15}. \tag{33}$$

Richardson's Extrapolation

Theorem 6.3 (Richardson's Extrapolation). Suppose that two approximations of order $\boldsymbol{O}(h^{2k})$ for $f'(x_0)$ are $D_{k-1}(h)$ and $D_{k-1}(2h)$ and that they satisfy

$$f'(x_0) = D_{k-1}(h) + c_1 h^{2k} + c_2 h^{2k+2} + \cdots$$

and

$$f'(x_0) = D_{k-1}(2h) + 4^k c_1 h^{2k} + 4^{k+1} c_2 h^{2k+2} + \cdots$$

Then an improved approximation has the form

$$f'(x_0) = D_k(h) + \boldsymbol{O}\left(h^{2k+2}\right) = \frac{4^k D_{k-1}(h) - D_{k-1}(2h)}{4^k - 1} + \boldsymbol{O}\left(h^{2k+2}\right).$$

MATLAB Code

Program 6.1 (Differentiation Using Limits). To approximate $f'(x)$ numerically by generating the sequence

$$f'(x) \approx D_k = \frac{f(x + 10^{-k}h) - f(x - 10^{-k}h)}{2(10^{-k}h)} \quad \text{for } k = 0, \ldots, n$$

until $|D_{n+1} - D_n| \geq |D_n - D_{n-1}|$ or $|D_n - D_{n-1}| <$ tolerance, which is an attempt to find the best approximation $f'(x) \approx D_n$.

```
function [L,n]=difflim(f,x,toler)
%Input - f is the function input as a string 'f'
%      - x is the differentiation point
%      - toler is the tolerance for the error
%Output-L=[H' D' E']:
%          H is the vector of step sizes
%          D is the vector of approximate derivatives
%          E is the vector of error bounds
%       - n is the coordinate of the ``best approximation''
max1=15;
h=1;
H(1)=h;
D(1)=(feval(f,x+h)-feval(f,x-h))/(2*h);
```

MATLAB Code

```
E(1)=0;
R(1)=0;
for n=1:2
   h=h/10;
   H(n+1)=h;
   D(n+1)=(feval(f,x+h)-feval(f,x-h))/(2*h);
   E(n+1)=abs(D(n+1)-D(n));
   R(n+1)=2*E(n+1)/(abs(D(n+1))+abs(D(n))+eps);
end
n=2;
while((E(n)>E(n+1))&(R(n)>toler))&n<max1
   h=h/10;
   H(n+2)=h;
   D(n+2)=(feval(f,x+h)-feval(f,x-h))/(2*h);
   E(n+2)=abs(D(n+2)-D(n+1));
   R(n+2)=2*E(n+2)/(abs(D(n+2))+abs(D(n+1))+eps);
   n=n+1;
end
n=length(D)-1;
L=[H' D' E'];
```

MATLAB Code

Program 6.2 (Differentiation Using Extrapolation). To approximate $f'(x)$ numerically by generating a table of approximations $D(j,k)$ for $k \leq j$, and using $f'(x) \approx D(n,n)$ as the final answer. The approximations $D(j,k)$ are stored in a lower-triangular matrix. The first column is

$$D(j,0) = \frac{f(x+2^{-j}h) - f(x-2^{-j}h)}{2^{-j+1}h}$$

and the elements in row j are

$$D(j,k) = D(j,k-1) + \frac{D(j,k-1) - D(j-1,k-1)}{4^k - 1} \quad \text{for } 1 \leq k \leq j.$$

```
function [D,err,relerr,n]=diffext(f,x,delta,toler)

%Input   -f is the function input as a string 'f'
%        - delta is the tolerance for the error
%        - toler is the tolerance for the relative error
%Output - D is the matrix of approximate derivatives
%        - err is the error bound
```

MATLAB Code

```
%       - relerr is the relative error bound
%       - n is the coordinate of the ``best approximation''

err=1;
relerr=1;
h=1;
j=1;
D(1,1)=(feval(f,x+h)-feval(f,x-h))/(2*h);

while relerr>toler & err>delta &j<12
   h=h/2;
   D(j+1,1)=(feval(f,x+h)-feval(f,x-h))/(2*h);
   for k=1:j
      D(j+1,k+1)=D(j+1,k)+(D(j+1,k)-D(j,k))/((4^k)-1);
   end
   err=abs(D(j+1,j+1)-D(j,j));
   relerr=2*err/(abs(D(j+1,j+1))+abs(D(j,j))+eps);
   j=j+1;
end
[n,n]=size(D);
```

Numerical Differentiation Formulas

More Central-Difference Formulas

- Taylor series can be used to obtain central-difference formulas for the higher derivatives. The popular choices are those of order $\boldsymbol{O}(h^2)$ and $\boldsymbol{O}(h^4)$

- An example

$$f(x+h) = f(x) + hf'(x) + \frac{h^2 f''(x)}{2} + \frac{h^3 f^{(3)}(x)}{6} + \frac{h^4 f^{(4)}(x)}{24} + \cdots$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2 f''(x)}{2} - \frac{h^3 f^{(3)}(x)}{6} + \frac{h^4 f^{(4)}(x)}{24} - \cdots$$

$$f(x+h) + f(x-h) = 2f(x) + \frac{2h^2 f''(x)}{2} + \frac{2h^4 f^{(4)}(x)}{24} + \cdots$$

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - \frac{2h^2 f^{(4)}(x)}{4!}$$

$$- \frac{2h^4 f^{(6)}(x)}{6!} - \cdots - \frac{2h^{2k-2} f^{(2k)}(x)}{(2k)!} - \cdots$$

More Central-Difference Formulas

Table 6.3 Central-Difference Formulas of Order $\boldsymbol{O}(h^2)$

$$f'(x_0) \approx \frac{f_1 - f_{-1}}{2h}$$

$$f''(x_0) \approx \frac{f_1 - 2f_0 + f_{-1}}{h^2}$$

$$f^{(3)}(x_0) \approx \frac{f_2 - 2f_1 + 2f_{-1} - f_{-2}}{2h^3}$$

$$f^{(4)}(x_0) \approx \frac{f_2 - 4f_1 + 6f_0 - 4f_{-1} + f_{-2}}{h^4}$$

Table 6.4 Central-Difference Formulas of Order $\boldsymbol{O}(h^4)$

$$f'(x_0) \approx \frac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h}$$

$$f''(x_0) \approx \frac{-f_2 + 16f_1 - 30f_0 + 16f_{-1} - f_{-2}}{12h^2}$$

$$f^{(3)}(x_0) \approx \frac{-f_3 + 8f_2 - 13f_1 + 13f_{-1} - 8f_{-2} + f_{-3}}{8h^3}$$

$$f^{(4)}(x_0) \approx \frac{-f_3 + 12f_2 - 39f_1 + 56f_0 - 39f_{-1} + 12f_{-2} - f_{-3}}{6h^4}$$

An example

Example 6.4. Let $f(x) = \cos(x)$.

(a) Use formula (6) with $h = 0.1, 0.01,$ and 0.001 and find approximations of $f''(0.8)$. Carry nine decimal places in all calculations.

(b) Compare with the true value $f''(0.80) = -\cos(0.8)$.

(a) The calculation for $h = 0.01$ is

$$f''(0.8) \approx \frac{f(0.81) - 2f(0.80) + f(0.79)}{0.0001}$$
$$\approx \frac{0.689498433 - 2(0.696706709) + 0.703845316}{0.0001}$$
$$\approx -0.696690000.$$

(b) The error in this approximation is -0.000016709. The other calculations are summarized in Table 6.5. The error analysis will illuminate this example and show why $h = 0.01$ was best.

An example

Example 6.4. Let $f(x) = \cos(x)$.

(a) Use formula (6) with $h = 0.1, 0.01,$ and 0.001 and find approximations of $f''(0.8)$. Carry nine decimal places in all calculations.

(b) Compare with the true value $f''(0.80) = -\cos(0.8)$.

Table 6.5 Numerical Approximations to $f''(x)$ for Example 6.4

Step size	Approximation by formula (6)	Error using formula (6)
$h = 0.1$	-0.696126300	-0.000580409
$h = 0.01$	-0.696690000	-0.000016709
$h = 0.001$	-0.696000000	-0.000706709

Error Analysis

Let $f_k = y_k + e_k$, where e_k is the error in computing $f(x_k)$, including noise in measurement and round-off error. Then formula (6) can be written

(7) $$f''(x_0) = \frac{y_1 - 2y_0 + y_{-1}}{h^2} + E(f,h).$$

The error term $E(f,h)$ for the numerical derivative (7) will have a part due to round-off error and a part due to truncation error:

(8) $$E(f,h) = \frac{e_1 - 2e_0 + e_{-1}}{h^2} - \frac{h^2 f^{(4)}(c)}{12} \; .$$

If it is assumed that each error e_k is of the magnitude ϵ, with signs that accumulate errors, and that $|f^{(4)}(x)| \leq M$, then we get the following error bound:

(9) $$|E(f,h)| \leq \frac{4\epsilon}{h^2} + \frac{Mh^2}{12} \; .$$

Error Analysis

If h is small, then the contribution $4\epsilon/h^2$ due to round-off error is large. When h is large, the contribution $Mh^2/12$ is large. The optimal step size will minimize the quantity

(10) $$\mathrm{g}(h) = \frac{4\epsilon}{h^2} + \frac{Mh^2}{12}.$$

Setting $\mathrm{g}'(h) = 0$ results in $-8\epsilon/h^3 + Mh/6 = 0$, which yields the equation $h^4 = 48\epsilon/M$, from which we obtain the optimal value:

(11) $$h = \left(\frac{48\epsilon}{M}\right)^{1/4}.$$

When formula (11) is applied to Example 6.4, use the bound $|f^{(4)}(x)| \leq |\cos(x)| \leq 1 = M$ and the value $\epsilon = 0.5 \times 10^{-9}$. The optimal step size is $h = (24 \times 10^{-9}/1)^{1/4} = 0.01244666$, and we see that $h = 0.01$ was closest to the optimal value.

Error Analysis

- Since the portion of the error due to round off is inversely proportional to the square of h, this term grows when h gets small. This is sometimes referred to as the ***step-size dilemma***.

- One partial solution to this problem is to use a formula of higher order so that a larger value of h will produce the desired accuracy.

(12) $$f''(x_0) = \frac{-f_2 + 16f_1 - 30f_0 + 16f_{-1} - f_{-2}}{12h^2} + E(f,h)\,.$$

The error term for (12) has the form

(13) $$E(f,h) = \frac{16\epsilon}{3h^2} + \frac{h^4 f^{(6)}(c)}{90}\ ,$$

where c lies in the interval $[x - 2h, x + 2h]$. A bound for $|E(f,h)|$ is

Error Analysis

$$|E(f,h)| \le \frac{16\epsilon}{3h^2} + \frac{h^4 M}{90} \ ,$$

where $|f^{(6)}(x)| \le M$. The optimal value for h is given by the formula

$$h = \left(\frac{240\epsilon}{M}\right)^{1/6} .$$

Example 6.5. Let $f(x) = \cos(x)$.

(a) Use formula (12) with $h = 1.0, 0.1,$ and 0.01 and find approximations to $f''(0.8)$. Carry nine decimal places in all the calculations.

(b) Compare with the true value $f''(0.8) = -\cos(0.8)$.

(c) Determine the optimal step size.

Error Analysis

(a) The calculation for $h = 0.1$ is

$$
\begin{aligned}
f''(0.8) \\
&\approx \frac{-f(1.0) + 16f(0.9) - 30f(0.8) + 16f(0.7) - f(0.6)}{0.12} \\
&\approx \frac{-0.540302306 + 9.945759488 - 20.90120127 + 12.23747499 - 0.825335615}{0.12} \\
&\approx -0.696705958.
\end{aligned}
$$

(b) The error in this approximation is -0.000000751. The other calculation are summarized in Table 6.6

Table 6.6 Numerical Approximations to $f''(x)$ for Example 6.5

Step size	Approximation by formula (12)	Error using formula (12)
$h = 1.0$	-0.689625413	-0.007081296
$h = 0.1$	-0.696705958	-0.000000751
$h = 0.01$	-0.696690000	-0.000016709

(c) When formula (15) is applied, we can use the bound $|f^{(6)}(x)| \leq |\cos(x)| \leq 1 = M$ and the value $\epsilon = 0.5 \times 10^{-9}$. These values give the optimal step size $h = (120 \times 10^{-9}/1)^{1/6} = 0.070231219$.

Error Analysis

- Generally, if numerical differentiation is performed, only about half the accuracy of which the computer is capable is obtained.
- This severe loss of significant digits will almost always occur unless we are fortunate to find a step size that is optimal.
- Hence we must always proceed with caution when numerical differentiation is performed.
- The difficulties are more pronounced when working with experimental data, where the function values have been rounded to only a few digits.
- If a numerical derivative must be obtained from data, we should consider curve fitting, by using least-squares techniques, and differentiate the formula for the curve.

Differentiation of the Lagrange Polynomial

- If the function must be evaluated at abscissas that lie on one side of x_0, the central difference formulas cannot be used.

- Formulas for equally spaced abscissas that lie to the right (or left) of x_0 are called forward (or backward) -difference formulas.

- These formulas can be derived by differentiation of the Lagrange interpolation polynomial.

Example 6.6. Derive the formula

$$f''(x_0) \approx \frac{2f_0 - 5f_1 + 4f_2 - f_3}{h^2} .$$

Start with the Lagrange interpolation polynomial for $f(t)$ based on the four points x_0, x_1, x_2, and x_3.

Differentiation of the Lagrange Polynomial

$$f(t) \approx f_0 \frac{(t-x_1)(t-x_2)(t-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} + f_1 \frac{(t-x_0)(t-x_2)(t-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}$$
$$+f_2 \frac{(t-x_0)(t-x_1)(t-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} + f_3 \frac{(t-x_0)(t-x_1)(t-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} .$$

Differentiate the products in the numerators twice and get

$$f''(t)$$
$$\approx f_0 \frac{2((t-x_1)+(t-x_2)+(t-x_3))}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} + f_1 \frac{2((t-x_0)+(t-x_2)+(t-x_3))}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}$$
$$+f_2 \frac{2((t-x_0)+(t-x_1)+(t-x_3))}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}$$
$$+ f_3 \frac{2((t-x_0)+(t-x_1)+(t-x_2))}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} .$$

Differentiation of the Lagrange Polynomial

Then substitution of $t = x_0$ and the fact that $x_i - x_j = (i-j)h$ produces

$$
\begin{aligned}
f''(x_0) \approx\ & f_0 \frac{2\big((x_0 - x_1) + (x_0 - x_2) + (x_0 - x_3)\big)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} \\
& + f_1 \frac{2((x_0 - x_0) + (x_0 - x_2) + (x_0 - x_3))}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} \\
& + f_2 \frac{2\big((x_0 - x_0) + (x_0 - x_1) + (x_0 - x_3)\big)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} \\
& + f_3 \frac{2((x_0 - x_0) + (x_0 - x_1) + (x_0 - x_2))}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} \\
& = f_0 \frac{2((-h) + (-2h) + (-3h))}{(-h)(-2h)(-3h)} + f_1 \frac{2(0 + (-2h) + (-3h))}{(h)(-h)(-2h)} \\
& + f_2 \frac{2((0) + (-h) + (-3h))}{(2h)(h)(-h)} + f_3 \frac{2((0) + (-h) + (-2h))}{(3h)(2h)(h)} \\
& = f_0 \frac{-12h}{-6h^3} + f_1 \frac{-10h}{2h^3} + f_2 \frac{-8h}{-2h^3} + f_3 \frac{-6h}{6h^3} = \frac{2f_0 - 5f_1 + 4f_2 - f_3}{h^2}\ .
\end{aligned}
$$

Differentiation of the Lagrange Polynomial

Example 6.7. Derive the formula

$$f'''(x_0) \approx \frac{-5f_0 + 18f_1 - 24f_2 + 14f_3 - 3f_4}{2h^3}.$$

Start with the Lagrange interpolation polynomial for $f(t)$ based on the five points x_0, x_1, x_2, x_3, and x_4.

$$\begin{aligned}
f(t) \approx f_0 &\frac{(t-x_1)(t-x_2)(t-x_3)(t-x_4)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)(x_0-x_4)} \\
&+ f_1 \frac{(t-x_0)(t-x_2)(t-x_3)(t-x_4)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)(x_1-x_4)} \\
&+ f_2 \frac{(t-x_0)(t-x_1)(t-x_3)(t-x_4)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)(x_2-x_4)} \\
&+ f_3 \frac{(t-x_0)(t-x_1)(t-x_2)(t-x_4)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)(x_3-x_4)} \\
&+ f_4 \frac{(t-x_0)(t-x_1)(t-x_2)(t-x_3)}{(x_4-x_0)(x_4-x_1)(x_4-x_2)(x_4-x_3)}
\end{aligned}$$

Differentiate the numerators three times, then use the substitution $x_i - x_j = (i-j)h$ in the denominators and get

Differentiation of the Lagrange Polynomial

$$
\begin{aligned}
f'''(t) \approx f_0 &\frac{6((t-x_1)+(t-x_2)+(t-x_3)+(t-x_4))}{(-h)(-2h)(-3h)(-4h)} \\
&+ f_1 \frac{6((t-x_0)+(t-x_2)+(t-x_3)+(t-x_4))}{(h)(-h)(-2h)(-3h)} \\
&+ f_2 \frac{6((t-x_0)+(t-x_1)+(t-x_3)+(t-x_4))}{(2h)(h)(-h)(2h)} \\
&+ f_3 \frac{6((t-x_0)+(t-x_1)+(t-x_2)+(t-x_4))}{(3h)(2h)(h)(-h)} \\
&+ f_4 \frac{6((t-x_0)+(t-x_1)+(t-x_2)+(t-x_3))}{(4h)(3h)(2h)(h)}.
\end{aligned}
$$

Then substitution of $t = x_0$ in the form $t - x_j = x_0 - x_j = -jh$ produces

$$
\begin{aligned}
f'''(x_0) \approx f_0 &\frac{6((-h)+(-2h)+(-3h)+(-4h))}{24h^4} + f_1 \frac{6((0)+(-2h)+(-3h)+(-4h))}{-6h^4} \\
&+ f_2 \frac{6((0)+(-h)+(-3h)+(-4h))}{4h^4} + f_3 \frac{6((0)+(-h)+(-2h)+(-4h))}{-6h^4} \\
&+ f_4 \frac{6((0)+(-h)+(-2h)+(-3h))}{24h^4} \\
&= f_0 \frac{-60h}{24h^4} + f_1 \frac{54h}{6h^4} + f_2 \frac{-48h}{4h^4} + f_3 \frac{42h}{6h^4} + f_4 \frac{-36h}{24h^4} \\
&= \frac{-5 f_0 + 18 f_1 - 24 f_2 + 14 f_3 - 3 f_4}{2h^3},
\end{aligned}
$$

and the formula is established. ∎

Differentiation of the Lagrange Polynomial

Table 6.7 Forward- and Backward-Difference Formulas of Order $\boldsymbol{O}(h^2)$

$f'(x_0) \approx \dfrac{-3f_0 + 4f_1 - f_2}{2h}$	(forward difference)
$f'(x_0) \approx \dfrac{3f_0 - 4f_{-1} + f_{-2}}{2h}$	(backward difference)
$f''(x_0) \approx \dfrac{2f_0 - 5f_1 + 4f_2 - f_3}{h^2}$	(forward difference)
$f''(x_0) \approx \dfrac{2f_0 - 5f_{-1} + 4f_{-2} - f_{-3}}{h^2}$	(backward difference)
$f^{(3)}(x_0) \approx \dfrac{-5f_0 + 18f_1 - 24f_2 + 14f_3 - 3f_4}{2h^3}$	
$f^{(3)}(x_0) \approx \dfrac{5f_0 - 18f_{-1} + 24f_{-2} - 14f_{-3} + 3f_{-4}}{2h^3}$	
$f^{(4)}(x_0) \approx \dfrac{3f_0 - 14f_1 + 26f_2 - 24f_3 + 11f_4 - 2f_5}{h^4}$	
$f^{(4)}(x_0) \approx \dfrac{3f_0 - 14f_{-1} + 26f_{-2} - 24f_{-3} + 11f_{-4} - 2f_{-5}}{h^4}$	

Differentiation of the Newton Polynomial

- Here we show the relationship between the three formulas of order $\boldsymbol{O}(h^2)$ for approximating $f'(x_0)$, and a general algorithm is given for computing the numerical derivative.
- Start with Newton polynomial $P(t)$ of degree $N = 2$ that approximates $f(t)$ using nodes t_0, t_1, and t_2

(16) $$P(t) = a_0 + a_1(t - t_0) + a_2(t - t_0)(t - t_1)\ ,$$

where $a_0 = f(t_0), a_1 = (f(t_1) - f(t_0))/(t_1 - t_0)$, and

$$a_2 = \frac{\dfrac{f(t_2) - f(t_1)}{t_2 - t_1} - \dfrac{f(t_1) - f(t_0)}{t_1 - t_0}}{t_2 - t_0}\ .$$

The derivative of $P(t)$ is

(17) $$P'(t) = a_1 + a_2\big((t - t_0) + (t - t_1)\big),$$

and when it is evaluated at $t = t_0$, the result is

(18) $$P'(t_0) = a_1 + a_2(t_0 - t_1) \approx f'(t_0).$$

Differentiation of the Newton Polynomial

- Observe that the nodes $\{t_k\}$ do not need to be equally spaced for formulas (16) through (18) to hold. Choosing the abscissas in different orders will produce different formulas for approximating $f'(x)$.

Case (*i*): If $t_0 = x, t_1 = x + h$, and $t_2 = x + 2h$, then

$$a_1 = \frac{f(x+h) - f(x)}{h},$$

$$a_2 = \frac{f(x) - 2f(x+h) + f(x+2h)}{2h^2}.$$

When these values are substituted into (18), we get

$$P'(x) = \frac{f(x+h) - f(x)}{h} + \frac{-f(x) + 2f(x+h) - f(x+2h)}{2h}.$$

This is simplified to obtain

$$P'(x) = \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} \approx f'(x).$$

which is the second-order forward-difference formula for $f'(x)$.

Differentiation of the Newton Polynomial

Case (*ii*): If $t_0 = x, t_1 = x + h$, and $t_2 = x - h$, then

$$a_1 = \frac{f(x+h) - f(x)}{h} \ ,$$

$$a_2 = \frac{f(x+h) - 2f(x) + f(x-h)}{2h^2}$$

When these values are substituted into (18), we get

$$P'(x) = \frac{f(x+h) - f(x)}{h} + \frac{-f(x+h) - 2f(x) - f(x-h)}{2h} \ .$$

This is simplified to obtain

$$P'(x) = \frac{f(x+h) - f(x-h)}{2h} \approx f'(x).$$

which is the second-order central-difference formula for $f'(x)$.

Differentiation of the Newton Polynomial

Case (*iii*): If $t_0 = x, t_1 = x - h$, and $t_2 = x - 2h$, then

$$a_1 = \frac{f(x) - f(x-h)}{h} \ ,$$

$$a_2 = \frac{f(x) - 2f(x-h) + f(x-2h)}{2h^2}$$

These values are substituted into (18) and simplified to obtain

$$P'(x) = \frac{3f(x) - 4f(x-h) + f(x-2h)}{2h} \approx f'(x).$$

which is the second-order backward-difference formula for $f'(x)$.

Differentiation of the Newton Polynomial

The Newton polynomial $P(t)$ of degree N that approximates $f(t)$ using the nodes $t_0, t_1, \ldots, t_N$ is

$$P(t) = a_0 + a_1(t - t_0) + a_2(t - t_0)(t - t_1)$$
$$+a_3(t - t_0)(t - t_1)(t - t_2) + \cdots + a_N(t - t_0)\cdots(t - t_{N-1}).$$

The derivative of $P(t)$ is

$$P'(t) = a_1 + a_2((t - t_0) + (t - t_1))$$
$$+a_3((t - t_0)(t - t_1) + (t - t_0)(t - t_2) + (t - t_1)(t - t_2).$$
$$+ \cdots + a_N \sum_{k=0}^{N-1} \prod_{\substack{j=0 \\ j\neq k}}^{N-1} (t - t_j) \ .$$

When $P'(t)$ is evaluated at $t = t_0$, several of the terms in the summation are zero, and $P'(t_0)$ has the simpler form

$$P'(t_0) = a_1 + a_2(t_0 - t_1) + a_3(t_0 - t_1)(t_0 - t_2) + \cdots$$
$$+a_N(t_0 - t_1)(t_0 - t_2)(t_0 - t_3)\cdots(t_0 - t_{N-1}) \ . \qquad (24)$$

Differentiation of the Newton Polynomial

The kth partial sum on the right side of equation (24) is the derivative of the Newton polynomial of degree k based on the first k nodes. If

$$|t_0 - t_1| \le |t_0 - t_2| \le \cdots \le |t_0 - t_N|, \qquad \text{and if} \quad \{(t_j, 0)\}_{j=0}^{N}$$

forms a set of $N + 1$ equally spaced points on the real axis, the kth partial sum is an approximation to $f'(t_0)$ of order $O\left(h^{k-1}\right)$.

- Suppose that $N = 5$. If the five nodes are $t_k = x + hk$ for $k = 0, 1, 2, 3,$ and 4, then (24) is an equivalent way to compute the forward-difference formula for $f'(x)$ of order $\boldsymbol{O}(h^4)$.
- If the five nodes $\{t_k\}$ are chosen to be $t_0 = x$, $t_1 = x + h$, $t_2 = x - h$, $t_3 = x + 2h$, and $t_4 = x - 2h$, then (24) is the central-difference formula for $f'(x)$ of order $\boldsymbol{O}(h^4)$.
- When the five nodes are $t_k = x - kh$, then (24) is the backward-difference formula for $f'(x)$ of order $\boldsymbol{O}(h^4)$

Matlab code

Program 6.3 (Differentiation Based on $N+1$ Nodes). To approximate $f'(x)$ numerically by constructing the Nth-degree Newton polynomial

$$P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2) + \cdots + a_N(x - x_0)\cdots(x - x_{N-1})$$

and using $f'(x_0) \approx P'(x_0)$ as the final answer. The method must be used at x_0. The points can be rearranged $\{x_k, x_0, \ldots, x_{k-1}, x_{k+1}, \ldots, x_N\}$ to compute $f'(x_k) \approx P'(x_k)$.

```
function [A,df]=diffnew(X,Y)

%Input  - X is the 1xn abscissa vector
%       - Y is the 1xn ordinate vector
%Output - A is the 1xn vector containing the coefficients of
%         the Nth-degree Newton polynomial
%       - df is the approximate derivative

A=Y;
N=length(X);

for j=2:N
   for k=N:-1:j
      A(k)=(A(k)-A(k-1))/(X(k)-X(k-j+1));
   end
end

x0=X(1);
df=A(2);
prod=1;
n1=length(A)-1;

for k=2:n1
   prod=prod*(x0-X(k));
   df=df+prod*A(k+1);
end
```