

Chapter 7

Numerical Differentiation and Numerical Integration

*** 3/1/13 EC

What's Ahead

- A Case Study on Numerical Differentiation: Velocity Gradient for Blood Flow
- Finite Difference Formulas and Errors
- Interpolation-Based Formulas and Errors
- Richardson Extrapolation Technique
- Finite Difference and Interpolation-based Formulas for Second Derivatives
- Finite Difference Formulas for Partial Derivatives

PART I. Numerical Differentiation

7.1 Numerical Differentiation and Applications

In an elementary calculus course, the students learn the concept of the derivative of a function $y = f(x)$, denoted by $f'(x)$, $\frac{dy}{dx}$ or $\frac{d}{dx}(f(x))$, along with various scientific and engineering applications.

These applications include:

- Velocity and acceleration of a moving object
- Current, power, temperature, and velocity gradients
- Rates of growth and decay in a population
- Marginal cost and marginal profits in economics, etc.

The need for **numerical differentiation** arises from the fact that very often, either

- $f(x)$ is not explicitly given and only the values of $f(x)$ at certain discrete points are known

or

- $f'(x)$ is difficult to compute analytically.

We will learn various ways to compute $f'(x)$ numerically in this Chapter.

We start with the following **biological application**.

A Case Study on Numerical Differentiation

Velocity Gradient for Blood Flow

Consider the blood flow through an artery or vein. It is known that the nature of viscosity dictates a flow profile, where the velocity v increases toward the center of the tube and is zero at the wall, as illustrated in the following diagram:

Let v be the velocity of blood that flows along a blood vessel which has radius R and length l at a distance r from the central axis. Let ΔP = Pressure difference between the ends of the tube and η = Viscosity of blood.

From the **law of laminar flow** which gives the relationship between v and r , we have

$$v(r) = v_m \left(1 - \frac{r^2}{R^2} \right) \quad (7.1)$$

where

$$v_m = \frac{1}{4\eta} \frac{\Delta P}{l} R^2 \text{ is the } \mathbf{maximum\ velocity}.$$

Substituting the expression for v_m in (7.1), we obtain

$$v(r) = \frac{1}{4\eta} \frac{\Delta P}{l} (R^2 - r^2) \quad (7.2)$$

Thus, if ΔP and l are constant, then the velocity v of the blood flow is a function of r in $[0, R]$.

In an experimental set up, one then can measure velocities at several different values of r , given η , ΔP , l and R .

The problem of interest is now to **compute the velocity gradient** (that is, $\frac{dv}{dr}$) **from** $r = r_1$ to $r = r_2$. We will consider this problem later with numerical values.

7.2 Problem Statement

Numerical Differentiation Problem

Given the functional values, $f(x_0), f(x_1), \dots, f(x_n)$, of a function $f(x)$ which is not explicitly known, at the points x_0, x_1, \dots, x_n in $[a, b]$ or a differentiable function $f(x)$ on $[a, b]$.

Find an approximate value of $f'(x)$ for $a < x < b$.

7.2.1 Finite Difference Formulas for Derivatives

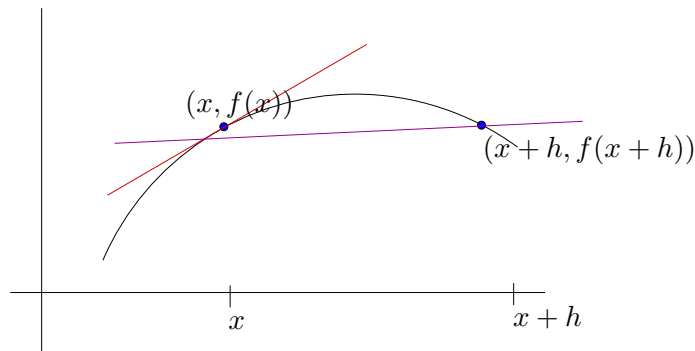
The derivative of a function $f(x)$ is defined as:

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

Thus, it is the slope of the tangent line at the point $(x, f(x))$. The **difference quotient** (DQ)

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

is the slope of the secant line passing through $(x, f(x))$ and $(x + h, f(x + h))$.



As h gets smaller and smaller, the difference quotient gets closer and closer to $f'(x)$. However, if h is too small, then the round-off error becomes large, yielding an inaccurate value of the derivative.

In any case, if the DQ is taken as an approximate value of $f'(x)$, then it is called **two-point forward difference formula (FDF)** for $f'(x)$.

Thus, two-point backward difference and two-point central difference formulas, are similarly defined, respectively, in terms of the functional values $f(x - h)$ and $f(x)$, and $f(x - h)$ and $f(x + h)$.

Two-point Forward Difference Formula (FDF):

$$f'(x) \approx \frac{f(x + h) - f(x)}{h} \quad (7.3)$$

Two-point Backward Difference Formula (BDF):

$$f'(x) \approx \frac{f(x) - f(x - h)}{h} \quad (7.4)$$

Two-point Central Difference Formula (CDF):

$$f'(x) \approx \frac{f(x + h) - f(x - h)}{2h} \quad (7.5)$$

Example 7.1

Given the following table, where the functional values correspond to $f(x) = x \ln x$:

x	$f(x)$
1	0
2	1.3863
3	3.2958

Approximate $f'(2)$ by using two-point (i) FDF, (ii) BDF, and (iii) CDF. (Note that $f'(x) = 1 + \ln x$; $f'(2) = 1 + \ln 2 = 1.6931$.)

Input Data: $x = 2, h = 1, x + h = 3$.

Solution.

Two-point FDF: $f'(x) \approx \frac{f(x+h)-f(x)}{h} = \frac{f(3)-f(2)}{1} = 1.9095$.

Absolute Error: $|(1 + \ln 2) - 1.9095| = |1.6931 - 1.9095| = 0.2164$.

Two-point BDF: $\frac{f(x)-f(x-h)}{h} = \frac{f(2)-f(1)}{h} = 1.3863$.

Absolute Error: $|1.6931 - 1.3863| = 0.3068$.

Two-point CDF: $\frac{f(x+h)-f(x-h)}{2h} = \frac{f(3)-f(1)}{2} = \frac{3.2958}{2} = 1.6479$.

Absolute Error: $|1.6931 - 1.6479| = 0.0452$.

Remarks: The above example shows that two-point CDF is more accurate than two-point FDF and BDF. The reason will be clear from our discussion on truncation errors in the next section.

Derivations of the Two-Point Finite Difference Formulas and Errors: Taylor Series Approach

In this section, we will show how to derive the two-point difference formulas and the truncation errors associated with them using the **Taylor series** and state without proofs, the three-point FDF and BDF. The derivatives of these and other higher-order formulas and their errors will be given in Section 7.2.3, using Lagrange interpolation techniques.

Consider the **two-term Taylor series expansion** of $f(x)$ about the points $x + h$ and $x - h$, respectively:

$$f(x + h) = f(x) + hf'(x) + \frac{h^2}{2}f''(\xi_0), \text{ where } x < \xi_0 < x + h \quad (7.6)$$

and

$$f(x - h) = f(x) - hf'(x) + \frac{h^2}{2}f''(\xi_1), \text{ where } x - h < \xi_1 < x \quad (7.7)$$

Solving for $f'(x)$ from (7.6), we get

$$f'(x) = \left[\frac{f(x+h) - f(x)}{h} \right] - \frac{h}{2} f''(\xi_0) \quad (7.8)$$

- The term within brackets on the right-hand side of (7.8) is the **two-point FDF**.
- The second term (remainder) on the right-hand side of (7.8) is the **truncation error for two-point FDF**.

Similarly, solving for $f'(x)$ from (7.7), we get

$$f'(x) = \frac{f(x) - f(x-h)}{h} + \frac{h}{2} f''(\xi_1) \quad (7.9)$$

- The first term within brackets on the right-hand side of (7.9) is the **two-point BDF**.
- The second term (remainder) on the right-hand side of (7.9) is the **truncation error for two-point BDF**.

Assume that $f''(x)$ is continuous. Consider this time a **three-term Taylor series expansion** of $f(x)$ about the points $(x+h)$ and $(x-h)$:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(x) + \frac{h^3}{3!} f'''(\xi_2) \quad (7.10)$$

where $x < \xi_2 < x+h$, and

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2} f''(x) - \frac{h^3}{3!} f'''(\xi_3) \quad (7.11)$$

where $x-h < \xi_3 < x$.

Subtracting (7.11) from (7.10), we obtain

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{h^3}{3!} [f'''(\xi_2) + f'''(\xi_3)] \quad (7.12)$$

Solving (7.12) for $f'(x)$, we get

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{12} [f'''(\xi_2) + f'''(\xi_3)] \quad (7.13)$$

To simplify the expression within the brackets in (7.13), we need the following theorem from Calculus:

Intermediate Value Theorem for Continuous Functions (IVT)

Let

- (i) $f(x)$ be a continuous function on $[a, b]$,
- (ii) x_1, x_2, \dots, x_n be n points in $[a, b]$,
- (iii) c_1, c_2, \dots, c_n be n real numbers - all of the same sign.

Then

$$\sum_{i=1}^n f(x_i)c_i = f(c) \sum_{i=1}^n c_i, \quad \text{for some } c \text{ in } [a, b].$$

We now apply IVT to $f'''(x)$ in (7.13), with $n = 2$ and $c_1 = c_2 = 1$. For this, we note that

- (i) $f'''(x)$ is continuous on $[x - h, x + h]$ (*Hypothesis (i) of IVT is satisfied*)
- (ii) $c_1 = c_2 = 1$ are two possible numbers (*Hypothesis (iii) is satisfied*)
and
- (iii) ξ_1 and ξ_2 are two numbers in $[x - h, x + h]$ (*Hypothesis (ii) is satisfied*).

Thus, by IVT, there exists a number ξ in $[x - h, x + h]$ such that

$$f'''(\xi_2) + f'''(\xi_3) = 2f'''(\xi) \quad (\text{Note that } c_1 = c_2 = 1).$$

Thus, (7.13) becomes

$$f'(x) = \left[\frac{f(x+h) - f(x-h)}{2h} \right] - \frac{h^2}{6} f'''(\xi) \quad (7.14)$$

The term within brackets on the right-hand side of (7.14) is the **two-point CDF**, and the term $-\frac{h^2}{6} f'''(\xi)$ is the **truncation error for two-point CDF**.

Errors for Two-point Difference Formulas:

$$\left\{ \begin{array}{ll} \text{Error for Two-point FDF: } & \frac{h}{2} f''(\xi_0), \quad x < \xi_0 < x + h \\ \text{Error for Two-point BDF: } & \frac{h}{2} f''(\xi_1), \quad x - h < \xi_1 < x \\ \text{Error for Two-point CDF: } & \frac{h^2}{6} f'''(\xi), \quad x - h < \xi < x + h \end{array} \right.$$

Two-Point FDF and BDF Versus Two-Point CDF

- Two-point FDF and BDF are $O(h)$ (they are first-order approximations).
- Two-point CDF are $O(h^2)$ (this is a second-order approximation).

It is now clear why two-point CDF is more accurate than both two-point FDF and BDF. This is because, both two-point FDF and BDF are $O(h)$ while two-point CDF is $O(h^2)$. Note that Example 7.1 supports this statement.

7.2.2 Three-point and Higher Order Formulas for $f'(x)$: Lagrange Interpolation Approach

Three-point and higher-order derivative formulas and their truncation errors can be derived in the similar way as in the last section. Three-point FDF and BDF approximate $f'(x)$ in terms of the functional values at three points: $x, x+h$, and $x+2h$ for FDF and $x, x-h, x-2h$ for BDF, respectively.

Three-point FDF for $f'(x)$: $f'(x) \approx \frac{-3f(x)+4f(x+h)-f(x+2h)}{2h}$

$$\begin{array}{ccccc} X & - & - & - & - & X & - & - & - & - & X \\ x & & & & & x+h & & & & & x+2h \end{array}$$

Three-point BDF for $f'(x)$: $f'(x) \approx \frac{f(x-2h)-4f(x-h)+3f(x)}{2h}$

$$\begin{array}{ccccc} X & - & - & - & - & - & X & - & - & - & - & - & X \\ x-2h & & & & & & x-h & & & & & & x \end{array}$$

The derivations and error formulas of these and other higher-order approximations are given in the next section.

The difference formulas are simple to use but they are only good for approximating $f'(x)$ where x is one of the tabulated points x_0, x_1, \dots, x_n . On the other hand, if x is a nontabulated point in $[a, b]$, and $f'(x)$ is sought at that point, then the simplest thing to do is to:

- Find an interpolating polynomial $P_n(x)$ passing through x_0, x_2, \dots, x_n (we will use Lagrange interpolations here).
- Accept $P'_n(x)$ as an approximation to $f'(x)$.

As we will see later, if x coincides with one of the points x_0, x_1, \dots, x_n , then we can recover some of the finite difference formulas of the last section, as special cases.

Let $(x_0, f_0), (x_1, f_1), \dots, (x_n, f_n)$ be $(n+1)$ distinct points in $[a, b]$ and $a = x_0 < x_1 < x_2 \dots < x_{n-1} < x_n = b$. Then recall the Lagrange interpolating polynomial $P_n(x)$ of degree at most, n is given by:

$$P_n(x) = L_0(x)f_0 + L_1(x)f_1 + \dots + L_nf_n \quad (7.15)$$

where

$$L_i(x) = \frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}, \quad i = 0, 1, \dots, n. \quad (7.16)$$

So,

$$P'_n(x) = L'_0(x)f_0 + L'_1(x)f_1 + \dots + L'_n(x)f_n. \quad (7.17)$$

Thus, the derivative of $f(x)$ at any point \bar{x} (tabulated or nontabulated) can be approximated by $P'_n(\bar{x})$. Obviously, the procedure is quite tedious as it requires computation of all the Lagrangian polynomials $L_k(x)$ and their derivatives at $x = \bar{x}$. We will derive below the derivative formulas in three special cases: $n = 1$, $n = 2$ and $n = 4$, which are commonly used. *These formulas will become the respective difference formulas in the special case when $\bar{x} = x$ is a tabulated point.* In order to distinguish these formulas from the corresponding finite difference formulas, these will be called, respectively, **two-point**, **three-point**, and **five-point** formulas.

n=1 (Two-point formula and Two-point FDF and BDF) Here the two tabulated points are x_0 and x_1 .

$$\begin{aligned} L_0(x) &= \frac{x-x_1}{x_0-x_1}, & L_1(x) &= \frac{x-x_0}{x_1-x_0} \\ P_1(x) &= f_0L_0(x) + f_1L_1(x) \\ &= f_0\frac{x-x_1}{x_0-x_1} + f_1\frac{x-x_0}{x_1-x_0} \\ &= \frac{-f_0(x-x_1)+f_1(x-x_0)}{x_1-x_0} = \frac{f_1(x-x_0)-f_0(x-x_1)}{x_1-x_0} \\ P'_1(x) &= \frac{(x_1-x_0)(f_1-f_0)+[f_1(x-x_0)-f_0(x-x_1)]x_0}{(x_1-x_0)^2} && \text{(using Quotient Rule)} \\ &= \frac{(x_1-x_0)(f_1-f_0)}{(x_1-x_0)^2} = \frac{f_1-f_0}{x_1-x_0} \end{aligned}$$

This gives us the **two-point formula** for $f'(x)$: $f'(x) \approx \frac{f_1-f_0}{x_1-x_0}$

Setting $x_0 = x$ and $x_1 = x + h$, we have the **two-point FDF**.

Similarly, setting $x_0 = x$ and $x_1 = x - h$, we have the **two-point BDF**.

Summarizing:

$$\begin{aligned} \text{Two-Point Formula:} \quad & f'(x) \approx \frac{f_1 - f_0}{x_1 - x_0} \\ \left\{ \begin{array}{l} \text{Two-Point FDF: } f'(x) \approx \frac{f(x+h) - f(x)}{h} \\ \text{Two-Point BDF: } f'(x) \approx \frac{f(x) - f(x-h)}{h} \end{array} \right. \end{aligned}$$

n=2 (Three-point Formula and Three-point FDF and BDF)

$$\begin{array}{ccccc} X & - & - & - & - & X & - & - & - & - & X \\ x_0 & & & & & x_1 & & & & & x_2 \end{array}$$

$$\begin{aligned} P_2(x) &= f_0 L_0(x) + f_1 L_1(x) + f_2 L_2(x). \\ L_0(x) &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}, \\ L_1(x) &= \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}, \\ \text{and} \\ L_2(x) &= \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}. \end{aligned} \tag{7.18}$$

$$\begin{aligned} P'_2(x) &= f_0 L'_0(x) + f_1 L'_1(x) + f_2 L'_2(x) \\ L'_0(x) &= \frac{d}{dx}(L_0(x)) = \frac{d}{dx} \left[\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} \right] \\ &= \frac{2x-x_1-x_2}{(x_0-x_1)(x_0-x_2)} \end{aligned} \tag{7.19}$$

Similarly, $L'_1(x) = \frac{2x-x_0-x_2}{(x_1-x_0)(x_1-x_2)}$ and $L'_2(x) = \frac{2x-x_0-x_1}{(x_2-x_0)(x_2-x_1)}$.

Thus, $P'_2(x) = f_0 \left[\frac{2x-x_1-x_2}{(x_0-x_1)(x_0-x_2)} \right] + f_1 \left[\frac{2x-x_0-x_2}{(x_1-x_0)(x_1-x_2)} \right] + f_2 \left[\frac{2x-x_0-x_1}{(x_2-x_0)(x_2-x_1)} \right]$, which is a **three-point formula** for $f'(x)$.

- Setting $x_0 = x$, $x_1 = x + h$, and $x_2 = x + 2h$, we obtain **three-point FDF**.
- Setting $x_0 = x$, $x_1 = x - h$, and $x_2 = x - 2h$, we obtain **three-point BDF**.

Summarizing:

Three-point Formula:

$$f'(x) \approx f_0 \left[\frac{2x-x_1-x_2}{(x_0-x_1)(x_0-x_2)} \right] + f_1 \left[\frac{2x-x_0-x_2}{(x_1-x_0)(x_1-x_2)} \right] + f_2 \left[\frac{2x-x_0-x_1}{(x_2-x_0)(x_2-x_1)} \right]. \tag{7.20}$$

$$\begin{cases} \text{Three-point FDF: } f'(x) \approx [-3f(x) + 4f(x+h) - f(x+2h)]/2h \\ \text{Three-point BDF: } f'(x) \approx \frac{1}{2h}[f(x-2h) - 4f(x-h) + 3f(x)] \end{cases}$$

n=4 Five-point formula and associated **four-point CDF** difference formula can similarly be obtained [Exercise]. We list below the **five-point FDF** and **four-point CDF**, for easy references:

Five-Point FDF:

$$f'(x) \approx \frac{1}{12h} [-25f(x) + 48f(x+h) - 36f(x+2h) + 16f(x+3h) - 3f(x+4h)]. \quad (7.21)$$

Four-point CDF:

$$f'(x) \approx \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h}. \quad (7.22)$$

Note: Analogous to two-point CDF, we call the above formula as **four-point CDF**, because the function value $f(x)$ does not appear in the above formula. It uses only four function values: $f(x-2h)$, $f(x-h)$, $f(x+h)$, and $f(x+2h)$.

Errors in Polynomial Approximations to $f'(x)$

Recall that the error term in $(n+1)$ point polynomial interpolation of $f(x)$ is given by (**Theorem 6.7**):

$$E_n(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_n)}{(n+1)!} f^{(n+1)}(\xi_x), \text{ where } x_0 < \xi_x < x_n. \quad (7.23)$$

Differentiating $E_n(x)$ with respect to x and remembering that ξ_x is a function of x , we obtain

$$E'_n(x) = \frac{d}{dx} \left[\frac{(x-x_0)(x-x_1)\cdots(x-x_n)}{(n+1)!} \right] f^{(n+1)}(\xi_x) + \frac{(x-x_0)(x-x_1)\cdots(x-x_n)}{(n+1)!} \frac{d}{dx} [f^{(n+1)}(\xi_x)]. \quad (7.24)$$

Simplification. The error formula (7.24) can be simplified if the point x at which the derivative is to be evaluated happens to be one of the nodes x_i , as in the case of finite difference formulas.

First, if $x = x_i$, then the second term on the right-hand side of (7.24) becomes zero, because $(x - x_i)$ is a factor.

Secondly,

$\frac{d}{dx}(x-x_0)(x-x_1)\cdots(x-x_n)$ at $x = x_i$ is $(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)$.

Thus, in this case (7.24) becomes

$$E'_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\eta) \prod_{\substack{j=0 \\ j \neq i}}^n (x_i - x_j)$$

for some $\eta \in (a, b)$.

Furthermore, if the nodes are equidistant of length h , then

$$\prod_{\substack{j=0 \\ i \neq j}}^n (x_i - x_j) = h^n.$$

Thus, at $x = x_i$, we obtain

$$E_n(x) = \frac{1}{(n+1)!} h^n f^{(n+1)}(\eta).$$

Theorem 7.2. (Error Theorem for Numerical Differentiation).

Let

- (i) $f(x)$ be continuously differentiable on $[a, b]$,
- (ii) x_0, x_1, \dots, x_n be $(n+1)$ distinct points in $[a, b]$,
- (iii) $P_n(x)$ be the interpolating polynomial of degree at most n with x_0, x_1, \dots, x_n as nodes.

(a)

6.1. Then the derivative error at $x = x_i$ is given by

$$E_n(x) = |f'(x_i) - P'_n(x_i)| = \frac{1}{(n+1)!} f^{(n+1)}(\eta) \prod_{\substack{j=0 \\ j \neq i}}^n (x_i - x_j) \quad (7.25)$$

for some $\eta \in (a, b)$.

(b) Furthermore, if the nodes are equidistant of spacing h , then

$$E_n(x) = \frac{h^n}{(n+1)!} f^{(n+1)}(\eta)$$

Special cases: Since finite difference formulas concern finding the derivative at a tabulated point, $x = x_i$, we immediately recover the following error results established earlier:

- Two-point FDF and BDF ($\mathbf{n} = \mathbf{1}$) are $O(h)$. (**First-order approximation**)
- Two-point CDF and three-point FDF and BDF ($\mathbf{n} = \mathbf{2}$) are $O(h^2)$. (**Second-order approximation**)
- Four-point CDF and five-point FDF and BDF ($\mathbf{n} = \mathbf{3}$) are $O(h^3)$. (**Third-order approximation**)

and so on.

Example 7.3

Given the following table of $f(x) = x \ln x$:

x	$f(x)$
1	0
2	1.3863
2.5	2.2907

Approximate $f'(2.1)$ using Lagrange interpolation.

Input Data:

(i) Nodes:	$x_0 = 1, x_1 = 2, x_2 = 2.5$
(ii) Functional values:	$f_0 = 0, f_1 = 1.3863, f_2 = 2.2907$
(iii) The point at which the derivative is to be approximated:	$x = 2.1$
(iv) The degree of interpolation:	$n = 2$

Solution Strategy and Formulas to be used.

- Compute $P_2(x)$ - Lagrange interpolating polynomial of degree 2, using Equation (7.18).
- Compute $P'_2(x)$ using Equation (7.22) and accept it as an approximation to $f'(x)$.

Solution:

Using three-point Lagrangian interpolation, we have

$$f'(x) \approx P'_2(x) = f_0 \left[\frac{2x - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)} \right] + f_1 \left[\frac{2x - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)} \right] + f_2 \left[\frac{2x - x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)} \right]$$

So, $f'(2.1) \approx P'_2(2.1) = 0 + 1.3863 \left[\frac{4.2-1-2.5}{1 \times (-0.5)} \right] + 2.2907 \left[\frac{4.2-1-2}{1.5 \times 0.5} \right] = 1.7243$.

Absolute Error = $|1.7419 - 1.7243| = 0.0176$. (Note that $f'(2.1) = 1.7419$)

7.2.3 Richardson's Extrapolation for Derivative Approximations

Richardson's extrapolation technique is a clever idea to extrapolate information with higher accuracy from two less accurate informations. As we will see later, this technique forms the basis of the popular **Romberg integration method**. In the context of numerical differentiation, the *idea is to find a derivative formula with higher-order truncation error by combining two formulas, each having the same lower-order truncation errors*.

The following example will help understand the technique. Here we will show how the Richardson extrapolation technique can be used to derive four-point CDF which has error $O(h^4)$ by combining two two-point CDFs with spacing h and $\frac{h}{2}$, respectively, each of which has error $O(h^2)$.

Richardson's Technique from Two-point CDF to Four-point CDF

Recall that two-point CDF was derived from three-term Taylor series expansions of $f(x+h)$ and $f(x-h)$. If instead, five-term Taylor series expansions are used, then proceeding exactly in the same way as in Section 7.2.1 and assuming that $f''(x)$ is continuous in $[x-h, x+h]$, we can write

$$f'(x) = \left[\frac{f(x+h) - f(x-h)}{2h} \right] - \frac{f'''(x)}{3!} h^2 + O(h^4). \quad (7.26)$$

The *first term in the brackets on the right-hand side is the two-point CDF which has error $O(h^2)$, with spacing h* . Now, suppose that $f'(x)$ is evaluated with spacing $\frac{h}{2}$. Then we have

$$f'(x) = \left[\frac{f(x+\frac{h}{2}) - f(x-\frac{h}{2})}{h} \right] - \frac{f'''(x)}{4 \cdot 3!} h^2 + O(h^4). \quad (7.27)$$

The *first term in the brackets on the right-hand side is the two-point CDF with spacing $\frac{h}{2}$* , which has also error $O(h^2)$. So the order of truncation error remains the same.

It turns out, however, that these two approximations can be combined to obtain an approximation which has error $O(h^4)$. This can be done by eliminating the term involving h^2 , as follows:

First, multiply Equation (7.27) by 4 to obtain

$$4f'(x) = 4 \frac{f(x + \frac{h}{2}) - f(x - \frac{h}{2})}{h} - \frac{f''(x)}{3!} h^2 + O(h^4).$$

Next, subtract the last equation from Equation (7.26) and divide the result by 3, yielding

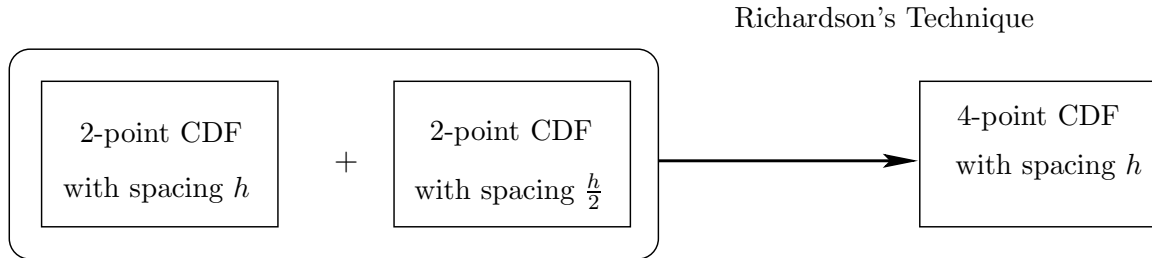
$$f'(x) = \frac{1}{3} \left[4 \frac{f(x + \frac{h}{2}) - f(x - \frac{h}{2})}{h} - \frac{f(x + h) - f(x - h)}{2h} \right] + O(h^4).$$

This is an approximation of $f'(x)$ with error $O(h^4)$. Indeed, the reader will easily recognize the term in the bracket on the right-hand side, as the *four-point CDF with spacing $\frac{h}{2}$* ; that is with the points: $x - h, x - \frac{h}{2}, x, x + \frac{h}{2}, x + h$. Replacing h by $2h$ in the above formula, we have

$$f'(x) \approx \frac{8f(x + h) - 8f(x - h) - f(x + 2h) + f(x - 2h)}{12h} \quad (7.28)$$

which is the *four-point CDF with spacing h* and we already know that this approximation has error $O(h^4)$.

Figure 7.1: Richardson's Technique from 2-point CDF to 4-point CDF



Summarizing: Starting from two two-point CDFs with spacing h and $\frac{h}{2}$, each of which has truncation error $O(h^2)$, Richardson extrapolating technique enables us to obtain a four-point CDF with error $O(h^4)$.

General Case from $O(h^{2k})$ to $O(h^{2k+1})$. We will now consider the general case of deriving a derivative formula of $O(h^{2k+2})$ starting from two formulas of $O(h^{2k})$.

k=1. From $O(h^2)$ to $O(h^4)$. Let $D_0(h)$ and $D_0(\frac{h}{2})$ be two approximate derivative formulas of $O(h^2)$, with spacing h and $\frac{h}{2}$, respectively, which can be written as:

$$f'(x) = D_0(h) + A_1 h^2 + A_2 h^4 + A_3 h^6 + \cdots \quad (7.29)$$

$$f'(x) = D_0\left(\frac{h}{2}\right) + A_1\left(\frac{h}{2}\right)^2 + A_2\left(\frac{h}{2}\right)^4 + A_3\left(\frac{h}{2}\right)^6 + \cdots, \quad (7.30)$$

where A_1, A_2, A_3 , etc. are constants independent of h .

As in the last section, a formula of $O(h^4)$ can now be obtained by eliminating two terms involving h^2 from the above two equations. This is done as follows:

Subtract (7.29) from $4 \times (7.30)$ to obtain:

$$\begin{aligned} 3f'(x) &= 4D_0\left(\frac{h}{2}\right) - D_0(h) - \frac{3}{4}A_2h^4 + \cdots \\ \text{or} \quad f'(x) &= \frac{4}{3}D_0\left(\frac{h}{2}\right) - \frac{1}{3}D_0(h) - \frac{1}{4}A_2h^4 + \cdots \\ \text{or} \quad f'(x) &= D_0\left(\frac{h}{2}\right) + \frac{D_0(\frac{h}{2}) - D_0(h)}{3} - \frac{1}{4}A_2h^4 + \cdots \end{aligned} \quad (7.31)$$

Set

$$D_1(h) = D_0\left(\frac{h}{2}\right) + \frac{D_0(\frac{h}{2}) - D_0(h)}{3}.$$

Then

$$f'(x) = D_1(h) - \frac{1}{4}A_2h^4 + \cdots$$

Thus, $D_1(h)$ is an $O(h^4)$ approximation of $f'(x)$.

k=2. From $O(h^4)$ to $O(h^6)$. To start with, we have $f'(x) = D_1(h) - \frac{1}{4}A_2h^4 + \cdots$.

Replace h by $\frac{h}{2}$ to obtain

$$f'(x) = D_1\left(\frac{h}{2}\right) - \frac{1}{4}A_2\left(\frac{h}{2}\right)^4 + \cdots \quad (7.32)$$

To obtain an $O(h^6)$ approximation, subtract (7.29) from $16 \times (7.30)$ to obtain

$$\begin{aligned} 15f'(x) &= 16D_1\left(\frac{h}{2}\right) - D_1(h) + O(h^6) \\ \text{or} \quad f'(x) &= \frac{16}{15}D_1\left(\frac{h}{2}\right) - \frac{1}{15}D_1(h) + O(h^6) \\ &= D_1\left(\frac{h}{2}\right) + \frac{D_1(\frac{h}{2}) - D_1(h)}{15} + O(h^6) \end{aligned} \quad (7.33)$$

Set

$$D_2(h) = D_1\left(\frac{h}{2}\right) + \frac{D_1(\frac{h}{2}) - D_1(h)}{15}$$

So,

$$f'(x) = D_2(h) + O(h^6).$$

Thus, $D_2(h)$ is an $O(h^6)$ approximation of $f'(x)$.

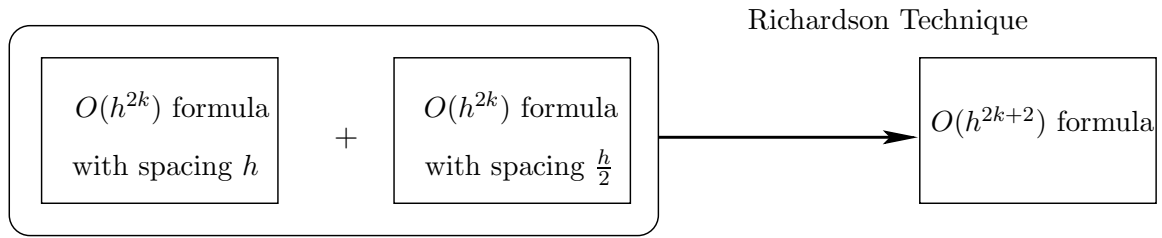
General Case: From $O(h^{2k})$ to $O(h^{2k+2})$.

The pattern is now quite clear.

In the general case,

- Start with two approximations $D_{k-1}(h)$ and $D_{k-1}(\frac{h}{2})$, each of order $O(h^{2k})$.
- Combine $D_{k-1}(h)$ and $D_{k-1}(\frac{h}{2})$ using Richardson's extrapolation technique to obtain an approximation of $O(h^{2k+2})$.

Figure 7.2: Richardson Extrapolation Technique from $O(h^{2k})$ to $O(h^{2k+2})$



Theorem 7.4 (Richardson Extrapolation Theorem for Even Order Approximation).

Let $D_{k-1}(h)$ and $D_{k-1}(\frac{h}{2})$ be two approximations of $O(h^{2k})$ for $f'(x)$ which, respectively, satisfy the error equations:

$$\begin{cases} f'(x) = D_{k-1}(h) + A_1 h^{2k} + A_2 h^{2k+2} + \dots \\ \text{and} \\ f'(x) = D_{k-1}(\frac{h}{2}) + A_1 (\frac{h}{2})^{2k} + A_2 (\frac{h}{2})^{2k+2} + \dots \end{cases}$$

Then

$$D_k(h) = D_{k-1}(\frac{h}{2}) + \frac{D_{k-1}(\frac{h}{2}) - D_{k-1}(h)}{4^k - 1} \quad (7.34)$$

is an $O(h^{2k+2})$ approximation of $f'(x)$.

□

Richardson Extrapolation Table

The above computation can be systematically arranged in the form of the following table, to be called **Richardson Extrapolation Table**.

The arrow \rangle pointing towards an entry of the table shows the dependence of that entry on the two entries of the previous column.

Richardson Extrapolation Table (An Illustration)

$O(h^2)$	$O(h^4)$	$O(h^6)$
$D_0(h)$		
\rangle	$D_1(h)$	
$D_0(\frac{h}{2})$	\rangle	$D_2(h)$
\rangle	$D_1(\frac{h}{2})$	
$D_0(\frac{h}{4})$		

Algorithm 7.5. Richardson Extrapolation Technique for Derivative Approximations

Inputs: (i) h = Spacing of the equally-spaced points

(ii) $D_0(h)$ = An $O(h^2)$ approximation of $f'(x)$ with spacing h

(iii) $D_0(\frac{h}{2})$ = An $O(h^2)$ approximation of $f'(x)$ with spacing $\frac{h}{2}$

Output:

$D_k(h)$ = an $O(h^{2k+2})$ approximation of $f'(x)$, $k = 1, 2, 3, \dots$

For $k = 1, 2, \dots$ do

$$D_k(h) = D_{k-1}\left(\frac{h}{2}\right) + \frac{D_{k-1}\left(\frac{h}{2}\right) - D_{k-1}(h)}{4^k - 1}$$

End

Example 7.6

Given $f(x) = x \ln x$, $h = 0.5$, compute an approximation of $f'(x)$ of $O(h^6)$ starting with two-point CDF with spacing h and $\frac{h}{2}$.

Inputs:

$$\begin{aligned}
 \text{(i)} \quad & \left\{ \begin{aligned} D_0(h) &= \frac{f(x+h)-f(x-h)}{2h} = \frac{f(1.05)-f(0.5)}{1} && \text{(Two-point CDF} \\ &= 1.05 \times \ln(1.05) - 0.5 \times \ln(0.5) && \text{with spacing } h) \\ &= 0.9548 \end{aligned} \right. \\
 \text{(ii)} \quad & \left\{ \begin{aligned} D_0(\frac{h}{2}) &= \frac{f(x+\frac{h}{2})-f(x-\frac{h}{2})}{h} = \frac{f(1.25)-f(0.75)}{0.5} && \text{(Two-point CDF} \\ &= \frac{1.25 \times \ln(1.25) - 0.75 \times \ln(0.75)}{0.5} && \text{with spacing } \frac{h}{2}) \\ &= 0.9894 \end{aligned} \right.
 \end{aligned}$$

k=1: ($O(h^4)$ Approximation)

$$\begin{aligned}
 & D_0(h) \\
 & \quad \rangle \quad D_1(h) \\
 & D_0(\frac{h}{2})
 \end{aligned}$$

Compute $D_1(h)$ by **Richardson's technique** (Set $k = 1$) in Formula (7.34).

$$\begin{aligned}
 D_1(h) &= D_0(\frac{h}{2}) + \frac{D_0(\frac{h}{2}) - D_0(h)}{3} \\
 &= 0.9894 + \frac{0.9894 - 0.9548}{3} \\
 &= 1.0009
 \end{aligned}$$

k=2: ($O(h^6)$ Approximation)

$$\begin{aligned}
 & D_1(h) \\
 & \quad \rangle \quad D_2(h) \\
 & D_1(\frac{h}{2})
 \end{aligned}$$

Compute $D_0(\frac{h}{4})$ by replacing h by $\frac{h}{4}$ in the formula of $D_0(h)$.

$$\begin{aligned}
 D_0(\frac{h}{4}) &= \frac{f(x+\frac{h}{4})-f(x-\frac{h}{4})}{\frac{h}{2}} = \frac{f(1.125)-f(0.8750)}{0.25} \\
 &= \frac{1.25 \times \ln(1.25) - 0.8750 \times \ln(0.8750)}{0.25} \\
 &= 0.9974
 \end{aligned}$$

Compute $D_1(\frac{h}{2})$ by replacing h by $\frac{h}{2}$ in the formula of $D_1(h)$.

$$\begin{aligned}
 D_1(\frac{h}{2}) &= D_0(\frac{h}{4}) + \frac{D_0(\frac{h}{4}) - D_0(\frac{h}{2})}{3} \\
 &= 0.9974 + \frac{0.9974 - 0.9894}{3} \\
 &= 1.0001
 \end{aligned}$$

Compute $D_2(h)$ by Richardson's technique (set $k = 2$) in Formula (7.34).

$$\begin{aligned} D_2(h) &= D_1\left(\frac{h}{2}\right) + \frac{D_1\left(\frac{h}{2}\right) - D_1(h)}{15} \\ &= 1.0000 \end{aligned}$$

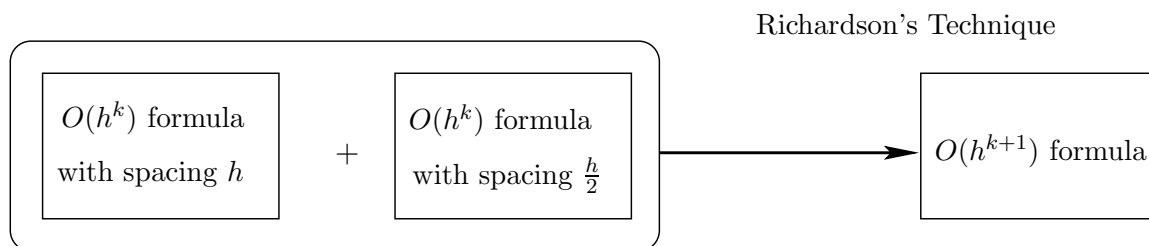
Richardson Extrapolation Table for Example 7.4:

$O(h^2)$	$O(h^4)$	$O(h^6)$
0.9548		
0.9894	1.0009	1.0000
0.9974	1.0001	

Richardson's Extrapolation for $f'(x)$ from $O(h^k)$ to $O(h^{k+1})$

In the preceding section, we described how two approximations of $O(h^{2k})$ can be combined to obtain an approximation of $O(h^{2k+2})$. The underlying assumption there was that *error can be expressed in terms of even powers of h* . The same type of technique can be used to obtain an approximation of $O(h^{k+1})$ starting from two approximations of $O(h^k)$, in a similar way.

Figure 7.3: Richardson Extrapolation from $O(h^k)$ to $O(h^{k+1})$



We state the result in the following theorem. The proof is left as an **Exercise**.

Theorem 7.7. (Richardson Extrapolation Technique from $O(h^k)$ to $O(h^{k+1})$)

Let $D_k(h)$ and $D_k(\frac{h}{2})$ be two $O(h^k)$ approximations of $f'(x)$, which, respectively, satisfy:

$$f'(x) = D_k(h) + A_1 h^k + A_2 h^{k+1} + \dots \quad (7.35)$$

$$f'(x) = D_k\left(\frac{h}{2}\right) + A_1 \left(\frac{h}{2}\right)^k + A_2 \left(\frac{h}{2}\right)^{k+1} + \dots \quad (7.36)$$

Then

$$D_k(h) = D_{k-1}\left(\frac{h}{2}\right) + \frac{D_{k-1}\left(\frac{h}{2}\right) - D_{k-1}(h)}{2^{k-1} - 1}, \quad k = 2, 3, 4, \dots \quad (7.37)$$

is an $O(h^{k+1})$ approximation of $f'(x)$.

□

Richardson Extrapolation Table from $O(h^k)$ to $O(h^{k+1})$

$O(h)$	$O(h^2)$	$O(h^3)$	$O(h^4)$
$D_1(h)$			
\rangle	$D_2(h)$		
$D_1(\frac{h}{2})$	\rangle	$D_3(h)$	
\rangle	$D_2(\frac{h}{2})$	\rangle	$D_4(h)$
$D_1(\frac{h}{4})$	\rangle	$D_3(\frac{h}{2})$	
\rangle	$D_2(\frac{h}{4})$		
$D_1(\frac{h}{8})$			

Example 7.8

Given $f(x) = x \ln x$, $h = 0.5$, (i) Compute an $O(h^2)$ approximation of $f'(1)$ starting from an $O(h)$ approximation. (ii) Compare the result with that obtained by two-point CDF, which is also an $O(h^2)$ approximation.

Solution (i). Step 0. Compute $D_1(h)$ using two-point FDF (which is an $O(h)$ approximation).

$$\begin{aligned} D_1(h) &= \frac{f(x+h) - f(x)}{h} = \frac{f(1.5) - f(1)}{0.5} \\ &= \frac{1.5 \times \ln(1.5) - 1 \times \ln(1)}{0.5} \\ &= 1.2164 \end{aligned}$$

Compute $D_1(\frac{h}{2})$ by replacing h by $\frac{h}{2}$:

$$\begin{aligned} D_1\left(\frac{h}{2}\right) &= \frac{f\left(x + \frac{h}{2}\right) - f(x)}{\frac{h}{2}} = \frac{f(1.25) - f(1)}{0.25} \\ &= \frac{1.25 \times \ln(1.25) - 1 \times \ln(1)}{0.25} \\ &= 1.1157 \end{aligned}$$

Step 1. Compute $D_2(h)$ by Richardson extrapolation (substitute $k = 2$ in Equation (7.37)).

$$\begin{array}{c} D_1(h) \\ \rangle \quad D_2(h) \\ D_1\left(\frac{h}{2}\right) \end{array}$$

$$\begin{aligned}
D_2(h) &= D_1\left(\frac{h}{2}\right) + \frac{D_1\left(\frac{h}{2}\right) - D_1(h)}{2-1} \\
&= 1.1157 + \frac{1.1157 - 1.2164}{1} \\
&= 1.0150
\end{aligned}$$

Absolute Error: $|1.0150 - 1| = 0.0150$. (Note that $f'(1) = 1$).

Note: $D_1(h)$ and $D_1\left(\frac{h}{2}\right)$ are of $O(h)$ and $D_2(h)$ is of $O(h^2)$.

Solution (ii). Comparison with Two-Point CDF:

$$\begin{aligned}
\text{Two-point CDF} &= \frac{f(x+h) - f(x-h)}{2h} = \frac{f(1.5) - f(0.5)}{2 \times 0.5} \\
&= \frac{1.5 \times \ln(1.5) - 0.5 \times \ln(0.5)}{1} \\
&= 0.9548
\end{aligned}$$

Absolute Error: $|0.9548 - 1| = 0.0452$.

Clearly, an $O(h^2)$ Richardson extrapolation technique is more accurate than two-point CDF, which is also $O(h^2)$.

7.2.4 Effects of Round-off Errors and an Optimal Choice of h

So far, we have considered truncation error for approximations of $O(h^k)$, obtained from truncation of a Taylor series or polynomial interpolation.

These error formulas suggest that the smaller h is, the better the approximation. However, this is not entirely true. To come up with an optimal choice of h , we must take into consideration of the round-off error due to the floating point computations as well. We illustrate the idea with two-point CDF.

The two-point central difference formula is given by:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} \quad (7.38)$$

For the sake of notational convenience, write $f(x+h) = f_1$, and $f(x-h) = f_{-1}$.

First, consider the **round-off error**. From the laws of Floating Point Arithmetic (**Theorem 3.7**), we obtain

$$fl\left(\frac{f_1 - f_{-1}}{2h}\right) = \frac{f_1 - f_{-1}}{2h}(1 + \delta), |\delta| \leq 2\mu, \quad (7.39)$$

where μ is the **machine precision**. Thus, *round-off error in computing the CDF* $\leq \frac{\mu}{h}$.

Next, let's consider the **truncation error**. Assume that $|f^{(3)}(x)| \leq M$. Then from (7.17), we obtain the truncation error for two CDF's is $\leq M\frac{h^2}{6}$.

So, the **total error (absolute)** = round-off error + truncation error $\leq \frac{\mu}{h} + M\frac{h^2}{6}$.

Significance of this result is the following:

Though $\frac{h^2}{6}$ becomes smaller as h becomes smaller, the contribution from the round-off error, $\frac{\mu}{h}$, becomes larger as h becomes smaller. Eventually, **when h is too small, the large round-off error will dominate and the computation will become inaccurate.**

This simple illustration reveals the fact that too small of a value of h is hardly an advantage.

What is important is how to choose a value of h that will minimize the total error.

Choosing an optimal h

Since the maximum total absolute error

$$E(h) = \frac{\mu}{h} + \frac{h^2}{6}M$$

is a function of h , an optimal value of h may be obtained by minimizing $E(h)$, assuming of course, one can find the upper bound M for $f'''(x)$. We give a simple example to illustrate this.

Example 7.9

Given $f(x) = e^x$, $0 \leq x \leq 1$. Find the optimal value of h for which the total error in computing two-point CDF approximation to $f'(x)$ will be as small as possible.

Solution. We need to find h for which the maximum absolute total error (which is a function of h):

$$E(h) = \frac{\mu}{h} + M\frac{h^2}{6}$$

will be minimized.

Find M :

$$f(x) = e^x \qquad f^{(3)}(x) = e^x$$

$$|f^{(3)}(x)| \leq |e^x| \leq e \text{ in } 0 \leq x \leq 1$$

$$\text{So,} \qquad M = e$$

Thus, $|E(h)| \leq \frac{\mu}{h} + e\frac{h^2}{6}$.

Minimize $E(h)$.

$$E(h) = \frac{\mu}{h} + e \frac{h^2}{6}$$

is a simple function of h . It is easy to see that $E(h)$ will be minimized if $h = \sqrt[3]{\frac{3\mu}{e}}$. Assume now $\mu = 2 \times 10^{-7}$ (**single-precision**). Then $E(h)$ will be minimized if

$$h = \sqrt[3]{\frac{3 \times 2 \times 10^{-7}}{e}} \approx 0.0060$$

Verification. The readers are invited to verify [**Exercise**] the above result by computing errors with different values of h in the neighborhood of 0.0060.

7.2.5 Approximations of Higher-order Derivatives and Partial Derivatives

Needs for computing higher-order and partial derivatives arises in a wide variety of scientific and engineering applications. Mathematical models of many of these applications are either second- or higher-order differential equations and/or partial differential equations. Typically, such equations are solved in two stages:

Stage I The differential equations are approximated by means of finite differences or finite element techniques that lead to a system of algebraic equations.

Stage II Solution to the system of algebraic equations gives the solution of the applied problem.

We will consider finite difference approximation of second-order derivatives and first- and second-order partial derivatives. *These formulas can be derived exactly in the same way as their counterparts for the first-order derivatives.* We will illustrate the derivation of **three-point difference formulas for $f''(x)$** .

Three-Point Difference Formulas for Second Derivatives and their Truncation Errors

Suppose that the functional values of x , $x+h$, and $x+2h$ are known. Using three-term Taylor's series expansion of $f(x+h)$ with error terms, we can write

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f''(\xi_1) \quad \text{where } x < \xi_1 < x+h. \quad (7.40)$$

Similarly,

$$f(x+2h) = f(x) + 2hf'(x) + \frac{4h^2}{2!}f''(x) + \frac{8h^3}{3!}f''(\xi_2) \quad \text{where } x < \xi_1 < x+2h. \quad (7.41)$$

Eliminate now $f'(x)$ from these two equations. To do this, multiply Equation (7.40) by 2 and subtract it from Equation (7.41),

$$f(x+2h) - 2f(x+h) = -f(x) + h^2 f''(x) + h^3(f''(\xi_2) - f''(\xi_1)) \quad (7.42)$$

Assuming that $f'''(x)$ is continuous on $[x, x+2h]$, by the **Intermediate Value Theorem**, there exists a number ξ between ξ_1 and ξ_2 such that

$$f''(x) = \frac{f'''(\xi_1) + f'''(\xi_2)}{2}. \quad (7.43)$$

Thus, we have

$$f(x+2h) - 2f(x+h) = -f(x) + h^2 f''(x) + \frac{h^3}{2} f''(\xi) \quad (7.44)$$

Solving for $f''(x)$, we have the three-point FDF for $f''(x)$.

In the same way, we can derive three-point BDF and CDF, and other higher-order formulas for $f''(x)$. We state some of these formulas below. Their derivations are left as **Exercises**.

Three-point FDF for $f''(x)$ with Error Term:

$$f''(x) = \frac{f(x) - 2f(x+h) + f(x+2h)}{h^2} + \frac{h}{2} f''(\xi) \quad (7.45)$$

Three-point BDF for $f''(x)$ with Error Term:

$$f''(x) = \frac{f(x-2h) - 2f(x-h) + f(x)}{h^2} + O(h) \quad (7.46)$$

Three-point CDF for $f''(x)$ with Error Term:

$$f''(x) = \frac{1}{h^2} [f(x-h) - 2f(x) + f(x+h)] - \frac{h^2}{12} f^4(\xi), \quad \text{where } x-h < \xi < x+h. \quad (7.47)$$

Five-point CDF for $f''(x)$ with Error Term:

$$f''(x) = \frac{-f(x-2h) + 16f(x-h) - 30f(x) + 16f(x+h) - f(x+2h)}{12h^2} + O(h^4) \quad (7.48)$$

Example 7.10

For the function $f(x) = x \ln x$, approximate $f''(x)$ at $x = 1$, using the five-point forward difference formula, with $h = 0.1$.

Solution:

Input Data: $x - 2h = 0.8000, x - h = 0.9000, x = 1, x + h = 1.01, x + 2h = 1.2$.

Formula to be used: Equation (7.50).

$$f''(x) \approx \frac{-f(0.8000) + 16f(0.9000) - 30f(1) + 16f(1.1) - f(1.2)}{12 \times (0.1)^2} = 1.0000$$

Exact Value of $f''(x)$ at $x = 1$ is 1.

Derivative Formulas for Second-Order Derivatives from Polynomial Interpolation

As in the case of the first-order derivative, formulas for the second and higher-order derivatives can also be derived based on Lagrange or Newton interpolations, and finite-difference formulas can be recovered as special cases. Of course, *the primary advantage with the interpolating formulas is that they can be used to approximate the derivatives also at non-tabulated points*. As noted before, these formulas are, however, computation intensive. As an illustration, we will derive here: the four-point formula for $f''(x)$ based on Lagrange interpolation.

The four points are: x_0, x_1, x_2, x_3 . The Lagrange polynomial $P_3(x)$ of degree 3 is given by

$$\begin{aligned} P_3(x) &= L_0(x)f_0 + L_1(x)f_2 + L_2(x)f_3 + L_3(x)f_3 \\ &= \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)}f_0 + \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}f_1 \\ &\quad + \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}f_2 + \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}f_3 \end{aligned}$$

Differentiating two times, we get the **four-point formula for $f''(x)$** :

$$\begin{aligned} f''(x) \approx P_3''(x) &= \frac{2(x-x_1)+(x-x_2)+(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)}f_0 + \frac{2(x-x_0)+(x-x_2)+(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}f_1 \\ &\quad + \frac{2(x-x_0)+(x-x_1)+(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}f_2 + \frac{2(x-x_0)+(x-x_1)+(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}f_3 \end{aligned}$$

Note: The above formula can be used to approximate $f''(x)$ at *any point* x in $[x_0, x_3]$, not necessarily only at tabulated points.

Error in Four-point Formula

Recall that the error term for four-point interpolation is

$$E_3(x) = \frac{(x-x_0)(x-x_1)(x-x_2)(x-x_3)}{4!}f^{(4)}(\xi_x), \quad \text{where } x_0 < \xi_x < x_3.$$

It is easy to see that $E_3''(x)$ is $O(h^2)$, where h is the distance of equally spaced nodes.

The four-point FDF for $f''(x)$ can be obtained by setting: $x_0 = x$, $x_1 = x + h$, $x_2 = x + 2h$, and $x_3 = x + 3h$.

Four-point FDF of $f''(x)$:

$$f''(x) \approx \frac{2f(x) - 5f(x+h) + 4f(x+2h) - f(x+3h)}{h^2} + O(h^2)$$

The other formulas for $f''(x)$ and higher-order derivatives can similarly be computed (though not easily).

Finite-Difference Approximations of the Partial Derivatives

Let $u(x, t)$ be a function of two variables x and t . Then the partial derivatives of the first-order with respect to x and t are defined by

$$\begin{aligned}\frac{\partial u}{\partial x}(x, t) &= \lim_{h \rightarrow 0} \frac{u(x+h, t) - u(x, t)}{h} \\ \frac{\partial u}{\partial t}(x, t) &= \lim_{k \rightarrow 0} \frac{u(x, t+k) - u(x, t)}{k}\end{aligned}$$

Idea: Apply the corresponding finite difference formulas of the single variable to one of the two variables for which the approximation is sought, while keeping the other variable constant.

- Two-point FDF for $\frac{\partial u}{\partial x}(x, t)$ and $\frac{\partial u}{\partial t}(x, t)$ are given by:

$$\begin{aligned}\frac{\partial u}{\partial x}(x, t) &= \frac{u(x+t, t) - u(x, t)}{h} + O(h) \\ \frac{\partial u}{\partial t}(x, t) &= \frac{u(x, t+k) - u(x, t)}{k} + O(k)\end{aligned}$$

- Two-point BDF for $\frac{\partial u}{\partial x}$ and $\frac{\partial u}{\partial t}$ are given by:

$$\begin{aligned}\frac{\partial u}{\partial x}(x, t) &= \frac{u(x, t) - u(x-h, t)}{h} + O(h) \\ \frac{\partial u}{\partial t}(x, t) &= \frac{u(x, t) - u(x, t-k)}{k} + O(k)\end{aligned}$$

- Two-point CDF for $\frac{\partial u}{\partial x}$ and $\frac{\partial u}{\partial t}$ are given by:

$$\begin{aligned}\frac{\partial u}{\partial x}(x, t) &= \frac{u(x+h, t) - u(x-h, t)}{2h} + O(h) \\ \frac{\partial u}{\partial t}(x, t) &= \frac{u(x, t+k) - u(x, t-k)}{2k} + O(k)\end{aligned}$$

- Three-point CDF for second-order partial derivatives are given by:

$$\frac{\partial^2 u}{\partial x^2}(x, t) = \frac{u(x-h, t) - 2u(x, t) + u(x+h, t)}{h^2} + O(h^2)$$

$$\frac{\partial^2 u}{\partial t^2}(x, t) = \frac{u(x, t-k) - 2u(x, t) + u(x, t+k)}{k^2} + O(k^2)$$

List of Important Finite-Difference Formulas

A. For First-order Derivative

$$\begin{aligned}
 O(h) & \left\{ \begin{array}{l} \textbf{Two-Point FDF:} \\ f'(x) = \frac{f(x+h) - f(x)}{h} \\ \textbf{Two-point BDF:} \\ f'(x) = \frac{f(x) - f(x-h)}{h} \end{array} \right. \\
 O(h^2) & \left\{ \begin{array}{l} \textbf{Two-point CDF:} \\ f'(x) = \frac{f(x+h) - f(x-h)}{2h} \\ \textbf{Three-point FDF:} \\ f'(x) = \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} \\ \textbf{Three-point BDF:} \\ f'(x) = \frac{f(x-2h) - 4f(x-h) + 3f(x)}{2h} \end{array} \right. \\
 O(h^4) & \left\{ \begin{array}{l} \textbf{Four-point CDF:} \\ f'(x) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) + f(x+2h)}{12h} \\ \textbf{Five-point FDF:} \\ f'(x) = \frac{-25f(x) + 48f(x+h) - 36f(x+2h) + 16f(x+3h) - 3f(x+4h)}{12h} \end{array} \right.
 \end{aligned}$$

B. For Second-Order Derivatives

$$O(h) \left\{ \begin{array}{l} \textbf{Three-point FDF:} \\ f''(x) = \frac{f(x) - 2f(x+h) + f(x+2h)}{h^2} \\ \textbf{Three-point BDF:} \\ f''(x) = \frac{f(x-2h) - 2f(x-h) + f(x)}{h^2} \end{array} \right.$$

$$\begin{aligned}
O(h^2) & \left\{ \begin{array}{l} \textbf{Four-point FDF:} \\ f''(x) = \frac{2f(x)-5f(x+h)+4f(x+2h)-f(x+3h)}{h^2} \\ \\ \textbf{Four-point BDF:} \\ f''(x) = \frac{-f(x-3h)+4f(x-2h)-5f(x-h)+2f(x)}{h^2} \\ \\ \textbf{Three-point CDF:} \\ f''(x) = \frac{f(x-h)-2f(x)+f(x+h)}{h^2} \end{array} \right. \\
O(h^4) & \left\{ \begin{array}{l} \textbf{Five-point CDF:} \\ f''(x) = \frac{-f(x-2h)+16f(x-h)-30f(x)+16f(x+h)-f(x+2h)}{12h^2} \end{array} \right.
\end{aligned}$$

C. For Partial Derivatives

$u(x, t)$ is a function of two variables x and t .

$$\textbf{Two-point FDF:} \left\{ \begin{array}{l} \frac{\partial u}{\partial x}(x, t) = \frac{u(x+h, t)-u(x, t)}{h} + O(h) \\ \frac{\partial u}{\partial t}(x, t) = \frac{u(x, t+k)-u(x, t)}{k} + O(k) \end{array} \right.$$

$$\textbf{Two-point BDF:} \left\{ \begin{array}{l} \frac{\partial u}{\partial x}(x, t) = \frac{u(x, t)-u(x-h, t)}{h} + O(h) \\ \frac{\partial u}{\partial x}(x, t) = \frac{u(x+h, t)-u(x-h, t)}{2h} + O(h) \end{array} \right.$$

$$\textbf{Two-point CDF:} \left\{ \begin{array}{l} \frac{\partial u}{\partial x}(x, t) = \frac{u(x+h, t)-u(x-h, t)}{2h} + O(h) \\ \frac{\partial u}{\partial t}(x, t) = \frac{u(x, t+k)-u(x, t-k)}{2h} + O(k) \end{array} \right.$$

$$\textbf{Three-point CDF:} \frac{\partial^2 u}{\partial t^2}(x, t) = \frac{u(x, t-k)-2u(x, t)+u(x, t+k)}{k^2} + O(k^2)$$

Example 7.11

Given $f(x) = x \ln x$, $h = 1$.

(a) Approximate $f''(1)$ using three-point FDF.

(b) Approximate $f''(3)$ using three-point BDF.

Analytical Formula: $f'(x) = 1 + \ln x$

$$f''(x) = \frac{1}{x}$$

Solutions:

(a) **Using three-point FDF at $x = 1$:**

Input Data: $x = 1, x + h = 2, x + 2h = 3$.

$$\begin{aligned} f''(x) &\approx \frac{f(x) - 2f(x+h) + f(x+2h)}{h^2} \\ f''(1) &\approx \frac{f(1) - 2f(2) + f(3)}{1} = 0 - 2 \times 1.3863 + 3.2958 \\ &= 0.5232. \end{aligned}$$

Absolute Error: $|f''(1) - 0.5232| = |1 - 0.5232| = 0.4768 = 47.68\%$.

(b) **Using three-point BDF at $x = 3$:**

Input Data: $x = 3, x - 2h = 1, x - h = 2$.

$$\begin{aligned} f''(x) &\approx \frac{f(x-2h) - 2f(x-h) + f(x)}{h^2} \\ f''(3) &\approx \frac{f(1) - 2f(2) + f(3)}{1} = -2 \times 1.3863 + 3.2958 \\ &= 0.5232. \end{aligned}$$

Absolute Error: $|f''(3) - 0.5232| = |0.3333 - 0.5232| = 0.1899$.

Remarks: Clearly, the above approximations are not good. The readers are invited to compute **three-point** and **five-point CDF** approximations of $f''(x)$ using **Formulas (7.47) and (7.48)** and verify the improvement in accuracy with these formulas.

Exercises on Part I

7.1. **(Computational)** Given the following table of functional values:

x	$f(x) = \sin x$	$f(x) = \cos x$	$f(x) = xe^x$
0	0	1	0
0.5	0.4794	0.8776	0.8249
1	0.8415	0.5403	2.7183
1.5	0.9975	0.0707	6.7225
2	0.9093	-0.4161	14.7781
2.5	0.5985	-0.811	30.4562
3	0.1411	-0.9900	60.2566

For each function, approximate the following derivative values and compare them with actual values.

- (a) $f'(1.8)$ using an interpolating polynomial approximation,
- (b) $f'(0.5)$ using three-point FDF,
- (c) $f'(1.5)$ using three-point CDF,
- (d) $f'(2.5)$ using three-point BDF.

7.2. **(Computational)** Given the following tables of functional values for the functions (as indicated):

(a)	x	$f(x) = \sqrt{x} \sin x$
	0	0
	$\frac{\pi}{4}$	0.6267
	$\frac{\pi}{2}$	1.2533
	$\frac{3\pi}{4}$	1.0856
	π	0
(b)	x	$f(x) = \sin(\sin(\sin(x)))$
	0	0
	$\frac{\pi}{4}$	0.6049
	$\frac{\pi}{2}$	0.7456
	$\frac{3\pi}{4}$	0.6049
	π	0

	x	$f(x) = \sqrt{x + \sqrt{x + \sqrt{x}}}$	
	0	0	
(c)	0.5	1.2644	
	1	1.5538	
	1.5	1.7750	
	2	1.9616	
	x	$f(x) = x \ln x + x^2$	$f'(x)$
	0.1	-0.2168	
(d)	0.5	0.7738	
	0.8	6.8763	
	1	20.0855	

- A. For the functions in Tables (a) and (b), do the following:
- Estimate $f'(0)$ using two-point and three-point forward difference formulas and compare your results.
 - Estimate $f'(\frac{\pi}{2})$ using two-point forward difference and backward difference formulas; and two-point and four-point central difference formulas. Compare the results among both the difference formulas and themselves and with actual values of $f'(x)$ obtained by analytical differentiation.
 - Estimate $f'(\frac{2\pi}{3})$ using Lagrangian interpolation and compare the result with the actual value of $f'(\frac{2\pi}{3})$ obtained by analytical differentiation.
- B. For functions in Table (c), repeat Part A with $x = 0$ in (i), $x = 1$ in (ii), and $x = 1.2$ in (iii).
- C. Fill in the missing entries, as accurately as possible, in Table (d) using the appropriate formulas.

7.3. **(Analytical)** Derive the following formulas with their associated truncation errors:

- Three-point forward-difference formula: $f'(x) \approx \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h}$.
- Four-point central difference formula: $f'(x) \approx \frac{f(x-2h) - 8f(x-h) + 8f(x+h)}{12h}$.
- Three-point forward difference formula for $f''(x)$: $f''(x) \approx \frac{f(x) - 2f(x+h) + f(x+2h)}{h^2}$.
- Three-point central difference formula for $f''(x)$: $f''(x) \approx \frac{f(x-h) - 2f(x) + f(x+h)}{h^2}$.

7.4. **(Applied)** The amount of force F needed to move an object along a horizontal plane is given by

$$F(\theta) = \frac{\mu W}{\mu \sin \theta + \cos \theta}$$

where

W = weight of the object

μ = frictional *constant*

θ = angle between the attached string to the object makes with the plane.

The following table gives F versus θ :

θ (radians)	F (lb)
0.5	25.7458
1	28.7031
1.5	44.8274
2	231.7826

Given $\mu = 0.6$, $W = 50\text{lb}$. Find, using a FDF:

- (a) at what rate the force is changing when $\theta = 1.5$.
- (b) at what rate the force is changing when $\theta = 1.8$.
- (c) at what angle the rate of change is zero.

7.5. **(Applied)** The following table gives the estimated world population (in millions) at various dates:

Year	Population
1960	2,982
1970	3,692
1980	4,435
1990	5,263
2000	6,070
2010	6,092

Estimate the rate of the world's population growth in 1980, 2010, and 1985; using the appropriate derivative formulas (as accurately as possible).

7.6. **(Applied)** Heat Conduction through Material

The famous **Fourier law** of heat conduction states that the unit time rate of heat transfer through a material is *proportional to the negative gradient in the temperature*. In its simpler form, it can be expressed as:

$$Q_x = -k \frac{dT}{dx}$$

where

x = distance (m) along the path of heat flow

T = temperature ($^{\text{deg}}\text{C}$)

k = thermal conductivity

Q_x = heat flux (W/m^2)

Given the following table:

x	0	0.1	0.2	0.3
T	15	10	5	3

Compute k if Q_x at $x = 0$ is 40 W/m^2 .

7.7. (Applied) US Trade Deficit

The derivatives market, subprime mortgage crisis, and a declining dollar value, contributed to an economic crisis in 2008 in U.S.A. On December 1, 2008, a recession was officially declared. The recession, however, led to a record trade deficit. The trade deficit occurs when the total good and services of a country's imports is greater than the total exports.

The following table shows the trade deficits (in billions) for the years 2006-2010.

Year	Trade Deficit (Approximate)
2006	753
2007	696
2008	698
2009	381
2010	500

Using the above table, predict the rate of change for the trade deficits (as accurately as possible) for the years 2007 and 2010.

7.8. (Computational) Given $f(x) = x + e^x$, $h = 0.5$.

- Starting with two-point CDF, compute an $O(h^6)$ approximation of $f'(x)$ using the Richardson extrapolation technique.
- Ppresent your results in the form of a Richardson extrapolation table.

7.9. (Computational) Given $f(x) = xe^x$, $h = 0.1$.

- Starting from two-point FDF, compute an $O(h^3)$ approximation of $f'(0.5)$ using the Richardson extrapolation technique.
- Ppresent your results in the form of a Richardson extrapolation table.

7.10. (Computational) Verify the claim made in Exercise 7.6 for the optimal value of $h = 0.0060$, by computing the errors with several different values of h in the neighborhood of 0.0060.

7.11. (Analytical) Using Newton's interpolation, give a proof of the Error Theorem for Numerical Differentiation (**Theorem 7.2**).

7.12. (Analytical) Give a proof of Theorem 7.7.

- 7.13. For the functions in Exercise 7.1 and 7.2 (a), Find an optimal value of h for which the error in computing a two-point CDF approximation to $f'(x)$ will be as small as possible.

MORE TO COME

3/1/13 EC

PART II. Numerical Integration

What's Ahead

- A Case Study of Numerical Integration: Blood Flow and Cardiac Input
- Basic Quadrature Rules: Trapezoidal, Simpson, Simpson's $\frac{3}{8}$ th, Corrected Trapezoidal
- Composite Rules (via Monomial and Lagrange Interpolation)
- Romberg Integration
- Gaussian Quadrature
- Improper Integral
- Multiple Integrals
- MATLAB Functions and Their Uses

7.3 Statement and some Applications of the Numerical Integration Problem

In a beginning calculus course, the students learn various analytical techniques for finding the integral

$$\int_a^b f(x) dx$$

and a variety of applications in science and engineering. A few of these applications include

- Finding the **area between two curves** $y = f_1(x)$ and $y = f_2(x)$ and the lines $x = a$ and $x = b$:

$$\begin{aligned} \text{Area} &= \int_a^b f(x) dx, \text{ where} \\ f(x) &= f_1(x) - f_2(x), \quad f_1(x) \geq f_2(x), \end{aligned}$$

and both functions $f_1(x)$ and $f_2(x)$ are continuous on $[a, b]$.

- **Volume of a solid** obtained by rotating a curve or a region bounded by two curves about a line
- The **arc length of a function** $f(x)$ given by $a(x) = \int_a^b \sqrt{1 + [f'(x)]^2} dx$
- The **area of a surface of revolution**: the area of the surface obtained by rotating the curve $y = f(x)$, $a \leq x \leq b$ about the x -axis is $S = \int_a^b 2\pi f(x) \sqrt{1 + [f'(x)]^2} dx$
- The **center of mass or the centroid of a region** at (\bar{x}, \bar{y}) :

$$\begin{aligned} \bar{x} &= \frac{1}{A} \int_a^b x f(x) dx \\ \bar{y} &= \frac{1}{A} \int_a^b \frac{1}{2} [f(x)]^2 dx \end{aligned}$$

where A is the area.

- Statistical applications, such as computing the **mean of any probability density function** $f(x)$:

$$\mu = \int_{-s}^s x f(x) dx$$

Of particular interest is the probability density function of the **normal distribution**:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

where σ = **standard deviation**

- Biological applications, such as the **volume of the blood flow** in the heart and the **cardiac output** when dye is injected.

For some detailed discussions of some of these applications and many others, the readers may refer to the authoritative calculus book by Stewart [1].

Why Numerical Integration?

As in the case of numerical differentiation, the need to numerically evaluate an integral comes from the fact that

- In many practical applications, the integrand is not explicitly known—all that's known are the certain discrete values of the integrand from experimental measurements.

or

- The integral is difficult to compute analytically.

We will learn various techniques of numerical integration in this Chapter.

We conclude this section with a biological application of integrals.

A Biological Application: Blood Flow and Cardiac Input

Blood Flow: Recall in Section 7.1.1, we considered the **velocity of blood flow** in a tube using the law of laminar flow:

$$v(r) = \frac{1}{4\eta} \frac{\Delta P}{l} (R^2 - r^2).$$

Here we consider the *volume of blood flow*. It can be shown (see Stewart [1]) that the volume V , of the blood that passes a cross section per unit time is given by

$$V = \int_0^R 2\pi r \frac{\Delta P}{4\eta l} (R^2 - x^2) dx$$

Thus, V is a function of r . If the integrand is explicitly known, then it is quite easy to compute this integral. However, in many practical applications, only certain discrete values of r will be known in $[0, R]$. The integration thus needs to be computed numerically. We will later consider this application with numerical data.

Cardiac output: The **cardiac output** of the heart is the volume of blood pumped by the heart per unit time. Usually, a dye is injected into the right atrium to measure the cardiac output.

Let $c(t)$ denote the concentration of the dye at time t and let the dye be injected for the time interval $[0, T]$. Then the cardiac output C_O is given by:

$$C_O = \frac{A}{\int_0^T c(t) dt}$$

where A is the amount of dye.

Again, in practical applications, $c(t)$ will be measured at certain equally spaced times over the interval $[0, T]$. Thus, all that will be known to a user is some discrete values of $c(t)$ at these instants of time, from where the integral must be computed numerically.

A solution of this problem with numerical data will be considered later in this Chapter.

Numerical Integration Problem

Given

- (i) the functional values f_0, f_1, \dots, f_n , of a function $f(x)$, at x_0, x_1, \dots, x_n , where $a = x_0 < x_1 < x_2 \dots < x_{n-1} < x_n = b$,

or

- (ii) an integratable function $f(x)$ itself over $[a, b]$.

Compute: an approximate value of $I = \int_a^b f(x)dx$ using these functional values or those computed from the given function.

7.4 Numerical Integration Techniques: General Idea

The numerical techniques discussed in this chapter have the following form:

$$\int_a^b f(x)dx \approx I = w_0f(x_0) + w_1f(x_1) + \dots + w_nf(x_n) = w_0f_0 + w_1f_1 + \dots + w_nf_n$$

where x_0, x_1, \dots, x_n are called the **nodes**, and w_0, w_1, \dots, w_n are called the **weights**. Thus, we can have two types of formulas:

Type 1. The $(n + 1)$ nodes are given and the $(n + 1)$ weights are determined by the rule. The well-known classical quadrature rules, such as the **Trapezoidal rule** and **Simpson's rule** are examples of this type.

Type 2. Both nodes and weights are determined by the rule. **Gaussian quadrature** is an example of this type.

We will discuss Type 1 first.

General Idea. The polynomial functions are easier to integrate. Thus, to evaluate $\int_a^b f(x)$ numerically, the obvious things to do are:

- Find the interpolating polynomial $P_n(x)$ of degree at most n , passing through the $(n + 1)$ points: $(x_0, f_0), (x_1, f_1), \dots, (x_n, f_n)$.
- Evaluate $I = \int_a^b P_n(x)dx$ and accept this value as an approximation to I .

Exactness of the Type 1 Quadrature Rules: These rules, by construction, should be exact for all polynomials of degree less than or equal to n .

We know that there are different ways to construct the unique interpolating polynomial using different basis functions.

We will describe here the quadrature rules based on **monomial** and **Lagrange interpolations**.

7.5 Numerical Integration Techniques Based on Monomial Interpolation

A numerical quadrature formula based on monomial basis interpolation, by definition, is exact for the basis polynomials $\{1, x, x^2, \dots, x^n\}$. This observation leads to a system of $(n + 1)$ equations in $(n + 1)$ unknowns, w_0, w_1, \dots, w_n , as shown below.

$$\begin{aligned}
f(x) = 1 : \quad & \int_a^b 1 dx = w_0 + w_1 + \cdots + w_n \\
\implies \quad & b - a = w_0 + w_1 + \cdots + w_n \\
\\
f(x) = x : \quad & \int_a^b x dx = w_0 x_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n \\
\implies \quad & \frac{b^2 - a^2}{2} = w_0 x_0 + w_1 x_1 + \cdots + w_n x_n \\
\\
& \vdots \\
f(x) = x^n : \quad & \int_a^b x^n dx = w_0 x_0^n + w_1 x_1^n + \cdots + w_n x_n^n \\
\implies \quad & \frac{b^{n+1} - a^{n+1}}{n+1} = w_0 x_0^n + w_1 x_1^n + \cdots + w_n x_n^n
\end{aligned}$$

The above $(n + 1)$ equations are assembled as:

$$\begin{cases}
w_0 + w_1 + \cdots + w_n & = b - a \\
w_0 x_0 + w_1 x_1 + \cdots + w_n x_n & = \frac{b^2 - a^2}{2} \\
\vdots & \\
w_0 x_0^n + w_1 x_1^n + \cdots + w_n x_n^n & = \frac{b^{n+1} - a^{n+1}}{n+1}
\end{cases}$$

In matrix-vector notations:

$$\begin{pmatrix}
1 & 1 & \cdots & 1 \\
x_0 & x_1 & \cdots & x_n \\
\vdots & & \ddots & \vdots \\
x_0^{n-1} & x_1^{n-1} & \cdots & x_n^{n-1} \\
x_0^n & x_1^n & \cdots & x_n^n
\end{pmatrix}
\begin{pmatrix}
w_0 \\
w_1 \\
\vdots \\
w_{n-1} \\
w_n
\end{pmatrix}
=
\begin{pmatrix}
b - a \\
\frac{b^2 - a^2}{2} \\
\vdots \\
\frac{b^n - a^n}{n} \\
\frac{b^{n+1} - a^{n+1}}{n+1}
\end{pmatrix}$$

The readers will recognize that the matrix on the left-hand side is a **Vandermonde matrix**, which is **nonsingular** by virtue of the fact that x_0, x_1, \dots, x_n are **distinct**. The **unique solution** of this system will yield the unknowns w_0, w_1, \dots, w_n .

Uniqueness. Thus, we have the following **uniqueness result**:

Given $x_0 < x_1 < x_2 \dots < x_n$, there exist an unique set of weights w_0, w_1, \dots, w_n such that

$$\int_a^b f(x) dx \approx w_0 f_0 + w_1 f_1 + \cdots + w_n f_n$$

where $f_i = f(x_i)$, $i = 0, 1, \dots, n$.

The above process is sometimes called the method of **undetermined coefficients**.

Special cases. The following two special cases give rise to two famous quadrature rules.

$n = 1 \rightarrow$ **Trapezoidal rule** ($x_0 = a, x_1 = b$)

$$\int_a^b f(x)dx \approx \frac{b-a}{2}[f(a) + f(b)]$$

$n = 2 \rightarrow$ **Simpson's rule** ($x_0 = a, x_1 = \frac{a+b}{2}, x_2 = b$)

$$\int_a^b f(x)dx \approx \frac{b-a}{6}[f(a) + 4f(\frac{a+b}{2}) + f(b)]$$

7.6 Numerical Integration Rules Based on Lagrange Interpolation

We will now derive Trapezoidal, Simpson's, and other rules based on Lagrange interpolation. The use of Lagrange interpolation also will help us derive the error formulas for these rules.

The **Idea** is as follows:

- Given n , find the Lagrange interpolating polynomial $P_n(x)$ of degree at most n , approximating $f(x)$:

$$P_n(x) = f_0L_0(x) + f_1L_1(x) + \cdots + f_nL_n(x)$$

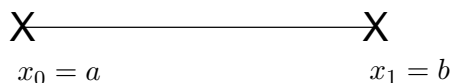
where $L_0(x), L_1(x), \dots, L_n(x)$ are Lagrange polynomials each of degree n .

- Compute $\int_a^b P_n(x)dx$ and accept the result as an approximation to $\int_a^b f(x)dx$.

7.6.1 Trapezoidal Rule

$f(x) \approx \text{Lagrange Interpolating polynomial of degree 1} \implies \text{Trapezoidal Rule}$
--

In this case, there are only two nodes: x_0, x_1 .



The Lagrange interpolating polynomial $P_1(x)$ of degree 1 is:

$$P_1(x) = L_0(x)f_0 + L_1(x)f_1.$$

$$\text{So, } I = \int_{a=x_0}^{b=x_1} f(x)dx \approx \int_{x_0}^{x_1} [L_0(x)f_0 + L_1(x)f_1]dx.$$

$$\text{Recall that } L_0(x) = \frac{x - x_1}{x_0 - x_1}, \text{ and } L_1(x) = \frac{x - x_0}{x_1 - x_0}.$$

Thus, I_T = trapezoidal rule approximation of I is given by

$$\begin{aligned} I_T &= \int_{x_0}^{x_1} \left[\frac{x - x_1}{x_0 - x_1} f_0 + \frac{x - x_0}{x_1 - x_0} f_1 \right] dx \\ &= \frac{f_0}{x_0 - x_1} \int_{x_0}^{x_1} (x - x_1) dx + \frac{f_1}{x_1 - x_0} \int_{x_0}^{x_1} (x - x_0) dx \\ &= \frac{f_0}{x_0 - x_1} \left[\frac{(x - x_1)^2}{2} \right]_{x_0}^{x_1} + \frac{f_1}{x_1 - x_0} \left[\frac{(x - x_0)^2}{2} \right]_{x_0}^{x_1} = \frac{(x_1 - x_0)}{2} (f_0 + f_1). \end{aligned}$$

Let $x_1 - x_0 = h$. Since $f_0 = f(x_0) = f(a)$, and $f_1 = f(x_1) = f(b)$, we obtain

Trapezoidal Rule

$$I_T = \frac{(x_1 - x_0)}{2} (f_0 + f_1) = \frac{h}{2} (f_0 + f_1) = \frac{b - a}{2} (f(a) + f(b)) \quad (7.49)$$

Error in Trapezoidal Rule

Since the above formula only gives a crude approximation to the actual value of the integral, we need to assess the error.

To obtain an error formula for this integral approximation, recall that *the error formula for interpolation with a polynomial of degree at most n* is given by

$$E_n(x) = \frac{f^{(n+1)}(\xi(x))\Psi_n(x)}{(n+1)!}, \quad (7.50)$$

where $\Psi_n(x) = (x - x_0)(x - x_1) \dots (x - x_n)$, and $a \leq \xi \leq b$ ($a \equiv x_0, b \equiv x_n$).

Since in case of the Trapezoidal rule, $n = 1$, the error associated interpolation error is:

$$E_1(x) = \frac{f''(\xi(x))\Psi_1(x)}{2!},$$

where $\Psi_1(x) = (x - x_0)(x - x_1)$, and $f''(x)$ is the second derivative of $f(x)$.

Integrating this formula we have the following **error formula for the Trapezoidal rule** (denoted by $E_T(x)$):

$$E_T(x) = \int_{x_0}^{x_1} \frac{f''(\xi(x))}{2!} (x - x_0)(x - x_1) dx = \int_{x_0}^{x_1} \frac{f''(\xi(x))}{2!} \Psi_1(x) dx. \quad (7.51)$$

We now show how the above formula can be simplified. The **Weighted Mean Value Theorem** (WMT) from calculus will be needed for this purpose.

Weighted Mean Value Theorem for Integrals (WMT)

Let

- (i) $f(x)$ be continuous on $[a, b]$
- (ii) $g(x)$ does not change sign on $[a, b]$

Then there exists a number c in (a, b) such that

$$\int_a^b f(x)g(x)dx = f(c) \int_a^b g(x)dx$$

To apply the WMT to (7.51), we note that

- (i) $f''(x)$ is continuous on $[x_0, x_1]$ (*Hypothesis (i) of WMT is satisfied*).
- (ii) $\Psi_1(x) = (x - x_0)(x - x_1)$ *does not change sign* over $[x_0, x_1]$. This is because for any x in $[x_0, x_1]$, $(x - x_0) > 0$ and $(x - x_1) < 0$ (*Hypothesis (ii) of WMT is satisfied*).

So, by applying the WMT to $E_T(x)$, with $g(x) = \Psi_1(x)$ and noting that $h = x_1 - x_0$, we obtain

Error in the Trapezoidal Rule

$$\begin{aligned}
 E_T &= \frac{f''(\eta)}{2!} \int_{x_0}^{x_1} (x - x_0)(x - x_1) dx = \frac{-h^3}{12} f''(\eta) \\
 &= \frac{-(b-a)^3}{12} f''(\eta) = \frac{-(b-a)}{12} h^2 f''(\eta)
 \end{aligned} \tag{7.52}$$

where $a < \eta < b$.

Trapezoidal Rule with Error Formula

$$\int_{x_0=a}^{x_1=b} f(x) dx = \underbrace{\frac{b-a}{2} [f(a) + f(b)]}_{\text{Trapezoidal Rule}} - \underbrace{\frac{b-a}{12} h^2 f''(\eta)}_{\text{Error}}, \quad a < \eta < b.$$

Exactness of Trapezoidal Rule: From the above error formula it follows that the Trapezoidal rule is exact only for polynomials of degree 1 or less. This is because, for all these polynomials, $f''(x) = 0$, and is *non zero*, whenever $f(x)$ is of degree 2 and higher.

Geometrical Representation of the Trapezoidal Rule

Trapezoidal rule approximates the area under the curve $y = f(x)$ from $x_0 = a$ to $x_1 = b$ by the area of the trapezoid as shown below:

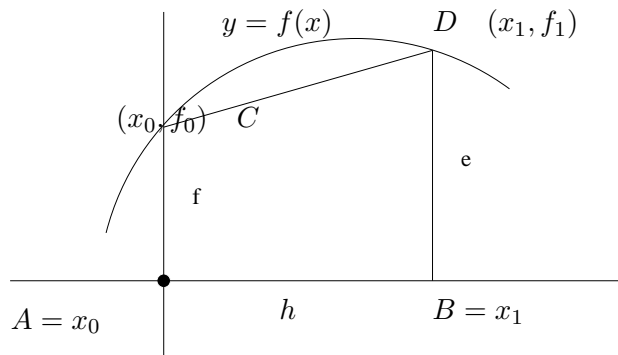


Figure 7.1: Illustration of the Trapezoidal Rule.

Note: The area of the trapezoid $ABCD = \text{Length of the base} \times \text{average height} = h \cdot \frac{1}{2}(f_0 + f_1) = \frac{h}{2}(f_0 + f_1) = \frac{h}{2}[f(a) + f(b)]$.

7.6.2 Simpson's Rule

If $f(x)$ is approximated by Lagrange interpolating polynomial of degree 2 and the integration is taken over $[a, b]$ with the interpolating polynomial as the integrand, the result is *Simpson's rule*.

$$\boxed{f(x) \approx \text{Lagrange Interpolating polynomial of degree 2} \implies \text{Simpson's Rule}}$$

The three points of interpolation in this case are: x_0, x_1 , and x_2 .

$$\begin{array}{ccc} \times & \bullet & \times \\ x_0 = a & x_1 & x_2 = b \end{array}$$

The Lagrange interpolating polynomial $P_2(x) = L_0(x)f_0 + L_1(x)f_1 + L_2(x)f_2$.

So,

$$I = \int_{a=x_0}^{b=x_2} f(x)dx \approx \int_{x_0}^{x_2} [L_0(x)f_0 + L_1(x)f_1 + L_2(x)f_2]dx \quad (7.53)$$

$$\text{Now, } L_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}, \quad L_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)},$$

$$\text{and } L_2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}.$$

Let h be the distance between two consecutive points of interpolation, assumed to be equally spaced. That is, $x_1 - x_0 = h$ and $x_2 - x_1 = h$.

Substituting these expressions of $L_0(x), L_1(x)$ and $L_2(x)$ into (7.53) and integrating, we obtain [Exercise] the famous **Simpson's Rule**:

$$I_S = \frac{h}{3}(f_0 + 4f_1 + f_2) \quad (7.54)$$

Noting that

$$\begin{cases} h = \frac{b-a}{2} \\ f_0 = f(x_0) = f(a) \\ f_1 = f(x_1) = f(x_0 + h) = f(a + \frac{b-a}{2}) = f(\frac{a+b}{2}) \\ f_2 = f(x_2) = f(b) \end{cases}$$

We can rewrite (7.54) as:

$$\int_a^b f(x)dx \approx I_S = \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right].$$

Error in Simpson's Rule

Since $n = 2$, the error formula (7.50) becomes

$$E_2(x) = \frac{1}{3!} f^3(\xi(x)) \Psi_2(x) dx,$$

where $\Psi_2(x) = (x - x_0)(x - x_1)(x - x_2)$

Since $\Psi_2(x)$ **does change sign** in $[x_0, x_2]$, we *can not apply WMT directly* to obtain the error formula for Simpson's rule. In this case, we use the following modified formula:

Modified Integration Error Formula

Let

- (i) $\Psi_n(x) = (x - x_0)(x - x_1) \cdots (x - x_n)$ be such that it changes sign on (a, b) , but

$$\int_a^b \Psi_n(x) dx = 0$$

- (ii) x_{n+1} be a point such that $\Psi_{n+1}(x) = (x - x_{n+1})\Psi_n(x)$ is of one sign in $[a, b]$.
 (iii) $f(x)$ is $(n + 2)$ times continuously differentiable,

then application of the MVT for integration yields:

$$E_{n+1}(x) = \frac{1}{(n + 2)!} f^{(n+2)}(\eta) \int_a^b \Psi_{n+1}(x) dx \quad (7.55)$$

where $a < \eta < b$.

To apply the above modified error formula to obtain an error expression for Simpson's rule, we note the following:

- (i) $\int_{x_0}^{x_2} \Psi_2(x) dx = \int_{x_0}^{x_2} (x - x_0)(x - x_1)(x - x_2) dx = 0$ (**Hypothesis (i) is satisfied**).

- (ii) If a point x_3 is chosen as $x_3 \equiv x_1$, then

$$\Psi_3(x) = (x - x_3)\Psi_2(x) = (x - x_1)(x - x_0)(x - x_1)(x - x_2) = (x - x_1)^2(x - x_0)(x - x_2).$$

is of the **same sign** in $[x_0, x_3]$ (*Hypothesis (ii) is satisfied*).

Assume further that $\mathbf{f}(\mathbf{x})$ is **4 times continuously differentiable** (*Hypothesis (iii) is satisfied*). Then by (7.55) we have the following modified error formula for Simpson's rule:

$$\begin{aligned} E_S &= \frac{1}{4!} f^{(4)}(\eta) \int_{x_0}^{x_2} \Psi_3(x) \\ &= \frac{1}{24} f^{(4)}(\eta) \int_{x_0}^{x_2} (x - x_1)^2 (x - x_0)(x - x_2) dx \\ &= \frac{1}{24} f^{(4)}(\eta) \left(\frac{-4}{15} \right) h^5 = -\frac{h^5}{90} f^{(4)}(\eta) \quad a < \eta < b. \end{aligned}$$

Substituting $h = \frac{b-a}{2}$, we have

Error in Simpson's Rule:
$$E_S = -\frac{\left(\frac{b-a}{2}\right)^5}{90} f^{(4)}(\eta), \quad \text{where } a < \eta < b.$$

Simpson's Rule with Error Formula

$$\begin{aligned} \int_a^b f(x) dx = \int_{a=x_0}^{b=x_2} f(x) dx &= \underbrace{\frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]}_{\text{Simpson's Rule}} \\ &\quad + \underbrace{\frac{\left(\frac{b-a}{2}\right)^5}{90} f^{(4)}(\eta)}_{\text{Error formula}}, \end{aligned}$$

where $a < \eta < b$.

Exactness of Simpson's Formula

Since $f^4(x)$ is zero for all polynomials of degree less than or equal to 3, but is nonzero for all polynomials of higher degree, we conclude from the above error formula:

Simpson's rule is exact for all polynomials of degree less than or equal to 3.

Remarks: Because of the use of the modified error formula (7.55), the error for Simpson's rule is of one order higher than that warranted by the usual error formula for interpolation. That is why, Simpson's rule is exact for all polynomials of degree less than or equal to 3, even when Simpson's formula is obtained by approximating $f(x)$ by a polynomial of degree 2.

Precision of a Quadrature Rule

A quadrature rule is said to have the degree of precision k if the error term of that rule is zero for all polynomials of degree less than or equal to k , but it is different from zero for some polynomial of degree $k + 1$. Thus,

- Trapezoidal rule has a degree of precision 1.
- Simpson's rule has a degree of precision 3.

7.6.3 Simpson's Three-Eighth Rule

Simpson's rule was developed by approximating $f(x)$ with a polynomial of degree 2. If $f(x)$ is approximated using a polynomial of degree 3, then we have **Simpson's Three-Eighth Rule** [Exercise]:

Let

- (i) x_0, x_1, x_2 , and x_3 be the points of subdivisions of the interval $[a, b]$.
- (ii) $h = x_{i+1} - x_i, i = 0, 1, 2$.

Then

Simpson's $\frac{3}{8}$ th rule is:

$$\int_{x_0}^{x_3} f(x)dx \approx \frac{3h}{8} [f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)].$$

The Error term in this case is of the same order as the Simpson's rule.

Specifically, **the error in the Simpson's $\frac{3}{8}$ th rule**, denoted by $E_S^{\frac{3}{8}}$ is given by:

$$E_S^{\frac{3}{8}} = -\frac{3h^5}{80}f^{(4)}(\eta), \quad \text{where } x_0 < \eta < x_3.$$

Remarks: It is clear that applications of Simpson's rule and Simpson's Three-Eighth rule are restricted to the even and odd number of subintervals, respectively. Thus, often these two rules are used in conjunction with each other.

7.6.4 Corrected Trapezoidal Rule

Simpson's rule and Simpson's $\frac{3}{8}$ th rule were developed by approximating $f(x)$ by polynomials of degree 2 and 3, respectively. Yet, another rule can be developed by approximating $f(x)$

by **Hermite interpolating** polynomial of degree 3, with a special choice of the nodes as: $x_0 = x_1 = a$ and $x_2 = x_3 = b$.

This rule, for obvious reasons, is called the **corrected trapezoidal rule** (ICT). This rule along with the error expression is stated below. Their proofs are left as an **[Exercise]**.

Corrected Trapezoidal Rule with Error Formula

$$\int_a^b f(x)dx \approx I_{TC} = \underbrace{\frac{b-a}{2}[f(a) + f(b)] + \frac{(b-a)^2}{12}[f'(a) - f'(b)]}_{\text{Corrected Trapezoidal Rule}} + \underbrace{\frac{(b-a)^5}{720}f^{(4)}(\eta)}_{\text{Error Formula}}$$

Remarks: Comparing the error formulas of the trapezoidal rule and the corrected trapezoidal rule, it is obvious that the corrected trapezoidal rule is much more accurate than the trapezoidal rule.

However, the price to pay for this gain is that I_{CT} requires computations of $f'(a)$ and $f'(b)$.

Example 7.12 (a) Approximate $\int_0^1 \cos x dx$ using

- (i) Trapezoidal rule
 - (ii) Simpson's rule
 - (iii) Simpson's $\frac{3}{8}$ th rule
 - (iv) Corrected trapezoidal rule
- (b) In each case, compute the maximum error and compare this maximum error with the actual error, obtained by an analytical formula.

Solution (a).

(i) **Trapezoidal Rule Approximation**

$$\begin{cases} \text{Input Data: } x_0 = a = 0, & x_1 = b = 1, & f(x) = \cos x \\ \text{Formula to be used: } I_T = \frac{b-a}{2}[f(a) + f(b)] \end{cases}$$

$$I_T = \frac{1}{2}[\cos(0) + \cos(1)] = \frac{1}{2}(1 + 0.5403) = 0.7702.$$

(ii) **Simpson's Rule Approximation**

$$\begin{cases} \text{Input Data: } x_0 = 0, x_1 = 0.5, x_2 = 1, & f(x) = \cos x, \quad a = 0, b = 1 \\ \text{Formula to be used: } I_S = \frac{b-a}{6}[f(a) + 4f(\frac{a+b}{2}) + f(b)] \end{cases}$$

$$I_S = \frac{1}{6}[\cos(0) + 4\cos(\frac{1}{2}) + \cos(1)] = \frac{1}{6}[1 + 4 \times 0.8776 + 0.5403] = 0.8418.$$

(iii) **Simpson's $\frac{3}{8}$ Rule Approximation**

$$\begin{cases} \text{Input Data: } x_0 = 0, x_1 = \frac{1}{3}, x_2 = \frac{2}{3}, x_3 = 1, & h = \frac{1}{3} \\ \text{Formula to be used: } I_S^{\frac{3}{8}} = \frac{3h}{8}[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] \end{cases}$$

$$I_S^{\frac{3}{8}} = \frac{1}{8}[\cos(0) + 3\cos(\frac{1}{3}) + 3\cos(\frac{2}{3}) + \cos(1)] = \frac{1}{8}[1 + 3 \times 0.9450 + 3 \times 0.7859 + 0.5403] = 0.8416.$$

(iv) **Corrected Trapezoidal Rule Approximation**

$$\begin{cases} \text{Input Data: } a = 0, b = 1, f(x) = \cos x \\ \text{Formula to be used: } I_{TC} = \frac{b-a}{2}[f(a) + f(b)] + \frac{(b-a)^2}{12}[f'(a) - f'(b)] \end{cases}$$

$$I_{TC} = \frac{1}{12}[\cos(0) + \cos(1)] + \frac{1}{12}[-\sin(0) + \sin(1)] = 0.8403.$$

Solution (b).

Maximum Errors vs. Actual Errors

To compute maximum absolute errors and compare them with the actual error, we need

(i) $f''(x) = -\cos x; \max_{0 \leq x \leq 1} |f''(x)| = 1$ (for trapezoidal rule)

(ii) $f^{(iv)}(x) = \cos x; \max_{0 \leq x \leq 1} |f^{(iv)}(x)| = 1$ (for Simpson's rule)

(iii) $I = \int_0^1 \cos x dx = [\sin(1) - \sin(0)] = 0.8415$ (analytical value of I)

• **For Trapezoidal Method**

$$\begin{cases} \text{Error formula: } E_T = \frac{b-a}{12}h^2 f''(\eta) \\ \text{Maximum absolute error: } \frac{1}{12} \times \max_{0 \leq x \leq 1} |f''(x)| = \frac{1}{12} = 0.0833 \\ \text{Actual absolute error: } |I - I_T| = |0.8415 - 0.7702| = 0.0713 \end{cases}$$

- **For Simpson's Method**

$$\begin{cases} \text{Error formula : } E_S = -\frac{(\frac{b-a}{2})^5}{90} f^{(iv)}(\eta) \\ \text{Maximum absolute error : } \frac{1}{32} \times \frac{1}{90} \max_{0 \leq x \leq 1} |f^{(4)}(x)| = \frac{1}{32} \times \frac{1}{90} = 3.477 \times 10^{-4} \\ \text{Actual absolute error : } |I - I_S| = |0.8415 - 0.8418| = 3 \times 10^{-4} \end{cases}$$

- **For Corrected Trapezoidal Rule**

$$\begin{cases} \text{Error formula : } \frac{(b-a)^5}{720} f^{(4)}(\eta) \\ \text{Maximum absolute error : } \frac{1}{720} \max_{0 \leq x \leq 1} |f^{(4)}(x)| = \frac{1}{720} = 0.0014 \\ \text{Actual absolute error : } |I - I_{TC}| = |0.8415 - 0.8403| = 0.0012 \end{cases}$$

- **For Simpson's $\frac{3}{8}$ th Rule [Exercise]**

Observations: (i) Actual absolute error in each case is comparable with the corresponding maximum (**worst possible**) error, but is always less than the latter.

(ii) The corrected trapezoidal rule is more accurate than the trapezoidal rule.

(iii) Simpson's rule is most accurate.

7.7 Newton-Cotes Quadrature

Trapezoidal, Simpson's and Simpson's Three-Eighth rules, developed in the last section are special cases of a more general rule, known as, the **Closed Newton-Cotes** (CNC) rule.

An n -point **closed Newton-Cotes** rule over $[a, b]$ has the $(n+1)$ nodes: $x_i = a + i \frac{(b-a)}{n}$, $i = 0, 1, \dots, n$.

Thus, it is easy to see that for

- $n = 1 \Rightarrow$ Trapezoidal Rule (Two nodes: $x_0 = a, x_1 = b$)
- $n = 2 \Rightarrow$ Simpson's Rule (Three nodes: $x_0 = a, x_1 = \frac{a+b}{2}, x_2 = b$)
- $n = 3 \Rightarrow$ Simpson's $\frac{3}{8}$ th Rule.
- $n = 4 \Rightarrow$ Boole's Rule [Exercise]

$$\int_{x_0}^{x_4} f(x) dx \approx \frac{2h}{45} (7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4).$$

(Note that CNC Rule includes the end points of the nodes.)

The **open Newton-Cotes** has the $(n + 1)$ nodes **which do not include the end points**. These nodes are given by: $x_i = a + i \frac{(b-a)}{n+2}, i = 1, 2, \dots, n$.

Mid-Point Rule: A well-known example of the n -point open Newton-Cotes rule is the **mid-point rule** (with $n = 0$). Thus, **the midpoint rule is based on interpolation of $f(x)$ with a constant function**. The only node in this case is: $x_1 = \frac{a+b}{2}$.

So,

$$I_M = \text{Midpoint Approximation to the Integral } \int_a^b f(x)dx = (b-a)f\left(\frac{a+b}{2}\right).$$

Error Formula for the Midpoint Rule:

In this case $\Psi_0(x) = x - x_1 = x - \frac{a+b}{2}$ *changes sign* in (a, b) .

However, note that if we let $x_0 = x_1$, then

$$\Psi_1(x) = (x - x_1)^2 = \left(x - \frac{a+b}{2}\right)^2$$

is always of the same sign. Thus, as in the case of Simpson's rule, we can derive the error formula for I_M [**Exercise**]:

$$E_M = f''(\eta) \frac{(b-a)^3}{24}, \quad \text{where } a < \eta < b.$$

Midpoint Rule with Error Formula

$$\int_a^b f(x)dx \approx I_M = \underbrace{(b-a)f\left(\frac{a+b}{2}\right)}_{\text{Midpoint Rule}} + \underbrace{f''(\eta) \frac{(b-a)^3}{24}}_{\text{Error Formula}}, \quad \text{where } a < \eta < b.$$

Remark: Comparing the error terms of I_M and I_T , we easily see that the midpoint rule is more accurate than the trapezoidal rule. **The following simple example compares the accuracy of these different rules:** trapezoidal, Simpson's, midpoint, and corrected trapezoidal.

Example 7.13

Apply the midpoint, trapezoidal, corrected trapezoidal, and Simpson's rule to approximate

$$\int_0^1 e^x dx$$

$$I_M = f(0.5) = e^{0.5} = 1.6487$$

$$I_T = \frac{1}{2}(1 + e) = 1.8591$$

$$I_S = \frac{1}{6} \left(e^0 + 4e^{\frac{1}{2}} + e^1 \right) = 1.7189$$

$$I = \int_0^1 e^x dx \approx 1.7183 \text{ (correct to four decimal digits).}$$

$$I_{TC} = \frac{1}{2}[1 + e] + \frac{1}{12}[1 - e] = 1.7160$$

Error Comparisons:

$$E_M = |I - I_m| = 0.0696 \text{ (midpoint error)}$$

$$E_T = |I - I_T| = 0.1408 \text{ (trapezoidal error)}$$

$$E_S = |I - I_S| = 6 \times 10^{-4} \text{ (Simpson's error)}$$

$$E_{TC} = |I - I_{TC}| = 0.0023 \text{ (corrected trapezoidal error)}$$

Observations:

- (i) As predicted by theory, the corrected trapezoidal rule is more accurate than the trapezoidal rule.
- (ii) Simpson's rule is most accurate.
- (iii) The midpoint rule is also more accurate than the trapezoidal rule.

(Other Higher-order Open Newton-Cotes Rules) Higher-order open Newton-Cotes rules can be derived with values of $n = 1, 2, 3$, etc. These are left as Exercises.

7.8 The Composite Rules

To obtain a greater accuracy, *the idea then will be to subdivide the interval $[a, b]$ into smaller intervals, apply these quadrature formulas in each of these smaller intervals and add up the results to obtain a more accurate approximation.* The resulting quadrature rule is called the **composite rule**. Thus, a procedure for constructing a composite rule will be as follows:

- Divide $[a, b]$ into n equal subintervals. Let the points of the subdivisions be:

$$a = x_0 < x_1 < x_2 \dots < x_{n-1} < x_n = b.$$

Assume that each of these intervals are of equal length. Let $h = \frac{b-a}{n}$ = the length of each of these subintervals.

Then $x_0 = a$, $x_1 = a + h$, $x_2 = a + 2h$, \dots , $x_n = b = a + nh$.

- Apply a quadrature rule in each of these subintervals.
- Add the results.

The Composite Trapezoidal Rule

$$\int_a^b f(x)dx = \int_{x_0=a}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \dots + \int_{x_{n-1}}^{x_n=b} f(x)dx$$

Applying the basic trapezoidal rule (7.49) to each of the integrals on the right-hand side and adding the results, we obtain the composite trapezoidal rule, I_{CT} .

$$\begin{aligned} \text{Thus, } \int_a^b f(x)dx &\approx I_{CT} = \frac{h}{2}(f_0 + f_1) + \frac{h}{2}(f_1 + f_2) + \dots + \frac{h}{2}(f_{n-1} + f_n) \\ &= h \left(\frac{f_0}{2} + f_1 + f_2 + \dots + f_{n-1} + \frac{f_n}{2} \right). \end{aligned}$$

Noting that

$$\left\{ \begin{array}{l} f_0 = f(x_0) = f(a) \\ f_1 = f(x_1) = f(a + h) \\ \vdots \\ f_{n-1} = f(x_{n-1}) = f(a + (n-1)h) \\ f_n = f(x_n) = f(b) \end{array} \right.$$

We can write I_{CT} as

$$I_{CT} = \frac{h}{2} [f(a) + 2(f(a + h) + 2f(a + 2h) + \dots + 2f(a + (n-1)h) + f(b))]$$

Using the summation notation, we have the following formula for the **composite trapezoidal rule**:

$$I_{CT} = \frac{h}{2} \left[f(a) + 2 \sum_{i=1}^{h-1} f(a + ih) + f(b) \right]$$

Note: $n = 1 \Rightarrow I_{CT}$ = the basic trapezoidal rule (7.65).

7.8.1 The Error Formula for Composite Trapezoidal Rule

The error formula for I_{CT} is obtained by adding the individual error terms of the trapezoidal rule in each of the subintervals. Thus, the error formula for the composite trapezoidal rule, denoted by E_{CT} , is given by:

$$E_{CT} = \frac{-h^3}{12} [f''(\eta_1) + f''(\eta_2) + \dots + f''(\eta_n)]$$

where $x_{i-1} < \eta_i < x_i$, $i = 2, 3, \dots, n$.

Simplification of the Error Formula

.

To simplify the expression within the brackets, we assume that $f''(x)$ is **continuous on** $[a, b]$. Then by **Intermediate Value Theorem** (IVT), we can write

$$f''(\eta_1) + \dots + f''(\eta_n) = n f''(\eta), \text{ where } \eta_1 < \eta < \eta_n.$$

Thus, the above formula for E_{CT} reduces to:

$$\begin{aligned} E_{CT} &= -\frac{nh^3}{12} f''(\eta), \text{ where } \eta_1 < \eta < \eta_n. \\ &= -n \frac{(b-a)}{n} \cdot \frac{h^2}{12} f''(\eta) = -\frac{(b-a)}{12} h^2 f''(\eta), \\ &\quad \left(\text{because } h = \frac{b-a}{n} \right). \end{aligned}$$

The Composite Trapezoidal Rule With Error Term

$$\int_{a=x_0}^{b=x_n} f(x)dx = \underbrace{\frac{h}{2} \left[f(a) + 2 \sum_{i=1}^{n-1} f(a+ih) + f(b) \right]}_{\text{Composite Trapezoidal Rule}} - \underbrace{\frac{b-a}{12} h^2 f''(\eta)}_{\text{Error}}$$

where $a < \eta < b$.

7.8.2 The Composite Simpson's Rule (I_{CS})

Since Simpson's rule was obtained with two subintervals, in order to derive the CSR, we divide the interval $[a, b]$ into *even number of subintervals*, say $n = 2m$, where m is a positive integer and then apply Simpson's rule in each of those subintervals and finally, add up the results.

Obtaining Composite Simpson's Rule

- Divide the interval $[a, b]$ into n **even** number of equal subintervals: $[x_0, x_2], [x_2, x_4] \dots, [x_{n-2}, x_n]$. Set $h = \frac{b-a}{n}$.

Then we have

$$\int_a^b f(x)dx = \int_{x_0}^{x_2} f(x)dx + \int_{x_2}^{x_4} f(x)dx + \dots + \int_{x_{n-2}}^{x_{2n}} f(x)dx.$$

- Apply the basic Simpson's rule (**Formula 7.54**) to each of the integrals on the right-hand side and add the results to obtain the **Composite Simpson's rule**, I_{CS} .

$$\begin{aligned} I_{CS} &= \frac{h}{3} [(f_0 + 4f_1 + f_2) + (f_2 + 4f_3 + f_4) + \dots + (f_{n-2} + 4f_{n-1} + f_n)] \\ &= \frac{h}{3} [(f(a) + f(b) + 4(f_1 + f_3 + \dots + f_{n-1}) + 2(f_2 + f_4 + \dots + f_{n-2}))]. \end{aligned} \tag{7.56}$$

(Note that $f_0 = f(x_0) = f(a)$ and $f_n = f(x_n) = f(b)$).

Note: $n = 2 \Rightarrow$ The **basic Simpson's rule** (7.54).

The Error in Composite Simpson's Rule

The *error in composite Simpson's Rule*, denoted by E_C^S , is given by

$$E_C^S = \frac{-h^5}{90} \left[f^{(iv)}(\eta_1) + f^{(iv)}(\eta_2) + \dots + f^{(iv)}(\eta_{\frac{n}{2}}) \right],$$

where $x_{2i-2} < \eta_i < x_{2i}$, $i = 1, \dots, \frac{n}{2}$.

A Simplified Expression for the Error

As before, we can now invoke **IVT** to simplify the above error expression. *Assuming that $f^{(iv)}(x)$ is continuous on $[a, b]$* , by IVT, we can write

$$f^{(iv)}(\eta_1) + f^{(iv)}(\eta_2) + \dots + f^{(iv)}(\eta_n) = n f^{(iv)}(\eta_{\frac{n}{2}}),$$

where $\eta_1 < \eta < \eta_{\frac{n}{2}}$.

Thus, the error formula for the composite Simpson's rule is simplified to:

$$\begin{aligned} E_{CS} &= \frac{-h^5}{90} \times \frac{n}{2} f^{(iv)}(\eta) \\ &= \frac{-h^5}{180} \frac{(b-a)}{h} f^{(iv)}(\eta) = \frac{-h^4}{180} (b-a) f^{(iv)}(\eta). \\ &\quad \left(\text{Since } n = \frac{b-a}{h} \right). \end{aligned}$$

The Composite Simpson's Rule with Error Term

$$\begin{aligned} \int_a^b f(x) dx &= \frac{h}{3} [(f_0 + f_n) + 4(f_1 + \dots + f_{n-1}) + 2(f_2 + \dots + f_{n-2})] - \frac{h^4}{180} (b-a) f^{(4)}(\eta) \\ &= \underbrace{\frac{h}{3} [f(a) + f(b) + 4(f_1 + \dots + f_{n-1}) + 2(f_2 + \dots + f_{n-2})]}_{\text{Composite Simpson's Rule}} - \underbrace{\frac{h^4}{180} (b-a) f^{(4)}(\eta)}_{\text{Error}} \end{aligned}$$

Example 7.14

$$\text{Let } f(x) = \begin{cases} x & \text{if } 0 \leq x \leq \frac{1}{2} \\ 1-x & \text{if } \frac{1}{2} \leq x \leq 1 \end{cases}$$

(a) **Trapezoidal Rule over $[0, 1]$ with $h = 1$**

$$I_T = \frac{1}{2}(f(0) + f(1)) = \frac{1}{2}(0 + 0) = \boxed{0}$$

(b) **Composite Trapezoidal Rule with $h = \frac{1}{2}$**

$$\begin{aligned} I_{CT} &= \frac{1}{4} (f(0) + f(\frac{1}{2})) + \frac{1}{4} (f(\frac{1}{2}) + f(1)) \\ &= \frac{1}{4} (0 + \frac{1}{2} + \frac{1}{2} + 0) = \frac{1}{4}(1) = \boxed{\frac{1}{4}} \end{aligned}$$

Solution (c). Simpson's Rule over $[0, 1]$

$$\begin{aligned} I_S &= \frac{1}{6} [f(0) + 4f(\frac{1}{2}) + f(1)] \\ &= \frac{1}{6} (0 + 4(\frac{1}{2}) + 0) = \frac{1}{6} \times 2 = \frac{1}{3} \end{aligned}$$

Example 7.15 Applications of the Error Formula

Determine h to approximate

$$I = \int_{0.1}^{10} \frac{1}{t e^t} dt$$

with an accuracy of $\epsilon = 10^{-3}$ using the **composite trapezoidal rule**.

Solution.

Input Data: $f(t) = \frac{1}{te^t}$; $a = 0.1, b = 10$.

Formula to be used: $E_{CT} = -\frac{b-a}{12} h^2 f''(\eta)$.

Step 1. Find the **maximum value** (in magnitude) of E_{CT} in the interval $[0.1, 10]$.

We then need to find $f''(\eta)$.

Now, $f(t) = \frac{1}{te^t}$ (given)

So, $f''(t) = \frac{1}{te^t} \left(\frac{2}{t^2} + \frac{2}{t} + 1 \right)$.

Thus, $f''(t)$ is maximum in $[0.1, 10]$ when $t = 0.1$.

and

$$\max |f''(t)| = \frac{1}{0.1 \times e^{0.1}} (200 + 20 + 1) = 1999.69$$

So, the *absolute maximum value* of $E_{CT} = \left(\frac{9.9}{12} \right) h^2 \times 1999.69$.

Step 2. Find h .

To approximate I with an accuracy of $\epsilon = 10^{-3}$, h has to be such that the absolute maximum error is less than or equal to ϵ . That is,

$$\left(\frac{9.9}{12}\right) h^2 \times 1999.69 \leq 10^{-3}.$$

or,

$$h^2 \leq 12 \cdot 10^{-3} \times \frac{1}{9.9 \times 1999.69}$$

or

$$h \leq 7.7856 \times 10^{-4}.$$

The readers are invited to repeat the calculations with a smaller interval, $[0, 1]$, and compare the results [**Exercise**].

The Composite Corrected Trapezoidal Rule (I_{CTC})

I_{CTC} and the associated error formula can be developed in the same way as the composite trapezoidal and Simpson's rule. We leave the derivations as [**Exercises**].

The Composite Corrected Trapezoidal Rule with Error Term

$$\int_a^b f(x)dx = \underbrace{h(f_1 + f_2 + \cdots + f_{n-1}) + \frac{h}{2}(f(a) + f(b))}_{\text{Composite Corrected Trapezoidal Rule}} + \underbrace{\frac{h^2}{12}[f'(a) - f'(b)] + \frac{h^4(b-a)}{720}f^{(iv)}(\eta)}_{\text{Error Term}}$$

7.9 Romberg Integration

As said before Romberg integration is based on Richardson's extrapolation.

Idea for Romberg Integration

Starting with two lower-order approximations with the same order of truncation errors, obtained by a certain quadrature rule with step-size h and $\frac{h}{2}$, respectively, generate successively higher-order approximations.

We will take **Trapezoidal rule** (with two subintervals) which is $O(h^2)$, as our basis quadrature rule and show how Richardson's extrapolation technique can be applied to obtain a sequence of approximations of $O(h^4), O(h^6), \dots$

An implementation of the above idea is possible, because one can prove that *the error for the trapezoidal rule satisfies an equation involving only even powers of h* . Thus, proceeding exactly in the same way, as in the case of numerical differentiation, the following result, analogous to the Richardson's theorem for numerical differentiation, can be established [**Exercise**].

Theorem 7.16. Richardson's Theorem for Numerical Integration

Define $h_k = \frac{b-a}{2^{k-1}}$. Let $R_{k,j-1}$ and $R_{k-1,j-1}$ ($k = 2, 3, \dots, h; j = 2, 3, \dots, k$) be two trapezoidal approximations of $I = \int_a^b f(x)dx$, respectively, of step sizes h_k and h_{k-1} that satisfy the equations:

$$\begin{aligned} I &= R_{k,j-1} + A_1 h_k^2 + A_2 h_k^4 + \dots \\ I &= R_{k-1,j-1} + A_1 h_{k-1}^2 + A_2 h_{k-1}^4 + \dots \end{aligned}$$

Then, $R_{k-1,j-1}$ and $R_{k,j-1}$ can be combined to have an improved approximation R_{kj} of $O(h_k^{2j})$:

$$R_{k,j} = R_{k,j-1} + \frac{R_{k,j-1} - R_{k-1,j-1}}{4^{j-1} - 1} \quad (7.57)$$

□

Romberg Table. The numbers $\{R_{kj}\}$ are called **Romberg numbers** and can be arranged in the form of the following table.

Romberg Table

$O(h^2)$	$O(h^4)$	$O(h^6)$	\dots	$O(h^{2n})$
R_{11}				
\rangle	R_{22}			
R_{21}	\rangle	R_{33}		
\rangle	R_{32}			
R_{31}	\rangle	R_{43}		
\rangle	R_{42}	\vdots	\ddots	
R_{41}	\dots	\vdots	\ddots	\rangle
\vdots	\vdots	\vdots		R_{nn}
\rangle	R_{n2}	R_{n3}		
R_{n1}				

A Recursive Relation for the 1st Column of the Romberg Table

The entries $R_{11}, R_{21}, \dots, R_{n1}$ of the 1st column of the Romberg table are just trapezoidal approximations with spacing h_1, h_2, \dots, h_n , respectively. These numbers can be computed by using the composite trapezoidal rule. However, once R_{11} is computed, the other entries can be computed recursively, as shown below, without repeatedly applying the composite trapezoidal rule with increasing intervals.

R_{11} = Trapezoidal rule approximation with 1 interval:

$$R_{11} = \frac{h_1}{2}[f(a) + f(b)] = \frac{b-a}{2}[f(a) + f(b)].$$

R_{21} = Trapezoidal rule with 2 intervals:

$$\begin{aligned} R_{21} &= \frac{h_2}{2}[f(a) + f(b) + 2f(a + h_2)] \\ &= \frac{h_2}{2} \left[f(a) + f(b) + 2f\left(a + \frac{b-a}{2}\right) \right] \\ &= \frac{b-a}{4} \left[f(a) + f(b) + 2f\left(a + \frac{b-a}{2}\right) \right] \end{aligned}$$

It is easily verified that R_{21} can be written in terms of R_{11} as follows:

$$R_{21} = \frac{1}{2}[R_{11} + h_1 f(a + h_2)]$$

Similarly, $R_{31} = \frac{1}{2}[R_{21} + h_2[f(a + h_3) + f(a + 3h_3)]]$

The trend is now clear. In general, we can write **[Exercise]**:

$$\begin{aligned} R_{k1} &= \frac{1}{2}[R_{k-1,1} + h_{k-1}[f(a + h_k) + f(a + 3h_k) + \dots + f(a + (2^{k-1} - 1)h_k)]] \\ &= \frac{1}{2} \left[R_{k-1,1} + h_{k-1} \sum_{i=1}^{2^{k-2}} f(a + (2i - 1)h_k) \right]. \end{aligned} \tag{7.58}$$

Constructing the Romberg Table

The Romberg table can be generated row-by-row as shown below:

The 1st row:

$$R_{11} = \frac{1}{2}[(b-a)(f(a) + f(b))].$$

The 2nd row:

- (i) Compute R_{21} by setting $k = 1$ in (7.58):

$$R_{21} = \frac{1}{2}[R_{11} + h_1 f(a + h_2)].$$

- (ii) Compute R_{22} by combining R_{11} and R_{21} using (7.57):

$$R_{22} = R_{21} + \frac{R_{21} - R_{11}}{3}.$$

The 3rd row:

- (i) Compute R_{31} by setting $k = 3$ in (7.58):

$$R_{31} = \frac{1}{2}[R_{21} + h_2(f(a + h_3) + f(a + 3h_3))].$$

- (ii) Compute R_{32} by combining R_{21} and R_{31} using (7.57):

$$R_{32} = R_{31} + \frac{R_{31} - R_{21}}{3}.$$

- (iii) Compute R_{33} by combining R_{22} and R_{32} using (7.57):

$$R_{33} = R_{32} + \frac{R_{32} - R_{22}}{15}.$$

In general, **compute the k th row** as:

- (i) Compute R_{k1} from $R_{k-1,1}$ using (7.58):

$$R_{k1} = \frac{1}{2} \left[R_{k-1,1} + h_{k-1} \sum_{i=1}^{2^{k-2}} f(a + (2i-1)h_k) \right].$$

- (ii) Compute $R_{k2}, R_{k3}, \dots, R_{kk}$ using (7.57):

$$R_{kj} = R_{k,j-1} + \frac{R_{k,j-1} - R_{k-1,j-1}}{4^{k-1} - 1}, \quad j = 2, 3, \dots, k.$$

Algorithm 7.17. Romberg Integration

Inputs: $\begin{cases} \text{(i) } f(x) - \text{Integrand} \\ \text{(ii) } a, b - \text{The endpoints of the interval} \\ \text{(iii) } n - \text{A positive integer} \\ \text{(iv) } \epsilon - \text{A tolerance} \end{cases}$

Outputs: The Romberg numbers $\{R_{kj}\}$

Step 0. Set $h = b - a$.

Step 1. Compute the 1st row:

Step 2. For $k = 2, 3, \dots, n$ do (Compute 2nd through n th row).

 Compute $R_{k1} = \left[R_{k-1,j-1} + h \sum_{i=1}^{2^{k-2}} f(a + (2i-1)h) \right]$ (**Formula 7.58**)

 For $j = 2, \dots, k$ do

 Compute $R_{kj} = R_{k,j-1} + \frac{R_{k,j-1} - R_{k-1,j-1}}{4^{j-1} - 1}$ (**Formula 7.57**)

 Stop if $|R_{k,k} - R_{k-1,k-1}| < \epsilon$.

End

End

Example 7.18

Compute

$$I = \int_1^{1.5} x^2 \ln x \, dx \text{ with } n = 3$$

Input Data: $\begin{cases} f(x) = x^2 \ln x; a = 1, b = 1.5 \\ n = 3; h = 1.5 - 1 = 0.5 \end{cases}$

Solution.

Step 1. Compute the 1st row of the Romberg table. $R_{11} = \frac{1}{2}(1.5-1)[f(1)+f(1.5)] = 0.2280741$

Step 2. Compute the second row of the Romberg table.

2.1 Compute $R_{21} = \frac{1}{2}(R_{11} + (1.5-1) \left(f\left(1 + \frac{1.5-1}{2}\right) \right)) = 0.2012025$

2.2 Compute $R_{22} = R_{21} + \frac{R_{21}-R_{11}}{r^1-1}$

Step 3. Compute the 3rd row of the Romberg table.

3.1 Compute $R_{31} = 0.1944945$

3.2 Compute $R_{32} = 0.1922585$

3.3 Compute $R_{33} = 0.192259$

Romberg Table for Example 7.16

$R_{11} = 0.2280741$		
$R_{21} = 0.2012025$		$R_{22} = 0.1922453$
$R_{31} = 0.1944945$		$R_{32} = 0.1922585$
		$R_{33} = 0.1922593$

Exact value of I (up to six decimal figures) = 0.192259

7.10 Adaptive Quadrature Rule

Adaptive quadrature rule is a way to adaptively adjust the value of the step-size h such that the error becomes less than a prescribed tolerance (say ϵ).

A generic procedure can be stated as follows:

A Generic Adaptive Quadrature Process

Step 1. Pick a favorite quadrature rule F .

Step 2. Compute I_F^h , an approximation to $I = \int_a^b f(x)dx$ with step size h applying the rule F .

Step 3. Compute an error estimate:

$$E_F^h = |I - I_F^h|$$

If $E_F^h < \epsilon$, the given error tolerance, then stop and accept E_F^h as the approximation.

Step 4. If $E_F^h \geq \epsilon$, then compute $E_F^{\frac{h}{2}}$ as follows:

Divide the interval into two equal subintervals and apply the chosen rule to each subinterval. Add the results.

If the approximate value of each integral is less than $\frac{\epsilon}{2}$, then accept $I_F^{\frac{h}{2}}$ as the final approximation.

If not, continue the process of subdividing until the desired accuracy of ϵ is obtained or a stopping criterion is satisfied.

We will now develop a stopping criterion when Simpson's rule is adopted as the favorite rule.

Adaptive Simpson's Quadrature Rule

Using Simpson's rule as our favorite technique we will derive a computable stopping criterion.

With the step-size h , that is, with the points of subdivisions as $a, a + h$, and b ,

$$\begin{array}{ccccc} & & \text{X} & & \\ & & | & & \\ \text{X} & \text{-----} & \text{X} & \text{-----} & \text{X} \\ a & & a+h & & b \end{array}$$

we have Simpson's rule approximation:

$$I_S^h = \frac{h}{3} [f(a) + 4f(a+h) + f(b)] \quad (\text{Formula 7.70})$$

with **Error** $= E_S^h = -\frac{h^5}{90} f^{(iv)}(\eta)$, where $a < \eta < b$.

Now let's use 4 intervals; that is, this time the points of subdivision are:

$$a, \quad a + \frac{h}{2}, \quad a + h, \quad a + \frac{3h}{2}, \quad b.$$

$$\begin{array}{ccccccc} & & \text{X} & & \text{X} & & \text{X} & & \text{X} & & \text{X} \\ & & | & & | & & | & & | & & | \\ \text{X} & \text{-----} & \text{X} & \text{-----} & \text{X} & \text{-----} & \text{X} & \text{-----} & \text{X} \\ a & & a+\frac{h}{2} & & a+h & & a+\frac{3h}{2} & & b \end{array}$$

Then, using **Composite Simpson's rule** (Formula 7.73) with $n = 4$, that is, with the length of each subinterval $\frac{h}{2}$, we have

$$I_S^{\frac{h}{2}} = \frac{h}{6} \left[f(a) + 4f\left(a + \frac{h}{2}\right) + 2f(a+h) + 4f\left(a + \frac{3h}{2}\right) + f(b) \right]$$

and **error this time**, $E_S^{\frac{h}{2}} = -\frac{1}{16} \left(\frac{h^5}{90} \right) f^{(4)}(\bar{\eta})$, where $a < \bar{\eta} < b$.

That is, $I - I_S^{\frac{h}{2}} = -\frac{1}{16} \left(\frac{h^5}{90} \right) f^{(4)}(\bar{\eta})$, where $a < \bar{\eta} < b$ through (7.76).

Now the question is: *how well does $I_S^{\frac{h}{2}}$ approximate I over I_S^h ?*

Assume that $\eta \approx \bar{\eta}$. Then, we have

$$I_{h/2}^S - \frac{1}{16} \left(\frac{h^5}{90} \right) f^{(4)}(\eta) \approx I_h^S - \frac{h^5}{90} f^{(4)}(\eta)$$

$$\text{or} \quad \frac{h^5}{90} f^{(4)}(\eta) \approx \frac{16}{15} [I_S^h - I_s^{h/2}].$$

From (7.76), we then have

$$|I - I_s^{h/2}| \approx \frac{1}{15} |I_S^h - I_s^{h/2}| \quad (7.59)$$

This is a rather pleasant result, because it is easy to compute $|I_S^h - I_s^{\frac{h}{2}}|$. Suppose this quantity is denoted by δ .

Then, if we choose h such that $\delta = 15\epsilon$, we see that $|E_S^{\frac{h}{2}}| < \epsilon$.

Thus a **stopping criterion** for the adaptive Simpson's rule will be:

Stop the process of subdivision as soon as the

$$\frac{1}{15} |I_S^h - I_s^{\frac{h}{2}}| < \epsilon. \quad (7.60)$$

Example 7.19

Approximate $\int_0^{\frac{\pi}{2}} \cos x dx$ using Simpson's adaptive quadrature rule with an accuracy of $\epsilon = 10^{-3}$.

Step 1. Simpson's rule approximation with two intervals.

Input Data: $f(x) = \cos x; a = 0, b = \frac{\pi}{2}; h = \frac{\pi}{4}$.

$$x_0 = a = 0, x_1 = \frac{\pi}{4}, x_2 = b = \frac{\pi}{2}.$$

Formula to be used: $I_S^h = \frac{h}{3} [f(a) + 4f(\frac{a+b}{2}) + f(b)]$.

$$\begin{aligned} I_S^h &= \frac{\pi}{12} \left[f(0) + 4f\left(\frac{\pi}{4}\right) + f\left(\frac{\pi}{2}\right) \right] \quad \left(\text{Note that } h = \frac{\pi}{4} \right). \\ &= \frac{\pi}{12} \left[\cos(0) + 4\cos\left(\frac{\pi}{4}\right) + \cos\left(\frac{\pi}{2}\right) \right] \\ &= 1.0023 \end{aligned}$$

Step 2. Simpson's rule approximation with four intervals.

Input Data: $f(x) = \cos x; a = 0, b = \frac{\pi}{2}$.

$$x_0 = a = 0, x_1 = \frac{\pi}{8}, x_2 = \frac{\pi}{4}, x_3 = \frac{3\pi}{8}, x_4 = b = \frac{\pi}{2}.$$

Formula to be used:

$$\begin{aligned} I_S^{\frac{h}{2}} &= \frac{\pi}{6} \left[f(0) + 4f\left(\frac{\pi}{8}\right) + 2f\left(\frac{\pi}{4}\right) + 4f\left(\frac{3\pi}{8}\right) + f\left(\frac{\pi}{2}\right) \right] \\ &= \frac{\pi}{24} \left[\cos(0) + 4\cos\left(\frac{\pi}{8}\right) + 2\cos\left(\frac{\pi}{4}\right) + 4\cos\left(\frac{3\pi}{8}\right) + \cos\left(\frac{\pi}{2}\right) \right] = 1.0001 \end{aligned}$$

Now $\frac{1}{15} \left| I_S^h - I_S^{\frac{h}{2}} \right| = 1.4667 \times 10^{-4}$. Since $\epsilon = 10^{-3}$, we can stop.

Note the **actual error** is

$$\left| \int_a^b f(x)dx - I_S^{\frac{h}{2}} \right| = 10^{-4}$$

7.11 Gaussian Quadrature

So far, we have discussed the quadrature rules of the form:

$$\int_a^b f(x)dx \approx w_0 f(x_0) + w_1 f(x_1) \dots + w_{n-1} f(x_{n-1}), \quad (7.61)$$

where the numbers x_0, x_1, \dots, x_{n-1} , called **nodes**, are **given**, and the numbers, w_0, \dots, w_{n-1} , called **weights**, are determined by a **quadrature rule**.

For example, in

• **Trapezoidal Rule:** $\int_a^b f(x)dx \approx \frac{h}{2}[f(x_1) + f(x_2)] = \frac{h}{2}f(x_0) + \frac{h}{2}f(x_1)$, the nodes x_0 and x_1 are specified and the weights $w_0 = w_1 = \frac{h}{2}$ are determined by the rule.

In fact, recall that in this case, $w_0 = w_1 = \frac{h}{2}$ are determined as follows:

$$\begin{aligned} w_0 &= \int_{x_0}^{x_1} \frac{x - x_1}{x_0 - x_1} dx = \frac{x_1 - x_0}{2} = \frac{h}{2} \\ w_1 &= \int_{x_0}^{x_1} \frac{x - x_0}{x_1 - x_0} dx = \frac{x_1 - x_0}{2} = \frac{h}{2} \end{aligned}$$

• **Simpson's Rule:** $\int_a^b f(x)dx \approx \frac{h}{3}[f(x_1) + 4f(x_1) + f(x_2)] = \frac{h}{3}f(x_0) + \frac{4h}{3}f(x_1) + \frac{h}{3}f(x_2)$.

The nodes x_0, x_1 , and x_2 are specified and the weights $w_1 = w_3 = \frac{h}{3}$ and $w_2 = \frac{4h}{3}$ are determined by Simpson's rule.

It is natural to wonder if we can devise a quadrature rule by determining both the nodes and the weights. *The idea is to obtain a quadrature rule that is exact for all polynomials of degree less than or equal to $2n - 1$* , which is the largest class of polynomials for which one can hope for the rules to be exact. This is because, we have $2n$ parameters: **n nodes** and **n weights** and a polynomial of degree $2n - 1$ can contain at most $2n$ parameters. The process is known as **Gaussian Quadrature** rule.

7.11.1 Derivation of Gaussian Quadrature Rule

We first derive this rule in the simple cases when $n = 2$ and $n = 3$ with the interval $[-1, 1]$.

Case n=2. Here the abscissas is x_0 and x_1 . The weights w_0 and w_1 are to be found such that the rule will be exact for all polynomials of degree less than or equal to 3.

Since the polynomials $1, x, x^2$, and x^3 form a basis of all polynomials of degree less than or equal to 3, we can set $f(x) = 1, x, x^2$ and x^3 successively in the formula (7.61) to determine w_0 and w_1 .

- $f(x) = 1$: $\int_{-1}^1 1 \, dx = w_0 f(x_0) + w_1 f(x_1)$ or $2 = w_0 + w_1$.
- $f(x) = x$: $\int_{-1}^1 x \, dx = w_0 f(x_0) + w_1 f(x_1)$ or $0 = w_0 x_0 + w_1 x_1$.
- $f(x) = x^2$: $\int_{-1}^1 x^2 \, dx = w_0 f(x_0) + w_1 f(x_1)$ or $\frac{2}{3} = w_0 x_0^2 + w_1 x_1^2$.
- $f(x) = x^3$: $\int_{-1}^1 x^3 \, dx = w_0 f(x_0) + w_1 f(x_1)$ or $0 = w_0 x_0^3 + w_1 x_1^3$.

Thus, we have the following systems of nonlinear equations to solve:

$$\begin{aligned}
 w_0 + w_1 &= 2 \\
 w_0 x_0 + w_1 x_1 &= 0 \\
 w_0 x_0^2 + w_1 x_1^2 &= \frac{2}{3} \\
 w_0 x_0^3 + w_1 x_1^3 &= 0
 \end{aligned} \tag{7.62}$$

A solution of this system of equations is

$$w_0 = w_1 = 1; x_0 = -\frac{1}{\sqrt{3}}, x_1 = \frac{1}{\sqrt{3}}.$$

Thus, for $n = 2$, we have the **2-point Gaussian Quadrature rule**

$$\int_{-1}^1 f(x)dx \approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right).$$

Case n=3. Here we seek to find x_0, x_1 , and x_2 ; and w_0, w_1 , and w_2 such that the rule will be exact for all polynomials of degree up to five (note that $n = 3$; so $2n - 1 = 5$).

Taking $f(x) = 1, x, x^2, x^3, x^4$, and x^5 , and proceeding exactly as above, we can show [Exercise] that the following systems of equations are obtained:

$$\begin{aligned} w_0 + w_1 + w_2 &= 2 \\ w_0x_0 + w_1x_1 + w_2x_2 &= 0 \\ w_0x_0^2 + w_1x_1^2 + w_2x_2^2 &= \frac{2}{3} \\ w_0x_0^3 + w_1x_1^3 + w_2x_2^3 &= 0 \\ w_0x_0^4 + w_1x_1^4 + w_2x_2^4 &= \frac{2}{5} \\ w_0x_0^5 + w_1x_1^5 + w_2x_2^5 &= 0 \end{aligned} \tag{7.63}$$

The above **nonlinear system** is rather difficult to solve. However, it turns out that the system will be satisfied if x_0, x_1 , and x_2 are chosen as the roots of an orthogonal polynomial, called **Legendre polynomial** and with this particular choice of x 's, the weights, w_0, w_1 , and w_2 are computed as

$$w_i = \int_{-1}^1 L_i(x)dx, \quad i = 0, 1, 2$$

where $L_i(x)$ is the i th Lagrange polynomial of degree 3. This is also true for $n = 2$, and in fact for any n , as the following discussions show.

7.11.2 Determining Nodes and Weights via Legendre Polynomials

We have encountered the notion of orthogonal polynomials in Chapter 6 in the context of interpolation with the zeros of the orthogonal Chebyshev polynomial. Recall that when the Chebyshev zeros are used as the nodes of interpolation, the error can be substantially reduced. Here we show that the zeros of another orthogonal polynomial, called the **Legendre polynomial**, plays a vital role in Gaussian quadrature. We first remind the readers of the definition of an orthogonal polynomial.

A polynomial $P_n(x)$ of degree n is an **orthogonal polynomial** with respect to the weight function $w(x)$ in $[a, b]$ if, for each polynomial $P_m(x)$ of degree $m < n$,

$$\int_a^b w(x) P_n(x) P_m(x) dx = 0$$

We will discuss more about orthogonal polynomials later in *Chapter 10*.

As in the case of Chebyshev polynomials, the Legendre polynomials can also be generalized recursively.

Recursion to generate Legendre polynomials

Set $P_0(x) = 1$, $P_1(x) = x$.

Then, the Legendre polynomials $\{P_k(x)\}$ can be generated from the recursive relation:

$$P_{k+1}(x) = \frac{(2k+1)xP_k(x) - kP_{k-1}(x)}{k+1}, \quad k = 1, 2, 3, \dots, n.$$

Thus,

$$\mathbf{k=1:} \quad P_2(x) = \frac{3xP_1(x) - P_0(x)}{2} = \frac{3}{2}x^2 - \frac{1}{2}. \quad (\text{Legendre polynomial of degree 2})$$

$$\mathbf{k=2:} \quad P_3(x) = \frac{5}{2} \left(x^3 - \frac{3}{5}x \right). \quad (\text{Legendre polynomial of degree 3})$$

and so on.

Choosing the Nodes and Weights

The following theorem shows *how to choose the nodes* x_0, x_1, \dots, x_{n-1} and *the weights* w_0, w_1, \dots, w_{n-1} such that the quadrature formula $\sum_{i=0}^{n-1} w_i f(x_i)$ is exact for all polynomials of degree less than or equal to $2n-1$.

Theorem 7.20. (Choosing the Nodes and Weights in n -point Gaussian Quadrature)

Let

- (i) the nodes x_0, x_1, \dots, x_{n-1} be chosen as the n -zeros of the n th degree Legendre polynomial $P_n(x)$ of degree n .
- (ii) the weights w_0, w_1, \dots, w_{n-1} be defined by

$$w_i = \int_{-1}^1 L_i(x) dx, \quad i = 0, \dots, n-1$$

where $L_i(x)$ is the i th Lagrange polynomial, each of degree $(n - 1)$, given by:

$$L_i(x) = \prod_{j=1}^{n-1} \frac{(x - x_j)}{x_i - x_j}, \quad j \neq i \quad (7.64)$$

Then

for any polynomial $P(x)$ of degree at most $(2n - 1)$, we have

$$\int_{-1}^1 P(x) dx = \sum_{i=0}^{n-1} w_i P(x_i). \quad (7.65)$$

Proof. Case 1: First, assume that $P(x)$ is a polynomial of degree at most $n - 1$. Write $P(x)$ as:

$$P(x) = L_0(x)P(x_0) + L_1(x)P(x_1) + \cdots + L_{n-1}(x)P(x_{n-1}), \quad (7.66)$$

where x_0, x_1, \dots, x_{n-1} are the zeros of the Legendre polynomial of degree n . This representation is exact, since the error term

$$P^{(n)}(\xi_x)(x - x_0) \cdots (x - x_{n-1})$$

is zero by virtue of the fact that the n -th derivative $P^{(n)}(x)$ of $P(x)$, which is a polynomial of degree at most $n - 1$, is zero.

So, integrating (7.84) from $a = -1$ to $a = 1$ and noting that $P(x_i), i = 0, \dots, n - 1$ are constants, we obtain

$$\begin{aligned} \int_{-1}^1 P(x) dx &= P(x_0) \int_{-1}^1 L_0(x) dx + P(x_1) \int_{-1}^1 L_1(x) dx + \cdots + P(x_{n-1}) \int_{-1}^1 L_{n-1}(x) dx \\ &= \sum_{i=0}^{n-1} w_i P(x_i). \end{aligned}$$

Remembering that $w_i = \int_{-1}^1 L_i(x) dx$, $i = 0, \dots, n - 1$, the above equation becomes

$$\int_{-1}^1 P(x) dx = w_0 P_0(x) + w_1 P_1(x) + \cdots + w_{n-1} P_{n-1}(x).$$

Case 2: Next, let's assume that the degree of $P(x)$ is at most $2n - 1$, but at least n . Then $P(x)$ can be written in the form:

$$P(x) = Q_{n-1}(x)P_n(x) + R_{n-1}(x) \quad (7.67)$$

where $P_n(x)$ is a n th-degree Legendre polynomial and $Q_{n-1}(x)$ and $R_{n-1}(x)$ are polynomials of degree at most $n-1$. Substituting $x = x_i$ in (7.85), we get

$$P(x_i) = Q_{n-1}(x_i)P_n(x_i) + R_{n-1}(x_i)$$

Since x_i are the zeros of $P_n(x)$, $P_n(x_i) = 0$. Thus, $P(x_i) = R_{n-1}(x_i)$.

From (7.85), we have

$$\int_{-1}^1 P(x)dx = \int_{-1}^1 Q_{n-1}(x)P_n(x)dx + \int_{-1}^1 R_{n-1}(x)dx$$

By the orthogonal property of the Legendre polynomial $P_n(x)$, $\int_{-1}^1 Q_{n-1}(x)P_n(x)dx = 0$.

$$\text{So, } \int_{-1}^1 P(x)dx = \int_{-1}^1 R_{n-1}(x)dx.$$

Again, since $R_{n-1}(x)$ is a polynomial of degree at-most $n-1$, we can write, by Case I:

$$\int_{-1}^1 R_{n-1}(x)dx = \sum_{i=0}^{n-1} w_i R_{n-1}(x_i)$$

Thus,

$$\int_{-1}^1 P(x)dx = \sum_{i=0}^{n-1} w_i R_{n-1}(x_i) = \sum_{i=0}^{n-1} w_i P(x_i).$$

□

Gauss-Legendre n-point Rule

Step 1. Pick n

Step 2. Compute the n zeros, x_0, x_1, \dots, x_{n-1} of the n th degree Legendre polynomial.

Step 3. Compute the corresponding weights:

$$w_i = \int_{-1}^1 L_i(x)dx \quad , \quad i = 0, 1, \dots, n-1$$

where w_i is the i th Lagrange polynomial.

Step 4. Compute the integral approximation:

$$I_{GL}^n = w_0 f(x_0) + w_1 f(x_1) + \cdots + w_{n-1} f(x_{n-1}).$$

Remarks: The zeros of the Legendre polynomials and the corresponding weights are often available from a table.

Working with an arbitrary interval $[a, b]$: So far our discussion on Gaussian quadrature has been to approximate $\int_{-1}^1 f(x) dx$.

If we need to approximate $\int_a^b f(x)$ by Gaussian quadrature, we must transform the interval $[a, b]$ into $[-1, 1]$ as follows:

Substitute

$$x = \frac{1}{2}[(b-a)t + a + b].$$

Then

$$dx = \frac{b-a}{2} dt.$$

$$\int_a^b f(x) dx = \int_{-1}^1 f\left(\frac{(b-a)t}{2} + a + b\right) \frac{(b-a)}{2} dt$$

Thus, to approximate $I = \int_a^b f(x)$ using Gauss-Quadrature rule, do the following:

Step 1. Transform the integral $\int_a^b f(x) dx$ to $\frac{b-a}{2} \int_{-1}^1 f\left(\frac{(b-a)t}{2} + a + b\right) dt = \frac{b-a}{2} \int_{-1}^1 F(t) dt$.

Step 2. Apply n -point Gauss-Quadrature rule to $\int_{-1}^1 F(t) dt$.

Step 3. Multiply the result from Step 2 by $\frac{b-a}{2}$ to approximate I .

Error in n -point Gaussian Quadrature: It can be shown that the error in n -point Gaussian quadrature is given by

$$E_C^n = \frac{f^{(2n)}(\xi_x)}{2n!} \int_a^b [\psi(x)]^2 dx$$

where $\psi(x) = (x - x_0)(x - x_1) \dots (x - x_n)$, and $a < \xi_x < b$.

Example 7.21

Apply 3-point Gaussian Quadrature rule to approximate $\int_1^2 e^{-x} dx$.

Input Data:
$$\begin{cases} f(x) &= e^{-x}; a = 1, b = 2 \\ n &= 3. \end{cases}$$

Solution.

Step 1. Transform the integral to change the interval from $[1, 2]$ to $[-1, 1]$.

$$\begin{aligned} \int_1^2 f(x) dx &= \int_1^2 e^{-x} dx = \frac{1}{2} \int_{-1}^1 f\left(\frac{t+3}{2}\right) dt \\ &= \frac{1}{2} \int_{-1}^1 e^{-(\frac{t+3}{2})} dt \end{aligned}$$

Step 2. Approximate $\int_{-1}^1 e^{-(\frac{t+3}{2})} dt$ using Gauss-Legendre rule with $n = 3$.

Three zeros of the Legendre polynomial $P_3(t)$ are:

$$t_0 = -\sqrt{\frac{3}{5}}, t_1 = 0, t_2 = \sqrt{\frac{3}{5}}.$$

The corresponding weights are (*in 4 significant digits*): $w_0 = 0.5555, w_1 = 0.8889, w_2 = 0.5555$.

Thus, using the 3-point Gauss-Legendre rule we have:

$$\int_{-1}^1 e^{-(\frac{t+3}{2})} dt = w_0 e^{-(\frac{t_0+3}{2})} + w_1 e^{-(\frac{t_1+3}{2})} + w_2 e^{-(\frac{t_2+3}{2})} = 0.4652$$

Step 3. Compute $\int_1^2 f(x) dx = \frac{0.4652}{2} = 0.2326$.

Exact Solution: $\int_1^2 e^{-x} dx = 0.2325$ (up to four significant digits).

Relative Error: $\frac{|0.2325 - 0.2326|}{|0.2325|} = 4.3011 \times 10^{-4}$

7.12 Exercises on Part II

7.1. (**Computational**) Suppose $x_0 = 0$, $x_1 = 1$, and $x_2 = 2$ are the three nodes and we are required to find a quadrature rule of the form $\int_0^2 f(x)dx = w_0f(0) + w_1f(1) + w_2f(2)$ which should be exact for all polynomials of degree less than or equal to 2. What are w_0 , w_1 , and w_2 ?

7.2. (**Computational**) Suppose a quadrature rule of the form $\int_0^1 f(x)dx = w_0f(0) + w_1f(x_1)$ has to be devised such that it is exact for all polynomials of degree less than or equal to 1. What are the values of x_1 and w_0 ?

7.3. (**Computational**) Is it possible to devise a quadrature rule of the form $\int_1^{20} f(x)dx = w_0f(1) + w_1f(2) + w_2f(3) + \cdots + w_{19}f(20)$ that is exact for all polynomials of as high of a degree as possible? Determine why solving the system for the weights will not be computationally effective.

7.4. (**Computational**) Using the appropriate data from **Exercise 7.1** in (Exercises on Numerical Differentiation),

A. Approximate the following integrals:

(i) $\int_0^3 \sin x dx$

(ii) $\int_0^3 \cos x dx$

(iii) $\int_0^3 xe^x dx$

by each of the following quadrature rules:

(a) trapezoidal rule with $h = 3$.

(b) Simpson's rule with $h = 1.5$.

(c) Simpson's $\frac{3}{8}$ th rule with $h = 1$.

(d) corrected trapezoidal rule with $h = 3$.

B. In each case (above), compute the maximum absolute error and compare these errors with those obtained by the analytical formulas.

7.5. (**Computational**) Using the appropriate data of **Exercise 7.2** in approximate the following integrals,

(a) $\int_0^\pi \sqrt{x} \sin x dx$

(b) $\int_0^\pi \sin(\sin \sin(x)) dx$

$$(c) \int_0^2 \left[\sqrt{x} + \sqrt{x + \sqrt{x}} \right] dx$$

by each of the following rules:

- (i) trapezoidal rule with $h = \pi$.
- (ii) Simpson's rule with $h = \frac{h}{2}$.
- (iii) Boole's rule with $h = \frac{\pi}{4}$.

7.6. (**Analytical**) Derive the trapezoidal and Simpson's rule using Newton's interpolation with forward differences.

7.7. (**Analytical**)

- (a) Derive Boole's rule.
- (b) Prove that Boole's rule has a degree of precision of 5.

7.8. (**Analytical**) Derive Simpson's $\frac{3}{8}$ th rule with error formula.

7.9. (**Analytical**) Drive the corrected trapezoidal rule with error formula.

7.10. Derive error formula for Simpson's rule using Taylor's Series of order 3 about x_1 .

7.11. (**Computational**) Using the appropriate data set of **Exercise 1** in **Section 7.3**,

A. Approximate the following integrals:

- (i) $\int_0^3 \sin x dx$
- (ii) $\int_0^3 \cos x dx$
- (iii) $\int_0^3 x e^x dx$

by the

- (a) composite trapezoidal rule with $h = 0.5$.
- (b) composite Simpson's rule with $h = 0.5$.

B. Compute the maximum absolute error in each case above, and compare the errors with the actual absolute errors obtained by analytical rules.

C. For each of the integrals in part A, determine how the result obtained by the basic rule compare with those by the composite rule?

7.12. () Determine h and N to approximate the integral $\int_0^1 e^{-t^2} dt$ within an accuracy of $G = 10^{-5}$ using the

- (i) composite trapezoidal rule.

(ii) composite Simpson's rule.

Which one is better and why?

7.13. **True or False.** The degree of precision of a quadrature rule is the degree of the interpolating polynomial on which the rule is based. Explain your answer.

7.14. (a) Show that when Simpson's rule is applied to $\int_0^{2\pi} \sin x dx$, there is zero error, assuming no round-off.

(b) Obtain a generalization to the other integrals $\int_a^b f(x) dx$. Show that Simpson's rule is exact for $\int_0^{\frac{\pi}{2}} \cos^2 x dx$.

7.15. Prove that the quadrature rule of the form $\int_1^3 f(x) dx \approx w_0 f(1) + w_1 f(2) + w_2 f(3)$ is exact for all polynomials of as high a degree as possible, and is nothing but Simpson's rule.

7.13 MATLAB Problems on Numerical Quadrature and their Applications

M7.1