Semester II

Session: 2023-24

Name of Program		**Program Code**	
Name of the Course	**Numerical Analysis**	**Course Code**	**23MAT502SE01**
Hours per Week	**04**	**Credits**	**03**
Maximum Marks	**50**	**Time of Examinations**	**03 Hours**

Note:
The question paper will consist of four sections. Each of the first three sections (*I-III*) will contain two questions and the students shall be asked to attempt one question from each section. Section – IV (Question No. 7) will contain three short answer type questions without any internal choice covering the entire syllabus and shall be compulsory

Course Learning Outcomes (CLO):

- **CLO 1** Learn about interpolation with equal and unequal intervals.
- **CLO 2** Apply forward, backward, central and divided difference formulae for interpolation.
- **CLO 3** Apply standard probability distributions to the concerned problems.
- **CLO 4** Understand the method of numerical differentiation and various methods for finding solution of eigen value problems..
- **CLO 5** Know how to solve integration and ordinary differential equation using numerical data

Unit 1:
Finite Differences operators and their relations. Finding the missing terms and effect of error in a difference tabular values, Interpolation with equal intervals: Newton's forward and Newton's backward interpolation formulae. Interpolation with unequal intervals: Newton's divided difference, Lagrange's Interpolation formulae, Hermite Formula.

Unit 2:
Central Differences: Gauss forward and Gauss's backward interpolation formulae, Sterling, Bessel Formula.
Numerical Differentiation: Derivative of a function using interpolation formulae.
Eigen Value Problems: Power method, Jacobi's method, Given's method, House-Holder's method, QR method, Lanczos method.

Unit 3:
Numerical Integration: Newton-Cote's Quadrature formula, Trapezoidal rule, Simpson's one- third and three-eighth rule, Chebychev formula, Gauss Quadrature formula.
Numerical solution of ordinary differential equations: Single step methods-Picard's method, Taylor's series method, Euler's method, Runge-Kutta Methods. Multiple step methods: Predictor-corrector method, Modified Euler's method, Milne-Simpson's method.

Part-B (Practical)

Time: 3 Hours **Max. Marks: 25**

There will be a separate practical paper consisting of implementation of numerical methods in C Programming Language, studied in the theory paper **24MAT402SE01** (Part-A).

References:

1. Babu Ram, Numerical Methods: Pearson Publication.
2. R.S. Gupta, Elements of Numerical Analysis, Macmillan's India 2010.
3. M. K. Jain, S.R.K. Iyengar and R.K. Jain, Numerical Method, Problems and Solutions, New Age International (P) Ltd., 1996
4. M. K. Jain, S.R.K. Iyengar and R.K. Jain, Numerical Method for Scientific and Engineering Computation, New Age International (P) Ltd., 1999
5. C. E. Froberg, Introduction to Numerical Analysis (2nd Edition).
6. Melvin J. Maaron, Numerical Analysis-A Practical Approach, Macmillan Publishing Co., Inc., New York
7. R.Y. Rubnistein, Simulation and the Monte Carlo Methods, John Wiley, 1981

Unit - 1

Finite differences operators :-

Let $y = f(x)$ be a function of x then the independent variable x is called argument of the function and the value of function is called Entry.

$$y = f(x)$$

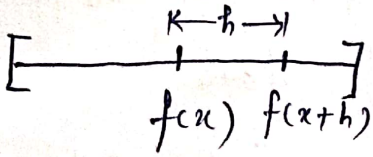
↓ Entry ↓ Argument

$x = a, a+h, a+2h, \ldots$

$y = f(a), f(a+h), f(a+2h), \ldots$

The differences between consecutive values of argument x is called Interval differencing.

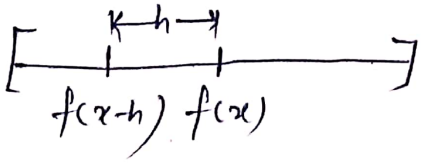
→ Forward difference :-



$$\Delta f(x) = f(x+h) - f(x)$$

Δ = forward differences operator

→ Backward differences :



$$\nabla f(x) = f(x) - f(x-h)$$

∇ = Backward difference operators.

* Forward difference Table :

Arguments	Entry	first diff.	Second diff.	3rd diff.
0	y_0	$y_1 - y_0 = \Delta y_0$	$\Delta y_1 - \Delta y_0 = \Delta^2 y_0$	
1	y_1			$\Delta^3 y_0 = \Delta^2 y_1 - \Delta^2 y_0$
2	y_2	$y_2 - y_1 = \Delta y_1$	$\Delta y_2 - \Delta y_1 = \Delta^2 y_1$	
3	y_3	$y_3 - y_2 = \Delta y_2$	$\Delta y_3 - \Delta y_2 = \Delta^2 y_2$	$\Delta^3 y_1 = \Delta^2 y_2 - \Delta^2 y_1$
4	y_4	$y_4 - y_3 = \Delta y_3$		

Example :

Construct a forward difference table for the following data :

x	48	50	53	60	65
$f(x)$	2.871	2.404	2.083	1.862	1.712

✱ Backward difference Table :-

x	y	∇y	$\nabla^2 y$
0	y_0		
1	y_1	$y_1 - y_0 = \nabla y_1$	$\nabla y_2 - \nabla y_1 = \nabla^2 y_2$
2	y_2	$y_2 - y_1 = \nabla y_2$	$\nabla y_3 - \nabla y_2 = \nabla^2 y_3$
3	y_3	$y_3 - y_2 = \nabla y_3$	$\nabla y_4 - \nabla y_3 = \nabla^2 y$
4	y_4	$y_4 - y_3 = \nabla y_4$	

✱ Relation b/w operators :-

Δ = forward operator

∇ = Backward operator

E = Shift operator

1. $E = 1 + \Delta$

Proof :

$$E f(x) = f(x+h)$$

$$\Delta f(x) = f(x+h) - f(x)$$

$$\Delta f(x) = E f(x) - f(x)$$

$$\Delta f(x) = (E-1) f(x)$$

$$\boxed{\Delta + 1 = E}$$ ✱

2. $\nabla = 1 - E^{-1}$

Proof:

$$\nabla f(x) = f(x) - f(x-h)$$

$$E^{-1} f(x) = f(x-h) = f(x) - E^{-1} f(x)$$

$$\nabla f(x) = (1 - E^{-1}) f(x)$$

$$\boxed{\nabla = 1 - E^{-1}}$$

3. $\nabla = \Delta E^{-1}$

$$\nabla f(x) = f(x) - f(x-h)$$

$$= \Delta f(x-h)$$

$$\nabla f(x) = \Delta E^{-1} f(x)$$

$$\boxed{\nabla = \Delta E^{-1}}$$

Note:

$$E f(x) = f(x+h)$$

$$E^{-1} f(x) = f(x-h)$$

$$\Delta f(x) = f(x+h) - f(x)$$

$$\nabla f(x) = f(x) - f(x-h)$$

✧ Error In a difference Tabular Values :-

Example :- One of the following numbers is a misprint. Correct it.

1 2 4 8 16 26 42 64 93

Soln :-

$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$	$\Delta^4 f(x)$
1				
	1			
2		1		
	2		1	
4		2		1
	4		2	
8		4		-4
	8		-2	
16		2		6
	10		4	
26		6		-4
	16		0	
42		6		1
	22		1	
64		7		
	29			
93				

The sum of all values in column $\Delta^4 f(x)$ is zero and 6 is the numerically greatest value of this difference column which corresponds to functional value 16.

Hence, 16 is incorrect value.

$$(1-p)^4 = 1 - 4p + 6p^2 - 4p^3 + p^4$$

$$1, -4, 6, -4, 1.$$

Example

Find the error and correct the wrong figure in the following functional value:

$$2, 5, 10, 18, 26, 37, 50.$$

* Interpolation :-

Let us assume that the function $y = f(x)$ is known for certain values of x say for $x_0, x_1, x_2 \cdots, x_n$ as $f(x_0), f(x_1), \cdots f(x_n)$. The process of finding the values of $f(x)$ corresponds to $x = x_i$ where $x_0 < x_i < x_n$ with the help of given data is called Interpolation.

→ Newton-Gregory formula for forward Interpolation :-

Let $y = f(x)$ be a function of 'x'.

$$x = a, \; a+h, \; a+2h, \cdots a+nh.$$

$$y = f(x) = f(a), f(a+h), \cdots, f(a+nh).$$

Let us assume that $f(x)$ is a polynomial in 'x' of degree 'n'.

$$f(x) = A_0 + A_1(x-a) + A_2(x-a)(x-\overline{a+h})$$
$$+ A_3(x-a)(x-\overline{a+h}) + \cdots$$
$$\cdots + A_n(x-a)(x-\overline{a+h})(x-\overline{a+2h})(x-\overline{a+n-1}$$

Put $x = a$ in eq (1)

$$f(a) = A_0$$

Put $x = a+h$

$$f(a+h) = A_0 + A_1(a+h-a) + A_2(a+h-a)$$

$$f(a+h) = f(a) + A_1 h$$

$$A_1 h = f(a+h) - f(a)$$

$$A_1 = \frac{\Delta f(a)}{h} \quad - (2)$$

Put $x = a+2h$ in eq (1)

$$f(a+2h) = A_0 + 2A_1 h + 2A_2 h^2$$

$$f(a+2h) = f(a) + 2[\Delta f(a)] + 2A_2 h^2$$

$$A_2 = \frac{\Delta^2 f(a)}{2h^2}$$

A.

Figure 3.1

In reviewing these data, we might ask whether they could be used to provide a reasonable estimate of the population, say, in 1975 or even in the year 2020. Predictions of this type can be obtained by using a function that fits the given data. This process is called interpolation and is the subject of this chapter.

Now stating generally, the problem is: given a set of tabular values (x_0, y_0), (x_1, y_1), ... , (x_n, y_n) satisfying $y = f(x)$ where $f(x)$ is not known explicitly and it is asked to find the values of $f(x)$ corresponding to certain given values of x in the range $x_0 < x < x_n$. For this we find a function $g(x)$ in such a way that $f(x)$ and $g(x)$ agree at the set of tabulated values given. Then the required value of $f(x)$ at some point x is taken as the value of $g(x)$ at x. This process is called **interpolation**. If $g(x)$ is a polynomial then the process is called **polynomial interpolation**.

3.1 Interpolation and the Lagrange polynomial

One of the most useful and well-known classes of functions mapping the set of real numbers into itself is the algebraic polynomials, the set of functions of the form

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0.$$

where n is a nonnegative integer and $a_0, \ldots, a_n$ real constants. One reason for their importance is that they uniformly approximate continuous functions. By this we mean that given any function, defined and continuous on a closed and bounded interval, there exists a polynomial that is as "close" to the given function as desired.

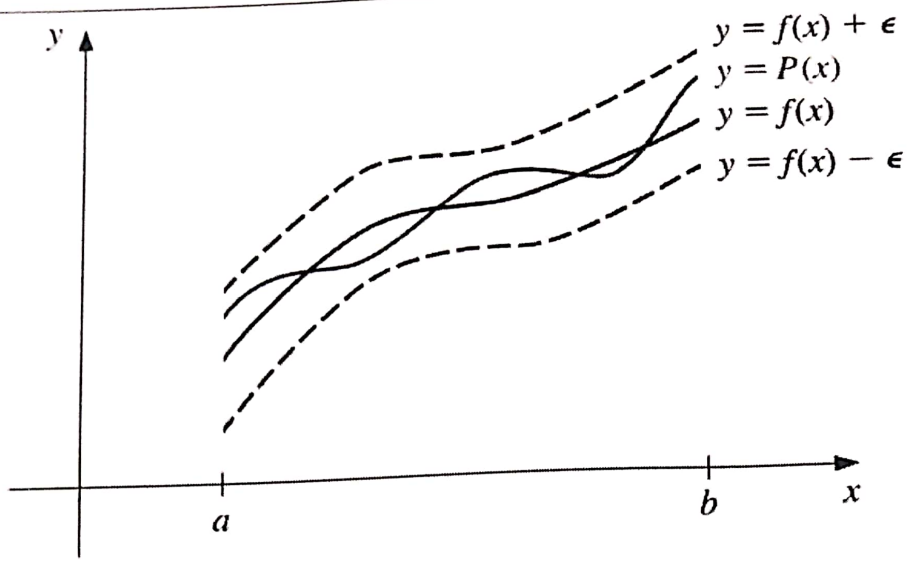


Figure 3.2

Theorem 3.1

(Weierstrass Approximation Theorem)

Suppose f is defined and continuous on $[a, b]$. For each $\epsilon > 0$, there exists a polynomial $P(x)$, with the property that

$$|f(x) - P(x)| < \epsilon, \quad for\ all\ x\ in\ [a, b].$$

Lagrange Interpolating Polynomials

The problem of determining a polynomial of degree one that passes through the distinct points (x_0, y_o) and (x_1, y_1) is the same as approximating a function f for which $f(x_0) = y_0$ and $f(x_1) = y_1$ by means of a first-degree polynomial **interpolating**, or agreeing with, the values of f at the given points. Using this polynomial for approximation within the interval given by the endpoints is called polynomial **interpolation**.

c.

n	$x_0, x_1 \ldots x_n$	$P_n(0.25)$
1	0.2, 0.3	-0.13869287
2	0.2, 0.3, 0.4	-0.132559734
3	0.2, 0.3, 0.4, 0.1	-0.13277477

d.

n	$x_0, x_1 \ldots x_n$	$P_n(0.9)$
1	0.8, 1.0	0.44086280
2	0.8, 1.0, 0.7	0.43841352
3	0.8, 1.0, 0.7, 0.6	0.44198500

2. a. We have $\sqrt{3} \approx P_4(\frac{1}{2}) = 1.7083.$

b. We have $\sqrt{3} \approx P_4(3) = 1.690607.$

c. Absolute error in part (a) is approximately 0.0237, and the absolute error in part (b) is 0.0414, so part (a) is more accurate.

3. $y = 4.25$

4. $P_2 = f(0.5) = 4$

5. $P_{0,1,2,3}(2.5) = 2.875$

3.3 Divided Differences

Iterated interpolation was used in the previous section to generate successively higher-degree polynomial approximations at a specific point. Divided-difference methods introduced in this section are used to successively generate the polynomials themselves.

Definition

For a given sequence $\{p_n\}_{n=0}^{\infty}$ the forward difference Δp_n (read "delta p_n") is defined by $\Delta p_n = p_{n+1} - p_n$, for $n \geq 0$.

Higher powers of the operator Δ are defined recursively by

$\Delta^k p_n = \Delta(\Delta^{k-1} p_n)$, for $k \geq 2$.

The definition implies that

$$\Delta^2 p_n = \Delta(p_{n+1} - p_n) = \Delta p_{n+1} - \Delta p_n$$
$$= (p_{n+2} - p_{n+1}) - (p_{n+1} - p_n).$$

So $\Delta^2 p_n = p_{n+2} - 2p_{n+1} + p_n$.

Divided Differences

Suppose that $P_n(x)$ is the nth interpolating polynomial that agrees with the function f at the distinct numbers $x_0, x_1, \ldots, x_n$. Although this polynomial is unique, there are alternate algebraic representations that are useful in certain situations. The divided differences of f with respect to $x_0, x_1, \ldots, x_n$ are used to express $P_n(x)$ in the form

$$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots$$
$$+ a_n(x - x_0) \ldots (x - x_{n-1}), \qquad (5)$$

for appropriate constants a_0, a_1, a_n. To determine the first of these constants, a_0, note that if $P_n(x)$ is written in the form of Eq. (5), then evaluating $P_n(x)$ at x_0 leaves only the constant term a_0; that is,

$$a_0 = P_n(x_0) = f(x_0);$$

Similarly, when $P(x)$ is evaluated at x_1, the only nonzero terms in the evaluation of $P_n(x_1)$ are the constant and linear terms

$$f(x_0) + a_1(x_1 - x_0) = P_n(x_1) = f(x_1)$$

So,

$$a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \tag{6}$$

We now introduce the divided-difference notation.

The zeroth divided difference of the function f with respect to x_i, denoted $f[x_i]$, is simply the value of f at x,:

$$f[x_i] = f(x_i) \tag{7}$$

The remaining divided differences are defined recursively; the first divided difference of f with respect to x_i and x_{i+1} is denoted $f[x_i, x_{i+1}]$ and defined as

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i} \tag{8}$$

The second divided difference, $f[x_i, x_{i+1}, x_{i+2}]$, is defined as

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}$$

Similarly, after the $(k-1)$st divided differences,

$f[x_i, x_{i+1}, \ldots, x_{i+k-1}, x_{i+k}]$ and $f[x_{i+1}, x_{i+2}, \ldots, x_{i+k-1}, x_{i+k}]$ have been determined, the ***k*th divided difference** relative to $x_i, x_{i+1}, x_{i+2} \ldots, x_{i+k}$ is

$$f[x_i, x_{i+1}, \ldots, x_{i+k-1}, x_{i+k}] = \frac{f[x_{i+1}, x_{i+2}, \ldots, x_{i+k}] - f[x_i, x_{i+1}, \ldots, x_{i+k-1}]}{x_{i+k} - x_i} \quad (9)$$

The process ends with the single nth divided difference,

$$f[x_0, x_1, \ldots, x_n] = \frac{f[x_1, x_2, \ldots, x_n] - f[x_0, x_1, \ldots, x_{n-1}]}{x_n - x_0}$$

Because of Eq. (6), we can write $a_1 = f[x_0, x_1]$, just as a_0 can be expressed as $a_0 = f(x_0) = f[x_0]$. Hence, the interpolating polynomial in Eq. (5) is

$$P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)(x - x_1) \ldots (x - x_{n-1}).$$

As might be expected from the evaluation of a_0 and a_1, the required constants are

$$a_k = f[x_0, x_1, x_2, \ldots, x_k],$$

for each $k = 0,1, \ldots, n$. So, $P_n(x)$ can be rewritten in a form called **Newton's Divided Difference**:

$$P_n(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](x - x_0) \ldots (x - x_{k-1}) \quad (10)$$

The value of $f[x_0, x_1, \ldots, x_k]$ is independent of the order of the numbers $x_0, x_1, \ldots, x_k$.

The generation of the divided differences is outlined in Table 3.9. Two fourth and one fifth difference can also be determined from these data.

Table 3.9

x	$f(x)$	First divided difference	Second divided difference	Thrid divided difference
x_0	$f[x_0]$			
		$f[x_0x_1]=\frac{f[x_1]-f[x_0]}{x_1x_0}$		
x_1	$f[x_1]$		$f[x_0,x_1,x_2]=\frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}$	
		$f[x_1,x_2]=\frac{f[x_2]-f[x_1]}{x_2x_1}$		$f[x_0,x_1,x_2,x_3]=\frac{f[x_1x_2,x_3]-f[x_0,x_2]}{x_3x_0}$
x_2	$f[x_2]$		$f[x_1,x_2,x_3]=\frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1}$	

$$f[x_2, x_3] = \frac{f[x_3] - f[x_2]}{x_3 - x_2}$$

$$f[x_1, x_2, x_3, x_4] = \frac{f[x_2, x_3, x_4] - f[x_1, x_2, x_3]}{x_4 - x_1}$$

$x_3 \quad f[x_3]$

$$f[x_2, x_3, x_4] = \frac{f[x_3, x_4] - f[x_2, x_3]}{x_4 - x_2}$$

$$f[x_3, x_4] = \frac{f[x_4] - f[x_3]}{x_4 - x_3}$$

$$f[x_2, x_3, x_4, x_5] = \frac{f[x_3, x_4, x_5] - f[x_2, x_3, x_4]}{x_5 - x_2}$$

$x_4 \quad f[x_4]$

$$f[x_3, x_4, x_5] = \frac{f[x_4, x_5] - f[x_3, x_4]}{x_5 - x_3}$$

$$f[x_4, x_5] = \frac{f[x_3] - f[x_4]}{x_5 - x_4}$$

$x_5 \quad f[x_5]$

Example 1. Complete the divided difference table for the data used in Table 3.10 and construct the interpolating polynomial that uses all these data.

Table 3.10

x	$f(x)$

$$g(x) = f(x) - P_n(x)$$

Since $f(x_i) = P_n(x_i)$ for each $i = 0,1, \ldots, n$, the function g has $n + 1$ distinct zeros in $[a, b]$. Generalized Rolle's Theorem implies that a number ξ in (a, b) exists with $g^{(n)}(\xi) = 0$, so

$$0 = f^{(n)}(\xi) - P_n^{(n)}(\xi)$$

Since $P_n(x)$is a polynomial of degree n whose leading coefficient is $f[x_0, x_1, \ldots, x_n]$,

$$P_n^{(n)}(x) = n!\, f[x_0, x_1, \ldots, x_n],$$

For all values of x. As a consequence,

$$f[x_0, x_1, \ldots, x_n] = \frac{f^{(n)}(\xi)}{n!}.$$

Newton's divided-difference formula can be expressed in a simplified form when the nodes are arranged consecutively with equal spacing. In this case, we introduce the notation $h = x_{i+1} - x_i$, for each $i = 0, 1, \ldots, n-1$ and let $x = x_0 + sh$. Then the difference $x - x_i$ is $x - x_i = (s - i)h$. So, Eq. (10) becomes

$$\begin{aligned} P_n^{(n)}(x) &= P_n(x_0 + sh) \\ &= f[x_0] + shf\,[x_0, x_1] + s(s-1)h^2 f[x_0, x_1, x_2] \\ &\quad + \cdots + s(s-1)\ldots(s-n+1)h^n f[x_0, x_1, \ldots, x_n] \\ &= f[x_0] + \sum_{k=1}^{n} \frac{s(s-1)\ldots(s-k+1)}{k!}, \end{aligned}$$

we can express $P_n(x)$ compactly as

$$\begin{aligned} P_n(x) &= P_n(x_0 + sh) \\ &= f[x_0] \\ &\quad + \sum_{k=1}^{n} \binom{S}{k} k!\, h^k f[x_0, x_i, \ldots, x_k] \end{aligned} \qquad (11)$$

Forward Differences

The Newton forward-difference formula is constructed by making use of the forward difference notation Δ. With this notation.

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{1}{h}\left(f(x_1) - f(x_0)\right) = \frac{1}{h}\Delta f\ (x_0)$$

$$f[x_0, x_1, x_2] = \frac{1}{2h}\left[\frac{\Delta f(x_1) - \Delta f(x_0)}{h}\right] = \frac{1}{2h^2}\Delta^2 f(x_0)$$

and, in general,

$$f[x_0, x_1, \ldots, x_k] = \frac{1}{k!\, h^k}\Delta^k f(x_0)$$

Since $f[x_0, x_1, \ldots, x_k] = \frac{1}{k!h^k}\Delta^k f(x_0)$.

Since $f[x_0] = f(x_0)$ Eq. (11) has the following form.

Newton Forward-Difference Formula

$$P_n(x) = f(x_0) + \sum_{k=1}^{n} \binom{S}{k} \Delta^k f(x_0) \qquad (12)$$

Backward Differences

If the interpolating nodes are reordered from last to first as $x_n, x_{n-1}, \ldots, x_0$, we can write the interpolatory formula as

$$P_n(x) = f[x_n] + f[x_n, x_{n-1}](x - x_n) + f[x_n, x_{n-1}, x_{n-2}](x - x_{n-1})$$

$$+ \cdots + f[x_n, \ldots, x_0](x - x_n)(x - x_{n-1}) \ldots (x - x_1).$$

If, in addition, the nodes are equally spaced with $x = x_n + sh$ and $x = x_i + (s + n - i)h$, then

$$P_n(x) = P_n(x_n + sh)$$

$$= f[x_n] + shf[x_n, x_{n-1}] + s(s+1)h^2 f[x_n, x_{n-1}, x_{n-2}] + \cdots$$

$$+ s(s+1) \ldots (s+n-1)h^n f[x_n, \ldots, x_0]$$

This is used to derive a commonly applied formula known as the **Newton backward difference formula**. To discuss this formula, we need the following definition.

Definition 3.7. Given the sequence $\{p_n\}_{n=0}^{\infty}$, define the backward-difference ∇p_n (read nabla p_n) by

$$\nabla p_n = p_n - p_{n-1}, \text{for } n \geq 1$$

Higher powers are defined recursively by

$$\nabla^k p_n = \nabla(\nabla^{k-1} p_n), \text{for } k \geq 2.$$

Definition 3.7 implies that

$$f[x_n, x_{n-1}] = \frac{1}{h}\nabla f(x_n), f[x_n, x_{n-1}, x_{n-2}] = \frac{1}{2h^2}\nabla^2 f(x_n),$$

and, in general,

$$f[x_n, x_{n-1}, \dots, x_{n-k}] = \frac{1}{k!\,h^k}\nabla^k f(x_n)$$

Consequently,

$$P_n(x) = f[x_n] + s\nabla f(x_n) + \frac{s(s+1)}{2}\nabla^2 f(x_n) + \cdots + \frac{s(s+1)\dots(s+n-1)}{n!}\nabla^n f(x_n).$$

If we extend the binomial coefficient notation to include all real values of s by letting

$$\binom{-s}{k} = \frac{-s\,(-s-1)\dots(-s-k+1)}{k!} = (-1)^k \frac{s\,(s+1)\dots(s+k-1)}{k!}$$

Then

$$P_n(x) = f[x_n] + (-1)^1\binom{-s}{1}\nabla f(x_n) + (-1)^2\binom{-s}{2}\nabla^2 f(x_n) + \cdots + (-1)^n\binom{-s}{n}\nabla^n f(x_n)$$

This gives the following result.

Newton Backward-Difference Formula

Unit-2nd

10.3 POWER METHOD FOR APPROXIMATING EIGENVALUES

In Chapter 7 we saw that the eigenvalues of an $n \times n$ matrix A are obtained by solving its characteristic equation

$$\lambda^n + c_{n-1}\lambda^{n-1} + c_{n-2}\lambda^{n-2} + \cdots + c_0 = 0.$$

For large values of n, polynomial equations like this one are difficult and time-consuming to solve. Moreover, numerical techniques for approximating roots of polynomial equations of high degree are sensitive to rounding errors. In this section we look at an alternative method for approximating eigenvalues. As presented here, the method can be used only to find the eigenvalue of A that is largest in absolute value—we call this eigenvalue the **dominant eigenvalue** of A. Although this restriction may seem severe, dominant eigenvalues are of primary interest in many physical applications.

Definition of Dominant Eigenvalue and Dominant Eigenvector

Let $\lambda_1, \lambda_2, \ldots,$ and λ_n be the eigenvalues of an $n \times n$ matrix A. λ_1 is called the **dominant eigenvalue** of A if

$$|\lambda_1| > |\lambda_i|, \quad i = 2, \ldots, n.$$

The eigenvectors corresponding to λ_1 are called **dominant eigenvectors** of A.

Not every matrix has a dominant eigenvalue. For instance, the matrix

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

(with eigenvalues of $\lambda_1 = 1$ and $\lambda_2 = -1$) has no dominant eigenvalue. Similarly, the matrix

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(with eigenvalues of $\lambda_1 = 2$, $\lambda_2 = 2$, and $\lambda_3 = 1$) has no dominant eigenvalue.

EXAMPLE 1 *Finding a Dominant Eigenvalue*

Find the dominant eigenvalue and corresponding eigenvectors of the matrix

$$A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}.$$

Solution From Example 4 of Section 7.1 we know that the characteristic polynomial of A is $\lambda^2 + 3\lambda + 2 = (\lambda + 1)(\lambda + 2)$. Therefore the eigenvalues of A are $\lambda_1 = -1$ and $\lambda_2 = -2$, of which the dominant one is $\lambda_2 = -2$. From the same example we know that the dominant eigenvectors of A (those corresponding to $\lambda_2 = -2$) are of the form

$$\mathbf{x} = t\begin{bmatrix} 3 \\ 1 \end{bmatrix}, \quad t \neq 0.$$

The Power Method

Like the Jacobi and Gauss-Seidel methods, the power method for approximating eigenvalues is iterative. First we assume that the matrix A has a dominant eigenvalue with corresponding dominant eigenvectors. Then we choose an initial approximation $\mathbf{x}_0$ of one of the dominant eigen*vectors* of A. This initial approximation must be a *nonzero* vector in R^n. Finally we form the sequence given by

$$\begin{aligned}
\mathbf{x}_1 &= A\mathbf{x}_0 \\
\mathbf{x}_2 = A\mathbf{x}_1 &= A(A\mathbf{x}_0) = A^2\mathbf{x}_0 \\
\mathbf{x}_3 = A\mathbf{x}_2 &= A(A^2\mathbf{x}_0) = A^3\mathbf{x}_0 \\
&\vdots \\
\mathbf{x}_k = A\mathbf{x}_{k-1} &= A(A^{k-1}\mathbf{x}_0) = A^k\mathbf{x}_0.
\end{aligned}$$

For large powers of k, and by properly scaling this sequence, we will see that we obtain a good approximation of the dominant eigenvector of A. This procedure is illustrated in Example 2.

EXAMPLE 2 ***Approximating a Dominant Eigenvector by the Power Method***

Complete six iterations of the power method to approximate a dominant eigenvector of

$$A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}.$$

Solution We begin with an initial nonzero approximation of

$$\mathbf{x}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

We then obtain the following approximations.

Iteration	*Approximation*
$\mathbf{x}_1 = A\mathbf{x}_0 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -10 \\ -4 \end{bmatrix}$	$\rightarrow -4\begin{bmatrix} 2.50 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_2 = A\mathbf{x}_1 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} -10 \\ -4 \end{bmatrix} = \begin{bmatrix} 28 \\ 10 \end{bmatrix}$	$\rightarrow 10\begin{bmatrix} 2.80 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_3 = A\mathbf{x}_2 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} 28 \\ 10 \end{bmatrix} = \begin{bmatrix} -64 \\ -22 \end{bmatrix}$	$\rightarrow -22\begin{bmatrix} 2.91 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_4 = A\mathbf{x}_3 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} -64 \\ -22 \end{bmatrix} = \begin{bmatrix} 136 \\ 46 \end{bmatrix}$	$\rightarrow 46\begin{bmatrix} 2.96 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_5 = A\mathbf{x}_4 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} 136 \\ 46 \end{bmatrix} = \begin{bmatrix} -280 \\ -94 \end{bmatrix}$	$\rightarrow -94\begin{bmatrix} 2.98 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_6 = A\mathbf{x}_5 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} -280 \\ -94 \end{bmatrix} = \begin{bmatrix} 568 \\ 190 \end{bmatrix}$	$\rightarrow 190\begin{bmatrix} 2.99 \\ 1.00 \end{bmatrix}$

Note that the approximations in Example 2 appear to be approaching scalar multiples of

$$\begin{bmatrix} 3 \\ 1 \end{bmatrix},$$

which we know from Example 1 is a dominant eigenvector of the matrix

$$A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}.$$

In Example 2 the power method was used to approximate a dominant eigenvector of the matrix A. In that example we already knew that the dominant eigenvalue of A was $\lambda = -2$. For the sake of demonstration, however, let us assume that we do not know the dominant eigenvalue of A. The following theorem provides a formula for determining the eigenvalue corresponding to a given eigenvector. This theorem is credited to the English physicist John William Rayleigh (1842–1919).

Theorem 10.2

Determining an Eigenvalue from an Eigenvector

If $\mathbf{x}$ is an eigenvector of a matrix A, then its corresponding eigenvalue is given by

$$\lambda = \frac{A\mathbf{x} \cdot \mathbf{x}}{\mathbf{x} \cdot \mathbf{x}}.$$

This quotient is called the **Rayleigh quotient.**

Proof Since $\mathbf{x}$ is an eigenvector of A, we know that $A\mathbf{x} = \lambda\mathbf{x}$, and we can write

$$\frac{A\mathbf{x} \cdot \mathbf{x}}{\mathbf{x} \cdot \mathbf{x}} = \frac{\lambda\mathbf{x} \cdot \mathbf{x}}{\mathbf{x} \cdot \mathbf{x}} = \frac{\lambda(\mathbf{x} \cdot \mathbf{x})}{\mathbf{x} \cdot \mathbf{x}} = \lambda.$$

In cases for which the power method generates a good approximation of a dominant eigenvector, the Rayleigh quotient provides a correspondingly good approximation of the dominant eigenvalue. The use of the Rayleigh quotient is demonstrated in Example 3.

EXAMPLE 3 *Approximating a Dominant Eigenvalue*

Use the result of Example 2 to approximate the dominant eigenvalue of the matrix

$$A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}.$$

Solution After the sixth iteration of the power method in Example 2, we had obtained.

$$\mathbf{x}_6 = \begin{bmatrix} 568 \\ 190 \end{bmatrix} \approx 190 \begin{bmatrix} 2.99 \\ 1.00 \end{bmatrix}.$$

With $\mathbf{x} = (2.99, 1)$ as our approximation of a dominant eigenvector of A, we use the Rayleigh quotient to obtain an approximation of the dominant eigenvalue of A. First we compute the product $A\mathbf{x}$.

$$A\mathbf{x} = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix} \begin{bmatrix} 2.99 \\ 1.00 \end{bmatrix} = \begin{bmatrix} -6.02 \\ -2.01 \end{bmatrix}$$

Then, since

$$A\mathbf{x} \cdot \mathbf{x} = (-6.02)(2.99) + (-2.01)(1) \approx -20.0$$

and

$$\mathbf{x} \cdot \mathbf{x} = (2.99)(2.99) + (1)(1) \approx 9.94,$$

we compute the Rayleigh quotient to be

$$\lambda = \frac{A\mathbf{x} \cdot \mathbf{x}}{\mathbf{x} \cdot \mathbf{x}} \approx \frac{-20.0}{9.94} \approx -2.01,$$

which is a good approximation of the dominant eigenvalue $\lambda = -2$.

From Example 2 we can see that the power method tends to produce approximations with large entries. In practice it is best to "scale down" each approximation before proceeding to the next iteration. One way to accomplish this **scaling** is to determine the component of $A\mathbf{x}_i$ that has the largest absolute value and multiply the vector $A\mathbf{x}_i$ by the reciprocal of this component. The resulting vector will then have components whose absolute values are less than or equal to 1. (Other scaling techniques are possible. For examples, see Exercises 27 and 28.

EXAMPLE 4 ***The Power Method with Scaling***

Calculate seven iterations of the power method with *scaling* to approximate a dominant eigenvector of the matrix

$$A = \begin{bmatrix} 1 & 2 & 0 \\ -2 & 1 & 2 \\ 1 & 3 & 1 \end{bmatrix}.$$

Use $\mathbf{x}_0 = (1, 1, 1)$ as the initial approximation.

Solution One iteration of the power method produces

$$A\mathbf{x}_0 = \begin{bmatrix} 1 & 2 & 0 \\ -2 & 1 & 2 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 5 \end{bmatrix},$$

and by scaling we obtain the approximation

$$\mathbf{x}_1 = \tfrac{1}{5} \begin{bmatrix} 3 \\ 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 0.60 \\ 0.20 \\ 1.00 \end{bmatrix}.$$

A second iteration yields

$$A\mathbf{x}_1 = \begin{bmatrix} 1 & 2 & 0 \\ -2 & 1 & 2 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 0.60 \\ 0.20 \\ 1.00 \end{bmatrix} = \begin{bmatrix} 1.00 \\ 1.00 \\ 2.20 \end{bmatrix}$$

and

$$\mathbf{x}_2 = \frac{1}{2.20} \begin{bmatrix} 1.00 \\ 1.00 \\ 2.20 \end{bmatrix} = \begin{bmatrix} 0.45 \\ 0.45 \\ 1.00 \end{bmatrix}.$$

Continuing this process, we obtain the sequence of approximations shown in Table 10.6.

TABLE 10.6

$\mathbf{x}_0$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$
$\begin{bmatrix} 1.00 \\ 1.00 \\ 1.00 \end{bmatrix}$	$\begin{bmatrix} 0.60 \\ 0.20 \\ 1.00 \end{bmatrix}$	$\begin{bmatrix} 0.45 \\ 0.45 \\ 1.00 \end{bmatrix}$	$\begin{bmatrix} 0.48 \\ 0.55 \\ 1.00 \end{bmatrix}$	$\begin{bmatrix} 0.51 \\ 0.51 \\ 1.00 \end{bmatrix}$	$\begin{bmatrix} 0.50 \\ 0.49 \\ 1.00 \end{bmatrix}$	$\begin{bmatrix} 0.50 \\ 0.50 \\ 1.00 \end{bmatrix}$	$\begin{bmatrix} 0.50 \\ 0.50 \\ 1.00 \end{bmatrix}$

From Table 10.6 we approximate a dominant eigenvector of A to be

$$\mathbf{x} = \begin{bmatrix} 0.50 \\ 0.50 \\ 1.00 \end{bmatrix}.$$

Using the Rayleigh quotient, we approximate the dominant eigenvalue of A to be $\lambda = 3$. (For this example you can check that the approximations of $\mathbf{x}$ and λ are exact.)

REMARK: Note that the *scaling factors* used to obtain the vectors in Table 10.6,

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$
↓	↓	↓	↓	↓	↓	↓
5.00	2.20	2.82	3.13	3.02	2.99	3.00,

are approaching the dominant eigenvalue $\lambda = 3$.

In Example 4 the power method with scaling converges to a dominant eigenvector. The following theorem tells us that a sufficient condition for convergence of the power method is that the matrix A be diagonalizable (and have a dominant eigenvalue).

Theorem 10.3
Convergence of the Power Method

If A is an $n \times n$ diagonalizable matrix with a dominant eigenvalue, then there exists a nonzero vector $\mathbf{x}_0$ such that the sequence of vectors given by

$$A\mathbf{x}_0, \quad A^2\mathbf{x}_0, \quad A^3\mathbf{x}_0, \quad A^4\mathbf{x}_0, \quad \ldots, \quad A^k\mathbf{x}_0, \quad \ldots$$

approaches a multiple of the dominant eigenvector of A.

Proof Since A is diagonalizable, we know from Theorem 7.5 that it has n linearly independent eigenvectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ with corresponding eigenvalues of $\lambda_1, \lambda_2, \ldots, \lambda_n$. We assume that these eigenvalues are ordered so that λ_1 is the dominant eigenvalue (with a corresponding eigenvector of $\mathbf{x}_1$). Because the n eigenvectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ are linearly independent, they must form a basis for R^n. For the initial approximation $\mathbf{x}_0$, we choose a nonzero vector such that the linear combination

$$\mathbf{x}_0 = c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \cdots + c_n\mathbf{x}_n$$

has nonzero leading coefficients. (If $c_1 = 0$, the power method may not converge, and a different $\mathbf{x}_0$ must be used as the initial approximation. See Exercises 21 and 22.) Now, multiplying both sides of this equation by A produces

$$\begin{aligned} A\mathbf{x}_0 &= A(c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \cdots + c_n\mathbf{x}_n) \\ &= c_1(A\mathbf{x}_1) + c_2(A\mathbf{x}_2) + \cdots + c_n(A\mathbf{x}_n) \\ &= c_1(\lambda_1\mathbf{x}_1) + c_2(\lambda_2\mathbf{x}_2) + \cdots + c_n(\lambda_n\mathbf{x}_n). \end{aligned}$$

Repeated multiplication of both sides of this equation by A produces

$$A^k\mathbf{x}_0 = c_1(\lambda_1^k\mathbf{x}_1) + c_2(\lambda_2^k\mathbf{x}_2) + \cdots + c_n(\lambda_n^k\mathbf{x}_n),$$

which implies that

$$A^k\mathbf{x}_0 = \lambda_1^k\left[c_1\mathbf{x}_1 + c_2\left(\frac{\lambda_2}{\lambda_1}\right)^k\mathbf{x}_2 + \cdots + c_n\left(\frac{\lambda_n}{\lambda_1}\right)^k\mathbf{x}_n\right].$$

Now, from our original assumption that λ_1 is larger in absolute value than the other eigenvalues it follows that each of the fractions

$$\frac{\lambda_2}{\lambda_1}, \quad \frac{\lambda_3}{\lambda_1}, \quad \ldots, \quad \frac{\lambda_n}{\lambda_1}$$

is less than 1 in absolute value. Therefore each of the factors

$$\left(\frac{\lambda_2}{\lambda_1}\right)^k, \quad \left(\frac{\lambda_3}{\lambda_1}\right)^k, \quad \ldots, \quad \left(\frac{\lambda_n}{\lambda_1}\right)^k$$

must approach 0 as k approaches infinity. This implies that the approximation

$$A^k\mathbf{x}_0 \approx \lambda_1^k c_1\mathbf{x}_1, \quad c_1 \neq 0$$

improves as k increases. Since $\mathbf{x}_1$ is a dominant eigenvector, it follows that any scalar multiple of $\mathbf{x}_1$ is also a dominant eigenvector. Thus we have shown that $A^k\mathbf{x}_0$ approaches a multiple of the dominant eigenvector of A.

The proof of Theorem 10.3 provides some insight into the rate of convergence of the power method. That is, if the eigenvalues of A are ordered so that

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|,$$

then the power method will converge quickly if $|\lambda_2| / |\lambda_1|$ is small, and slowly if $|\lambda_2| / |\lambda_1|$ is close to 1. This principle is illustrated in Example 5.

EXAMPLE 5 ***The Rate of Convergence of the Power Method***

(a) The matrix

$$A = \begin{bmatrix} 4 & 5 \\ 6 & 5 \end{bmatrix}$$

has eigenvalues of $\lambda_1 = 10$ and $\lambda_2 = -1$. Thus the ratio $|\lambda_2| / |\lambda_1|$ is 0.1. For this matrix, only four iterations are required to obtain successive approximations that agree when rounded to three significant digits. (See Table 10.7.)

TABLE 10.7

$\mathbf{x}_0$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$
$\begin{bmatrix} 1.000 \\ 1.000 \end{bmatrix}$	$\begin{bmatrix} 0.818 \\ 1.000 \end{bmatrix}$	$\begin{bmatrix} 0.835 \\ 1.000 \end{bmatrix}$	$\begin{bmatrix} 0.833 \\ 1.000 \end{bmatrix}$	$\begin{bmatrix} 0.833 \\ 1.000 \end{bmatrix}$

(b) The matrix

$$A = \begin{bmatrix} -4 & 10 \\ 7 & 5 \end{bmatrix}$$

has eigenvalues of $\lambda_1 = 10$ and $\lambda_2 = -9$. For this matrix, the ratio $|\lambda_2| / |\lambda_1|$ is 0.9, and the power method does not produce successive approximations that agree to three significant digits until sixty-eight iterations have been performed, as shown in Table 10.8.

TABLE 10.8

$\mathbf{x}_0$	$\mathbf{x}_1$	$\mathbf{x}_2$		$\mathbf{x}_{66}$	$\mathbf{x}_{67}$	$\mathbf{x}_{68}$
$\begin{bmatrix} 1.000 \\ 1.000 \end{bmatrix}$	$\begin{bmatrix} 0.500 \\ 1.000 \end{bmatrix}$	$\begin{bmatrix} 0.941 \\ 1.000 \end{bmatrix}$	$\cdots$ $\cdots$	$\begin{bmatrix} 0.715 \\ 1.000 \end{bmatrix}$	$\begin{bmatrix} 0.714 \\ 1.000 \end{bmatrix}$	$\begin{bmatrix} 0.714 \\ 1.000 \end{bmatrix}$

In this section we have discussed the use of the power method to approximate the *dominant* eigenvalue of a matrix. This method can be modified to approximate other eigenvalues through use of a procedure called **deflation.** Moreover, the power method is only one of several techniques that can be used to approximate the eigenvalues of a matrix. Another popular method is called the **QR algorithm.**

This is the method used in most computer programs and calculators for finding eigenvalues and eigenvectors. The algorithm uses the QR–factorization of the matrix, as presented in Chapter 5. Discussions of the deflation method and the QR algorithm can be found in most texts on numerical methods.

SECTION 10.3 EXERCISES

In Exercises 1–6, use the techniques presented in Chapter 7 to find the eigenvalues of the given matrix A. If A has a dominant eigenvalue, find a corresponding dominant eigenvector.

1. $A = \begin{bmatrix} 2 & 1 \\ 0 & -4 \end{bmatrix}$

2. $A = \begin{bmatrix} -3 & 0 \\ 1 & 3 \end{bmatrix}$

3. $A = \begin{bmatrix} 1 & -5 \\ -3 & -1 \end{bmatrix}$

4. $A = \begin{bmatrix} 4 & -5 \\ 2 & -3 \end{bmatrix}$

5. $A = \begin{bmatrix} 2 & 3 & 1 \\ 0 & -1 & 2 \\ 0 & 0 & 3 \end{bmatrix}$

6. $A = \begin{bmatrix} -5 & 0 & 0 \\ 3 & 7 & 0 \\ 4 & -2 & 3 \end{bmatrix}$

In Exercises 7–10, use the Rayleigh quotient to compute the eigenvalue λ of A corresponding to the given eigenvector $\mathbf{x}$.

7. $A = \begin{bmatrix} 4 & -5 \\ 2 & -3 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$

8. $A = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} -3 \\ 1 \end{bmatrix}$

9. $A = \begin{bmatrix} 1 & 2 & -2 \\ -2 & 5 & -2 \\ -6 & 6 & -3 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix}$

10. $A = \begin{bmatrix} 3 & 2 & -3 \\ -3 & -4 & 9 \\ -1 & -2 & 5 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix}$

C In Exercises 11–14, use the power method with scaling to approximate a dominant eigenvector of the matrix A. Start with $\mathbf{x}_0 = (1, 1)$ and calculate five iterations. Then use $\mathbf{x}_5$ to approximate the dominant eigenvalue of A.

11. $A = \begin{bmatrix} 2 & 1 \\ 0 & -7 \end{bmatrix}$

12. $A = \begin{bmatrix} -1 & 0 \\ 1 & 6 \end{bmatrix}$

13. $A = \begin{bmatrix} 1 & -4 \\ -2 & 8 \end{bmatrix}$

14. $A = \begin{bmatrix} 6 & -3 \\ -2 & 1 \end{bmatrix}$

C In Exercises 15–18, use the power method with scaling to approximate a dominant eigenvector of the matrix A. Start with $\mathbf{x}_0 = (1, 1, 1)$ and calculate four iterations. Then use $\mathbf{x}_4$ to approximate the dominant eigenvalue of A.

15. $A = \begin{bmatrix} 3 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 2 & 8 \end{bmatrix}$

16. $A = \begin{bmatrix} 1 & 2 & 0 \\ 0 & -7 & 1 \\ 0 & 0 & 0 \end{bmatrix}$

17. $A = \begin{bmatrix} -1 & -6 & 0 \\ 2 & 7 & 0 \\ 1 & 2 & -1 \end{bmatrix}$

18. $A = \begin{bmatrix} 0 & 6 & 0 \\ 0 & -4 & 0 \\ 2 & 1 & 1 \end{bmatrix}$

C In Exercises 19 and 20, the given matrix A does not have a dominant eigenvalue. Apply the power method with scaling, starting with $\mathbf{x}_0 = (1, 1, 1)$, and observe the results of the first four iterations.

19. $A = \begin{bmatrix} 1 & 1 & 0 \\ 3 & -1 & 0 \\ 0 & 0 & -2 \end{bmatrix}$

20. $A = \begin{bmatrix} 1 & 2 & -2 \\ -2 & 5 & -2 \\ -6 & 6 & -3 \end{bmatrix}$

21. (a) Find the eigenvalues and corresponding eigenvectors of

$$A = \begin{bmatrix} 3 & -1 \\ -2 & 4 \end{bmatrix}.$$

(b) Calculate two iterations of the power method with scaling, starting with $\mathbf{x}_0 = (1, 1)$.

(c) Explain why the method does not seem to converge to a dominant eigenvector.

22. Repeat Exercise 21 using $\mathbf{x}_0 = (1, 1, 1)$, for the matrix

$$A = \begin{bmatrix} -3 & 0 & 2 \\ 0 & -1 & 0 \\ 0 & 1 & -2 \end{bmatrix}.$$

C 23. The matrix

$$A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}$$

has a dominant eigenvalue of $\lambda = -2$. Observe that $A\mathbf{x} = \lambda\mathbf{x}$ implies that

$$A^{-1}\mathbf{x} = \frac{1}{\lambda}\mathbf{x}.$$

Apply five iterations of the power method (with scaling) on A^{-1} to compute the eigenvalue of A with the smallest magnitude.

C 24. Repeat Exercise 23 for the matrix

$$A = \begin{bmatrix} 2 & 3 & 1 \\ 0 & -1 & 2 \\ 0 & 0 & 3 \end{bmatrix}.$$

25. (a) Compute the eigenvalues of

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \text{ and } B = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}.$$

(b) Apply four iterations of the power method with scaling to each matrix in part (a), starting with $\mathbf{x}_0 = (-1, 2)$.

(c) Compute the ratios λ_2/λ_1 for A and B. For which do you expect faster convergence?

26. Use the proof of Theorem 10.3 to show that

$$A(A^k\mathbf{x}_0) \approx \lambda_1(A^k\mathbf{x}_0)$$

for large values of k. That is, show that the scale factors obtained in the power method approach the dominant eigenvalue.

C In Exercises 27 and 28, apply four iterations of the power method (with scaling) to approximate the dominant eigenvalue of the given matrix. After each iteration, scale the approximation by dividing by its length so that the resulting approximation will be a unit vector.

27. $A = \begin{bmatrix} 5 & 6 \\ 4 & 3 \end{bmatrix}$

28. $A = \begin{bmatrix} 7 & -4 & 2 \\ 16 & -9 & 6 \\ 8 & -4 & 5 \end{bmatrix}$

10.4 APPLICATIONS OF NUMERICAL METHODS

Applications of Gaussian Elimination with Pivoting

In Section 2.5 we used least squares regression analysis to find *linear* mathematical models that best fit a set of n points in the plane. This procedure can be extended to cover polynomial models of any degree as follows.

Regression Analysis for Polynomials

The least squares regression polynomial of degree m for the points $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ is given by

$$y = a_m x^m + a_{m-1}x^{m-1} + \cdots + a_2x^2 + a_1x + a_0,$$

where the coefficients are determined by the following system of $m + 1$ linear equations.

$$\begin{aligned}
na_0 + (\Sigma x_i)a_1 + (\Sigma x_i^2)a_2 + \cdots + (\Sigma x_i^m)a_m &= \Sigma y_i \\
(\Sigma x_i)a_0 + (\Sigma x_i^2)a_1 + (\Sigma x_i^3)a_2 + \cdots + (\Sigma x_i^{m+1})a_m &= \Sigma x_i y_i \\
(\Sigma x_i^2)a_0 + (\Sigma x_i^3)a_1 + (\Sigma x_i^4)a_2 + \cdots + (\Sigma x_i^{m+2})a_m &= \Sigma x_i^2 y_i \\
&\vdots \\
(\Sigma x_i^m)a_0 + (\Sigma x_i^{m+1})a_1 + (\Sigma x_i^{m+2})a_2 + \cdots + (\Sigma x_i^{2m})a_m &= \Sigma x_i^m y_i
\end{aligned}$$

Note that if $m = 1$ this system of equations reduces to

$$\begin{aligned}
na_0 + (\Sigma x_i)a_1 &= \Sigma y_i \\
(\Sigma x_i)a_0 + (\Sigma x_i^2)a_1 &= \Sigma x_i y_i,
\end{aligned}$$

§9.3 Householder's method

In this section, we will use Householder's transformation to find a symmetric **tridiagonal** matrix T that is similar to a given symmetric matrix A.

In next section, we will discuss how to find all eigenvalues of a symmetric tridiagonal matrix.

Definition 9.16: Let $w \in \mathbf{R}^n$ with $w^t w = 1$. Then the $n \times n$ matrix

$$P = I - 2ww^t$$

is called a Householder transformation (or Householder matrix).

Theorem 9.17: If $P = I - 2ww^t$ is a householder's matrix, then P is symmetric and orthogonal. So $P^{-1} = P^t = P$.

Proof: The symmetry is from

$$P^t = (I - 2ww^t)^t = I^t - (2ww^t)^t = I - 2(w^t)^t(w)^t = I - 2ww^t = P.$$

The orthogonality is from

$$\begin{aligned} PP^t &= (I - 2ww^t)(I - 2ww^t) = I - 4ww^t + 4ww^t ww^t \\ &= I - 4ww^t + 4w(w^t w)w^t = I - 4ww^t + 4ww^t = I. \end{aligned}$$

Some basics about the Householder matrix:

(a) $Pw = -w$. For any x orthogonal to w (i.e., $w^t x = 0$), then $Px = x$. These imply that P has eigenvalues 1 (multiplicity $n-1$) and -1.

(b) For any two distinct vectors x and y in $\mathbf{R}^n$ with **same length**, the Householder matrix

$$P = I - 2ww^t, \qquad w = (x - y)/\|x - y\|_2$$

satisfies

$$Px = y.$$

Proof of part (b): Let's figure out how to prove it. If $Px = y$, then

$$x - 2\frac{x - y}{\|x - y\|_2}\frac{(x - y)^t}{\|x - y\|_2}x = y.$$

$$(x - y) - 2\frac{(x - y)(x - y)^t x}{\|x - y\|_2^2} = 0.$$

$$\frac{x - y}{\|x - y\|_2^2}\left(\|x - y\|_2^2 - 2(x - y)^t x\right) = 0.$$

$$\frac{x - y}{\|x - y\|_2^2}\left((x - y)^t(x - y) - 2(x - y)^t x\right) = 0.$$

$$\frac{x - y}{\|x - y\|_2^2}\left(x^t x - x^t y - y^t x + y^t y - 2x^t x + 2y^t x\right) = 0.$$

The last equation is true since $x^t y = y^t x$ and $x^t x = y^t y$. For the final proof, you write above statements backwards.

By (b) above, for any $x \in \mathbf{R}^n$, let $y = (\pm\|x\|_2, 0, \ldots, 0)^t \in \mathbf{R}_n$, then $\|x\|_2 = \|y\|_2$. There is a Householder matrix P such that $Px = y$. This is the key step of Householder method. You may pick "+" or "−" in y.

Ex. If $x = (3,0,4)^t$, choose $y = (\|x\|_2, 0, 0)^t = (5,0,0)^t$. Let $w = (x-y)/\|x-y\|_2$ and $P = I - 2ww^t$. Then $Px = y$. (You can verify it.)

Ex. If $x = (1,1,3,0,4)^t$, find a Householder matrix to change x to the form $y = (1,1,*,0,0)$.

Sol. To make $\|x\|_2 = \|y\|_2$, $y = (1,1,5,0,0)$. Choose $w = (x-y)/\|x-y\|_2$ and $P = I - 2ww^t$ as before. You can verify that $Px = y$.

Just to remind you, for a Householder matrix P, if $Px = y$, then $x^tP = y^t$ since P is symmetric.

We use the following example to explain the Householder method for changing a symmetric matrix to a **similar** triangular matrix.

Ex. Use Householder matrices to transform

$$A = \begin{bmatrix} 4 & 1 & -2 & 2 \\ 1 & 2 & 0 & 1 \\ -2 & 0 & 3 & -2 \\ 2 & 1 & -2 & -1 \end{bmatrix}$$

to a **similar symmetric** tridiagonal matrix.

Sol. Let $x = (4,1,-2,2)^t$. We want to change x to $y = (4,*,0,0)^4$. So, let $y = (4,3,0,0)^t$ and $w = (x-y)/\|x-y\|_2 = (0,-2,-2,2)^t/\sqrt{12} = (0,-1,-1,1)/\sqrt{3}$. Let

$$P_1 = I - 2ww^t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/3 & -2/3 & 2/3 \\ 0 & -2/3 & 1/3 & 2/3 \\ 0 & 2/3 & 2/3 & 1/3 \end{bmatrix}.$$

Then

$$P_1 \begin{bmatrix} 4 \\ 1 \\ -2 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 4,1,-2,2 \end{bmatrix} P_1 = \begin{bmatrix} 4,3,0,0 \end{bmatrix}$$

Then,

$$P_1AP_1 = \begin{bmatrix} 4 & 3 & 0 & 0 \\ 3 & 10/3 & -4/3 & -1 \\ 0 & -4/3 & -1 & -4/3 \\ 0 & -1 & -4/3 & 5/3 \end{bmatrix}.$$

Now, for $x = (3, 10/3, -4/3, -1)$, choose $y = (3, 10/3, 5/3, 0)$ and $w = (x-y)^t/\|x-y\|_2$. Then

$$P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -4/5 & -3/5 \\ 0 & 0 & -3/5 & 4/5 \end{bmatrix}.$$

and

$$P_2P_1AP_1P_2 = \begin{bmatrix} 4 & 3 & 0 & 0 \\ 3 & 10/3 & 5/3 & 0 \\ 0 & 5/3 & -33/25 & -68/75 \\ 0 & 0 & -68/75 & 149/75 \end{bmatrix} = T.$$

This matrix T is tridiagonal. Note that

$$T = P_2P_1AP_1P_2 = P_2^tP_1^tAP_1P_2 = (P_1P_2)^tAP_1P_2.$$

Hence, T and A are similar and have same eigenvalues.

You can easily generalize this method to deal with any symmetric matrices.

Remark: This method can NOT be applied to transforming a symmetric matrix to a **similar** diagonal matrix D. (which step doesn't work in the method?) (If can, the diagonal elements of D will be the eigenvalues of A!)

Definition: A matrix $H = (h_{ij})$ is called an upper Hessenberg if $h_{ij} = 0$ for all $i \geq j + 2$.

Remark: If we apply the method above to any matrix, the result will be an upper Hessenberg matrix. You will have a question in your homework.

Remark: Householder method can also be used for solving systems of linear equations. The method is much more stable than Gaussian elimination method. But this method takes more time to get the solution than Gaussian method does.

of the data points closest to x. This requires using the Newton backward-divided-difference formula with $s = -\frac{2}{3}$ and the divided differences in Table 3.12 that have a wavy underline (__). Notice that the fourth divided difference is used in both formulas:

$$P_4(2.0) = P_4\left(2.2 - \frac{2}{3}(0.3)\right)$$

$$= 0.1103623 - \frac{2}{3}(0.3)(-0.5715210)$$
$$- \frac{2}{3}\left(\frac{1}{3}\right)(0.3)^2(0.0118183)$$
$$- \frac{2}{3}\left(\frac{1}{3}\right)\left(\frac{4}{3}\right)(0.3)^3(0.0680685)$$
$$- \frac{2}{3}\left(\frac{1}{3}\right)\left(\frac{4}{3}\right)\left(\frac{7}{3}\right)(0.3)^4(0.0018251)$$

$$= 0.2238754$$

Centered Differences

The Newton forward- and backward-difference formulas are not appropriate for approximating $f(x)$ when x lies near the center of the table because neither will permit the highest-order difference to have x_0 close to x. A number of divided-difference formulas are available for this case, each of which has situations when it can be used to maximum advantage. These methods are known as centered-difference formulas. We will consider only one centered-difference formula, Stirling's method.

For the centered-difference formulas, we choose near the point being approximated and label the nodes directly below x_0 as

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} a_i f(x_i),$$

With error given by

$$E(f) = \frac{1}{(n+1)!}\int_a^b \prod_{i=0}^{n}(x - x_i) f^{(n+1)}(\xi(x))\, dx,$$

Let us consider formulas produced by using first and second Lagrange polynomials with equally spaced nodes. This gives the **Trapezoidal rule** and **Simpson's rule**

The Trapezoidal Rule

To derive a Trapezoidal rule for approximating $\int_a^b f(x)dx$, let $x_0 = a,\ x_1 = b,\ h = b - a$ and use the linear Lagrange polynomial:

$$P_1(x) = \frac{(x - x_1)}{(x_0 - x_1)} f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)} f(x_1)$$

Then

$$\int_a^b f(x)dx = \int_{x_0}^{x_1} \left[\frac{(x - x_1)}{(x_0 - x_1)} f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)} f(x_1)\right] dx$$

$$+ \frac{1}{2}\int_{x_0}^{x_1} f''(\xi(x))(x - x_0)(x - x_1)dx. \quad (7)$$

The product $(x-x_0)(x-x_1)$does not change sign on $[x_0, x_1]$, so the Weighted Mean Value Theorem for Integrals can be applied to the error term to give, for some ξ in $(x-x_0)$

$$\int_{x_0}^{x_1} f''(\xi(x))(x-x_0)(x-x_1)dx$$

$$= f''(\xi)\int_{x_0}^{x_1}(x-x_0)(x-x_1)dx.$$

$$= f''(\xi)\left[\frac{x^3}{3} - \frac{(x_1+x_0)}{2}x^2 + x_0x_1x\right]_{x_0}^{x_1}$$

$$= \frac{h^3}{6}f''(\xi)$$

Consequently Eq. (7) implies that

$$\int_a^b f(x)dx = \left[\frac{(x-x_1)^2}{2(x_0-x_1)}f(x_0) + \frac{(x-x_0)^2}{2(x_1-x_0)}f(x_1)\right]_{x_0}^{x_1}$$

$$-\frac{h^3}{12}f''(\xi)$$

$$= \frac{(x-x_0)}{2}[f(x_0)+f(x_1)] - \frac{h^3}{12}f''(\xi)$$

(Weighted Mean Value Theorem for Integrals) Suppose $f \in C[a,b]$, the Riemann integral of g exists on $[a,b]$, and $g(x)$ does not change sign on $[a,b]$. Then there exists a number c in (a,b) with $\int_a^b f(x)g(x)\,dx = f(c)\int_a^b g(x)\,dx$. When $g(x) = 1$, this is the usual Mean Value Theorem for Integrals. It gives the

average value of the function f over the interval $[a, b]$, $f(c) = \frac{1}{b-a}\int_a^b f(x)\,dx$.

Using the notation $h = x_1 - x_0$ gives the following rule:

Trapezoidal Rule:

$$\int_a^b f(x)dx = \frac{h}{2}[f(x_0) + f(x_1)] - \frac{h^3}{12}f''(\xi)$$

(8)

This is called the Trapezoidal rule because when f is a function with positive values, $\int_a^b f(x)dx$ is approximated by the area in a trapezoid, as shown in Figure 4.3.

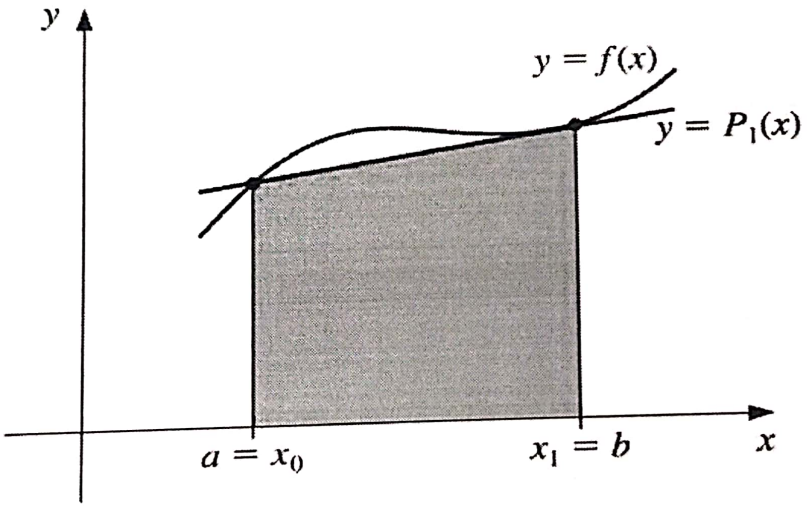


Figure 4.3.

The error term for the Trapezoidal rule involves f'', so the rule gives the exact result when applied to any function whose second derivative is identically zero, that is, any polynomial of degree one or less.

Simpson's Rule

Simpson's rule results from integrating over [a, b] the second Lagrange polynomial with equally spaced nodes $x_0 = a, x_2 = b,$, and $x_1 = a + h$, where $h = \frac{(b-a)}{2}$(See Figure 4.4.)

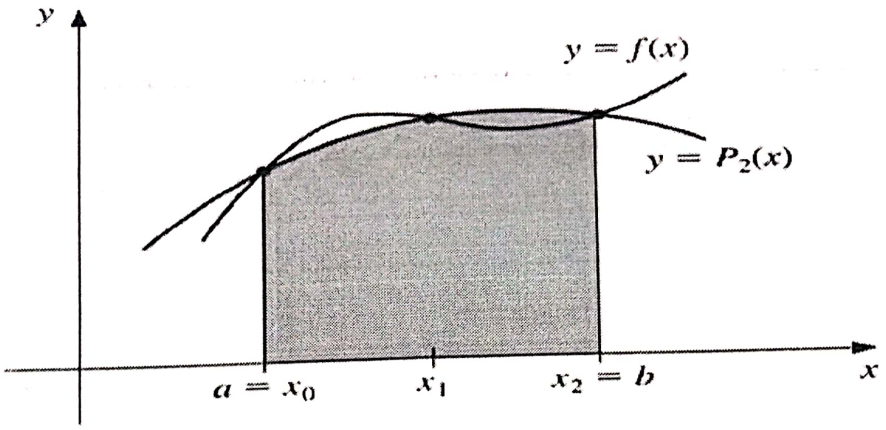


Figure 4.4

Therefore

$$\int_a^b f(x)dx = \int_{x_0}^{x_2} [\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}] dx + \int_{x_0}^{x_2} \frac{(x-x_0)(x-x_1)(x-x_2)}{6} f^{(3)}(\xi(x))dx$$

Deriving Simpson's rule in this manner, however, provides only an $O(h^4)$ error term involving$f^{(3)}$. By approaching the problem in another way, a higher-order term involving $f^{(4)}$ can be derived.

To illustrate this alternative method, suppose that f is expanded in the third Taylor polynomial about x_1. Then, for each x in $[x_0, x_2]$, a number $\xi(x)$ in (x_0, x_2) exists with

$$f(x) = f(x_1) + f'(x_1)(x - x_1) + \frac{f''(x_1)}{2}(x - x_1)^2 + \frac{f'''(x_1)}{6}(x - x_1)^3 + \frac{f^{(4)}(\xi(x))}{24}(x - x_1)^4$$

and

$$\int_{x_0}^{x_2} f(x)\,dx = \left[f(x_1)(x - x_1) + \frac{f'(x_1)}{2}(x - x_1)^2 + \frac{f''(x_1)}{6}(x - x_1)^3 + \frac{f'''(x_1)}{24}(x - x_1)^4 \right]_{x_0}^{x_2} + \frac{1}{24}\int_{x_0}^{x_2} f^{(4)}(\xi(x))(x - x_1)^4\,dx. \qquad (4.24)$$

Because $(x - x_1)^4$ is never negative on $[x_0, x_2]$, the Weighted Mean Value Theorem for Integral implies that

$$\frac{1}{24}\int_{x_0}^{x_2} f^{(4)}(\xi(x))(x - x_1)^4\,dx = \frac{f^{(4)}(\xi_1)}{24}\int_{x_0}^{x_2} (x - x_1)^4\,dx = \frac{f^{(4)}(\xi_1)}{120}(x - x_1)^5\Big]_{x_0}^{x_2},$$

For some number ξ_1 in (x_0, x_2).

However, $h = x_2 - x_1 = x_1 - x_0$, so

$$(x_2 - x_1)^2 - (x_0 - x_1)^2 = (x_2 - x_1)^4 - (x_0 - x_1)^4 = 0.$$

whereas

$(x_2 - x_1)^3 - (x_0 - x_1)^3 = 2h^3$ and $(x_2 - x_1)^5 - (x_0 - x_1)^5 = 2h^5$

Consequently,

$$\int_{x_0}^{x_2} f(x)dx = 2hf(x_1) + \frac{h^3}{3} f''(x_1) + \frac{f^{(4)}(\xi)}{60} h^5$$

If we replace $f''(x_1)$ by the approximation given Eq. (6) of section 4.1, we have

$$\int_{x_0}^{x_2} f(x)dx = 2hf(x_1) + \frac{h^3}{3}\left\{\frac{1}{h^2}[f(x_0) - 2f(x_1) + f(x_2)] - \frac{h^2}{12} f^{(4)}(\xi_2)\right\} + \frac{f^{(4)}(\xi_1)}{60} h^5$$

$$= \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{12}\left[\frac{1}{3} f^{(4)}(\xi_2) - \frac{1}{5} f^{(4)}(\xi_1)\right].$$

It can be shown by alternative methods that the values ξ_1 and ξ_2 in this expression can be replaced by a common value ξ in (x_0, x_2). This gives Simpson's rule.

Simpson's Rule:

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}[f(x_0) + 4f(x_2) + f(x_2)] - \frac{h^5}{90} f^{(4)}(\xi). \qquad (9)$$

The error term in Simpson's rule involves the fourth derivative of f, so it gives exact results when applied to any polynomial of degree three or less.

Example 1. Compare the Trapezoidal rule and Simpson's rule approximations to $\int_0^2 f(x)dx$ when $f(x)$ is

(a) x^2 (b) x^4 (c) $(x+1$

(d) $\sqrt{1+x^2}$ (e) $\sin x$ (f) e^x

Solution. On [0, 2], the Trapezoidal and Simpson's rules have the forms

Trapezoidal: $\int_0^2 f(x)dx \approx f(0) + f(2)$ and

Simpson's: $\int_0^2 f(x)dx \approx \frac{1}{3}[f(0) + 4f(1) + f(2)]$

When $f(x) = x^2$,

Trapezoidal: $\int_0^2 f(x)dx \approx 0^2 + 2^2 = 4$ and

Simpson's: $\int_0^2 f(x)dx \approx \frac{1}{3}[(0^2) + 4 \cdot 1^2 + 2^2] = \frac{8}{3}$

The approximation from Simpson's rule is exact because its truncation error involves $f^{(4)}$, which is identically 0 when f(x) = x^2.

The results to three places for the functions are summarized in Table 4.7. Notice that in each instance, Simpson's rule is significantly superior.

Table 4.7.

	(a)	(b)	(c)	(d)	(c)	(f)
$f(x)$	x^2	lx^2	$(x+1)^{-1}$	$\sqrt{1+x^2}$	sin x	ℓ^1
Exact value	2.667	6.400	1.099	2.958	1.416	6.389
Trapezoidal	4.000	16.000	1.333	3.326	0.909	8.389
Simpson's	2.667	6.667	1.111	2.964	1.425	6.421

Measuring Precision

The standard derivation of quadrature error formulas is based on determining the class of polynomials for which these formulas produce exact results. The next definition is used to facilitate the discussion of this derivation.

Definition. The **degree of accuracy**, or **precision**, of a quadrature formula is the largest positive integer n such that the formula is exact for x^k, for each ^k= 0,1, ... , n..

This implies that the Trapezoidal and Simpson's rules have degrees of precision one and three respectively.

Integration and summation are linear operations; that is,

$$\int_a^b (\alpha f(x) + \beta g(x))dx = \alpha \int_a^b f(x)dx + \beta \int_a^b g(x)dx$$

and

Unit - 3rd.

$$\sum_{i=0}^{n}(\alpha f(x_i) + \beta g(x_i)) = \alpha \sum_{i=0}^{n} f(x_i) + \beta \sum_{i=0}^{n} g(x_i),$$

for each pair of integrable functions f and g and each pair of real constants α and β. This implies that the degree of precision of a quadrature formula is n if and only if the error is zero for all polynomials of degree $k = 0,1, \ldots, n.$ but is not zero for some polynomial of degree n + 1.

The Trapezoidal and Simpson's rules are examples of a class of methods known as Newton- Cotes formulas. There are two types of Newton-Cotes formulas: open and closed.

Closed Newton-Cotes Formulas

The $(n + 1)$point closed Newton-Cotes formula uses node? $x_1 = x_0 + ih$,, for $i = 0,1, \ldots, n.$,where $x_0 = a$, $x_n = b$and $h = \frac{(b-a)}{n}$. (See Figure 4.5.) It is called closed because the endpoints of the closed interval [a, b] are included as nodes.

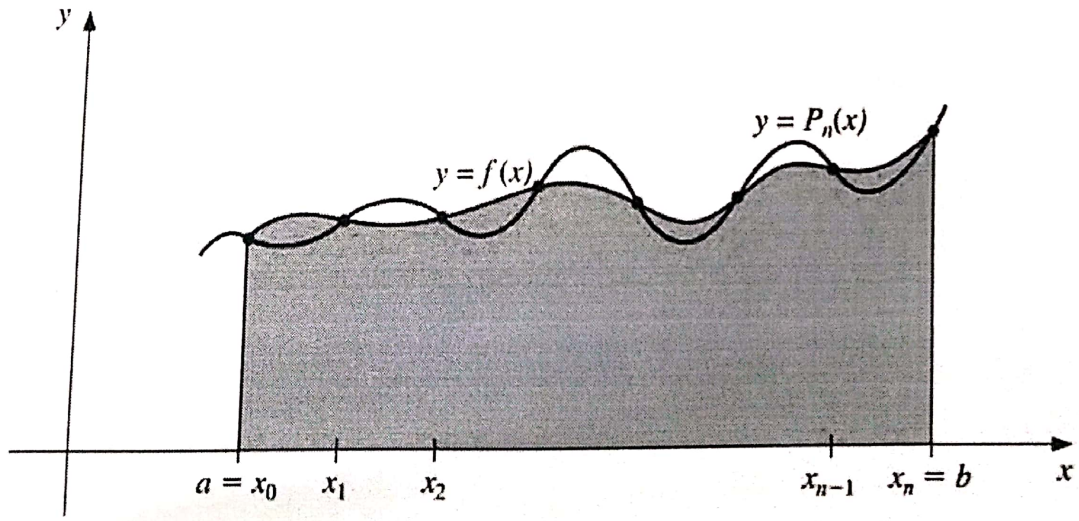


Figure 4.5

The formula assumes the form

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} a_i f(x_i),$$

where

$$a_i = \int_{x_0}^{x_n} L_i(x)dx = \int_{x_0}^{x_n} \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{(x-x_j)}{(x_i-x_j)} dx.$$

Theorem 4.2 Suppose that $\sum_{i=0}^{n} a_i f(x_i)$ denotes the $(n+1)$point closed Newton Cotes formula with $x_0 = a, x_n = b,$ and$h = \frac{(b-a)}{n}$. There exists $\xi \in (a,b)$for which

$$\int_a^b f(x)dx = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+3} f^{n+2}(\xi)}{(n+2)!} \int_o^n t^2(t-1)\dots(t-n)dt,$$

If n is even and $f \in C^{n+2}[a,b]$, and

$$\int_a^b f(x)dx = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!} \int_o^n t(t-1)\dots(t-n)dt,$$

If n is odd and $f \in C^{n+1}[a,b]$,

Note that when n is an even integer, the degree of precision is $n+1$, although the interpolation polynomial is of degree at most n. When n is odd, the degree of precision is only n.

Some of the common closed **Newton-Cotes formulas** with their error terms are listed. Note that in each case the unknown value ξ lies in (a, b).

n = 1: Trapezoidal rule

$$\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}[f(x_0) + f(x_1)] - \frac{h^3}{12}f''(\xi),$$

where $x_0 < \xi < x_1$.

(10)

n = 2: Simpson's rule

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{90}f^{(4)}(\xi),$$

where $x_0 < \xi < x_2$.

(11)

n = 3: Simpson's Three-Eighths rule

$$\int_{x_0}^{x_3} f(x)dx = \frac{3h}{8}[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] - \frac{3h^5}{80}f^{(4)}(\xi),$$

where $x_0 < \xi < x_3$.

(12)

n = 4:

$$\int_{x_0}^{x_4} f(x)dx = \frac{2h}{45}[7f(x_0) + 32f(x_1) + 12f(x_2) + 7f(x_4)] - \frac{8h^7}{945} f^{(6)}(\xi),$$

where $x_0 < \xi < x_4$.

(13)

Open Newton-Cotes Formulas

The open *Newton-Cotes formulas* do not include the endpoints of [a, b] as nodes. They use the nodes $x_i = x_0 + ih$, for each $i =$ 0,1, ... , n,, where $h = (b - a)/(n + 2)$ and $x_0 = a + h$. This implies that $x_n = b - h$, so we label the endpoints by setting x_{-1} $= a$ and $x_{n+1} = b$, as shown in Figure 4.6. Open formulas contain all the nodes used for the approximation within the open interval (a, b). The formulas become

$$\int_a^b f(x)dx = \int_{x_{-1}}^{x_{n+1}} f(x)dx \approx \sum_{i=0}^{n} a_i f(x_i).$$

where $a_i = \int_a^b L_i(x)dx$.

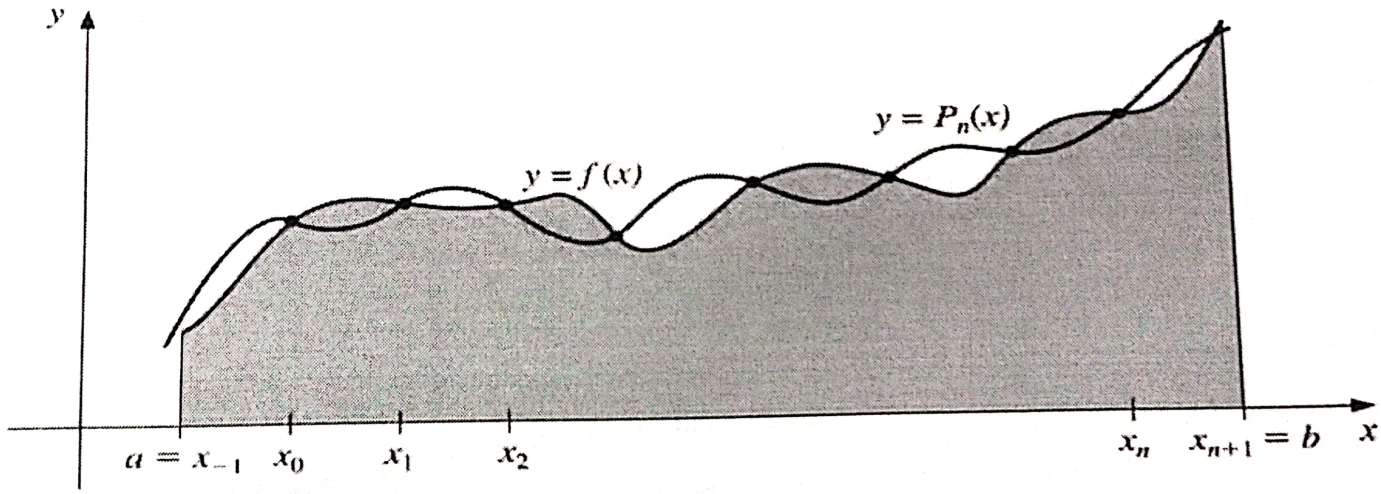


Figure 4.6

Theorem 4.3 Suppose that $\sum_{i=0}^{n} a_i f(x_i)$ denotes the (n + 1)-point open Newton-Cotes formula with $x_{-1} = a, x_{n+1} = b$, and $h = (b-a)/(n+2)$. There exists $\xi \in (a,b)$ for which

$$\int_a^b f(x)dx = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+3} \; f^{(n+2)} \; (\xi)}{(n+2)!} \int_{-1}^{n+1} t^2(t - 1) \dots (t-n)dt,$$

if n is even and $f \in C^{n+2} \; [a,b]$,and

$$\int_a^b f(x)dx = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+2} \; f^{(n+1)} \; (\xi)}{(n+1)!} \int_{-1}^{n+1} t^2(t - 1) \dots (t-n)dt,$$

if n is off and $f \in C^{n+1} \; [a,b]$.

Some of the common **open Newton-Cotes** formulas with their error terms are as follows:

n = 0: Midpoint rule

$$\int_{x_{-1}}^{x_1} f(x)dx = 2hf(x_0) + \frac{h^3}{3} f''(\xi),$$

where $x_{-1} < \xi < x_1$.

(14)

n = 1:

$$\int_{x_{-1}}^{x_2} f(x)dx = \frac{3h}{2}[f(x_0) + f(x_1)] + \frac{3h^3}{4}f''(\xi),$$

where $x_{-1} < \xi < x_2$.

(15)

n = 2:

$$\int_{x_{-1}}^{x_3} f(x)dx = \frac{4h}{3}[2f(x_0) - f(x_1) + 2f(x_2)] + \frac{14h^5}{45}f^{(4)}(\xi),$$

where $x_{-1} < \xi < x_3$.

(16)

n=3:

$$\int_{x_{-1}}^{x_4} f(x)dx = \frac{5h}{24}[11f(x_0) + f(x_1) + f(x_2) + 11f(x_3)] + \frac{95}{144}h^5 f^{(4)}(\xi),$$

where $x_{-1} < \xi < x_4$.

(17)

Example 2. Compare the results of the closed and open Newton-Cotes formulas listed as Eq. (10) through (13) and Eq. (14) through (17) to approximate

$$\int_0^{\frac{\pi}{4}} \sin x \, dx = 1 - \frac{\sqrt{2}}{2} \approx 0.29289322.$$

Solution. For the closed formulas, we have

$$n = 1: \quad \frac{\left(\frac{\pi}{4}\right)}{2}[\sin 0 + \sin \pi/4] \approx 0.27768018$$

$$n = 2: \quad \frac{\left(\frac{\pi}{8}\right)}{3}[\sin 0 + 4 \sin \frac{\pi}{8} \sin \pi/4] \approx 0.29293264$$

$$n = 3: \frac{3\left(\frac{\pi}{12}\right)}{8}[\sin 0 + 3 \sin \frac{\pi}{12} + 3 \sin \frac{\pi}{6} + \sin \pi/4] \approx 0.29291070$$

$$n = 4: \frac{2\left(\frac{\pi}{16}\right)}{45}[7 \sin 0 + 32 \sin \frac{\pi}{16} + 12 \sin \frac{\pi}{8} + 32 \sin \frac{3\pi}{16} + 7 \sin \frac{\pi}{4}] \approx 0.29289318$$

And for the open formulas, we have

$$n = 0: \quad 2\left(\frac{\pi}{8}\right)[\sin \frac{\pi}{8}] \approx 0.30055887$$

$$n = 1: \quad \frac{3\left(\frac{\pi}{12}\right)}{2}[\sin \frac{\pi}{12} + \sin \frac{\pi}{6}] \approx 0.29798754$$

$$n = 2: \quad \frac{4\left(\frac{\pi}{16}\right)}{3}[2 \sin \frac{\pi}{16} - \sin \frac{\pi}{8} + 2 \sin \frac{3\pi}{16}] \approx 0.29285866$$

$$n = 4: \quad \frac{5\left(\frac{\pi}{20}\right)}{24}[11 \sin \frac{\pi}{20} + \sin \frac{\pi}{10} + \sin \frac{3\pi}{20} + 11 \sin \frac{\pi}{5}] \approx 0.29286923$$

6.

a. The Composite Trapezoidal rule requires $h < 0.04382$ and $n > 46$. The approximation is 0.405471.

b. The Composite Simpson's rule requires $h < 0.44267$ and $n > 6$. The approximation is 0.405466.

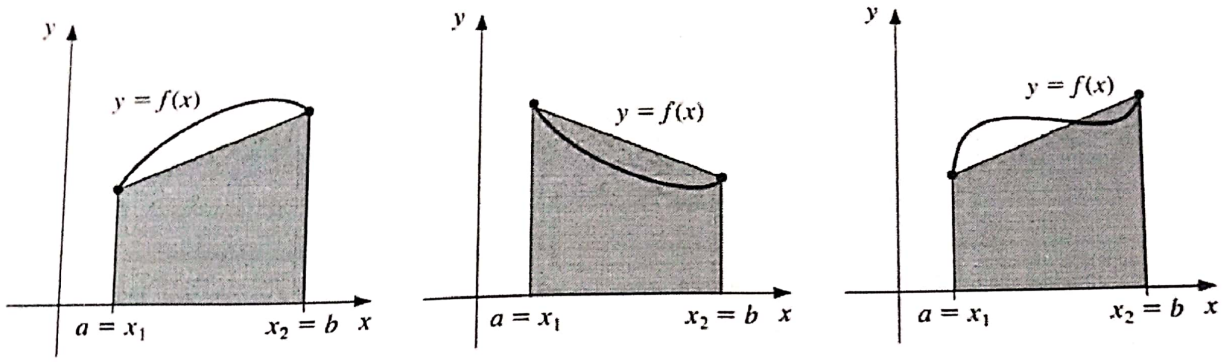
c. The Composite Midpoint rule requires $h < 0.03098$ and $n > 64$. The approximation is 0.405460

4.4 Gaussian Quadrature

The Newton-Cotes formulas in Section 4.3 were derived by integrating interpolating polynomials. The error term in the interpolating polynomial of degree n involves the $(n+1)$st derivative of the function being approximated, so a Newton-Cotes formula is exact when approximating the integral of any polynomial of degree less than or equal to n.

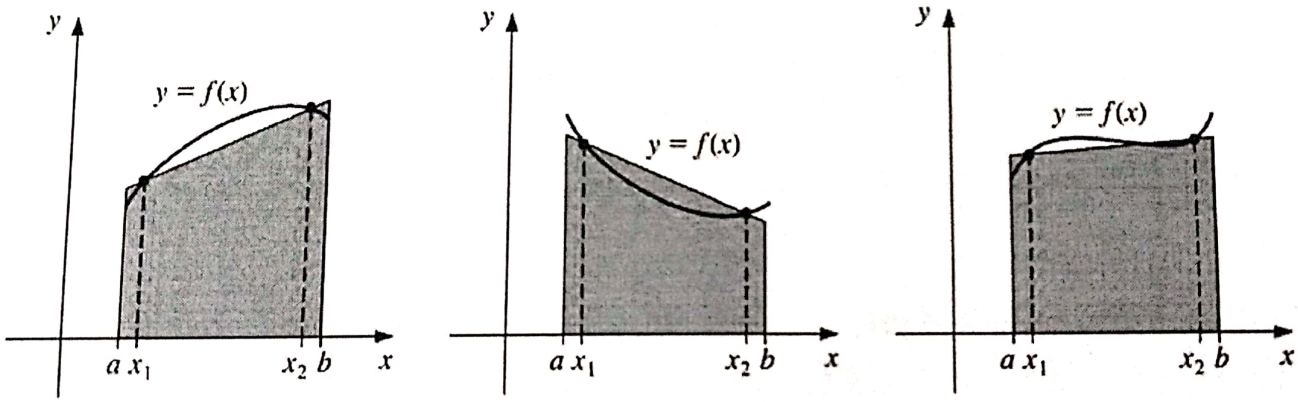
All the Newton-Cotes formulas use values of the function at equally spaced points. This restriction is convenient when the formulas are combined to form the composite rules, but it can significantly decrease the accuracy of the approximation. Consider, for example, the Trapezoidal rule applied to determine the integrals of the functions whose graphs are shown in Figure 4.10.

Figure 4.10



The Trapezoidal rule approximates the integral of the function by integrating the linear function that joins the endpoints of the graph of the function. But this is not likely the best line for approximating the integral. Lines such as those shown in Figure 4.11 would likely give much better approximations in most cases.

Figure 4.11



In Gaussian quadrature, the points for evaluation are chosen in an optimal rather than an equally spaced way. The nodes $x_1, x_2, \ldots, x_n$ in the interval [a, b] and coefficients $c_1, c_2, \ldots, c_n$ are chosen to minimize the expected error obtained in the approximation

d. The function f does not satisfy a Lipschitz condition, so Theorem 5.3 cannot be used

5.2 Euler's Method

Euler's method is the most elementary approximation technique for solving initial-value problems. Although it is seldom used in practice, the simplicity of its derivation can be used to illustrate the techniques involved in the construction of some of the more advanced

techniques, without the cumbersome algebra that accompanies these constructions. The object of Euler's method is to obtain approximations to the well-posed initial-value problem

$$\frac{dy}{dt} = f(t,y), \qquad a \le t \le b, \qquad y(a) = \alpha \quad (6)$$

A continuous approximation to the solution $y(t)$ will not be obtained; instead, approximations to y will be generated at various values, called **mesh points**, in the interval $[a,b]$. Once the approximate solution is obtained at the points, the approximate solution at other points in the interval can be found by interpolation.

We first make the stipulation that the mesh points are equally distributed throughout the interval $[a,b]$. This condition is ensured by choosing a positive integer N, setting $h = (b-a)/N$, and selecting the mesh points

$$t_i = a + ih, \qquad for\ each\ i = 0,1,2,\ldots,N.$$

The common distance between the points $h = t_{i+1} - t_i$, is called the step size.

We will use Taylor's Theorem to derive Euler's method. Suppose that $y(t)$, the unique solution to (6), has two continuous derivatives on $[a, b]$, so that for each $i = 0, 1, 2, \ldots, N - 1$

$$y(t_{i+1}) = y(t_i) + (t_{i+1} - t_i)y'(t_i) + \frac{(t_{i+1} - t_i)^2}{2} y''(\xi_i),$$

for some number ξ_i in (t_i, t_{i+1}). Because $h = t_{i+1} - t_i$, we have

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2} y''(\xi_i),$$

and, because $y(t)$ satisfies the differential equation (6),

$$\begin{aligned} & y(t_{i+1}) \\ &= y(t_i) + hf(t_i, y(t_i)) \\ &+ \frac{h^2}{2} y''(\xi_i) \end{aligned} \qquad (7)$$

Euler's method constructs $w_i \approx y(t_i)$, for each $i = 1, 2, .., N$, by deleting the remainder term. Thus, Euler's method is

$w_0 = \alpha$

$w_{i+1} = w_i + hf(t_i, w_i)$, for each $i = 0, 1, \ldots, N - 1$. (8)

Illustration

Use Euler's method to approximate the solution to

$$y' = y - t^2 + 1, \qquad 0 \le t \le 2, \qquad y(0) = 0.5,$$

at $t = 2$. Here we will simply illustrate the steps in the technique when we have $h = 0.5$. For this problem, $f(t, y) = y - t^2 + 1$; so,

Figure 5.2

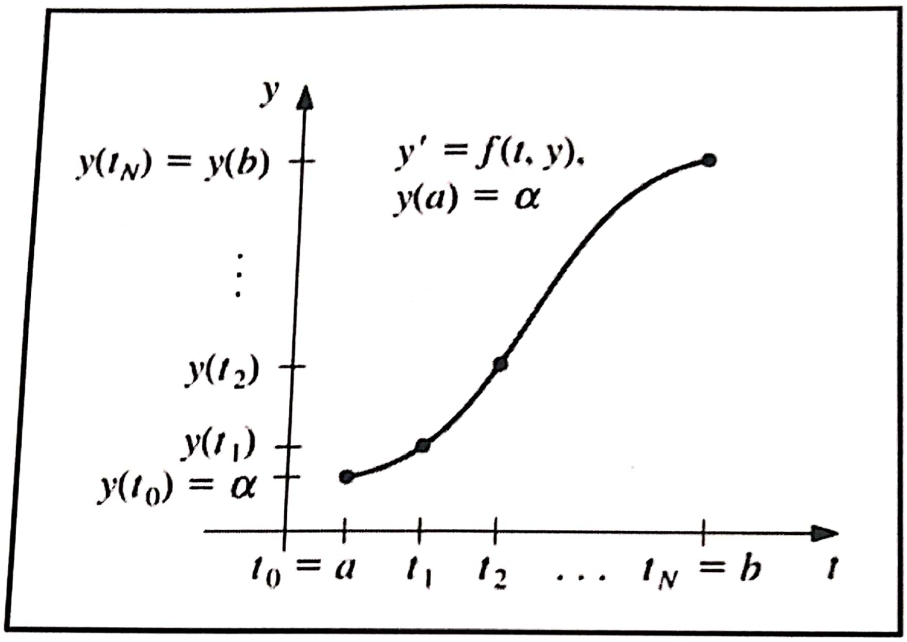


Figure 5.3

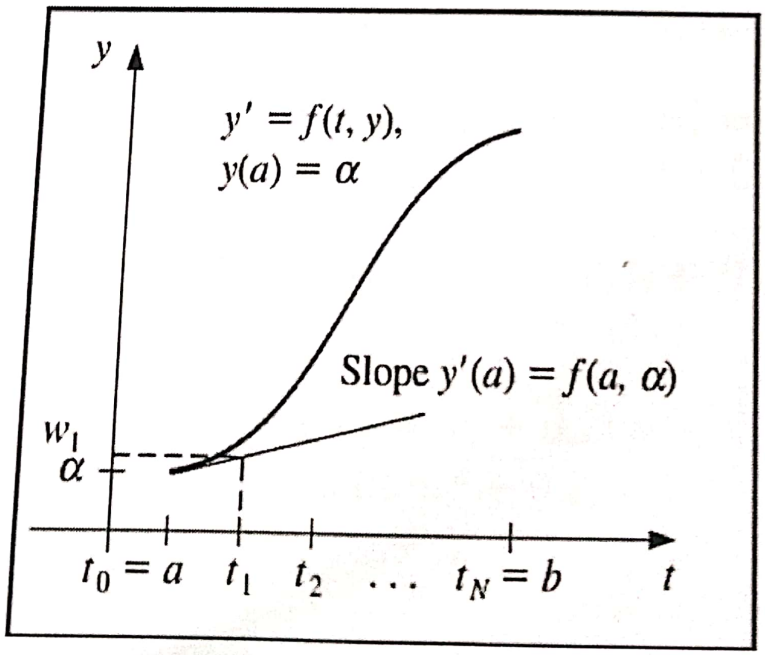
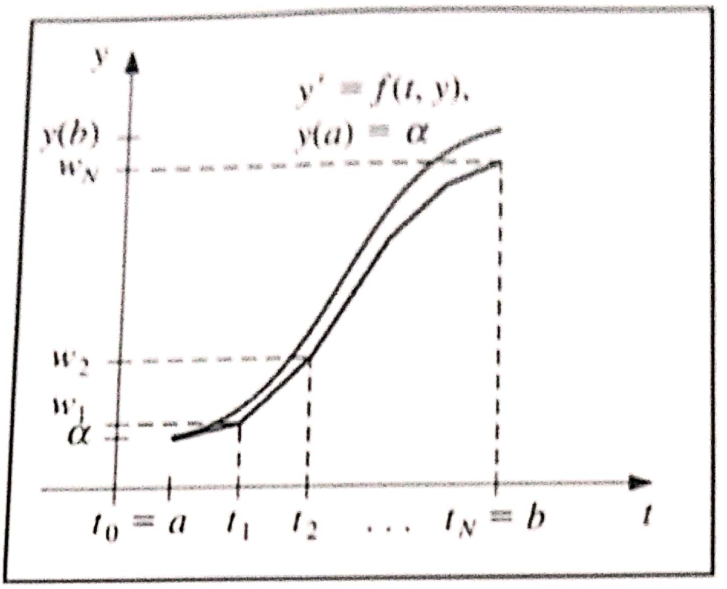


Figure 5.4



Example 1.

Euler's method was used in the first illustration with $h = 0.5$ to approximate the solution to the initial-value problem

$$y' = y - t^2 + 1, \qquad 0 \le t \le 2, \qquad y(0) = 0.5.$$

Use $N = 10$ to determine approximations and compare these with the exact values given by $y(t) = (t + 1)^2 - 0.5e^t$.

Solution. With $N = 10$, we have $h = 0.2, t_i = 0.2i, w_0 = 0.5$, and

$$w_{i+1} = w_i + h(w_i - t_i^2 + 1) = w_i + 0.2[w_i - 0.04i^2 + 1]$$
$$= 1.2w_i - 0.008i^2 + 0.2,$$

for $i = 0, 1, \ldots, 9$. So,

$$w_1 = 1.2(0.5) - 0.008(0)^2 + 0.2 = 0.8, w_2$$
$$= 1.2(0.8) - 0.008(1)^2 + 0.2 = 1.152,$$

15	1.5	2.152599
20	2.0	2.095185

5.4 Runge-Kutta method

The Taylor methods outlined in the previous section have the desirable property of high order local truncation error but the disadvantage of requiring the computation and evaluation of the derivatives of $f(t, y)$.

Runge-Kutta methods have the high-order local truncation error of the Taylor methods but eliminate the need to compute and evaluate the derivatives of $f(t, y)$

The difference-equation method resulting from replacing $T^{(2)}(t, y)$ in Taylor'smethod of order two by $f(t + (h/2), y + (h/2)f(t, y))$ is a specific Runge-Kutta method known as the **Midpoint method**.

Midpoint Method

$$w_0 = \alpha$$

$$w_{i+1} = w_i + hf\left(t_i + \frac{h}{2}, w_i + \frac{h}{2}f(t_i, w_i)\right), for\ i = 0,1, \ldots, N-1$$

Only three parameters are present in $a_1(f + \alpha_1, y + \beta_1)$, and all are needed in the match of $T^{(2)}$. So, a more complicated form is required to satisfy the conditions for any of the higher-order Taylor methods.

The most appropriate four-parameter form for approximating

1.8	4.8151763	4.8166575	0.0014812	4.8150236	0.0001527
2.0	5.3054720	5.3075838	0.0021119	5.3052587	0.0002132

Predictor-Corrector Methods

In Example 4, the implicit Adams-Moulton method gave better results than the explicit Adams-Bashforth method of the same order. Although this is generally the case, the implicit methods have the inherent weakness of first having to convert the method algebraically to an explicit representation for w_{i+1}. This procedure is not always possible, as can be seen by considering the elementary initial-value problem

$$y' = e^y, \quad 0 \le t \le 0.25, \quad y(0) = 1$$

Because $f(t, y) = e^y$, the three-step Adams-Moulton method has

$$w_{i+1} = w_i + \frac{h}{24}[9e^{w_{i+1}} + 19e^{w_i} - 5e^{w_{i-1}} + e^{w_{i-2}}]$$

as its difference equation, and this equation cannot be algebraically solved for w_{i+1}.

We could use Newton's method or the secant method to approximate w_{i+1}, but this complicates the procedure considerably. In practice, implicit multistep methods are not used as described above. Rather, they are used to improve approximations obtained by explicit methods. The combination