ADVANCED NUMERICAL ANALYSIS

Initial and Boundary value problems

F.A. ABD EL- SALAM

Initial-Value Problems for Ordinary Differential Equations

The Elementary Theory of Initial-Value Problems

Differential equations are used to model problems in science and engineering that involve the change of some variable with respect to another. Most of these problems require the solution of an *initial-value problem*, that is, the solution to a differential equation that satisfies a given initial condition.

In common real-life situations, the differential equation that models the problem is too complicated to solve exactly, and one of two approaches is taken to approximate the solution. The first approach is to modify the problem by simplifying the differential equation to one that can be solved exactly and then use the solution of the simplified equation to approximate the solution to the original problem. The other approach, which we will examine in this chapter, uses methods for approximating the solution of the original problem. This is the approach that is most commonly taken because the approximation methods give more accurate results and realistic error information.

The methods that we consider in this chapter do not produce a continuous approximation to the solution of the initial-value problem. Rather, approximations are found at certain specified, and often equally spaced, points. Some method of interpolation, commonly Hermite, is used if intermediate values are needed.

We need some definitions and results from the theory of ordinary differential equations before considering methods for approximating the solutions to initial-value problems.

Definition (1):

A function $f\ (x, y)$is said to satisfy a **Lipschitz condition** in the variable y on a set $D\ \subset \mathbb{R}^2$ if a constant $L > 0$ exists with

$|f\ (x, y_1)\ -\ f\ (x, y_2)|\ \leq\ L|\ y_1\ -\ y_2|$, Whenever $(x, y_1)$and $(x, y_2)$are in D. The constant L is called a **Lipschitz constant** for f.

Example 1

Show that $f\ (x, \text{y}) = x|y|$ satisfies a Lipschitz condition on the interval

$$D\ =\ \{(x, y)\ |\ 1\ \leq\ x\ \leq 2 \text{ and } -\ 3\ \leq\ y\ \leq\ 4\}$$

Solution

For each pair of points $(x, y_1)$and (x, y_2) in D we have

$$|f\ (x, y_1) -\ f\ (x, y_2)| = \big|x|y_1| - x|y_2|\big| = |x|\big||y_1| - |y_2|\big| \leq 2\ |y_1 - y_2|$$

Thus f satisfies a Lipschitz condition on D in the variable y with Lipschitz constant 2. The smallest value possible for the Lipschitz constant for this problem is $L = 2$, because, for example,

$|f\,(2, 1) - f\,(2, 0)| = |2 - 0| = 2|1 - 0|$.

Definition (2):

A set $D\ \subset \mathbb{R}^2$ is said to be **convex** if whenever (x_1, y_1) and(x_2, y_2) belong to D, then $((1 - \lambda)x_1 + \lambda x_2, (1 - \lambda)y_1 + \lambda y_2)$also belongs to D for every λ in [0, 1].

In geometric terms, Definition (2) states that a set is convex provided that whenever two points belong to the set, the entire straight-line segment between the points also belongs to the set. The sets we consider in this chapter are generally of the form $D = \{(t, y) \mid a\ \ x \leq\ b \text{ and } -\infty < y\ \ \infty\}$ for some constants a and b. It is easy to verify that these sets are convex.

Theorem (1):
Suppose $f\ (x, y)$ is defined on a convex set $D\ \subset \mathbb{R}^2$. If a constant $L > 0$ exists with

$$\frac{\partial f(x, y)}{\partial x} \leq L \; for\ all\ (x, y) \in D$$

Then *f* satisfies a Lipschitz condition on *D* in the variable *y* with Lipschitz constant *L*.

As the next theorem will show, it is often of significant interest to determine whether the function involved in an initial-value problem satisfies a Lipschitz condition in its second variable, and condition (1) is generally easier to apply than the definition. We should note, however, that Theorem (1) gives only sufficient conditions for a Lipschitz condition to hold. The function in Example 1, for instance, satisfies a Lipschitz condition, but the partial derivative with respect to *y* does not exist when *y* = 0.
The following theorem is a version of the fundamental existence and uniqueness theorem for first-order ordinary differential equations. Although the theorem can be proved with the hypothesis reduced somewhat, this form of the theorem is sufficient for our purposes.

Theorem (2):
Suppose that $D = \{(x, y) \mid a \leq x \leq\ b \text{ and } -\infty\ < y\ < \infty\}$ and that $f(x, y)$ is continuous on *D*. If *f* satisfies a Lipschitz condition on *D* in the variable *y*, then the initial-value problem
$y' = f(x, y), a \leq x \leq b, y(a) = \alpha$
has a unique solution $y(x)$ for $a \leq x \leq b$.

Example 2
Use Theorem (2) to show that there is a unique solution to the initial-value problem
$y' = 1 + x\ sinxy, 0 \leq x \leq 2, y(0) = 0$
Solution
Holding *x* constant and applying the Mean Value Theorem to the function
$f(x, y) = 1 + x\ sinxy$
We find that when $y_1\ <\ y_2$, a number ξ in (y_1, y_2), exists with

$$\frac{f(x, y_1) - f(x, y_2)}{y_1 - y_2} = \frac{\partial f(x, \xi)}{\partial y} = x^2 cos\xi x$$

Thus

$$|y_1 - y_2| = |y_1 - y_2||x^2 cos\xi x| \leq 4|y_1 - y_2|$$

and *f* satisfies a Lipschitz condition in the variable *y* with Lipschitz constant *L* = 4.
Additionally, $f(x, y)$ is continuous when $0 \leq x \leq 2$ and $-\infty < y < \infty$, so Theorem (2) implies that a unique solution exists to this initial-value problem.

EXERCISE (1)

(1) Use theorem (2) to show that each of the following initial-value problems has a unique solution, and find the solution.

(a) $y = y\cos x, \quad 0 \le x \le 1, \quad y(0) = 1.$

(b) $y' = \dfrac{2y}{x} + x^2 e^x,\ 1 \le x \le 2, \quad y(1) = 0.$

(c) $y' = -\dfrac{2y}{x} + x^2 e^x,\ 1 \le x \le 2, \quad y(1) = \sqrt{2e}$

(d) $y' = \dfrac{4x^3 y}{1 + x^4},\ 0 \le x \le 1, \quad y(0) = 1.$

(2) *Picard's method* for solving the initial-value problem

$$y' = f(x, y), a \le x \le b, y(a) = \alpha,$$

is described as follows:

Let $y_0(x) = \alpha$ for each x in $[a, b]$.

Define a sequence $\{y_k(x)\}$ of functions by

$$y_k(x) = \alpha + \int_{\alpha}^{x} f(\tau, y_{k-1}(\tau))\, d\tau \ , k = 1, 2, \ldots.$$

(a) Integrate $y' = f(x, y)$, and use the initial condition to derive Picard's method.

(b) Generate $y_0(x), y_1(x), y_2(x) and\ y_3(x)$ for the initial-value problem $y' = -y + x + 1, 0 \le x \le 1, y(0) = 1.$

(c) Compare the result in part (b) to the Maclaurin series of the actual solution $y(x) = x + e^{-x}$.

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

$$y' = f(x, y), a \le x \le b, y(a) = \alpha \qquad (1)$$

A continuous approximation to the solution $y(x)$ 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 and selecting the mesh points

$$x_k = a + kh, \quad \text{for each } k = 0, 1, 2, 3, \dots, N$$

The common distance between the points $h = \frac{b-a}{N} = x_{k+1} - x_k$ is called the **step size**.

We will use Taylor's Theorem to derive Euler's method. Suppose that $y(x)$, the unique solution to (1), has two continuous derivatives on $[a, b]$, so that for each $k = 0,1,2,3,\dots,N-1$

$$y(x_{k+1}) = y(x_k) + (x_{k+1} - x_k)y'(x_k) + \frac{(x_{k+1} - x_k)^2}{2} y''(\xi_k) \qquad (2)$$

For som number $\xi_k \in (x_k, x_{k+1})$ becouse $h = x_{k+1} - x_k$, we have

$$y(x_{k+1}) = y(x_k) + h\,y'(x_k) + \frac{h^2}{2} y''(\xi_k) \qquad (3)$$

and, because $y(x)$ satisfies the differential equation (1),

$$y(x_{k+1}) = y(x_k) + hf(x_k, y_k) + \frac{h^2}{2} y''(\xi_k) \qquad (4)$$

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

$w_0 = \alpha,$

$$w_{k+1} = w_k + h\, f(x_k, w_k), \text{for each } k=0,1,\dots, N-1. \qquad (5)$$

Illustration

Euler's method to approximate the solution to

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

at x = 2. Here we will simply illustrate the steps in the technique when we have h = 0.5.

For this problem $f(x, y) = y - x^2 + 1$, so

$w_{k+1} = w_k + hf(x_k, w_k)$, for each $k = 0,1, \dots, N-1$.

$w_{k+1} = w_k + h(w_k - x_k^2 + 1)$, for each $k = 0,1, \dots, N-1$.

$w_0 = y(0) = 0.5;$

$w_{k+1} = w_k + 0.5(w_k - x_k^2 + 1)$, for each $k = 0,1, \dots, N-1$.

$$w_1 = y(x_1) = w(0.5) = y_0 + 0.5\left(w_0 - (0.0)^2 + 1\right) = 0.5 + 0.5(1.5) = 1.25;$$

$$w_2 = y(x_2) = w(1.0) = w_1 + 0.5\left(w_1 - (0.5)^2 + 1\right) = 1.25 + 0.5(2.0) = 2.25;$$

$$w_3 = y(x_3) = y(1.5) = w_2 + 0.5\left(w_2 - (1.0)^2 + 1\right) = 2.25 + 0.5(2.25) = 3.375;$$

$$w_4 = y(x_4) = y(2) = w_3 + 0.5\left(w_3 - (1.5)^2 + 1\right) = 3.375 + 0.5(2.125) = 4.4375.$$

Equation (5) is called the **difference equation** associated with Euler's method. As we will see later in this chapter, the theory and solution of difference equations parallel, in many ways, the theory and solution of differential equations.

Example 3

Euler's method was used in the first illustration with $h = 0.2$ to approximate the solution to the initial-value problem

$$y' = y - x^2 + 1, 0 \leq t \leq 2, \quad y(0) = 0.5,$$

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

Solution

With $N = 10$ we have $h = 0.2$, $x_k = 0.2k, y_0 = 0.5$, and

$$w_{k+1} = w_k + h(w_k - x_k{}^2 + 1)$$

$$= w_k + 0.2h(w_k - (0.2k)^2 + 1)$$

$$= 1.2w_k - 0.008k^2 + 0.2$$

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

$$w_1 = 1.2(0.5) - 0.008(0)^2 + 0.2 = 0.8$$

$$w_2 = 1.2(0.8) - 0.008(1)^2 + 0.2 = 1.152$$

and so on. Table (1) shows the comparison between the approximate values at x_k and the actual values

k	x_k	$y_k = y(x_k)$	$\lvert y_{exact} - y_k \rvert$
0	0	0.5	0
1	0.2	0.8	0.029298621
2	0.4	1.152	0.062087651
3	0.6	1.5504	0.0985406
4	0.8	1.98848	0.138749536
5	1	2.458176	0.182683086
6	1.2	2.9498112	0.230130339
7	1.4	3.45177344	0.280626577
8	1.6	3.950128128	0.33335566
9	1.8	4.428153754	0.387022514
10	2	4.865784504	0.439687446

Table (1)

Note that the error grows slightly as the value of t increases. This controlled error growth is a consequence of the stability of Euler's method, which implies that the error is expected to grow in no worse than a linear manner

Error Bounds for Euler's Method

Although Euler's method is not accurate enough to warrant its use in practice, it is sufficiently elementary to analyze the error that is produced from its application.

Theorem (3)
Suppose f is continuous and satisfies a Lipschitz condition with constant L on $D = \{(x, y) \mid a \le x \le b \; and \; -\infty < y < \infty\}$ and that a constant M exists with $|y'| \le M, for \; all \; x \in [a, b]$, where $y(x)$ denotes the unique solution to the initial-value problem $y' = f(x, y), a \le x \le b, y(a) = \alpha$.
Let $w_0, w_1, \ldots, w_N$ be the approximations generated by Euler's method for some positive integer N. Then, for each $K = 0, 1, 2, \ldots, N$,

$$|y_k - w_k| \le \frac{hM}{2L}(e^{L(x_k - a)} - 1)$$

Proof
When k= 0 the result is clearly true, since $y(x_0) = w_0 = \alpha$.
From equation (4)

$$y(x_{k+1}) = y(x_k) + hf(x_k, y_k) + \frac{h^2}{2} y''(\xi_k)$$

for $i = 0, 1, \ldots, N - 1$, and from the equations in (5),
$w_{k+1} = w_k + hf(x_k, w_k)$
Using the notation $y_k = y(x_k) \; and \; y_{k+1} = y(x_{k+1})$, we subtract these two equations to obtain

$$y_{k+1} - w_{k+1} = (y_k - w_k) + h(f(x_k, y(x_k)) - f(x_k, w_k)) + \frac{h^2}{2} y''(\xi_k)$$

Hence

$$|y_{k+1} - w_{k+1}| \le |y_k - w_k| + h|f(x_k, y_k) - f(x_k, w_k)| + \frac{h^2}{2}|y''(\xi_k)|$$

Now f satisfies a Lipschitz condition in the second variable with constant L, and $|y'(t)| \le M$, so

$$|y_{k+1} - w_{k+1}| \le |y_k - w_k| + hL|y_k - w_k| + \frac{h^2}{2}|y''(\xi_k)|$$

$$|y_{k+1} - w_{k+1}| \le |y_k - w_k|(1 + hL) + \frac{h^2 M}{2}$$

$$\le |y_{k-1} - w_{k-1}|(1 + hL)^2 + \frac{h^2 M}{2}(1 + hL) + \frac{h^2 M}{2}$$

$$\le |y_{k-2} - w_{k-2}|(1 + hL)^3 + \frac{h^2 M}{2}(1 + hL)^2 + \frac{h^2 M}{2}(1 + hL) + \frac{h^2 M}{2}$$

$1 + (1 + hL) + (1 + hL)^2 + \ldots + (1 + hL)^k$ is a geometric series with ratio $(1 + hL)$

Its sum $\dfrac{1 - (1 + hL)^{k+1}}{1 - (1 + hL)}$ then

$$\left|y_{k+1}-w_{k+1}\right| \le \left|y_0-w_0\right|(1+hL)^{k+1}+\frac{h^2M}{2}\left(1+(1+hL)+(1+hL)^2+\ldots+(1+hL)^k\right)$$

$$\le \left|y_0-w_0\right|(1+hL)^{k+1}+\frac{h^2M}{2}\left(\frac{1-(1+hL)^{k+1}}{1-(1+hL)}\right)$$

Because $\left|y_0-w_0\right|=0$, $x_{k+1}-x_0=x_{k+1}-a$ and $(1+hL)^{k+1}\le e^{hL(k+1)}$, this implies that

$$\left|y_{k+1}-w_{k+1}\right| \le \frac{h^2M}{2hL}\left((1+hL)^{k+1}-1\right) \le \frac{h^2M}{2hL}\left(\mathrm{e}^{hL(k+1)}-1\right) \le \frac{h^2M}{2hL}\left(\mathrm{e}^{L(x_{k+1}-a)}-1\right)$$

for each $k = 0, 1, \ldots, N-1$.

The weakness of Theorem (3) lies in the requirement that a bound be known for the second derivative of the solution. Although this condition often prohibits us from obtaining a realistic error bound, it should be noted that if $\partial f/\partial t$ and $\partial f/\partial y$ both exist, the chain rule for partial differentiation implies that

$$y''(x) = dy'/dx = df/dx$$
$$= \partial f/\partial x(x, y(x)) + \partial f/\partial y(x, y(x)) \cdot f(x, y(x)).$$

So it is at times possible to obtain an error bound for $y'(t)$ without explicitly knowing $y(t)$.

Example 4

The solution to the initial-value problem

$$y' = y - x^2 + 1, 0 \le x \le 2, y(0) = 0.5,$$

was approximated in Example 3 using Euler's method with $h = 0.2$. Use the inequality in Theorem (3) to find a bounds for the approximation errors and compare these to the actual errors.

Solution

Since $f(x, y) = y - x^2 + 1$, we have $\partial f(x, y)/\partial y = 1$ for all y, so $L = 1$. For this problem, the exact solution is

$y(x) = (x + 1)^2 - 0.5e^x$,

so $y''(x) = 2 - 0.5e^x$ and

$|y''(t)| \le 0.5e^2 - 2$, for all $x \in [0, 2]$.

Using the inequality in the error bound for Euler's method with $h = 0.2$, $L = 1$, and $M = 0.5e^2 - 2$ gives

$$|y_k - w_k| \le 0.1(0.5e^2 - 2)(e^{x_k} - 1).$$

Hence

$$|y(0.2) - w_1| \le 0.1(0.5e^2 - 2)(e^{0.2} - 1) = 0.03752;$$
$$|y(0.4) - w_2| \le 0.1(0.5e^2 - 2)(e^{0.4} - 1) = 0.08334;$$

and so on. Table (2) lists the actual error found in Example 3, together with this error bound. Note that even though the true bound for the second derivative of the solution was used, the error bound is considerably larger than the actual error, especially for increasing values of *x*.

x_k	Approximate solution	exact solution	Actual Error	Error Bound
0.2	0.8	0.829298621	0.029298621	0.037517318
0.4	1.152	1.214087651	0.062087651	0.083341075
0.6	1.5504	1.6489406	0.0985406	0.139310337
0.8	1.98848	2.127229536	0.138749536	0.207671348
1	2.458176	2.640859086	0.182683086	0.291167676
1.2	2.949811	3.179941539	0.230130339	0.39315032
1.4	3.451773	3.732400017	0.280626577	0.517712204
1.6	3.950128	4.283483788	0.33335566	0.669852432
1.8	4.428154	4.815176268	0.387022514	0.855676927
2	4.865785	5.305471951	0.439687447	1.082643477

Table (2)

The principal importance of the error-bound formula given in Theorem (3) is that the bound depends linearly on the step size *h*. Consequently, diminishing the step size should give correspondingly greater accuracy to the approximations.
Neglected in the result of Theorem (3) is the effect that round-off error plays in the choice of step size. As *h* becomes smaller, more calculations are necessary and more round off error is expected. In actuality then, the difference-equation form

$w_0 = \alpha,$

$w_{k+1} = w_k + h\, f(x_k, w_k),$ for each $k=0,1,\ldots, N-1.$

is not used to calculate the approximation to the solution y_k at a mesh point x_k. We use instead an equation of the form

$w_0 = \alpha + \delta_0$

$u_{k+1} = u_k + h\, f(x_k, u_k) + \delta_{k+1},$ for each $k=0,1,\ldots, N-1.$

where δ_k denotes the round-off error associated with u_k. Using methods similar to those in the proof of Theorem (3), we can produce an error bound for the finite-digit approximations to y_k given by Euler's method.

EXERCISE (2)

(1) Use Euler's method to approximate the solutions for each of the following initial-value problems.

(a) $y' = xe^{3x} - 2y, \quad 0 \le x \le 1, \quad y(0) = 0, \; with \quad h = 0.5$

(b) $y' = 1 + (x - y)^2, \quad 2 \le x \le 3, \quad y(2) = 1,$ with $h = 0.5$

(c) $y' = 1 + \frac{y}{x}, \quad 1 \le x \le 2, \quad y(1) = 2,$ with $h = 0.25$

(d) $y' = \cos\ 2x + \sin\ 3x, \quad 0 \le x \le 1, \quad y(0) = 1,$ with $h = 0.25$

(2) The actual solutions to the initial-value problems in Exercise 1 are given here. Compare the actual error at each step to the error bound.

(a) $y(x) = \frac{1}{5}xe^{3x} - \frac{1}{25}e^{3x} + \frac{1}{25}e^{-2x}$

(b) $y(x) = x + \dfrac{1}{1-x}$

(c) $y(x) = x\ln x + 2x$

(d) $y(x) = \frac{1}{2}\sin 2x - \frac{1}{3}\cos 3x + \frac{4}{3}$

Higher-Order Taylor Methods

Since the object of a numerical techniques is to determine accurate approximations with minimal effort, we need a means for comparing the efficiency of various approximation methods. The first device we consider is called the *local truncation error* of the method.

The local truncation error at a specified step measures the amount by which the exact solution to the differential equation fails to satisfy the difference equation being used for the approximation at that step. This might seem like an unlikely way to compare the error of various methods. We really want to know how well the approximations generated by the methods satisfy the differential equation, not the other way around. However, we don't know the exact solution so we cannot generally determine this, and the local truncation will serve quite well to determine not only the local error of a method but the actual approximation error.

Definition (3)

Consider the initial value problem

$$y' = f(x, y), a \le x \le b, y(a) = \alpha.$$

The difference equation

$w_{k+1} = w_k + h\,f(x_k, w_k),\ w_0 = \alpha,$ for each $k = 0, 1, \ldots, N-1$.

has **local truncation error** τ_{i+1} where

$$\tau_{i+1} = \frac{y_{i+1} - (y_i + hf(x_i, y_i))}{h} = \frac{y_{i+1} - y_i}{h} - f(x_i, y_i), \quad \text{for each } i = 0, 1, \ldots, N-1$$

Where y_i and y_{i+1} denote the solution at x_i and x_{i+1}, respectively.
For example, Euler's method has local truncation error at the *i*th step

$$\tau_{i+1} = \frac{y_{i+1} - y_i}{h} - f(x_i, y_i),$$

for each $i = 0, 1, \ldots, N - 1$.
This error is a ***local error*** because it measures the accuracy of the method at a specific step, assuming that the method was exact at the previous step. As such, it depends on the differential equation, the step size, and the particular step in the approximation. By considering Eq. (4) in the previous section, we see that Euler's method has

$\tau_{i+1}(h) = \frac{h}{2} y''(\xi_i)$, for some ξ_i in (x_i, x_{i+1}).
When $y''(x)$ is known to be bounded by a constant M on $[a, b]$, this implies

$|\tau_{i+1}(h)| \leq \frac{h}{2} M$, so the local truncation error in Euler's method is $O(h)$.

One way to select difference-equation methods for solving ordinary differential equations is in such a manner that their local truncation errors are $O(h)^p$ for as large a value of p as possible, while keeping the number and complexity of calculations of the methods within a reasonable bound.
Since Euler's method was derived by using Taylor's Theorem with $n = 1$ to approximate the solution of the differential equation, our first attempt to find methods for improving the convergence properties of difference methods is to extend this technique of derivation to larger values of n.
Suppose the solution $y(x)$ to the initial-value problem

$$y' = f(x, y), a \leq x \leq b, y(a) = \alpha,$$

has $(n + 1)$ continuous derivatives. If we expand the solution, $y(x)$, in terms of its nth Taylor polynomial about x_i and evaluate at x_{i+1}, we obtain

$$y(x_{i+1}) = y(x_i) + hy'(x_i) + \frac{h^2}{2!} y''(x_i) + \ldots + \frac{h^n}{n!} y^{(n)}(x_i) + \frac{h^{n+1}}{(n+1)!} y^{(n+1)}(\xi i), \qquad (6)$$

for some $\xi_i \in (x_i, \mathrm{x}_{i+1})$
Successive differentiation of the solution, $y(x)$, gives

$y'(\mathrm{x}) = f(x, y(x)), y''(x) = f'(x, y(x))$, and, generally, $y^{(k)}(x) = f^{(k-1)}(x, y(x))$.
Substituting these results into Eq. (6) gives

$$y(x_{i+1}) = y(x_i) + hf(x_i, y_i) + \frac{h^2}{2!} f'(x_i, y_i) + \ldots + \frac{h^n}{n!} f^{(n-1)}(x_i, y_i)$$
$$+ \frac{h^{n+1}}{(n+1)!} f^{(n)}(\xi_i, y(\xi_i)), \qquad (7)$$

The difference-equation method corresponding to Eq. (7) is obtained by deleting the remainder term involving ξ_i.

Taylor method of order *n*

$w_0 = \alpha$

$$w_{i+1} = w_i + hf(x_i, w_i) + \frac{h^2}{2!} f'(x_i, w_i) + \ldots + \frac{h^n}{n!} f^{(n-1)}(x_i, w_i) \quad \text{for } i = 1,2,3,\ldots,\text{N}-1$$

Euler's method is Taylor's method of order one.

Example 5

Apply Taylor's method of orders **(a)** two and **(b)** four with $N = 10$ to the initial-value problem

$$y' = y - x^2 + 1, \quad 0 \le tx \le 2, y(0) = 0.5.$$

Solution

(a) For the method of order two we need the first derivative of $f(x, y(x)) = y - x^2 + 1$ with respect to the variable x. Because $y' = y - x^2 + 1$

we have $f'(t, y(t)) = \frac{d}{dx}(y - x^2 + 1) = y' - 2x = y - x^2 + 1 - 2x$

so

$$w_{i+1} = w_i + hf(t_i, w_i) + \frac{h^2}{2} f'(x_i, w_i)$$

$$w_{i+1} = w_i + h\left(w_i - x_i^2 + 1\right) + \frac{h^2}{2}\left(w_i - x_i^2 + 1 - 2x_i\right)$$

$$w_{i+1} = w_i + h\left(w_i - x_i^2 + 1\right) + \frac{h^2}{2}\left(w_i - x_i^2 + 1 - 2x_i\right)$$

$$w_{i+1} = w_i + h\left[\left(1 + \frac{h}{2}\right)\left(w_i - x_i^2 + 1\right) - hx_i\right]$$

Because $N = 10$ we have $h = 0.2$, and $x_i = 0.2i$ for each $i = 1, 2, \ldots, 10$. Thus the

$w_0 = 0.5,$

$$w_{i+1} = w_i + 0.2\left[\left(1 + \frac{0.2}{2}\right)\left(w_i - 0.04i^2 + 1\right) - 0.04i\right]$$

$$w_{i+1} = 1.22w_i - 0.0088i^2 - 0.0088i + 0.22$$

The first two steps give the approximations

$y(0.2) \approx w1 = 1.22(0.5) - 0.0088(0)2 - 0.008(0) + 0.22 = 0.83;$

$y(0.4) \approx w2 = 1.22(0.83) - 0.0088(0.2)2 - 0.008(0.2) + 0.22 = 1.2158$

All the approximations and their errors are shown in Table (3)

i	x_i	Approximate solution	exact solution	Error
0	0	0.5	0.5	0
1	0.2	0.83	0.829298621	0.000701379
2	0.4	1.2158	1.214087651	0.001712349
3	0.6	1.652076	1.6489406	0.0031354
4	0.8	2.13233272	2.127229536	0.005103184
5	1.0	2.648645918	2.640859086	0.007786833
6	1.2	3.19134802	3.179941539	0.011406482
7	1.4	3.748644585	3.732400017	0.016244568
8	1.6	4.306146394	4.283483788	0.022662606
9	1.8	4.8462986	4.815176268	0.031122332
10	2.0	5.347684292	5.305471951	0.042212342

Table 3: Taylor Order 2

(b) For Taylor's method of order four we need the first three derivatives of $f\ (x, y(x))$ with respect to *x*. Again using $y' = y - x^2 + 1$ we have

$$f'(x,y) = y - x^2 + 1 - 2x$$
$$f''(x,y) = y' - 2x - 2$$
$$= y - x^2 + 1 - 2x - 2 = y - x^2 - 2x - 1,$$

$$w_0 = \alpha$$

$$w_{i+1} = w_i + hf(x_i, w_i) + \frac{h^2}{2!} f'(x_i, w_i) + \frac{h^3}{3!} f'(x_i, w_i)$$

$$= w_i + h\left(w_i - x_i^2 + 1\right) + \frac{h^2}{2!}\left(w_i - x_i^2 + 1 - 2x_i\right) + \frac{h^3}{3!}\left(w_i - x_i^2 - 2x_i - 1\right)$$

$$= w_i + \left(h + \frac{h^2}{2} + \frac{h^3}{6}\right)\left(w_i - x_i^2 + 1\right) - h^2 x_i - \frac{h^3}{3}\left(x_i + 1\right)$$

for $i = 0, 1, \ldots, N - 1$.

Because *N* = 10 and *h* = 0.2 the method becomes

$$w_{i+1} = 1.2214 w_i - 0.008856 i^2 - 0.00856 i + 0.2186$$

for each $i = 0, 1, \ldots, 9$. The first two steps give the approximations

$y(0.2) \approx w1 = 1.2214(0.5) - 0.008856(0)2 - 0.00856(0) + 0.2186 = 0.8293;$

$y(0.4) \approx w2 = 1.2214(0.8293) - 0.008856(0.2)2 - 0.00856(0.2) + 0.2186$
$= 1.214091$

All the approximations and their errors are shown in Table (4).

i	x_i	Approximate solution	exact solution	Error
0	0	0.5	0.5	0
1	0.2	0.8293	0.82929862	1.37908E-06
2	0.4	1.21409102	1.21408765	3.36882E-06
3	0.6	1.648946772	1.6489406	6.17202E-06
4	0.8	2.127239587	2.12722954	1.00514E-05
5	1.0	2.640874432	2.64085909	1.53459E-05
6	1.2	3.179964031	3.17994154	2.24922E-05
7	1.4	3.732432067	3.73240002	3.20507E-05
8	1.6	4.283528527	4.28348379	4.47392E-05
9	1.8	4.815237743	4.81517627	6.14751E-05
10	2.0	5.305555379	5.30547195	8.34286E-05

Table (4) Taylor Order 4

Compare these results with those of Taylor's method of order 2 in Table (4) and you will see that the fourth-order results are vastly superior. The results from Table (4) indicate the Taylor's method of order 4 results are quite accurate at the nodes 0.2, 0.4, etc. But suppose we need to determine an approximation to an intermediate point in the table, for example, at $x = 1.25$. If we use linear interpolation on the Taylor method of order four approximations at $x = 1.2$ and $x = 1.4$, we have

$$y(1.25) \approx (\frac{1.25 - 1.4}{1.2 - 1.4})3.1799640 + \frac{1.25 - 1.2}{1.4 - 1.2})3.7324321 = 3.3180810$$

The true value is $y(1.25) = 3.3173285$, so this approximation has an error of 0.0007525,which is nearly 30 times the average of the approximation errors at 1.2 and 1.4.We can significantly improve the approximation by using cubic Hermit interpolation. To determine this approximation for $y(1.25)$ requires approximations to $y(1.2)$ *and* $y(1.4)$as well as approximations to $y(1.2)$ and $y(1.4)$. However, the approximations for $y(1.2)$ and $y(1.4)$ are in the table, and the derivative approximations are available from the differential equation, because $y'(x) = f(x, y)$. In our example $y'(x) = y(x) - x^2 + 1$, so

$$y'(1.2) = y(1.2) - (1.2)^2 + 1 \approx 3.1799640 - 1.44 + 1 = 2.7399640$$

and

$$y'(1.4) = y(1.4) - (1.4)^2 + 1 \approx 3.7324327 - 1.96 + 1 = 2.7724321.$$

EXERCISE (3)

(1) Use Taylor's method of order two to approximate the solutions for each of the following initial-value problems.

(a) $y' = xe^{3x} - 2y, \quad 0 \le x \le 1, \quad y(0) = 0,$ with $h = 0.5$

(b) $y' = 1 + (x - y)^2, \quad 2 \le x \le 3, \quad y(2) = 1,$ with $h = 0.5$

(c) $y' = 1 + \dfrac{y}{x}, \quad 1 \le x \le 2, \quad y(1) = 2,$ with $h = 0.25$

(d) $y' = cos\ 2x + sin\ 3x, \quad 0 \le x \le 1, \quad y(0) = 1, \ withh = 0.25$

(2) Repeat Exercise 1 using Taylor's method of order four.

(3) Given the initial- value problem

$$x^2 y' = 1 - xy - x^2 y^2, \ 1 \le x \le 2, \ y(1) = -1$$

With the exact solution $y = -x^{-1}$

(a) Use Taylor's method of order two with $h = 0.05$ to approximate the solution and compare it with the actual values of y .

(b) Use the answer generated in part (a) and linear interpolation polynomial to approximate the value $y(1.052)$ and $y(1.555)$.

(c) Use Taylor's method of order four with $h = 0.05$ to approximate the solution and compare it with the actual values of y .

(d) Use the answer generated in part (a) and cubic interpolation polynomial to approximate the value $y(1.052)$ and $y(1.555)$.

Runge-Kutta Methods

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 $f(x, y)$. This is a complicated and time-consuming procedure for most problems, so the Taylor methods are seldom used in practice.
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 $f(x, y)$.

Before presenting the ideas behind their derivation, we need to consider Taylor's Theorem in two variables.

Theorem (4) 5.13

Suppose that $f(x, y)$ and all its partial derivatives of order less than or equal to $n + 1$ are continuous on

$D = \{(x, y) \mid a \le x \le b, c \le y \le d\}$,

and let $(x_0, y_0) \in D$. For every $(x, y) \in D$, there exists ξ between x and x_0, and μ between y and y_0 with

$f(x, y) = P_n(x, y) + R_n(x, y)$,

Where

$$P_n(x,y)=f(x_0,y_0)+\left[(x-x_0)\frac{\partial f(x_0,y_0)}{\partial x}+(y-y_0)\frac{\partial f(x_0,y_0)}{\partial y}\right]$$

$$+\left[\frac{(x-x_0)^2}{2}\frac{\partial^2 f(x_0,y_0)}{\partial x^2}+(x-x_0)(y-y_0)\frac{\partial^2 f(x_0,y_0)}{\partial x\partial y}+\frac{(y-y_0)^2}{2}\frac{\partial^2 f(x_0,y_0)}{\partial y^2}\right]+\ldots$$

$$+\left[\frac{1}{n!}\sum_{j=0}^{n}\binom{n}{j}(x-x_0)^{n-j}(y-y_0)^j\frac{\partial^n f(x_0,y_0)}{\partial x^{n-j}\partial y^j}\right]$$

and $$R_n(x,y)=\frac{1}{(n+1)!}\sum_{j=0}^{n+1}\binom{n+1}{j}(x-x_0)^{n+1-j}(y-y_0)^j\frac{\partial^{n+1} f(\xi,\mu)}{\partial x^{n+1-j}\partial y^j}$$

The function $P_n(x,y)$ is called the ***n*th Taylor polynomial in two variables** for the function f about (x_0,y_0), and $R_n(x,y)$ is the remainder term associated with $P_n(x,y)$

Runge-Kutta Methods of Order Two

Consider the differential equation

$$y'=f(x,y),\ \ a\le x\le b,\ \ y(a)=\alpha \qquad (8)$$

Since we want to construct a second-order method, we start with the Taylor expansion

$$y(x+h)=y(x)+hy'(x)+\frac{h^2}{2}y''(x)+O(h^3)$$

The first derivative can be replaced by the right-hand side of the differential equation (8), and the second derivative is obtained by differentiating (8),

$$y''(x)=\frac{d}{dx}f(x,y)=\frac{\partial f}{\partial x}+\frac{\partial f}{\partial y}y' \quad =\frac{\partial f}{\partial x}+\frac{\partial f}{\partial y}f$$

Substitute in (1)

$$y(x+h)=y(x)+hf(x,y)+\frac{h^2}{2}\left(\frac{\partial f}{\partial x}+\frac{\partial f}{\partial y}f(x,y)\right)+O(h^3)$$

$$=y(x)+\frac{h}{2}f(x,y)+\frac{h}{2}\left(f(x,y)+h\frac{\partial f}{\partial x}+h\frac{\partial f}{\partial y}f(x,y)\right)+O(h^3) \qquad (9)$$

Recalling the multivariate Taylor expansion

$$f(x+h,y+k)=f(x,y)+h\frac{\partial f}{\partial x}+k\frac{\partial f}{\partial y}+O(h^2) \qquad (10)$$

We see that the expression in bract in (9) is

$$f(x+h,y+hf)=f(x,y)+h\frac{\partial f}{\partial x}+hf(x,y)\frac{\partial f}{\partial y}+\ldots$$ Therefor we get

$$y(x+h)= y(x)+\tfrac{h}{2}f(x,y)+\tfrac{h}{2}f(x+h,y+hf)+O(h^3)$$
$$= y(x)+\tfrac{h}{2}\big(f(x,y)+f(x+h,y+hf)\big)+O(h^3)$$
$$= y(x)+\tfrac{h}{2}\big(k_1+k_2\big)+O(h^3)$$

Runge-Kutta Methods of Order Two

$w_0=\alpha$

$k_1=f(x_i,y_i)$

$k_2=f(x_i+h,y_i+hk_1)$

$w_{i+1}=w_i+\dfrac{h}{2}\big(k_1+k_2\big)$ for $i=0,1,2,...,N-1$

Midpoint Method

$w_0=\alpha$

$k=hf\ (x_i,w_i)$

$w_{i+1}=w_i+hf(x_i+\tfrac{h}{2},w_i+\tfrac{1}{2}k),$ for $i=0,\ 1,...,N-1.$

Modified Euler Method

$w_0=\ \alpha,$

$k=hf\ \ x_i\ ,w_i$

$w_{i+1}=w_i+\dfrac{h}{2}[f\ \ x_i\ ,w_i\ \ +f\ \ x_{i+1}\ ,\mathrm{w}_i+k)]$ for $i=0,1,\ .\ .\ .\ ,N-1$

Example 6

Use the Midpoint method and the Modified Euler method with N = 10, h = 0.2, $x_i\ =\ 0.2i$,and $w_0\ =\ 0.5,$ to approximate the solution to our usual example,

$y'=y-x^2+1,\ 0\le x\le 2,\quad y(0)=0.5.$

Solution

The difference equations produced from the various formulas are

Midpoint method: $w_{i+1}=1.22w_i-0.0088i^2-0.008i+0.218;$

Modified Euler method: $w_{i+1}=1.22w_i-0.0088i^2-0.008i+0.216$,

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

The first two steps of these methods give

Midpoint method: $w_1=1.22(0.5)-0.0088(0)^2-0.008(0)+0.218=0.828;$

Modified Euler method: $w_1=1.22(0.5)-0.0088(0)^2-0.008(0)+0.216=0.826$, *Midpoint method:*

$w_2 = 1.22(0.828) - 0.0088(0.2)^2 - 0.008(0.2) + 0.218 = 1.21136$

Modified Euler method:

$w_2 = 1.22(0.826) - 0.0088(0.2)^2 - 0.008(0.2) + 0.216 = 1.20692$

Table 5 lists all the results of the calculations. For this problem, the Midpoint method is superior to the Modified Euler method.

x_i	$y(x_i)$	Midpoint Method	Error	Modified Euler Method	Error
0.0	0.5000000	0.5000000	0	0.50000000	0
0.2	0.8292986	0.8280000	0.0012986	0.8260000	0.0032986
0.4	1.2140877	1.2113600	0.0027277	1.2069200	0.0071677
0.6	1.6489406	1.6446592	0.0042814	1.6372424	0.0116982
0.8	2.1272295	2.1212842	0.0059453	2.1102357	0.0169938
1.0	2.6408591	2.6331668	0.0076923	2.6176876	0.0231715
1.2	3.1799415	3.1704634	0.0094781	3.1495789	0.0303627
1.4	3.7324000	3.7211654	0.0112346	3.6936862	0.0387138
1.6	4.2834838	4.2706218	0.0128620	4.2350972	0.0483866
1.8	4.8151763	4.8009586	0.0142177	4.7556185	0.0595577
2.0	5.3054720	5.2903695	0.0151025	5.2330546	0.0724173

Table (5)

Higher-Order Runge-Kutta Methods

The term$T^{(3)}(x, y)$) can be approximated with error $O(h^3)$ by an expression of the form $f(x+\alpha_1, y+\delta_1 f(x+\alpha_2, y+\delta_2 f(x, y)))$, involving four parameters, the algebra involved in the determination of $\alpha_1, \delta_1, \alpha_2,$ and δ_2 is quite involved. The most common $O(h^3)$ is Heun's method, given by

Heun's method

$w_0 = \alpha$

$k_1 = hf(x_i, y_i)$

$k_2 = hf(x_i + \frac{1}{3}\text{h}, w_i + \frac{1}{3}k_1)$

$k_3 = hf(x_i + \frac{2}{3}\text{h}, w_i + \frac{2}{3}k_2)$

$w_{i+1} = w_i + \frac{1}{4}\ k_1 + 3k_3$ for $i = 0,\ 1, \ldots, N-1.$

Illustration

Applying Heun's method with $N=10, h=0.2, x_i=0.2i,$ and $w_0=0.5$ to approximate the solution to our usual example,

$y' = y - x^2 + 1, \quad 0 \le x \le 2, \quad y(0)=0.5.$

gives the values in Table (6) Note the decreased error throughout the range over the Midpoint and Modified Euler approximations.

x_i	$y(x_i)$	Heun's Method	Error
0.0	0.5000000	0.5000000	0
0.2	0.8292986	0.8292444	0.0000542
0.4	1.2140877	1.2139750	0.0001127
0.6	1.6489406	1.6487659	0.0001747
0.8	2.1272295	2.1269905	0.0002390
1.0	2.6408591	2.6405555	0.0003035
1.2	3.1799415	3.1795763	0.0003653
1.4	3.7324000	3.7319803	0.0004197
1.6	4.2834838	4.2830230	0.0004608
1.8	4.8151763	4.8146966	0.0004797
2.0	5.3054720	5.3050072	0.0004648

Table (6)

Runge-Kutta methods of order three are not generally used. The most common Runge-Kutta method in use is of order four in difference-equation form, is given by the following.

Runge-Kutta Order Four

$w0 = \alpha,$

$k_1 = hf(x_i, w_i),$

$k_2 = hf(x_i + \frac{h}{2}, wi + \frac{1}{2}k_1),$

$k_3 = hf(x_i + \frac{h}{2}, wi + \frac{1}{2}k_2),$

$k_4 = hf(x_{i+1}, w_i + k_3),$

$w_{i+1} = w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \qquad$ for $i = 0,\ 1, \ldots, N-1.$

This method has local truncation error $O(h_4)$, provided the solution $y(x)$ as five continuous derivatives. We introduce the notation k_1, k_2, k_3, k_4 into the method is to eliminate the need for successive nesting in the second variable of $f(t, y)$.

Example 7

Use the Runge-Kutta method of order four with h = 0.2, N = 10, and $x_i = 0.2i$ to obtain approximations to the solution of the initial-value problem

$y' = y - x^2 + 1,\ 0 \le x \le 2, \quad y(0) = 0.5.$

Solution

The approximation to $y(0.2)$ is obtained by

$w_0 = 0.5$

$k_1 = 0.2f(0,\ 0.5) = 0.2(1.5) = 0.3$

$k_2 = 0.2f(0.1,\ 0.65) = 0.328$

$k_3 = 0.2f(0.1,\ 0.664) = 0.3308$

$k_4 = 0.2f(0.2,\ 0.8308) = 0.35816$

$w_1 = 0.5 + 16(0.3 + 2(0.328) + 2(0.3308) + 0.35816) = 0.8292933.$

The remaining results and their errors are listed in Table (7).

x_i	Exact $y_i = y(x_i)$	Runge-Kutta Order Four w_i	Error $\lvert y_i - w_i \rvert$
0.0	0.5000000	0.5000000	0.0
0.2	0.8292986	0.8292933	0.0000053
0.4	1.2140877	1.2140762	0.0000114
0.6	1.6489406	1.6489220	0.0000186
0.8	2.1272295	2.1272027	0.0000269
1.0	2.6408591	2.6408227	0.0000364
1.2	3.1799415	3.1798942	0.0000474
1.4	3.7324000	3.7323401	0.0000599
1.6	4.2834838	4.2834095	0.0000743
1.8	4.8151763	4.8150857	0.0000906
2.0	5.3054720	5.3053630	0.0001089

Table (7)

EXERCISE (4)

(1) Use the Modified Euler method to approximate the solutions to each of the following initial-value problems, and compare the results to the actual values.

(a) $y' = xe^{3x} - 2y, \quad 0 \le x \le 1,\ y(0) = 0,\ \textit{with}\ \ h = 0.5;$

Actual solution $\quad y(x) = \frac{1}{5}xe^{3x} - \frac{1}{25}e^{3x} + \frac{1}{25}e^{-2x}.$

(b) $y' = 1 + (x - y)^2, \quad 2 \le x \le 3, \quad y(2) = 1, \quad$ with $h = 0.5;$

Actual solution $\quad y(x) = x + \frac{1}{1-x}$.

(c) $y' = 1 + \frac{y}{x},\ 1 \le x \le 2,\ \ y(1) = 2,$ with $h = 0.25$;

Actual solution $y(x) = x \ln x + 2x.$

(d) $y' = \cos\ 2x + \sin\ 3x,\ 0 \le x \le 1, y(0) = 1,$ with $h = 0.25;$

Actual solution $y(x) = \frac{1}{2}\ \sin\ 2x - \frac{1}{3}\cos\ 3x + \frac{4}{3}$.

(4) Repeat Exercise 1 using the Midpoint method.

(5) Repeat Exercise 1 using the Runge-Kutta method of order four.

(6) Show that Heun's method can be expressed in difference form, similar to that of the Runge-Kutta method of order four, as

$w_0 = \alpha,$

$k_1 = hf(x_i, w_i),$

$k_2 = hf(x_i + \frac{h}{3}, wi + \frac{1}{3}k_1),$

$k_3 = hf(x_i + \frac{2h}{3}, wi + \frac{2h}{3}k_2,$

$w_{i+1} = w_i + \frac{1}{4}(k_1 + 3k_3),$

for each $\ i = 0,\ 1, \ldots, N - 1$

(7) The Runge-Kutta method of order four can be written in the form

$w_0 = \alpha,$

$$w_{i+1} = w_i + \frac{h}{6}f\ (x_i, w_i) + \frac{h}{3}f(x_i + \alpha_1 h, w_i + \delta_1 hf\ (x_i, \mathrm{w}_i))$$

$$+ \frac{h}{3}f(x_i + \alpha_2 h, w_i + \delta_2 hf(x_i + \gamma_2 h, w_i + \gamma_3 hf\ (x_i, w_i)))$$

$$+ \frac{h}{6}f(x_i + \alpha_3 h, w_i + \delta_3 hf(x_i + \gamma_4 h, w_i + \gamma_5 hf(x_i + \gamma_6 h, w_i + \gamma_7 hf(x_i, \mathrm{w}_i)))).$$

Find the values of the constants

$\alpha_1, \alpha_2, \alpha_3, \delta_1, \delta_2, \delta_3, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6,$ and $\gamma_7.$

Multistep Methods

The methods discussed to this point in the chapter are called **one-step methods** because the approximation for the mesh point x_{i+1} involves information from only one of the previous mesh points, x_i. Although these methods might use function evaluation information at points between x_i and x_{i+1}, they do not retain that information for direct use in future approximations. All the information used by these methods is obtained within the subinterval over which the solution is being approximated.

The approximate solution is available at each of the mesh points $x_0, x_1, ..., x_i$ before the approximation at x_{i+1} is obtained, and because the error $\left|w_j - y(x_j)\right|$ tends to increase with *j*, so it seems reasonable to develop methods that use these more accurate previous data when approximating the solution at x_{i+1}. Methods using the approximation at more than one previous mesh point to determine the approximation at the next point are called *multistep* methods. The precise definition of these methods follows, together with the definition of the two types of multistep methods.

Definition (4):

An ***m*-step multistep method** for solving the initial-value problem

$$y' = f\,(x, y), a \le x \le b, \qquad y(a) = \alpha, \qquad (11)$$

has a difference equation for finding the approximation w_{i+1} at the mesh point x_{i+1} represented by the following equation, where *m* is an integer greater than 1:

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m}$$

$$+h[b_m f(x_{i+1}, x_{i+1}) + b_{m-1} f(x_i, x_i) + \cdots + b_0 f(x_{i+1-m}, x_{i+1-m})] \qquad (12)$$

for $i = m - 1, m, \ldots, N - 1,$

where $h = (b - a)/N$, the $a_{m-1}, a_{m-2}, ..., a_0$ and $b_m, b_{m-1}, ..., b_0$.
are constants, and the starting values

$$w_0 = \alpha, w_1 = \alpha_1, w_2 = \alpha_2, \ldots, w_{m-1} = \alpha_{m-1}$$

are specified.

When $b_m = 0$ the method is called **explicit**, or **open**, because Eq. (12) then gives w_{i+1} explicitly in terms of previously determined values. When $b_m \neq 0$ the method is called **implicit**, or **closed**, because w_{i+1} occurs on both sides of Eq. (12), so w_{i+1} is specified only implicitly.

For example,
The equations

$$w_0 = \alpha, w_1 = \alpha_1, w_2 = \alpha_2, w_3 = \alpha_3$$

$$w_{i+1} = w_i + \frac{h}{24}[55f(x_i, w_i) - 59f(x_{i-1}, w_{i-1}) + 37f(x_{i-2}, w_{i-2}) - 9f(x_{i-3}, w_{i-3})],$$

for each $i = 3, 4, \ldots, N-1$, (13)

define an *explicit* four-step method known as the **fourth-order Adams-Bashforth technique**.

The equations

$$w_0 = \alpha, w_1 = \alpha_1, w_2 = \alpha_2$$

$$w_{i+1} = w_i + \frac{h}{24}[9f(x_{i+1}, w_{i+1}) + 19f(x_i, w_i) - 5f(x_{i-1}, w_{i-1}) + f(x_{i-2}, w_{i-2})],$$

for each $i = 2, 3, \ldots, N-1$, (14)

define an *implicit* three-step method known as the **fourth-order**

Adams-Moulton technique.

The starting values in either equation (13) or (14) must be specified, generally by assuming $w_0 = \alpha$ and generating the remaining values by either a Runge-Kutta or Taylor method. We will see that the implicit methods are generally more accurate then the explicit methods, but to apply an implicit method such as (14) directly, we must solve the implicit equation for w_{i+1}. This is not always possible, and even when it can be done the solution for w_{i+1} may not be unique.

Example 8

In Example (7) see Table (7) we used the Runge-Kutta method of order four with $h = 0.2$ to approximate the solutions to the initial value problem

$$y' = y - x^2 + 1, 0 \le x \le 2, y(0) = 0.5.$$

The first four approximations were found to be

$y(0) = w0 = 0.5,$

$y(0.2) \approx w1 = 0.8292933,$

$y(0.4) \approx w2 = 1.2140762$

$, and\ y(0.6) \approx w3 = 1.6489220.$

Use these as starting values for the fourth-order Adams-Bashforth method to compute new approximations for $y(0.8)$ and $y(1.0)$, and compare these new approximations to those produced by the Runge-Kutta method of order four.

Solution

For the fourth-order Adams-Bashforth we have

$$y(0.8) \approx w_4 = w_3 + \frac{0.2}{24}[55f(0.6, w_3) - 59f(0.4, w_2) + 37f(0.2, w_1) - 9f(0, w0)]$$
$$= 1.6489220 + 0.224[55f(0.6, 1.6489220) - 59f(0.4, 1.2140762)$$
$$+ 37f(0.2, 0.8292933) - 9f(0, 0.5)]$$
$$= 1.6489220 + 0.0083333[55(2.2889220) - 59(2.0540762) + 37(1.7892933) - 9(1.5)]$$
$$= 2.1272892$$

and

$$y(1.0) \approx w_5 = w_4 + \frac{0.2}{24}(55f(0.8, w_4) - 59f(0.6, w_3)$$
$$+ 37f(0.4, w_2) - 9f(0.2, w_1))$$
$$= 2.1272892 + \frac{0.2}{24}(55f(0.8,\ 2.1272892) - 59f(0.6,\ 1.6489220)$$
$$+ 37f(0.4,\ 1.2140762) - 9f(0.2,\ 0.8292933))$$
$$= 2.1272892 + 0.0083333(55(2.4872892) - 59(2.2889220)$$
$$+ 37(2.0540762) - 9(1.7892933)) = 2.6410533$$

The error for these approximations at x= 0.8 and x = 1.0 are, respectively

$|2.1272295 - 2.1272892| = 5.97 \times 10{-}5$ and $|2.6410533 - 2.6408591| = 1.94 \times 10^{-4}$.
The corresponding Runge-Kutta approximations had errors
$|2.1272027 - 2.1272892| = 2.69 \times 10{-}5$ and $|2.6408227 - 2.6408591| = 3.64 \times 10^{-5}$.

Adams-Bashforth method

To begin the derivation of a multistep method, note that the solution to the initial-value problem

$$y' = f(x, y), a \leq x \leq b, y(a) = \alpha,$$

if integrated over the interval $[x_i, x_{i+1}]$, has the property that

$$y(x_{i+1}) - y(x_i) = \int_{x_i}^{x_{i+1}} y'(x)dx = \int_{x_i}^{x_{i+1}} f(x, y(x))dx.$$

Consequently,

$$y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} f(x, y(x))dx.. \qquad (15)$$

However we cannot integrate $f(x, y(x))$ without knowing $y(x)$, the solution to the problem, so we instead integrate an interpolating polynomial $P(x)$ to $f(x, y(x))$, one that is determined by some of the previously obtained data points
$(x_0, w_0), (x_1, w_1), ..., (x_i, w_i)$. When we assume, in addition, that $y(x_i) \approx w_i$,
Eq. (15) becomes

$$y(x_{i+1}) \approx w_i + \int_{x_i}^{x_{i+1}} P\ t\ dt. \qquad 16$$

To derive an Adams-Bashforth explicit *m*-step technique, we form the backward difference polynomial $P_{m-1}(x)$ through $(x_i, f_i), (x_{i-1}, f_{i-1}), ...,$ (x_{i-1+m}, f_{i-1+m}) where $f_i = f(x_i, y(x_i))$.

Since P_{m-1}(x) is an interpolator polynomial of degree *m*−1, some number ξ_i in x_{i+1-m}, x_i exists with

$$f(x, y(x)) = P_{m-1}(x) + \frac{f^{(m)}(\xi_i, y(\xi_i))}{m!}(x - x_i)(x - x_{i-1})...(x - x_{i+1-m})$$

Introducing the variable substitution $x = x_i + sh$, with *dt* = *h ds*, into $P_{m-1}(t)$ and the error term implies that

$$\int_{x_i}^{x_{i+1}} f\ (x, y)dx = \int_{x_i}^{x_{i+1}} P_{m-1}dx = \int_{x_i}^{x_{i+1}} \sum_{k=0}^{m-1} (-1)^k \binom{-s}{k} \nabla^k f(x_i, y_i)dx$$

$$+ \int_{x_i}^{x_{i+1}} \frac{f^{(m)}(\xi_i, y(\xi_i))}{m!}(x - x_i)(x - x_{i-1})...(x - x_{i+1-m})dx$$

$$\int_{x_i}^{x_{i+1}} f(x, y)dx = \sum_{k=0}^{m-1} h\nabla^k f(x_i, y_i)(-1)^k \int_0^1 \binom{-s}{k} ds$$

$$+ \frac{h^{m+1}}{m!} \int_0^1 (s)(s+1)...(s+m-1) f^{(m)}(\xi_i, y(\xi_i))ds$$

The integral $I_k = (-1)^k \int_0^1 \binom{-s}{k} ds$ for various values of *k* are easily evaluated and are listed in Table(8)

k	I_k
0	1
1	$\frac{1}{2}$
2	$\frac{5}{12}$
3	$\frac{3}{8}$
4	$\frac{251}{720}$
5	$\frac{95}{288}$

Table (8)

And we use the relation $\binom{-s}{k} = (-1)^k \binom{s+k-1}{k}$

For example, when k = 3, $\binom{-s}{3} = (-1)^3 \binom{s+2}{3} = \frac{(s+2)(s+1)s}{3!}$

$$(-1)^k \int_0^1 \binom{-s}{k} ds = (-1)^3 \int_0^1 \binom{-s}{3} ds = \frac{(-1)^3(-1)^3}{3!} \int_0^1 (s)(s+1)(s+2) = \frac{3}{8}$$

As a consequence,

$$\int_{x_i}^{x_{i+1}} f(x,y)dx = h\left[f(x_i,y_i) + \frac{1}{2}\nabla f(x_i,y_i) + \frac{5}{12}\nabla^2 f(x_i,y_i) + ...\right]$$

$$+ \frac{h^{m+1}}{m!} \int_0^1 (s)(s+1)...(s+m-1) f^{(m)}(\xi_i, y(\xi_i))ds \qquad (17)$$

Because $s(s+1)\dots(s+m-1)$ does not change sign on [0, 1], the Weighted Mean Value Theorem for Integrals can be used to deduce that for some number μ_i , where $x_{i+1-m} < \mu_i < x_{i+1}$, the error term in Eq. (17) becomes

$$\frac{h^{m+1}}{m!} \int_0^1 (s)(s+1)...(s+1-m) f^{(m)}(\xi_i, y(\xi_i))ds$$

$$= \frac{h^{m+1} f^{(m)}(\mu_i, y(\mu_i))}{m!} \int_0^1 (s)(s+1)...(s+m-1)ds$$

$$= h^{m+1} f^{(m)}(\mu_i, y(\mu_i)) \int_0^1 \frac{(s)(s+1)...(s+m-1)}{m!} ds$$

Hence the error in (17) simplifies to

$$E_m = h^{m+1} f^{(m)}(\mu_i, y(\mu_i))(-1)^m \int_0^1 \binom{-s}{m} ds \qquad (18)$$

But $y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} f(x, y(x))dx.$

so Eq. (15) can be written as

$$y(x_{i+1}) = y(x_i) + h\sum_{k=0}^{m-1} \nabla^k f(x_i,y_i).I_k + h^{m+1} f^{(m)}(\mu_i, y(\mu_i))\ .I_m \qquad (19)$$

Where $(-1)^k \int_0^1 \binom{-s}{k} ds$

Example 9
Use Eq. (19) with *m* = 3 to derive the three-step Adams-Bashforth technique.
Solution
We have

$$y(x_{i+1}) = y(x_i) + h\left[f(x_i, y_i) + \frac{1}{2}\nabla f(x_i, y_i) + \frac{5}{12}\nabla^2 f(x_i, y_i) + ...\right]$$

$$= y(x_i) + h\left[\begin{array}{l} f(x_i, y_i) + \frac{1}{2}\ f(x_i, y_i) - f(x_{i-1}, y_{i-1}) \\ + \frac{5}{12}\ f(x_i, y_i) - 2f(x_{i-1}, y_{i-1}) + f(x_{i-2}, y_{i-2}) \end{array}\right]$$

$$y(x_{i+1}) = y(x_i) + \frac{h}{12}\left[23f(x_i, y_i) - 16f(x_{i-1}, y_{i-1}) + 5f(x_{i-2}, y_{i-2})\right]$$

Then the general form

$$y(x_{i+1}) = y(x_i) + \frac{h}{12}\ 23f(x_i, y_i) - 16f(x_{i-1}, y_{i-1}) + 5f(x_{i-2}, y_{i-2})$$

$$E_m = h^4\ f^{\ (3)}(\mu_i, y(\mu_i))I_3 = \frac{3}{8}h^4\ f^{\ (3)}(\mu_i, y(\mu_i))$$

The three-step Adams-Bashforth method is, consequently,

$w_0 = \alpha,\ w_1 = \alpha_1,\ w_2 = \alpha_2,$

$$w_{i+1} = w_i + \frac{h}{12}\ 23f(x_i, w_i) - 16f(x_{i-1}, w_{i-1}) + 5f(x_{i-2}, w_{i-2})$$

for $i = 2, 3, \ldots, N - 1$.
Multistep methods can also be derived using Taylor series. An example of the procedure involved is considered in following Exercise

EXERCISE (5)

(1) Derive the Adams-Bashforth Three-Step method by the following method. Set
$$y(x_{i+1}) = y(x_i) + ahf(x_i, y_i) + bhf(x_{i-1}, y_{i-1}) + chf(x_{i-2}, y_{i-2})$$
Expand $f(x_i, y_i), f(x_{i-1}, y_{i-1})$ and $f(x_{i-2}, y_{i-2})$ in Taylor series about (x_i, y_i), and equate the coefficients of *h*, *h*2 and *h*3 to obtain *a*, *b*, and *c*.

(2) (a) Derive the Adams-Bashforth Two-Step method by using the Lagrange form of the interpolating polynomial.
(b) Derive the Adams-Bashforth Four-Step method by using Newton's backward-difference form of the interpolating polynomial.
(3) Derive Simpson's method by applying Simpson's rule to the integral

$$y(x_{i+1}) - y(x_i) = \int_{x_i}^{x_{i+1}} f(x, y(x))dx.$$

The local truncation error for multistep methods is defined analogously to that of One-step methods. As in the case of one-step methods, the local truncation error provides a measure of how the solution to the differential equation fails to solve the difference equation.

Definition (5)
If $y(x)$ is the solution to the initial-value problem

$$y' = f(x,y), a \le x \le b, y(a) = \alpha,$$

and

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \cdot \cdot \cdot + a_0 w_{i+1-m}$$
$$+ h[b_m f(x_{i+1}, w_{i+1}) + b_{m-1} f(x_i, w_i) + \cdot \cdot \cdot + b_0 f(x_{i+1-m}, w_{i+1-m})]$$

is the $(i+1)st$ step in a multistep method, the **local truncation error** at this step is

$$\tau_{i+1} = \frac{w_{i+1} - a_{m-1}w_i - a_{m-2}w_{i-1} - ... - a_0 w_{i+1-m}}{h}$$
$$-[b_m f(x_{i+1}, w_{i+1}) + b_{m-1} f(x_i, w_i) + ... + b_0 f(x_{i+1-m}, w_{i+1-m})] \quad (20)$$

for each $i = m-1, m, \ldots, N-1$.

Example 10
Determine the local truncation error for the three-step Adams-Bashforth method derived in Example 9.

Solution
Considering the form of the error given in Eq. (18) and the appropriate entry in Table 8 gives

$$E_m = h^{m+1} f^{(m)}(\mu_i, y(\mu_i))(-1)^m \int_0^1 \binom{-s}{m} ds$$

$$E_3 = h^4 f^{(3)}(\mu_i, y(\mu_i))(-1)^3 \int_0^1 \binom{-s}{3} ds = \frac{3h^4}{8} h^4 f^{(3)}(\mu_i, y(\mu_i))$$

Using the fact that $f^{(3)}(\mu_i, y(\mu_i)) = y^{(4)}(\mu_i)$ and the difference equation derived in Example 9, the truncation error is

$$\tau_{i+1} = \frac{y(x_{i+1}) - y(x_i)}{h} - \frac{1}{12}\ 23f(x_i, y_i) - 16f(x_{i-1}, y_{i-1}) + 5f(x_{i-2}, y_{i-2})$$
$$= \frac{1}{h}\left[\frac{3h^4}{12} f^{(3)}(\mu_i, y(\mu_i))\right] = \frac{3h^3}{8} f^{(3)}(\mu_i, y(\mu_i)) \qquad \text{For } \mu_i \in\ x_{i-2}, x_{i+1}$$

Adams-Bashforth Explicit Methods

Some of the explicit multistep methods together with their required starting values and local truncation errors are as follows. The derivation of these techniques is similar to the procedure in Examples 9 and 10.

Adams-Bashforth Two-Step Explicit Method

Use Eq. (19) with m = 2 to derive the two-step Adams-Bashforth technique.

$$y(x_{i+1}) = y(x_i) + h\sum_{k=0}^{m-1} \nabla^k f(x_i, y_i)(-1)^k \int_0^1 \binom{-s}{k} ds$$

$$+ h^{m+1} f^{(m)}(\mu_i, y(\mu_i))\ (-1)^m \int_0^1 \binom{-s}{m} ds$$

$$y(x_{i+1}) = y(x_i) + h\left[f(x_i, y_i) + \frac{1}{2}\nabla f(x_i, y_i)\right] + \frac{5}{12}h^3 f^{(2)}(\mu_i, y(\mu_i))$$

$$y(x_{i+1}) = y(x_i) + h\left[f(x_i, y_i) + \frac{1}{2}\nabla f(x_i, y_i)\right] + \frac{5}{12}h^3 f^{(2)}(\mu_i, y(\mu_i))$$

$$y(x_{i+1}) = y(x_i) + \frac{h}{2}\left[3f(x_i, y_i) - f(x_{i-1}, y_i - 1)\right] + \frac{5}{12}h^3 y^{(3)}(\mu_i)$$

$w_0 = \alpha, w_1 = \alpha_1,$

$$w_{i+1} = w_i + \frac{h}{2}\ 3f(x_i, w_i) - f(x_{i-1}, w_i - 1) \qquad (21)$$

Where $i = 1, 2, \ldots, N - 1$.

The local truncation error is $\tau_{i+1} = \frac{5}{12}h^2\ y^{(3)}(\mu_i)$ for some $\mu_i \in\ x_{i-1}, x_{i+1}$

Adams-Bashforth Three-Step Explicit Method

When m = 3 we derive the three-step Adams-Bashforth technique

$w_0 = \alpha,\ w_1 = \alpha_1,\ w_2 = \alpha_2,$

$$w_{i+1} = w_i + \frac{h}{12}\ 23f(x_i, w_i) - 16f(x_{i-1}, w_{i-1}) + 5f(x_{i-2}, w_{i-2}) \qquad (22)$$

Where $i = 2, 3, \ldots, N - 1$.

The local truncation error is $\tau_{i+1} = \frac{3h^3}{8} f^{(3)}(\mu_i, y(\mu_i))$ for some $\mu_i \in\ x_{i-2}, x_{i+1}$

Adams-Bashforth Four-Step Explicit Method

when $m = 4$ to derive the four-step Adams-Bashforth technique

$w_0 = \alpha,\ w_1 = \alpha_1,\ w_2 = \alpha_2, w_3 = \alpha_3$

$$w_{i+1} = w_i + \frac{h}{24}\ 55f(x_i, w_i) - 59f(x_{i-1}, w_{i-1}) + 37f(x_{i-2}, w_{i-2}) - 9f(x_{i-3}, w_{i-3}) \qquad (23)$$

Where $i = 3, 4, \ldots, N - 1$.

The local truncation error is $\tau_{i+1} = \dfrac{251h^4}{720} f^{(4)}(\mu_i, y(\mu_i))$

for some $\mu_i \in x_{i-3}, x_{i+1}$

Adams-Bashforth Five-Step Explicit Method

Put $m = 5$ to derive the five-step Adams-Bashforth technique

$w_0 = \alpha,\ w_1 = \alpha_1,\ w_2 = \alpha_2, w_3 = \alpha_3, w_4 = \alpha_4$

$$\boxed{w_{i+1} = w_i + \frac{h}{720}\left[\begin{array}{l} 1901f(x_i, w_i) - 2774f(x_{i-1}, w_{i-1}) + 2616f(x_{i-2}, w_{i-2}) \\ -1274f(x_{i-3}, w_{i-3}) + 251f(x_{i-4}, w_{i-4}) \end{array}\right]} \quad (24)$$

Where $i = 4, 5, \ldots, N - 1$.

The local truncation error is $\tau_{i+1} = \dfrac{95\ h^5}{288} f^{(5)}(\mu_i, y(\mu_i))$

for some $\mu_i \in x_{i-4}, x_{i+1}$

Adams-Moulton Implicit Methods

Implicit methods are derived by using $(x_{i+1}, f(x_{i+1}, y(x_{i+1})))$ as an additional interpolation node in the approximation of the integral

$$\int_{x_i}^{x_{i+1}} f(x, y)dx$$

To begin the derivation of **Adams-Moulton Implicit Methods**, note that the solution to the initial-value problem

$$y' = f(x, y), a \leq x \leq b, y(a) = \alpha,$$

if integrated over the interval $[x_i, x_{i+1}]$, has the property that

$$y(x_{i+1}) - y(x_i) = \int_{x_i}^{x_{i+1}} y'(x)dx = \int_{x_i}^{x_{i+1}} f(x, y(x))dx.$$

Consequently,

$$y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} f(x, y(x))dx.. \quad (15)$$

However we cannot integrate $f(x, y(x))$ without knowing $y(x)$, the solution to the problem, so we instead integrate an interpolating polynomial $P(x)$ to $f(x, y(x))$, one that is determined by some of the previously obtained data points $(x_0, w_0), (x_1, w_1), \ldots, (x_i, w_i), (x_{i+1}, w_{i+1})$ When we assume, in addition, that $y(x_i) \approx w_i$, Eq. (15) becomes $y(x_{i+1}) \approx w_i + \int_{x_i}^{x_{i+1}} P\ t\ dt.$ (16)

Although any form of the interpolating polynomial can be used for the derivation, it is most convenient to use the Newton backward-difference formula including the point (x_{i+1}, w_{i+1}) , because this form more easily incorporates the most recently calculated data.

To derive an **Adams-Moulton Implicit Methods** *m*-step technique, we form the backward difference polynomial $P_m(x)$ through the set of points

$(x_{i+1}, f_{i+1}), (x_i, f_i), (x_{i-1}, f_{i-1}), \ldots, (x_{i+1-m}, f_{i+1-m})$ where $f_i = f(x_i, y(x_i))$.

Since $P_m(x)$ is an interpolator polynomial of degree *m*, some number ξ_i in x_{i+1-m}, x_{i+1}

exists with

$$f(x, y(x)) = P_m(x) + \frac{f^{(m+1)}(\xi_i, y(\xi_i))}{(m+1)!}(x - x_{i+1})(x - x_i)(x - x_{i-1}) \cdot \cdot (x - x_{i+1-m})$$

Introducing the variable substitution $x = x_i + sh$, with *dt* = *h ds*, into $P_m(t)$ and the error term implies that

$$\int_{x_i}^{x_{i+1}} f\ (x, y)dx = \int_{x_i}^{x_{i+1}} P_m\,dx = \int_{x_i}^{x_{i+1}} \sum_{k=0}^{m} (-1)^k \binom{-s}{k} \nabla^k f(x_{i+1}, y_{i+1})dx$$

$$+ \int_{x_i}^{x_{i+1}} \frac{f^{(m+1)}(\xi_i, y(\xi_i))}{(m+1)!}(x - x_{i+1})(x - x_i)\ldots(x - x_{i+1-m})dx$$

$$\int_{x_i}^{x_{i+1}} f(x, y)dx = \sum_{k=0}^{m} h\nabla^k f(x_{i+1}, y_{i+1})(-1)^k \int_0^1 \binom{-s+1)}{k} ds$$

$$+ \frac{h^{m+2}}{(m+1)!} \int_0^1 (s-1)(s)(s+1)\ldots(s+m-1) f^{(m)}(\xi_i, y(\xi_i))ds$$

The integral $I_k = (-1)^k \int_0^1 \binom{-s+1}{k} ds$ for various values of *k* are easily evaluated and are listed in the following table

And we use the relation $\binom{-s}{k} = (-1)^k \binom{s+k-1}{k}$

For example, when *k* = 3, $\binom{-s}{3} = (-1)^3 \binom{s+2}{3} = \frac{(s+2)(s+1)s}{3!}$

$$(-1)^k \int_0^1 \binom{-s}{k} ds = (-1)^3 \int_0^1 \binom{-s}{3} ds = \frac{(-1)^3(-1)^3}{3!} \int_0^1 (s)(s+1)(s+2) = \frac{3}{8}$$

k	I_k
0	1
1	$\frac{-1}{2}$
2	$\frac{-1}{12}$
3	$\frac{-1}{24}$
4	$\frac{-53}{60}$

As a consequence,

$$\int_{x_i}^{x_{i+1}} f\,(x,y)dx = h\left[f(x_{i+1},y_{i+1}) + \frac{1}{2}\nabla f(x_{i+1},y_{i+1}) + \frac{5}{12}\nabla^2 f(x_{i+1},y_{i+1}) + \ldots\right]$$
$$+\frac{h^{m+2} f^{\,(m+1)}(\mu_i, y(\mu_i))}{(m+1)!}\int_0^1 (\text{s}-1)(s)(s+1)\ldots(s+m-1)ds \qquad (17)$$

Because $(s-1)s(s+1)\ldots(s+m-1)$ does not change sign on [0, 1], the Weighted Mean Value Theorem for Integrals can be used to deduce that for some number μ_i , where $x_{i+1-m} < \mu_i < x_{i+1}$, the error term in Eq. (17) becomes

$$\frac{h^{m+2}}{(m+1)!}\int_0^1 (s-1)(s)(s+1)\ldots(s+1-m) f^{\,(m+1)}(\xi_i, y(\xi_i))ds$$
$$= \frac{h^{m+2} f^{\,(m+1)}(\mu_i, y(\mu_i))}{(m+1)!}\int_0^1 (s-1)(s)(s+1)\ldots(s+m-1)ds$$
$$= h^{m+2} f^{\,(m+1)}(\mu_i, y(\mu_i))\int_0^1 \frac{(s-1)(s)(s+1)\ldots(s+m-1)}{(m+1)!}ds$$

Hence the error in (17) simplifies to

$$E_m = h^{m+2}\, f^{(m+1)}(\mu_i, y(\mu_i))(-1)^{m+1}\int_0^1 \binom{-(s-1)}{m+1} ds \qquad (18)$$

But $y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} f(x, y(x))dx.$

Then the difference equation is

$$\boxed{y(x_{i+1}) = y(x_i) + h\sum_{k=0}^{m} \nabla^k f(x_{i+1}, y_{i+1}).I_k + h^{m+2}\, f^{(m+1)}(\mu_i, y(\mu_i))\ .I_m} \qquad (19)$$

Where $I_k = (-1)^k \int_0^1 \binom{-s+1}{k} ds$ and $\boxed{I_m = (-1)^{m+1}\int_0^1 \binom{-s+1}{m+1} ds}$

When $m = 2$ we drive Adams-Moulton Two-Step method

Consider $f_i = f(x_{i+1}, y_{i+1})$.

$$y(x_{i+1}) = y(x_i) + h\sum_{k=0}^{2} \nabla^k f(x_{i+1}, y_{i+1}).I_k + h^4\, f^{(3)}(\mu_i, y(\mu_i))\ .I_m$$

$$y(x_{i+1}) = y(x_i) + h\left[I_0 f_{i+1}. + I_1 \nabla f_{i+1} + I_2 \nabla^2 f_{i+1}\right]$$

$$- h^4\, f^{(3)}(\mu_i, y(\mu_i)) \int_0^1 \binom{-s+1}{3} ds$$

$$= y(x_i) + h\left[f_{i+1}. - \tfrac{1}{2}\nabla f_{i+1} - \tfrac{1}{12}\nabla^2 f_{i+1}\right] - \frac{h^4}{24} f^{(3)}(\mu_i, y(\mu_i))$$

$$= y(x_i) + h\left[f_{i+1}. - \left(f_{i+1} - f_i\right) - \frac{1}{12}\left(f_{i+1} - 2f_i + f_{i-1}\right)\right] - \frac{h^4}{24} f^{(3)}(\mu_i, y(\mu_i))$$

$$= y(x_i) + \frac{h}{12}\left[f_{i+1}. + 8f_i - f_{i-1}\right] - \frac{h^4}{24} f^{(3)}(\mu_i, y(\mu_i))$$

Adams-Moulton Two-Step Implicit Method

$w_0 = \alpha,\ w_1 = \alpha_1$

$$\boxed{w_{i+1} = w_i + \frac{h}{12}\left[5f(x_{i+1}, w_{i+1}) + 8f(x_i, w_i) - f(x_{i-1}, w_{i-1})\right]} \qquad (25)$$

where $i = 1, 2, 3, \ldots, N - 1.$

The local truncation error is $\tau_{i+1} = \frac{-h^3}{24} f^{(3)}(\mu_i, y(\mu_i))$

for some $\mu_i \in x_{i-1}, x_{i+1}$

Adams-Moulton Three-Step Implicit Method

$w_0 = \alpha,\ w_1 = \alpha_1,\ w_2 = \alpha_2$

$$w_{i+1} = w_i + \frac{h}{24}\left[9f(x_{i+1}, w_{i+1}) + 19f(x_i, w_i) - 5f(x_{i-1}, w_{i-1}) + f(x_{i-2}, w_{i-2})\right] \qquad (26)$$

where $i = 2, 3, \ldots, N - 1.$

The local truncation error is $\tau_{i+1} = \dfrac{-19h^4}{720} f^{(4)}(\mu_i, y(\mu_i))$

for some $\mu_i \in x_{i-2}, x_{i+1}$

Adams-Moulton Four-Step Implicit Method

$w_0 = \alpha,\ w_1 = \alpha_1,\ w_2 = \alpha_2,\ w_3 = \alpha_3$

$$w_{i+1} = w_i + \frac{h}{720}[251f(x_{i+1}, w_{i+1}) + 646f(x_i, w_i) - 264f(x_{i-1}, w_{i-1}) + 106f(x_{i-2}, w_{i-2}) - 19f(x_{i-3}, w_{i-3})] \qquad (27)$$

where $i = 3, 4, \ldots, N - 1.$

The local truncation error is $\tau_{i+1} = \dfrac{-3h^5}{160} f^{(5)}(\mu_i, y(\mu_i))$

for some $\mu_i \in x_{i-3}, x_{i+1}$

It is interesting to compare an *m*-step Adams-Bashforth explicit method with an $(m-1)$ −step Adams-Moulton implicit method. Both involve *m* evaluations of *f* per step, and both have the terms $f^{\,m+1}(\mu_i, \mu_i)h^m$ in their local truncation errors. In general, the coefficients of the terms involving *f* in the local truncation error are smaller for the implicit methods than for the explicit methods. This leads to greater stability and smaller round-off errors for the implicit methods.

Example 11

Consider the initial-value problem

$$y' = y - x^2 + 1, 0 \leq x \leq 2, \qquad y(0) = 0.5.$$

Use the exact values given from $y(x) = (x + 1)^2 - 0.5e^x$ as starting values and *h* = 0.2 to compare the approximations from

(a) By the explicit Adams-Bashforth four-step method and
(b) The implicit Adams-Moulton three-step method.

Solution

(a) The Adams-Bashforth method has the difference equation

$$w_0 = \alpha,\ w_1 = \alpha_1,\ w_2 = \alpha_2, w_3 = \alpha_3$$

$$w_{i+1} = w_i + \frac{h}{24}\ 55f(x_i, w_i) - 59f(x_{i-1}, w_{i-1}) + 37f(x_{i-2}, w_{i-2}) - 9f(x_{i-3}, w_{i-3})$$

For $i = 3, 4, \ldots, 9$.
When simplified using $f(x, y) = y - x^2 + 1, \quad h = 0.2,$

$$w_{i+1} = \frac{1}{24}[35w_{i+1} - 11.8w_{i-1} + 7.4w_{i-2} - 1.8w_{i-3} - 0.192i^2 - 0.192i + 4.736]$$

(b) The Adams-Moulton method has the difference equation

$$w_{i+1} = w_i + \frac{h}{24}[9f(x_{i+1}, w_{i+1}) + 19f(x_i, w_i) - 5f(x_{i-1}, w_{i-1}) + f(x_{i-2}, w_{i-2})],$$

for $i = 2, 3, \ldots, 9$. This reduces to

$$w_{i+1} = \frac{1}{24}[1.8\, w_{i+1} + 27.8\, w_i - w_{i-1} + 0.2w_{i-2} - 0.192i^2 - 0.192i + 4.736].$$

To use this method explicitly, we need to solve the equation explicitly solve for w_{i+1}
This gives

$$w_{i+1} = \frac{1}{22.2}[27.8w_i - w_{i-1} + 0.2w_{i-2} - 0.192i^2 - 0.192i + 4.736],$$

for $i = 2, 3, \ldots, 9$.

The results in Table (9) were obtained using the exact values from $y(x) = (x + 1)^2 - 0.5e^x$ for $\alpha,\ \alpha_1,\ \alpha_2,$ and α_3 in the explicit Adams-Bashforth case and for $\alpha,\ \alpha_1,$, and α_2 in the implicit Adams-Moulton case. Note that the implicit Adams-Moulton method gives consistently better results.

xi	Exact *wi*	AB	Error	AM	Error
0.0	0.5000000				
0.2	0.8292986				
0.4	1.2140877				
0.6	1.6489406			1.6489341	0.0000065
0.8	2.1272295	2.1273124	0.0000828	2.1272136	0.0000160
1.0	2.6408591	2.6410810	0.0002219	2.6408298	0.0000293
1.2	3.1799415	3.1803480	0.0004065	3.1798937	0.0000478
1.4	3.7324000	3.7330601	0.0006601	3.7323270	0.0000731
1.6	4.2834838	4.2844931	0.0010093	4.2833767	0.0001071
1.8	4.8151763	4.8166575	0.0014812	4.8150236	0.0001527
2.0	5.3054720	5.3075838	0.0021119	5.3052587	0.0002132

Table (9)

Predictor-Corrector Methods

In Example 9 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

$$y' = e^y, 0 \le x \le 0.25, \qquad y(0) = 1.$$

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

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

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 of an explicit method to predict and an implicit to improve the prediction is called a **predictor-corrector method**.

Consider the following fourth-order method for solving an initial-value problem. The first step is to calculate the starting values w_0, w_1, w_2 and w_3 for the four-step explicit Adams-Bashforth method. To do this, we use a fourth-order one-step method, the Runge-Kutta method of order four. The next step is to calculate an approximation, w_{4p} , to $y(x_4)$ using the explicit Adams-Bashforth method as predictor:

$$w_{4p} = w_3 + \frac{h}{24}[55f(x_3, w_3) - 59f(x_2, w_2) + 37f(x_1, w_1) - 9f(x_0, w_0)].$$

This approximation is improved by inserting *w*4*p* in the right side of the three-step implicit Adams-Moulton method and using that method as a corrector. This gives

$$w_4 = w_3 + \frac{h}{24}[9f(x_4, w_{4p}) + 19f(x_3, w_3) - 5f(x_2, w_2) + f(x_1, w_1)].$$

The only new function evaluation required in this procedure is $f(x_4, w_{4p})$ in the corrector equation; all the other values of f have been calculated for earlier approximations.

The value w_4 is then used as the approximation to $y(x_4)$, and the technique of using the Adams-Bashforth method as a predictor and the Adams-Moulton method as a corrector is repeated to find w_{5p} and w_5, the initial and final approximations to $y(x_5)$

This process is continued until we obtain an approximation *wc* to $y(x_N) = y(b)$.

Improved approximations to $y(x_{i+1})$ might be obtained by iterating the Adams-Moulton formula, but these converge to the approximation given by the implicit formula rather than to the solution $y(x_{i+1})$. Hence it is usually more efficient to use a reduction in the step size if improved accuracy is needed.

Example 12

Apply the Adams fourth-order predictor-corrector method with h = 0.2 and starting values from the Runge-Kutta fourth order method to the initial-value problem

$$y' = y - x^2 + 1, 0 \le x \le 2, \qquad y(0) = 0.5.$$

Solution

This is continuation and modification of the problem considered in Example 9 at the beginning of the section. In that example we found that the starting approximations from Runge-Kutta are

$y(0) = w_0 = 0.5,\ y(0.2) \approx w_1 = 0.8292933,\ y(0.4) \approx w_2 = 1.2140762$, and $y(0.6) \approx w_3 = 1.6489220$. and the fourth-order Adams-Bashforth method gave

$$y(0.8) \approx w_{4p} = w_3 + \tfrac{0.2}{24}(55f(0.6, w_3) - 59f(0.4, w_2) + 37f(0.2, w_1) - 9f(0, w_0))$$

$$= 1.6489220 + \tfrac{0.2}{24}(55f(0.6,\ 1.6489220) - 59f(0.4,\ 1.2140762)$$

$$+ 37f(0.2,\ 0.8292933) - 9f(0,\ 0.5))$$

$$= 1.6489220 + 0.0083333(55(2.2889220) - 59(2.0540762) + 37(1.7892933)$$

$$- 9(1.5)) = 2.1272892.$$

We will now use $w4p$ as the predictor of the approximation to $y(0.8)$ and determine the corrected value $w4$, from the implicit Adams-Moulton method. This gives

$$y(0.8) \approx w_4$$

$$= w_3 + \tfrac{0.2}{24}[9f(0.8, w_{4p}) + 19f(0.6, w_3) - 5f(0.4, w_2) + f(0.2, w_1)]$$

$$= 1.6489220 + \tfrac{0.2}{24}(9f(0.8,\ 2.1272892) + 19f(0.6,\ 1.6489220)$$

$$- 5f(0.4,\ 1.2140762) + f(0.2,\ 0.8292933))$$

$$= 1.6489220 + 0.0083333(9(2.4872892) + 19(2.2889220)$$

$$- 5(2.0540762) + (1.7892933)) = 2.1272056$$

Now we use this approximation to determine the predictor, w_{5p}, for $y(1.0)$ as

$$y(1.0) \approx w_{5p}$$

$$= w_4 + \tfrac{0.2}{24}(55f(0.8, w_4) - 59f(0.6, w_3) + 37f(0.4, w_2) - 9f(0.2, w_1))$$

$$= 2.1272056 + \tfrac{0.2}{24}(55f(0.8,\ 2.1272056) - 59f(0.6,\ 1.6489220)$$

$$+ 37f(0.4,\ 1.2140762) - 9f(0.2,\ 0.8292933))$$

$$= 2.1272056 + 0.0083333(55(2.4872056) - 59(2.2889220) + 37(2.0540762)$$

$$- 9(1.7892933)) = 2.6409314$$

and correct this with

$$y(1.0) \approx w_5 = w4 + \frac{0.2}{24}[9f(1.0, w_{5p}) + 19f(0.8, w_4) - 5f(0.6, w_3) + f(0.4, w_2)]$$

$$= 2.1272056 + \frac{0.2}{24}(9f(1.0,\ 2.6409314) + 19f(0.8,\ 2.1272892)$$

$$- 5f(0.6,\ 1.6489220) + f(0.4,\ 1.2140762))$$

$$= 2.1272056 + 0.0083333(9(2.6409314) + 19(2.4872056) - 5(2.2889220)$$

$$+ (2.0540762)) = 2.6408286$$

In Example 9 we found that using the explicit Adams-Bashforth method alone produced results that were inferior to those of Runge-Kutta. However, these approximations to $y(0.8)$and $y(1.0)$ are accurate to within

$|2.1272295 - 2.1272056| = 2.39 \times 10^{-5}$
and $|2.6408286 - 2.6408591| = 3.05 \times 10^{-5}$.

respectively, compared to those of Runge-Kutta, which were accurate, respectively, to within

$|2.1272027 - 2.1272892| = 2.69 \times 10^{-5}$
and $|2.6408227 - 2.6408591| = 3.64 \times 10^{-5}$.

The remaining predictor-corrector approximations were generated using Algorithm 5.4 and are shown in Table (10)

x_i	$y_i = y(x_i)$	w_i	$\lvert y_i - w_i \rvert$
0.0	0.5000000	0.5000000	0
0.2	0.8292986	0.8292933	0.0000053
0.4	1.2140877	1.2140762	0.0000114
0.6	1.6489406	1.6489220	0.0000186
0.8	2.1272295	2.1272056	0.0000239
1.0	2.6408591	2.6408286	0.0000305
1.2	3.1799415	3.1799026	0.0000389
1.4	3.7324000	3.7323505	0.0000495
1.6	4.2834838	4.2834208	0.0000630
1.8	4.8151763	4.8150964	0.0000799
2.0	5.3054720	5.3053707	0.0001013

Table (10)

Stability

A number of methods have been presented in this chapter for approximating the solution to an initial-value problem. Although numerous other techniques are available, we have chosen the methods described here because they generally satisfied three criteria:

- Their development is clear enough so that you can understand how and why they work.
- One or more of the methods will give satisfactory results for most of the problems that are encountered by students in science and engineering.
- Most of the more advanced and complex techniques are based on one or a combination of the procedures described here.

One-Step Methods

In this section, we discuss why these methods are expected to give satisfactory results when some similar methods do not. Before we begin this discussion, we need to present two definitions concerned with the convergence of one-step difference-equation methods to the solution of the differential equation as the step size decreases.

Definition (6)5.18

A one-step difference-equation method with local truncation error $\tau_i(h)$ at the *i*th step is said to be **consistent** with the differential equation it approximates if

$$\lim_{h\to 0} \ \underset{1\le i\le N}{Max} \ |\tau_i(h)| = 0.$$

Note that this definition is a *local* definition since, for each of the values $\tau_i(h)$, we are assuming that the approximation w_{i-1} and the exact solution $y(x_{i-1})$ are the same. A more realistic means of analyzing the effects of making *h* small is to determine the *global* effect of the method. This is the maximum error of the method over the entire range of the approximation, assuming only that the method gives the exact result at the initial value.

Definition (7)

A one-step difference-equation method is said to be **convergent** with respect to the differential equation it approximates if

$$\lim_{h\to 0} \ \underset{1\le i\le N}{Max} \ |w_i - y(x_i)| = 0.$$

where $y(x_i)$ denotes the exact value of the solution of the differential equation and w_i is the approximation obtained from the difference method at the *i*th step.

A method is convergent if the solution to the difference equation approaches the solution to the differential equation as the step size goes to zero.

Example 13
Show that Euler's method is convergent.
Solution
the error-bound formula for Euler's method, is

$$\underset{1\le k\le N}{Max} |y_k - w_k| \le \frac{hM}{2L}(e^{L(b-a)} - 1)$$

However, M, L, a, and b are all constants and

$$\lim_{h\to 0} \underset{1\le k\le N}{Max} |y_k - w_k| \le \lim_{h\to 0} \frac{hM}{2L}(e^{L(b-a)} - 1) = 0$$

So Euler's method is convergent with respect to a differential equation satisfying the conditions of this definition. The rate of convergence is $O(h)$. A consistent one-step method has the property that the difference equation for the method approaches the differential equation when the step size goes to zero. So the local truncation error of a consistent method approaches zero as the step size approaches zero. The other error-bound type of problem that exists when using difference methods to approximate solutions to differential equations is a consequence of not using exact results. In practice, neither the initial conditions nor the arithmetic that is subsequently performed is represented exactly because of the round-off error associated with finite-digit arithmetic.

Theorem (5):
Suppose the initial-value problem

$$y' = f(x,y), a \le x \le b, y(a) = \alpha,$$

is approximated by a one-step difference method in the form

$$w_0 = \alpha, \qquad w_{i+1} = w_i + h\varphi(x_i, w_i, h).$$

Suppose also that a number $h_0 > 0$ exists and that $\varphi(x,w,h)$ is continuous and satisfies a Lipschitz condition in the variable w with Lipschitz constant L on the set

$$D = \{(x,w,h) \mid a \le x \le b \text{ and } -\infty < w < \infty, 0 \le h \le h_0\}.$$

Then
(i) The method is stable;
(ii) The difference method is convergent if and only if it is consistent, which is equivalent to $\varphi(x,y,0) = f(x,y), for\ all\ a \le x \le b$;
(iii) If a function τ exists and, for each $i = 1, 2, \ldots, N$, the local truncation error $\tau_i(h)$ satisfies $|\tau_i(h)| \le \tau(h)$ whenever $0 \le h \le h_0$, then

$$|y(x_i) - w_i| \le \frac{\tau(h)}{L} e^{L(x_i - a)}.$$

Example 14
The Modified Euler method is given by

$$w_0 = \alpha,$$

$$w_{i+1} = w_i + \frac{h}{2}\left[f(x_i, w_i) + f(x_{i+1}, w_i + hf(x_i, w_i)\right],$$

for $i = 0, 1, \ldots, N-1$.

Verify that this method is stable by showing that it satisfies the hypothesis of Theorem **(5)**.

Solution

For this method,

$$\varphi = \frac{1}{2}\left[f(x_i, w_i) + f(x_{i+1}, w_i + hf\ (x_i, w_i)\right]$$

If f satisfies a Lipschitz condition on $\{(x, w) \mid a \le x \le b \ and\ -\infty < w < \infty\}$ in the variable w with constant L, then, since

$$\varphi(t, w, h) - \varphi(t, \overline{w}, h)$$
$$= \frac{1}{2}\ f(x, w) + f(x + h, w + hf\ (x, w)\ - \frac{1}{2}\left[f(x, \overline{w}) + f(x + h, \overline{w} + hf\ (x, \overline{w})\right]$$

the Lipschitz condition on f leads to

$$\left|\varphi(t, w, h) - \varphi(t, \overline{w}, h)\right|$$
$$= \left|\frac{1}{2}\ f(x, w) + f(x + h, w + hf\ (x, w)\ - \frac{1}{2}\left[f(x, \overline{w}) + f(x + h, \overline{w} + hf\ (x, \overline{w})\right]\right|$$
$$= \left|\frac{1}{2}\right|f(x, w) - f(x, \overline{w})\left| + \frac{1}{2}\right|f(x + h, w + hf\ (x, w) - f(x + h, \overline{w} + hf\ (x, \overline{w})\Big\|$$
$$= \frac{1}{2}L\left|w - \overline{w}\right| + \frac{1}{2}L\left|w + hf\ (x, w) - \overline{w} + hf\ (x, \overline{w})\right|$$
$$= \frac{1}{2}L\left|w - \overline{w}\right| + \frac{1}{2}L\left|w - \overline{w}\right| + \frac{1}{2}L\left|hf\ (x, w) - hf\ (x, \overline{w})\right|$$
$$= \frac{1}{2}L\left|w - \overline{w}\right| + \frac{1}{2}L\left|w - \overline{w}\right| + \frac{1}{2}hL^2\left|w - \overline{w})\right|$$
$$= L\left|w - \overline{w}\right| + \frac{1}{2}hL^2\left|w - \overline{w})\right| = \left(L + \frac{1}{2}hL^2\right)\left|w - \overline{w})\right|$$

Therefore, φ satisfies a Lipschitz condition in w on the set

$$D = \{(x, w, h) \mid a \le x \le b \ and\ -\infty < w < \infty, 0 \le h \le h_0\}.$$

for any $h_0 > 0$with constant $L' = L + \frac{1}{2}hL^2$.

Finally, if f is continuous on $\{(x, w) \mid a \le x \le b \ and\ -\infty < w < \infty\}$ then φ is continuous on

$$D = \{(x, w, h) \mid a \le x \le b \ and\ -\infty < w < \infty, 0 \le h \le h_0\}.$$

so Theorem (5) implies that the Modified Euler method is stable. Letting h = 0, we have

$$\varphi(t, w,\ 0) = \frac{1}{2}f(x, w) + \frac{1}{2}f(x + 0, w + 0\ \ f\ (x, w)) = f\ (x, w),$$

so the consistency condition expressed in Theorem (5), part (ii), holds. Thus, the method is convergent. Moreover, we have seen that for this method the local truncation error is $O(h^2)$, so the convergence of the Modified Euler method is also $O(h^2)$.

Multistep Methods

For multistep methods, the problems involved with consistency, convergence, and stability are compounded because of the number of approximations involved at each step. In the one-step methods, the approximation w_{i+1} depends directly only on the previous approximation w_i, whereas the multistep methods use at least two of the previous approximations, and the usual methods that are employed involve more.

The general multistep method for approximating the solution to the initial-value problem

$$y' = f(x, y), a \leq x \leq b, y(a) = \alpha, \quad (28)$$

has the form

$$w_0 = \alpha, w_1 = \alpha_1, \ldots, w_{m-1} = \alpha_{m-1},$$

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m}$$
$$+ hF\left(x_i, h, w_{i+1}, w_i, \ldots, w_{i+1-m}\right), \quad (29)$$

for each $i = m - 1, m, \ldots, N - 1$, where $a_{m-1}, a_{m-2}, \ldots, a_0$ are constants and, as usual, $h = (b - a)/N$ and $x_i = a + ih$. The local truncation error for a multistep method expressed in this form is

$$\tau_{i+1}(h) = \frac{y(x_{i+1}) - a_{m-1}y(x_i) - \ldots - a_0 y(x_{i+1-m})}{h}$$
$$- F(x_i, h, y(x_{i+1}), y(x_i), \ldots, y(x_{i+1-m})),$$

for each $i = m - 1, m, \ldots, N - 1$. As in the one-step methods, the local truncation error measures how the solution y to the differential equation fails to satisfy the difference equation.

For the four-step Adams-Bashforth method,

The local truncation error is

$$\tau_{i+1} = \frac{251h^4}{720} f^{(4)}(\mu_i, y(\mu_i)) \quad \text{for some} \quad \mu_i \in \left(x_{i-3}, x_{i+1}\right)$$

whereas the three-step Adams-Moulton method has local truncation error is

$$\tau_{i+1} = \frac{-19h^4}{720} f^{(4)}(\mu_i, y(\mu_i)) \quad \text{for some} \quad \mu_i \in \left(x_{i-2}, x_{i+1}\right)$$

provided, of course, that $y \in C5[a, b]$.

Throughout the analysis, two assumptions will be made concerning the function F:

- If $f \equiv 0$ (that is, if the differential equation is homogeneous), then $F \equiv 0$ also.
- F satisfies a Lipschitz condition with respect to $\{w_j\}$, in the sense that a constant L exists and,

for every pair of sequences $\{v_j\}_{j=0}^{N}$ and $\{v_j\}_{j=0}^{N}$
and for $i = m - 1, m, \ldots, N - 1$, we have

$$|F(x_i,h,v_{i+1},...,v_{i+1-m}) - F(x_i,h,v_{i+1},...,v_{i+1-m})| \leq L \sum_{j=0}^{m} |v_{i+1-j} - v_{i+1-j}|.$$

The explicit Adams-Bashforth and implicit Adams-Moulton methods satisfy both of these conditions, provided *f* satisfies a Lipschitz condition.
The concept of convergence for multistep methods is the same as that for one-step methods.
• A multistep method is **convergent** if the solution to the difference equation approaches the solution to the differential equation as the step size approaches zero. This means that

$$\lim_{h\to\infty} \max_{0\leq i\leq N} |w_i - y(x_i)| = 0 .$$

For consistency, however, a slightly different situation occurs. Again, we want a multistep method to be consistent provided that the difference equation approaches the differential equation as the step size approaches zero; that is, the local truncation error approaches zero at each step as the step size approaches zero. The additional condition occurs because of the number of starting values required for multistep methods. Since usually only the first starting value, *w*0 = *α*, is exact, we need to require that the errors in all the starting values $\{\alpha_i\}$ approach zero as the step size approaches zero. So

$$\lim_{h\to 0} |\tau_i| = 0 \qquad \text{for all } i = m, m+1, ..., N \qquad (30)$$

And $\lim_{h\to 0} |\alpha_i - y(x_i)| = 0$ for all $i = 1, 2,, m-1$ (31)

must be true for a multistep method in the form (5.55) to be **consistent**. Note that (31) implies that a multistep method will not be consistent unless the one-step method generating the starting values is also consistent.
The following theorem for multistep methods is similar to Theorem (5), part (iii), and gives a relationship between the local truncation error and global error of a multistep method. It provides the theoretical justification for attempting to control global error by controlling local truncation error. 8.

Theorem (6):
Suppose the initial-value problem

$$y' = f(x,y), a \leq x \leq b, y(a) = \alpha,$$

is approximated by an explicit Adams predictor-corrector method with an *m*-step Adams-Bashforth predictor equation

$$w_{i+1} = w_i + h[b_{m-1} f(x_i, w_i) + ... + b_0 f(x_{i+1-m}, w_{i+1-m})],$$

With local truncation error $\tau_{i+1}(h)$, and an $(m - 1)$ −step implicit Adams-Moulton corrector equation

$$w_{i+1} = w_i + h[b_{m-1} f(x_i, w_{i+1}) + b_{m-2} f(x_i, w_i)$$

$$+ \ldots + b_0 f(x_{i+2-m}, w_{i+2-m})]$$

with local truncation error $\tau_{i+1}(h)$,. In addition, suppose that $f(x, y)$and$f_y(x, y)$ are continuous

on $D = \{(x, y) \mid a \leq x \leq b \; and \; -\infty < y < \infty\}$

and that $f_y(x, y)$ is bounded. Then the local truncation error $\sigma_{i+1}(h)$ of the predictor-corrector method is

$$\sigma_{i+1}(h) = \tau_{i+1}(h) + \tau_{i+1}(h) b_{m-1} \frac{\partial f(x_{i+1}, \theta_{i+1})}{\partial y},$$

Where θ_{i+1} is a number between zero and $h\tau_{i+1}(h)$.

Moreover, there exist constants$k_1$and k_2 such that

$$|w_i - y(x_i)| \leq \left[\underset{0 \leq j \leq m-1}{Max} \left| w_j - y(x_j) \right| + k_1 \sigma(h) \right] e^{k_2 (x_i - a)},$$

Where $\sigma(h) = \underset{\leq j \leq N}{Max} |\sigma_j(h)|$.

Before discussing connections between consistency, convergence, and stability for multistep methods, we need to consider in more detail the difference equation for a multistep method. In doing so, we will discover the reason for choosing the Adams methods as our standard multistep methods.

Associated with the difference equation (29) given at the beginning of this discussion,

$$w_0 = \alpha, w_1 = \alpha_1, \ldots, w_{m-1} = \alpha_{m-1},$$

$$w_{i+1} = a_{m-1} w_i + a_{m-2} w_{i-1} + \cdots + a_0 w_{i+1-m}$$

$$+ hF\ x_i, h, w_{i+1}, w_i, \ldots, w_{i+1-m}\ ,$$

is a polynomial, called the **characteristic polynomial** of the method, given by

$$P\ \lambda\ = \lambda^m - a_{m-1} \lambda^{m-1} - a_{m-2} \lambda^{m-2} - \ldots - a_1 \lambda - a_0. \qquad (32)$$

The stability of a multistep method with respect to round-off error is dictated the by magnitudes of the zeros of the characteristic polynomial. To see this, consider applying the standard multistep method (5.55) to the trivial initial-value problem

$$y' \equiv 0, y(a) = \alpha, where \; \alpha \neq 0. \qquad (33)$$

This problem has exact solution $y(x) \equiv \alpha$. We can see that any multistep method will, in theory, produce the exact solution $w_n = \alpha$ for all n. The only deviation from the exact solution is due to the round-off error of the method. The right side of the differential equation in (33) has $f(x,y) \equiv 0$, so by assumption (1), we have $F(x_i, h, w_{i+1}, w_{i+2}, \ldots, w_{i+1-m}) = 0$
in the difference equation (29). As a consequence, the standard form of the difference equation becomes

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \ldots + a_0 w_{i+1-m}. \qquad (34)$$

Suppose λ is one of the zeros of the characteristic polynomial associated with (29). Then $w_n = \lambda_n$ for each n is a solution to (33) since

$$\lambda^{i+1} - a_{m-1}\lambda^i - a_{m-2}\lambda^{i-1} - \ldots - a_0\lambda^{i+1-m} = \lambda^{i+1-m}[\lambda^m - a_{m-1}\lambda^{m-1} - \ldots - a_0] = 0 .$$

In fact, if $\lambda_1, \lambda_2, \ldots, \lambda_m$ are distinct zeros of the characteristic polynomial for (29), it can be shown that *every* solution to (34) can be expressed in the form

$$w_n = \sum_{i=1}^{m} c_i \lambda_i^n , \qquad (35)$$

for some unique collection of constants $c_1, c_2, \ldots, c_m$.
Since the exact solution to (33) is $y(t) = \alpha$, the choice $w_n = \alpha_n$, for all n, is a solution to (34). Using this fact in (34) gives
$0 = \alpha - \alpha a_{m-1} - \alpha a_{m-2} - \ldots - \alpha a_0 = \alpha[1 - a_{m-1} - a_{m-2} - \ldots - a_0]$. This implies that $\lambda = 1$ is one of the zeros of the characteristic polynomial (32). We will assume that in the representation (35) this solution is described by $\lambda_1 = 1$ and $c_1 = \alpha$, so
all solutions to (33) are expressed as

$$w_n = \alpha + \sum_{i=2}^{m} c_i \lambda_i^{\,n} . \qquad (36)$$

If all the calculations were exact, all the constants $c_2, c_3, \ldots, c_m$ would be zero. In practice, the constants $c_2, c_3, \ldots, c_m$ are not zero due to round-off error. In fact, the round-off error grows exponentially unless $|\lambda_i| \le 1|$ for each of the roots $\lambda_2, \lambda_3, \ldots, \lambda_m$. The smaller the magnitude of these roots, the more stable the method with respect to the growth of round-off error. In deriving (36), we made the simplifying assumption that the zeros of the characteristic polynomial are distinct. The situation is similar when multiple zeros occur. For example, if $\lambda_k = \lambda_{k+1} = \ldots = \lambda_{k+p}$ for some k and p, it simply requires replacing the sum

$$c_k \lambda_k^n + c_{k+1}\lambda_{k+1}^n + \ldots + c_{k+p}\lambda_{k+p}^n$$

In (36) with

$$c_k\lambda_k^n + c_{k+1}n\lambda_k^{n-1} + c_{k+2}n(n-1)\lambda_k^{n-2} + ... + c_{k+p}\ n(n-1)...(n-p+1)\ \lambda_k^{n-p}.\quad(37)$$

Although the form of the solution is modified, the round-off error if $|\lambda_k| > 1$ still grows exponentially.

Although we have considered only the special case of approximating initial-value problems of the form (33), the stability characteristics for this equation determine the stability for the situation when $f(x, y)$ is not identically zero. This is because the solution to the homogeneous equation (33) is embedded in the solution to any equation. The following definitions are motivated by this discussion.

Definition (8)

Let $\lambda_1, \lambda_2, \ldots, \lambda_m$ denote the (not necessarily distinct) roots of the characteristic equation

$$P(\lambda) = \lambda^m - a_{m-1}\lambda^{m-1} - a_{m-2}\lambda^{m-2} - ... - a_1\lambda - a_0 = 0$$

Associated with the multistep difference method

$$w_0 = \alpha, w_1 = \alpha_1, \ldots, w_{m-1} = \alpha_{m-1}$$

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} + hF(x_i, h, w_{i+1}, w_i, \ldots, w_{i+1-m}).$$

If $|\lambda_i| \leq 1$, for each $i = 1, 2, \ldots, m,$ and all roots with absolute value 1 are simple roots, then the difference method is said to satisfy the **root condition**.

Definition (9)

(i) Methods that satisfy the root condition and have $\lambda = 1$ as the only root of the characteristic equation with magnitude one are called **strongly stable**.

(ii) Methods that satisfy the root condition and have more than one distinct root with magnitude one are called **weakly stable**.

(iii) Methods that do not satisfy the root condition are called **unstable**.

Consistency and convergence of a multistep method are closely related to the round-off stability of the method. The next theorem details these connections. For the proof of this result and the theory on which it is based.

Theorem (7)

A multistep method of the form

$$w_0 = \alpha, w_1 = \alpha_1, \ldots, w_{m-1} = \alpha_{m-1}$$

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} + hF(x_i, h, w_{i+1}, w_i, \ldots, w_{i+1-m}).$$

is stable if and only if it satisfies the root condition. Moreover, if the difference method is consistent with the differential equation, then the method is stable if and only if it is convergent

Example 15

The fourth-order Adams-Bashforth method can be expressed as

$w_{i+1} = w_i + hF(x_i, h, w_{i+1}, w_i, ..., w_{i-3}),$

where

$F(x_i, h, w_{i+1}, w_i, ..., w_{i-3})$

$$= \frac{h}{24}[55f(x_i, w_i) - 59f(x_{i-1}, w_{i-1}) + 37f(x_{i-2}, w_{i-2}) - 9f(x_{i-3}, w_{i-3})];$$

Show that this method is strongly stable.

Solution

In this case we have $m = 4$, $a_0 = 0, a_1 = 0, a_2 = 0$, and $a_3 = 1$, so the characteristic equation for this Adams-Bashforth method is

$0 = P(\lambda) = \lambda^4 - \lambda^3 = \lambda^3(\lambda - 1)$. This polynomial has roots $\lambda_1 = 1, \lambda_2 = 0, \lambda_3 = 0$, and $\lambda_4 = 0$. Hence it satisfies the root condition and is strongly stable.

The Adams-Moulton method has a similar characteristic polynomial, with zeros $\lambda_1 = 1, \lambda_2 = 0,$ and $\lambda_3 = 0$, and is also strongly stable.

Example 16

Show that the fourth-order Milne's method, the explicit multistep method given by

$$w_{i+1} = w_{i-3} + \frac{4h}{3}\left[2f(x_i, wi) - f(x_{i-1}, w_{i-1}) + 2f(x_{i-2}, w_{i-2})\right]$$

Satisfies the root condition, but it is only weakly stable.

Solution

The characteristic equation for this method, $0 = P(\lambda) = \lambda^4 - 1$, has four roots with magnitude one: $\lambda_1 = 1, \lambda_2 = -1, \lambda_3 = i$, and $\lambda_4 = -i$. Because all the roots have magnitude 1, the method satisfies the root condition. However, there are multiple roots with magnitude 1, so the method is only weakly stable.

Example 17

Apply the strongly stable fourth-order Adams-Bashforth method and the weakly stable Milne's method with $h = 0.1$ to the initial-value problem

$y' = -6y + 6,\ 0 \le x \le 1,\quad y(0) = 2$, which has the exact solution $y(x) = 1 + e^{-6x}$.

Solution

The results in Table (11) show the effects of a weakly stable method versus a strongly stable method for this problem.

x_i	Exact $y(x_i)$	Adams-Bashforth Method w_i	Error $\|y_i - w_i\|$	Milne's Method w_i	Error $\|y_i - w_i\|$
0.1	1.5488116				
0.2	1.3011942				
0.3	1.1652989				
0.4	1.0907180	1.0996236	8.906 × 10−3	1.0983785	7.661 × 10−3
0.5	1.0497871	1.0513350	1.548 × 10−3	1.0417344	8.053 × 10−3
0.6	1.0273237	1.0425614	1.524 × 10−2	1.0486438	2.132 × 10−2
0.7	1.0149956	1.0047990	1.020 × 10−2	0.9634506	5.154 × 10−2
0.8	1.0082297	1.0359090	2.768 × 10−2	1.1289977	1.208 × 10−1
0.9	1.0045166	0.9657936	3.872 × 10−2	0.7282684	2.762 × 10−1
1.0	1.0024788	1.0709304	6.845 × 10−2	1.6450917	6.426 × 10−1

Table (11)

The reason for choosing the Adams-Bashforth-Moulton as our standard fourth-order predictor-corrector technique over the Milne-Simpson method of the same order is that both the Adams-Bashforth and Adams-Moulton methods are strongly stable.
They are more likely to give accurate approximations to a wider class of problems than is the predictor-corrector based on the Milne and Simpson techniques, both of which are weakly stable.

EXERCISE (6)

(1) Consider the differential equation $y' = f(x,y), a \le x \le b, y(a) = \alpha.$

(a) Show that $y'(x_i) = \dfrac{-3y(x_i) + 4y(x_{i+1}) - y(x_{i+2})}{2h} + \dfrac{h^2}{3} y''' \xi_1$ for some ξ, where $x_i < \xi_i < x_{i+2}$.

(b) Part (a) suggests the difference method $w_{i+2} = 4w_{i+1} - 3w_i - 2hf(x_i, w_i)$, for $i = 0, 1, \ldots, N-2$. Use this method to solve $y' = 1 - y$, $0 \le x \le 1$, $y(0) = 0$, with $h = 0.1$. Use the starting value $w_0 = 0$ and $w_1 = y(x_1) = 1 - e^{-0.1}$.

(c) Repeat part (b) with $h = 0.01$ and $w_1 = y(x_1) = 1 - e^{-0.01}$.

(d) Analyze this method for consistency, stability, and convergence.

(2) Given the multistep method

$$w_{i+1} = \frac{-3}{2}w_i + 3w_{i-1} - \frac{1}{2}w_{i-2} + 3hf(x_i, w_i), \text{ for } i = 2, \ldots, N-1,$$

with starting values w_0, w_1, w_2 :

(a) Find the local truncation error.

(b) Comment on consistency, stability, and convergence.

(3) Investigate stability for the difference method
$w_{i+1} = -4w_i + 5w_{i-1} + 2h[f(x_i, w_i) + 2hf(x_{i-1}, w_{i-1})],$
for $i = 1, 2, \ldots, N-1,$ with starting values $w_0, w_1.$

Multistep Methods

In previous sections we have discussed numerical procedures for approximating the solution of the initial value problem

$$y' = f(x, y), y(t_0) = y_0, \qquad (1)$$

in which data at the point $x = x_n$ are used to calculate an approximate value of the solution $y(x_{n+1})$ at the next mesh point $x = x_{n+1}$. In other words, the calculated value of *y* at any mesh point depends only on the data at the preceding mesh point. Such methods are called **one-step methods**. However, once approximate values of the solution $y(x_n)$ have been obtained at a few points beyond x_0, it is natural to ask whether we can make use of some of this information, rather than just the value at the last point, x_0 calculate the value of y at the next point. Specifically, if y_1 at x_1, y_2 at $x_2, \ldots, y_n$ at x_n are known, how can we use this information to determine y_{n+1} at x_{n+1}? Methods that use information at more than the last mesh point are referred to as **multistep methods**. In this section we will describe two types of multistep methods:
Adams methods and backward differentiation formulas. Within each type, we can achieve various levels of accuracy, depending on the number of preceding data points that are used. For simplicity, we will assume throughout our discussion that the step size *h* is constant.

<u>Adams Methods</u>.
Integrate (1) in the interval $[x_n, x_{n+1}]$ we have

$$y_{n+1} - y_n = \int_{x_n}^{x_{n+1}} f(x_n, y_n)dx$$

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(x_n, y_n)dx \qquad (2)$$

Where y_n is the approximate solution of the initial value problem (1) at the point x_n. The basic idea of an Adams method is to approximate $f(x_n, y_n)$ by a polynomial $P_m(x)$ of degree *m* and to use the polynomial to evaluate the integral on the right side of equation (2).

Explicit schemes

Two-step Adams-Bashforth method

We derive the two-step Adams-Bashforth method,

$$y_{n+1} = y_n + h[b_1 f(x_n, y_n) + b_2 f(x_{n-1}, y_{n-1}))$$

The constants b_1, and b_2, are obtained by evaluating the integral from x_n to x_{n+1} of a polynomial $P_1(x)$ that passes through $f(x_n, y_n)$ and $f(x_{n-1}, y_{n-1})$

Because we can write

$$P_1(x_n, y_n) = f(x_n, y_n)L_0 + f(x_{n-1}, y_{n-1})L_1 + \frac{f''(\lambda_n, y(\lambda_n))}{2!}(x - x_n)(x - x_{n-1})$$

Where

$E = \frac{f''(\lambda_n, y(\lambda_n))}{2!}(x - x_n)(x - x_{n-1})$ is the truncation error and

$$L_0 = \frac{(x - x_{n-1})}{(x_n - x_{n-1})}, \quad L_1 = \frac{(x - x_n)}{(x_{n-1} - x_n)}$$

is the Lagrange polynomials for the interpolation points x_n and x_{n-1}, and because our final method expresses y_{n+1} as a linear combination of y_n and values of *f*, it follows that the constants b_1 and b_2, are the integrals of the Lagrange polynomials from x_n to x_{n+1} divided by h.

So equation (2) becomes

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} \left[f(x_n, y_n)\frac{(x - x_{n-1})}{(x_n - x_{n-1})} + f(x_{n-1}, y_{n-1})\frac{(x - x_n)}{(x_{n-1} - x_n)} \right] dx$$

$$y_{n+1} = y_n + f(x_n, y_n)\int_{x_n}^{x_{n+1}} \frac{(x - x_{n-1})}{(x_n - x_{n-1})}dx + f(x_{n-1}, y_{n-1})\int_{x_n}^{x_{n+1}} \frac{(x - x_n)}{(x_{n-1} - x_n)}dx$$

Put

$$x = x_n + sh$$

then $x - x_n = sh$ and $dx = hds$

$$x - x_{n-1} = (x - x_n) + (x_n - x_{n-1}) = sh + h = h(s+1)$$

$$y_{n+1} = y_n + hf(x_n, y_n)\int_{x_n}^{x_{n+1}} \frac{h(s+1)}{h}ds + hf(x_{n-1}, y_{n-1})\int_{x_n}^{x_{n+1}} \frac{sh}{-h}ds$$

$$y_{n+1} = y_n + hf(x_n, y_n)\int_0^1 (s+1)ds - hf(x_{n-1}, y_{n-1})\int_0^1 sds$$

$$y_{n+1} = y_n + h\left[f(x_n, y_n)\frac{(s+1)^2}{2}ds - f(x_{n-1}, y_{n-1})\frac{s^2}{2} \right]_0^1$$

$$y_{n+1} = y_n + h\left[\tfrac{3}{2}f(x_n, y_n) - \tfrac{1}{2}f(x_{n-1}, y_{n-1})\right]$$

We conclude that the two-step Adams-Bashforth method is

$$\boxed{y_{n+1} = y_n + \tfrac{h}{2}\; 3f(x_n, y_n) - f(x_{n-1}, y_{n-1})} \qquad (3)$$

Finally we determine the local error by the equation

$$E_\ell = \frac{f''(\lambda_n, y(\lambda_n))}{2!}\int_{x_k}^{x_{k+1}} (x - x_n)(x - x_{n-1})$$

$$= \frac{f''(\lambda_n, y(\lambda_n))}{2!}\int_0^1 h^3(s)(s+1)ds = \frac{5h^3}{12}f''(\lambda_n, y(\lambda_n))$$

If we use trapezoidal integral by using the points $(x_n, y_n), (x_{n+1}, y_{n+1})$ which in fact interpolation polynomial interpolate the function $f(x_n, y_n)$ at $(x_n, y_n), (x_{n+1}, y_{n+1})$ equation (2) becomes

$$\boxed{y_{n+1} = y_n + \tfrac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, y_{n+1})\right]} \qquad (4)$$

Predictor-corrector scheme

When an explicit scheme is combined with an implicit scheme in this manner, we have the so called *predictor-corrector* scheme. The equation (3) is *predictor* calculation to equation (4)

$$y_{n+1}^{\;P} = y_n + \tfrac{h}{2}\; 3f(x_n, y_n) - f(x_{n-1}, y_{n-1}) \qquad (5)$$

$$y_{n+1}^{\;C} = y_n + \tfrac{h}{2}\; f(x_n, y_n) + f(x_{n+1},\; y_{n+1}^{\;P}) \qquad (6)$$

Three-step Adams-Bashforth method

We derive the three-step Adams-Bashforth method,

$$y_{n+1} = y_n + h[b_1 f(x_n, y_n) + b_2 f(x_{n-1}, y_{n-1}) + b_3 f(x_{n-2}, y_{n-2})):$$

The constants b_1, b_2 and b_3, are obtained by evaluating the integral from x_n to x_{n+1} of a polynomial $P_2(x)$ that passes through $f(x_n, y_n), f(x_{n-1}, y_{n-1})$ and $f(x_{n-2}, y_{n-2})$. Because we can write

$$P_2(x_n,y_n)=f(x_n,y_n)L_0+f(x_{n-1},y_{n-1})L_1+f(x_{n-2},y_{n-2})L_2$$

$$+\frac{f'''(x_n,y_n)}{3!\,h^3}(x-x_n)(x-x_{n-1})(x-x_{n-2})$$

Where

$$L_0=\frac{(x-x_{n-1})(x-x_{n-2})}{(x_n-x_{n-1})(x_n-x_{n-2})},$$

$$L_1=\frac{(x-x_n)(x-x_{n-2})}{(x_{n-1}-x_n)(x_{n-1}-x_{n-2})},$$

$$L_2=\frac{(x-x_n)(x-x_{n-1})}{(x_{n-2}-x_n)(x_{n-2}-x_{n-1})}$$

is the Lagrange polynomials for the interpolation points x_n, x_{n-1} and x_{n-2}, and because our final method expresses y_{n+1} as a linear combination of y_n and values of f, it follows that the constants b_1, b_2 and b_3, are the integrals of the Lagrange polynomials from x_n to x_{n+1} divided by h.

Equation (2) becomes

$$y_{n+1}=y_n+\int_{x_n}^{x_{n+1}} f(x_n,y_n)\frac{(x-x_{n-1})(x-x_{n-2})}{(x_n-x_{n-1})(x_n-x_{n-2})}dx$$

$$+\int_{x_n}^{x_{n+1}} f(x_{n-1},y_{n-1})\frac{(x-x_n)(x-x_{n-2})}{(x_{n-1}-x_n)(x_{n-1}-x_{n-2})}dx$$

$$+\int_{x_n}^{x_{n+1}} f(x_{n-2},y_{n-2})\frac{(x-x_n)(x-x_{n-1})}{(x_{n-2}-x_n)(x_{n-2}-x_{n-1})}dx$$

Put

$$x=x_n+sh$$

then $x-x_n=sh$ and $dx=hds$

$$x-x_{n-1}=(x-x_n)+(x_n-x_{n-1})=sh+h=h(s+1)$$

$$x - x_{n-2} = (x - x_n) + (x_n - x_{n-2}) = sh + 2h = h(s+2)$$

$$y_{n+1} = y_n + hf(x_n, y_n)\int_0^1 \frac{(s+1)(s+2)}{(1)(2)}ds$$

$$+ hf(x_{n-1}, y_{n-1})\int_0^1 \frac{(s)(s+2)}{(-1)(-1)}ds + hf(x_{n-2}, y_{n-2})\int_0^1 \frac{(s)(s+1)}{(-2)(-1)}ds$$

We conclude that the three-step Adams-Bashforth method is

$$\boxed{y_{n+1} = y_n + \frac{h}{12}\, 23f(x_n, y_n) - 16f(x_{n-1}, y_{n-1}) + 5f(x_{n-2}, y_{n-2})} \qquad (6)$$

Finally wee determine the local error by the equation

$$E_\ell = \frac{f'''(x_n, y_n)}{3!}\int_{x_k}^{x_{k+1}} (x - x_n)(x - x_{n-1})(x - x_{n-2})$$

$$= \frac{f'''(\lambda_n, y(\lambda_n))}{3!}\int_0^1 h^4 (s)(s+1)(s+2)ds = \frac{3}{8}h^4 f^{(3)}(\mu_i, y(\mu_i))$$

If we use Simpson integral by integrate the equation (1) on the interval $\left[x_{n-1}, x_{n+1}\right]$ which in fact interpolation polynomial interpolate the function $f(x_n, y_n)$ at three points $(x_{n-1}, y_{n-1}), (x_n, y_n)$ and (x_{n+1}, y_{n+1}) equation (2) becomes

$$\boxed{y_{n+1} = y_{n-1} + \frac{h}{3}\left[f(x_{n-1}, y_{n-1}) + 2f(x_n, y_n) + f(x_{n+1}, y_{n+1})\right]} \qquad (7)$$

Adams-Moulton (AM) Methods

The same approach can be used to derive an implicit Adams method, which is known as an Adams-Moulton method. The only difference is that because x_{n+1} is an interpolation point. Because the resulting interpolating polynomial is of degree one greater than in the explicit case, the error in an $m - step$ Adams-Moulton method is O(h^{m+1}), as opposed to O(h^m) for an $m - step$ Adams-Bashforth method AM methods are implicit methods; in other words, they use information at x_{n+1} to compute y_{n+1} . Let us derive AM2, the second-order Adams-Moulton method.

Again, like the AB method, we will use the Lagrange interpolating polynomial of degree 1, as a linear interpolation. However, instead of fitting the interpolant to f at x_n and x_{n-1}, we will fit to f at x_n and x_{n+1}. Proceeding as for the AB2 case, we have

$$P_1(x_n, y_n) = f(x_{n+1}, y_{n+1})L_0 + f(x_n, y_n)L_1 + \frac{f''(\lambda_n, y(\lambda_n))}{2!}(\mathrm{x} - x_n)(\mathrm{x} - x_{n+1})$$

Where

$E = \frac{f''(\lambda_n, y(\lambda_n))}{2!}(\mathrm{x} - x_n)(\mathrm{x} - x_{n+1})$ is the truncation error and

$L_0 = \frac{(x - x_n)}{(x_{n+1} - x_n)}, \quad L_1 = \frac{(x - x_{n+1})}{(x_n - x_{n+1})}$ is the Lagrange polynomials for the interpolation points x_n and x_{n+1},

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} \left[f(x_{n+1}, y_{n+1}) \frac{(x - x_n)}{(x_{n+1} - x_n)} + f(x_n, y_n) \frac{(x - x_{n+1})}{(x_n - x_{n+1})} \right] dx$$

$$y_{n+1} = y_n + f(x_{n+1}, y_{n+1}) \int_{x_n}^{x_{n+1}} \frac{(x - x_n)}{(x_{n+1} - x_n)} dx + f(x_n, y_n) \int_{x_n}^{x_{n+1}} \frac{(x - x_{n+1})}{(x_n - x_{n+1})} dx$$

Put

$$x = x_n + sh$$

then $x - x_n = sh$ and $dx = hds$

$$x - x_{n+1} = (x - x_n) + (\mathrm{x}_n - x_{n+1}) = sh - h = h(s - 1)$$

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}) \int_{x_n}^{x_{n+1}} \frac{h(s)}{h} ds + hf(x_n, y_n) \int_{x_n}^{x_{n+1}} \frac{(s-1)h}{-h} ds$$

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}) \int_0^1 s\, ds - hf(x_n, y_n) \int_0^1 (1 - s) ds$$

$$y_{n+1} = y_n + h\left[f(x_{n+1}, y_{n+1}) \frac{(s)^2}{2} ds - f(x_n, y_n) \frac{(1 - s^2)}{2} \right]_0^1$$

$$y_{n+1} = y_n + h\left[\tfrac{1}{2} f(x_{n+1}, y_{n+1}) + \tfrac{1}{2} f(x_n, y_n) \right]$$

We conclude that the one-step Adams-Moulton method is

$$\boxed{y_{n+1} = y_n + \frac{h}{2}\left[f(x_{n+1}, y_{n+1}) + f(x_n, y_n) \right]} \qquad (3)$$

We determine the local error by the equation

$$E_\ell = \frac{f''(\lambda_n, y(\lambda_n))}{2!} \int_{x_k}^{x_{k+1}} (x - x_n)(x - x_{n+1})$$

$$= \frac{f''(\lambda_n, y(\lambda_n))}{2!} \int_0^1 h^3(s)(s-1)ds = \frac{-h^3}{6} f''(\lambda_n, y(\lambda_n)) \ , \quad x_n < \lambda_n < x_{n+1}$$

If we use trapezoidal integral by using the points $(x_n, y_n), (x_{n+1}, y_{n+1})$ which in fact interpolation polynomial interpolate the function $f(x_n, y_n)$ at $(x_n, y_n), (x_{n+1}, y_{n+1})$ equation (2) becomes

$$\boxed{y_{n+1} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, y_{n+1})\right]} \qquad (4)$$

Notice that the order 2 AM method only requires the use of one previous step for the same $O(h^3)$ LTE; again, the global error is $O(h^2)$. In general the s order AM method requires s-1 steps, despite using the same polynomial degree as an s order AB method. This is the benefit of going implicit. Of course, we now have a linear system to solve if we want to compute y_{n+1}

AM3

To derive Adam-Moulton Consider approximating the function $f(x_n, y_n)$. in equation (2) by the following *second* degree Lagrange polynomial from equation (1), when $m = 2$

$$P_2(x_n, y_n) = f(x_{n+1}, y_{n+1})L_0 + f(x_n, y_n)L_1 + f(x_{n-1}, y_{n-1})L_2$$
$$+ \frac{f'''(x_n, y_n)}{3!\,\mathrm{h}^3}(\mathrm{x} - x_{n+1})(\mathrm{x} - x_n)(\mathrm{x} - x_{n-1})$$

Where

$$L_0 = \frac{(x - x_n)(x - x_{n-1})}{(x_{n+1} - x_n)(x_{n+1} - x_{n-1})}, \quad L_1 = \frac{(x - x_{n+1})(x - x_{n-1})}{(x_n - x_{n+1})(x_{n-1} - x_{n-1})} \text{, and}$$

$$L_2 = \frac{(x - x_{n+1})(x - x_n)}{(x_{n-1} - x_{n+1})(x_{n-1} - x_n)}$$

is the Lagrange polynomials for the interpolation points x_{n-1} , x_n andx_{n+1},then equation (2) becomes

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(x_{n+1}, y_{n+1}) \frac{(x - x_n)(x - x_{n-1})}{(x_{n+1} - x_n)(x_{n+1} - x_{n-1})} dx$$

$$+ \int_{x_n}^{x_{n+1}} f(x_n, y_n) \frac{(x - x_{n+1})(x - x_{n-1})}{(x_n - x_{n+1})(x_n - x_{n-1})} dx$$

$$+ \int_{x_n}^{x_{n+1}} f(x_{n-1}, y_{n-1}) \frac{(x - x_{n+1})(x - x_n)}{(x_{n-1} - x_{n+1})(x_{n-1} - x_n)} dx$$

Put

$x = x_n + sh$

then $x - x_n = sh$ and $dx = hds$

$x - x_{n-1} = (x - x_n) + (x_n - x_{n-1}) = sh + h = h(s+1)$

$x - x_{n+1} = (x - x_n) + (x_n - x_{n+1}) = sh - h = h(s-1)$

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}) \int_0^1 \frac{(s)(s+1)}{(1)(2)} ds$$

$$+ hf(x_n, y_n) \int_0^1 \frac{(s-1)(s+1)}{(-1)(1)} ds$$

$$+ hf(x_{n-1}, y_{n-1}) \int_0^1 \frac{(s-1)(s)}{(-2)(-1)} ds$$

We conclude that the Two-Step Adams-Moulton method is

$$\boxed{y_{n+1} = y_n + \frac{h}{12}\left[5f(x_{n+1}, y_{n+1}) + 8f(x_n, y_n) - f(x_{n-1}, y_{n-1})\right]} \qquad (6)$$

We determine the local error by the equation

$$E_\ell = \frac{f'''(x_n, y_n)}{3!} \int_{x_k}^{x_{k+1}} (x - x_{n+1})(x - x_n)(x - x_{n-1}) dx$$

$$= \frac{f'''(\lambda_n, y(\lambda_n))}{3!} \int_0^1 h^4 (s-1)(s)(s+1) ds = \frac{-h^4}{24} f^{(3)}(\mu_i, y(\mu_i))$$

$x_{i-1} < \mu_i < x_{i+1}$

If we use **Simpson integral** by integrate the equation (1) on the interval $\left[x_{n-1}, x_{n+1}\right]$ which in fact interpolation polynomial interpolate the function $f(x_n, y_n)$ at three points $(x_{n-1}, y_{n-1}), (x_n, y_n)$ and (x_{n+1}, y_{n+1}) equation (2) becomes

$$\boxed{y_{n+1} = y_{n-1} + \frac{h}{3}\left[f(x_{n-1}, y_{n-1}) + 2f(x_n, y_n) + f(x_{n+1}, y_{n+1})\right]} \qquad (7)$$

Based on the general form of an AB method, we can now also write the general form of an AM method. We only need to adjust the indices so they go up to x_{n+1} on the interpolating polynomial

$$y_{n+1} = y_n + h\sum_{k=0}^{s-1} B_k f(x_{n+1-k}, y_{n+1-k})$$

Where $B_k = \frac{1}{h}\int_{x_i}^{x_{i+1}} L_k(x)dx$

Example 11

Consider the initial-value problem

$$y' = y - x^2 + 1, 0 \leq x \leq 2, \qquad y(0) = 0.5.$$

Use the exact values given from $y(x) = (x + 1)^2 - 0.5e^x$ as starting values and $h = 0.2$ to compare the approximations from

(c) By the explicit Adams-Bashforth four-step method and

(d) The implicit Adams-Moulton three-step method.

Backward Differentiation Formulas.

Another type of multistep method uses a polynomial $P_m(x)$ to approximate the solution $y(x_n)$) of the initial value problem (1) rather than its derivative $y'(x_n)$, as in the Adams methods. We then differentiate $P_m(x)$ and set $P_m(x_{n+1}))$ equal $f(x_{n+1}, y_{n+1})$ to obtain an implicit formula for y_{n+1}. These are called **backward differentiation formulas**.

The simplest case uses a first degree polynomial $P_1(x) = Ax + B$. The coefficients are chosen to match the computed values of the solution y_n and y_{n+1}. Thus A and B must satisfy

$$\begin{aligned} P_1(x_n) &= Ax_n + B = y_n \\ P_1(x_{n+1}) &= Ax_{n+1} + B = y_{n+1} \end{aligned} \qquad (8)$$

Since $P_1'(x) = A$, the requirement that $P_1'(x_{n+1}) = f(x_{n+1}, y_{n+1})$ is just

$A = f(x_{n+1}, y_{n+1})$

Another expression for A comes from subtracting the first of Equation. (8) from the second, which gives $A = \frac{(y_{n+1} - y_n)}{h}$.

we obtain the first order backward differentiation formula

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}). \qquad (9)$$

Note that Eq. (9) is just the backward Euler formula .

By using higher order polynomials and correspondingly more data points, we can obtain backward differentiation formulas of any order. The second order formula is Or use the three step backward difference formula

$y'(x_{n+1}) = \frac{3y_{n+1} - 4y_n + y_{n-1}}{2h}$ Substitute in the initial value problem (1) then

$$\frac{3y_{n+1} - 4y_n + y_{n-1}}{2h} = f(x_{n+1}, y_{n+1})$$

Arrange the terms we have

$$\boxed{y_{n+1} = \frac{1}{3}\ 4y_n - y_{n-1} + 2hf(x_{n+1}, y_{n+1})}$$

and the fourth order formula is

$$\boxed{y_{n+1} = 125\,[48y_n - 36y_{n-1} + 16y_{n-2} - 3y_{n-3} + 12hf(x_{n+1}, y_{n+1})]}$$

These formulas have local truncation errors proportional to h^3 and h^5, respectively.

A comparison between one-step and multistep methods must take several factors into consideration. The fourth order Runge–Kutta method requires four evaluations of f at each step, while the fourth order Adams–Bashforth method (once past the starting values) requires only one, and the predictor–corrector method only two.

Thus, for a given step size h, the latter two methods may well be considerably faster than Runge–Kutta. However, if Runge–Kutta is more accurate and therefore can use fewer steps, then the difference in speed will be reduced and perhaps eliminated.

The Adams–Moulton and backward differentiation formulas also require that the difficulty in solving the implicit equation at each step be taken into account. All multistep methods have the possible disadvantage that errors in earlier steps can feed back into later calculations with unfavorable consequences. On the other hand, the underlying polynomial approximations in multistep methods make it easy to approximate the solution at points between the mesh points, should this be desirable.

Multistep methods have become popular largely because it is relatively easy to estimate the error at each step and to adjust the order or the step size to control it.

Summary

1. An order s AB method combines f values in $[x_{n+1-s}, x_n]$ to update y_n ;

2. An order s AM method combines f values in $[x_{n+1-s}, x_{n+1}]$ to solve for

y_{n+1} ;

3. An order s BDF method combines y values in $[x_{n+1-s}, x_{n+1}]$, evaluates f at x_{n+1} alone, and solves for y_{n+1}.

our general formula will encompass all the methods in this document, but will account for possibly new multistep methods as well. Here it is:

$$\sum_{k=0}^{s} \alpha_k y_{n+1-k} = \sum_{k=0}^{s} B_k f_{n+1-k}$$

We can see how to recover AB, AM and BDF methods from this formula:

1. For AB methods, $\alpha_k = 0; k > 1$ and $B_0 = 0$
2. For AM methods, $\alpha_k = 0; k > 1$, but $B_0 \neq 0$
3. For BDF methods, $\alpha_k = 0; k > 0$.

We derived several multistep methods using integrals and derivatives of polynomial interpolants, and presented a general formula that covered all the cases discussed in this document. Using the polynomial framework, we were also able to derive estimates for local truncation errors (LTEs) for all these methods. We remarked on global truncation errors for every method.
However, we never discussed the stability of these methods, beyond pointing out that implicit methods may be more stable than explicit ones.

PROBLEMS

In each of Problems 1 through 6 determine an approximate value of the solution at $x = 0.4$ and $x = 0.5$ using the specified method. For starting values use the values given by the Runge–Kutta method; Compare the results of the various methods with each other and with the actual solution (if available).
(a) Use the fourth order predictor–corrector method with h = 0.1. Use the corrector formula once at each step.
(b) Use the fourth order Adams–Moulton method with h = 0.1.
(c) Use the fourth order backward differentiation method with h = 0.1.

(1) $y' = 3 + x - y$, $y(0) = 1$
(2) $y' = 5x - 3\sqrt{y}$, $y(0) = 2$
(3) $y' = 2y - 3x$, $y(0) = 1$
(4) $y' = 2x + ye^{-x}$, $y(0) = 1$
(5) $y' = y^2 + 2xy^3 + x^2$, $y(0) = 0.5$
(6) $y = (x^2 - y^2) sin\, y$, $y(0) = -1$
(7) Show that the first order Adams–Bashforth method is the Euler method and that the first order Adams–Moulton method is the backward Euler method.
(8) Show that the third order Adams–Bashforth formula is

$$y_{n+1} = y_n + \frac{h}{12}(23f_n - 16f_{n-1} + 5f_{n-2}).$$

(9) Show that the third order Adams–Moulton formula is

$$y_{n+1} = y_n + \frac{h}{12}(5f_{n+1} + 8f_n - f_{n-1}).$$

(10) Derive the second order backward differentiation formula

MULTISTEP METHODS (Tayler Expansion Technique)

Multistep methods make use of information about the solution and its derivative at more than one point in order to extrapolate to the next point. One specific class of multistep methods is based on the principle of numerical integration. If the differential equation $y' = f(x,y)$ is integrated from x_i to x_{i+1} we obtain

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x,y(x))dx \qquad (1)$$

To carry out the integration in (1), approximate $f(x,y(x))$ by a polynomial that interpolates $f(x,y(x))$at *k* points x_i , x_{i-1} ....x_{i-k+1}. If the Newton backward formula of degree *k-1* is used to interpolate $f(x,y(x))$, then the Adams-Bashforth formulas are generated and are of the form

$$y_{i+1} = y_i + h\sum_{j=1}^{k} b_j {y'}_{i-j+1} \qquad (2)$$

where

$${y'}_i = f(x_i, y(x_i))$$

This is called a *k-step* formula because it uses information from the previous *k* steps. Note that the Euler formula is a one-step formula ($k = 1$) with $b_1 = 1$.

Alternatively, if one begins with (1), the coefficients b_j can be chosen by assuming that the past values of *y* are exact and equating like powers of *h* in the expansion of (2) and of the local solution y_{i+1} about x_i. In the case of a three-step formula

$$y_{i+1} = y_i + h[b_1 {y'}_i + b_2 {y'}_{i-1} + b_3 {y'}_{i-2}]$$

Expand ${y'}_{i-1}, and\ {y'}_{i-2}$about x_i gives

$$y_{i+1} = y_i + h{y'}_i(b_1 + b_2 + b_3) - h^2{y''}_i(b_2 + 2b_3) + \frac{h^3}{3!}{y'''}_i(b_2 + 4b_3) + \cdots$$

Where

$${y'}_{i-1} = y'(x_i - h) = {y'}_i - h{y''}_i + \frac{h^2}{2!}{y'''}_i - \frac{h^3}{3!}{y''''}_i + \cdots$$

$${y'}_{i-2} = y'(x_i - 2h) = {y'}_i - 2h{y''}_i + \frac{4h^2}{2!}{y'''}_i - \frac{8h^3}{3!}{y''''}_i + \cdots$$

Multistep Methods

The Taylor's series expansion of y_{i+1} *is*

$${y'}_{i+1} = y'(x_i + h) = y_i + h{y'}_i + \frac{h^2}{2!}{y''}_i + \frac{h^3}{3!}{y'''}_i + \cdots$$

and upon equating like power of *h*, we have

$b_1 + b_2 + b_3$=1

$b_2 + 2b_3 = -\frac{1}{2}$

$b_2 + 2b_3 = \frac{1}{3}$

The solution of this set of linear equations is $b_1 = \frac{23}{12}$, $b_2 = \frac{-16}{12}$ and $b_3 = \frac{5}{12}$
Therefore, the three-step Adams-Bashforth formula is

$$y_{i+1} = y_i + \frac{h}{12}[23y'_i - 16y'_{i-1} - 5y'_{i-2}] \qquad (3)$$

A difficulty with multistep methods is that they are not self-starting. In (3) values for y_i y'_i, y'_{i-1} and y'_{i-2} are needed to compute y_{i+1} The traditional technique for computing starting values has been to use Runge-Kutta formulas of the same accuracy since they only require y_0 to get started. An alternative procedure, which turns out to be more efficient, is to use a sequence of s-step formulas with
$s = 1, 2, \ldots, k$. The computation is started with the one-step formulas in order to provide starting values for the two-step formula and so on. Also, the problem of getting started arises whenever the step-size *h* is changed. This problem is overcome by using a *k-step* formula whose coefficients depend upon the past step-size *h*. This kind of procedure is currently used in commercial multistep routines.
The previous multistep methods can be derived using polynomials that interpolated at the point x_i and at points backward from x_i these are sometimes known as formulas of explicit type. Formulas of implicit type can also be derived by basing the interpolating polynomial on the point x_{i+1} as well as on x_i and points backward from x_i The simplest formula of this type is obtained if the integral is approximated by the trapezoidal formula. This leads to

$$y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, y_{i+1})] \qquad (4)$$

If *f* is nonlinear, y_{i+1} cannot be solved for directly. However,
we can attempt to obtain y_{i+1} by means of iteration. Predict a first approximation $y^0{}_{i+1}$ to y_{i+1} by using the Euler method

$$y_{i+1} = y_i + hf(x_i, y_i) \qquad (5)$$

Then compute a corrected value with the trapezoidal formula

$$y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, y_i + hf(x_i, y_i)\ )] \qquad (6)$$

Which is called Modified Euler Method
For most problems occurring in practice, convergence generally occurs within one or two iterations. Equations (5) and (6) used as outlined above define the simplest predictor-corrector method.

Consistency and Convergence

We have learned that the numerical solution obtained from Euler's method,

$$y_{n+1} = y_n + hf(x_n, y_n);\ x_n = x_0 + nh;$$

converges to the exact solution $y(x)$ of the initial value problem

$$y_0 = hf(x, y);\ y(x_0) = y_0; as\ h \to 0.$$

We now analyze the convergence of a general one-step method of the form

$y_{n+1} = y_n + h\emptyset(x_n, y_n, h)$; for some continuous function$\emptyset(x_n, y_n, h)$. We define the local truncation error of this one-step method
by

$$T_n(h) = \frac{y_{n+1} - y_n}{h} - \emptyset(x_n, y_n, h):$$

That is, the local truncation error is the result of substituting the exact solution into the approximation of the ODE by the numerical method.
As $h \to 0$ and $n \to 1$, in such a way that $x_0 + nh = x \in [x_0; T]$, we obtain

$$T_n(h) \to y' - \emptyset(x, y(x), 0):$$

We therefore say that the one-step method is consistent if
$\emptyset(x, y(x), 0) = f(x, y)$:
A consistent one-step method is one that converges to the ODE as $h \to 0$.
We then say that a one-step method is stable if$\emptyset(x_n, y_n, h)$is Lipschitz continuous in y. That is,

$$|\emptyset(x, u, h) - \emptyset(x, v, h)| \leq L_\emptyset |u - v|; \; x \in [x_0, T], u, v \in R; \; h \in [0, h_0];$$

for some constant $L_\emptyset$
We now show that a consistent and stable one-step method is consistent. Using the same approach and notation as in the convergence proof of Euler's method, and the fact that the method is stable, we obtain the following bound for the global error

$$e_n = y(x_n) - y_n$$

$$|e_n| \leq \left(\frac{e^{L_\emptyset (T - x_0)} - 1}{L_\emptyset}\right) \max_{0 \leq m \leq n-1} |T_m(h)|$$

Because the method is consistent, we have

$$\lim_{h \to 0} \max_{0 < n < T/h} |T_n(h)| = 0:$$

It follows that as $h \to 0$ and $n \to 1$ in such a way that $x_0 + nh = x$, we have

$$\lim_{n \to \infty} |e_n| = 0;$$

and therefore the method is convergent.
In the case of Euler's method, we have

$$\emptyset(x, y, h) = f(x, y); \; T_n(h) = \frac{h}{2} f''(\mu); \mu \in (x_0, T):$$

Therefore, there exists a constant K such that

$$|T_n(h)| < Kh; \; 0 < h < h_0;$$

for some sufficiently small h_0. We say that Euler's method is first-order accurate.
More generally, we say that a one-step method has order of accuracy p if, for any sufficiently smooth solution $y(x)$,there exists constants K and h_0 such that

$$|T_n(h)| < Kh^p; \; 0 < h < h_0:$$

We now consider an example of a higher-order accurate method.

An Implicit One-Step Method

Suppose that we approximate the equation

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} y'(s)\, ds$$

by applying the Trapezoidal Rule to the integral. This yields a one-step method

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})]$$

known as the trapezoidal method. It follows from the error in the Trapezoidal Rule that

$$T_n(h) = \frac{y_{n+1} - y_n}{h} - \frac{1}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})] = -\frac{1}{12}h^2 y''(\tau_n)$$
$$\tau_n \in (x_n, x_{n+1})$$

Therefore, the trapezoidal method is second-order accurate.
To show convergence, we must establish stability by finding a suitable Lipschitz constant $L_\emptyset$for the function

$$\emptyset(x, y, h) = \frac{1}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})]$$

assuming that $L_f$is a Lipschitz constant for $f(x, y)$ in y. We have

$$|\emptyset(x, u, h) - \emptyset(x, v, h)|$$
$$= \frac{1}{2}|f(x, u, h) + f(x + h, u, \emptyset(x, u, h) - f(x, v, h) - f(x + h, v, \emptyset(x, v, h)|$$
$$\leq L_f|u - v| + \frac{h}{2}L_f|\emptyset(x, u, h) - \emptyset(x, v, h)|$$

Therefore

$$\left(1 - \frac{h}{2}L_f\right)|\emptyset(x, u, h) - \emptyset(x, v, h)| \leq L_f|u - v|$$

and therefore

$$L_\emptyset \leq \frac{L_f}{1 - \frac{h}{2}L_f}$$

provided that $\frac{h}{2}L_f < 1$. We conclude that for h sufficiently small, the trapezoidal method is stable,and therefore convergent, with $O(h^2)$ global error.
The trapezoidal method consist with Euler's method because it is an implicit method, due to the evaluation of $f(x, y)$ at y_{n+1}. It follows that it is generally necessary to solve a nonlinear equation to obtain y_{n+1} from y_n. This additional computational effort is offset by the fact that implicit methods are generally more stable than explicit methods such as Euler's method. Another example of an implicit method is backward Euler's method

$$y_{n+1} = y_n + \frac{h}{2}[f(x_{n+1}, y_{n+1})]$$

Like Euler's method, backward Euler's method is first-order accurate.

Boundary-Value Problems for Ordinary Differential Equations

In this chapter we show how to approximate the solution to **boundary-value** problems, differential equations with conditions imposed at different points. For first-order differential equations, only one condition is specified, so there is no distinction between initial-value and boundary-value problems. We will be considering second-order equations with two boundary values.
Physical problems that are position-dependent rather than time-dependent are often described in terms of differential equations with conditions imposed at more than one point. The two-point boundary-value problems in this chapter involve a second-order differential equation of the form

$$y'' = f\ (x, y, y'),\ \ for\ a \le x \le b, \qquad (1)$$

together with the boundary conditions

$$y(a) = \alpha \text{ and } y(b) = \beta. \qquad (2)$$

The Linear Shooting Method

The following theorem gives general conditions that ensure the solution to a second-order boundary value problem exists and is unique.

Theorem 1

Suppose the function *f* in the boundary-value problem

$$y'' = f\ (x, y, y'), \text{for } a \le x \le b, \text{with } y(a) = \alpha \text{ and } y(b) = \beta,$$

is continuous on the set

$$D = \{(x, y, y') | \text{for } a \le x \le b, \text{with } -\infty < y < \infty \text{ and } -\infty < y' < \infty\},$$

and that the partial derivatives f_y and $f_{y'}$ are also continuous on *D*. If

(i) $f_y\ (x, y, y') > 0$, for all $(x, y, y') \in D$, and

(ii) a constant *M* exists, with

$$\left|f_{y'}(x, y, y')\right| \le M, for\ all\ (x, y, y'') \in D,$$

then the boundary-value problem has a unique solution.

Example 1

Use Theorem (1) to show that the boundary-value problem

$y'' + e^{-xy} + sin\ y' = 0, for\ 1 \le x \le 2,$ with $y(1) = y(2) = 0,$

has a unique solution.

Solution

We have

$$f\ (x, y, y') = e^{-xy} + sin\ y'.$$

and for all *x* in [1, 2],

$f_y(x, y, y') = xe^{-xy} > 0\ and \left|f_{y'}(x, y, y')\right| = |-cos\ y'| \le 1.$

So the problem has a unique solution.

Linear Boundary-Value Problems

The differential equation

$y'' = f(x, y, y')$

is linear when functions $p(x), q(x)$, and $r(x)$ exist with

$f(x, y, y') = p(x)y' + q(x)y + r(x)$.
Problems of this type frequently occur, and in this situation, Theorem (1) can be simplified

Corollary 11.2

Suppose the linear boundary-value problem
$y'' = p(x)y' + q(x)y + r(x)$, for $a \le x \le b$, with $y(a) = \alpha$ and $y(b) = \beta$,
Satisfies

(i) $p(x), q(x)$, and $r(x)$ are continuous on $[a, b]$,
(ii) $q(x) > 0$ on $[a, b]$.

Then the boundary-value problem has a unique solution.
To approximate the unique solution to this linear problem, we first consider the initial value problems
$y'' = p(x)y' + q(x)y + r(x)$, for $a \le x \le b$,
with $y(a) = \alpha$, and $y'(a) = 0$, (3)
and
$y'' = p(x)y' + q(x)y + r(x)$, for $a \le x \le b$,
with $y(a) = 0$, and $y'(a) = 1$, (4)

Both problems have a unique solution.

Let $y_1(x)$ denote the solution to (3), and let $y_2(x)$ denote the solution to (4).
Assume that $y_2(x) \neq 0$.
Define

$$y(x) = y_1(x) + \frac{\beta - y_1(b)}{y_2(b)} y_2(x). \qquad (5)$$

Then $y(x)$ is the solution to the linear boundary problem (3). To see this, first note that

$$y'(x) = y_1'(x) + \frac{\beta - y_1(b)}{y_2(b)} y_2'(x) \qquad (6)$$

and

$$y''(x) = y_1''(x) + \frac{\beta - y_1(b)}{y_2(b)} y_2''(x) \qquad (7)$$

Substituting for $y_1''(x)$ *(x)* and $y_2''(x)$ in this equation gives

$$\begin{aligned} y'' &= p(x)y_1'(x) + q(x)y_1(x) + r(x) + \frac{\beta - y1(b)}{y2(b)}(p(x)y_2'(x) + q(x)y_2) \\ &= p(x)\left(y_1'(x) + \frac{\beta - y_1(b)}{y_2(b)} y_2'(x)\right) \\ &+ q(x)\left(y_1(x) + \frac{\beta - y_1(b)}{y_2(b)} y_2(x)\right) + r(x) \\ &= p(x)y'(x) + q(x)y(x) + r(x). \end{aligned}$$

Moreover,

$$y(a) = y_1(a) + \frac{\beta - y_1(b)}{y_2(b)} y_2(a) = \alpha + \frac{\beta - y_1(b)}{y_2(b)} \cdot 0 = \alpha$$

and

$$y(b) = y_1(b) + \frac{\beta - y_1(b)}{y_2(b)} y_2(b) = y_1(b) + \beta - y_1(b) = \beta$$

Linear Shooting

The Shooting method for linear equations is based on the replacement of the linear boundary value problem by the two initial-value problems (3) and (4). Numerous methods are available from Chapter 5 for approximating the solutions$y_1(x)$ and $y_2(x)$, and once these approximations are available, the solution to the boundary-value problem is approximated using Eq. (5). Graphically, the method has the appearance shown in Figure 1.

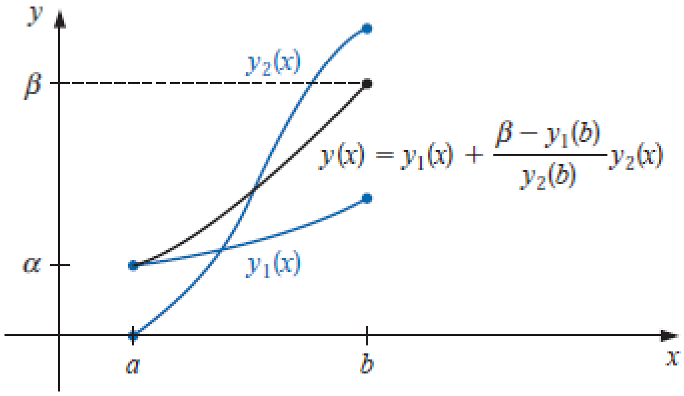


Fig.1

Example 2

Apply the Linear Shooting technique with $N = 10$ to the boundary-value problem $y'' = -\frac{2}{x}y' + \frac{2}{x^2}\, y + \frac{sin(ln\, x)}{x^2}$, for $1 \le x \le 2$, with $y(1) = 1$ and $y(2) = 2$, and compare the results to those of the exact solution

$$y = c_1 x + \frac{c_2}{x^2} - \frac{3}{10}\, sin(ln\, x) - \frac{1}{10} cos(ln\, x),$$

Where

$$c_2 = \frac{1}{70}[8 - 12\, sin(ln\, 2) - 4\, cos(ln\, 2)] \approx -0.03920701320$$

and

$$c_1 = \frac{11}{10} - c_2 \approx 1.1392070132.$$

Solution

$$y''_1 = -\frac{2}{x}y'_1 + \frac{2}{x^2}y_1 + \sin(\ln x)\, x^2 ,$$

for $1 \le x \le 2$, with $y_1(1) = 1$and $y'_1(1) = 0$

,

and

$$y''_2 = -\frac{2}{x}y'_2 + \frac{2}{x^2}y_2 ,$$

$, for\ 1 \le x \le 2,$with $y_2(1) = 0$ and $y'_2(1) = 1.$

The results of the calculations, with $N = 10$ and $h = 0.1$, are given in Table 1. The value listed as $u_{1,i}$ approximates $y_1(x_i)$, the value $v_{1,i}$approximates $y_2(x_i)$,, and w_i approximates

$$y'(x_i) = y_1'(x_i) + \frac{\beta - y_1(b)}{y_2(b)} y_2'(xx_i)$$

x_i	$u_{1,i} \approx y_1(x_i)$	$v_{1,i} \approx y_2(x_i)$	$w_i \approx y(x_i)$	$y(x_i)$	$\lvert y(x_i) - w_i \rvert$
1.0	1.00000000	0.00000000	1.00000000	1.00000000	
1.1	1.00896058	0.09117986	1.09262917	1.09262930	1.43×10^{-7}
1.2	1.03245472	0.16851175	1.18708471	1.18708484	1.34×10^{-7}
1.3	1.06674375	0.23608704	1.28338227	1.28338236	9.78×10^{-8}
1.4	1.10928795	0.29659067	1.38144589	1.38144595	6.02×10^{-8}
1.5	1.15830000	0.35184379	1.48115939	1.48115942	3.06×10^{-8}
1.6	1.21248372	0.40311695	1.58239245	1.58239246	1.08×10^{-8}
1.7	1.27087454	0.45131840	1.68501396	1.68501396	5.43×10^{-10}
1.8	1.33273851	0.49711137	1.78889854	1.78889853	5.05×10^{-9}
1.9	1.39750618	0.54098928	1.89392951	1.89392951	4.41×10^{-9}
2.0	1.46472815	0.58332538	2.00000000	2.00000000	

Table (12)

The accurate results in this example are due to the fact that the fourth-order Runge-Kutta method gives $O(h^4)$ approximations to the solutions of the initial-value problems. Unfortunately, because of round off errors, there can be problems hidden in this technique

EXERCISE SET 1

1. The boundary-value problem

$$y'' = 4(y - x), 0 \leq x \leq 1, y(0) = 0, y(1) = 2,$$

has the solution $y(x) = e^2(e^4 - 1) - 1(e^{2x} - e^{-2x}) + x$. Use the Linear Shooting method to approximate the solution, and compare the results to the actual solution.

a. With $h = 0.5$;

b. With $h = 0.25$.

2. The boundary-value problem

$$y'' = y' + 2y + \cos x, 0 \leq x \leq \frac{\pi}{2}, y(0) = -0.3, y\left(\frac{\pi}{2}\right) = -0.1$$

has the solution $y(x) = -110\,(\sin x + 3\cos x)$. Use the Linear Shooting method to approximate the solution, and compare the results to the actual solution.

a. With $h = (\pi/4)$;

b. With $h = (\pi/8)$.

3. Use the Linear Shooting method to approximate the solution to the following boundary-value problems.

a. $y'' = -3y' + 2y + 2x + 3,$
$0 \leq x \leq 1, y(0) = 2, y(1) = 1;$ use $h = 0.1.$
b. $y'' = -4x^{-1}y' - 2x^{-2}y + 2x^{-2}\ln x,$
$1 \leq x \leq 2, y(1) = -0.5, y(2) = ln\, 2;$ use $h = 0.05.$
c. $y'' = -(x + 1)y' + 2y + (1 - x^2)e^{-x},$
$0 \leq x \leq 1, y(0) = -1, y(1) = 0;$ use $h = 0.1.$
d. $y'' = x - 1y + 3x - 2y + x - 1 \ln x - 1,$
$1 \leq x \leq 2, y(1) = y(2) = 0;$ $use\ h = 0.1.$

4. Although $q(x) < 0$ in the following boundary-value problems, unique solutions exist and are given. Use the Linear Shooting Algorithm to approximate the solutions to the following problems, and compare the results to the actual solutions.
a. $y'' + y = 0, 0 \leq x \leq \pi/4, y(0) = 1, y(\pi/4) = 1;$ use $h = \pi/20$;
Actual solution $y(x) = cos\, x + (\sqrt{2} - 1)sin\, x.$
b. $y'' + 4y = \cos x, 0 \leq x \leq \pi/4, y(0) = 0, y(\pi/4) = 0;$ use $h = \pi/20$;
Actual solution $y(x) = -13\, cos\, 2x - \sqrt{26}\, sin\, 2x + 13\, cos\, x.$
c. $y'' = -4x^{-1}y' + 2x^{-2}y - 2x^{-2}\ln x,$
$y(1) = 1/2, y(2) = ln\, 2;$ use $h = 0.05;$
Actual solution $y(x) = 4x^{-1} - 2x^{-2} + ln\, x - 3/2.$
d. $y'' = 2y' - y + xe^x - x, 0 \leq x \leq 2,$
$y(0) = 0, y(2) = -4;$ use $h = 0.2;$
Actual solution $y(x) = 16\, x^3e^x - 53\, xe^x + 2e^x - x - 2.$

5. Use the Linear Shooting Algorithm to approximate the solution $y = e^{-10x}$ to the boundary-value problem
$$y'' = 100y, \qquad 0 \leq x \leq 1, y(0) = 1, y(1) = e^{-10}$$
Use h = 0.1 and 0.05.

Finite-Difference Methods for Linear Problems

The linear and nonlinear Shooting methods for boundary-value problems can present problems of instability. The methods in this section have better stability characteristics, but they generally require more computation to obtain a specified accuracy. Methods involving finite differences for solving boundary-value problems replace each of the derivatives in the differential equation with an appropriate difference-quotient approximation of the type considered in Section 4.1. The particular difference quotient and step size h are chosen to maintain a specified order of truncation error. However, h cannot be chosen too small because of the general instability of the derivative approximations.

Discrete Approximation

The finite difference method for the linear second-order boundary-value problem,

$y'' = p(x)y' + q(x)y + r(x),$ for $a \leq x \leq b,$ with $y(a) = \alpha$ and $y(b) = \beta$
(11.14)(8)

Requires that difference-quotient approximations be used to approximate both $y'$and y''. First, we select an integer N > 0 and divide the interval [a, b] into (N+1) equal subintervals whose endpoints are the mesh points

$x_i = a + ih$, for $i = 0, 1, \ldots, N+1$, where $h = (b-a)/(N+1)$.

Choosing the step size h in this manner facilitates the application of a matrix algorithm from Chapter 6, which solves a linear system involving an $N \times N$ matrix. At the interior mesh points, xi, for $i = 1, 2, \ldots, N$, the differential equation to be approximated is

$$y''(x_i) = p(x_i)y'(x_i) + q(x_i)y(x_i) + r(x_i). \qquad (9)(11.15)$$

Expanding y in a third Taylor polynomial about x_i evaluated at $x_{i+1}$and x_{i-1}, we have, assuming that $y \in C^4[x_{i-1}, x_{i+1}]$,

$$y(x_{i+1}) = y(x_i + h) = y(x_i) + hy'(x_i) + \frac{h^2}{2!}y''(x_i) + \frac{h^3}{3!}y'''(x_i) + \frac{h^4}{24}y^{(4)}(\xi_{+i})$$

$$y(x_{i-1}) = y(x_i - h) = y(x_i) - hy'(x_i) + \frac{h^2}{2!}y''(x_i) - \frac{h^3}{3!}y'''(x_i) + \frac{h^4}{24}y^{(4)}(\xi_{-i})$$

for some ξ_{+i} in (x_i, x_{i+1}), and for some ξ_{-i}in (x_{i-1}, x_i). If these equations are added, we have

$$y(x_{i+1}) + y(x_{i-1}) = 2y(x_i) + h^2y''(x_i) + \frac{h^4}{24}\left[y^{(4)}(\xi_{+i}) + y^{(4)}(\xi_{-i})\right],$$

and solving for $y''(x_i)$) gives

$$y''(x_i) = \frac{1}{h^2}[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})] - \frac{h^2}{24}\left[y^{(4)}(\xi_{+i}) + y^{(4)}(\xi_{-i})\right].$$

The Intermediate Value Theorem can be used to simplify the error term to give

$$y''(x_i) = \frac{1}{h^2}[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})] - \frac{h^2}{12}y^{(4)}(\xi_i). \qquad (10)(11.16)$$

for some ξ_i in (x_{i-1}, x_{i+1})

This is called the **centered-difference formula** for $y''(x_i)$.

A centered-difference formula for $y'(x_i)$is obtained in a similar manner (the details were considered in Section 4.1), resulting in

$$y'(x_i) = \frac{1}{2h}[y(x_{i+1}) - y(x_{i-1})] - \frac{h^2}{12}y^{(3)}(\eta_i). \qquad (11)(11.17)$$

for some η_i in (x_{i-1}, x_{i+1}) The use of these centered-difference formulas in Eq. (9) results in the equation

$$\frac{1}{h^2}[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})]$$
$$= p(x_i)\left(\frac{1}{2h}[y(x_{i+1}) - y(x_{i-1})]\right) + q(x_i)y(x_i) + r(x_i)$$
$$- \frac{h^2}{12}\left(2p(x_i)y^{(3)}(\eta_i) + y^{(4)}(\xi_i)\right)$$

A Finite-Difference method with truncation error of order $O(h^2)$ results by using this equation together with the boundary conditions $y(a) = \alpha$ and $y(b) = \beta$ to define the system of linear equations

$$w_0 = \alpha, w_{N+1} = \beta$$

and

$$\frac{1}{h^2}[w_{i+1} - 2w_i + w_{i-1}] - p(x_i)\left(\frac{1}{2h}[w_{i+1} - w_{i-1}]\right) - q(x_i)y(x_i) = r(x_i)$$

(12) (11.18)

for each $i = 1, 2, \ldots, N$.

In the form we will consider, Eq. (12) is rewritten as

$$-\left((1 + \tfrac{h}{2}p(x_i)\right)w_{i-1} + \left(2 + h^2 q(x_i)\right) w_i - \left((1 - \tfrac{h}{2}p(x_i)\right)w_{i+1} = -h^2 r(x_i)$$

and the resulting system of equations is expressed in the tridiagonal $N \times N$ matrix form

$$A\mathbf{w} = \mathbf{b}, \qquad (13)$$

$$A = \begin{bmatrix} 2 + h^2 q(x_1) & -1 + \frac{h}{2}p(x_1) & 0 & \cdots & 0 \\ -1 - \frac{h}{2}p(x_2) & 2 + h^2 q(x_2) & -1 + \frac{h}{2}p(x_2) & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & -1 + \frac{h}{2}p(x_{N-1}) \\ 0 & \cdots & 0 & -1 - \frac{h}{2}p(x_N) & 2 + h^2 q(x_N) \end{bmatrix},$$

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N-1} \\ w_N \end{bmatrix}, \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} -h^2 r(x_1) + \left(1 + \frac{h}{2}p(x_1)\right) w_0 \\ -h^2 r(x_2) \\ \vdots \\ -h^2 r(x_{N-1}) \\ -h^2 r(x_N) + \left(1 - \frac{h}{2}p(x_N)\right) w_{N+1} \end{bmatrix}.$$

Example 1

Consider $N = 9$ to approximate the solution to the linear boundary-value problem

$y'' = -\frac{2}{x}y' + \frac{2}{x^2}\, y + \frac{sin(ln\, x)}{x^2}$, for $1 \le x \le 2$, with $y(1) = 1$ and $y(2) = 2$,

and compare the results to exact solution

$$y = 1.1392070132.x + \frac{-0.03920701320}{x^2} - \frac{3}{10}\, sin(ln\, x) - \frac{1}{10} cos(ln\, x),$$

Solution

For this example, we will use $N = 9$, so $h = 0.1$

x_i	w_i	$y(x_i)$	$\|w_i - y(x_i)\|$
1.0	1.00000000	1.00000000	
1.1	1.09260052	1.09262930	2.88×10^{-5}
1.2	1.18704313	1.18708484	4.17×10^{-5}
1.3	1.28333687	1.28338236	4.55×10^{-5}
1.4	1.38140205	1.38144595	4.39×10^{-5}
1.5	1.48112026	1.48115942	3.92×10^{-5}
1.6	1.58235990	1.58239246	3.26×10^{-5}
1.7	1.68498902	1.68501396	2.49×10^{-5}
1.8	1.78888175	1.78889853	1.68×10^{-5}
1.9	1.89392110	1.89392951	8.41×10^{-6}
2.0	2.00000000	2.00000000	

Table (13)

These results are considerably less accurate than those obtained in Example 1. This is because the method used in that example involved a Runge-Kutta technique with local truncation error of order $O(h^4)$, whereas the difference method used here has local truncation error of order $O(h^2)$. To obtain a difference method with greater accuracy, we can proceed in a number of ways. Using fifth-order Taylor series for approximating $y''(x_i)$ and $y'(x_i)$ results in a truncation error term involving h^4. However, this process requires using multiples not only of $y(x_{i+1})$ and $y(x_{i-1})$, but also of $y(x_{i+2})$ and $y(x_{i-2})$ in the approximation formulas for $y''(x_i)$ and $y'(x_i)$. This leads to difficulty at $i = 0$, because we do not know w_{-1}, and at $i = N$, because we do not know w_{N+2}. Moreover, the resulting system of equations analogous to (13) is not in tridiagonal form, and the solution to the system requires many more calculations.

EXERCISE SET 11.3

1. The boundary-value problem

$$y'' = 4(y - x), 0 \le x \le 1, y(0) = 0, y(1) = 2$$

has the solution $y(x) = e^2(e^4 - 1)^{-1}(e^{2x} - e^{-2x}) + x$. Use the Linear Finite-Difference method to approximate the solution, and compare the results to the actual solution.

a. With $h = 0.5$

b. With $h = 0.25$

c. Use extrapolation to approximate $y(1/2)$.

2. The boundary-value problem

$$y'' = y' + 2y + \cos x, 0 \le x \le \pi/2, y(0) = -0.3, y(\pi/2) = -0.1$$

has the solution $y(x) = -1/10\,(\sin x + 3\cos x)$. Use the Linear Finite-Difference method to approximate the solution, and compare the results to the actual solution.

a. With $h = \pi/4$; **b.** With $h = \pi/8$.

c. Use extrapolation to approximate $y(\pi/4)$.

3. Use the Linear Finite-Difference Algorithm to approximate the solution to the following boundary value problems.

a. $y'' = -3y' + 2y + 2x + 3, \quad 0 \le x \le 1, y(0) = 2, y(1) = 1;$ use $h = 0.1$.

b. $y'' = -4x^{-1}y' + 2x^{-2}y - 2x^{-2}\ln x, \quad 1 \le x \le 2,$
$y(1) = -12, y(2) = \ln 2;$ use $h = 0.05$.

c. $y'' = -(x+1)y' + 2y + (1 - x^2)e^{-x}, \quad 0 \le x \le 1,$
$y(0) = -1, y(1) = 0;$ use $h = 0.1$.

d. $y'' = x^{-1}y' + 3x^{-2}y + x^{-1}\ln x - 1, 1 \le x \le 2,$
$y(1) = y(2) = 0;$ use $h = 0.1$.

4. Although $q(x) < 0$ in the following boundary-value problems, unique solutions exist and are given. Use the Linear Finite-Difference Algorithm to approximate the solutions, and compare the results to the actual solutions.

e. $y'' + y = 0, 0 \le x \le \pi/4, y(0) = 1, y(\pi/4) = 1;$ use $h = \pi/20$;
actual solution $y(x) = \cos x + (\sqrt{2} - 1)\sin x.$

f. $y'' + 4y = \cos x, 0 \le x \le \pi/4, y(0) = 0, y(\pi/4) = 0;$ use $h = \pi/20$;
actual solution $y(x) = -13\cos 2x - \sqrt{26}\sin 2x + 13\cos x.$

g. $y'' = -4x^{-1}y' + 2x^{-2}y - 2x^{-2}\ln x,$
$y(1) = 1/2, y(2) = \ln 2;$ use $h = 0.05$;
actual solution $y(x) = 4x^{-1} - 2x^{-2} + \ln x - 3/2.$

h. $y'' = 2y' - y + xe^x - x, 0 \le x \le 2,$
$y(0) = 0, y(2) = -4;$ use $h = 0.2$;
actual solution $y(x) = 16x^3e^x - 53xe^x + 2e^x - x - 2.$

5. Use the Linear Finite-Difference Algorithm to approximate the solution $y = e^{-10x}$ to the boundary value problem

$$y'' = 100y, 0 \le x \le 1, y(0) = 1, y(1) = e^{-10}$$

Use $h = 0.1$ and 0.05. Can you explain the consequences?

6. Repeat Exercise 3(a) and (b) using the extrapolation discussed in Example 2.

The Rayleigh-Ritz Method

The Shooting method for approximating the solution to a boundary-value problem replaced the boundary-value problem with pair of initial-value problems. The finite-difference approach replaces the continuous operation of differentiation with the discrete operation of finite differences. The Rayleigh-Ritz method is a variational technique that attacks the problem from a third approach. The boundary-value problem is first reformulated as a problem of choosing, from the set of all sufficiently differentiable functions satisfying the boundary conditions, the function to minimize a certain integral. Then the set of feasible functions is reduced in size, and an approximation is found from this set to minimize the integral. This gives our approximation to the solution of the boundary-value problem. To describe the Rayleigh-Ritz method, we consider approximating the solution to a linear two-point boundary-value problem from beam-stress analysis. This boundary-value problem is described by the differential equation

$$-\frac{d}{dx}(p(x)y') + q(x)y = f(x), \text{for } 0 \le x \le 1, \qquad (14) \quad (11.21)$$

with the boundary conditions

$$y(0) = y(1) = 0. \qquad (15)(11.22)$$

This differential equation describes the deflection *y(x)* of a beam of length 1 with variable cross section represented by $q(x)$. The deflection is due to the added stresses $p(x)$ and $f(x)$.

More general boundary conditions are considered in Exercises 6 and 9.

In the discussion that follows, we assume that $p \in C^1[0,1]$ and $q, f \in C[0,1]$. Further, we assume that there exists a constant $\delta > 0$ such that

$p(x) \geq \delta$, and that $q(x) \geq 0$, for each x in [0, 1].

These assumptions are sufficient to guarantee that the boundary-value problem given in (14) and (15) has a unique solution (see [BSW]).

Variational Problems

As is the case in many boundary-value problems that describe physical phenomena, the solution to the beam equation satisfies an integral minimization **variational** property. The variational principle for the beam equation is fundamental to the development of the Rayleigh-Ritz method and characterizes the solution to the beam equation as the function that minimizes an integral over all functions in $C^2[0,1]$, the set of those functions u in $C^2[0,1]$ with the property that $u(0) = u(1) = 0$. The following theorem gives the characterization.

Theorem (2)

Let $p \in C^1[0,1]$, $q, f \in C[0,1]$, and

$$p(x) \geq \delta > 0, q(x) \geq 0, for\ 0 \leq x \leq 1.$$

The function $y \in C_0^2[0, 1]$ is the unique solution to the differential equation

$$-\frac{d}{dx}(p(x)y') + q(x)y = f(x), \text{for } 0 \leq x \leq 1, \qquad (16)$$

if and only if y is the unique function in $C_0^2[0, 1]$ that minimizes the integral

$$I[u] = \int_0^1 \{p(x)[u'(x)]^2 + q(x)[u(x)]^2 - 2f(x)u(x)\}\, dx. \qquad (17)$$

Details of the proof of this theorem can be found in [Shul], pp. 88-89. It proceeds in three steps.

•First it is shown that any solution y to (16) also satisfies the equation

$$\int_0^1 f(x)u(x)dx = \int_0^1 p(x)y'(x)u'(x) + q(x)y(x)u(x)dx, \qquad (18)$$

for all $u \in C_0^2[0,1]$.

• The second step shows that $y \in C_0^2[0,1]$ is a solution to (17) if and only if (18) holds for all $u \in C_0^2[0,1]$.

• The final step shows that (18) has a unique solution. This unique solution will also be a solution to (16) and to (17), so the solutions to (16) and (17) are identical.

The Rayleigh-Ritz method approximates the solution y by minimizing the integral, not over all the functions in $C_0^2[0,1]$, but over a smaller set of functions consisting of linear combinations of certain basis functions $\varphi_1, \varphi_2, ..., \varphi_n$. The basis functions are linearly independent and satisfy

$$\varphi_i(0) = \varphi_i(1) = 0 \text{ , for each } i = 1, 2, \ldots, n.$$

An approximation $\varphi(x) = \sum_{i=1}^{n} c_i \varphi_i(x)$ to the solution $y(x)$ of Eq. (16) is then obtained by finding constants $c_1, c_2, \ldots, c_n$ to minimize the integral $I[\varphi(x)] = I[\sum_{i=1}^{n} c_i \varphi_i]$

From Eq. (17),

$$I[\varphi(x)] = I[\textstyle\sum_{i=1}^{n} c_i \varphi_i] \tag{19}$$
$$= \int_0^1 \{p(x)[\textstyle\sum_{i=1}^{n} c_i \varphi'_i]^2 + q(x)[\sum_{i=1}^{n} c_i \varphi_i]^2 - 2f(x) \sum_{i=1}^{n} c_i \varphi_i\} dx$$

and, for a minimum to occur, it is necessary, when considering I as a function of $c_1, c_2, \ldots, c_n$ to have

$$\frac{\partial I}{\partial c_j} = 0, \text{ for each } j = 1, 2, \ldots, n. \tag{20}$$

Differentiating (19) gives

$$\frac{\partial I}{\partial c_j} = \int_0^1 \left\{ 2p(x) \sum_{i=1}^{n} c_i \varphi'_i(x) c_i \varphi'_j(x) + 2q(x) \sum_{i=1}^{n} c_i \varphi_{i(x)} \varphi_{j(x)} - 2f(x) \varphi_j(x) \right\} dx$$

for each $j = 1, 2, \ldots, n.$

and substituting into Eq. (20) yields

$$\sum_{i=1}^{n} \left[\int_0^1 \left\{ p(x) \varphi'_i(x) c_i \varphi'_j(x) + q(x) \varphi_{i(x)} \varphi_{j(x)} \right\} c_i \right] dx - \int_0^1 f(x) \varphi_j(x) dx = 0$$

(21)(11.28)

for each $j = 1, 2, \ldots, n.$

The **normal equations** described in Eq. (21) produce an $n \times n$ linear system $A\mathbf{c} = \mathbf{b}$ in the variables $c_1, c_2, \ldots, c_n$, where the symmetric matrix A has

$$a_{ij} = \int_0^1 p(x) \varphi'_i(x) \varphi'_j(x) + q(x) \varphi_i(x) \varphi_j(x) \; dx$$

and **b** is defined by

$$b_i = \int_0^1 f(x) \varphi_j(x) dx$$

Piecewise-Linear Basis

The simplest choice of basis functions involves piecewise-linear polynomials. The first step is to form a partition of [0, 1] by choosing points $x_0, x_1, \ldots, x_{n+1}$ with $0 = x_0 < x_1 < \ldots < x_n < x_{n+1} = 1$. Letting $h_i = x_{i+1} - x_i$, for each $i = 0, 1, \ldots, n$, we define the basis functions $\varphi_1(x), \varphi_2(x), \ldots, \varphi_n(x)$ by

$$\varphi(x) = \begin{cases} 0 & \text{if } 0 \le x \le x_{i-1}, \\ \dfrac{(x - x_{i-1})}{h_{i-1}} & \text{if } x_{i-1} \le x \le x_i, \\ \dfrac{(x_{i+1} - x)}{h_i} & \text{if } x_i \le x \le x_{i+1}, \\ 0 & \text{if } x_{i+1} \le x \le 1. \end{cases} \tag{22}$$

for each $i = 1, 2, \ldots, n$.

The functions $\varphi_i(x)$ are piecewise-linear, so the derivatives $\varphi'_i(x)$, while not continuous, are constant on (x_j, x_{j+1}), for each $j = 0, 1, \ldots, n$, and

$$\varphi'_i(x) = \begin{cases} 0 & \text{if } 0 \le x \le x_{i-1}, \\ \dfrac{1}{h_{i-1}} & \text{if } x_{i-1} \le x \le x_i, \\ \dfrac{1}{h_i} & \text{if } x_i \le x \le x_{i+1}, \\ 0 & \text{if } x_{i+1} \le x \le 1. \end{cases} \tag{23}$$

for each $i = 1, 2, \ldots, n$.

Because $\varphi_i(x)$ and $\varphi'_i(x)$ are nonzero only on (x_i, x_{i+1}), $\varphi_i(x)\varphi_j(x) = 0$ and $\varphi'_i(x)\varphi'_j(x) = 0$, except when j is $i - 1$, i, or $i + 1$. As a consequence, the linear system given by (21) reduces to an $n \times n$ tridiagonal linear system. The nonzero entries in A are

$$\begin{aligned} a_{i,i} &= \int_0^1 \left\{p(x)\left[\varphi'_i(x)\right]^2 + q(x)[\varphi_i(x)]^2\right\} dx \\ &= \left(\frac{1}{h_{i-1}}\right)^2 \int_{x_{i-1}}^{x_i} p(x)dx + \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} p(x)dx \\ &= \left(\frac{1}{h_{i-1}}\right)^2 \int_{x_{i-1}}^{x_i} (x - x_{i-1})^2\, q(x)dx + \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} (x_{i+1} - x)^2\, q(x)dx \end{aligned}$$

for each $i = 1, 2, \ldots, n$;

$$a_{i,i+1} = \int_0^1 \{p(x)\varphi'_i(x)\varphi'_{i+1}(x) + q(x)\varphi_i(x)\varphi_{i+1}(x)\}dx$$
$$= \left(\frac{-1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} p(x)dx + \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} (x_{i+1} - x)(x - x_i)q(x)dx$$

for each $i = 1, 2, \ldots, n - 1$;

$$a_{i,i-1} = \int_0^1 \{p(x)\varphi'_i(x)\varphi'_{i-1}(x) + q(x)\varphi_i(x)\varphi_{i-1}(x)\}dx$$
$$= \left(\frac{-1}{h_{i-1}}\right)^2 \int_{x_{i-1}}^{x_i} p(x)dx + \left(\frac{1}{h_{i-1}}\right)^2 \int_{x_{i-1}}^{x_i} (x_i - x)(x - x_{i-1})q(x)dx$$

for each $i = 2, \ldots, n$;

$$b_i = \int_0^1 f(x)\varphi_i(x)dx$$
$$= \left(\frac{1}{h_{i-1}}\right) \int_{x_{i-1}}^{x_i} (x - x_{i-1})f(x)dx + \left(\frac{1}{h_i}\right) \int_{x_i}^{x_{i+1}} (x_{i+1} - x)f(x)dx$$

for each $i = 1,2, \ldots, n$;

There are six types of integrals to be evaluated:

$$Q_{1,i} = \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} (x_{i+1} - x)(x - x_i)\ q(x)dx,$$

for each $i = 1, 2, \ldots, n - 1$,

$$Q_{2,i} = \left(\frac{1}{h_{i-1}}\right)^2 \int_{x_{i-1}}^{x_i} (x - x_{i-1})^2\ q(x)dx,$$

for each $i = 1, 2, \ldots, n$,

$$Q_{3,i} = \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} (x_{i+1} - x)^2\ q(x)\ dx,$$

for each $i = 1, 2, \ldots, n$,

$$Q_{4,i} = \left(\frac{1}{h_{i-1}}\right)^2 \int_{x_{i-1}}^{x_i} p(x)\ dx,$$

for each $i = 1, 2, \ldots, n + 1$,

$$Q_{5,i} = \frac{1}{h_{i-1}} \int_{x_{i-1}}^{x_i} (x - x_{i-1})\ f(x)\ dx,$$

for each $i = 1, 2, \ldots, n1$,

And

$$Q_{6,i} = \frac{1}{h_i}\int_{x_i}^{x_{i+1}} (x_{i+1} - x)\, f(x)\, dx,$$

for each $i = 1, 2, \ldots, n$,

The matrix A and the vector **b** in the linear system $A\mathbf{c} = \mathbf{b}$ have the entries

$$a_{i,i} = Q_{4,i} + Q_{4,i+1} + Q_{2,i} + Q_{3,i}\,, \quad \text{for each } i = 1, 2, \ldots, n,,$$
$$a_{i,i+1} = -Q_{4,i+1} + Q_{1,i}, \quad \text{for each } i = 1, 2, \ldots, n - 1,$$
$$a_{i,i-1} = -Q_{4,i} + Q_{1,i-1}, \quad \text{for each } i = 2, 3, \ldots, n,$$

and

$$b_i = Q_{5,i} + Q_{6,i}, \quad \text{for each } i = 1, 2, \ldots, n.$$

The entries in **c** are the unknown coefficients $c_1, c_2, \ldots, c_n$, from which the Rayleigh-Ritz approximation φ, given by $\varphi(x) = \sum_{i=1}^{n} c_i \varphi_i(x)$, is constructed.

To employ this method requires evaluating 6*n* integrals, which can be evaluated either directly or by a quadrature formula such as Composite Simpson's rule.
An alternative approach for the integral evaluation is to approximate each of the functions *p*, *q*, and *f* with its piecewise-linear interpolating polynomial and then integrate the approximation. Consider, for example, the integral $Q_{1,i}$. The piecewise-linear interpolation of q is

$$P_q(x) = \sum_{i=0}^{n+1} q(x_i)\varphi_i(x)\,,$$

where $\varphi_1, \ldots, \varphi_n$ are defined in (11.30) and

$$\varphi_0 = \begin{cases} \frac{(x_1 - x)}{x_1} & 0 \le x \le x_1 \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \varphi_{n+1} = \begin{cases} \frac{(x - x_i)}{x_i} & 0 \le x \le x_1 \\ 0 & \text{otherwise} \end{cases}$$

The interval of integration is $[x_i, x_{i+1}]$, so the piecewise polynomial $P_q(x)$ reduces to

$$P_q(x) = q(x_i)\varphi_i(x) + q(x_{i+1})\varphi_{i+1}(x).$$
$$|q(x) - Pq(x)| = O(h2i\,), \quad \text{for} \quad x_i \le x \le x_{i+1},$$

if $q \in C^2[x_i, x_{i+1}]$. For $i = 1, 2, \ldots, n - 1$, the approximation to $Q_{1,i}$ is obtained by integrating the approximation to the integrand

$$Q_{1,i} = \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} (x_{i+1} - x)(x - x_i)\; q(x)dx$$
$$\approx \left(\frac{1}{h_i}\right)^2 \int_{x_i}^{x_{i+1}} (x_{i+1} - x)(x - x_i)\left[\frac{q(x_i)(x_{i+1} - x)}{h_i} + \frac{q(x_{i+1})(x - x_i)}{h_i}\right] dx$$

$$= \frac{h_i}{12}[q(x_i) + q(x_{i+1})]$$

Further, if $q \in C^2[x_i, x_{i+1}]$then

$$\left| Q_{1,i} - \frac{h_i}{12}[q(x_i) + q(x_{i+1})] \right| = O(h_i^3)$$

Approximations to the other integrals are derived in a similar manner and are given by

$$Q_{2,i} \approx \frac{h_{i-1}}{12}[3q(x_i) + q(x_{i-1})],$$
$$Q_{3,i} \approx \frac{h_i}{12}[3q(x_i) + q(x_{i+1})],$$
$$Q_{4,i} \approx \frac{h_{i-1}}{2}[p(x_i) + p(x_{i-1})],$$
$$Q_{5,i} \approx \frac{h_{i-1}}{6}[2f(x_i) + f(x_{i-1})],$$

and

$$Q_{6,i} \approx \frac{h_i}{6}[2f(x_i) + f(x_{i+1})]].$$

Illustration

Consider the boundary-value problem

$$-y'' + \pi^2 y = 2\pi^2 \sin(\pi x), \text{for } 0 \le x \le 1, \text{with } y(0) = y(1) = 0.$$

Let $h_i = h = 0.1$, so that $x_i = 0.1i$, for each $i = 0, 1, \dots, 9$.

The integrals are

$$Q_{1,i} = 100\int_{0.1i}^{0.1i+0.1} (0.1i + 0.1 - x)(x - 0.1i)\pi^2 \, dx = \frac{\pi^2}{60},$$

$$Q2, i = 100\int_{0.1i-0.1}^{0.1i} (x - 0.1i + 0.1)^2\pi^2 \, dx = \frac{\pi^2}{30},$$

$$Q3, i = 100\int_{0.1i}^{0.1i+0.1} (0.1i + 0.1 - x)^2 \pi^2 \, dx = \frac{\pi^2}{30},$$

$$Q4, i = 100\int_{0.1i-0.1}^{0.1i} dx = 10,$$

$$Q5, i = 10\int_{0.1i-0.1}^{0.1i+0.1} (x - 0.1i + 0.1) \;\; 2\pi^2 \sin \pi x \;\; dx$$

$$= -2\pi \cos 0.1\pi i + 20[\sin(0.1\pi i) - \sin(0.1i - 0.1)\pi)],$$

and

$$Q6,i = 10\int_{0.1i}^{0.1i+0.1} (0.1i + 0.1 - x) \;\; 2\pi^2 \, sin\, \pi x \; dx$$

$$= 2\pi \, \cos 0.1\pi i - 20[\sin((0.1i + 0.1)\pi) - \sin(0.1\pi i)].$$

The linear system *A***c** = **b** has

$a_{i,i} = 20 + \frac{\pi^2}{15}$, for each $i = 1,2,\ldots,9,$

$a_{i,i+1} = -10 + \frac{\pi^2}{60}$, for each $i = 1,2,\ldots,8,$

$a_{i,i-1} = -10 + \frac{\pi^2}{60}$, for each $i = 2,3,\ldots,9,$

And

$b_i = 40 \sin(0.1\pi i)[1 - \cos 0.1\pi]$, for each $i = 1,2,\ldots,9.$

The solution to the tridiagonal linear system is

$$c_9 = 0.3102866742, c_8 = 0.5902003271, c_7 = 0.8123410598,$$
$$c_6 = 0.9549641893, c_5 = 1.004108771, c_4 = 0.9549641893,$$
$$c_3 = 0.8123410598, c_2 = 0.5902003271, c_1 = 0.3102866742.$$

The piecewise-linear approximation is

$$\varphi(x) = \sum_{i=1}^{9} c_i \varphi_i(x),$$

and the actual solution to the boundary-value problem is *y(x)* = sin *πx*. Table (14) lists the error in the approximation at *xi*, for $each\ i = 1,\ldots,9.$

i	x_i	$\varphi(x_i)$	$y(x_i)$	$\lvert\varphi(x_i) - y(x_i)\rvert$
1	0.1	0.3102866742	0.3090169943	0.00127
2	0.2	0.5902003271	0.5877852522	0.00241
3	0.3	0.8123410598	0.8090169943	0.00332
4	0.4	0.9549641896	0.9510565162	0.00390
5	0.5	1.0041087710	1.0000000000	0.00411
6	0.6	0.9549641893	0.9510565162	0.00390
7	0.7	0.8123410598	0.8090169943	0.00332
8	0.8	0.5902003271	0.5877852522	0.00241
9	0.9	0.3102866742	0.3090169943	0.00127

Table (14)

It can be showing that the tridiagonal matrix A given by the pricewise linear basis functions is positive definite, so, the linear system is stable with respect to round off error. Under the hypotheses presented at the beginning of this section, we have $|\varphi(x) - y(x)| = O(h^2)$, for each *x* in [0, 1].

B-Spline Basis

The use of piecewise-linear basis functions results in an approximate solution to Eqs. (14) and (16) that is continuous but not differentiable on [0, 1]. A more sophisticated set of basis functions is required to construct an approximation that belongs to $C_0^2[0,1]$. These basis functions are similar to the cubic interpolatory splines . Recall that the cubic *interpolatory* spline S on the five nodes $x_0, x_1, x_2, x_3, and\ x_4$ for a function f is defined by:

(a) *S(x)* is a cubic polynomial, denoted $S_j(x)$, on the subinterval $[x_j, x_{j+1}]$ for each $j = 0,1,2,3$;

(b) $S_j(x_j) = f(x_j)\ and\ S_j(x_{j+1}) = f(x_{j+1})$ for each $j = 0,1,2,3$;

(c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ for each $j = 0,1,2$;

(d) $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ for each $j = 0,1,2$;

(e) $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ for each $j = 0,1,2$;

(f) One of the following sets of boundary conditions is satisfied:

(i) $S''(x_0) = S''(x_n) = 0$ (Natural (or free) boundary);

(ii) $S'(x_0) = f'(x_0)\ and\ S'(x_n) = f'(x_n)$ (Clamped boundary).

Since uniqueness of solution requires the number of constants in **(a)**, 16, to equal the number of conditions in **(b)** through **(f)**, only one of the boundary conditions in **(f)** can be specified for the interpolatory cubic splines.

The cubic spline functions we will use for our basis functions are called **B-splines**, or *bell-shaped splines*. These differ from interpolatory splines in that both sets of boundary conditions in **(f)** are satisfied. This requires the relaxation of two of the conditions in **(b)** through **(e)**. Since the spline must have two continuous derivatives on $[x_0, x_4]$, we delete two of the interpolation conditions from the description of the interpolatory splines. In particular, we modify condition **(b)** to **b.** $S(x_j) = f(x_j)$ for $j = 0,2,4$.

For example, the basic B-spline S defined next and shown in Figure 11.5 uses the equally spaced nodes

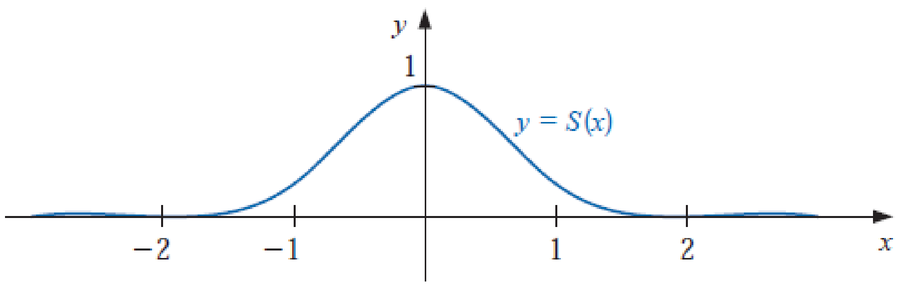
$$x_0 = -2, x_1 = -1, x_2 = 0, x_3 = 1, \text{and } x_4 = 2$$

It satisfies the interpolatory conditions **b.** $S(x_0) = 0, S(x_2) = 1, S(x_4) = 0$; as well as both sets of conditions

(i) $S''(x_0) = S''(x_4) = 0$ and

(ii) $S'(x_0) = S'(x_4) = 0$.

As a consequence, $S \in C_0^2(-\infty, \infty)$, and is given specifically as



$$S(x) = \begin{cases} 0 & \text{if} \quad x \le -2 \\ \frac{1}{4}(2+x)^3 & \text{if} \quad -2 \le x \le -1 \\ \frac{1}{4}[(2+x)^3 - 4(1+x)^3] & \text{if} \quad -1 < x \le 0 \\ \frac{1}{4}[(2-x)^3 - 4(1-x)^3] & \text{if} \quad 0 < x \le 1 \\ \frac{1}{4}(2-x)^3 & \text{if} \quad 1 \le x \le 2 \\ 0 & \text{if} \quad 2 < x \end{cases} \qquad (25)$$

We will now use this basic B-spline to construct the basis functions φ_i in $C_0^2[0,1]$. We first partition [0, 1] by choosing a positive integer n and defining $h = 1/(n+1)$. This produces the equally-spaced nodes

$$x_i = ih, \text{ for each } i = 0, 1, \dots, n+1.$$

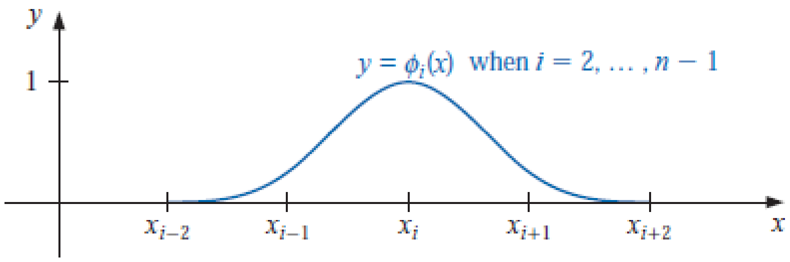
We then define the basis functions $\{\varphi_i\}_{i=0}^{n+1}$ as

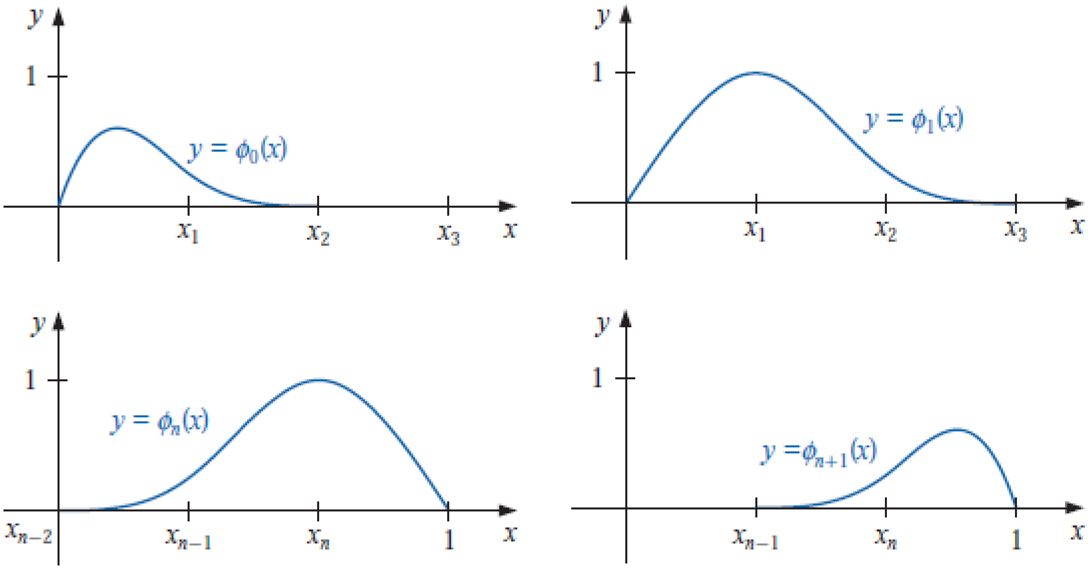
$$\varphi_i = \begin{cases} S\left(\frac{x}{h}\right) - 4S\left(\frac{x+h}{h}\right) & \text{if } i = 0 \\ S\left(\frac{x-h}{h}\right) - S\left(\frac{x+h}{h}\right) & \text{if } i = 1 \\ S\left(\frac{x-ih}{h}\right) & \text{if } 2 \le i \le n-1 \\ S\left(\frac{x-nh}{h}\right) - S\left(\frac{x-(n+2)h}{h}\right) & \text{if } i = n \\ S\left(\frac{x-(n+1)h}{h}\right) - 4S\left(\frac{x-(n+2)h}{h}\right) & \text{if } i = n+1 \end{cases}$$

It is not difficult to show that $\{\varphi_i\}_{i=0}^{n+1}$ is a linearly independent set of cubic splines satisfying

$$\varphi_i(0) = \varphi_i(1) = 0, \text{ for each } i = 0, 1, \dots, n, n+1$$

The graphs of φ_i, for $2 \le i \le n-1$, are shown in Figure 11.6, and the graphs of $\varphi_0, \varphi_1, \varphi_n$, *and* φ_{n+1} are in Figure 11.7.





Since $\varphi_i(x)$ and $\varphi'_i(x)$ are nonzero only for $x \in [x_{i-2}, x_{i+2}]$, the matrix in the Rayleigh-Ritz approximation is a band matrix with bandwidth at most seven:

$$A = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} & 0 & \cdots & & & 0 \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & \ddots & & & \vdots \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & \ddots & & \vdots \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & a_{n-2,n+1} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & a_{n-1,n+1} \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & a_{n,n+1} \\ 0 & \cdots & & 0 & a_{n+1,n-2} & a_{n+1,n-1} & a_{n+1,n} & & a_{n+1,n+1} \end{bmatrix}, \qquad (26)$$

where

$$a_{i,j} = \int_0^1 \left\{ p(x)\varphi'_i(x)\varphi'_j(x)(x) + q(x)\varphi_i(x)\varphi_j(x) \right\} dx,$$

for each $i, j = 0, 1, \ldots, n + 1$. The vector **b** has the entries

$$b_i = \int_0^1 f(x)\varphi_i(x)\, dx.$$

The matrix A is positive definite, so the linear system $A\mathbf{c} = \mathbf{b}$ can be solved by Cholesky's Algorithm or by Gaussian elimination.

Illustration

Consider the boundary-value problem

$$-y'' + \pi^2 y = 2\pi^2 \sin(\pi x), \text{for } 0 \le x \le 1, \text{with } y(0) = y(1) = 0.$$

we let $h = 0.1$ and generated approximations using piecewise-linear basis functions. Table (15) lists the results obtained by applying the B-splines

i	c_i	x_i	$\varphi(x_i)$	$y(x_i)$	$\|y(x_i) - \varphi(x_i)\|$
0	0.50964361×10^{-5}	0.0	0.00000000	0.00000000	0.00000000
1	0.20942608	0.1	0.30901644	0.30901699	0.00000055
2	0.39835678	0.2	0.58778549	0.58778525	0.00000024
3	0.54828946	0.3	0.80901687	0.80901699	0.00000012
4	0.64455358	0.4	0.95105667	0.95105652	0.00000015
5	0.67772340	0.5	1.00000002	1.00000000	0.00000020
6	0.64455370	0.6	0.95105713	0.95105652	0.00000061
7	0.54828951	0.7	0.80901773	0.80901699	0.00000074
8	0.39835730	0.8	0.58778690	0.58778525	0.00000165
9	0.20942593	0.9	0.30901810	0.30901699	0.00000111
10	0.74931285×10^{-5}	1.0	0.00000000	0.00000000	0.00000000

Table (15)

We recommend that the integrations in Steps 6 and 9 be performed in two steps. First, Construct cubic spline interpolatory polynomials for p, q, and f using the methods presented in Section 3.5. Then approximate the integrands by products of cubic splines or derivatives of cubic splines. The integrands are now piecewise polynomials and can be integrated exactly on each subinterval, and then summed. This leads to accurate approximations of the integrals.
The hypotheses assumed at the beginning of this section are sufficient to guarantee that

$$\left\{\int_0^1 |y(x) - \varphi(x)|^2 \, dx\right\}^{1/2} = O(h^4), \text{if } 0 \le x \le 1.$$

Another popular technique for solving boundary-value problems is the **method of collocation**. The word collocation has its root in the Latin "co-" and "locus "indicating together with and place. It is equivalent to what we call interpolation.
This procedure begins by selecting a set of basis functions $\{\varphi_1, \ldots, \varphi_N\}$, a set of numbers $\{x_i, \ldots, x_n\}$ in [0, 1], and requiring that an approximation $\sum_{i=1}^{N} c_i\, \varphi_i\,(x)$ satisfy the differential equation at each of the numbers x_i , for $1 \le i \le n$. If, in addition, it is required that $\varphi_i(0) = \varphi_i(1) = 0$, for $1 \le i \le N$, then the boundary conditions are automatically satisfied. Much attention in the literature has been given to the choice of the numbers $\{x_j\}$ and the basis functions$\{\varphi_i\}$. One popular choice is to let the φ_i be the basis functions for spline functions relative to a partition of $[0, 1]$, and to let the nodes $\{x_j\}$ be the Gaussian points or roots of certain orthogonal polynomials, transformed to the proper subinterval.

EXERCISE SET 11.5

1. Use the Piecewise Linear Algorithm to approximate the solution to the boundary-value problem

$$y'' + \frac{\pi^2}{4}y = \frac{\pi^2}{16}\cos\frac{\pi x}{4}, 0 \leq x \leq 1, y(0) = y(1) = 0$$

using $x_0 = 0,\ x_1 = 0.3,\ x_2 = 0.7,\ x_3 = 1$. Compare your results to the actual solution $y(x) = -\frac{1}{3}\cos\frac{\pi}{2}x - \frac{\sqrt{2}}{6}\sin\frac{\pi}{2}x + \frac{1}{3}\cos\frac{\pi}{4}x$.

2. Use the Piecewise Linear Algorithm to approximate the solution to the boundary-value problem

$$-\frac{d}{dx}(xy') + 4y = 4x2 - 8x + 1, 0 \leq x \leq 1, y(0) = y(1) = 0$$

using $x_0 = 0,\ x_1 = 0.4,\ x_2 = 0.8,\ x_3 = 1$. Compare your results to the actual solution $y(x) = x^2 - x$.

3. Use the Piecewise Linear Algorithm to approximate the solutions to the following boundary-value problems, and compare the results to the actual solution:

a. $-x^2y'' - 2xy' + 2y = -4x^2, 0 \leq x \leq 1, y(0) = y(1) = 0;$
use $h = 0.1$; actual solution $y(x) = x^2 - x$.

b. $-(x+1)y'' - y' + (x+2)y = [2 - (x+1)^2]e\ln 2 - 2e^x,$
$0 \leq x \leq 1, y(0) = y(1) = 0$; use $h = 0.05$;
actual solution $y(x) = e^x \ln(x+1) - (e\ln 2)x$.

4. Use the Cubic Spline Algorithm with $n = 3$ to approximate the solution to each of the following boundary-value problems, and compare the results to the actual solutions given in Exercises 1 and 2:

a. $y'' + \frac{\pi^2}{4}y = \frac{\pi^2}{16}\cos\frac{\pi x}{4}, 0 \leq x \leq 1, y(0) = y(1) = 0$

b. $-\frac{d}{dx}(xy') + 4y = 4x2 - 8x + 1, 0 \leq x \leq 1, y(0) = 0, y(1) = 0$

5. Show that the boundary-value problem

$$-\frac{d}{dx}(p(x)y') + q(x)y = f(x), 0 \leq x \leq 1, y(0) = \alpha, y(1) = \beta,$$

can be transformed by the change of variable
$z = y - \beta x - (1 - x)\alpha$ into the form

$$-\frac{d}{dx}(p(x)z') + q(x)z = F(x), 0 \leq x \leq 1, z(0) = 0, z(1) = 0.$$

6. Use Exercise 6 and the Piecewise Linear Algorithm with $n = 9$ to approximate the solution to the boundary-value problem

$$-y'' + y = x, 0 \leq x \leq 1, y(0) = 1, y(1) = 1 + e^{-1}.$$

Reference

[1] Richard L. Burden & J. Douglas Faires,"*Numerical Analysis*",9th edition,(2011).