Numerical Analysis

10th ed

R L Burden, J D Faires, and A M Burden

Beamer Presentation Slides
Prepared by
Dr. Annette M. Burden
Youngstown State University

August 9, 2015

Definition (5.1)

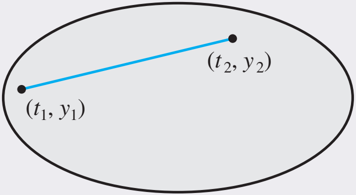
A function $f(t, y)$ is said to satisfy a **Lipschitz condition** in the variable y on a set $D \subset \Re^2$ if a constant $L > 0$ exists with

$$|f(t, y_1) - f(t, y_2,)| \leq L|y_1 - y_2|,$$

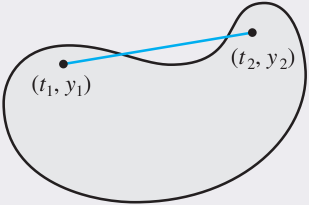
whenever (t, y_1) and (t, y_2) are in D. The constant L is called a **Lipschitz constant** for f.

Definition (5.2)

A set $D \subset \Re^2$ is said to be **convex** if whenever (t_1, y_1) and (t_2, y_2) belong to D, then $((1-\lambda)t_1 + \lambda t_2, (1-\lambda)y_1 + \lambda y_2)$ also belongs to D for every λ in $[0, 1]$.



Convex



Not convex

Theorem (5.3)

Suppose $f(t, y)$ is defined on a convex set $D \subset \Re^2$. If a constant $L > 0$ exists with

$$\left|\frac{\partial f}{\partial y}(t, y)\right| \leq L, \quad \text{for all } (t, y) \in D,$$

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

Theorem (5.4)

Suppose that $D = \{ (t, y) \mid a \leq t \leq b \text{ and } -\infty < y < \infty \}$ *and that* $f(t, y)$ *is continuous on* D*. If* f *satisfies a Lipschitz condition on* D *in the variable* y*, then the initial-value problem*

$$y'(t) = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha,$$

has a unique solution $y(t)$ *for* $a \leq t \leq b$*.*

Definition (5.5)

The initial-value problem

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

is said to be a **well-posed problem** if:

- A unique solution, $y(t)$, to the problem exists, and
- There exist constants $\varepsilon_0 > 0$ and $k > 0$ such that for any ε, in $(0, \varepsilon_0)$, whenever $\delta(t)$ is continuous with $|\delta(t)| < \varepsilon$ for all t in $[a, b]$, and when $|\delta_0| < \varepsilon$, the initial-value problem (perturbed problem)

$$\frac{dz}{dt} = f(t, z) + \delta(t), \quad a \le t \le b, \quad z(a) = \alpha + \delta_0,$$

has a unique solution $z(t)$ that satisfies

$$|z(t) - y(t)| < k\varepsilon \quad \text{for all } t \text{ in } [a, b].$$

Theorem (5.6)

Suppose $D = \{\,(t, y) \mid a \le t \le b \text{ and } -\infty < y < \infty\,\}$. If f is continuous and satisfies a Lipschitz condition in the variable y on the set D, then the initial-value problem

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

is well-posed.

The object of Euler's method is to obtain approximations to the well-posed initial-value problem

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

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

For equally distributed mesh points throughout the interval $[a, b]$, choose a positive integer N and select the mesh points

$$t_i = a + ih, \quad \text{for each } i = 0, 1, 2, \ldots, N.$$

The common distance between the points $h = (b - a)/N = t_{i+1} - t_i$ is called the **step size.** Euler's method constructs $w_i \approx y(t_i) = y(t_i) + hf((t_i, y(t_i)) + \frac{h^2}{2}y''(\xi_i)$, for each $i = 1, 2, \ldots, N$, by deleting the remainder term. Euler's method is $w_0 = \alpha$, with **difference equation**

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

Algorithm 5.1: EULER'S METHOD

To approximate the solution of the initial-value problem

$$y' = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha,$$

at $(N + 1)$ equally spaced numbers in the interval $[a, b]$:

INPUT endpoints a, b; integer N; initial condition α.

OUTPUT approximation w to y at the $(N + 1)$ values of t.

Step 1 Set $h = (b - a)/N$; $t = a$; $w = \alpha$;
OUTPUT (t, w).

Step 2 For $i = 1, 2, \ldots, N$ do Steps 3, 4.

Step 3 Set $w = w + hf(t, w)$; (*Compute* w_i.)
$t = a + ih$. (*Compute* t_i.)

Step 4 OUTPUT (t, w).

Step 5 STOP.

Lemma (5.7)

For all $x \geq -1$ *and any positive* m*, we have* $0 \leq (1+x)^m \leq e^{mx}$.

Lemma (5.8)

If s *and* t *are positive real numbers,* $\{a_i\}_{i=0}^{k}$ *is a sequence satisfying* $a_0 \geq -t/s$*, and*

$$a_{i+1} \leq (1+s)a_i + t, \quad \text{for each } i = 0, 1, 2, \ldots, k-1,$$

then

$$a_{i+1} \leq e^{(i+1)s}\left(a_0 + \frac{t}{s}\right) - \frac{t}{s}.$$

Theorem (5.9)

Suppose f is continuous and satisfies a Lipschitz condition with constant L on

$$D = \{\ (t, y) \mid a \le t \le b \text{ and } -\infty < y < \infty\ \}$$

and that a constant M exists with

$$|y''(t)| \le M, \quad \text{for all } t \in [a, b],$$

where $y(t)$ denotes the unique solution to the initial-value problem

$$y' = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha.$$

Let $w_0, w_1, \ldots, w_N$ be the approximations generated by Euler's method for some positive integer N. Then, for each $i = 0, 1, 2, \ldots, N$,

$$|y(t_i) - w_i| \le \frac{hM}{2L}\left[e^{L(t_i - a)} - 1\right].$$

Theorem (5.10)

Let $y(t)$ denote the unique solution to the initial-value problem

$$y' = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha$$

and $u_0, u_1, \ldots, u_N$ be the approximations obtained using

$$\begin{aligned} u_0 &= \alpha + \delta_0, \\ u_{i+1} &= u_i + hf(t_i, u_i) + \delta_{i+1}, \quad \text{for each } i = 0, 1, \ldots, N-1, \end{aligned}$$

. If $|\delta_i| < \delta$ for each $i = 0, 1, \ldots, N$ and the hypotheses of Theorem 5.9 hold for (5.12), then

$$|y(t_i) - u_i| \leq \frac{1}{L}\left(\frac{hM}{2} + \frac{\delta}{h}\right)[e^{L(t_i - a)} - 1] + |\delta_0| e^{L(t_i - a)},$$

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

Definition (5.11)

The difference method

$$w_0 = \alpha$$
$$w_{i+1} = w_i + h\phi(t_i, w_i), \quad \text{for each } i = 0, 1, \ldots, N-1,$$

has **local truncation error**

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

for each $i = 0, 1, \ldots, N-1$, where y_i and y_{i+1} denote the solution of the differential equation at t_i and t_{i+1}, respectively.

Taylor method of order n

The difference method

$$w_0 = \alpha$$
$$w_{i+1} = w_i + h\phi(t_i, w_i), \quad \text{for each } i = 0, 1, \ldots, N-1,$$

has **local truncation error**

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

for each $i = 0, 1, \ldots, N-1$, where y_i and y_{i+1} denote the solution of the differential equation at t_i and t_{i+1}, respectively.

Theorem (5.12)

If Taylor's method of order n is used to approximate the solution to

$$y'(t) = f(t, y(t)), \quad a \leq t \leq b, \quad y(a) = \alpha,$$

with step size h and if $y \in C^{n+1}[a, b]$*, then the local truncation error is* $O(h^n)$*.*

Theorem (5.13)

Suppose that $f(t, y)$ and all its partial derivatives of order less than or equal to $n+1$ are continuous on $D = \{ (t, y) \mid a \le t \le b, c \le y \le d \}$, and let $(t_0, y_0) \in D$. For every $(t, y) \in D$, there exists ξ between t and t_0 and μ between y and y_0 with $f(t, y) = P_n(t, y) + R_n(t, y)$, where

$$\begin{aligned} P_n(t, y) \quad = \quad & f(t_0, y_0) + \left[(t - t_0)\frac{\partial f}{\partial t}(t_0, y_0) + (y - y_0)\frac{\partial f}{\partial y}(t_0, y_0) \right] \\ & + \left[\frac{(t - t_0)^2}{2}\frac{\partial^2 f}{\partial t^2}(t_0, y_0) + (t - t_0)(y - y_0)\frac{\partial^2 f}{\partial t \partial y}(t_0, y_0) \right. \\ & \left. + \frac{(y - y_0)^2}{2}\frac{\partial^2 f}{\partial y^2}(t_0, y_0) \right] + \cdots \\ & + \left[\frac{1}{n!}\sum_{j=0}^{n} \binom{n}{j} (t - t_0)^{n-j}(y - y_0)^j \frac{\partial^n f}{\partial t^{n-j}\partial y^j}(t_0, y_0) \right] \end{aligned}$$

and

$$R_n(t, y) = \frac{1}{(n+1)!}\sum_{j=0}^{n+1} \binom{n+1}{j} (t - t_0)^{n+1-j}(y - y_0)^j \frac{\partial^{n+1} f}{\partial t^{n+1-j}\partial y^j}(\xi, \mu).$$

Midpoint Method

$$w_0 = \alpha,$$
$$w_{i+1} = w_i + hf\left(t_i + \frac{h}{2}, w_i + \frac{h}{2}f(t_i, w_i)\right), \text{ for } i = 0, \ldots, N-1.$$

Modified Euler Method

$$w_0 = \alpha,$$
$$w_{i+1} = w_i + \frac{h}{2}[f(t_i, w_i) + f(t_{i+1}, w_i + hf(t_i, w_i))], \ i = 0, \ldots, N-1.$$

Runge-Kutta Order Four

$$
\begin{aligned}
w_0 &= \alpha, \\
k_1 &= hf(t_i, w_i), \\
k_2 &= hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_1\right), \\
k_3 &= hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_2\right), \\
k_4 &= hf(t_{i+1}, w_i + k_3), \\
w_{i+1} &= w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),
\end{aligned}
$$

for each $i = 0, 1, \ldots, N-1$. This method has local truncation error $O(h^4)$, provided the solution $y(t)$ has five continuous derivatives. We introduce the notation k_1, k_2, k_3, k_4 into the method to eliminate the need for successive nesting in the second variable of $f(t, y)$.

Algorithm 5.2: RUNGE-KUTTA METHOD (order four)

To approximate the solution of the initial-value problem

$$y' = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha,$$

at $(N + 1)$ equally spaced numbers in the interval $[a, b]$:

INPUT endpoints a, b; integer N; initial condition α.

OUTPUT approximation w to y at the $(N + 1)$ values of t.

Step 1 Set $h = (b - a)/N$; $t = a$; $w = \alpha$; OUTPUT (t, w).
Step 2 For $i = 1, 2, \ldots, N$ do Steps 3–5.
Step 3 Set $K_1 = hf(t, w)$; $K_2 = hf(t + h/2, w + K_1/2)$;
$K_3 = hf(t + h/2, w + K_2/2)$; $K_4 = hf(t + h, w + K_3)$.
Step 4 Set $w = w + (K_1 + 2K_2 + 2K_3 + K_4)/6$; (*Compute* w_i.)
$t = a + ih$. (*Compute* t_i.)
Step 5 OUTPUT (t, w).
Step 6 STOP.

Computational Comparisons: Table 5.9

Evaluations per step	2	3	4	$5 \leq n \leq 7$	$8 \leq n \leq 9$	$10 \leq n$
Best local trunc. error	$O(h^2)$	$O(h^3)$	$O(h^4)$	$O(h^{n-1})$	$O(h^{n-2})$	$O(h^{n-3})$

Algorithm 5.3: RUNGE-KUTTA-FEHLBERG METHOD

To approximate the solution of the initial-value problem

$$y' = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha,$$

with local truncation error within a given tolerance:

INPUT endpoints a, b; initial condition α; tolerance TOL; maximum step size $hmax$; minimum step size $hmin$.

OUTPUT t, w, h where w approximates $y(t)$ and the step size h was used, or a message that the minimum step size was exceeded.

Step 1 Set $t = a$;
$w = \alpha$;
$h = hmax$;
$FLAG = 1$;
OUTPUT (t, w).

Algorithm 5.3: RUNGE-KUTTA-FEHLBERG METHOD CONT.

Step 2 While ($FLAG = 1$) do Steps 3–11.

Step 3 Set $K_1 = hf(t, w)$;

$$K_2 = hf\left(t + \frac{1}{4}h, w + \frac{1}{4}K_1\right);$$
$$K_3 = hf\left(t + \frac{3}{8}h, w + \frac{3}{32}K_1 + \frac{9}{32}K_2\right);$$
$$K_4 = hf\left(t + \frac{12}{13}h, w + \frac{1932}{2197}K_1 - \frac{7200}{2197}K_2 + \frac{7296}{2197}K_3\right);$$
$$K_5 = hf\left(t + h, w + \frac{439}{216}K_1 - 8K_2 + \frac{3680}{513}K_3 - \frac{845}{4104}K_4\right);$$
$$K_6 = hf\left(t + \frac{1}{2}h, w - \frac{8}{27}K_1 + 2K_2 - \frac{3544}{2565}K_3 + \frac{1859}{4104}K_4 - \frac{11}{40}K_5\right).$$

Step 4 Set $R = \frac{1}{h}\left|\frac{1}{360}K_1 - \frac{128}{4275}K_3 - \frac{2197}{75240}K_4 + \frac{1}{50}K_5 + \frac{2}{55}K_6\right|$.

(*Note:* $R = \frac{1}{h}|\tilde{w}_{i+1} - w_{i+1}| \approx |\tau_{i+1}(h)|$.)

Step 5 If $R \le TOL$ then do Steps 6 and 7.

Step 6 Set $t = t + h$; (*Approximation accepted.*)

$$w = w + \frac{25}{216}K_1 + \frac{1408}{2565}K_3 + \frac{2197}{4104}K_4 - \frac{1}{5}K_5.$$

Step 7 OUTPUT (t, w, h). (*End Step 5*)

Algorithm 5.3: RUNGE-KUTTA-FEHLBERG METHOD CONT.

Step 8 Set $\delta = 0.84(TOL/R)^{1/4}$.

Step 9 If $\delta \leq 0.1$ then set $h = 0.1h$
  else if $\delta \geq 4$ then set $h = 4h$
  else set $h = \delta h$. (*Calculate new h.*)

Step 10 If $h > hmax$ then set $h = hmax$.

Step 11 If $t \geq b$ then set $FLAG = 0$
  else if $t + h > b$ then set $h = b - t$
  else if $h < hmin$ then
  set $FLAG = 0$;
  OUTPUT (‘*minimum h exceeded*’).
  (*Procedure unsuccessful.*)(*End Step 3*)

Step 12 (*The procedure is complete.*)
  STOP.

Definition (5.14)

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

$$y' = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha,$$

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

$$\begin{aligned} w_{i+1} \quad = \quad & a_{m-1} w_i + a_{m-2} w_{i-1} + \cdots + a_0 w_{i+1-m} \\ & + h[b_m f(t_{i+1}, w_{i+1}) + b_{m-1} f(t_i, w_i) \\ & + \cdots + b_0 f(t_{i+1-m}, w_{i+1-m})], \end{aligned}$$

for $i = m-1, m, \ldots, N-1$, where $h = (b-a)/N$, the $a_0, a_1, \ldots, a_{m-1}$ and $b_0, b_1, \ldots, b_m$ are constants, and the starting values

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

are specified.

Definition (5.15)

If $y(t)$ is the solution to the initial-value problem

$$y' = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha,$$

and

$$\begin{aligned} w_{i+1} \quad = \quad & a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} \\ & + h[b_m f(t_{i+1}, w_{i+1}) + b_{m-1} f(t_i, w_i) + \cdots + b_0 f(t_{i+1-m}, w_{i+1-m})] \end{aligned}$$

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

$$\begin{aligned} \tau_{i+1}(h) \quad = \quad & \frac{y(t_{i+1}) - a_{m-1}y(t_i) - \cdots - a_0 y(t_{i+1-m})}{h} \\ & - [b_m f(t_{i+1}, y(t_{i+1})) + \cdots + b_0 f(t_{i+1-m}, y(t_{i+1-m}))], \end{aligned}$$

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

Definition (Adams-Bashforth Two-Step Explicit Method)

$$\begin{aligned} w_0 &= \alpha, \quad w_1 = \alpha_1, \\ w_{i+1} &= w_i + \frac{h}{2}[3f(t_i, w_i) - f(t_{i-1}, w_{i-1})], \end{aligned}$$

where $i = 1, 2, \ldots, N-1$. The local truncation error is $\tau_{i+1}(h) = \frac{5}{12}y'''(\mu_i)h^2$, for some $\mu_i \in (t_{i-1}, t_{i+1})$.

Definition (Adams-Bashforth Three-Step Explicit Method)

$$\begin{aligned} w_0 &= \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2, \\ w_{i+1} &= w_i + \frac{h}{12}[23f(t_i, w_i) - 16f(t_{i-1}, w_{i-1}) + 5f(t_{i-2}, w_{i-2})], \end{aligned}$$

where $i = 2, 3, \ldots, N-1$. The local truncation error is $\tau_{i+1}(h) = \frac{3}{8}y^{(4)}(\mu_i)h^3$, for some $\mu_i \in (t_{i-2}, t_{i+1})$.

Definition (Adams-Bashforth Four-Step Explicit Method)

$$\begin{aligned} w_0 &= \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2, \quad w_3 = \alpha_3, \\ w_{i+1} &= w_i + \frac{h}{24}[55f(t_i, w_i) - 59f(t_{i-1}, w_{i-1}) + 37f(t_{i-2}, w_{i-2}) \\ &- 9f(t_{i-3}, w_{i-3})], \end{aligned}$$

where $i = 3, 4, \ldots, N-1$. The local truncation error is $\tau_{i+1}(h) = \frac{251}{720} y^{(5)}(\mu_i) h^4$, for some $\mu_i \in (t_{i-3}, t_{i+1})$.

Definition (Adams-Bashforth Five-Step Explicit Method)

$$
\begin{aligned}
w_0 &= \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2, \quad w_3 = \alpha_3, \quad w_4 = \alpha_4, \\
w_{i+1} &= w_i + \frac{h}{720}[1901 f(t_i, w_i) - 2774 f(t_{i-1}, w_{i-1}) \\
&\quad + 2616 f(t_{i-2}, w_{i-2}) - 1274 f(t_{i-3}, w_{i-3}) + 251 f(t_{i-4}, w_{i-4})],
\end{aligned}
$$

where $i = 4, 5, \ldots, N-1$. The local truncation error is $\tau_{i+1}(h) = \frac{95}{288} y^{(6)}(\mu_i) h^5$, for some $\mu_i \in (t_{i-4}, t_{i+1})$.

Definition (Adams-Moulton Two-Step Implicit Method)

$$\begin{aligned} w_0 &= \alpha, \quad w_1 = \alpha_1, \\ w_{i+1} &= w_i + \frac{h}{12}[5f(t_{i+1}, w_{i+1}) + 8f(t_i, w_i) - f(t_{i-1}, w_{i-1})], \end{aligned}$$

where $i = 1, 2, \ldots, N-1$. The local truncation error is $\tau_{i+1}(h) = -\frac{1}{24}y^{(4)}(\mu_i)h^3$, for some $\mu_i \in (t_{i-1}, t_{i+1})$.

Definition (Adams-Moulton Three-Step Implicit Method)

$$\begin{aligned} w_0 &= \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2, \\ w_{i+1} &= w_i + \frac{h}{24}[9f(t_{i+1}, w_{i+1}) + 19f(t_i, w_i) - 5f(t_{i-1}, w_{i-1}) + f(t_{i-2}, w_{i-2})], \end{aligned}$$

where $i = 2, 3, \ldots, N-1$. The local truncation error is $\tau_{i+1}(h) = -\frac{19}{720}y^{(5)}(\mu_i)h^4$, for some $\mu_i \in (t_{i-2}, t_{i+1})$.

Definition (Adams-Moulton Four-Step Implicit Method)

$$
\begin{aligned}
w_0 &= \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2, \quad w_3 = \alpha_3, \\
w_{i+1} &= w_i + \frac{h}{720}[251f(t_{i+1}, w_{i+1}) + 646f(t_i, w_i) \\
&\quad - 264f(t_{i-1}, w_{i-1}) + 106f(t_{i-2}, w_{i-2}) \\
&\quad -19f(t_{i-3}, w_{i-3})],
\end{aligned}
$$

where $i = 3, 4, \ldots, N-1$. The local truncation error is

$$\tau_{i+1}(h) = -\frac{3}{160} y^{(6)}(\mu_i) h^5,$$

for some $\mu_i \in (t_{i-3}, t_{i+1})$.

Algorithm 5.4: ADAMS FOURTH-ORDER PREDICTOR-CORRECTOR

To approximate the solution of the initial-value problem

$$y' = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha,$$

at $(N+1)$ equally spaced numbers in the interval $[a, b]$:

INPUT endpoints a, b; integer N; initial condition α.

OUTPUT approximation w to y at the $(N+1)$ values of t.

Step 1 Set $h = (b-a)/N$; $t_0 = a$; $w_0 = \alpha$; OUTPUT (t_0, w_0).
Step 2 For $i = 1, 2, 3$, do Steps 3–5.
(*Compute starting values using Runge-Kutta method.*)
Step 3 Set $K_1 = hf(t_{i-1}, w_{i-1})$;
$K_2 = hf(t_{i-1} + h/2, w_{i-1} + K_1/2)$;
$K_3 = hf(t_{i-1} + h/2, w_{i-1} + K_2/2)$;
$K_4 = hf(t_{i-1} + h, w_{i-1} + K_3)$.

Algorithm 5.4: ADAMS FOURTH-ORDER PREDICTOR-CORRECTOR

Step 4 Set $w_i = w_{i-1} + (K_1 + 2K_2 + 2K_3 + K_4)/6;$
$t_i = a + ih.$

Step 5 OUTPUT (t_i, w_i).

Step 6 For $i = 4, \ldots, N$ do Steps 7–10.

Step 7 Set $t = a + ih;$

$$w = w_3 + h[55f(t_3, w_3) - 59f(t_2, w_2) + 37f(t_1, w_1) - 9f(t_0, w_0)]/24; \quad (\textit{Predict} \quad w_i.)$$

$$w = w_3 + h[9f(t, w) + 19f(t_3, w_3) - 5f(t_2, w_2) + f(t_1, w_1)]/24. \quad (\textit{Correct} \quad w_i.)$$

Step 8 OUTPUT (t, w).

Step 9 For $j = 0, 1, 2$
set $t_j = t_{j+1};$ (*Prepare for next iteration.*)
$w_j = w_{j+1}.$

Step 10 Set $t_3 = t;$
$w_3 = w.$

Step 11 STOP.

Algorithm 5.5: ADAMS VARIABLE STEP-SIZE PREDICTOR-CORRECTOR

To approximate the solution of the initial-value problem

$$y' = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha$$

with local truncation error within a given tolerance:

INPUT endpoints a, b; initial condition α; tolerance TOL; maximum step size $hmax$; minimum step size $hmin$.

OUTPUT i, t_i, w_i, h where at the ith step w_i approximates $y(t_i)$ and the step size h was used, or a message that the minimum step size was exceeded.

Step 1 Set up a subalgorithm for the Runge-Kutta fourth-order method to be called $RK4(h, v_0, x_0, v_1, x_1, v_2, x_2, v_3, x_3)$ that accepts as input a step size h and starting values $v_0 \approx y(x_0)$ and returns $\{(x_j, v_j) \mid j = 1, 2, 3\}$ defined by the following: for $j = 1, 2, 3$ set

$K_1 = hf(x_{j-1}, v_{j-1}); \quad K_2 = hf(x_{j-1} + h/2, v_{j-1} + K_1/2)$

$K_3 = hf(x_{j-1} + h/2, v_{j-1} + K_2/2); \quad K_4 = hf(x_{j-1} + h, v_{j-1} + K_3)$

$v_j = v_{j-1} + (K_1 + 2K_2 + 2K_3 + K_4)/6; \quad x_j = x_0 + jh.$

Algorithm 5.5: ADAMS VARIABLE STEP-SIZE PREDICTOR-CORRECTOR

Step 2 Set $t_0 = a$;
$w_0 = \alpha$;
$h = hmax$;
$FLAG = 1$; (*FLAG will be used to exit the loop in Step 4.*)
$LAST = 0$; (*LAST will indicate when the last value is calculated.*)
OUTPUT (t_0, w_0).

Step 3 Call $RK4(h, w_0, t_0, w_1, t_1, w_2, t_2, w_3, t_3)$;
Set $NFLAG = 1$; (*Indicates computation from RK4.*)
$i = 4$; $t = t_3 + h$.

Step 4 While ($FLAG = 1$) do Steps 5–20.

Step 5 Set
$$WP = w_{i-1} + \frac{h}{24}[55f(t_{i-1}, w_{i-1}) - 59f(t_{i-2}, w_{i-2}) + 37f(t_{i-3}, w_{i-3}) - 9f(t_{i-4}, w_{i-4})]; \quad (\textit{Predict } w_i.)$$
$$WC = w_{i-1} + \tfrac{h}{24}[9f(t, WP) + 19f(t_{i-1}, w_{i-1}) - 5f(t_{i-2}, w_{i-2}) + f(t_{i-3}, w_{i-3})]; \quad (\textit{Correct } w_i.)$$
$$\sigma = 19|WC - WP|/(270h).$$

Step 6 If $\sigma \leq TOL$ then do Steps 7–16 (*Result accepted.*)
else do Steps 17–19. (*Result rejected.*)

Algorithm 5.5: ADAMS VARIABLE STEP-SIZE PREDICTOR-CORRECTOR

Step 7 Set $w_i = WC$; (*Result accepted.*) $t_i = t$.

Step 8 If $NFLAG = 1$ then for $j = i-3, i-2, i-1, i$

OUTPUT (j, t_j, w_j, h);

(*Previous results also accepted.*)

else OUTPUT (i, t_i, w_i, h).

(*Previous results already accepted.*)

Step 9 If $LAST = 1$ then set $FLAG = 0$ (*Next step is* 20.)

else do Steps 10–16.

Step 10 Set $i = i + 1$;

$NFLAG = 0$.

Step 11 If $\sigma \leq 0.1\ TOL$ or $t_{i-1} + h > b$ then do Steps 12–16.

(*Increase h if it is more accurate than required or decrease h to include b as a mesh point.*)

Step 12 Set $q = (TOL/(2\sigma))^{1/4}$.

Step 13 If $q > 4$ then set $h = 4h$

else set $h = qh$.

Algorithm 5.5: ADAMS VARIABLE STEP-SIZE PREDICTOR-CORRECTOR

Step 14 If $h > hmax$ then set $h = hmax$.
Step 15 If $t_{i-1} + 4h > b$ then
set $h = (b - t_{i-1})/4$;
$LAST = 1$.
Step 16 Call $RK4(h, w_{i-1}, t_{i-1}, w_i, t_i, w_{i+1}, t_{i+1}, w_{i+2}, t_{i+2})$;
Set $NFLAG = 1$;
$i = i + 3$. (*True branch done; End Step 6; Next step 20.*)
Step 17 Set $q = (TOL/(2\sigma))^{1/4}$. (*False branch of Step 6: Result rejected.*)
Step 18 If $q < 0.1$ then set $h = 0.1h$
else set $h = qh$.
Step 19 If $h < hmin$ then set $FLAG = 0$;
OUTPUT ('*hmin* exceeded') else
if $NFLAG = 1$ then set $i = i - 3$;
(*Previous results also rejected.*)
Call $RK4(h, w_{i-1}, t_{i-1}, w_i, t_i, w_{i+1}, t_{i+1}, w_{i+2}, t_{i+2})$;
set $i = i + 3$; $NFLAG = 1$.(*End Step 6.*)
Step 20 Set $t = t_{i-1} + h$.(*End Step 4.*)
Step 21 STOP.

Algorithm 5.6: EXTRAPOLATION

To approximate the solution of the initial-value problem

$$y' = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha,$$

with local truncation error within a given tolerance:

INPUT endpoints a, b; initial condition α; tolerance TOL; maximum step size $hmax$; minimum step size $hmin$.

OUTPUT T, W, h where W approximates $y(t)$ and step size h was used, or a message that minimum step size was exceeded.

Step 1 Initialize the array $NK = (2, 4, 6, 8, 12, 16, 24, 32)$.
Step 2 Set $TO = a$; $WO = \alpha$;
$h = hmax$; $FLAG = 1$. (*FLAG is used to exit the loop in Step* 4.)
Step 3 For $i = 1, 2, \ldots, 7$
for $j = 1, \ldots, i$
set $Q_{i,j} = (NK_{i+1}/NK_j)^2$. (*Note:* $Q_{i,j} = h_j^2/h_{i+1}^2$.)

Algorithm 5.6: EXTRAPOLATION

Step 4 While ($FLAG = 1$) do Steps 5–20.

Step 5 Set $k = 1$;

$NFLAG = 0$. (*When desired accuracy is achieved, NFLAG is set to* 1.)

Step 6 While ($k \leq 8$ and $NFLAG = 0$) do Steps 7–14.

Step 7 Set $HK = h/NK_k$; $T = TO$; $W2 = WO$;

$W3 = W2 + HK \cdot f(T, W2)$; (*Euler's first step.*)

$T = TO + HK$.

Step 8 For $j = 1, \ldots, NK_k - 1$

set $W1 = W2$; $W2 = W3$;

$W3 = W1 + 2HK \cdot f(T, W2)$; (*Midpoint method.*)

$T = TO + (j + 1) \cdot HK$.

Step 9 Set $y_k = [W3 + W2 + HK \cdot f(T, W3)]/2$.

(*Endpoint correction to compute* $y_{k,1}$.)

Step 10 If $k \geq 2$ then do Steps 11–13.

(*Note:* $y_{k-1} \equiv y_{k-1,1}, y_{k-2} \equiv y_{k-2,2}, \ldots, y_1 \equiv y_{k-1,k-1}$ *since only the previous row of the table is saved.*)

Step 11 Set $j = k$;

$v = y_1$. (*Save* $y_{k-1,k-1}$.)

Algorithm 5.6: EXTRAPOLATION

Step 12 While ($j \geq 2$) do set $y_{j-1} = y_j + \frac{y_j - y_{j-1}}{Q_{k-1,j-1} - 1}$;

(*Extrapolation to compute* $y_{j-1} \equiv y_{k,k-j+2}$.)

$\left(\textit{Note:} \quad y_{j-1} = \frac{h_{j-1}^2 y_j - h_k^2 y_{j-1}}{h_{j-1}^2 - h_k^2} . \right)$; $j = j - 1$.

Step 13 If $|y_1 - v| \leq TOL$ then set $NFLAG = 1$.

(y_1 *is accepted as the neww*.)

Step 14 Set $k = k + 1$. (*End Step 6*)

Step 15 Set $k = k - 1$. (*Part of Step 4*)

Step 16 If $NFLAG = 0$ then do Steps 17 and 18 (*Result rejected.*)

else do Steps 19 and 20. (*Result accepted.*)

Step 17 Set $h = h/2$. (*New value for w rejected, decrease h.*)

Step 18 If $h < hmin$ then OUTPUT ('*hmin* exceeded');

Set $FLAG = 0$. (*End Step 16*)

(*True branch completed, next step is back to Step* 4.)

Step 19 Set $WO = y_1$; (*New value for w accepted.*)

$TO = TO + h$; (TO, WO, h).

Algorithm 5.6: EXTRAPOLATION

Step 20 If $TO \geq b$ then set $FLAG = 0$
(*Procedure completed successfully.*)
else if $TO + h > b$ then set $h = b - TO$
(*Terminate at* $t = b$.)
else if ($k \leq 3$ and $h < 0.5(hmax)$ set $h = 2h$.
(*Increase step size if possible.*) (*End of Step 4 and 16*)

Step 21 STOP.

Definition (5.16)

The function $f(t, y_1, \ldots, y_m)$, defined on the set

$$D = \{ (t, u_1, \ldots, u_m) \mid a \leq t \leq b; -\infty < u_i < \infty, \text{for each } i = 1, \ldots, m \}$$

is said to satisfy a **Lipschitz condition** on D in the variables $u_1, u_2, \ldots, u_m$ if a constant $L > 0$ exists with

$$|f(t, u_1, \ldots, u_m) - f(t, z_1, \ldots, z_m)| \leq L \sum_{j=1}^{m} |u_j - z_j|,$$

for all $(t, u_1, \ldots, u_m)$ and $(t, z_1, \ldots, z_m)$ in D.

Theorem (5.17)

Suppose that

$$D = \{ (t, u_1, \ldots, u_m) \mid a \le t \le b; -\infty < u_i < \infty, \text{for each } i = 1, \ldots, m \}$$

and let $f_i(t, u_1, \ldots, u_m)$, for each $i = 1, 2, \ldots, m$, be continuous and satisfy a Lipschitz condition on D. The system of first-order differential equations (5.45), subject to the initial conditions (5.46), has a unique solution $u_1(t), \ldots, u_m(t)$, for $a \le t \le b$.

Algorithm 5.7: RUNGE-KUTTA METHOD FOR SYSTEMS OF DE

Step 1 Set $h = (b-a)/N;\ t = a$.
Step 2 For $j = 1, 2, \ldots, m$ set $w_j = \alpha_j$.
Step 3 OUTPUT $(t, w_1, w_2, \ldots, w_m)$.
Step 4 For $i = 1, 2, \ldots, N$ do steps 5–11.
  Step 5 For $j = 1, 2, \ldots, m$ set
    $k_{1,j} = hf_j(t, w_1, w_2, \ldots, w_m)$.
  Step 6 For $j = 1, 2, \ldots, m$ set
    $k_{2,j} = hf_j\left(t + \frac{h}{2}, w_1 + \frac{1}{2}k_{1,1}, w_2 + \frac{1}{2}k_{1,2}, \ldots, w_m + \frac{1}{2}k_{1,m}\right)$.
  Step 7 For $j = 1, 2, \ldots, m$ set
    $k_{3,j} = hf_j\left(t + \frac{h}{2}, w_1 + \frac{1}{2}k_{2,1}, w_2 + \frac{1}{2}k_{2,2}, \ldots, w_m + \frac{1}{2}k_{2,m}\right)$.
  Step 8 For $j = 1, 2, \ldots, m$ set
    $k_{4,j} = hf_j(t + h, w_1 + k_{3,1}, w_2 + k_{3,2}, \ldots, w_m + k_{3,m})$.
  Step 9 For $j = 1, 2, \ldots, m$ set
    $w_j = w_j + (k_{1,j} + 2k_{2,j} + 2k_{3,j} + k_{4,j})/6$.
  Step 10 Set $t = a + ih$.
  Step 11 OUTPUT $(t, w_1, w_2, \ldots, w_m)$.
Step 12 STOP.

Definition (5.18)

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

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

Definition (5.19)

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

$$\lim_{h\to 0} \max_{1\le i\le N} |w_i - y(t_i)| = 0,$$

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

Theorem (5.20)

Suppose the initial-value problem $y' = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha$, *is approximated by a one-step difference method in the form*

$$w_0 = \alpha, \; w_{i+1} = w_i + h\phi(t_i, w_i, h).$$

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

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

Then

(i) *The method is stable;*

(ii) *The difference method is convergent if and only if it is consistent, which is equivalent to* $\phi(t, y, 0) = f(t, y), \quad$ *for all* $a \le t \le b$;

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

Theorem (5.21)

Suppose the initial-value problem $y' = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha$, *is approximated by an explicit Adams predictor-corrector method with an m-step Adams-Bashforth predictor equation*

$$w_{i+1} = w_i + h[b_{m-1} f(t_i, w_i) + \cdots + b_0 f(t_{i+1-m}, w_{i+1-m})],$$

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

$$w_{i+1} = w_i + h\left[\tilde{b}_{m-1} f(t_i, w_{i+1}) + \tilde{b}_{m-2} f(t_i, w_i) + \cdots + \tilde{b}_0 f(t_{i+2-m}, w_{i+2-m})\right],$$

with local truncation error $\tilde{\tau}_{i+1}(h)$. *In addition, suppose that* $f(t, y)$ *and* $f_y(t, y)$ *are continuous on* $D = \{ (t, y) \mid a \le t \le b \text{ and } -\infty < y < \infty \}$ *and that* f_y *is bounded. Then the local truncation error* $\sigma_{i+1}(h)$ *of the predictor-corrector method is* $\sigma_{i+1}(h) = \tilde{\tau}_{i+1}(h) + \tau_{i+1}(h)\tilde{b}_{m-1}\frac{\partial f}{\partial y}(t_{i+1}, \theta_{i+1})$, *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*

$$|w_i - y(t_i)| \le \left[\max_{0 \le j \le m-1} \left|w_j - y(t_j)\right| + k_1 \sigma(h)\right] e^{k_2(t_i - a)},$$

where $\sigma(h) = \max_{m \le j \le N} |\sigma_j(h)|$.

Definition (5.22)

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

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

associated with the multistep difference method

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

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

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 (5.23)

(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.**

Theorem (5.24)

A multistep method of the form

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

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

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.

Definition (5.25)

The **region *R* of absolute stability** for a one-step method is $R = \{\, h\lambda \in \mathcal{C} \mid |Q(h\lambda)| < 1 \,\}$, and for a multistep method, it is $R = \{\, h\lambda \in \mathcal{C} \mid |\beta_k| < 1$, for all zeros β_k of $Q(z, h\lambda) \,\}$.

Algorithm 5.8: TRAPEZOIDAL WITH NEWTON ITERATION

To approximate the solution of the initial-value problem

$$y' = f(t, y), \quad \text{for } a \leq t \leq b, \quad \text{with } y(a) = \alpha$$

at $(N + 1)$ equally spaced numbers in the interval $[a, b]$:

NPUT endpoints a, b; integer N; initial condition α; tolerance TOL; maximum number of iterations M at any one step.

OUTPUT approximation w to y at the $(N + 1)$ values of t or a message of failure.

Step 1 Set $h = (b - a)/N$;
$t = a$; $w = \alpha$; OUTPUT (t, w).
Step 2 For $i = 1, 2, \ldots, N$ do Steps 3–7.
Step 3 Set $k_1 = w + \frac{h}{2} f(t, w)$; $w_0 = k_1$; $j = 1$; $FLAG = 0$.

Algorithm 5.8: TRAPEZOIDAL WITH NEWTON ITERATION

Step 4 While $FLAG = 0$ do Steps 5–6.

Step 5 Set $w = w_0 - \dfrac{w_0 - \frac{h}{2} f(t+h, w_0) - k_1}{1 - \frac{h}{2} f_y(t+h, w_0)}$.

Step 6 If $|w - w_0| < TOL$ then set $FLAG = 1$
else set $j = j + 1$;
$w_0 = w$;
if $j > M$ then
OUTPUT ('The maximum number of iterations exceeded');
STOP.

Step 7 Set $t = a + ih$;
OUTPUT (t, w). *End of Step 2*

Step 8 STOP.