Numerical Analysis

Initial-Value Problems for Ordinary Differential Equations

OL Say

ol.say@itc.edu.kh

Institute of Technology of Cambodia

May 9, 2023

2. Outline

1. In mathematics, an ordinary differential equation (ODE) is a differential equation (DE) dependent on only a single independent variable.
2. The term "ordinary" is used in contrast with partial differential equation (PDE) which may be with respect to more than one independent variable.
3. Given f, a function of t, y, and derivatives of y. Then an equation of the form

$$y^{(n)} = f\left(t, y, y', \dots, y^{(n-1)}\right), \text{ where } y = y(t),$$

is called an explicit ODE of order n.

4. More generally, an implicit ODE of order n takes the form:

$$f\left(t, y, y', \dots, y^{(n)}\right) = 0, \text{ where } y = y(t).$$

5. A number of coupled DEs forms a system of DEs.
6. If Y is a vector whose elements are functions;

$$Y(t) = (y_1(t), y_2(t), \dots, y_m(t)),$$

and F is a vector-valued function of Y and its derivatives, then

$$Y^{(n)} = F\left(t, Y, Y', \dots, Y^{(n-1)}\right)$$

is an explicit system of ODEs of order n and dimension m.

7. In column vector form:

$$\begin{pmatrix} y_1^{(n)} \\ y_2^{(n)} \\ \vdots \\ y_m^{(n)} \end{pmatrix} = \begin{pmatrix} f_1(t, Y, Y', \dots, Y^{(n-1)}) \\ f_2(t, Y, Y', \dots, Y^{(n-1)}) \\ \vdots \\ f_m(t, Y, Y', \dots, Y^{(n-1)}) \end{pmatrix}$$

These are not necessarily linear.

8. The implicit analogue is:

$$F\left(t, Y, Y', \dots, Y^{(n-1)}\right) = \mathbf{0}$$

where $\mathbf{0} = (0, 0, \dots, 0)$ is the zero vector.

9. In matrix form

$$\begin{pmatrix} f_1(t, Y, Y', \dots, Y^{(n)}) \\ f_2(t, Y, Y', \dots, Y^{(n)}) \\ \vdots \\ f_m(t, Y, Y', \dots, Y^{(n)}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

10. Note that an explicit ODE of order m can always be transformed into m explicit ODEs of order 1 by using the notation $u_1 = y$ and $u_{j+1} = y^{(j)}$ for $j = 1, \dots, m-1$, and $U = (u_1, \dots, u_m)$.

11. In matrix form:

$$U' = F(t, U) \Leftrightarrow \begin{pmatrix} u_1' \\ u_2' \\ \vdots \\ u_m' \end{pmatrix} = \begin{pmatrix} u_2 \\ u_3 \\ \vdots \\ f(t, u_1, u_2, \dots, u_m) \end{pmatrix}$$

12. The solution of the equation requires the knowledge of m auxiliary conditions.
13. If these conditions are specified at the same value of t, the problem is said to be an initial value problem (IVP).
14. The auxiliary conditions, called initial conditions, have the form

$$y_0(a) = \alpha_1, y_1(a) = \alpha_2, \ldots, y_{m-1}(a) = \alpha_m$$

15. If y_j are specified at different values of t, the problem is called a boundary value problem (BVP).
16. In this chapter, we discus IVP of the following types
    - a. explicit ODE of order 1,
    - b. explicit system of ODEs of order 1 and dimension m,
    - c. explicit ODE of order m.

Definition 1 (Lipschitz)

A function $f(t, y)$ 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

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

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

Theorem 2

Suppose that $D = \{(t, y) | a \leq t \leq b \text{ and } -\infty < y < \infty\}$ 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

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

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

Definition 3

The initial-value problem (IVP)

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

is said to be a well-posed problem if:

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

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

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

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

In other words, and IVP is well-posed if it have the properties that:

1. a solution exists,
2. the solution is unique,
3. the solution's behaviour changes continuously with initial conditions

Theorem 4

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

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

is well-posed.

1. The object of Euler's method is to obtain approximations to the well-posed IVP

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

2. On an interval $I = [a,b]$, we defined $N+1$ mesh points in I by

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

3. Suppose that $y(t)$, the unique solution to the IVP, has two continuous derivatives on I, so that for each $i = 0, 1, \ldots, N-1$,

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

4. Euler's method constructs $w_i \approx y(t_i)$, for each $i = 1, 2, \ldots, N$, by dropping the remainder.

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

Algorightm: Forward Euler Method

To approximate the solution of the IVP

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

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.

1. Set $h = (b-a)/N; t = a; w = \alpha$; OUTPUT (t,w).
2. For i from 1 to N do
   a. Set $w = w + hf(t,w); t = a + ih$.
   b. OUTPUT (t,w).

Example 5

Use an algorithm for Euler's method to approximate the solution to

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

at $t = 2$ with step size $h = 0.2$. (Exact solution $y(t) = (t+1)^2 - e^t/2$.)

i	t	w	y	$\lvert y - w\rvert$
0	0.0	0.50000000	0.50000000	0.00000000
1	0.2	0.80000000	0.82929862	0.02929862
2	0.4	1.15200000	1.21408765	0.06208765
3	0.6	1.55040000	1.64894060	0.09854060
4	0.8	1.98848000	2.12722954	0.13874954
5	1.0	2.45817600	2.64085909	0.18268309
6	1.2	2.94981120	3.17994154	0.23013034
7	1.4	3.45177344	3.73240002	0.28062658
8	1.6	3.95012813	4.28348379	0.33335566
9	1.8	4.42815375	4.81517627	0.38702251
10	2.0	4.86578450	5.30547195	0.43968745

Theorem 6

Suppose f is continuous and satisfies a Lipschitz condition with constant L on $D = \{(t, y) | a \leq t \leq b \text{ and } -\infty < y < \infty\}$ and that a constant M exists with

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

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

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

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

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

1. Suppose the solution $y(t)$ to the initial-value problem

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

has $(n+1)$ continuous derivatives.

2. If we expand the solution, $y(t)$, in terms of its n-th Taylor polynomial about t_i and evaluate at t_{i+1}, we obtain

$$y(t_{i+1}) = y(t_i) + hy'(t) + \frac{h^2}{2}y''(t_i) + \cdots + \frac{h^n}{n!}y^{(n)}(t_i) + \frac{h^{n+1}}{(n+1)!}y^{(n+1)}(\xi_i)$$

for some ξ_i in (t_i, t_{i+1}).

3. Successive differentiation of the solution, $y(t)$, gives

$$y'(t) = f(t,y(t)), y''(t) = f'(t,y(t)), \cdots, y^{(k)}(t) = f^{(k-1)}(t,y(t)).$$

4. Then $y(t_{i+1}) = y(t_i) + hf(t_i, y_{t_i}) + \frac{h^2}{2} f'(t_i, y(t_i)) + \cdots$

$$+ \frac{h^n}{n!} f^{(n-1)}(t_i, y(t_i)) + \frac{h^{n+1}}{(n+1)!} f^{(n)}(\xi_i, y(\xi_i))$$

5. The difference-equation method corresponding to the above equation is obtained by deleting the remainder term involving ξ_i.
6. Taylor method of order n

$$w_0 = \alpha,$$
$$w_{i+1} = w_i + hT^{(n)}(t_i, w_i), \; i = 0, 1, \dots, N-1,$$

where $T^{(n)}(t_i, w_i) = f(t_i, w_i) + \frac{h}{2} f'(t_i, w_i) + \cdots + \frac{h^{n-1}}{n!} f^{(n-1)}(t_i, w_i)$.

7. From the definitions, we identify Euler's method as Taylor's method of order one.

Example 7

Apply Taylor's method of order two with $n = 10$ to the initial-value problem $y' = y - t^2 + 1,\ 0 \le t \le 2,\ y(0) = 0.5.$

1. $f(t, y) = y - t^2 + 1 \Rightarrow f'(t, y) = y' - 2t = y - t^2 + 1 - 2t$
2. $T^{(2)}(t_i, w_i) = f(t_i, w_i) + \frac{h}{2} f'(t_i, w_i) = \left(1 + \frac{h}{2}\right)(w_i - t_i^2 + 1) - ht_i$
3. The second-order method becomes

$$
\begin{aligned}
w_0 &= 0.5,\ t_0 = 0\\
\text{For } i &= 0, 1, \dots, 9:\\
w_{i+1} &= w_i + h\left[\left(1 + \frac{h}{2}\right)(w_i - t_i^2 + 1) - ht_i\right],\\
t_{i+1} &= t_i + h.
\end{aligned}
$$

Table: Taylor's method of order 2

i	t	w	y	$\|y - w\|$
0	0.0	0.50000000	0.50000000	0.00000000
1	0.2	0.83000000	0.82929862	0.00070138
2	0.4	1.21580000	1.21408765	0.00171235
3	0.6	1.65207600	1.64894060	0.00313540
4	0.8	2.13233272	2.12722954	0.00510318
5	1.0	2.64864592	2.64085909	0.00778683
6	1.2	3.19134802	3.17994154	0.01140648
7	1.4	3.74864458	3.73240002	0.01624457
8	1.6	4.30614639	4.28348379	0.02266261
9	1.8	4.84629860	4.81517627	0.03112233
10	2.0	5.34768429	5.30547195	0.04221234

1. The accuracy of numerical integration can be greatly improved by keeping more terms of the series.
2. Thus an n-th order Taylor method would use the truncated Taylor polynomial

$$P_n(t+h) = y(t) + y'(t)h + \frac{1}{2!}y''(t)h^2 + \cdots + \frac{1}{n!}y^{(n)}(t)h^n.$$

3. To arrive at the second-order Runge-Kutta method, we assume an integration formula of the form

$$y(t+h) = y(t) + b_1 f(t,y)h + b_2 f(t + c_2 h, y + a_{21} h f(t,y))h, \qquad (*)$$

and attempt to find the parameters a_1, a_2, b_1 and b_2 by matching above equation to the Taylor polynomial:

$$\begin{aligned} P_2(t+h) &= y(t) + y'(t)h + \frac{1}{2!}y''(t)h^2 \\ &= y(t) + f(t,y)h + \frac{1}{2}f'(t,y)h^2 \\ &= y(t) + f(t,y)h + \frac{1}{2}\left[f_t(t,y) + f_y(t,y)y'(t)\right]h^2 \\ &= y(t) + f(t,y)h + \frac{1}{2}f_t(t,y)h^2 + \frac{1}{2}f(t,y)f_y(t,y)h^2, \quad (**) \end{aligned}$$

4. By the fact that $f(x+h, y+k) = f(x,y) + f_x(x,y)h + f_y(x,y)k + R_2(x,y)$, we get

$$\begin{aligned} &f(t + c_2h, y + a_{21}hf(t,y)) \\ &\qquad = f(t,y) + f_t(t,y)c_2h + f_y(t,y)a_{21}hf(t,y) + R_2(t,y) \end{aligned}$$

5. The equation $(*)$ becomes

$$y(t,y) = y(t,y) + b_1 f(t,y) + b_2 f(t,y) + b_2 c_2 f_t(t,y)h^2 + b_2 a_{21} f(t,y) f_y(t,y) h^2 + R_2(t,y)$$

6. Truncate $R_2(t,y)$ of the last equation and compare it to $(**)$,

$$b_1 + b_2 = 1, b_2 c_2 = \frac{1}{2}, \text{ and } b_2 a_{21} = \frac{1}{2}.$$

7. These are three non-linear equations for the four unknowns. Using a_{21} as a free parameter, we have

$$b_1 = 1 - \frac{1}{2a_{21}}, b_2 = \frac{1}{2a_{21}}, c_2 = a_{21}.$$

8. Some of the popular choices and the names associated with the resulting formulas are as follows:
9. Explicit midpoint method: $(b_1, b_2, c_2, a_{21}) = (0, 1, 1/2, 1/2)$

$$w_0 = \alpha, t_0 = a,$$
$$\text{For } i = 0, 1, \ldots, N-1$$
$$w_{i+1} = w_i + hf\left(t_i + \frac{1}{2}h, w_i + \frac{1}{2}hf(t_i, w_i)\right), t_{i+1} = t_i + h.$$

Or equivalently,

$$k_1 = f(t_i, w_i),$$
$$k_2 = f\left(t_i + \frac{1}{2}h, w_i + \frac{1}{2}hk_1\right),$$
$$w_{i+1} = w_i + hk_2, t_{i+1} = t_i + h.$$

10. **Modified Euler's/Heun's second-order method:**
$(b_1, b_2, c_2, a_{21}) = (1/2, 1/2, 1, 1)$

$$w_0 = \alpha, t_0 = a,$$
For $i = 0, 1, \dots, N-1$
$$w_{i+1} = w_i + \frac{1}{2}hf(t_i, w_i) + \frac{1}{2}hf\left(t_i + h, w_i + hf(t_i, w_i)\right), t_{i+1} = t_i + h.$$

11. **Ralston's method:** $(b_1, b_2, c_2, a_{21}) = (1/4, 3/4, 2/3, 2/3)$

$$w_0 = \alpha, t_0 = a,$$
For $i = 0, 1, \dots, N-1$
$$w_{i+1} = w_i + \frac{1}{4}hf(t_i, w_i) + \frac{3}{4}hf\left(t_i + \frac{2}{3}h, w_i + \frac{2}{3}hf(t_i, w_i)\right), t_{i+1} = t_i + h.$$

1. The fourth-order Runge-Kutta method is obtained from the Taylor series along the same lines as the second-order method.
2. As the second-order case, there is no unique fourth-order Runge-Kutta formula.
3. The most popular version formula is described as the following sequence of operations:

$$
\begin{aligned}
w_0 &= \alpha, t_0 = a, \\
\text{For } i &= 0, 1, \ldots, N-1, \\
k_1 &= f(t_i, w_i) \\
k_2 &= f\left(t_i + \tfrac{1}{2}h, w_i + \tfrac{1}{2}hk_1\right) \\
k_3 &= f\left(t_i + \tfrac{1}{2}h, w_i + \tfrac{1}{2}hk_2\right) \\
k_4 &= f(t_i + h, w_i + hk_3) \\
w_{i+1} &= w_i + \tfrac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4), t_{i+1} = t_i + h.
\end{aligned}
$$

Example 8

Apply Runge-Kutta's method of order four with $n = 10$ to the initial-value problem $y' = y - t^2 + 1,\ 0 \le t \le 2,\ y(0) = 0.5.$

i	t	w	y	$\|y - w\|$
0	0.0	0.50000000	0.50000000	0.00000000
1	0.2	0.82929333	0.82929862	0.00000529
2	0.4	1.21407621	1.21408765	0.00001144
3	0.6	1.64892202	1.64894060	0.00001858
4	0.8	2.12720268	2.12722954	0.00002685
5	1.0	2.64082269	2.64085909	0.00003639
6	1.2	3.17989417	3.17994154	0.00004737
7	1.4	3.73234007	3.73240002	0.00005994
8	1.6	4.28340950	4.28348379	0.00007429
9	1.8	4.81508569	4.81517627	0.00009057
10	2.0	5.30536300	5.30547195	0.00010895

1. The family of explicit Runge-Kutta methods is a generalization of the fourth-order Runge-Kutta method mentioned above. It is given by

$$
\begin{aligned}
w_0 &= \alpha, t_0 = a, \\
\text{For } i &= 0, 1, \dots, N-1 : \\
k_1 &= f(t_i, w_i), \\
k_2 &= f(t_i + c_2 h, w_i + (a_{21} k_1) h), \\
k_3 &= f(t_i + c_3 h, w_i + (a_{31} k_1 + a_{32} k_2) h), \\
&\vdots \\
k_m &= f(t_i + c_m h, w_i + (a_{m1} k_1 + a_{m2} k_2 + \cdots + a_{m,m-1} k_{m-1}) h), \\
w_{i+1} &= w_i + h(b_i k_1 + b_2 k_2 + \cdots + b_m k_m), \ t_{i+1} = t_i + h.
\end{aligned}
$$

2. To specify a particular method, one needs to provide the integer m (the number of stages), and the coefficients a_{ij} (for $1 \leq j < i \leq m$), b_i (for $i = 1, 2, \ldots, m$) and c_i (for $i = 2, 3, \ldots, m$).
3. The matrix (a_{ij}) is called the Runge-Kutta matrix, while the b_i and $c \neq$ are known as the weights and the nodes.
4. These data are usually arranged in a mnemonic device, known as a Butcher tableau.

$$
\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & (0) & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_m & a_{m1} & a_{m2} & \cdots & a_{m,m-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{m-1} & b_m
\end{array}
$$

1. We list some explicit Runge-Kutta's methods (matrix (a_{ij}) is lower triangle) as the following.
2. Forward Euler's method: The Euler method is first order. The lack of stability and accuracy limits its popularity mainly to use as a simple introductory example of a numeric solution method.

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

3. Generic second-order method:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \alpha & \alpha & 0 \\ \hline & 1-\dfrac{1}{2\alpha} & \dfrac{1}{2\alpha} \end{array}$$

4. **Explicit midpoint method:** The (explicit) midpoint method is a second-order method with two stages (see also the implicit midpoint method below):

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array}$$

5. **Heun's method:** Heun's method is a second-order method with two stages. It is also known as the explicit trapezoid rule, improved Euler's method, or modified Euler's method.

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array}$$

6. **Ralston's method:** Ralston's method is a second-order method with two stages and a minimum local error bound:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 2/3 & 2/3 & 0 \\ \hline & 1/4 & 3/4 \end{array}$$

7. **Generic third-order method:** for $\alpha \neq 0, 2/3, 1$,

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \alpha & \alpha & 0 & 0 \\ 1 & 1 + \dfrac{1-\alpha}{\alpha(3\alpha-2)} & -\dfrac{1-\alpha}{\alpha(3\alpha-2)} & 0 \\ \hline & \dfrac{1}{2} - \dfrac{1}{6\alpha} & \dfrac{1}{6\alpha(1-\alpha)} & \dfrac{2-3\alpha}{6(1-\alpha)} \end{array}$$

8. Kutta's third-order method:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1 & -1 & 2 & 0 \\ \hline & 1/6 & 2/3 & 1/6 \end{array}$$

9. Heun's third-order method:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 \\ 2/3 & 0 & 2/3 & 0 \\ \hline & 1/4 & 0 & 3/4 \end{array}$$

10. Van der Houwen's/Wray third-order method:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 8/15 & 8/15 & 0 & 0 \\ 2/3 & 1/4 & 5/12 & 0 \\ \hline & 1/4 & 0 & 3/4 \end{array}$$

11. Ralston's third-order method:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 3/4 & 0 & 3/4 & 0 \\ \hline & 2/9 & 1/3 & 4/9 \end{array}$$

12. Third-order Strong Stability Preserving Runge-Kutta (SSPRK3):

$$
\begin{array}{c|ccc}
0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 \\
1/2 & 1/4 & 1/4 & 0 \\
\hline
 & 1/6 & 1/6 & 2/3
\end{array}
$$

13. Classic fourth-order method: The "original" Runge–Kutta method.

$$
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
1/2 & 1/2 & 0 & 0 & 0 \\
1/2 & 0 & 1/2 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
\hline
 & 1/6 & 1/3 & 1/3 & 1/6
\end{array}
$$

14 **3/8-rule fourth-order method:** This method doesn't have as much notoriety as the "classic" method, but is just as classic because it was proposed in the same paper (Kutta, 1901).

$$
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
1/3 & 1/3 & 0 & 0 & 0 \\
2/3 & -1/3 & 1 & 0 & 0 \\
1 & 1 & -1 & 1 & 0 \\
\hline
 & 1/8 & 3/8 & 3/8 & 1/8
\end{array}
$$

Definition 9

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

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

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 :

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

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

$$w_0 = \alpha, \; w_0 = \alpha_1, \; w_0 = \alpha_2, \; \cdots, \; w_{m-1} = \alpha_{m-1}$$

are specified.

1. When $b_m = 0$ the method is called explicit, or open because the equation gives w_{i+1} explicitly in terms of previously determined values.
2. When $b \neq 0$ the method is called implicit, or close, because w_{i+1} occurs on both sides of the equation, so w_{i+1} is specified only implicitly.
3. Some of the explicit multistep methods together with their required starting values and local truncation errors are as follows.
4. Adams-Bashforth Two-Step Explicit Method: $i = 1, \ldots, N-1$.

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

$$w_{i+1} = w_i + \frac{h}{2}[3f(t_i, w_i) - f(t_{i-1}, w_{i-1})],$$

$$\tau_{i+1}(h) = \frac{5}{12}y'''(\mu_i)h^2, \mu_i \in (t_{i-1}, t_{i+1}).$$

5. **Adams-Bashforth Three-Step Explicit Method:** $i = 2, \ldots, N-1$.

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

$$w_{i+1} = w_i + \frac{h}{12}[23f(t_i, w_i) - 16f(t_{i-1}, w_{i-1}) + 5f(t_{i-2}, w_{i-2})],$$

$$\tau_{i+1}(h) = \frac{3}{8}y^{(4)}(\mu_i)h^3, \mu_i \in (t_{i-2}, t_{i+1}).$$

6. **Adams-Bashforth Four-Step Explicit Method:** $i = 3, \ldots, N-1$.

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

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

$$\tau_{i+1}(h) = \frac{251}{720}y^{(5)}(\mu_i)h^4, \mu_i \in (t_{i-3}, t_{i+1}).$$

7. Adams-Bashforth Five-Step Explicit Method: $i = 4, \ldots, N-1$.

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

$$\begin{aligned} w_{i+1} = w_i + \frac{h}{720}[&1901f(t_i, w_i) - 2774f(t_{i-1}, w_{i-1}) \\ &+ 2616f(t_{i-2}, w_{i-2}) - 1274f(t_{i-3}, w_{i-3}) \\ &+ 251f(t_{i-4}, w_{i-4})], \end{aligned}$$

$$\tau_{i+1}(h) = \frac{95}{288} y^{(6)}(\mu_i)h^5, \mu_i \in (t_{i-4}, t_{i+1}).$$

8. Some of the more common implicit methods are as follows.

9. **Adams-Moulton Two-Step Implicit Method:** $i = 1, \ldots, N-1$

$$w_0 = \alpha, w_1 = \alpha_1,$$
$$w_{i+1} = w_i + \frac{h}{2}[5f(t_{i+1}, w_{i+1}) + 8f(t_i, w_w) - f(t_{i-1}, w_{i-1})]$$
$$\tau_{i+1}(h) = -\tfrac{1}{14}y^{(4)}(\mu_i)h^3, \mu_i \in (t_{i-1}, t_{i+1}).$$

10. **Adams-Moulton Three-Step Implicit Method:** $i = 2, \ldots, N-1$

$$w_0 = \alpha, w_1 = \alpha_1, w_2 = \alpha_2,$$
$$w_{i+1} = w_i + \frac{h}{24}[9f(t_{i+1}, w_{i+1}) + 19f(t_i, w_w) - 5f(t_{i-1}, w_{i-1}) + f(t_{i-2}, w_{i-2})]$$
$$\tau_{i+1}(h) = -\tfrac{19}{720}y^{(5)}(\mu_i)h^4, \mu_i \in (t_{i-2}, t_{i+1}).$$

11 **Adams-Moulton Four-Step Implicit Method:** $i = 3, \ldots, N-1$

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

$$w_{i+1} = w_i + \frac{h}{720}\big[251f(t_{i+1}, w_{i+1}) + 646f(t_i, w_w) - 264f(t_{i-1}, w_{i-1}) + 106f(t_{i-2}, w_{i-2}) - 19f(t_{i-3}, w_{i-3})\big]$$

$$\tau_{i+1}(h) = -\tfrac{3}{160}y^{(5)}(\mu_i)h^5, \mu_i \in (t_{i-3}, t_{i+1}).$$

1. The combination of an explicit method to predict and an implicit to improve the prediction is called a predictor-corrector method.
2. Consider the following fourth-order method for solving an initial-value problem.
3. 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.
4. To do this, we use a fourth-order one-step method, the Runge-Kutta method of order four.
5. The next step is to calculate an approximation, w_{4p}, to $y(t_4)$ using the explicit Adams-Bashforth method as predictor:

$$wp_4 = w_3 + \frac{h}{24}[55f(t_3, w_3) - 59f(t_2, w_2) + 37f(t_1, w_1) - 9f(t_0, w_0)]$$

6. This approximation is improved by inserting w_{4p} in the right side of the three-step implicit Adams-Moulton method and using that method as a corrector:

$$w_4 = w_3 + \frac{h}{24}[9f(t_4, wp_4) + 19f(t_3, w_3) - 5f(t_2, w_2) + f(t_1, w_1)].$$

Example 10

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

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

i	t_i	w_i	y_i	$\lvert y_i - w_i \rvert$
0	0.0	0.50000000	0.50000000	0.00000000
1	0.2	0.82929333	0.82929862	0.00000529
2	0.4	1.21407621	1.21408765	0.00001144
3	0.6	1.64892202	1.64894060	0.00001858
4	0.8	2.12720563	2.12722954	0.00002390
5	1.0	2.64082860	2.64085909	0.00003049
6	1.2	3.17990264	3.17994154	0.00003890
7	1.4	3.73235048	3.73240002	0.00004953
8	1.6	4.28342082	4.28348379	0.00006296
9	1.8	4.81509636	4.81517627	0.00007991
10	2.0	5.30537067	5.30547195	0.00010128

Theorem 11

Suppose that

$$D = \{(t, u_1, u_2, \ldots, u_m) \mid a \le t \le b,\ -\infty < u_j < \infty,\ j = 1, 2, \ldots, m\},$$

and let $f_j(t, u_1, u_2, \ldots, u_m)$, *for each* $j = 1, 2, \ldots, m$, *be continuous and satisfy a* *Lipschitz condition* *on* D. *The system of first-order differential equations*

$$\begin{cases} u_1' &= f_1(t, u_1, u_2, \ldots, u_m) \\ u_2' &= f_2(t, u_1, u_2, \ldots, u_m) \\ &\vdots \\ u_m' &= f_m(t, u_1, u_2, \ldots, u_m) \end{cases},$$

subject to the initial conditions

$$u_1(a) = \alpha_1, u_2(a) = \alpha_2, \ldots, u_m(a) = \alpha_m,$$

has a *unique solution* $u_1(t), u_2(t), \ldots, u_m(t)$, *for* $a \le t \le b$.

1. Methods to solve systems of first-order differential equations are generalizations of the methods for a single first-order equation.
2. For example, the classical Runge-Kutta method of order four can be generalized as follows.
3. Let an integer $N > 0$ be chosen and set $h = (b - a)/N$.
4. Partition $[a, b]$ into N subintervals with the mesh points

$$t_i = a + ih, i = 0, 1, \ldots, N.$$

5. Use the notation w_{ij}, for each $i = 0, 1, \ldots, N$ and $j = 1, 2, \ldots, m$, to denote an approximation to $u_j(t_i)$.
6. For the initial conditions, set

$$w_{0,1} = \alpha_1, w_{0,2} = \alpha_2, \ldots, w_{0,m} = \alpha_m,$$

7. For $i = 0, \ldots, N-1$:
   a. For $j = 1, \ldots, m$: $k_{1,j} = f_j(t_i, w_{i,1}, \ldots, w_{i,m})$
   b. For $j = 1, \ldots, m$: $k_{2,j} = f_j\left(t_i + \frac{h}{2}, w_{i,1} + \frac{h}{2}k_{1,1}, \ldots, w_{i,m} + \frac{h}{2}k_{1,m}\right)$
   c. For $j = 1, \ldots, m$: $k_{3,j} = f_j\left(t_i + \frac{h}{2}, w_{i,1} + \frac{h}{2}k_{2,1}, \ldots, w_{i,m} + \frac{h}{2}k_{2,m}\right)$
   d. For $j = 1, \ldots, m$: $k_{4,j} = f_j(t_i + h, w_{i,1} + hk_{3,1}, \ldots, w_{i,m} + hk_{3,m})$
   e. For $j = 1, \ldots, m$: $w_{i+1,j} = w_{i,j} + \frac{h}{6}(k_{1,j} + 2k_{2,j} + 2k_{3,j} + k_{4,j})$.
8. We can rewrite the formula in vector form by setting
   a. $Y = (y_1, \cdots, y_m)$
   b. $F(t, Y) = (f_1(t, Y), \cdots, f_m(t, Y))$
   c. $W_i = (w_{i,1}, \cdots, w_{i,m})$
   d. $K_1 = (k_{1,1}, \cdots, k_{1,m})$
   e. $K_2 = (k_{2,1}, \cdots, k_{2,m})$

f. $K_3 = (k_{3,1}, \cdots, k_{3,m})$
g. $K_4 = (k_{4,1}, \cdots, k_{4,m})$

9. Then the recurrence can be rewritten as $W_0 = (\alpha_1, \dots, \alpha_m)$.

10. For $i = 0, \dots, N-1$:
   a. $K_1 = F(t_i, W_i)$
   b. $K_2 = F(t_i + \frac{h}{2}, W_i + \frac{h}{2}K_1)$
   c. $K_3 = F(t_i + \frac{h}{2}, W_i + \frac{h}{2}K_2)$
   d. $K_4 = F(t_i + h, W_i + hK_3)$
   e. $W_{i+1} = W_i + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$.

Example 12

Apply Runge-Kutta's method of order four with $h = 0.1$ to the system of initial-value problems

$$\begin{cases} u_1'(t) = -4u_1(t) + 3u_2(t) + 6, & u_1(0) = 0, \\ u_2'(t) = -2.4u_1(t) + 1.6u_2(t) + 3.6, & u_2(0) = 0 \end{cases}$$

to determine the value of the functions at $t = 0.5$. Compute the exact error provided that the exact solution to this system is

$$\begin{cases} u_1(t) = -3.375e^{-2t} + 1.875e^{-0.4t} + 1.5, \\ u_2(t) = -2.25e^{-2t} + 2.25e^{-0.4t}. \end{cases}$$

i	t	w_1	w_2	u_1	u_2	E_1	E_2
0	0.0	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
1	0.1	0.53826	0.31963	0.53826	0.31963	0.00001	0.00001
2	0.2	0.96850	0.56878	0.96851	0.56879	0.00001	0.00001
3	0.3	1.31072	0.76073	1.31074	0.76074	0.00002	0.00001
4	0.4	1.58127	0.90632	1.58128	0.90633	0.00002	0.00001
5	0.5	1.79351	1.01440	1.79353	1.01442	0.00002	0.00001

Example 13

Transform the the second-order initial-value problem

$$y'' - 2y' + 2y = e^{2t}\sin t,\ \ y(0) = -0.4, y'(0) = -0.6,\ \ \text{for } 0 \le t \le 1$$

into a system of first order initial-value problems, and use the Runge-Kutta method with $h = 0.1$ to approximate the solution.

1. Let $u_1(t) = y(t)$ and $u_2(t) = y'(t)$.
2. This transforms the second-order equation into the system

$$\begin{cases} u_1'(t) = u_2(t) \\ u_2'(t) = e^{2t}\sin t - 2u_1(t) + 2u_2(t) \end{cases}$$

   with initial conditions $u_1(0) = -0.4, u_2(0) = -0.6$.
3. $F(t, U(t)) = F(t, (u_1(t), u_2(t))) = (u_2(t), e^{2t}\sin t - 2u_1(t) + 2u_2(t))$.

i	t	$u_1 = y$	$u_2 = y'$
0	0.0	-0.40000000	-0.60000000
1	0.1	-0.46173334	-0.63163124
2	0.2	-0.52555988	-0.64014895
3	0.3	-0.58860144	-0.61366381
4	0.4	-0.64661231	-0.53658203
5	0.5	-0.69356666	-0.38873810
6	0.6	-0.72115190	-0.14438087
7	0.7	-0.71815295	0.22899702
8	0.8	-0.66971133	0.77199180
9	0.9	-0.55644290	1.53478148
10	1.0	-0.35339886	2.57876634