

MULTISTEP METHODS

All of the methods we have studied until now are examples of *one-step methods*. These methods use one past value of the numerical solution, call it y_n , in order to compute a new value y_{n+1} . Multistep methods use more than one past value, and these are often more efficient than the earlier methods. For example,

$$y_{n+1} = y_n + \frac{h}{2} [3f(x_n, y_n) - f(x_{n-1}, y_{n-1})], \quad n \geq 1 \quad (1)$$

is an explicit, stable, second order method with truncation error of size $O(h^3)$. We derive it below.

Multistep methods can be derived in a number of ways, but the most popular methods are derived most easily using numerical interpolation and numerical integration.

Integrate the equation

$$Y'(x) = f(x, Y(x)) \quad (2)$$

over the interval $[x_n, x_{n+1}]$. This yields

$$Y(x_{n+1}) = Y(x_n) + \int_{x_n}^{x_{n+1}} f(x, Y(x)) dx \quad (3)$$

Other intervals $[x_{n-p}, x_{n+1}]$, $p > 0$, can also be used; but the most popular multistep methods are obtained with the choice $[x_n, x_{n+1}]$.

Approximate the integral by replacing the integrand with a polynomial interpolant of it. For example, the linear interpolant of a function $g(x)$ based on interpolation at $\{x_{n-1}, x_n\}$ is given by

$$P_1(x) = \frac{(x_n - x)g(x_{n-1}) + (x - x_{n-1})g(x_n)}{h}$$

Integrate this over the interval $[x_n, x_{n+1}]$, obtaining

$$\begin{aligned} \int_{x_n}^{x_{n+1}} g(x) dx &\approx \int_{x_n}^{x_{n+1}} P_1(x) dx \\ &= \frac{3h}{2}g(x_n) - \frac{h}{2}g(x_{n-1}) \end{aligned}$$

By using methods similar to those used in §5.2, it can be shown that

$$\int_{x_n}^{x_{n+1}} g(x) dx = \frac{3h}{2}g(x_n) - \frac{h}{2}g(x_{n-1}) + \frac{5}{12}h^3g''(\xi_n)$$

for some ξ_n , $x_{n-1} \leq \xi_n \leq x_{n+1}$. Using this in (3) with $g(x) = Y'(x) = f(x, Y(x))$ yields

$$Y(x_{n+1}) = Y(x_n) + \frac{h}{2} [3f(x_n, Y(x_n)) - f(x_{n-1}, Y(x_{n-1}))] + \frac{5}{12}h^3Y'''(\xi_n)$$

This leads to the method in (1) when the truncation error term is dropped:

$$y_{n+1} = y_n + \frac{h}{2} [3f(x_n, y_n) - f(x_{n-1}, y_{n-1})], \quad n \geq 1$$

This is a two-step method (a second order recurrence relation) and it requires values for both y_0 and y_1 before proceeding to find y_n for $n \geq 2$. A value for y_1 must be obtained by some other method.

NUMERICAL EXAMPLE. Consider solving

$$Y'(x) = -Y(x) + 2 \cos x, \quad Y(0) = 1$$

The true solution is $Y(x) = \sin x + \cos x$. We give numerical results for the method in (1) with stepsizes $h = 0.05$ and $2h = 0.1$.

x	$y_h(x)$	$Y(x) - y_{2h}(x)$	$Y(x) - y_h(x)$	<i>Ratio</i>
2	0.492597	2.13E-3	5.53E-4	3.9
4	-1.411170	2.98E-3	7.24E-4	4.1
6	-0.675004	-3.91E-3	-9.88E-4	4.0
8	0.843737	3.68E-4	1.21E-4	3.0
10	-1.383983	3.61E-3	8.90E-4	4.1

This is consistent with an error formula

$$Y(x_n) - y_h(x_n) = O(h^2)$$

with a constant of proportionality that depends on x_n .

Compare the error to that for a second order Runge-Kutta method.

It can be shown that if

$$Y(x_0) - y_0 = O(h^2) \quad [\text{usually zero}]$$

$$Y(x_1) - y_1 = O(h^2)$$

with standard assumptions on the differentiability of f and Y , then we have

$$\max_{x_0 \leq x_n \leq b} |Y(x_n) - y_h(x_n)| \leq c h^2$$

for some constant $c \geq 0$.

Moreover, if

$$Y(x_0) - y_0 \approx c_0 h^2$$

$$Y(x_1) - y_1 \approx c_1 h^2$$

for constants c_0, c_1 , then

$$Y(x_n) - y_h(x_n) = D(x_n)h^2 + O(h^3)$$

with $D(x)$ a continuous function for $x_0 \leq x \leq b$.

From this, we also have the Richardson error estimate

$$Y(x_n) - y_h(x_n) \approx \frac{1}{3}[y_h(x) - y_{2h}(x)]$$

NUMERICAL EXAMPLE
(continuation)

Continue with solving

$$Y'(x) = -Y(x) + 2 \cos x, \quad Y(0) = 1$$

using method (1). The true solution is $Y(x) = \sin x + \cos x$. We use stepsizes $h = 0.05$ and $2h = 0.1$.

x	$y_h(x)$	$Y(x) - y_h(x)$	$\frac{1}{3}[y_h(x) - y_{2h}(x)]$
2	0.492597	$5.53E - 4$	$5.26E - 4$
4	-1.411170	$7.24E - 4$	$7.52E - 4$
6	-0.675004	$-9.88E - 4$	$-9.73E - 4$
8	0.843737	$1.21E - 4$	$8.21E - 5$
10	-1.383983	$8.90E - 4$	$9.08E - 4$

ADAMS-BASHFORTH METHODS

Recall the integrated formula

$$Y(x_{n+1}) = Y(x_n) + \int_{x_n}^{x_{n+1}} f(x, Y(x)) dx \quad (4)$$

Approximate the integrand using a polynomial interpolant of degree q . In particular, do the interpolation using the $q + 1$ points $\{x_n, x_{n-1}, \dots, x_{n-q}\}$.

Denote the interpolant by $P_q(x)$:

$$\begin{aligned} P_q(x) &= \sum_{j=0}^q L_{j,n}(x) f(x_{n-j}, Y(x_{n-j})) \\ &= \sum_{j=0}^q L_{j,n}(x) Y'(x_{n-j}) \end{aligned}$$

with $L_{j,n}(x)$, $0 \leq j \leq q$, denoting the Lagrange interpolation basis functions using the points $\{x_n, \dots, x_{n-q}\}$.

Substituting $P_q(x)$ into (4) and integrating, we obtain a formula

$$\begin{aligned}
 Y(x_{n+1}) &= Y(x_n) \\
 &+ h \sum_{j=0}^q w_{j,q} f(x_{n-j}, Y(x_{n-j})) + E_n \quad (5) \\
 hw_{j,q} &= \int_{x_n}^{x_{n+1}} L_{j,n}(x) dx
 \end{aligned}$$

and E_n denotes the error for this numerical integration over $[x_n, x_{n+1}]$. Dropping E_n , we obtain a way to approximate $Y(x_{n+1})$ using values at past points $x_n, x_{n-1}, \dots, x_{n-q}$:

$$y_{n+1} = y_n + h \sum_{j=0}^q w_{j,q} f(x_{n-j}, y_{n-j}) \quad (6)$$

This is called an *Adams-Bashforth method*. It is a $(q + 1)$ -step explicit method, and its truncation error is of size $O(h^{q+2})$.

We give several of these numerical methods and their corresponding truncation error terms E_n .

$q = 0$:

$$y_{n+1} = y_n + hf(x_n, y_n) \quad [\text{Euler's method}]$$

$$E_n = \frac{1}{2}h^2 Y''(\xi_n)$$

$q = 1$: Use the notation $y'_k = f(x_k, y_k)$.

$$y_{n+1} = y_n + \frac{h}{2} [3y'_n - y'_{n-1}], \quad n \geq 1$$

$$E_n = \frac{5}{12}h^3 Y'''(\xi_n)$$

$q = 2$:

$$y_{n+1} = y_n + \frac{h}{12} [23y'_n - 16y'_{n-1} + 5y'_{n-2}]$$

$$E_n = \frac{3}{8}h^4 Y^{(4)}(\xi_n)$$

$q = 3$:

$$y_{n+1} = y_n + \frac{h}{24} [55y'_n - 59y'_{n-1} + 37y'_{n-2} - 9y'_{n-3}]$$

$$E_n = \frac{251}{720}h^5 Y^{(5)}(\xi_n)$$

CONVERGENCE. Note first that when computing y_{n+1} using (6) that we require the $q + 1$ initial values y_0, y_1, \dots, y_q in order to compute y_{q+1} . Then we can continue using (6) to compute y_n for $n \geq q + 1$. The initial values y_1, \dots, y_q must be computed by some other method, perhaps a Runge-Kutta method or perhaps a lower order Adams-Bashforth method.

We can prove the following as regards the convergence. Let

$$\begin{aligned}\eta(h) &= \max_{0 \leq j \leq q} |Y(x_j) - y_j| \\ \delta(h) &= \max_{x_q \leq x_n \leq b} |E_n|\end{aligned}$$

Then there is an $h_0 > 0$ such that for $0 < h \leq h_0$, the numerical method of (6) is computable, and

$$|Y(x_n) - y_n| \leq c \left[\eta(h) + \frac{1}{h} \delta(h) \right] \quad (7)$$

for some constant $c > 0$. From the examples, we see that

$$|\delta(h)| \leq dh^{q+2} \max_{x_0 \leq x \leq b} |Y^{(q+2)}(x)|$$

and this is true for general $q \geq 0$.

STABILITY. All Adams-Bashforth methods possess the basic type of numerical stability associated with the earlier methods studied (Euler, backward Euler, trapezoidal).

Consider solving

$$y_{n+1} = y_n + h \sum_{j=0}^q w_{j,q} f(x_{n-j}, y_{n-j})$$

with initial values

$$y_k = Y(x_k), \quad k = 0, 1, \dots, q$$

Then perturb these initial values and solve

$$z_{n+1} = z_n + h \sum_{j=0}^q w_{j,q} f(x_{n-j}, z_{n-j})$$

with

$$|z_k - y_k| \leq \varepsilon, \quad k = 0, 1, \dots, q$$

for all sufficiently small values of h , say $0 < h \leq h_0$.

Then there is a constant $c > 0$ with

$$\max_{x_q \leq x_n \leq b} |z_k - y_k| \leq c\varepsilon, \quad 0 < h \leq h_0 \quad (8)$$

ADAMS-MOULTON METHODS

These methods are derived in the same manner as the Adams-Bashforth methods. We begin with

$$Y(x_{n+1}) = Y(x_n) + \int_{x_n}^{x_{n+1}} f(x, Y(x)) dx$$

Approximate the integrand using a polynomial interpolant of degree q . In particular, do the interpolation using the $q + 1$ points $\{x_{n+1}, x_n, \dots, x_{n-q+1}\}$.

Denote the interpolant by $\hat{P}_q(x)$:

$$\begin{aligned} \hat{P}_q(x) &= \sum_{j=-1}^{q-1} \hat{L}_{j,n}(x) f(x_{n-j}, Y(x_{n-j})) \\ &= \sum_{j=-1}^{q-1} \hat{L}_{j,n}(x) Y'(x_{n-j}) \end{aligned}$$

with $\hat{L}_{j,n}(x)$, $-1 \leq j \leq q - 1$, denoting the Lagrange interpolation basis functions using the points $\{x_{n+1}, \dots, x_{n-q+1}\}$.

Substituting $\hat{P}_q(x)$ into the integral and integrating, we obtain a formula

$$\begin{aligned}
 Y(x_{n+1}) &= Y(x_n) \\
 &+ h \sum_{j=-1}^{q-1} v_{j,q} f(x_{n-j}, Y(x_{n-j})) + \hat{E}_n \\
 hv_{j,q} &= \int_{x_n}^{x_{n+1}} \hat{L}_{j,n}(x) dx
 \end{aligned}$$

and \hat{E}_n denotes the error for this numerical integration over $[x_n, x_{n+1}]$. Dropping \hat{E}_n , we obtain a way to approximate $Y(x_{n+1})$ using values at the points $x_{n+1}, \dots, x_{n-q+1}$:

$$y_{n+1} = y_n + h \sum_{j=-1}^{q-1} v_{j,q} f(x_{n-j}, y_{n-j})$$

This is called an *Adams-Moulton method*. It is a q -step implicit method, and its truncation error is of size $O(h^{q+2})$.

We give several of these numerical methods and their corresponding truncation error terms \hat{E}_n .

Recall the notation $y'_k = f(x_k, y_k)$, $k \geq 0$.

$q = 0$: It is the backward Euler method,

$$\begin{aligned}y_{n+1} &= y_n + hy'_{n+1} \\ \hat{E}_n &= -\frac{1}{2}h^2Y''(\xi_n)\end{aligned}$$

$q = 1$: It is the trapezoidal method,

$$\begin{aligned}y_{n+1} &= y_n + \frac{h}{2}[y'_{n+1} + y'_n] \\ \hat{E}_n &= -\frac{1}{12}h^3Y^{(3)}(\xi_n)\end{aligned}$$

$q = 2$:

$$\begin{aligned}y_{n+1} &= y_n + \frac{h}{12}[5y'_{n+1} + 8y'_n - y'_{n-1}] \\ \hat{E}_n &= -\frac{1}{24}h^4Y^{(4)}(\xi_n)\end{aligned}$$

$q = 3$:

$$\begin{aligned}y_{n+1} &= y_n + \frac{h}{24}[9y'_{n+1} + 19y'_n - 5y'_{n-1} + y'_{n-2}] \\ \hat{E}_n &= -\frac{19}{720}h^5Y^{(5)}(\xi_n)\end{aligned}$$

CONVERGENCE and STABILITY. The convergence and stability properties of Adams-Bashforth methods extend to Adams-Moulton methods. In particular, the results given in (7) and (8).

From earlier, the backward Euler method and the trapezoidal method are A-stable methods. This is not true of higher order Adams-Moulton methods. But such methods do have very desirable stability and convergence properties. These are important enough as to justify the cost of solving the implicit methods.

As earlier, we often use fixed point iteration to solve the implicit equation. For the example with $q = 2$, we use

$$y_{n+1}^{(k+1)} = y_n + \frac{h}{12} [5f(x_{n+1}, y_{n+1}^{(k)}) + 8y'_n - y'_{n-1}]$$

for $k = 0, 1, \dots$. The initial guess $y_{n+1}^{(0)}$ is often obtained by using an Adams-Bashforth method of a comparable order ($q = 1$ or $q = 2$ in this case).

GENERAL REMARKS

Most large scale packages for solving differential equations to high accuracy are based on using Adams-Bashforth and Adams-Moulton methods, usually varying both the stepsize h and the order $q + 1$ of the method.

Matlab contains such a code, called **ode113**.