Adams-Bashforth and Adams-Moulton methods

Definitions

Linear multistep methods are used for the numerical solution of ordinary differential equations, in particular the initial value problem

$$y' = f(t,y) \quad \text{with} \quad y(t_0) = y_0 \,.$$

The Adams-Bashforth methods and Adams-Moulton methods are described on the Linear multistep method page.

Derivation

There are (at least) two ways that can be used to derive the Adams-Bashforth methods and Adams-Moulton methods. We will demonstrate the derivations using polynomial interpolation and using Taylor's theorem for the two-step Adams-Bashforth method.

Derive the two-step Adams-Bashforth method by using polynomial interpolation

From the w:Fundamental theorem of Calculus we can get

(1)

Set

(2)

To get the value of A, we can use an interpolating polynomial $P(t)$ as an approximation of $f(t,y(t))$. The interpolation polynomial in the Lagrange form is a linear combination

$$L(x) := \sum_{j=0}^{k} y_j \ell_j(x)$$

where

$$\ell_j(x) := \prod_{\substack{0\le m\le k \\ m\ne j}} \frac{x-x_m}{x_j-x_m} \,.$$

Then the interpolation can be

$$P(t) = f(t_n, y_n)\frac{t - t_{n-1}}{t_n - t_{n-1}} + f(t_{n-1}, y_{n-1})\frac{t - t_n}{t_{n-1} - t_n}\,.$$

Thus, (2) becomes

(3)

Integrating and simplifying, the right hand side of equation (3) becomes

$$\tfrac{1}{2}(t_n + t_{n+1})(f(t_n, y_n) - f(t_{n-1}, y_{n-1})) - t_{n-1} f(t_n, y_n) + t_n f(t_{n-1}, y_{n-1})$$

$$= \quad \tfrac{1}{2}(t_n + t_{n+1} - 2t_{n-1}) f(t_n, y_n) + \tfrac{1}{2}(2t_n - t_{n+1} - t_n) f(t_{n-1}, y_{n-1})\,.$$

Since t_{n-1}, t_n and t_{n+1} are equally spaced, then $t_n - t_{n-1} = t_{n+1} - t_n = h$. Therefore, the value of A is

$$A = \frac{3}{2} h f(t_n, y_n) - \frac{1}{2} h f(t_{n-1}, y_{n-1})$$

Putting this value back to (1) yields

$$y(t_{n+1}) \approx y(t_n) + \frac{3}{2} h f(t_n, y_n) - \frac{1}{2} h f(t_{n-1}, y_{n-1})\,.$$

Thus, the equation $y_{n+1} = y_n + \dfrac{3}{2} h f(t_n, y_n) - \dfrac{1}{2} h f(t_{n-1}, y_{n-1})$ is the two-step Adams-Bashforth method.

Derive the two-step Adams-Bashforth method by using Taylor's theorem

To simplify, let's set $f_k = f(x_k, y_k)$. Then the general form of Adams-Bashforth method is

(4)

where $\displaystyle\sum_{k=1}^{r} \lambda_k = 1$. For the two-step Adams-Bashforth method, let's set $\lambda_1 = 1 - \lambda, \lambda_2 = \lambda$.
Then (4) becomes

$$\begin{aligned} y_{n+2} &= y_{n+1} + h((1-\lambda) f_{n+1} + \lambda f_n) \\ &= y(t_{n+1}) + h((1-\lambda) y'(t_{n+1}) + \lambda y'(t_n))\,. \end{aligned}$$

By using Taylor's theorem, expand $y'(t_n)$ at t_{n+1} to get

$$y_{n+2} = y(t_{n+1}) + h((1-\lambda)y'(t_{n+1}) + \lambda(y'(t_{n+1}) - hy''(t_{n+1}) + O(h^2))\,.$$

Thus, the simplified form is

(5)

Expanding $y(t_{n+2})$ at $y(t_{n+1})$ yields

(6)

Subtracting (5) from (6) and then requiring the h^2 term to cancel makes $\lambda = -\frac{1}{2}$. The two-step Adams-Bashforth method is then

$$y_{n+2} = y_{n+1} + \frac{3}{2}hf(t_{n+1}, y_{n+1}) - \frac{1}{2}hf(t_n, y_n)$$

Since

$$y(t_{n+2}) - y_{n+2} = O(h^3)\,.$$

the local truncation error is of order $O(h^3)$ and thus the method is second order. (See w: Linear multistep method#Consistency and order and w:Truncation error)

Exercises

1. Derive three-step Adams-Bashforth method by using polynomial interpolation

Solution:

The initial problem is $y' = f(t, y)$. Then we can get:

(8)

let's set:

(9)

Then,we can use P(t) as an interpolation of f(t,y(t)). To derive the three-step Adams-bashforth method, the interpolation polynomial is:

$$P(t) = f(t_n, y_n)\frac{(t-t_{n-1})(t-t_{n-2})}{(t_n-t_{n-1})(t_n-t_{n-2})} + f(t_{n-1}, y_{n-1})\frac{(t-t_n)(t-t_{n-2})}{(t_{n-1}-t_n)((t_{n-1}-t_{n-2})} + f(t_{n-2}, y_{n-2})\frac{(t-t_n)(t-t_{n-1})}{(t_{n-2}-t_n)(t_{n-2}-t_{n-1})}$$

Since t_{n-2}, t_{n-1}, t_n and t_{n+1} are equally spaced, then

$t_{n-1} - t_{n-2} = t_n - t_{n-1} = t_{n+1} - t_n = h.$

So the integral of each part of P(t) is :

$$\frac{f(t_n, y_n)}{2h^2} \int_{t_n}^{t_{n+1}} (t - t_{n-1})(t - t_{n-2})\, dt = \frac{23}{12} h f(t_n, y_n)$$

$$-\frac{f(t_{n-1}, y_{n-1})}{h^2} \int_{t_n}^{t_{n+1}} (t - t_n)(t - t_{n-2})\, dt = -\frac{4}{3} h f(t_{n-1}, y_{n-1})$$

$$\frac{f(t_{n-2}, y_{n-2})}{2h^2} \int_{t_n}^{t_{n+1}} (t - t_{n-2})(t - t_{n-1})\, dt = \frac{5}{12} h f(t_{n-2}, y_{n-2})$$

Thus the approximate value of A is

$$A \approx \frac{23}{12} h f(t_n, y_n) - \frac{4}{3} h f(t_{n-1}, y_{n-1}) + \frac{5}{12} h f(t_{n-2}, y_{n-2})$$

Put this value back to (8)

$$y(t_{n+1}) \approx y(t_n) + \frac{23}{12} h f(t_n, y_n) - \frac{4}{3} h f(t_{n-1}, y_{n-1}) + \frac{5}{12} h f(t_{n-2}, y_{n-2})$$

Thus, the three-step Adams-Bashforth method is

$$y_{n+1} = y_n + h\left(\frac{23}{12} f(t_n, y_n) - \frac{4}{3} f(t_{n-1}, y_{n-1}) + \frac{5}{12} f(t_{n-2}, y_{n-2})\right)$$

2. Derive the second-order Adams-Moulton method by using Taylor's theorem

Solution:

To simplify, note f, f'_t, f'_y to represent $f(t_n, y_n), f'_t(t_n, y_n), f'_y(t_n, y_n)$.From Taylor expansion of the binary function (w:Taylor's theorem):

$$f_{n+1} = f(t_{n+1}, y_{n+1}) = f + h f'_t + f'_y (y_{n+1} - y_n) + O[h^2 + (y_{n+1} - y_n)^2]$$

Then by using:

(10)

can get:

$$f_{n+1} = f + h f'_t + (1 - \lambda) h f f'_y + \lambda h f_{n+1} f'_y + O(h^2)$$

Therefore, simplify this equation to get:

$$f_{n+1} = \frac{f + hf'_t + (1-\lambda)hff'_y + O(h^2)}{1 - \lambda hf'_y}$$

$$= [f + hf'_t + (1-\lambda)hff'_y + O(h^2)][1 + \lambda hf'_y + O(h^2)]$$

$$= f + h(f'_t + ff'_y) + O(h^2)$$

$$= y'(t_n) + hy''(t_n) + O(h^2)$$

Thus, subtitute f_{n+1} into (**10**):

$$y_{n+1} = y_n + h[(1-\lambda)f_n + \lambda f_{n+1}]$$

$$= y(t_n) + hy'(t_n) + \lambda h^2 y''(x_n) + O(h^3)$$

Combining the equation:

$$y(t_{n+1}) - y_{n+1} = O(h^3)$$

and Taylor expansion:

$$y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{1}{2}hy''(t_n) + O(h^3)$$

can get $\lambda = \frac{1}{2}$. Therefore, the second-order Adams-moulton method is:

$$y_{n+1} = y_n + \frac{h}{2}(f(t_{n+1}, y_{n+1}) + f(t_n, y_n))$$

Predictor–corrector method

To solve an ordinary differential equation (ODE), a w:Predictor–corrector method is an algorithm that can be used in two steps. First, the prediction step calculates a rough approximation of the desired quantity, typically using an explicit method. Second, the corrector step refines the initial approximation using another means, typically an implicit method.

Here mainly discuss about using Adams-bashforth and Adams-moulton methods as a pair to construct a predictor–corrector method.

Example: Adams predictor–corrector method

Let's start from the two-step Adams method. The prediction step is to use two-step Adams-bashforth:

$$P_{n+1} = y_n + \frac{3}{2}hf(t_n, y_n) - \frac{1}{2}hf(t_{n-1}, y_{n-1})$$

Then, by using two-step Adams-moulton the corrector step can be:

$$y_{n+1} = y_n + \tfrac{1}{2}h\big(f(t_{n+1}, P_{n+1}) + f(t_n, y_n)\big)$$

Also, by using four-step Adams-bashforth and Adams-moulton methods together, the predictor-corrector formula is:

$$\begin{cases} P_{n+1} = y_n + \frac{h}{24}(55f(t_n, y_n) - 59f(t_{n-1}, y_{n-1}) + 37f(t_{n-2}, y_{n-2}) - 9f(t_{n-3}, y_{n-3})) \\ y_{n+1} = y_n + \frac{h}{24}(9f(t_{n+1}, P_{n+1}) + 19f(t_n, y_n) - 5f(t_{n-1}, y_{n-1}) + f(t_{n-2}, y_{n-2})) \end{cases}$$

Note, the four-step Adams-bashforth method needs four initial values to start the calculation. It needs to use other methods, for example Runge-Kutta, to get these initial values.

Matlab programs

Four-step Adams predictor-corrector method:

```
function [x y]=maadams4(dyfun,xspan,y0,h)
% use four-step Adams predictor-corrector method to solve an ODE
y'=f(x,y), y(x0)=y0
% inputs: dyfun -- the function f(x,y), as an inline
% xspan -- the interval [x0,xn]
% y0 -- the initial value
% h -- the step size
% output: x, y -- the node and the value of y
x=xspan(1):h:xspan(2);
% use Runge-Kutta method to get four initial values
[xx,yy]=marunge(dyfun,[x(1),x(4)],y0,h);
y(1)=yy(1);y(2)=yy(2);
y(3)=yy(3);y(4)=yy(4);
for n=4:(length(x)-1)
p=y(n)+h/24*(55*feval(dyfun,x(n),y(n))-59*feval(dyfun,x(n-1),y(n-
1))+37*feval(dyfun,x(n-2),y(n-2))-9*feval(dyfun,x(n-3),y(n-3)));
y(n+1)=y(n)+h/24*
(9*feval(dyfun,x(n+1),p)+19*feval(dyfun,x(n),y(n))-5*feval(dyfun,x(n-
1),y(n-1))+feval(dyfun,x(n-2),y(n-2)));
end
x=x';y=y';
```

Runge-Kutta method:

```
function [x y]=marunge(dyfun,xspan,y0,h)
x=xspan(1):h:xspan(2);
y(1)=y0;
for n=1:(length(x)-1)
 k1=feval(dyfun,x(n),y(n));
 k2=feval(dyfun,x(n)+h/2,y(n)+h/2*k1);
 k3=feval(dyfun,x(n)+h/2,y(n)+h/2*k2);
 k4=feval(dyfun,x(n+1),y(n)+h*k3);
 y(n+1)=y(n)+h*(k1+2*k2+2*k3+k4)/6;
end
x=x';y=y'
```

References

- http://www-users.cselabs.umn.edu/classes/Spring-2011/csci5302/Notes/Multistep_ABM.pdf
- http://www.phy.ornl.gov/csep/ode/node12.html
- https://www.pramanaresearch.org/gallery/prjp%20-%201509.pdf