Multistep Methods

Sunday, November 8, 2020 11:05 PM

The two main classes of ODE numerical solvers are:

① ~~Multi-stage~~ One-step, e.g., Runge-Kutta

w_{i+1} depends on w_i (and intermediate points)
but not on w_{i-1}, w_{i-2}, etc.

② Multi-step

Why use intermediate points when we could use previous points w_i, w_{i-1}, w_{i-2}, etc. "for free"

High-level advantages/disadvantages

One-step/multi-stage are easier to adapt stepsize since changing h is problematic for multi-step methods (they use a special interpolant to deal with this)

Multi-step methods save some computation, and it's a bit easier to estimate local error (since all methods are "embedded")

Both are used in practice

Multi-step Methods

Solving $y' = f(t,y)$, $a \le t \le b$, $y(a) = y_0$ ... as usual

Generic form of a m-step method is

$m > 1$ for "multi-step" methods

$$w_{i+1} = \overbrace{a_{m-1} w_i + a_{m-2} w_{i-1} + \dots + a_0 w_{i+1-m}}^{m\text{-terms}}$$

$$+ h \cdot \big(\underbrace{b_m f_{i+1} + b_{m-1} f_i + \dots + b_0 f_{i+1-m}}_{m+1 \text{ terms}} \big)$$

where $f_i := f(t_i, w_i)$ (and the point is we already calculated that)

Wait a minute! $f_{i+1} = f(t_{i+1}, w_{i+1})$ depends on (the unknown) w_{i+1} !

So, if $b_m \neq 0$, it's an implicit method (Burden + Faires say "open")

$b_m = 0$, it's an explicit method (" ___ " "closed")

Details: $w_0 = y_0$ as always.

but w_1 ? If $m > 1$, we need "w_{-1}" etc.

So usually we do RK until we have enough history to start the multi-step method.

3 main types of multi-step methods:

1. Adams-Bashforth "AB" (explicit)
2. Adams-Moulton "AM" (implicit)
3. Backward Differentiation "BD" (implicit)

$\boxed{\text{Adams - methods}}$ ①, ②

$a_{m-1} = 1$, all other $a_i = 0$

$$w_{i+1} = a_{m-1} w_i + \underbrace{\cancel{a_{m-2} w_{i-1} + \dots + a_0 w_{i+1-m}}}_{=0} + h \cdot \left(b_m f_{i+1} + b_{m-1} \cdot f_i + \dots + b_0 \cdot f_{i+1-m} \right)$$

	Name	Order	Steps m	b_m	b_{m-1}	b_{m-2}	b_{m-3}	b_{m-4}
Adams-Bashforth	AB1 = Euler	1	1	0	1			
	AB2	2	2	0	$3/2$	$-1/2$		
	AB3	3	3	0	$23/12$	$-16/12$	$5/12$	
	AB4	4	4	0	$55/24$	$-59/24$	$37/24$	$-9/24$
Adams-Moulton	AM1 = Backward Euler	1	1	1				
	AM2 = trapezoid	2	1	$1/2$	$1/2$			
	AM3	3	2	$5/12$	$8/12$	$-1/12$		
	AM4	4	3	$9/24$	$19/24$	$-5/24$	$1/24$	
	AM5	5	4	$251/720$	$646/720$	$-264/720$	$106/720$	$-19/720$

explicit ($b_m = 0$)

$\boxed{\text{Backward Differentiation}}$ ③

all $b_i = 0$ except b_m

$$w_{i+1} = a_{m-1} w_i + a_{m-2} w_{i-1} + \dots + a_0 w_{i+1-m} + h \cdot \left(b_m f_{i+1} \underbrace{\cancel{+ b_{m-1} \cdot f_i + \dots + b_0 \cdot f_{i+1-m}}}_{=0} \right)$$

Name	Order	Steps m	a_{m-1}	a_{m-2}	a_{m-3}	a_{m-4}	b_m
BD1 = Backward Euler	1	1	1				1
BD2	2	2	$4/3$	$-1/3$			$2/3$
BD3	3	3	$18/11$	$-9/11$	$2/11$		$6/11$
BD4	4	4	$48/25$	$-36/25$	$16/25$	$-3/25$	$12/25$

Note: (Forward) Euler is both a RK and AB method
(Backward/Implicit) Euler is a RK, AM and BD method

Where do the numbers come from?

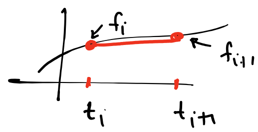
Adams methods (AB, AM):

$$y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_{i+1}} \underbrace{y'(t)}_{=f(t,y(t)) \text{ via ODE}} dt \quad \text{via F.T.C.}$$

so treat as integration problem.

Use t_j as nodes (if we include t_{i+1}, it's AM
else it's AB)

Ex: AM2 is aka trapezoid method
since we interpolate at t_i and t_{i+1} (so linear interpolant)



(this is implicit)

Error: AB interpolates on $m-1$ nodes, so degree $m-2$ polynomial, $O(h^{m-1})$ interpolation error
AM interpolates on m nodes, $m-1$, $O(h^m)$

so integrate this error, $O(h^m)$ / $O(h^{m+1})$ error, and local truncation error

divides this by h, so "order of error" is $O(h^{m-1})$ AB, $O(h^m)$ AM

Backward Differentiation Methods

As for Adams methods, interpolate w/ a polynomial $p(t)$,
then estimate $\underbrace{y'(t_{i+1})}_{\approx f(t_{i+1}, w_{i+1})}$ by $\underbrace{p'(t_{i+1})}_{\text{in terms of } w_{i+1}, w_i, w_{i-1}, \ldots}$.

and then re-arrange.

So, based on finite differences not quadrature

Predictor-Corrector Methods

Implicit methods, like AM, have nice properties but implementation is a pain since we have to solve a root-finding problem every iteration (except in a few lucky cases when you can solve it by hand).

Predictor-Corrector idea is if we want

$$w_{i+1} = \ldots + f(t_{i+1}, w_{i+1})$$

then replace with a "predicted" value,

$$\underset{\uparrow \text{"corrected"}}{w_{i+1}} = \ldots + f(t_{i+1}, w_{\text{predicted}})$$

Commonly use AB to compute w_p then AM for final "corrected" answer.

and sometimes called "ABM"

Sometimes referred to as $P(EC)^1$ methods

explicit corrector

you can do $P(EC)^k$ meaning predictor – multicorrector (you iterate)
but diminishing gains.

(Principle: don't solve the problem too accurately since it's only an approximation anyway. i.e., solve approximations approximately)

... this is what our modified Euler was doing!

modified Euler

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

means

$$k_1 = h\overbrace{f(t_i, w_i)}^{\text{"}f_i\text{"}}$$

$$k_2 = h f(\underbrace{t_i + 1\cdot h}_{t_{i+1}}, w_i + 1 k_1)$$

$$w_{i+1} = w_i + \tfrac{1}{2} k_1 + \tfrac{1}{2} k_2$$

So... modified Euler is like
trapezoid / Crank-Nicolson

$$w_{i+1} = w_i + h\left(\tfrac{1}{2} f_i + \tfrac{1}{2} f_{i+1}\right), \quad f_{i+1} = f(t_{i+1}, w_{i+1})$$

but using forward Euler as the predictor,

$$w_p = w_i + h f_i$$

$$w_{i+1} = w_i + h\left(\tfrac{1}{2} f_i + \tfrac{1}{2} f(t_{i+1}, w_p)\right)$$

Many other examples, e.g. book mentions explicit Milne's method (predictor)
w/ implicit Simpson's Method

You can analyze the order of these predictor – corrector methods
cf. Quarteroni et al.