Multistep Methods

Multistep methods for solving initial value problems are essentially interpolation formulae. A polynomial is used (although there are some rational function and splines sometimes employed) to interpolate some subset of solution approximations $y_k, y_{k-1}, ..., y_{k-p}$ and derivative approximations $y'_k, y'_{k-1}, ..., y'_{k-p}$. A method that uses history going back to y_{k-p} or y'_{k-p} is called a "*p+1* step method" (e.g. Euler's method is a one step method). Assuming equal spacing in the steps (i.e., $h_k = h_{k-1} = ... = h_{k-p}$), the basic form of a multistep method is

$$y_{k+1} = a_0 y_k + a_1 y_{k-1} + ... + a_p y_{k-p} + h_k (b_0 y'_k + b_1 y'_{k-1} + ... + b_p y'_{k-p}). \tag{0.1}$$

It should be emphasized that some of the coefficients a_i or b_i may not be present (equivalently, are set to zero). In the Adams methods the only non-zero a_i is a_0. In the backward differentiation methods (sometimes called "Gear methods") there is only one non-zero b_i.

The interpolation corresponds to the non-zero character of the coefficients:

> if a_i is allowed to be non-zero then the value y_{k-i} will be interpolated at x_{k-i} and if b_i is allowed to be non-zero then the value y'_{k-i} will be the derivative of the interpolant at x_{k-i}

The coefficients turn out to be independent of h_k and x_k (although this fact depends upon polynomial interpolation). A question one might ask is "why set any of the coefficients to zero? Why not use all of the information that is available?". The answer is that polynomial interpolation sometimes does a poor job of tracking the solutions to differential equations – sometimes these methods seem to take on lives of their own. As an example, consider the method

$$y_{k+1} = 11 y_k - 7 y_{k-1} - 3 y_{k-2} - 2hy'_k - 10hy'_{k-1} \text{ for } k = 3, 4, ..., \tag{0.2}$$

applied to the problem

$$y'(x) = -y(x)$$
$$y(0) = 1$$

The solution is the slowly decaying function $y(x) = e^{-x}$. Yet even with the small step size of $.01$ and with the values of y_0, y_1, y_2, y'_1 and y'_2 having no error at all, the multistep method not only fails to track the true solution but grows like e^{233x}.

x	computed solution	true solution	error
0.00	1.00000000000000	1.00000000000000	0.00E+00
0.01	0.99004983374917	0.99004983374917	0.00E+00
0.02	0.98019867330676	0.98019867330676	0.00E+00
0.03	0.97044552697118	0.97044553354851	-6.58E-09
0.04	0.96078936015833	0.96078943915232	-7.90E-08
0.05	0.95122859292337	0.95122942450071	-8.32E-07
0.06	0.94175592800951	0.94176453358425	-8.61E-06
0.07	0.93230495501850	0.93239381990595	-8.89E-05
0.08	0.92219892226816	0.92311634638664	-9.17E-04
0.09	0.90446014973899	0.91393118527123	-9.47E-03
0.10	0.80706342141785	0.90483741803596	-9.78E-02

x	computed solution	true solution	error
0.11	-0.11353289597880	0.89583413529653	-1.01E+00
0.12	-9.53325057068657	0.88692043671716	-1.04E+01
0.13	-1.0669423457E+02	0.87809543092056	-1.08E+02
0.14	-1.1096504373E+03	0.86935823539881	-1.11E+03
0.15	-1.1463557849E+04	0.86070797642506	-1.15E+04
0.16	-1.1835173678E+05	0.85214378896621	-1.18E+05
0.17	-1.2218086388E+06	0.84366481659638	-1.22E+06
0.18	-1.2613313542E+07	0.83527021141127	-1.26E+07
0.19	-1.3021318042E+08	0.82695913394336	-1.30E+08
0.20	-1.3442519589E+09	0.81873075307798	-1.34E+09

The form of the multistep method presented in (0.1) is called "explicit" because all of the data necessary to compute y_{k+1} are found on the right hand side of the equation. An alternative form of multistep methods is the "implicit" method

$$y_{k+1} = a_0 y_k + a_1 y_{k-1} + \ldots + a_p y_{k-p} + h_k(b_{-1} f(x_{k+1}, y_{k+1}) + b_0 y'_k + b_1 y'_{k-1} + \ldots + b_p y'_{k-p}) \quad (0.3)$$

where y_{k+1} on both sides of the equation. In general this is a non-linear equation in y_{k+1}. The extra complexity of solving such an equation for y_{k+1} is often compensated for by better tracking of the true solutions. As an implicit analog to the explicit method (0.2) of the previous example, consider

$$y_{k+1} = \frac{9}{17} y_k + \frac{9}{17} y_{k-1} - \frac{1}{17} y_{k-2} + h_k\left(\frac{6}{17} f(x_{k+1}, y_{k+1}) + \frac{18}{17} y'_k\right) \text{ for } k = 3, 4, \ldots, \quad (0.4)$$

Whereas method (0.2) uses derivative values at x_k and x_{k-1}, method (0.4) uses derivative values at x_{k+1} and x_k. Applied to the same problem as before with the same three starting values, we obtain:

x	computed solution	true solution	error
0.00	1.00000000000000	1.00000000000000	0.00E+00
0.01	0.99004983374917	0.99004983374917	0.00E+00
0.02	0.98019867330676	0.98019867330676	0.00E+00
0.03	0.97044553354678	0.97044553354851	-1.73E-12
0.04	0.96078943914972	0.96078943915232	-2.60E-12
0.05	0.95122942449677	0.95122942450071	-3.94E-12
0.06	0.94176453357926	0.94176453358425	-4.99E-12
0.07	0.93239381989978	0.93239381990595	-6.17E-12
0.08	0.92311634637941	0.92311634638664	-7.23E-12
0.09	0.91393118526290	0.91393118527123	-8.33E-12
0.10	0.90483741802659	0.90483741803596	-9.37E-12

x	computed solution	true solution	error
0.11	0.89583413528612	0.89583413529653	-1.04E-11
0.12	0.88692043670575	0.88692043671716	-1.14E-11
0.13	8.7809543091E-01	0.87809543092056	-1.24E-11
0.14	8.6935823539E-01	0.86935823539881	-1.34E-11
0.15	8.6070797641E-01	0.86070797642506	-1.43E-11
0.16	8.5214378895E-01	0.85214378896621	-1.52E-11
0.17	8.4366481658E-01	0.84366481659638	-1.62E-11
0.18	8.3527021139E-01	0.83527021141127	-1.70E-11
0.19	8.2695913393E-01	0.82695913394336	-1.79E-11
0.20	8.1873075306E-01	0.81873075307798	-1.88E-11

One approach to solving the equations specifying implicit methods is to "predict" a value of y_{k+1} with an explicit method, and the iteratively cycle on the implicit method with the left hand side of the equation being evaluated with the current estimate of y_{k+1} to produce a new estimate of y_{k+1}. This step is called "correction". Although one might apply correction until convergence is reached (i.e. the same value used on the right hand side is computed for the left hand side), generally only a few steps of correction are applied. (Furthermore, there is no guarantee in general that iterating the corrector in this fashion will produce convergence.)

One popular form of predictor-corrector pairs is the Adams methods. The explicit predictors have the form

$$y_{k+1} = y_k + h_k(b_0 y'_k + b_1 y'_{k-1} + \ldots + b_p y'_{k-p}) \quad (0.5)$$

and are called Adams-Bashforth methods. The implicit correctors have the form

$$y_{k+1} = y_k + h_k(b_{-1} f(x_{k+1}, y_{k+1}) + b_0 y'_k + b_1 y'_{k-1} + \ldots + b_p y'_{k-p}) \quad (0.6)$$

and are called Adams-Moulton methods. Backwards differentiation methods – sometimes called Gear methods – are implicit and of the form

$$y_{k+1} = a_0 y_k + a_1 y_{k-1} + \ldots + a_p y_{k-p} + h_k b_{-1} f(x_{k+1}, y_{k+1}).$$