HIGHER ORDER METHODS

There are two principal means to derive higher order methods

$$y_{n+1} = \sum_{j=0}^{p} a_j y_{n-j} + h \sum_{j=-1}^{p} b_j f(x_{n-j}, y_{n-j})$$

(a) Method of Undetermined Coefficients
(b) Numerical Integration

UNDETERMINED COEFFICIENTS

Pick $p > 0$ and $m > 1$. Solve the linear system

$$\sum_{j=0}^{p} a_j = 1$$

$$\sum_{j=0}^{p} (-1)^i a_j + i \sum_{j=-1}^{p} (-1)^{i-1} b_j = 1$$

for $i = 1, ..., m$.

EXAMPLE

For $p = 1$, the general two-step formula is

$$\begin{aligned} y_{n+1} &= a_0 y_n + a_1 y_{n-1} \\ &\quad + h\left[b_{-1} y_{n+1} + b_0 y_n + b_1 y_{n-1}\right] \end{aligned} \tag{1}$$

For order $m = 2$, we must have the coefficients satisfy the system

$$\begin{aligned} a_0 + a_1 &= 1 \\ -a_1 + b_{-1} + b_0 + b_1 &= 1 \\ a_1 + 2\left[b_{-1} - b_1\right] &= 1 \end{aligned}$$

These three equations have two degrees of freedom; and their general solution is

$$\begin{aligned} a_1 &= 1 - a_0 \\ b_{-1} &= 1 - \tfrac{1}{4} a_0 - \tfrac{1}{2} b_0 \\ b_1 &= 1 - \tfrac{3}{4} a_0 - \tfrac{1}{2} b_0 \end{aligned} \tag{2}$$

SOLVING THE LINEAR SYSTEM

The linear system has the matrix

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 0 & -2 & 1 \end{bmatrix}$$

By elementary row operations, this can be reduced to

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ \frac{1}{4} & 0 & 1 & \frac{1}{2} & 0 & 1 \\ \frac{3}{4} & 0 & 0 & \frac{1}{2} & 1 & 1 \end{bmatrix}$$

From this, we can solve for a_1, b_{-1}, b_1.

For the truncation error in the resulting method, recall the formula

$$T_n(Y) = \frac{c_{m+1}}{(m+1)!} h^{m+1} Y^{(m+1)}(x_n) + O\left(h^{m+2}\right)$$

$$c_{m+1} = 1 - \left[\sum_{j=0}^{p} (-j)^{m+1} a_j + (m+1) \sum_{j=-1}^{p} (-j)^m b_j\right]$$

For our case with $m = 2$, we have

$$T_n(Y) = \frac{c_3}{3!} h^3 y'''(x_n) + O(h^4) \tag{3}$$

$$c_3 = -4 + 2a_0 + 3b_0 \tag{4}$$

How might we choose a_0 and b_0?

To apply the earlier convergence theory of §6.3, we would want to require

$$0 \le a_0, a_1 \le 1$$

Subject to this, we could choose a_0 and b_0 to

$$\text{Minimize} \left| \frac{-4 + 2a_0 + 3b_0}{6} \right|$$

In fact, for any a_0 in $[0, 1]$, we can choose b_0 so as to make this quantity zero, namely let

$$b_0 = \frac{4 - 2a_0}{3}$$

Then

$$T_n(Y) = O(h^4)$$

and we could actually calculate a formula

$$T_n(Y) = \frac{c_4}{4!} h^4 y^{(4)}(x_n) + O(h^5)$$

In fact, we can then go ahead and choose $a_0 = 0$, $b_0 = \frac{4}{3}$, leading to the method

$$y_{n+1} = y_{n-1} + \frac{h}{3} \left[y'_{n+1} + 4y'_n + y'_{n-1} \right] \qquad (5)$$

This method has a truncation error of

$$T_n(Y) = -\frac{h^5}{90} y^{(5)}(\xi_n)$$

It arises from applying Simpson's numerical integration rule to

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

See pages 256-258 for Simpson's quadrature rule.

The above formula (5) is called *Milne's method*. It appears to be quite a good formula. It is convergent with an error of $O(h^4)$; and it is stable in the original sense of §6.2 and §6.3. Unfortunately, it is only weakly stable, and this appears when x_n is sufficiently large.

EXAMPLE - CONTINUED

Recall (1)

$$\begin{aligned} y_{n+1} &= a_0 y_n + a_1 y_{n-1} \\ &\qquad + h\left[b_{-1} y_{n+1} + b_0 y_n + b_1 y_{n-1}\right] \end{aligned}$$

with the choice of coefficients from (2),

$$\begin{aligned} a_1 &= 1 - a_0 \\ b_{-1} &= 1 - \tfrac{1}{4} a_0 - \tfrac{1}{2} b_0 \\ b_1 &= 1 - \tfrac{3}{4} a_0 - \tfrac{1}{2} b_0 \end{aligned}$$

If we want an explicit method, then we must force $b_{-1} = 0$. Doing so yields the choice of coefficients

$$a_1 = 1 - a_0, \quad b_0 = 2 - \frac{1}{2} a_0, \quad b_1 = -\frac{1}{2} a_0$$

Again, our earlier convergence and stability theory works if we choose $0 \leq a_0 \leq 1$. The truncation error from (3)-(4) becomes

$$T_n(Y) = \left(\frac{a_0}{12} + \frac{1}{3}\right) h^3 Y'''(x_n) + O(h^4)$$

For $0 \leq a_0 \leq 1$, the leading coefficient is minimized by choosing $a_0 = 0$. This yields the numerical method

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n)$$

which is simply the midpoint method. Thus choosing a_0 so as to just minimize the truncation error is not an adequate basis for choosing a method.

In contrast, choosing $a_0 = 1$ leads to the method

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

with a truncation error

$$T_n(Y) = \frac{5}{12}h^3 Y'''(x_n) + O(h^4)$$

This is the *Adams-Bashforth method* of order 2, and it does not have the weak stability property.

NUMERICAL INTEGRATION

The most popular multistep formulas in use today are based on numerical integration, by means of suitably chosen polynomial interpolation formulas. Integrate the equation

$$Y'(x) = f\left(x, Y(x)\right)$$

over some interval $[x_{n-r}, x_{n+1}]$. This yields

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

Approximate $Y'(x) = f\left(x, Y(x)\right)$ with a polynomial interpolant, and then integrate this polynomial to approximate the integral. The most satisfactory formulas have been found by doing the case

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

and these are called the *Adams family* of methods.

POLYNOMIAL INTERPOLATION

Let $t_0, ..., t_p$ be $p + 1$ distinct node points, and let $z_0, ..., z_p$ be corresponding function values. The unique polynomial of degree $\leq p$ which interpolates these values at the given node points is given by

$$\mathcal{P}_p(t) = \sum_{j=0}^{p} z_j \ell_j(t) \tag{6}$$

$$\ell_j(t) = \prod_{\substack{i=0 \\ i \neq j}}^{p} \left(\frac{t - t_i}{t_j - t_i} \right)$$

With this definition, $\deg(\ell_j) = p$ and

$$\ell_j(t_i) \equiv \delta_{i,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

Formula (6) is called the *Lagrange form* of the interpolation polynomial.

ERROR FORMULA

Let the data be generated by

$$z_j = g(t_j), \quad j = 0, 1, ..., p$$

Then

$$\begin{aligned} g(t) - \mathcal{P}_p(t) &= \frac{(t - t_0) \cdots (t - t_p)}{(p + 1)!} g^{(p+1)}(\zeta_t) \\ &= (t - t_0) \cdots (t - t_p) g[t_0, ..., t_p, t] \end{aligned}$$

with $g[t_0, ..., t_p, t]$ a *Newton divided difference* of order $p + 1$ (cf. §3.2).

BACKWARD DIFFERENCE FORMULA

Assume the data points are evenly spaced, say

$$t_j = t_0 + jh, \quad j = 0, 1, ..., p$$

Introduce the *backward differences*:

$$\nabla g_j \equiv \nabla g(t_j) = g(t_j) - g(t_{j-1}) = g_j - g_{j-1}$$

$$\nabla^2 g_j = \nabla g_j - \nabla g_{j-1}$$

$$\nabla^k g_j = \nabla^{k-1} g_j - \nabla^{k-1} g_{j-1}$$

Then the interpolation polynomial $\mathcal{P}_n(t)$ can be written as

$$\begin{aligned} \mathcal{P}_p(t) &= g_p + \frac{(t - t_p)}{h} \nabla g_p \\ &\quad + \frac{(t - t_p)(t - t_{p-1})}{2!\, h^2} \nabla^2 g_p + \cdots \\ &\quad + \frac{(t - t_p) \cdots (t - t_1)}{p!\, h^p} \nabla^p g_p \end{aligned}$$

With this, we can estimate the error with

$$\begin{aligned} g(t) - \mathcal{P}_p(t) &\approx \mathcal{P}_{p+1}(t) - \mathcal{P}_p(t) \\ &= \frac{(t - t_p) \cdots (t - t_0)}{(p + 1)!\, h^{p+1}} \nabla^{p+1} g_p \end{aligned}$$

where the interpolation nodes for constructing $\mathcal{P}_{p+1}(t)$ are $t_p, ..., t_0, t_{-1}$. This will lead to a convenient way to estimate the truncation error in our numerical methods.

ADAMS-BASHFORTH METHODS

Interpolate $Y'(x) = f\left(x, Y(x)\right)$ at the node points

$$x_n, x_{n-1}, ..., x_{n-p}$$

For linear interpolation, the nodes are x_n, x_{n-1}, and

$$\mathcal{P}_1(x) = \frac{(x_n - x)Y'(x_{n-1}) + (x - x_{n-1})Y'(x_n)}{h}$$

Integrating this over $[x_n, x_{n+1}]$,

$$\begin{aligned}\int_{x_n}^{x_{n+1}} f(x, Y(x))\, dx &\approx \int_{x_n}^{x_{n+1}} \mathcal{P}_1(x)\, dx \\ &= \frac{h}{2}\left[3Y_n' - Y_{n-1}'\right]\end{aligned}$$

This leads to the 2-step method

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

The truncation formula is

$$T_n(Y) = \frac{5}{12} h^3 Y'''(x_n) + O(h^4)$$

We can continue with this, using interpolation of degree p. This leads to a $(p+1)$-step formula of order $p+1$. The resulting formulas are given on page 387 of the text. For example, with $p = 2$,

$$\begin{aligned} Y_{n+1} \;&=\; Y_n + \frac{h}{12}\left[23Y'_n - 16Y'_{n-1} + 5Y'_{n-2}\right] \\ &\qquad\qquad + \frac{3}{8} h^4 Y^{(4)}(\xi_n) \end{aligned}$$

The third order Adams-Bashforth method is the 3-step method

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

with $y'_j \equiv f(x_j, y_j)$.

USING BACKWARD DIFFERENCES

Using the backword interpolation formula

$$Y'(x) \approx \mathcal{P}_1(x) = Y_n' + \frac{(x - x_n)}{h} \nabla Y_p'$$

on the nodes x_n, x_{n-1}, we have the equivalent quadrature formula

$$\begin{aligned} \int_{x_n}^{x_{n+1}} f(x, Y(x))\, dx &\approx hY_n' + \frac{\nabla Y_n'}{h} \int_{x_n}^{x_{n+1}} (x - x_n) dx \\ &= hY_n' + \frac{h}{2} \nabla Y_n' \end{aligned}$$

This leads to an equivalent formula for the 2-step method,

$$y_{n+1} = y_n + h \left[y_n' + \frac{1}{2} \nabla y_n' \right], \quad n \geq 1$$

with $y_j' \equiv f(x_j, y_j)$.

If we use interpolation of degree p at the nodes x_n, x_{n-1}, ..., x_{n-p}, we obtain the approximation

$$\int_{x_n}^{x_{n+1}} f(x, Y(x))\, dx \approx h \sum_{j=0}^{p} \gamma_j \nabla^j Y_n'$$

with

$$\gamma_0 = 1, \quad \gamma_1 = \frac{1}{2}, \quad \gamma_2 = \frac{5}{12}, \quad \gamma_3 = \frac{3}{8}, \quad \cdots$$

This leads to the Adams-Bashforth method of order $p+1$:

$$y_{n+1} = y_n + h \sum_{j=0}^{p} \gamma_j \nabla^j y_n', \quad n \geq p$$

The truncation error formula is

$$\begin{aligned} T_n(Y) &= \gamma_{p+1} h^{p+2} Y^{(p+2)}(\xi_n) \\ &= \gamma_{p+1} h \nabla^{p+1} Y_n' + O(h^{p+3}) \end{aligned}$$

with ξ_n some point in $[x_{n-p}, x_{n+1}]$. We can estimate the truncation error using

$$T_n(Y) \approx \gamma_{p+1} h \nabla^{p+1} y_n'$$

ADAMS-MOULTON METHODS

The idea behind the *Adam-Moulton methods* is the same as for the Adams-Bashforth methods. The main difference is that we now interpolate $Y'(x) = f(x, Y(x))$ at the $p+1$ node points

$$x_{n+1}, x_n, ..., x_{n-p+1}$$

We can again write the interpolation polynomial $\mathcal{P}_n(x)$ in its Lagrange form or its backward difference formula. The resulting formulas based on the Lagrange form are given on page 388.

The backward difference formula leads to

$$\int_{x_n}^{x_{n+1}} f(x, Y(x))\, dx \approx h \sum_{j=0}^{p} \delta_j \nabla^j Y'_{n+1}$$

with

$$\delta_0 = 1, \quad \delta_1 = -\frac{1}{2}, \quad \delta_2 = -\frac{1}{12}, \quad \delta_3 = -\frac{1}{24}, \quad \cdots$$

The *Adam-Moulton method* of order $p+1$ is given by

$$y_{n+1} = y_n + h \sum_{j=0}^{p} \delta_j \nabla^j y'_{n+1}, \quad n \geq p-1$$

It is a p-step method. Its truncation error is

$$\begin{aligned} T_n(Y) &= \delta_{p+1} h^{p+2} Y^{(p+2)}(\xi_n) \\ &= \delta_{p+1} h \nabla^{p+1} Y'_{n+1} + O(h^{p+3}) \end{aligned}$$

with ξ_n some point in $[x_{n-p+1}, x_{n+1}]$. We can estimate the truncation error using

$$T_n(Y) \approx \delta_{p+1} h \nabla^{p+1} y'_{n+1}$$

EXAMPLES

Backward Euler method: Let $p = 0$.

$$\begin{aligned} y_{n+1} &= y_n + hy'_{n+1} \\ &= y_n + hf(x_{n+1}, y_{n+1}), \quad n \geq 0 \end{aligned}$$

$$T_n(Y) = -\frac{h^2}{2} Y''(\xi_n)$$

Even though this is a very low order formula, it turns out to have desirable properties when used to solve a number of types of problems.

Trapezoidal method: Let $p = 1$. Then

$$\begin{aligned} y_{n+1} &= y_n + hy'_{n+1} - \frac{h}{2} \nabla y'_{n+1} \\ &= y_n + \frac{h}{2} \left[f(x_{n+1}, y_{n+1}) + f(x_n, y_n) \right], \quad n \geq 0 \end{aligned}$$

$$T_n(Y) = -\frac{h^3}{12} Y'''(\xi_n)$$

PREDICTOR-CORRECTOR FORMULAS

Use the $(p+1)$-order Adams-Bashforth formula as a predictor of the solution of the $(p+1)$-order Adams-Moulton method. Thus

$$y_{n+1}^{(0)} = y_n + h \sum_{j=0}^{p} \gamma_j \nabla^j y_n' \tag{7}$$

$$y_{n+1}^{(k+1)} = y_n + h \sum_{j=0}^{p} \delta_j \left(\nabla^j y_{n+1}' \right)^{(k)}, \quad k = 0, 1, \ldots \tag{8}$$

Some people use instead the pair

$$y_{n+1}^{(0)} = y_n + h \sum_{j=0}^{p-1} \gamma_j \nabla^j y_n'$$

$$y_{n+1}^{(k+1)} = y_n + h \sum_{j=0}^{p} \delta_j \left(\nabla^j y_{n+1}' \right)^{(k)}, \quad k = 0, 1, \ldots$$

For (7)-(8) with $p = 0$, this gives the predictor-corrector method

$$y_{n+1}^{(0)} = y_n + hf(x_n, y_n)$$

$$y_{n+1}^{(k+1)} = y_n + hf(x_{n+1}, y_{n+1}^{(k)}), \quad k = 0, 1, ...$$

For the case $p = 1$, this gives the method

$$y_{n+1}^{(0)} = y_n + hy_n' + \frac{h}{2}\nabla y_n'$$

$$\begin{aligned} y_{n+1}^{(k+1)} &= y_n + h\left(y_{n+1}'\right)^{(k)} \\ &\quad -\frac{h}{2}\left(\nabla y_{n+1}'\right)^{(k)}, \quad k = 0, 1, ... \end{aligned}$$

with

$$\left(y_{n+1}'\right)^{(k)} = f(x_{n+1}, y_{n+1}^{(k)})$$

$$\left(\nabla y_{n+1}'\right)^{(k)} = \left(y_{n+1}'\right)^{(k)} - y_n'$$

Note in this last example that

$$y_{n+1} = y_{n+1}^{(0)} + \frac{h}{2}\nabla^2 y'_{n+1} \tag{9}$$

To see this,

$$\begin{aligned} y_{n+1}^{(0)} + \frac{h}{2}\nabla^2 y'_{n+1} &= y_n + \frac{h}{2}\left[3y'_n - y'_{n-1}\right] \\ &\quad + \frac{h}{2}\left[y'_{n+1} - 2y'_n + y'_{n-1}\right] \\ &= y_n + \frac{h}{2}\left[y'_{n+1} + y'_n\right] \end{aligned}$$

Formula (9) is a special case of the general result

$$y_{n+1} = y_{n+1}^{(0)} + \gamma_p h \nabla^{p+1} y'_{n+1}$$

for the predictor and corrector formulas of (7)-(8). This is taken up in problem 23. It provides a simpler approach to doing the iteration process. Note that

$$\begin{aligned}
\left(\nabla^{p+1} y'_{n+1}\right)^{(k)} &= \left(\nabla^{p} y'_{n+1}\right)^{(k)} - \nabla^{p} y'_{n} \\
&\vdots \\
\left(\nabla^{2} y'_{n+1}\right)^{(k)} &= \left(\nabla y'_{n+1}\right)^{(k)} - \nabla y'_{n} \\
\left(\nabla y'_{n+1}\right)^{(k)} &= \left(y'_{n+1}\right)^{(k)} - y'_{n}
\end{aligned}$$

EXAMPLE

Use (7)-(8) with $p = 3$. This yields the predictor-corrector method

$$\begin{aligned} y_{n+1}^{(0)} &= y_n + hy_n' + \tfrac{1}{2}h\nabla y_n' \\ &\quad + \tfrac{5}{12}h\nabla^2 y_n' + \tfrac{3}{8}h\nabla^3 y_n' \\ y_{n+1}^{(k+1)} &= y_n + h\left(y_{n+1}'\right)^{(k)} - \tfrac{1}{2}h\nabla\left(y_{n+1}'\right)^{(k)} \\ &\quad - \tfrac{1}{12}h\nabla^2\left(y_{n+1}'\right)^{(k)} - \tfrac{1}{24}h\nabla^3\left(y_{n+1}'\right)^{(k)} \end{aligned}$$

The predictor formula $y_{n+1}^{(0)}$ uses the values $y_n, \ldots, y_{n-3}$; and the corrector formula uses y_{n+1}, ...,y_{n-2}. To start the method, we require y_0, ...,y_3. To generate y_1, y_2, y_3, a one-step method (usually a Runge-Kutta method) is used. This is inconvenient and expensive, and you want to minimize the number of changes of the stepsize.

The truncation error in the Adams-Moulton formula in this case is

$$\begin{aligned} T_n(Y) &= -\tfrac{19}{720} h \nabla^4 Y'_{n+1} + O(h^6) \\ &= -\tfrac{19}{720} h^5 Y^{(5)}(x_n) + O(h^6) \end{aligned}$$

We can estimate the local error in y_{n+1} using

$$LE \approx -\frac{19}{720} h \nabla^4 y'_{n+1}$$

There are also formulas for LE which are based on some difference of $y_{n+1}^{(0)}$ and y_{n+1}.

AUTOMATIC CODES

Beginning in the late 1960s, multistep codes were developed which varied both the stepsize and the order of the method; and these were all based on the Adams-Bashforth and Adams-Moulton methods. These use the tools as described above.

The first such codes were due to Fred Krogh (at JPL) and Bill Gear (University of Illinois). These have been generalized and extended in a number of ways. The principal such codes today are from groups at Sandia (Albuquerque) and Lawrence Livermore Laboratory, and these are referenced in the text. The Sandia group was directed by Larry Shampine (now at SMU); and the group at LLL is headed by Alan Hindmarsh.

The first generation Sandia code was called DE/STEP; and a more current version is called DDEABM. The latter is given in the class account, together with a driver program. The program contains its own documentation in its leading statements.

These codes attempt to make the local error LE satisfy

$$\left|(LE)_j\right| \leq ABSERR + RELERR * \left|y_{n,j}\right|$$

for each component $y_{n,j}$ of the numerical solution y_n. The codes give solutions at user supplied points, while generating their own node points internally. There are many options available. However, calculating an estimate of the global error is not an option.