Computer Science/Mathematics 455
Lecture Notes
Lecture # 33

Embedded Runge–Kutta Methods

Again we consider the initial value problem (IVP)

$$\begin{aligned} \mathbf{x}'(t) &= \mathbf{f}(t, \mathbf{x}(t)) \\ \mathbf{x}(t_0) &= \mathbf{x}_0 \end{aligned}$$

where t is usually time, and $\mathbf{x}(t)$ is a vector valued function.

In our notation

$$\mathbf{x}_n \approx \mathbf{x}(t_n).$$

Last time, we discussed the Runge–Kutta–Fehlberg 45 method.

It produced a first approximation of the form.

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}(t_n, \mathbf{x}_n) \\ \mathbf{k}_2 &= \mathbf{f}(t_n + c_1 h, \mathbf{x}_n + h a_{21} \mathbf{k}_1) \\ \mathbf{k}_3 &= \mathbf{f}(t_n + c_3 h, \mathbf{x}_n + h(a_{31}\mathbf{k}_1 + a_{32}\mathbf{k}_2)) \\ \mathbf{k}_4 &= \mathbf{f}(t_n + c_4 h, \mathbf{x}_n + h(a_{41}\mathbf{k}_1 + a_{42}\mathbf{k}_2 + a_{43}\mathbf{k}_3)) \\ \mathbf{k}_5 &= \mathbf{f}(t_n + c_5 h, \mathbf{x}_n + h(a_{51}\mathbf{k}_1 + a_{52}\mathbf{k}_2 + a_{53}\mathbf{k}_3 + a_{54}\mathbf{k}_4)) \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + h \sum_{i=1}^{5} b_i \mathbf{k}_i \end{aligned}$$

It has the Butcher array (unfilled spaces are zeros)

$$\begin{array}{c|ccccc} 0 & & & & & \\ 1/4 & 1/4 & & & & \\ 3/8 & 3/32 & 9/32 & & & \\ 12/13 & 1932/2197 & -7200/2197 & 7296/2197 & & \\ 1 & 439/216 & -8 & 3680/513 & -845/4104 & \\ \hline & 25/216 & 0 & 1408/2565 & 2197/4104 & -1/5 \end{array} \tag{1}$$

We then added one more stage

$$\mathbf{k}_6 = \mathbf{f}(t_n + h/2, \mathbf{x}_n + h\sum_{i=1}^{5} a_{6j}\mathbf{k}_i).$$

where $a_{6j}, j = 1, 2, 3, 4, 5$ are given by

$$-8/27, 2, -3544/2565, 1859/4104, -11/40,$$

to get a fifth order method. It has the form

$$\hat{\mathbf{x}}_{n+1} = \mathbf{x}_n + h\sum_{i=1}^{6} \hat{b}_i\mathbf{k}_i.$$

where $\hat{b}_i, i = 1, 2, 3, 4, 5, 6$ is given by

$$16/135, 0, 6656/12825, 28561/56430, -9/50, 2/55.$$

To model local truncation error, let

$$\begin{aligned} \mathbf{z}_n'(t) &= \mathbf{f}(t, z_n(t)) \\ \mathbf{z}_n(t_n) &= \mathbf{x}_n \end{aligned}$$

where we start solving the IVP at the most recent approximation. Then we showed last time that

$$\|\hat{\mathbf{x}}_{n+1} - \mathbf{x}_{n+1}\| = \|\mathbf{z}_n(t_{n+1}) - \mathbf{x}_{n+1}\| + O(h^6) = \alpha_5 h^5 + O(h^6).$$

We have an estimate of the highest order term of the local error. The only extra effort is that of computing an extra stage.

We represent this method in the modified Butcher array like so.

$$\begin{array}{c|cccccc}
0 & & & & & & \\
1/4 & 1/4 & & & & & \\
3/8 & 3/32 & 9/32 & & & & \\
12/13 & 1932/2197 & -7200/2197 & 7296/2197 & & & \\
1 & 439/216 & -8 & 3680/513 & -845/4104 & & \\
1/2 & -8/27 & 2 & -3544/2565 & 1859/4104 & -11/40 & \\
\hline
 & 25/216 & 0 & 1408/2565 & 2197/4104 & -1/5 & \\
 & 16/135 & 0 & 6656/12825 & 28561/56430 & -9/50 & 2/55
\end{array} \quad (2)$$

This kind of Runge–Kutta method is probably the most commonly coded.

The main competitors to the Runge–Kutta methods are the Adams methods.

Adams Methods

Two types.

Explicit methods – All of our methods so far have been explicit.

Implicit methods – require the solution of a nonlinear equation to get to the next step.

Notation.

$$\begin{aligned}
\mathbf{x}_n &\approx \mathbf{x}(t_n), \\
\mathbf{f}_n &= \mathbf{f}(t_n, \mathbf{x}_n) \approx \mathbf{f}(t_n, \mathbf{x}(t_n)) = \mathbf{x}'(t_n)
\end{aligned}$$

A N-step Adams method has the form

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h \sum_{k=0}^{N} \beta_k \mathbf{f}_{n+1-k}$$

where

$$\begin{aligned}
\beta_0 = 0 &\quad \text{explicit} \\
\beta_0 \neq 0 &\quad \text{implicit}
\end{aligned}$$

To start, we develop a two step implicit method from Taylor series for a scalar equation.

$$x(t_{n+1}) = x(t_n) + hx'(t_n) + \frac{h^2}{2}x''(t_n) + \frac{h^3}{6}x'''(t_n) + O(h^4)$$

which we write

$$x_{n+1} = x_n + hx'_n + \frac{h^2}{2}x''_n + \frac{h^3}{6}x'''_n + O(h^4).$$

Using finite difference approximations,

$$\begin{aligned}
x'_n &= f_n = f(t_n, x_n) \\
x''_n &\approx \frac{x'_{n+1} - x'_{n-1}}{2h} = \frac{f_{n+1} - f_{n-1}}{2h} \\
x'''_n &\approx \frac{x'_{n+1} - 2x'_n + x'_{n-1}}{h^2} = \frac{f_{n+1} - 2f_n + f_{n-1}}{h^2}
\end{aligned}$$

Thus

$$\begin{aligned} x_{n+1} &= x_n + hf_n + \frac{h^2}{2}\frac{f_{n+1}-f_{n-1}}{2h} + \frac{h^3}{6}\frac{f_{n+1}-2f_n+f_{n-1}}{h^2} \\ &= x_n + \frac{h}{12}(5f_{n+1}+8f_n-f_{n-1}). \end{aligned}$$

This is the two-step, third order Adams-Moulton method. It is implicit and given by

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{12}[5\mathbf{f}(t_{n+1}, \mathbf{x}_{n+1}) + 8\mathbf{f}_n - \mathbf{f}_{n-1}].$$

We have to solve a nonlinear equation to get $\mathbf{x}_{n+1}$, we could use Newton's method.

Onw justification for the Adams methods is numerical integration.

Consider

$$x(t_{n+1}) - x(t_n) = \int_{t_n}^{t_{n+1}} x'(t)dt = \int_{t_n}^{t_{n+1}} f(t, x(t))dt.$$

Forget the dependence upon x and consider

$$x_{n+1} - x_n = \int_{t_n}^{t_{n+1}} f(t)dt.$$

Using the change of variable,

$$t = t_n + sh, \quad 0 \le s \le 1$$

we have

$$x_{n+1} - x_n = \int_{t_n}^{t_{n+1}} f(t)dt = h\int_0^1 f(t_n + hs)ds = h\int_0^1 F(s)ds.$$

We make the last integral exact for polynomials up to a certain degree.

Adams–Bashforth, Explicit (3–step)

$$\int_0^1 F(t)dt = \beta_1 F(0) + \beta_2 F(-1) + \beta_3 F(-2)$$

If we make this exact matching for $F(t) = 1, t, t^2$ then we have the Vandermonde matrix

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 1 & 4 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1/2 \\ 1/3 \end{pmatrix}.$$

The solution is

$$\beta_1 = 23/12, \quad \beta_2 = -4/3, \quad \beta_3 = 5/12,$$

so that we have

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{12}(23\mathbf{f}_n - 16\mathbf{f}_{n-1} + 5\mathbf{f}_{n-2}).$$

This is a third order explicit method. Unstable if h is even of moderate size.

To get the implicit Adams–Moulton fourth order method, we fit

$$\int_0^1 F(t)dt = \beta_0 F(1) + \beta_1 F(0) + \beta_2 F(-1) + \beta_3 F(-2)$$

and make it exact matching for $F(t) = 1, t, t^2, t^3$. It is one of your homework problems to solve for this, so I will not. The resulting method is

$$\begin{aligned}
\mathbf{x}_{n+1} &= \mathbf{x}_n + \frac{h}{24}(9\mathbf{f}_{n+1} + 19\mathbf{f}_n - 5\mathbf{f}_{n-1} + \mathbf{f}_{n-2}) \\
&= \mathbf{x}_n + \frac{h}{24}(9\mathbf{f}(t_{n+1}, \mathbf{x}_{n+1}) + 19\mathbf{f}_n - 5\mathbf{f}_{n-1} + \mathbf{f}_{n-2})
\end{aligned}$$

This method is much more stable, but requires the solution of a nonlinear equation. We could use Newton's method, but that would require the evaluation of a Jacobian (yecch!).

The better idea is to use the predictor–corrector method.

I will start with the Adams predictor–corrector method of order 1. The Adams–Bashforth method of order 1 is

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}_n, \qquad \text{Euler's method}$$

The Adams–Moulton method of order 1 is

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}_{n+1} = \mathbf{x}_n + h\mathbf{f}(t_{n+1}, \mathbf{x}_{n+1}).$$

which is the backward Euler method. This requires us to solve a nonlinear equation to do a step. To solve it approximately, we use Euler's method to get an initial guess (called a predictor step).

$$\mathbf{x}_{n+1}^{(0)} = \mathbf{x}_n + h\mathbf{f}_n.$$

and then perform the fixed point iteration

$$\mathbf{x}_{n+1}^{(k+1)} = \mathbf{x}_n + h\mathbf{f}(t_{n+1}, \mathbf{x}_{n+1}^{(k)})$$

for $k = 0, 1, \ldots, \ell-$ until we obtain a satisfactory solution. We would prefer to do this only one or two times. We then let

$$\mathbf{f}_{n+1} = \mathbf{f}(t_{n+1}, \mathbf{x}_{n+1}^{(\ell)})$$

be the value used in subsequent steps.

Next time, we discuss predictor–corrector methods in more detail.