Lecture 3

Zed

March 9, 2017

1 General Formulation of Explicit 1-Step Method

We now write 1-step methods in a general form,

$$y_{n+1} = y_n + h\Phi(t_n, y_n, h) \quad (\dagger)$$

Def. **Truncation Error**: We insert true values of the solution into the formulation above, and define the truncation error:

$$T_n = \frac{y(t_{n+1}) - y(t_n)}{h} - \Phi(t_n, y(t_n), h)$$

Def. **Consistency**: The numerical method (†) is *consistent* with $y' = f(t, y)$ if the truncation error goes to 0 as $h \to 0$, i.e. $\lim_{h\to 0} T_n = 0$.

Cor. Let $\Phi(t_n, y(t_n), \cdot)$ continuous on h, then the definition of consistency is equivalent to

$$\lim_{h\to 0} \frac{y(t_n + h) - y(t_n)}{h} = \lim_{h\to 0} \Phi(t_n, y(t_n), h)$$

i.e. $y'(t_n) = \Phi(t_n, y(t_n), 0)$. Hence we have consistency $\iff$

$$y'(t_n) = \Phi(t_n, y(t_n), 0) \iff f(t_n, y(t_n)) = \Phi(t_n, y(t_n), 0)$$

Thm. (**Convergence and Zero-Stability**) If the following conditions holds

1. $f(t, y)$ is Lipschitz in y, continuous in t.
2. $\Phi(t, y, h)$ is continuous in t, Lipschitz in y in the same region as for the Lipschitz condition of $f(t, y)$, such that

$$|\Phi(x, y, h) - \Phi(x, z, h)| \leq L_\Phi |y - z|$$

3. $\Phi(\cdot, \cdot, h)$ is uniformly continuous in h, $\forall\ h \in [0, h_0]$.

Then the numerical solution is zero-stable (zero-stability), and it converges to the real solution. (convergence)

Proof. We still use the similar trick as in the euler method's proof. I.e. write the numerical method in discrete terms and also in the exact values

$$y_{n+1} = y_n + h\Phi(t_n, y_n, h) \quad (*)$$

$$y(t_{n+1}) = y(t_n) + h\Phi(t_n, y(t_n), h) + hT_n \quad (**)$$

the second equation minus the first one, and let $e_n := y_n - y(t_n)$, T_n be the truncation error

$$\begin{aligned} e_{n+1} &= e_n + h\left[\Phi(t_n, y_n, h) - \Phi(t_n, y(t_n), h)\right] - hT_n \\ |e_{n+1}| &\le |e_n| + hL_\Phi|e_n| + hT_n \\ &\le (1 + hL_\Phi)|e_n| + hT \quad (\text{with } T = \max_n T_n) \end{aligned} \tag{1}$$

Solve this recursive equation, and let $e_0 := y_0 - y(t_0)$, the (round off) error of the initial data, we get

$$\begin{aligned} |e_n| &\le (1 + hL_\Phi)^n|e_0| + [(1 + hL_\Phi)^n - 1]\frac{T}{L_\Phi} \\ &\le \exp(nhL_\Phi)|e_0| + (\exp(nhL_\Phi) - 1)\frac{T}{L_\Phi} \end{aligned} \tag{2}$$

We have convergence: since the first term only concerns the error of initial data, which we can, using some technique, make it close to zero. The second term is constant times T, which is at least $O(h) \to 0$ as $h \to 0$. So the error itself is always bounded.
The zero stability goes in a similar fashion. Recall the definition, we have two initial data y_0 and $\tilde{y}_0$, and using the same numerical method, we obtain sequence $\{y_0, y_1, ..., y_n\}$ and $\{\tilde{y}_0, \tilde{y}_1, ..., \tilde{y}_n\}$. Substitute $y(t_n)$ in $(**)$ to $\tilde{y}_n$, we obtain

$$|y_n - \tilde{y}_n| \le \exp(nhL_\Phi)|y_0 - \tilde{y}_0| + O(h)$$

which matches the definition of zero-stability ($|y_n - \tilde{y}_n| \le c \max_{0\le j\le s-1} |y_j - \tilde{y}_j|$ as $h \searrow 0$.)

2 Construction of Φ

2.1 The Taylor Expansion

One way to construct a higher-ordered method is using Taylor expansion of $y(t_{n+1})$ at $y(t_n)$:

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

The complexity lies in calculating the derivatives. We do some low ordered terms:

$$\begin{aligned} &y'(t_n) = f(t_n, y(t_n)) \\ &y''(t_n) = f_t + f_y y'(t_n) = f_t + f_y f \\ &y'''(t_n) = f_{tt} + f_{ty}y' + y'f_{yt} + y'^2 f_{yy} + y''f_y = f_{tt} + 2f_{ty}f + f^2 f_{yy} + f_y f_t + f_y^2 f \end{aligned} \tag{3}$$

The number of terms grows exponentially, very hard to calculate...

2.2 Runge-Kutta Construction

We've got another idea, we want to approximate

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t))dt$$

with

$$y(t_{n+1}) = y(t_n) + h\sum_{j=1}^{N} b_j f(t_n + c_j h, y(t_n + c_j h))$$

where we make a finer partition of $[t_n, t_{n+1}]$ to N *stages*: $t_n + c_j h$, $c_1 \le c_2 \le ... \le c_N$. And b_j being the weights assigned to each value.

Def. **N-Stages Runge Kutta Method**: we partition the interval $[t_n, t_{n+1}]$ to N stages, and update y by

$$y_{n+1} = y_n + h\sum_{j=1}^{N} b_j k_j$$

where

$$k_1 = f(t_n, y_n); \quad k_i = f(t_n + c_i h, y_n + h\sum_{j=1}^{i-1} a_{ij}k_j) \;\; 2 \le i \le N$$

There are three sets of parameters, $\boldsymbol{b}^\top$: the weights, $\boldsymbol{c}$: the nodes of partition of $[t_n, t_{n+1}]$ and $\boldsymbol{A} = \{a_{ij}\}$: the *Runge-Kutta matrix*. We can put then together in a $n+1 \times n+1$ table, which is often referred to as *butcher table*.

Prop. By the corollary listed in the previous section, consistency of 1-step method $\iff$ $\Phi(t_n, y(t_n), 0) = f(t_n, y(t_n))$. When $h = 0$, all the ks degenerate to $f(t_n, y(t_n))$. And Runge-Kutta has $\Phi(t_n, y(t_n), h) = \sum_{j=1}^{N} b_j k_j \Rightarrow \sum_{j=1}^{N} b_j f(t_n, y(t_n)) = f(t_n, y(t_n)) \Rightarrow \sum_{j=1}^{N} b_j = 1$. Hence the consistency requires all the weights sum to 1.

Ex. **Runge (1895)**:

$$\begin{array}{c|cccc} c_1 & & & & \\ c_2 & a_{21} & & & \\ c_3 & a_{31} & a_{32} & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & a_{s2} & \dots & a_{s,s-1} \\ \hline & b_1 & b_2 & \dots & b_{s-1} \quad b_s \end{array} = \begin{array}{c|c} \boldsymbol{c} & \boldsymbol{A} \\ \hline & \boldsymbol{b}^\top \end{array} = \begin{array}{c|cc} 0 & & \\ 1/2 & 1/2 & \\ \hline & 0 & 1 \end{array}$$

Gives

$$y_{n+1} = y_n + hk_2; \quad k_2 = f\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}f(t_n, y_n)\right)$$

Ex.

$$\begin{array}{c|c} \boldsymbol{c} & \boldsymbol{A} \\ \hline & \boldsymbol{b}^\top \end{array} = \begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & 1/2 & 1/2 \end{array}$$

Gives

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{2}k_1 + \frac{h}{2}k_2 \\ k_1 &= f(t_n, y_n), \\ k_2 &= f\left(t_n + h, y_n + f(t_n, y_n)\right) \end{aligned} \tag{4}$$

3 RK Methods of Different Orders

We now consider how to solve the coefficients

- $N = 1$ case is just $y_{n+1} = y_n + hf(t_n, y_n)$, which is the Euler method (order 1).
- $N = 2$:

$$\begin{aligned} y_{n+1} &= y_n + h(b_1 k_1 + b_2 k_2) \\ k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + c_2 h, y_n + a_{21} h k_1) \end{aligned} \tag{5}$$

substitute y_n with $y(t_n)$, and compute with taylor's expansion of two variables (expand $f(t_n + c_2h, y_n + a_{21}hf(t_n, y_n))$ at $(t_n, y(t_n))$)

$$\begin{aligned} y(t_{n+1}) &= y(t_n) + hb_1 f(t_n, y(t_n)) + hb_2 f(t_n + c_2h, y_n + a_{21}hf(t_n, y_n)) \\ &= y(t_n) + hb_1 y'(t_n) + hb_2[f(t_n, y_n) + f_t c_2 h + f_y a_{21} hf + \\ &\quad \tfrac{1}{2}(f_{tt}c_2^2h^2 + 2f_{ty}c_2 h a_{21} hf + f_{yy}h^2 a_{21}^2 f^2)] + O(h^3) \\ &= y(t_n) + h(b_1 + b_2)f + h^2(b_2 c_2 f_t + b_2 a_{21} f_y f) + \\ &\quad \tfrac{1}{2}h^3 b_2(f_{tt}c_2^2 + 2f_{ty}f c_2 a_{21} + f_{yy}f^2 a_{21}^2) + O(h^4) \end{aligned} \tag{6}$$

Compare this with the Taylor expansion that we have calculated in section 2.1:

$$y(t_{n+1}) = y(t_n) + hf + \tfrac{h^2}{2}(f_t + f_y f) + \tfrac{h^3}{6}(f_{tt} + 2f_{ty}f + f^2 f_{yy} + f_y f_t + f_y^2 f) + O(h^4) \tag{7}$$

Hence we match the coefficients to make the method a higher order:

- h vanishes: $b_1 + b_2 = 1$.
- h^2 vanishes: $b_2 c_2 = \frac{1}{2}$, $b_2 a_{21} = \frac{1}{2}$.
- h^3 cannot vanish, because the expansion of RK2 misses the terms $f_y f_t, f_y^2 f$.

I.e. it suffices to solve algebraic system $b_1 + b_2 = 1, b_2 c_2 = \frac{1}{2}, b_2 a_{21} = \frac{1}{2}$ to obtain a Runge-Kutta method of order 2.

- $N = 3$: is too complicated, oftentimes the calculation is left to symbolic computing programs. We give the results (**Henn, 1900**):

$$\begin{array}{c|ccc} c_1 & & & \\ c_2 & a_{21} & & \\ c_3 & a_{31} & a_{32} & \\ \hline & b_1 & b_2 & b_3 \end{array} = \begin{array}{c|ccc} 0 & & & \\ 1/3 & 1/3 & & \\ 1/3 & 0 & 2/3 & \\ \hline & 1/4 & 0 & 3/4 \end{array}$$

And another solution:

$$\begin{array}{c|ccc} c_1 & & & \\ c_2 & a_{21} & & \\ c_3 & a_{31} & a_{32} & \\ \hline & b_1 & b_2 & b_3 \end{array} = \begin{array}{c|ccc} 0 & & & \\ 1/2 & 1/2 & & \\ 1 & -1 & 2 & \\ \hline & 1/6 & 2/3 & -1/6 \end{array}$$

Note that we consider (**Henn**) better than the second recipie, because it has more zeros, which saves the computation time.

- $N = 4$: is the classical *RK4* routine impliemented in many computing libraries. It is due to (**Runge-Konniz, 1924**)

$$\begin{array}{c|cccc} c_1 & & & & \\ c_2 & a_{21} & & & \\ c_3 & a_{31} & a_{32} & & \\ c_4 & a_{41} & a_{42} & a_{43} & \\ \hline & b_1 & b_2 & b_3 & b_4 \end{array} = \begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

If $\boldsymbol{A}$ is a lowertriangular matrix, then the corresponding RK method is an explicit method. Note that solving the algebraic system of parameters , with as many zeros as possible, is not a trivial task. Nevertheless, we have some methods to the order like 12 (RK14) nowadays.