Computational methods-Lecture 13

Runga-Kutta methods and the convergence of one step method

1 Local Truncation error and the order

The local truncation error describes how well the **exact solution** satisfies the numerical scheme.

The consistency is measured by the local truncation error (LTE) where u^n is replaced by $u(t_n)$. For a method

$$u_{n+1} = u_n + h\phi(t_n, u_0, u_1, \cdots, u_m, h),$$

the local truncation error (LTE) is defined by

$$\tau_n = \frac{1}{h}\Big[u(t_{n+1}) - \Big(u(t_n) + h\phi(t_n, u(t_0), u(t_1), \cdots, u(t_m), h)\Big)\Big]$$

Note that we have divided h here because $\frac{u^{n+1}-u^n}{h}$ is in the same order as the derivative.

Remark 1. *This is different from the definiton in the Chinese reference book, which is given by*

$$T_{n+1} = u(t_{n+1}) - u(t_n) - h\phi(t_n, u(t_0), u(t_1), \cdots, u(t_m), h).$$

This error is often called the "one-step error" instead of "local truncation error" in some literature.

Example: For the forward Euler (FE):

$$\begin{aligned}\tau_n &= \frac{1}{h}(u(t_{n+1}) - u(t_n) - hf(t_n, u(t_n)))\\ &= \frac{1}{h}(u(t_{n+1}) - u(t_n) - ku'(t_n)) = O(h).\end{aligned}$$

Definition 1. *The ODE solvers are said to be consistent if the local truncation error goes to zero as $h \to 0$.*

Definition 2. *If there exists a largest number $p > 0$ such that*

$$\tau_n = O(h^p),$$

then the method is said to be of order p.

Direct Taylor expansion shows that the two Euler methods are first order while the trapezoidal method is a second order method.

****

(This part is not required in exam)

To improve the order, one can consider the higher Taylor schemes as follows.

2 Runga Kutta method (Not required for exam)

The Runga-Kutta methods are one-step methods, but with multi-stage to improve the accuracy, but possibly avoid computing high order derivatives. (Recall that the higher order Taylor schemes involve many derivatives.)

The starting point is again the integral relation:

$$u(t_{n+1}) = u(t_n) + \int_{t_n}^{t_{n+1}} f(s, u(s))\, ds.$$

The idea is to use many nodes to approximate the integral.

$$\int_{t_n}^{t_{n+1}} f(s, u(s))\, ds \approx h \sum_{i=1}^{r} c_i f(t_n + \lambda_i h, u(t_n + \lambda_i h)).$$

Hence, a possible way is to do the following:

$$u_{n+1} = u_n + h \sum_{i=1}^{r} c_i f(t_n + \lambda_i h, U_i) =: u_n + h \sum_{i=1}^{r} c_i K_i,$$

where U_j is approximation of $u(t_n + \lambda_j h)$, which can be found by setting

$$U_i = u_n + h \sum_{j=1}^{r} \mu_{ij} f(t_n + \lambda_j h, U_j), i = 1, 2, \ldots, r$$

Since U_i is an approximation of the value at $t_n + \lambda_i h$, one should have

$$\sum_{j=1}^{r} \mu_{ij} = \lambda_i.$$

Moreover, for the consistency, we must have

$$\sum_{i=1}^{r} c_i = 1.$$

This class of methods are called the r**-stage Runga-Kutta** method. If we have

$$\mu_{ij} = 0, \quad j \geq i,$$

then the method is explicit. Otherwise, we have *implicit Runga-Kutta* methods. The most frequently used schemes are $RK2, RK3, RK4$. In general the Runga-Kutta methods are not unique. **Often, we require RK-r methods to have order** r**.**

In general, to determine the coefficients, you need to do a lot of tedious Taylor expansions.

Here, we explore a quick way for you to choose the coefficients μ_{ij} and c_i. We apply the method to the model problem with $f(t, u) = \sigma u$. Then,

$$U_i = u^n + h \sum_{j=1}^{r} \mu_{ij} \sigma U_j, \qquad u_{n+1} = u_n + h \sum_{i=1}^{r} c_i \sigma U_i$$

However, we know that $u(t_{n+1}) = e^{\sigma h} u(t_n)$. Note

$$e^{\sigma h} = \sum_{n \geq 0} \frac{(\sigma h)^n}{n!}$$

We can therefore solve Y_j out in the first equation and determine the coefficients in

$$\sum_{n \geq 0} \frac{(\sigma h)^n}{n!} u_n \approx u^n + h \sum_{i=1}^{r} c_i \sigma U_i,$$

by comparing the powers of h.

2.1 Derivation of Runga-Kutta 2 methods

Let us consider the explicit RK-2 methods.

$$u_{n+1} = u_n + h \sum_{i=1}^{2} c_i f(t_n + \lambda_i h, U_i),$$

where

$$U_i = u_n + h \sum_{j=1}^{i-1} \mu_{ij} f(t_n + \lambda_j h, U_j), i = 1, 2.$$

Hence, for explicit method, we must have

$$U_1 = u_n, \quad \lambda_1 = 0.$$

Then,

$$u_{n+1} = u_n + h(c_1 f(t_n, u_n) + c_2 f(t_n + \lambda_2 h, U_2))$$

and

$$U_2 = u_n + h\mu_{21} f(t_n + \lambda_1 h, U_1).$$

By the consistency conditions

$$c_1 + c_2 = 1, \quad \mu_{21} = \lambda_2.$$

Now, we set

$$f(t, u) = \sigma u.$$

Then,

$$U_2 = u_n + h\mu_{21}\sigma U_1 = u_n(1 + \sigma h \mu_{21})$$

Hence,

$$u_{n+1} = u_n + c_1 h\sigma u_n + c_2 h\sigma u_n(1 + \sigma h \mu_{21}).$$

To compare with

$$u_n e^{\sigma h},$$

we ask

$$c_1 + c_2 = 1, \quad c_2 \mu_{21} = \frac{1}{2}.$$

The midpoint method

Let us choose $\lambda_2 = \mu_{21} = \frac{1}{2}$. Then, $c_2 = 1$ and $c_1 = 0$. Then, method is then given by

$$U_2 = u_n + \frac{h}{2} f(t_n, u_n), \quad u_{n+1} = u_n + hf(t_n + \frac{h}{2}, U_2).$$

The improved Euler method

If we choose $\mu_{21} = \lambda_2 = 1$, then $c_2 = \frac{1}{2} = c_1$.
Hence,

$$U_2 = u_n + hf(t_n, u_n), \quad u_{n+1} = u_n + \frac{h}{2}(f(t_n, u_n) + f(t_n + h, U_2)).$$

Clearly, this can be viewed as the modification of trapezoidal method where the value at t_{n+1} is obtained by the forward Euler's method. This is called the **improved Euler's method**. This is a **predictor-corrector** method.

2.2 A RK4 method

The following Runga-Kutta 4 method is also used frequently in practice.

$$u_{n+1} = u_n + \frac{h}{6}[K_1 + 2K_2 + 2K_3 + K_4],$$

where

$$K_1 = f(t_n, u_n), \quad K_2 = f(t_n + \frac{h}{2}, u_n + \frac{h}{2}K_1),$$
$$K_3 = f(t_n + \frac{h}{2}, u_n + \frac{h}{2}K_2), \quad K_4 = f(t_n + h, u_n + hK_3).$$

In other words, $U_1 = u_n, U_2 = u_n + \frac{h}{2}K_1, \;\; U_3 = u_n + \frac{h}{2}K_2, \;\; U_4 = u_n + hK_3$.

3 The convergence of one step method

An ODE solver is convergent if for a problem $u' = f(t,u)$ where f is continuous and Lipschitz continuous in u on $[0,T]$, we have

$$\lim_{k\to 0, nk=T} |u^n - u(T)| = 0,$$

where T is in the largest interval of existence.

f is Lipschitz in u means

$$\sup_{0\le t\le T} |f(t,u_1) - f(t,u_2)| \le L(T)|u_1 - u_2|$$

Claim:

For one step solvers, $u_{n+1} = u_n + k\Psi(u_n, t_n, h)$, as long as Ψ is continuous and uniformly Lipschitz continuous in u, the solver is stable. Here, 'stable' means that the global error introduced by the m-th step error will not be amplified too much. If further it is consistent, then it is convergent.

Proof. Let τ_j be the local truncation error. Then, the one step error is

$$u(t_{j+1}) - u(t_j) - h\Psi(u(t_j), t_j, h) = h\tau_j.$$

Let $E^j = |u(t_j) - u_j|$. Then, we have

$$E^{j+1} \le E^j + h|\Psi(u_j, t_j, h) - f(u(t_j), t_j, h)| + h|\tau_j| \le E^j + hLE^j + Ch^2.$$

Then,

$$E^n \le E^0 e^{nkL} + \sum_{j=0}^{n-1} h|\tau_{n-1-j}|(1+hL)^j \le Ce^{TL}h$$

□

In this proof, we have implicitly used the so-called *stability*. The error term $h\tau_j$ is amplified by $(1+hL)^{n-j}$ which is uniformly bounded by e^{TL}, so we have stability.

Example We take the forward Euler as the example. Again f is assumed to be Lipschitz.