Numerical Analysis

Iydon

2018 年 10 月 23 日

目录

1 The Elementary Theory of Initial-Value Problems

定义 1.1 (Lipschitz Condition) *A function $f(t, y)$ is said to satisfy a* Lispschitz condition *in the variable Y on a set $D \subset \mathbb{R}^2$ if a constant $L > 0$ exists with*

$$|f(t, y_1) - f(t, y_2)| \leq L\,|y_1 - y_2|$$

whenever (t_1, y_1), (t_2, y_2) are in D. The constant L is called a Lipschitz constant *for f.*

定义 1.2 (Convex) *A set $D \subset \mathbb{R}^2$ is said to be convex if whenever $(t_1, y_1), (t_2, y_2) \in D$, then for every $\lambda \in [0, 1]$,*

$$((1-\lambda)t_1 + \lambda t_2, (1-\lambda)y_1 + \lambda y_2) \in D.$$

定理 1.1 *Suppose $f(t, y)$ is defined on a convex set $D \subset \mathbb{R}^2$, if a constant $L > 0$ exists with*

$$\left|\frac{\partial f}{\partial y}(t, y)\right| \leq L$$

for all $(t, y) \in D$, then f satisfies a Lipschitz condition on D in the variable y with Lipschitz constant L.

定理 1.2 *Suppose that $D = \{(t, y) | a \leq t \leq b, y \in \mathbb{R}\}$ and $f(t, y)$ is continuous on D. If f satisfies a Lipschitz condition on D in the variable y, then the initial-value problem*

$$\begin{cases} y'(t) = f(t, y) & a \leq t \leq b \\ y(a) = \alpha \end{cases}$$

has a unique solution $y(t)$ for $a \leq t \leq b$.

1.1 Well-Posed Problems

定理 1.3 (Well-Posed) *Suppose that $D = \{(t, y) | a \leq t \leq b, y \in \mathbb{R}\}$ and $f(t, y)$, if f is continuous and satisfies a Lipschitz condition in the variable y on the set D, then the initial-value problem*

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha$$

is well-posed.

2 Euler's Method

The object of *Euler's method* is to obtain approximations to the well-posed initial-value problem

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha$$

We will use Taylor's Theorem to derive Euler's method. Suppose that $y(t)$, the unique solution has two continuous derivations on $[a, b]$, so that for each $i = 0, 1, \cdots, N-1$

$$y(t_{i+1}) = y(t_i) + (t_{i+1} - t_i)y'(t_i) + \frac{(t_{i+1} - t_i)^2}{2} y''(\xi_i)$$

for some number ξ_i in t_i, t_{i+1}. Because $h = t_{i+1} - t_i$, we have

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h)^2}{2} y''(\xi_i)$$

Euler's method constructs $\omega_i \approx y(t_i)$, for each $i = 1, 2, \cdots, N$, by deleting the remainder term, then Euler's method is

$$\begin{cases} \omega_0 = \alpha \\ \omega_{i+1} = \omega_i + hf(t_i, \omega_i) \quad fori = 0, 1, \cdots, N-1 \end{cases}$$

2.1 Errors Bounds for Euler's Method

定理 2.1 *Suppose f is continuous and satisfies a Lipschitz condition with constant L on*

$$D = \{(t,y) | a \le t \le b, y \in \mathbb{R}\}$$

and that a constant M exists with

$$|y''(t)| \le M, \quad \text{for all } t \in [a,b]$$

where $y(t)$ denotes the unique solution to the initial-value problem

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t,y), \quad a \le t \le b, \quad y(a) = \alpha$$

Let $\omega_0, \cdots, \omega_N$ be the approximations generated by Euler's method for some positive integer N, then for each $i = 0, 1, \cdots, N$

$$|y(t_i) - \omega_i| \le \frac{hM}{2L}\left[e^{L(t_i - a)} - 1\right].$$

[Proof]

$$\begin{aligned}
|y_{i+1} - \omega_{i+1}| &\le |y_i - \omega_i| + h\,|f(t_i, y_i) - f(t_i, \omega_i)| + \frac{h^2}{2}\,|y''(\xi_i)| \\
&\le (1 + hL)\,|y_i - \omega_i| + \frac{h^2 M}{2} \\
&\le e^{(i+1)hL}(|y_0 - \omega_0| + \frac{h^2 M}{2hL}) - \frac{h^2 M}{2hL} \\
&= \frac{hM}{2L}\left(e^{(t_{i+1} - a)L} - 1\right).
\end{aligned}$$

□

定理 2.2 *If $u_0, u_1, \cdots, u_N$ be the approximations and $|\delta_i| < \delta$, then*

$$|y(t_i) - u_i| \le \frac{1}{L}\left(\frac{hM}{2} + \frac{\delta}{h}\right)\left[e^{L(t_i - a)} - 1\right] + \delta e^{L(t_i - a)}$$

The minimal value of $E(f)$ occurs when $h = \sqrt{\frac{2\delta}{M}}$

3 Higher-Order Taylor Method

定义 3.1 (Local Truncation Error) *The difference method*

$$\begin{cases} \omega_0 = \alpha \\ \omega_{i+1} = \omega_i + h\phi(t_i + \omega_i) \quad \textit{for each } i = 0, 1, \cdots, N-1 \end{cases}$$

has local truncation error

$$\tau_{i+1}(x) = \frac{y_{i+1} - (y_i + h\phi(t_i + y_i))}{h} = \frac{y_{i+1} - y_i}{h} - \phi(t_i, y_i)$$

for each $i = 0, 1, \cdots, N-1$ *where* y_i *and* y_{i+1} *denote the accuracy at a specific step, assuming that the method was exact at the previous step.*

Euler's method has $\tau_{i+1} = \frac{h}{2}y''(\xi_i)$, so the local truncation error in Euler's method is $O(h)$.

3.1 Taylor Method of Order n

$$\begin{cases} \omega_0 = \alpha \\ \omega_{i+1} = \omega_i + h\phi(t_i + \omega_i) \quad \text{for each } i = 0, 1, \cdots, N-1 \end{cases}$$

where $T^{(n)}(t_i, \omega_i) = f(t_i, \omega_i) + \frac{h}{2}f'(t_i, \omega_i) + \cdots + \frac{h^{n-1}}{n!}f^{(n-1)}(t_i, \omega_i)$.

定理 3.1 *If Taylor's method of order n is used to approximate the solution to*

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha$$

with step size h and if $y \in C^{n+1}[a, b]$, *then the local truncation error is* $O(h^n)$.

[Proof]

$$\begin{aligned} y_{i+1} &= y_i + hf(t_i, y_i) + \cdots + \frac{h^n}{n!}f^{(n-1)}(t_i, y_i) + \frac{h^{n+1}}{(n+1)!}f^{(n)}(\xi_i, y(\xi_i)) \\ \Rightarrow \tau_{i+1}(h) &= \frac{y_{i+1} - y_i}{h} - T^{(n)}(t_i, y_i) = \frac{h^n}{(n+1)!}f^{(n)}(\xi_i, y(\xi_i)) \end{aligned}$$

for each $i = 0, 1, \cdots, N-1$. Since $y \in C^{n+1}[a,b]$, we have $y^{(n+1)}(t) = f^{(n)}(t, y(t))$ bounded on $[a,b]$ and $\tau_i(h) = O(h^n)$ for each $i = 1, 2, \cdots, N$. □

4 Runge-Kutta Methods

定理 4.1 *Suppose that $f(t,y)$ and all its partial derivatives of order less or equal to $n+1$ are continuous on $D = \{(t,y) | a \le t \le b, c \le y \le d\}$ ($D = [a,b] \times [c,d]$) and let $(t_0, y_0) \in D$. For every $(t,y) \in D$, there exists ξ between t and t_0 and μ between y and y_0 with*

$$f(t,y) = P_n(t,y) + R_n(t,y)$$

where

$$\begin{aligned} P_n(t,y) = f(t_0,y_0) &+ \left[(t-t_0)\frac{\partial f}{\partial t}(t_0,y_0) + (y-y_0)\frac{\partial f}{\partial y}(t_0,y_0)\right] \\ &+ \left[\frac{(t-t_0)^2}{2}\frac{\partial^2 f}{\partial t^2}(t_0,y_0) + (t-t_0)(y-y_0)\frac{\partial^2 f}{\partial t \partial y}(t_0,y_0) + \frac{(y-y_0)^2}{2}\frac{\partial^2 f}{\partial y^2}(t_0,y_0)\right] \\ &+ \left[\frac{1}{n!}\sum_{j=0}^{n+1}\binom{n}{j}(t-t_0)^{n-j}(y-y_0)^j\frac{\partial^{n+1} f}{\partial t^{n+1}\partial y^i}(t_0,y_0)\right] \end{aligned}$$

$$\text{and} \quad R_n(t,y) = \frac{1}{(n+1)!}\sum_{j=0}^{n+1}\binom{n+1}{j}(t-t_0)^{n+1-j}(y-y_0)\frac{\partial^{n+1} f}{\partial t^{n+1}\partial y^i}(\xi,\mu)$$

*The function $P_n(t,y)$ is called the n*th Taylor polynomial in two variables *for the function f about (t_0,y_0), and $R_n(t,y)$ is the remainder term associated with $P_n(t,y)$.*

4.1 Runge-Kutta Methods of Order Two

$$\begin{cases} y_{n+1} = y_n + h(c_1 k_1 + c_2 k_2) \\ k1 = f(x_n, y_n) \\ k1 = f(x_n + \lambda_2 h, y_n + \mu_{21} h k_1) \end{cases}$$

$$
\begin{aligned}
T_{n+1} &= y(x_{n+1}) - y(x_n) - h\left[c_1 f(x_n, y_n) + c_2 f(x_n + \lambda_2 h, y_n + \mu_{21} h f_n)\right] \\
&= h f_n + \frac{h^2}{2}\left[f'_x(x_n, y_n) + f'_y(x_n, y_n) f_n\right] \\
&\quad - h\left[c_1 f_n + c_2\left(f_n + \lambda_2 f'_x(x_n, y_n) h + \mu_{21} f'_y(x_n, y_n) f_n h\right)\right] + O(h^3) \\
&= (1 - c_1 - c_2) f_n h + \left(\frac{1}{2} - c_2 \lambda_2\right) f'_x(x_n, y_n) h^2 \\
&\quad + \left(\frac{1}{2} - c_2 \mu_{21}\right) f'_y(x_n, y_n) f_n h^2 + O(h^3) \\
\Rightarrow & y_{n+1} = y_n + h f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} f(x_n, y_n)\right)
\end{aligned}
$$

4.2 Midpoint Method

$$
\begin{cases}
\omega_0 = \alpha \\
\omega_{i+1} = \omega_i + h f\left(t_i + \frac{h}{2}, \omega_i + \frac{h}{2} f(t_i, \omega_i)\right) \quad \text{for } i = 0, \cdots, N-1
\end{cases}
$$

Local truncation error: $O(h^2)$.

4.3 Modified Euler Method

$$
\begin{cases}
\omega_0 = \alpha \\
\omega_{i+1} = \omega_i + \frac{h}{2}\left[f(t_i, \omega_i), f\left(t_{i+1}, \omega_i + h f(t_i, \omega_i)\right)\right] \quad \text{for } i = 0, \cdots, N-1
\end{cases}
$$

4.4 Higher-Order Runge-Kutta Methods

Runge-Kutta Order Three:

$$
\begin{cases}
\omega_0 = \alpha \\
k_1 = h f(t_i, \omega_i) \\
k_2 = h f(t_i + \frac{h}{2}, \omega_i + \frac{1}{2} k_1) \\
k_3 = h f(t_i + h, \omega - k_1 + 2k_2) \\
\omega_{i+1} = \omega_i + \frac{1}{6}(k_1 + 4k_2 + k_3)
\end{cases}
$$

Runge-Kutta Order Four:

$$
\begin{cases}
\omega_0 = \alpha \\
k_1 = hf(t_i, \omega_i) \\
k_2 = hf(t_i + \frac{h}{2}, \omega_i + \frac{1}{2}k_1) \\
k_3 = hf(t_i + \frac{h}{2}, \omega - k_1 + \frac{1}{2}k_2) \\
k_4 = hf(t_i + h, \omega_i + k_3) \\
\omega_{i+1} = \omega_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{cases}
$$

4.5 Computational Comparisons

Evaluations per step	$n \in [2,4]$	$n \in [5,7]$	$n \in [8,9]$	$n \in [10,\infty]$
Best possible local truncation error	$O(h^n)$	$O(h^{n-1})$	$O(h^{n-2})$	$O(h^3)$

5 Error Control and the Runge-Kutta-Fehlberg Method

hello world!