

第六次上机作业

曾梦辰*

2024 年 6 月 25 日

摘要

本次作业中我们给出 Euler 法, 修正 Euler 法以及经典 Runger-Kutter 法求解常微分方程数值解的通用程序. 同时, 我们通过计算一个初值问题验证以上方法的收敛阶.

1 问题

本次作业解决如下两个问题.

问题 1. 分别使用 Euler 法, 修正 Euler 法以及经典 Runger-Kutter 法, 编写一个计算常微分方程初值问题数值解的通用程序, 要求以初值问题的方程 $f(x, y)$ 与初值 y_0 , 区间个数 n , 起点与终点 x_0, x_{end} 作为参数.

问题 2. 利用问题 1 中的通用程序计算初值问题

$$\begin{cases} y' = -x^2 y^2, & 0 \leq x \leq 1.5 \\ y(0) = 3 \end{cases} \quad (1.1)$$

与精确解 $y(x) = 3/(1+x^3)$ 作比较, 取不同的步长验证各个方法的收敛阶.

2 计算公式

给定初值问题

$$\begin{cases} y' = f(x, y), & x \in [a, b] \\ y(a) = y_0 \end{cases}$$

将 $[a, b]$ 等分为 n 个小区间 $a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b$, 记 $y(x_i)$ 的近似值为 y_i . 设步长 $h = \frac{b-a}{n}$, 那么我们有计算公式:

*学号: 202011999050

- Euler 法:

$$y_{i+1} = y_i + hf(x_i, y_i)$$

- 改进 Euler 法:

$$\begin{cases} y_{i+1}^{(p)} = y_i + hf(x_i, y_i) \\ y_{i+1} = y_i + \frac{h}{2}(f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{(p)})) \end{cases}$$

也即

$$y_{i+1} = y_i + \frac{h}{2}(f(x_i, y_i) + f(x_{i+1}, y_i + hf(x_i, y_i)))$$

- 经典 Runger-Kutter 法:

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

其中

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \\ k_3 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right) \\ k_4 &= f(x_n + h, y_n + hk_3) \end{aligned}$$

在头文件`ode.h`中, 我们编写了这三种方法对应的程序, 输入的方程与精确解通过函数指针给出. 对给定的初值问题 (1.1), 我们分别取区间个数 $n = 1 \times 10^2, 5 \times 10^2, 1 \times 10^3, 5 \times 10^3, 1 \times 10^4$ 进行数值计算, 并计算在 $y_0 = 0$ 处的局部截断误差

$$R = \frac{1}{h^p}(y(x_1) - y(x_0) - g(x_0, x_1, y_0, y_1))$$

其中 Euler 法取 $g(x_0, x_1, y_0, y_1) = hf(x_0, y_0)$, $p = 2$; 修正 Euler 法取 $g(x_0, x_1, y_0, y_1) = \frac{h}{2}(f(x_0, y_0) + f(x_1, y_0 + hf(x_0, y_0)))$, $p = 3$; 经典 Runger-Kutter 法取 $g(x_0, x_1, y_0, y_1) = \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$, $p = 4$. 我们将数值验证 R 在 0 处收敛到一个固定值. 执行计算的程序源代码在`program6.c`中.

3 计算结果

我们在本节给出三种方法在点 $x = 0.5, 1, 1.5$ 处的值以及截断误差.

$n =$	1×10^2	5×10^2	1×10^3	5×10^3	1×10^4
$x = 0.5$	2.660361	2.667187	2.666037	2.666718	2.666604
$x = 1.0$	1.492714	1.496308	1.499281	1.498956	1.499591
$x = 1.5$	0.666441	0.681845	0.683779	0.685327	0.685521
$R =$	-0.045000	-0.009000	-0.004500	-0.000900	-0.000450

表 1: Euler 法的结果

$n =$	1×10^2	5×10^2	1×10^3	5×10^3	1×10^4
$x = 0.5$	2.648524	2.664881	2.664885	2.666489	2.666489
$x = 1.0$	1.488815	1.495506	1.498876	1.498875	1.499888
$x = 1.5$	0.670155	0.682553	0.684130	0.685397	0.685556
$R =$	1.500010	1.500000	1.500000	1.500013	1.499931

表 2: 改进 Euler 法的结果

$n =$	1×10^2	5×10^2	1×10^3	5×10^3	1×10^4
$x = 0.5$	2.648653	2.664887	2.664887	2.666489	2.666489
$x = 1.0$	1.488778	1.495505	1.498875	1.499550	1.499888
$x = 1.5$	0.670052	0.682549	0.684129	0.685397	0.685556
$R =$	0.000042	0.000002	0.000004	0.043033	-0.462192

表 3: 经典 Runger-Kutter 法的结果

结论

命题 3.1. *Euler* 法具有 2 阶精度, 改进 *Euler* 法具有 3 阶精度, 经典 *Runger-Kutter* 法具有 4 阶精度.

评注 3.2. 注意到当 n 很大时, *Runger-Kutter* 法的 R 在偏离 0, 我们猜测这是因为两个很小的数相除带来了较大的误差.

4 运行环境

程序运行环境: Dell Inspiron 14 Plus 7420, Linux 5.15.146.1-microsoft-standard-WSL2.

编译器: gcc version 9.4.0 (Ubuntu 9.4.0-Ubuntu 20.04.2)