数值分析 code10 实验报告

张景浩 PB20010399

2023.5.18

1 问题介绍

编写 RKF45 或 RKF54 的子程序，实现自适应方法，求解如下微分方程初值问题：

$$\begin{cases} y' = e^{xy} + cos(x-y) \\ y(1) = 3 \end{cases}$$

初始步长取 h=0.01，自适应方法选择：

$$h = 0.9h(\frac{\delta}{|e|})^{\frac{1}{p+1}}$$

其中 δ 是预先设定好的容差，e 是四阶 Runge-Kutta 公式与五阶 Runge-Kutta 公式的差，p 是第一个 Runge-Kutta 公式的阶。

要求在解溢出前终止程序，输出解的范围：[1,?]，并输入一个介于上述范围间的值，应用简单的线性插值计算出对应的函数值。

2 解决方法

基本思路为将四阶方法的五次函数求值包含在五阶方法的六次函数求值中，从而减少计算量，同时应用两种方法得到的结果的差作为对局部截断误差的估计，并通过自适应调整步长来使得局部截断误差 e 的值限定在预先给定的容限 δ 之内。

- 五阶方法

$$y(x+h) = y(x) + \frac{16}{135}F1 + \frac{6656}{12825}F3 + \frac{28561}{56430}F4 - \frac{9}{50}F5 + \frac{2}{55}F6$$

- 四阶方法

$$\bar{y}(x+h) = y(x) + \frac{25}{216}F1 + \frac{1408}{2565}F3 + \frac{2197}{4104}F4 - \frac{1}{5}F5$$

- 六次函数求值

$F1 = hf(x, y)$

$F2 = hf(x + h/4, y + F1/4)$

$F3 = hf(x + \frac{3}{8}h, y + \frac{3}{32}F1 + \frac{9}{32}F2)$

$F4 = hf(x + \frac{12}{13}h, y + \frac{1932}{2197}F1 - \frac{7200}{2197}F2 + \frac{7296}{2197}F3)$

$F5 = hf(x + h, y + \frac{439}{216}F1 - 8F2 + \frac{3680}{513}F3 - \frac{845}{4104}F4)$

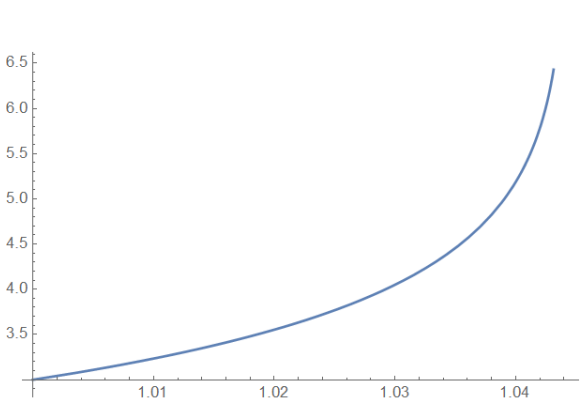
$F6 = hf(x + \frac{1}{2}h, y - \frac{8}{27}F1 + 2F2 - \frac{3544}{2565}F3 + \frac{1859}{4104}F4 - \frac{11}{40}F5)$

本次实验实验 RKF54 方法，容差 $\delta = 10^{-6}$，$e = y(x+h) - \bar{y}(x+h)$。

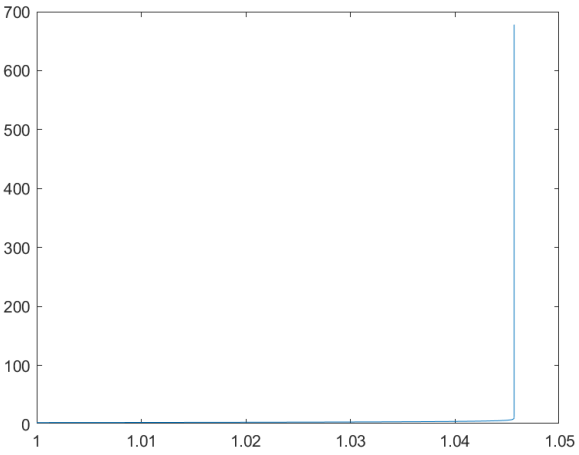
3 编译环境及使用方法

本程用 matlab 编译，使用时调用 outcome.m 文件。

4 实验结果



Mathematica 隐函数图像



RKF54 计算结果图像

图 1: 微分方程初值问题解图像

解的范围：[1,1.045644]

输入值	函数值
1.03	3.984478
1.02	3.534228
1.01	3.222551

表 1: 解范围内的函数值

5 总结

通过 RKF54 解的函数图像可以看出在解溢出之前的部分结果与 Mathematica 给出的隐函数图像基本一致，但是在解的溢出点附近函数值迅速增大，这说明此时步长非常的小，通过对比数据也可以发现绝大部分的步数都集中在了溢出点非常小的一个邻域内。同时由于自适应步长的调整保证了每一步的局部截断误差不超过给定的容差，这使得对于给定点的函数值具有非常高的精度。

A Computer Code

Here we include the computer code.

```
function [x,t]=rkf54(f,a,delta,x0,h,M)
% 输入：一阶微分f(t,x)，区间起点a，容限delta，初值x0，初始步长h，步数上界M
% 输出：函数值x，取值点t
p=5;
t=a;x=x0;
e=1; % 局部截断误差
k=1;
while ~isnan(e)&&~isinf(e)&&k<=M % 解溢出或到达步数上界
    s=t(end);
    y=x(end);

    F1=h*f(s,y);
    F2=h*f(s+h/4,y+F1/4);
    F3=h*f(s+3*h/8,y+3*F1/32+9*F2/32);
    F4=h*f(s+12*h/13,y+1932*F1/2197-7200*F2/2197+7296*F3/2197);
    F5=h*f(s+h,y+439*F1/216-8*F2+3680*F3/513-845*F4/4104);
    F6=h*f(s+h/2,y-8*F1/27+2*F2-3544*F3/2565+1859*F4/4104-11*F5/40);

    e=F1/360-128*F3/4275-2197*F4/75240+F5/50+2*F6/55;
    if abs(e)<=delta
        t(end+1)=t(end)+h;
        x(end+1)=y+16*F1/135+6656*F3/12825+28561*F4/56430-9*F5/50+2*F6/55;
%         if abs((x(end)-x(end-1))/h-f(t(end),x(end)))>10,break;end % 解失去意义
        k=k+1;
    end
    h=0.9*h*(delta/abs(e))^(1/(1+p)); % 自适应调整步长

end
x(end)=[];t(end)=[];
end
```