HW7 Romberg 算法

王启骅 PB20020580

2022 年 12 月 2 日

1 算法描述

首先定义函数 vx,vy，参数为 t（时刻），M（最大迭代数），采用 Romberg 积分算法求解 t 时刻的速度。从区间左端点 a=0 积分到区间右端点 b=t, 分别对 a_x, a_y 积分, 分割区间 h=$\frac{t}{n} = t$, 积分流程与课本相同，首先求得 k=1 情况

$$R_{1,1} = \frac{a(0) + a(t)}{2}h \tag{1}$$

其中 a 代表 x 或 y 方向加速度。

接下来求解当 k>1 时，利用递推式求解。首先取 h=$\frac{h}{2}$ 作为本次迭代得区间划分长度 h 值。对于 k=2-M

$$R_{k,1} = \frac{R_{k-1,1} + 2h\sum_{i=1}^{2^{k-2}} a(0 + (2i-1)h)}{2} \tag{2}$$

对于 j=2-k

$$R_{k,j} = R_{k,j-1} + \frac{R_{k,j-1} - R_{k-1,j-1}}{4^{j-1} - 1} \tag{3}$$

当循环到 $|R_{k,k} - Rk - 1, k - 1| < e$ 时退出循环。

在这里的计算中，为了节省内存，取了 $2 \times M$ 的数组 R，并且取了 count=0，对于每次循环，通过 count 的奇偶来判断 R[0] 或 R[1] 作为本次赋值对象，即用 count 对 2 取模来实现。每次循环结束 count=count+1 。最后输出 $R_{k,k}$ 即为在 t 时刻点的速度。

对于 x,y 坐标点的求解采用完全相同的算法，只是将以上积分过程中的 a_x, a_y 函数替换为 v_x, v_y，这里不再赘述。

2 结果与讨论

在 M=8 情况下，计算得到的轨迹图如图1

计算达到精度的比例如图2 在 M=4-8 时比例迅速增长，总比例从 0.157445 增长到 0.650570 。而到了 M=12 后比例以达到 1，说明已经全部达到精度。但是根据对 x，y 方向分别的讨论，可以看出对于不同的函数，积分达到精度的情况是不同的，y 方向收敛速度明显快于 x 方向。M 是在总体上起到对精度和计算效率的调控的作用，在使迭代能尽量达到精度的情况下，又防止某一特定点迭代所需次数过多而计算速率过慢。所以我们在进行实际运算中，需要根据对于计算的精度、效率双重考虑，取恰当的 M 值使在能够达到需要精度的前提下尽量快速的完成。而且要根据函数的不同，相应的迭代收敛速度也不同，选取不同的 M。

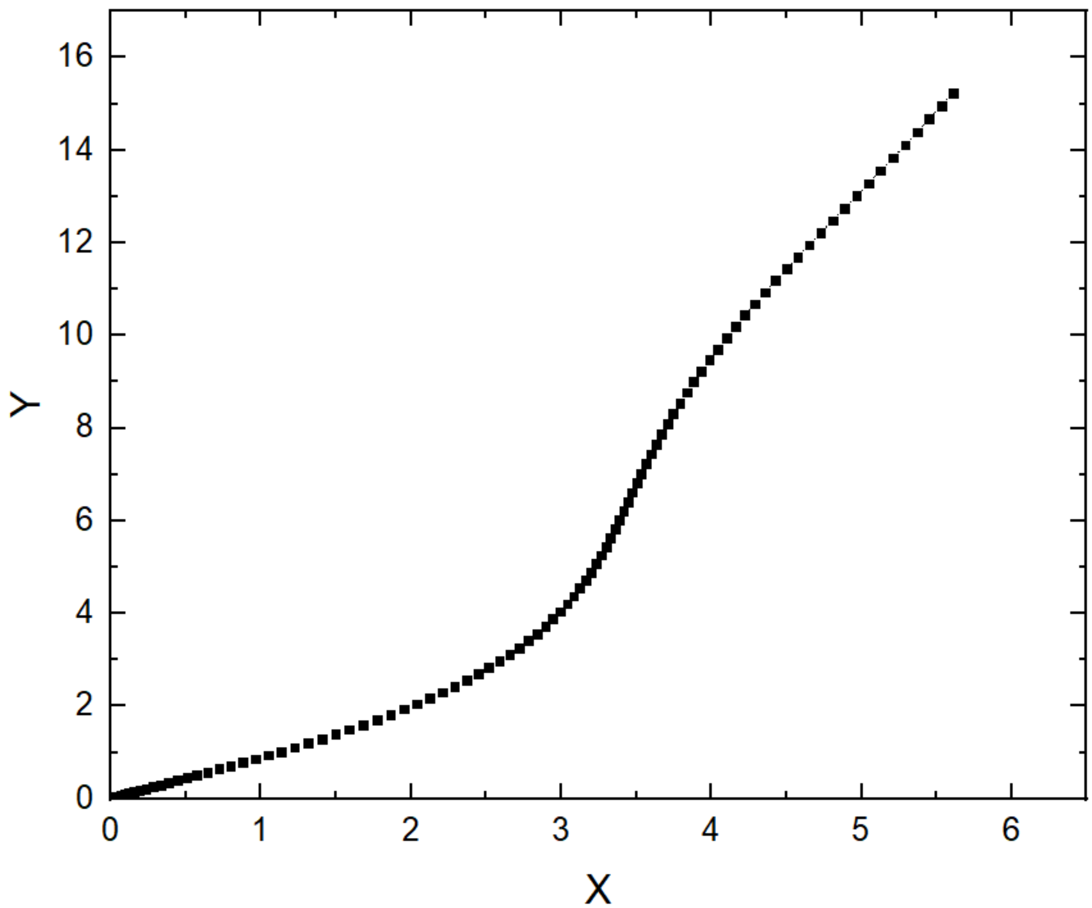

图 1: 轨迹图

M	x方向达到精度比例	y方向达到精度比例	总积分达到精度比例
4	0.119478	0.195565	0.157445
8	0.372393	0.985947	0.650570
12	1.000000	1.000000	1.000000
16	1.000000	1.000000	1.000000
20	1.000000	1.000000	1.000000

图 2: 达到精度的比例