计算方法第七次编程作业

PB20511896 王金鑫

1 题目

模拟一个位于二维平面上的质点运动的问题。质点初始时 (t = 0) 位于原点，速度为 0。质点受到来自水平方向 (x 方向) 和垂直方向 (y 方向) 的两个独立的力，力会随时间变化，记为 Fx(t)，Fy(t)。

因此速度和位移的表达式如下（以 x 分量为例）：

$$v_x(t) = \int_0^t a_x(s)ds$$

$$x(t) = \int_0^t v_x(s)ds = \int_0^t \int_0^s a_x(r)drds$$

给定两个方向的加速度如下：

$$a_x(t) = \frac{\sin(t)}{\sqrt{t}+1}$$

$$a_y(t) = \frac{\ln(t+1)}{t+1}$$

使用 Romberg 积分进行数值积分，计算出质点在时刻 $t \in \{0.1, 0.2, 0.3, \ldots, 10\}$ 的位移 $(x(t), y(t))$ 。其中，Romberg 的初始区间数取 $n = 1$，精度控制值取 $e = 10^{-6}$ ，最大迭代次数取 $M = 8$ 。

2 原理

Romberg 积分递推公式如下：

$$R_{k,j} = R_{k,j-1} + \frac{R_{k,j-1} - R_{k-1,j-1}}{4^{j-1}-1}, \qquad k = 2, 3, \ldots$$

Where j=1,2,...分别代表梯形、Simpson、Cotes 积分，k 代表分点数为 $n2^{k-1}$（步长 $h_k = \frac{h}{2^{k-1}}$）。

如果将 $R_{k,j}$ 看作下三角矩阵，则通过上述递推公式可实现按行递推。按列递推可通过下式实现：

$$R_{k,1} = \frac{R_{k-1,1}}{2} + h_k \sum_{i=1}^{2^{k-2}} f(a + (2i-1)h_k)$$

通过递推公式计算 $R_{k,k}$。当 $|R_{k,k} - Rk-1, k-1| < e$ 时停止迭代。Where e 为给定的控制精度。

对于题给积分，由于加速度解析式已知，故计算速度的积分可直接使用 Romberg 算法进行积分；计算位移时需要用到的不同时间的速度值可用 Romberg 积分计算出之后再代入计算位移。

3 结果

运行结果如下图所示。

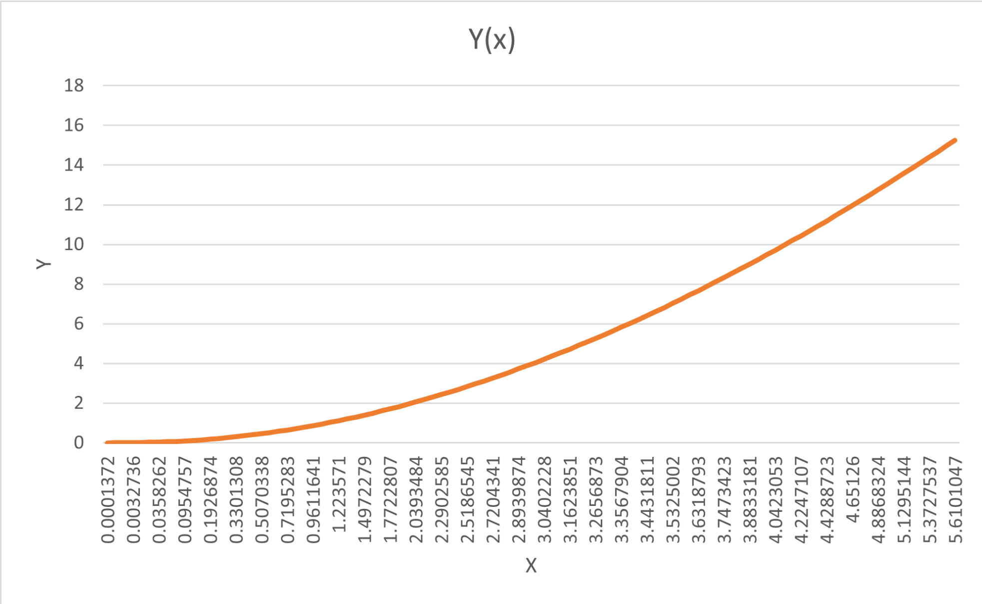

图 1: 质点的轨迹

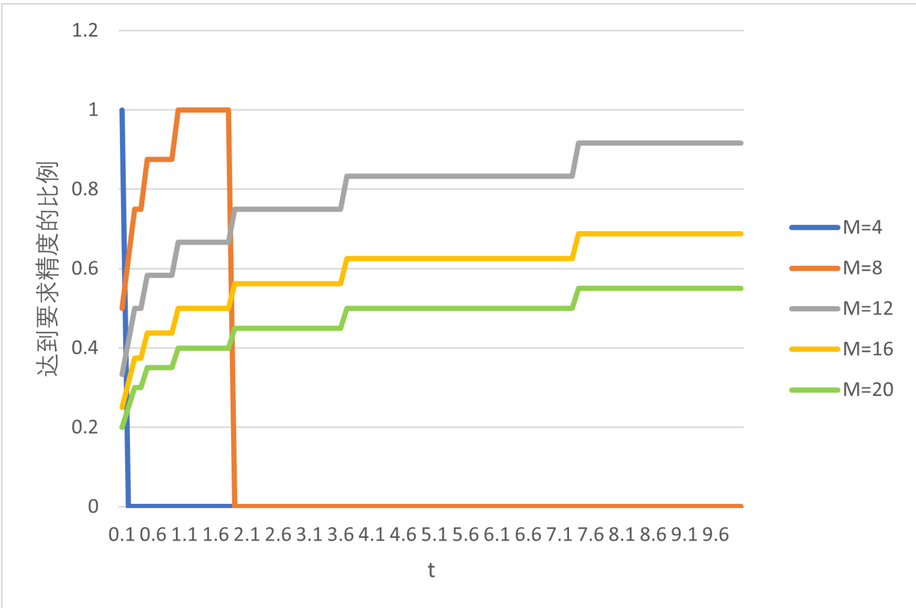

图 2: 以 V_x 为例，取不同 M 值时，不同 t 时刻 Romberg 积分达到要求精度的迭代次数与 M 的比例

M	4	8	12	16	20
Rate	0.0100000	0.1800000	1.0000000	1.0000000	1.0000000

图 3: 以 V_x 为例，取不同 M 值时 Romberg 积分达到要求精度的比例

4 结果分析

图 3 为在不同的 M 取值下，Romberg 积分达到要求精度的比例，即“达到误差要求的次数/调用的总次数”。由图可知，当 M=4、8 时，比例很小，不到 20%。而当 $M \geq 12$ 时，比例均为 1。比例随 M 的增大而增大。

由图 2 也可看出，当 M=4 时，只有 t=0.1 时刻的积分达到了要求精度；当 M=8 时，t=0.19 时刻及以后的积分都没有达到要求精度。从图中可明显看出，当 t 越大，即积分范围越大时，达到要求精度所需的迭代次数也会增大。当需要的迭代次数超过 M 时就不会得到满足要求精度的结果。

由此可知，M 越大，得到的解的精度越高。但是迭代的次数也会随着 M 增大而增大，即计算耗时会增加。且这里用到的算法程序使用矩阵来存储 $R_{k,k}$ 的值，因此 M 增大也会增加对内存的占用。如果 M 过大，在迭代所需次数小于 M，则多余的内存将会被浪费。但若 M 过小，则有可能会达不到我们所要的精度。但若精度设置的过小，那么将会耗费大量算力和时间。所以一个可能的选取 M 的较好的方法为：将 M 设置为与所能接受的最大运算时长相对应的迭代步数。