解题报告

黄于翀

2023 年 7 月 14 日

目录

1 问题一 两相横流问题

这是一道求解颗粒在稳定槽流的运动轨迹的问题，只要进行一定的受力分析，通过加速度和速度位移之间的关系，便可以得到轨迹图。

1.1 解题思路

首先明确一点，画图的时候找到一系列点的集合，使用plot函数实现绘图，故使用两个数组分别存储x坐标和y坐标。

这道题首先需要解决的是受力分析，这一部分为了配合画图时的x和y坐标因此在受力分析时分为x方向和y方向。

有了受力分析就有了水平和竖直方向上的加速度，也就可以据此算出速度和位移。因为其受力状态在不同时间，不同位置都是不一样的，所以加速度也在一直发生着变化，所以要在每次改变位置后重新进行分析。在这里的解决方法是选取一个十分小的时间间隔dt，进行迭代，直到颗粒超出范围。

1.2 公式推导

这里给出关键公式的推导，主要是受力分析时两个方向的受力情况。

已知

$$\overrightarrow{F_d} = \frac{1}{2}C_d\rho_{fluid}\left|\overrightarrow{V}_{fluid} - \overrightarrow{V}_s\right|\left(\overrightarrow{V}_{fluid} - \overrightarrow{V}_s\right)A_s + m\overrightarrow{g}$$

观察不难发现，除了重力之外，另一个力的大小与速度之差的平方成正比，方向由矢量差的方向决定，故现在将其分为水平和竖直两个方向研究。

鉴于已经分为水平和竖直两个方向，力的方向已经确定，故以下计算只进行标量间的计算。

相对速度

$$V_{relative} = \sqrt{\left(V_{fluid} - V_s\right)^2 + {V_y}^2}$$

水平方向

$$F_x = \frac{1}{2}C_d\rho_{fluid}V_{relative}\left(V_{fluid} - V_x\right)A_s$$

竖直方向

$$F_y = \frac{1}{2}C_d\rho_{fluid}V_{relative}\left(0 - V_x\right)A_s + mg$$

有了这两个方向的受力，那么剩下的只需要应用牛顿第二定律以及简单的运动学的公式，就可以计算出每一个小的时间间隔 dt的位移，并更新新的速度和加速度。

1.3 代码实现

在这里主要展示迭代的部分，其余部分放在压缩包中一并上交（LaTeX环境未配置好）

```
while x0>=0 && x0<=length && y0>=-width/2 && y0<=width/2
    vf = U_max*(1-y0*y0);   % 算出该位置流体的速度
    v_relative = sqrt((vf-vx)^2 + vy^2); % 颗粒和流体相对速度绝对值
    Fx = 0.5*Cd*den_f*A_s*v_relative*(vf-vx);% 水平方向受力
    Fy = 0.5*Cd*den_f*A_s*v_relative*(0-vy) - m*g; % 竖直方向受力
    ax = Fx/m;
    ay = Fy/m;
    vx = vx + ax*dt;
    vy = vy + ay*dt;
    x0 = x0 + vx*dt;
    y0 = y0 + vy*dt;      % 算出加速度速度及位置
    x(end+1) = x0;
    y(end+1) = y0;  % 将位置信息储存起来
```

图 1: 问题一迭代部分

1.4 运行结果

在进行测试的时候，我写了一个脚本对函数进行四次调用，因此在同一个图上同时画出四个颗粒的轨迹

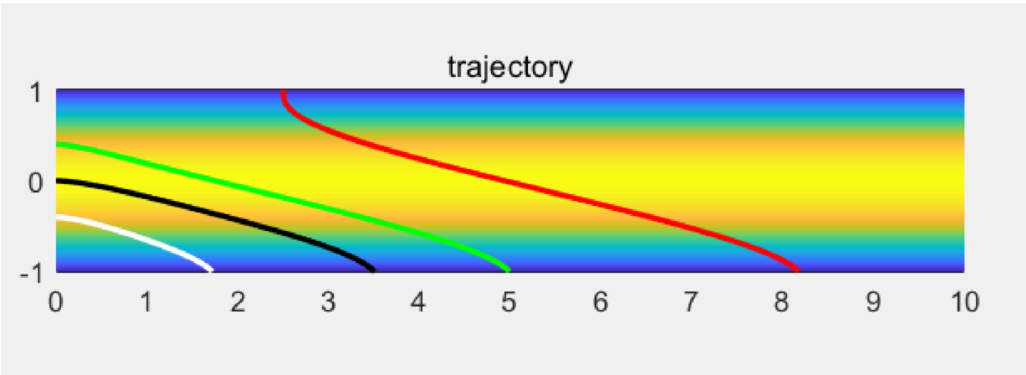

图 2: 问题一运行结果

由图2,我们可以清楚地看到颗粒运动的轨迹。

2 问题二 颗粒轨迹

这一道题实际上可以说是问题一的弱化版，没有了问题一受力分析的部分，直接按照运动学的规律进行编写即可。唯一的不同就是题目要求返回一个矩阵 $state$。

2.1 代码实现

```
function state = ComputeTrajectory(dt,timesteps)
    t = 0;
    x0 = cos(0)*dt;  % 第一个点
    y0 = sin(0)*dt;
    x = x0;          % 用来储存坐标
    y = y0;
    i = 1;
    while(i<timesteps)
        t = t + dt;
        vx = cos(t*6);
        vy = sin(t*6);       % 速度
        x0 = x0 + vx*dt;
        y0 = y0 + vy*dt;     % 新的坐标
        x(end+1) = x0;       % 更新
        y(end+1) = y0;
        i = i+1;
    end

    state = [x;y];      % 合成state
end
```

图 3: 问题二函数源代码

如上所示,得到矩阵。

2.2 运行结果

在这里，我将结果解析解进行比较

通过积分的方式，我们不难算出颗粒轨迹的解析解

$$x = \frac{1}{6}\sin 6t$$

$$y = -\frac{1}{6}\cos 6t + \frac{1}{6}$$

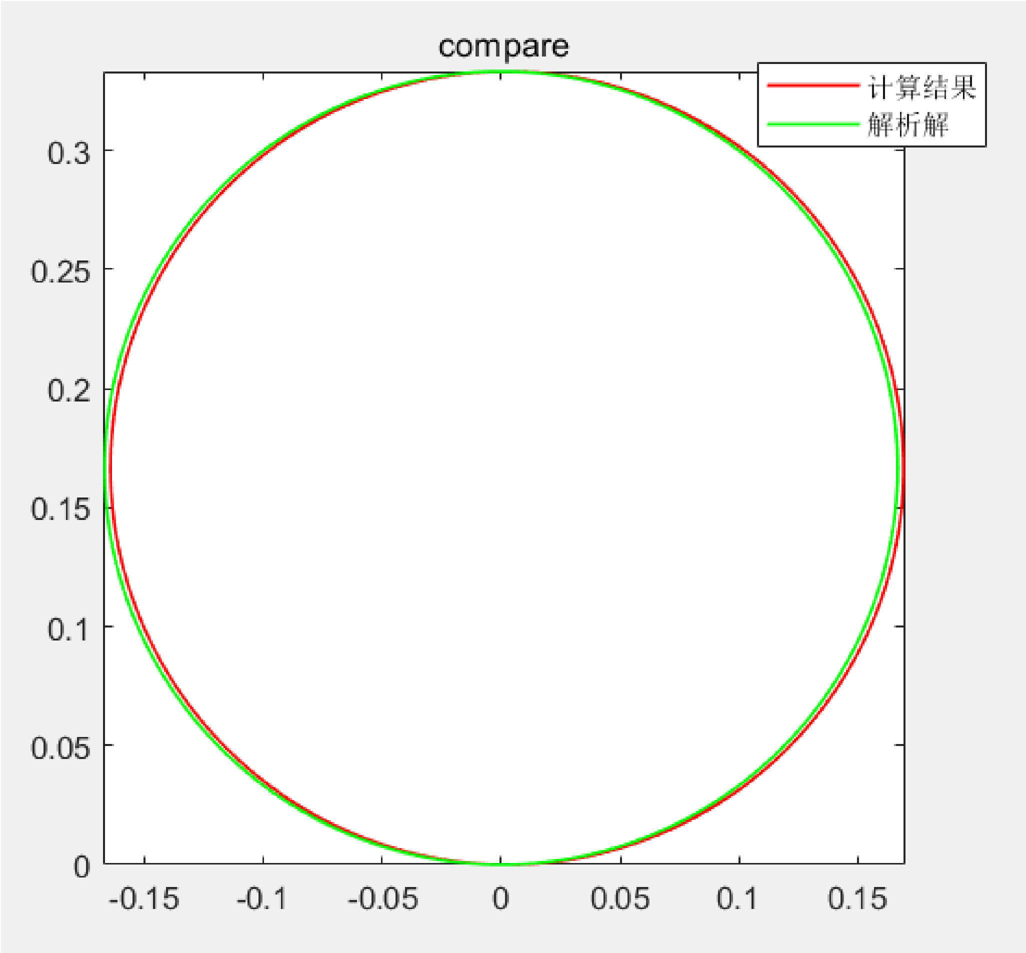

图 4: 与解析解进行比较

通过图4,我们可以看到，运行结果和解析解非常相近，但还是有一定的偏差。

3 问题三 二维稳定热传导模型

这道题的难点其实在于如何去将拉普拉斯进行转换，使程序比较好写。虽然MATLAB 有一些直接求解微分方程的工具，但我不认为直接使用它们求解是一个优秀的做法。

3.1 公式推导

关键点在于如何对拉普拉斯方程进行转换，以获得可以应用于编程的递推公式，这样就可以应用迭代的方式进行求解。

拉普拉斯方程

$$\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}=0$$

下面对公式进行简化，以得到迭代时所需的递推式。

$$\frac{\partial T}{\partial x}=\frac{T\left(x+dx,y\right)-T\left(x,y\right)}{dx}$$

$$\frac{\partial^2 T}{\partial x^2} = \frac{T(x+dx,y) - 2T(x,y) + T(x-dx,y)}{dx^2}$$

同理，对 y 求偏导，可以得到

$$\frac{\partial^2 T}{\partial y^2} = \frac{T(x,y+dy) - 2T(x,y) + T(x,y-dy)}{dy^2}$$

那么只要在处理的时候，将dx和dy取相同的值，那么我们就可以得到

$$T(x,y) = \frac{T(x+dx,y) + T(x-dx,y) + T(x,y+dy) + T(x,y-dy)}{4}$$

也就是说中间一个点的值，是周围四个之和的四分之一。至此，我们将最关键的迭代式子推了出来。

3.2 代码实现

这里我们同样给出迭代部分的代码

```
while diff > tolerance
    T_old = T; % 上一次的数据
    for i = 2:n-1       % y方向
        for j = 2:n-1   % x方向
            % 现在的
            T(i,j) = (T(i-1,j) + T(i+1,j) + T(i,j-1) + T(i,j+1)) / 4;
            % 经过数学推导及部分近似得到的递推式
        end
    end
    diff = max(max(abs(T - T_old))); %误差
end
```

图 5: 问题三迭代法代码

如图5所示，我们可以自行调整精度值，或者调整dx和dy的大小，来获得不同精度的图

3.3 运行结果

在这里我给出两种作图方式，精度为10^{-9},两个方向各分为200份，即有200×200的网格，运行时间大概需要5秒。

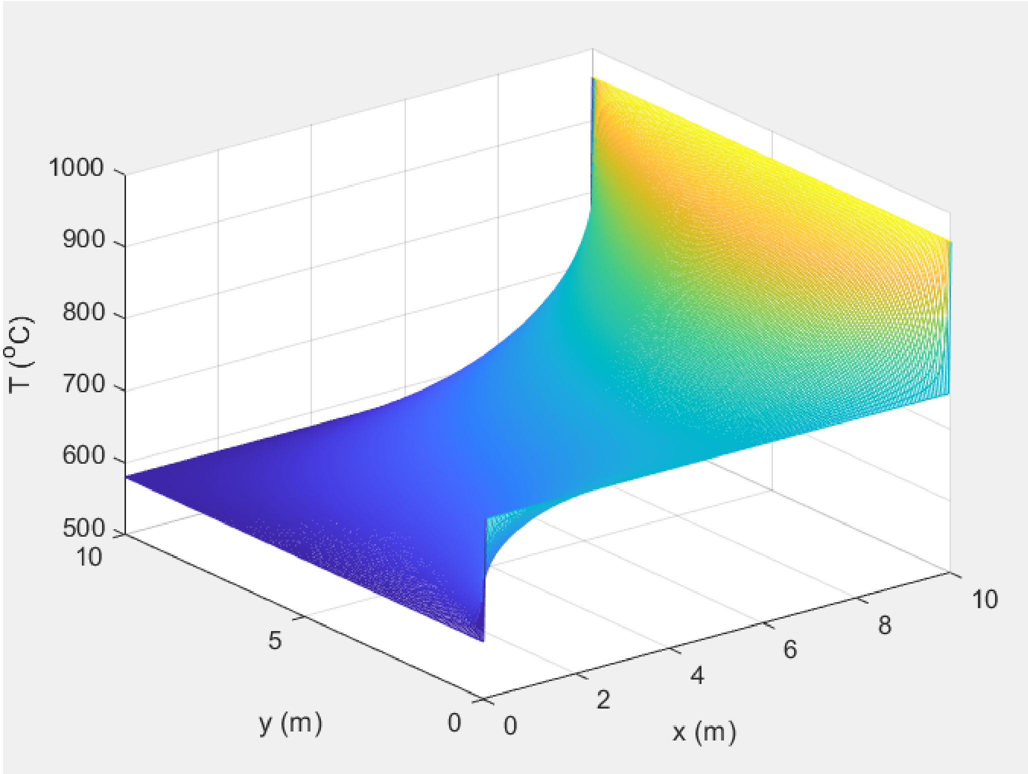

图 6: 使用**mesh**函数作图

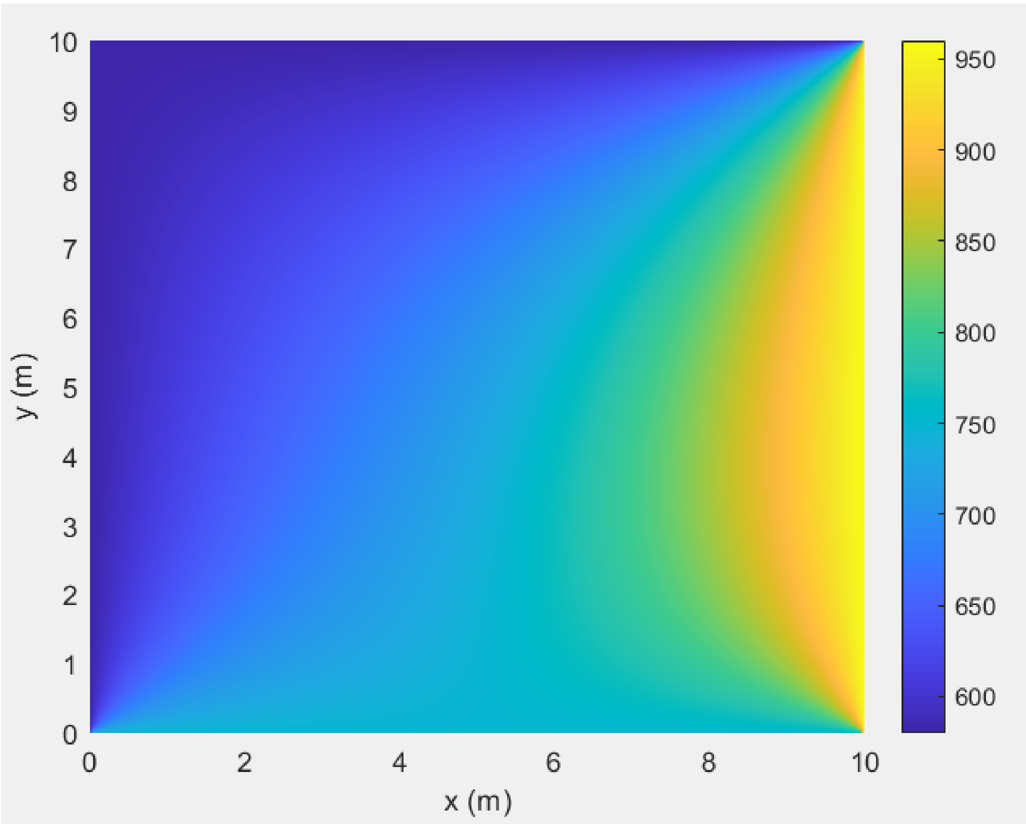

图 7: 使用**surface**函数作图

以上两个图都能将温度分布可视化。

致谢

感谢库晓珂老师一个星期的教学和指导。

感谢许多同学相互间的讨论，碰撞出思想上的火花。

感谢杨挺同学在保存数据上为我提供了许多帮助。

在此致谢所有帮助过我的人。