1D and 3D Numerical Thermo-conduction Simulation of Solid (CN-version)

1D & 3D 固体数值传热仿真

有三个传热仿真引擎

需要安装 numpy, matplotlib, seaborn 第三方库

本文档包含**项目框架**，**原理讲解**，**使用说明**三部分

原理部分有大量公式，使用 Latex 语法写成（一般的Markdown 编辑器均可正确编译显示，但是 Github 的不行）

有任何疑问欢迎邮箱联系 xiewansen@outlook.com

Framework of Projects

- README.md
- 1D_Dynamics
  - **dynamic_Thermo_Conduction.py**
  - **simple_Dynamic_Thermo_Conduction.py**
  - initialState.txt
  - readme.txt
  - setup.txt
  - Graph
    - simple_ThermoDynamic_Simulation.png
    - thermoTransfarDiagram.png
- 3D_BlockThermoConductionSimulation
  - main.py
  - shapeGenerate.py
  - **3D_5.py**
  - visualization.py
  - Data
    - Guide.npy
    - Structure.npy
    - Tdistribution.npy
    - TdistributionRes.npy
  - Graph
    - Result.png
  - Setup

- readme.txt
- setup.txt

原理讲解

这是固体中的传热方程

$$\frac{\partial T}{\partial t}=\alpha\nabla^2T+\frac{\dot{q}}{k}$$

符号	物理量	单位
k	热导率	$W\cdot m^{-2}\cdot K^{-1}$
$\dot{q}$	产热率	$W\cdot m^{-3}$
T	温度	K
t	时间	s
α	热扩散系数	$K\cdot s^{-1}$

计算过程中使用三维直角坐标系，欧拉（控制体）观点，衡算正交网格边界的热通量，离散计算散度。

在一维空间中

假设物体内部不产生热且具有各向同性

则方程为:

$$\frac{\partial T}{\partial t}=\alpha\frac{\partial^2}{\partial x^2}T$$

离散化后(i为空间角标，w为时间角标，上限值为n)

$$\frac{T_{i,w+1}-T_{i,w}}{\Delta t}=\alpha\frac{(T_{i+1,w}-T_{i,w})-(T_{i,w}-T_{i-1,w})}{(\Delta x)^2}$$

设定空间微元 $\Delta x=0.0001m$ 与时间微元 $\Delta t=0.0001s$ (具体数值可由人为给定，若后续计算不收敛，请尝试调大 Δx 并减小 Δt，二者越小计算结果越精确)

通过设定边界条件获得

$$T_{i=0,w}=T_1=f_1(w)$$

$$T_{i=n,w}=T_2=g_1(w)$$

通过设定初始条件获得

$$T_{i,w=0} = T_i = h_1(i)$$

通过欧拉前项法求解偏微分方程

$$T_{i,w+1} = \alpha \frac{T_{i+1,w} + T_{i-1,w} - 2T_{i,w}}{(\Delta x)^2} \Delta t + T_{i,w}$$

计算每个点的温度值并绘制图像

在三维空间中

假设物体内部不产生热且具有各向同性

在直角坐标系中，方程为：

$$\frac{\partial T}{\partial t} = \alpha(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2})T$$

离散化后(i, j, k 为空间角标，w为时间角标，上限值为n)

$$\begin{aligned}\frac{T_{i,j,k,w+1} - T_{i,j,k,w}}{\alpha \Delta t} = & \frac{T_{i+1,j,k,w} + T_{i-1,j,k,w} - 2T_{i,j,k,w}}{(\Delta x)^2} \\ & + \frac{T_{i,j+1,k,w} + T_{i,j-1,k,w} - 2T_{i,j,k,w}}{(\Delta y)^2} \\ & + \frac{T_{i,j,k+1,w} + T_{i,j,k-1,w} - 2T_{i,j,k,w}}{(\Delta z)^2}\end{aligned}$$

设定空间微元 $\Delta x = \Delta y = \Delta z = 0.0001m$ 与时间微元 $\Delta t = 0.0001s$ (具体数值可由人为给定，若后续计算不收敛，请尝试调大 Δx 并减小 Δt，二者越小计算结果越精确)

通过设定边界条件获得

$$T_{i,j,k,w} = T_1 = f_3(i,j,k,w)|_{surface}$$

$$T_{i+1,j,k,w} = T_2 = g_3(i,j,k,w)|_{surface}$$

通过设定初始条件获得

$$T_{i,j,k,w=0} = T_0 = h_3(i,j,k)$$

通过欧拉中项法求解偏微分方程

$$T_{i,j,k,w+1} = \frac{\alpha \Delta t}{\Delta x^2}(T_{i+1,j,k,w} + T_{i-1,j,k,w} + T_{i,j+1,k,w} + T_{i,j,k+1,w} + T_{i,j,k-1,w} + T_{i,j-1,k,w} - 6T_{i,j,k,w}) + T_{i,j,k,w}$$

计算每个点的温度值

由于工作量巨大，这个程序被分成了三个部分，用 numpy 数组进行交互。

构建结构的 **shapeGenerate.py**
传热计算（仿真）的 **3D_5.py**
结果可视化 **visualization.py**

详细内容请看下面的使用说明

1D-传热仿真

simple_Dynamic_Thermo_Conduction.py

直接运行这个文件，根据提示进行操作最终计算结果生成于 Graph\Simple_ThermoDynamic_Simulation.png 文件，有其他生成格式要求请在程序中更改操作。模拟材质为铁，若有其他需求请在程序脚本中编辑。

dynamic_Thermo_Conduction.py

1. 配置 Setup.txt

在 setup.txt 文件中写入边界条件，半角逗号分割元素，回车分割组。

写入格式：

```
初始平均温度/K，左端环境温度/K，右端环境温度/K，杆长/m，模拟时间/s，空间模拟点数目，时间模拟点数目
```

示例(金属铁杆传热温度分布)，允许多行，计算结果将会叠加到同一张图中

```
295,340,373,0.003,0.0,30,1000
295,340,373,0.003,0.1,30,1000
295,340,373,0.003,1.0,30,1000
295,340,373,0.003,5.0,30,1000
295,340,373,0.003,9.0,30,1000
295,340,373,0.003,15,30,10000
295,340,373,0.003,60,30,10000
```

左右环境温度输入为0时，认为杆件绝热

2. 配置 initialState.txt

杆件初始温度设定（初始条件设定，每一个模拟点的初始温度）

在 initialState.txt 文件中输入杆内个空间模拟点的温度与平均温度之差，默认为0

不写即默认为0，但是注意，该文件夹中至少有一个数字，即使不做个更改也要在内部加一个0，否则运行过程会出问题。

输入顺序从左向右

3. 运行 dynamic_Thermo_Conduction.py

最终计算结果生成为 Graph\ThermoConductionDiagram.png 文件，有其他生成格式要求请在程序中更改操作。模拟材质为铁，若有其他需求请在程序中输入。

注：千万不要给 setup.txt 与 initialState.txt 两个文件改名，保存文本使用UTF-8格式，半角符号。

3D-传热仿真

总体来讲要远比 1D-传热仿真复杂，分为：

```
1. 结构体设置与生成 shapeGenerate.py
2. 材质配置 & 传热计算 3D_5.py
3. 数据可视化 visualization.py
```

在 Setup\setup.txt 文件中写入脚本，半角分号分割元素

1.写入格式：（不要换行！！不要写括号的单位！！）

```
空间微元尺寸(m);时间微元长度(s);仿真时间(s);物体导热率(W/(K*m^2));比热容(J/(K*kg));密度(kg/m^3)
```

示例（果冻材质）：

```
0.001;1;800;0.683;4180;1000
```

注意：

```
应至少进行 100 次循环，即：仿真时间/时间微元长度 >= 100
否则进度条可能出现异常，但是显示进度的数值是正常的
```

2.仿真物体设置：

在 Data\shapeGenerate.py 中生成结构数组 Data\structure.npy，温度分布数组 Data\Tdistribution.npy 默认脚本是绘制实心圆的程序，请自行仿照书写，随后使用Setup\setup.py进行操作。结构数组中：0表示虚无，1表示边界，2表示模拟的微元。详情请见注释。

3.运行模拟：

为方便使用，特集成了 main.py 辅助使用。（您也可以依次使用 shapeGenerate.py, 3D_5.py, visualization.py）

打开 main.py 并运行，随后系统会主动调用 3D_5.py 进行传热的PDE计算，使用 Data\structure.npy 与 Data\Tdistribution.npy 作为参数计算温度分布并获得温度分布结果 Data\TdistributionRes.npy 与引导矩阵结果（本体数值 = 1,非本体数值 = 0）Data\Guide.npy 。

随后 main.py 会调用 visualization.py 进行数据可视化输出，该程序会调用 Data\TdistributionRes.npy 与 Data\Guide.npy 进行数据加工，输出某个截面的温度分布，以热图的形式展现，具体信息在对应程序中配置。

4.可视化输出

最后的结果会以图片形式展示模拟物体（示例程序为球体）各个截面温度分布。最终保存为 Graph\Result.png

可以根据 visualization.py 的注释根据需求自行编辑可视化。

注意：

千万不要给所有文件重命名 ，保存文本使用UTF-8格式，半角符号