说明: index1024_01

**《Python与机器学习》**

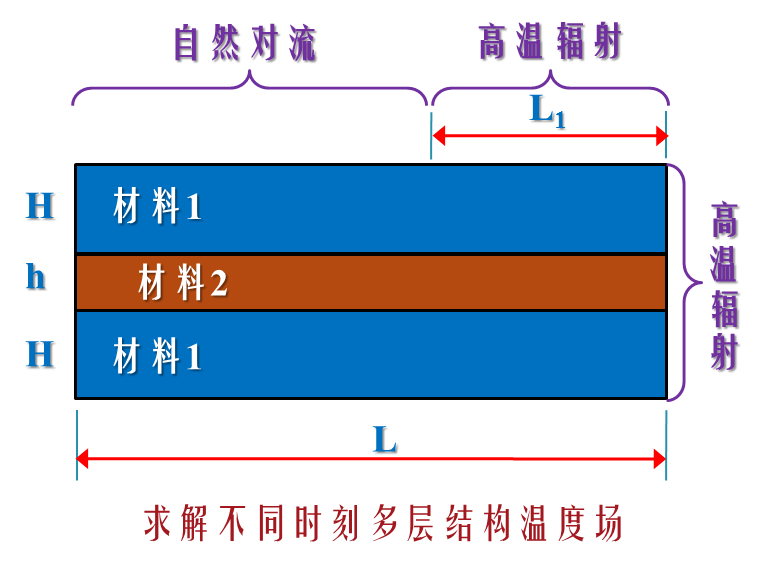
**课程大作业**

|  |  |
| --- | --- |
| 姓名 | 李宗霖 |
| 学号 | 2021300045 |
| 所在学院 | 教育实验学院 |
| 指导老师 | 黄湛钧 |
| 课程代码 | U01M11182.01 |

**西北工业大学航空学院**

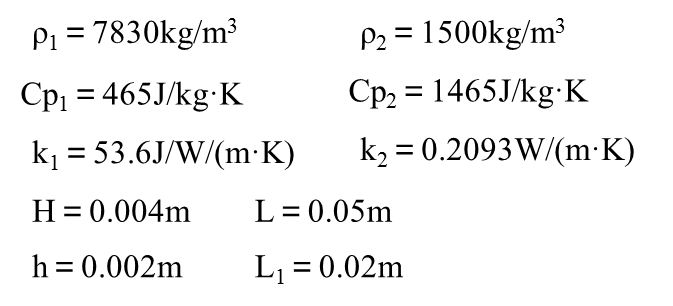
公式章 1 节 1**一、问题分析**

如图1所示，材料1和材料2组成三明治结构，其中右端处于973k的燃气中，下边界和左边界为绝热边界，上边界左段处于大气中，大气温度为283 k，求5s、15s、30s和60s时刻计算域内温度分布。



**图1 膏体发动机示意图**

其中参数为：



* 1. **控制方程**

分析问题可得，此问题可视为常物性二维非稳态导热问题，控制方程为：

|  |  |
| --- | --- |
|  | (1-1) |

* 1. **边界条件**

（1）左边界、下边界均为绝热边界，即，属于第二类边界条件（Dirichlet条件）。

（2）左上边界是自然对流边界，属于第三类边界条件（Robin条件）。

（3）右上边界、右边界是辐射换热边界，属于辐射边界条件。

* 1. **网格划分**

在进行非稳态问题求解计算时，需要对方程进行时间离散，可以采用显式、全隐、C-N格式。

显式格式用当前步的值直接表示下一步的值，计算较为简单，但步长受限制，一般时间步长要很小。显式格式稳定性判定准则：

隐式格式需要解方程组，求解一步计算量大，尤其对大型非线性方程组求解困难，但对步长没有要求（步长过大也会导致结果的精确性下降），稳定性好。

故我们：

（1）所有网格大小均采用0.5mm x 0.5mm，网格数为；

（2）先Matlab编写隐式格式进行计算，时间步长0.05s；

（3）在有余力的情况下，使用Python撰写三种时间格式的程序，其中显式格式时间步长取0.002s。

**二、控制方程推导**

* 1. **离散控制方程**

在时间之间和节点P的控制容积内对式(1-1)进行积分：

|  |  |
| --- | --- |
|  | (1-2) |

以下分别对式(1-2)中各个项进行积分运算：

非稳态项：

|  |  |
| --- | --- |
|  | (1-3) |

扩散项：

|  |  |
| --- | --- |
|  | (1-4) |

，

对时间的积分可一般化的表达为

|  |  |
| --- | --- |
|  | (1-5) |

其中，为位于之间变化的加权因子。

得到

|  |  |
| --- | --- |
|  | (1-6) |

同理得，

|  |  |
| --- | --- |
|  | (1-7) |

源项：

|  |  |
| --- | --- |
|  | (1-8) |

将式(1-3)、式(1-6)、式(1-7)、式(1-8)的积分结果代入式(1-2)，则得到所对应的离散方程：

|  |  |
| --- | --- |
|  | (1-9) |

整理后得：

|  |  |
| --- | --- |
|  | (1-10) |

写为离散方程的通用形式：

|  |  |
| --- | --- |
|  | (1-11) |

其中

|  |  |
| --- | --- |
|  | (1-12) |

* 1. **边界条件**
     1. **左边界**：绝热边界，令，可推出：

|  |  |
| --- | --- |
|  | (1-13) |

* + 1. **下边界**：绝热边界，令，可推出：

|  |  |
| --- | --- |
|  | (1-14) |

* + 1. **左上边界**：自由对流边界，已知大气温度为，即。查表可得此时大气物性：

求得：

对理想气体，热膨胀系数

取，计算得

故

其他

|  |  |
| --- | --- |
|  | (1-15) |

可推出：

|  |  |
| --- | --- |
|  | (1-16) |

* + 1. **右上边界**：辐射换热边界，已知右端高温燃气973K，取物体表面发射率为，可以推导出：

|  |  |
| --- | --- |
|  | (1-17) |

可推出：

|  |  |
| --- | --- |
|  | (1-18) |

* + 1. **右边界**：辐射换热边界，同右上边界（4），可得：

|  |  |
| --- | --- |
|  | (1-19) |

最终得到各边界条件的系数，如表1所示。

**表 1 离散方程系数表**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 界面 |  |  |  |  |  |  |  |  |
| 左边 |  |  | 0 |  |  |  | 0 | 0 |
| 下边 |  |  |  |  | *0* |  | 0 | 0 |
| 右边 |  |  |  | 0 |  |  |  | 0 |
| 左上 |  |  |  |  |  | 0 |  | 0 |
| 右上 |  |  |  |  |  | 0 |  | 0 |

* 1. **物性交界面**

传热系数采用调和平均法进行计算，在物性交界面处，有：

|  |  |
| --- | --- |
|  | (1-20) |

# 开发工具

Visual Studio Code

Anaconda3

# 运行环境

CPU：11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz

内存：16GB

GPU：NVIDIA GeForce RTX 3050 Laptop GPU，独立显存：4G

Python版本3.10.12

第三方模块：

os

numpy == 1.24.4

scipy == 1.10.1

matplotlib == 3.7.2

random

tqdm == 4.65.2

multiprocessing

imageio == 2.31.5

time

functools

torch == 2.1.0.dev20230601+cu121

torchaudio == 2.1.0.dev20230601+cu121

torchvision == 0.16.0.dev20230601+cu121

# 代码结构

1. Node\_info

存储每个节点属性的类，包括但不限于加权因子f 、各种系数、历史时间与温度列表等，以及一些批量处理函数\_\_init\_\_(节点类定义的初始化), \_\_repr\_\_(输出样式),normalize(各类系数初始化函数),a\_p (A系数计算) S\_calculate(源项计算函数)

1. generate\_nodes()

使用内节点法划分网格

1. def Coefficient\_normalize()

控制方程系数初始化

1. A\_init()

矩阵A的计算函数

1. B\_init()

源项系数B的计算函数

1. T\_plot()

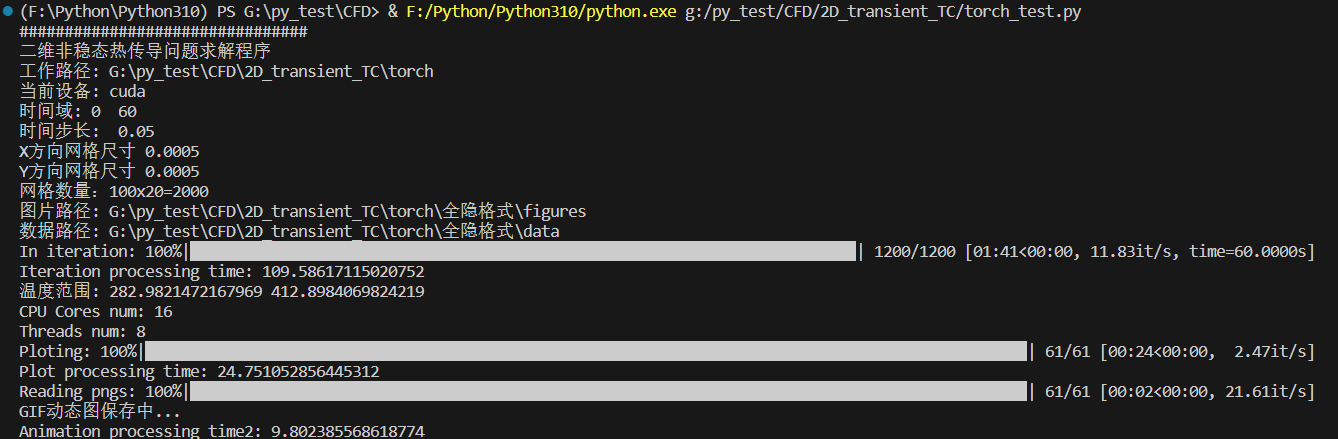
温度云图绘制函数 -

1. data\_write\_csv()

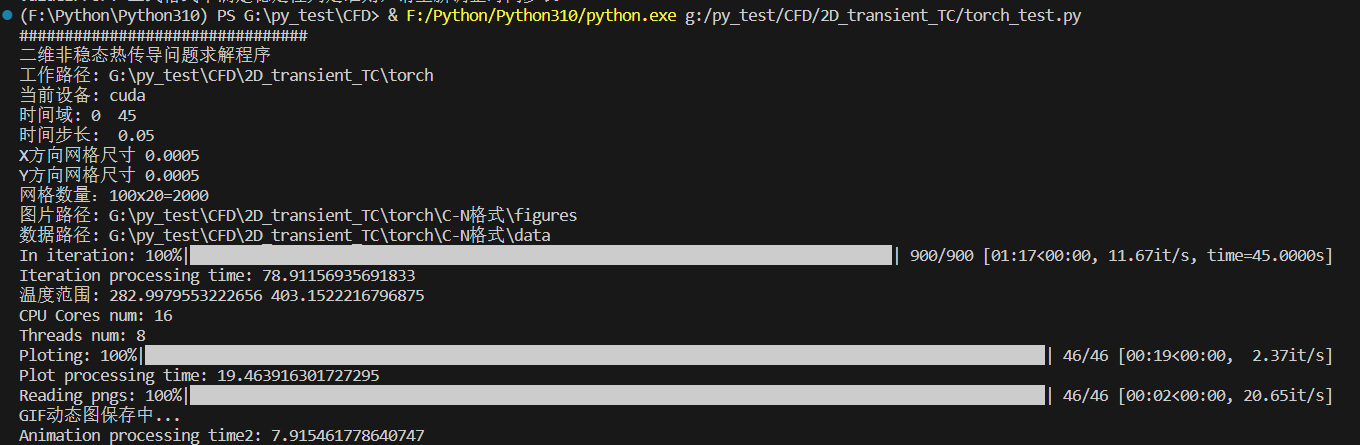
将数据写成csv文件保存的函数

# 运行截图

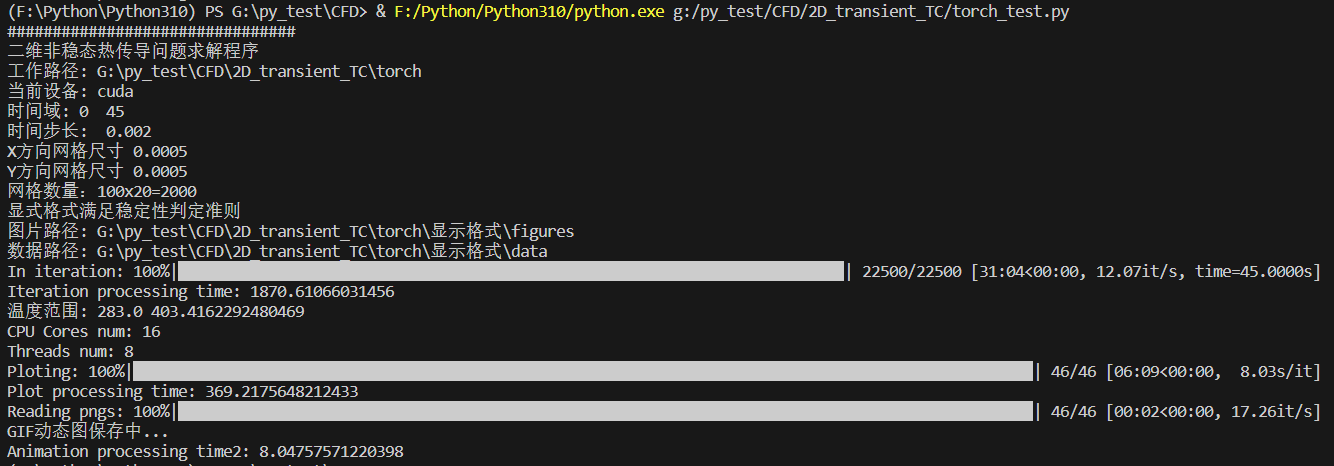
1. 全隐格式



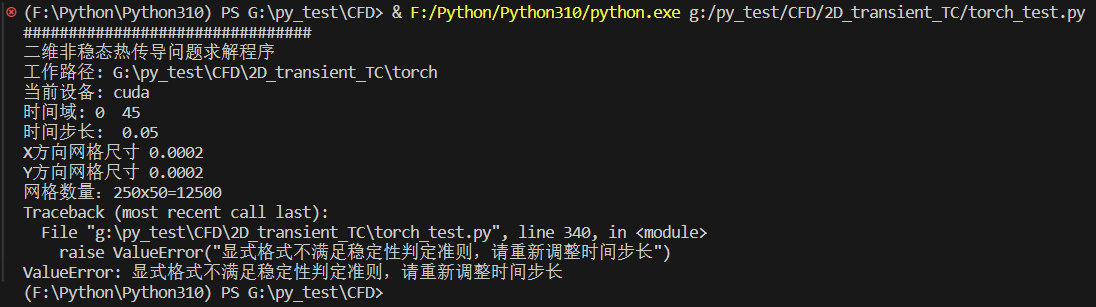
1. C-N格式



1. 显示格式



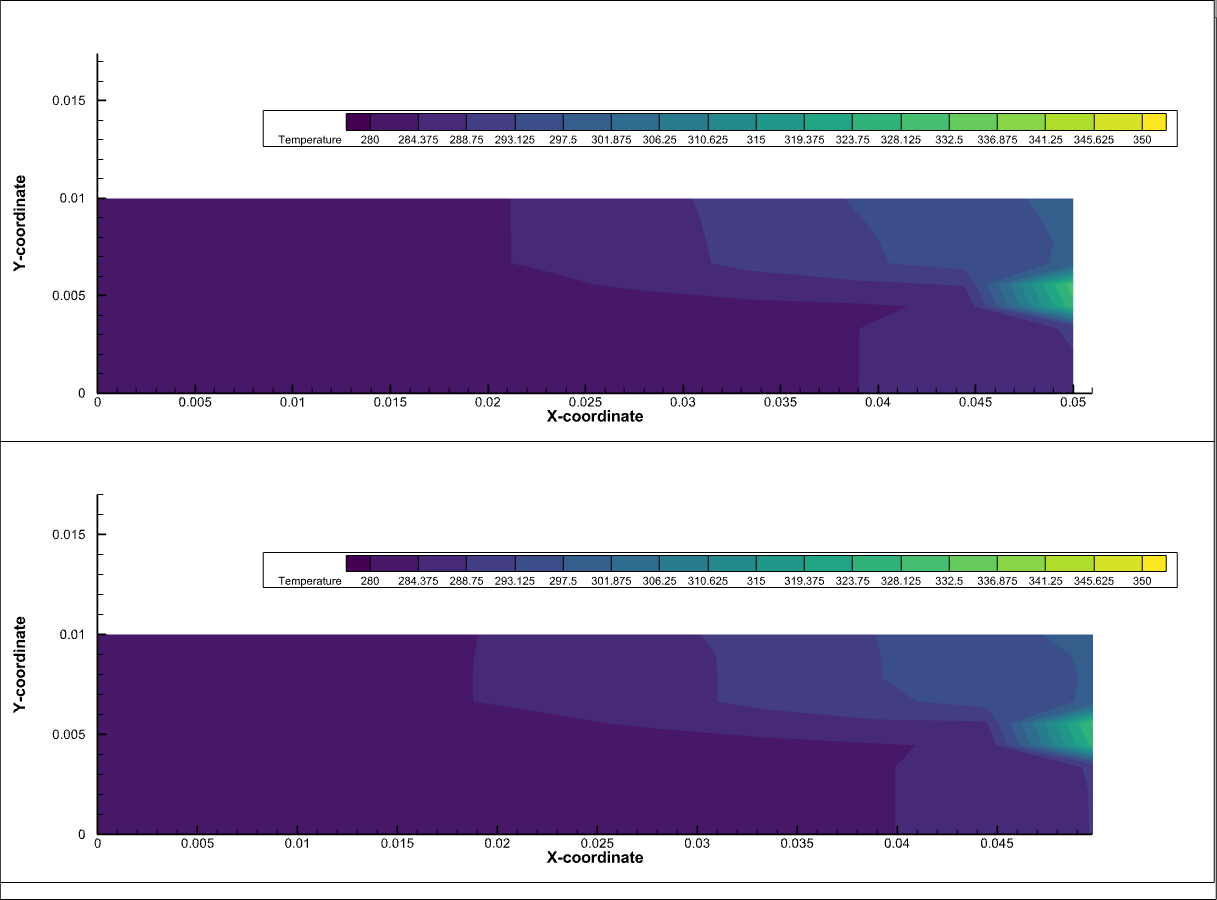
1. 显示格式报错提醒



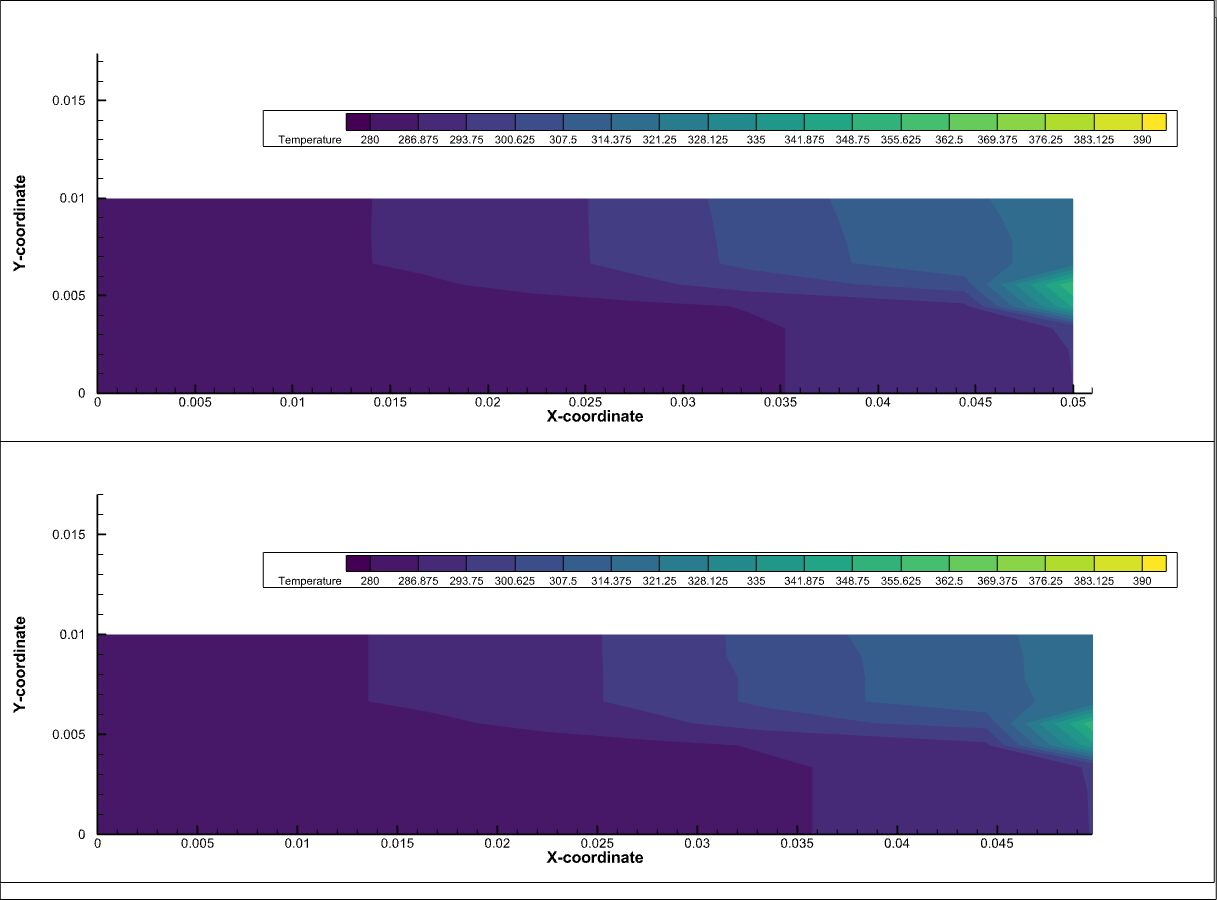
# 计算结果

1.编程计算结果与商业仿真软件结果对比

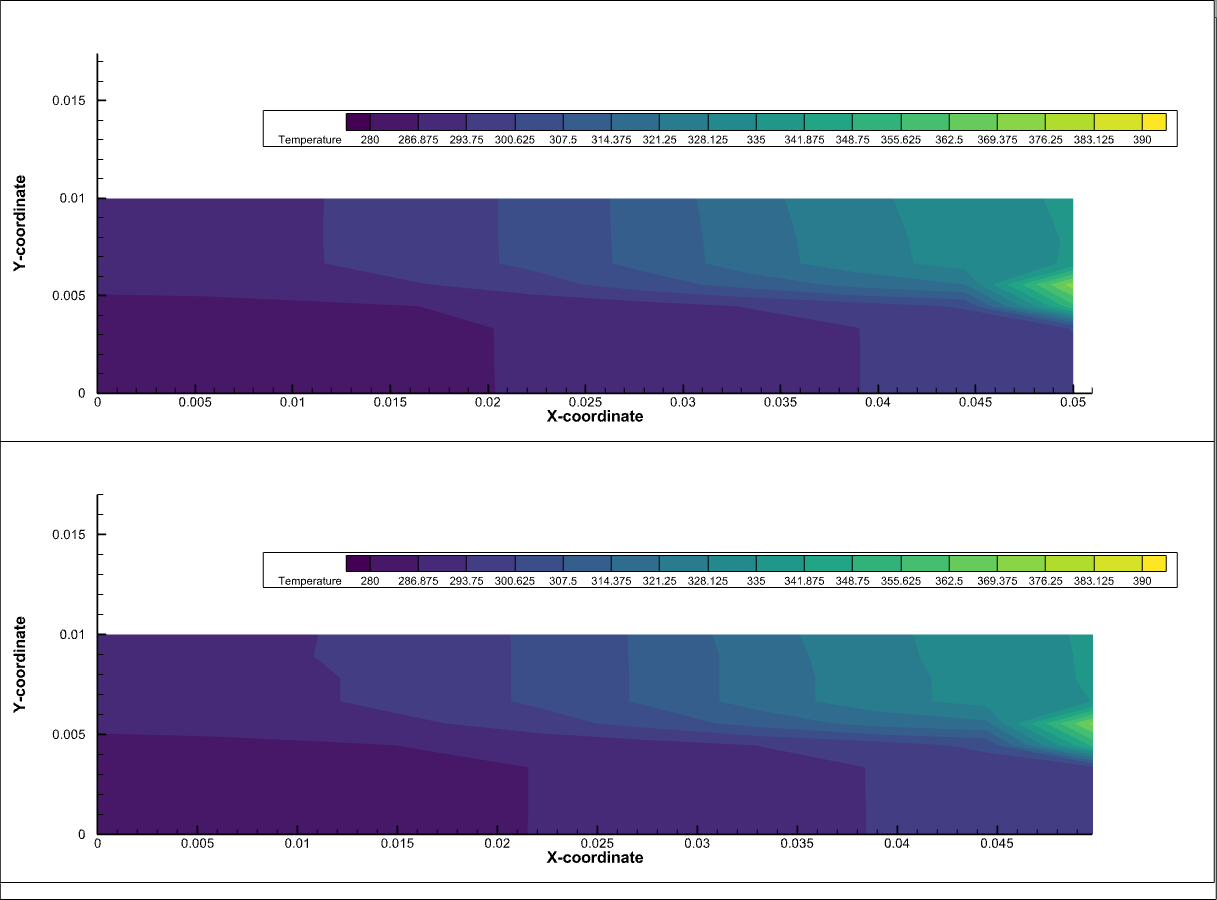
（1）5s时：



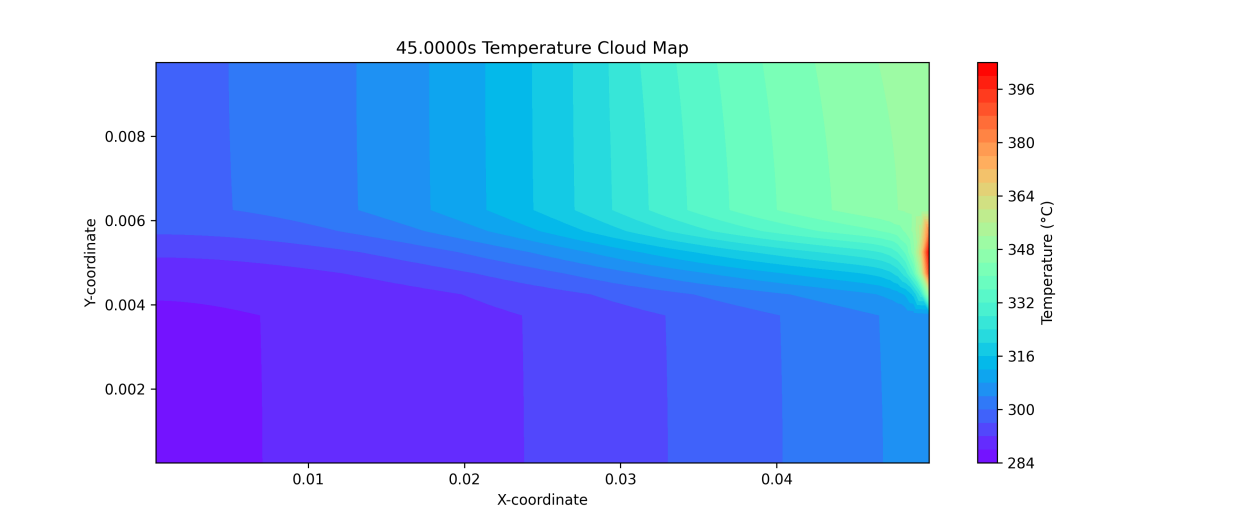
（2）15s时：



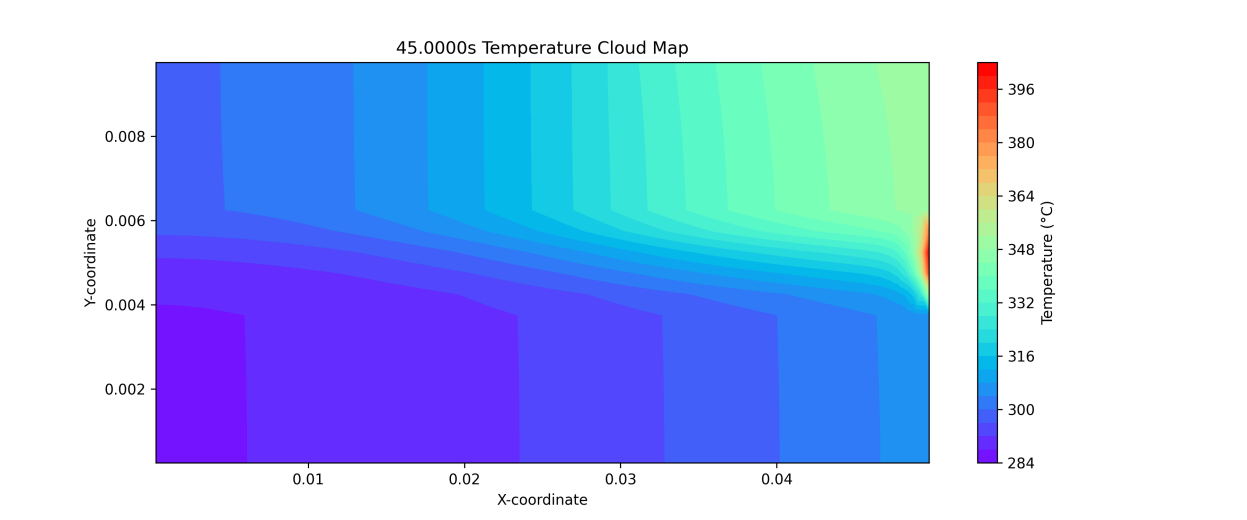
（3）30s时：



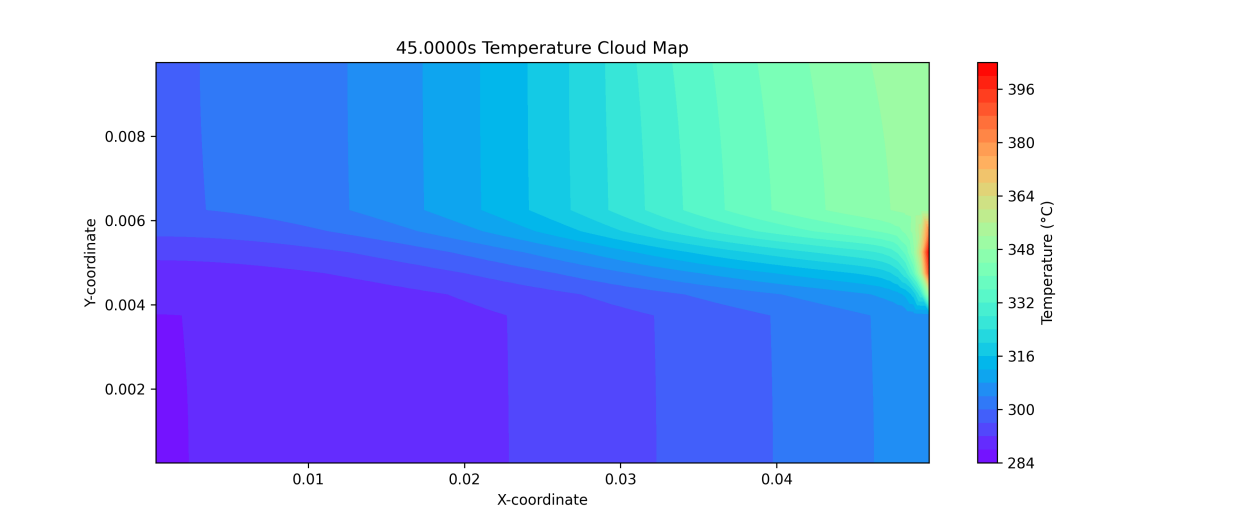
1. 三种时间格式45s结果对比
2. 全隐格式



1. C-N格式

****

1. 显式格式



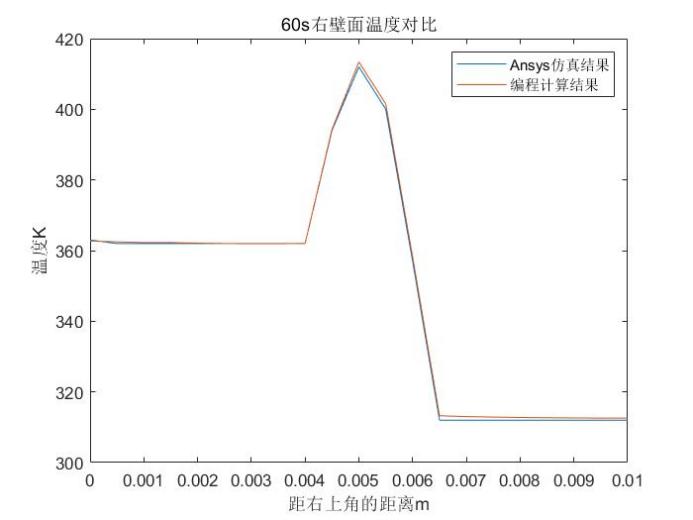
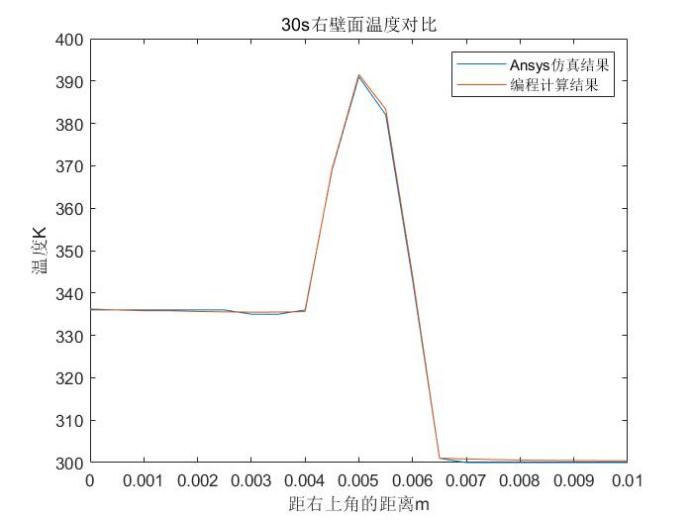
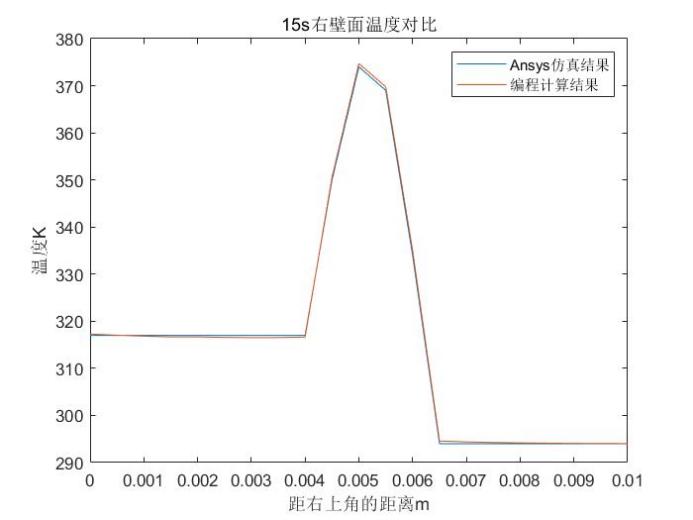
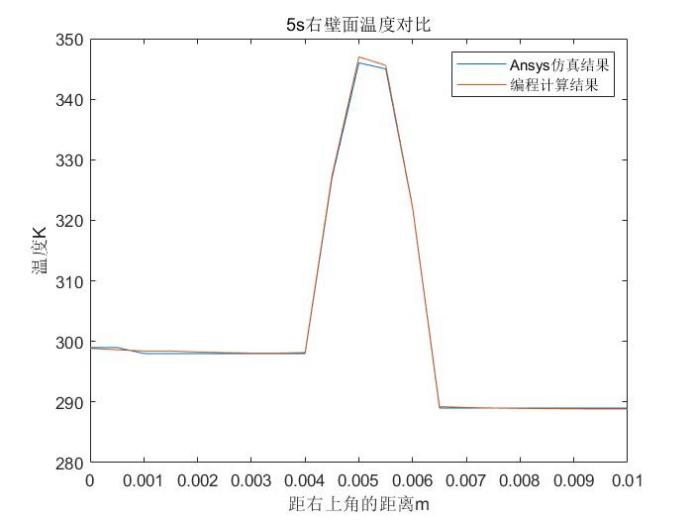
# **结果讨论**

## **1.计算结果合理性分析**

1. 由于材料1和材料2的物性参数不同，材料2的导温系数比材料1小，导致材料2积温，热量不能快速传递出去；
2. 右上角材料1同时受两边高温辐射，导致温度高于其他地方；
3. 材料2上下表面的温度不同，形成了倾斜的等温面；
4. 综上所述，本次计算所得模型大致符合物理规律，基本正确。

## **2.计算与仿真结果对比**

分别以5s，15s，30s，60s各时刻的右壁面网格中心点温度进行对比，可以看出程序计算结果与商业软件计算所得温度基本吻合，材料2处有略微差异，其中相对误差最大处为5s的材料2（0.005m处），为0.28%。



通过对比分析，编程计算结果与商业软件仿真结果基本接近，误差在可接受范围内。

**3.时间格式分析**

同个计算分析发现三种时间格式差异较小，最大差异0.13%，工程应用可以忽略不计。同时，通过分析45s时的云图，不难发现隐式格式的温度发展稍稍晚于显式格式，其中C-N格式计算结果相较全隐格式平均高0.038%，显式格式计算结果相较全隐格式平均高0.124%。

**4.代码优化**

代码优化前使用scipy的linalg.solve计算，一次迭代耗时13秒，优化后使用pytorch的torch.linalg.solve计算，实现1s内迭代10-13次，计算时间大大缩短。

同时使用并行池同时绘制8张图像，可以减少80%的绘图等待时间。

此外，还加入tqdm进度条避免代码运行时，没有相应的漫长等待。

# **结果讨论**

[1]田瑞峰，刘平安. 传热与流体流动的数值计算[M/OL]. 哈尔滨工程大学出版社, 2015[2024-01-14].

[2]李人宪. 有限体积法基础[M/OL]. 国防工业出版社, 2008[2024-01-14].

[3]陶文铨. 传热学：第五版[M/OL]. 高等教育出版社, 2019[2024-01-14].

[4]阿斯顿·张（ASTON ZHANG）, 李沐（MU LI）, [美]扎卡里·C. 立顿（ZACHARY C. LIPTON）, 等. 动手学深度学习[M/OL]. 何孝霆（XIAOTING HE）, 瑞潮儿·胡（RACHEL HU）, 译. PyTorch版. 人民邮电出版社, 2023[2023-07-11].