

超声探测技术在高温固体介质密度检测中的应用

摘要

5.2.2 热传导模型的工作原理

a). 目标方程

热传导模型中的目标函数就是题目中给出的一维热传导方程，需要研究此目标函数中自变量参数之间的关系：

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) \quad \#14$$

在一维热传导方程模型中， ρ 表示的是被测高温固体介质的密度， c 代表被测高温固体介质的比热容数值， k 表示被测高温固体介质的热传导系数，其是固体介质的固有属性，由固体介质决定其具体数值。 T 表示的就是当前高温固体介质中的温度， t 表示的是在这个一维热传导模型，一端加热的加热时间。 x 表示的是一维高温固体介质以加热端为原点的位移距离，即一维高温固体介质中的任意一点到加热原点的距离。

b). 约束条件

根据题目的描述和需求，由题中给出的数据，我们可以给出题目中所要求的约束条件为：

$$\begin{cases} T|_{x=0} = T_0 = 26K \\ -k \frac{\partial T}{\partial x}|_{x=0} = q \\ -k \frac{\partial T}{\partial x}|_{x=L} = 0 \end{cases} \quad 15$$

即起始条件中的一维介质固体的温度是 26K，同时一维热传导模型还应该满足热边界条件，加热条件为 q ，另一端不受热。

5.2.3 超声波运动学模型的工作原理

我们要找出超声波通过材料的时间 τ 与超声波到达材料的时刻 t 的关系。实际上，超声波通过材料需要一段时间，在这段时间内材料上各点的温度仍在变化，但为了方便计算，在以下计算中我们取超声波到达材料的时刻的温度。

通过热传导模型和第一题的波速与温度的数学关系，建立公式，确定超声波的传播时间 dt ，距离一维空间原点 x 距离处的小段距离 dx 和超声波的传播速度之间的关系，公式如下：

$$d\tau = \frac{dx}{v} \quad \#16$$

该公式的求解过程涉及到热传导模型和问题一波速与温度规划模型中的数学关系结论。我们通过问题一中波速和温度之间的数学关系结论，将超声波在公式中的传播速度 v 转化成一个温度变量 T ，

$$d\tau = \frac{dx}{v(T(x, t))} \quad \#17$$

将最终所要求解的超声波传播时间转化为距离一维空间原点 x 距离处的小段距离

dx 和超声波传播速度对应温度之间的关系，如下：

$$\tau(t) = \int_0^L \frac{dx}{v(T(x,t))}$$

5.3 模型的求解

5.3.1 热传导模型的求解与分析

1) 目标函数

我们要找出一维固体介质的每一点的温度关于该点与加热原点的距离、加热时间的函数关系，也即：

$$T(x, t)$$

2) 求解过程

我们采用分离变数法进行求解。

Step1. 构造函数，分离变量。令：

$$T(x, t) = \chi(x)\kappa(t)$$

其中 $\chi(x)$ 和 $\kappa(t)$ 分别是只包含 x 和 t 的函数。如此一来，有：

$$\frac{\partial T(x, t)}{\partial t} = \chi(x)\kappa'(t)$$

$$\frac{\partial T(x, t)}{\partial x} = \chi''(x)\kappa(t)$$

Step2. 代入原方程，得到：

$$\frac{\chi''(x)}{\chi(x)} = \frac{\kappa'(t)}{a^2\kappa(t)} = -\lambda$$

求解得：

$$\begin{cases} \kappa'(t) + \lambda a^2 \kappa(t) = 0 \\ \chi''(x) + \lambda \chi(x) = 0 \end{cases}$$

Step3. 加入边界条件，对 λ 进行分类讨论。

(I) 当 $\lambda \leq 0$ 时：

$$\chi(x) = Ae^{x\sqrt{-\lambda}} + Be^{x\sqrt{-\lambda}}$$

代入边界条件，得到：

$$\begin{cases} A + B = 0 \\ Ae^{l\sqrt{-\lambda}} + Be^{l\sqrt{-\lambda}} = 0 \end{cases}$$

得 $A = B = 0$ ，不符合题意，因此舍去。

(II) 当 $\lambda > 0$ 时，不妨设： $\lambda = \beta^2$ ， $\beta > 0$ ，有：

$$\chi(x) = A\cos\beta x + B\sin\beta x$$

代入边界条件，得到：

$$\begin{cases} A = 0 \\ B\sin\beta l = 0 \end{cases} \quad \beta = \frac{n\pi}{l} \quad (n = 1, 2, 3, \dots)$$

因此，我们可以得到：

$$\chi(x) = B \cdot n \cdot \sin \frac{n\pi}{l} x \quad (n = 1, 2, 3, \dots)$$

Step4. 将 $\chi(x)$ 代入原方程，求出 $\kappa(t)$ ，结果如下：

$$\kappa(t) = c \cdot e^{-\lambda a^2 t}$$

Step5. 将 $\chi(x)$ 和 $\kappa(t)$ 的结果结合起来，得出 $T(x, t)$ ，由于 $T(x, t)$ 的表达式中含有求和下的积分，表达式较为复杂，我们将在下文中显示其数值形式。

3) 热传导模型解析式

通过初始方案策划的步骤流程，一步步利用 Matlab 程序和人力计算出热传导模型的解析式，得到解析式如下：

$$T(x,t) = \frac{|a|}{a^6} \cdot \left(0.00019a^2 e^{-\frac{x^2}{4a^2t}} \sqrt{t}(2a^2t + x^2) \cdot (16a^2t + x^2) + 0.00017x \right. \\ \cdot (60a^4t^2 + 20a^2tx^2 + x^4) \cdot \left(-a + a \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2a\sqrt{t}}} e^{-\eta^2} d\eta \right) \left. + T_0(1 \right. \\ \left. - \frac{0.5a \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2a\sqrt{t}}} e^{-\eta^2} d\eta}{|a|} + 0.5 \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2|a|\sqrt{t}}} e^{-\eta^2} d\eta) \right)$$

其中一维空间位移 x 的取值范围为 $[0,0.05]$ ，加热时间 t 的取值范围为 $[0,500]$ ， T_0 表示起始加热温度， a^2 表示热传导系数除以材料密度和材料比热容，即：

$$a^2 = \frac{k}{\rho c}$$

可以将题目所给的数据带入热传导模型解析式，可以得到温度与位移和加热时间的关系的具体值。

4)结果呈现

图 5-9 呈现的是一个一维高温固体介质随时间变化的热力分布图，所有图的左端位置表示的是原点，即加热端，另一端是绝热端。红色表示的高温；蓝色表示低温。图中的热力线能够清晰的表示温度的明显分布。整个一维空间中，在同一时段，介质的温度变化呈现如图所示的趋势，随位移 x 越大，温度越低。



图 5 一维介质 100s 温度分布热力图



图 6 一维介质 200s 温度分布热力图

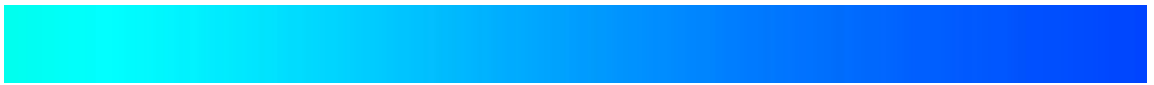


图 7 一维介质 300s 温度分布热力图



图 8 一维介质 400s 温度分布热力图



图 9 一维介质 500s 温度分布热力图

图 5 是 100s 时的分布图，整个一维介质的温度还没有产生明显差异，当加热到 300s 时，可以从图 7 看到，一维介质距离加热段不同位移 x 处的温度有较明显的差异，原点的温度明显高。当加热到 500s 时，每段位移的温度差异非常明显，加热段呈深红色，

表示高温，绝热段呈绿色，温度与 100s 时刻有明显的区分。

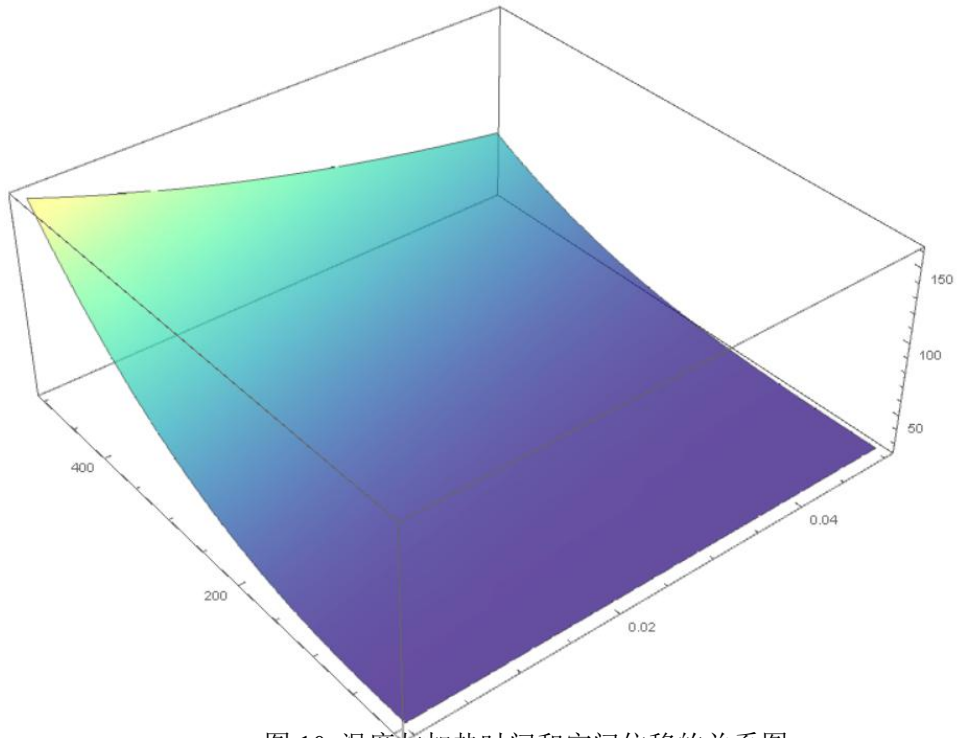


图 10 温度与加热时间和空间位移的关系图

如图 10 所示，呈现出的是一个热传导模型中一维介质解析式的三维坐标图。展现了一维介质中温度与加热时间和一维空间位移之间的关系， x, y 轴表示的是加热时间和一维空间位移， z 轴表示的是一维固体介质的温度，其受加热时间和一维空间位移坐标的影响。

5.3.2 超声波运动学模型的求解

1) 目标函数

$$\tau(t) = \int_0^L \frac{dx}{v(T(x, t))}$$

2) 求解过程

Step1. 将 $T(x, t)$ 代入 $v = v(T)$ ，得：

$$v(T) = \left(\left(0.00019a^2 e^{-\frac{x^2}{4a^2t}} \sqrt{t} (2a^2t + x^2) \cdot (16a^2t + x^2) + 0.00017x \cdot (60a^4t^2 + 20a^2tx^2 + x^4) \right. \right. \\ \cdot \left(-a + a \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2a\sqrt{t}}} e^{-\eta^2} d\eta \right) \left. \right) \frac{1}{a^5} + T_0 \left(1 - \frac{0.5a \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2a\sqrt{t}}} e^{-\eta^2} d\eta}{|a|} \right. \\ \left. \left. + 0.5 \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2|a|\sqrt{t}}} e^{-\eta^2} d\eta \right) \cdot k_1 + b_1 \right)$$

其中 k_1 与 b_1 为第一题中求出的常数，其值如下：

$$\begin{cases} k_1 = -0.4521 \\ b_1 = 3383.391115 \end{cases}$$

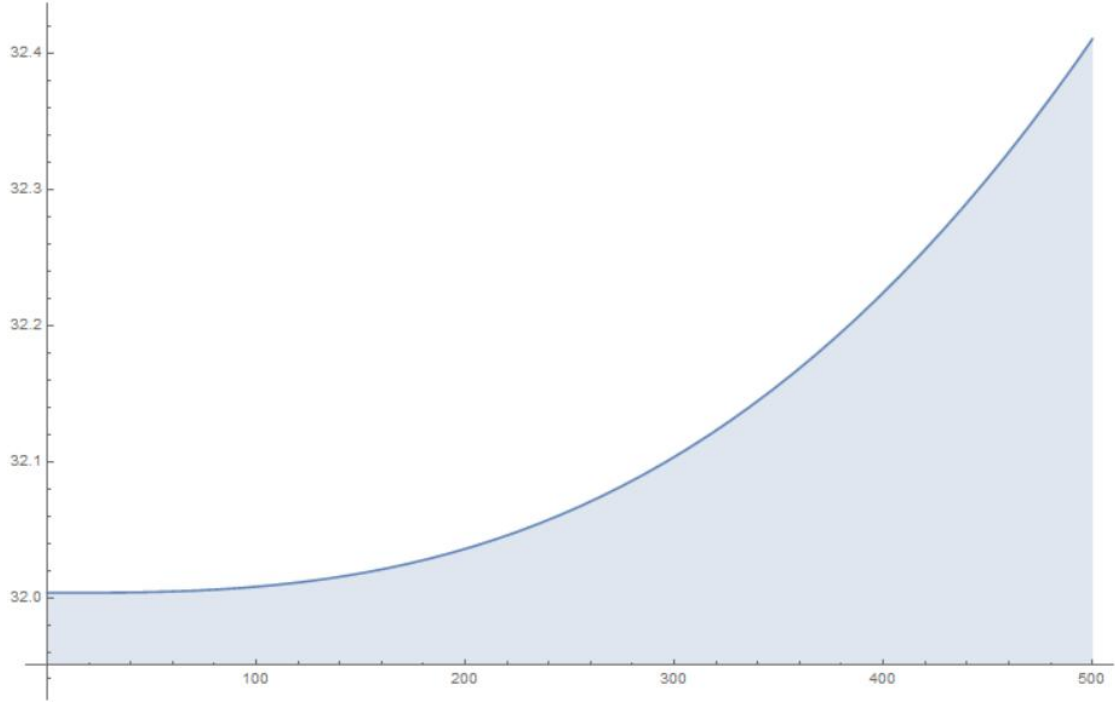
Step2. 将上式求出 $v(T)$ 的代入目标函数，并代入具体数值求积分。

5.3.3 问题二模型的结果

利用求解步骤和方法。我们依据数学解析式和热传导模型以及超声波运动学模型的结果，列出问题二的最终结果解析式，结果公式展现如下：

$$\tau(t) = \int_0^{0.05} \left(2/(3124.65 + 0.004521t^2x + \frac{0.001507tx^3}{a^2} + \frac{0.00007535x^5}{a^4} + \frac{5.8773a \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2a\sqrt{t}}} e^{-\eta^2} d\eta}{|a|} \right. \\ \left. + \frac{|a|}{a^5} \left(ae^{-\frac{x^2}{4a^2t}} \sqrt{t} (-0.002721a^4t^2 - 0.00153a^2tx^2 - 0.00008502x^4) \right. \right. \\ \left. \left. + (-0.004521a^4t^2x - 0.001507a^2tx^3 - 0.00007535x^5) \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2a\sqrt{t}}} e^{-\eta^2} d\eta \right) \right. \\ \left. - 5.8773 \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2|a|\sqrt{t}}} e^{-\eta^2} d\eta \right) dx$$

利用上式，可以做出 $\tau(t)$ 的图像：



上图表示一个函数解析式的坐标曲线。体现出因变量传播时间受自变量 t 影响发生变化。图可以清晰地通过图像展现两者之间的函数关系。

5.3.4 问题三模型的结果与分析

1) 求解过程

从上文的描述中，我们可以得到对于每一个确定密度的材料，在给定材料长度和测温时间的条件下，超声波通过材料学所需的时间是确定的。并且同材料的温度有正相关的关系。

具体来说，超声波通过材料学所需的时间 $\tau(\rho, t)$ 和材料密度 ρ 以及测温时刻具有如下关系：

$$\int_0^{0.05} 2dx \left(C_8 - 0.4521(373.15 - FUNC + C_7 \cdot \text{Erfc}[\sqrt{2}\sqrt{\frac{p}{t}}x] - C_7 \cdot \text{GR}[\frac{1}{2}, \frac{2px^2}{t}]) \right)$$

其中：

$$\begin{aligned} & e^{\frac{2px^2}{t}} \sqrt{t} \left(-\frac{32}{3000} t^2 - 0.048 p t x^2 - \frac{64}{3000} p^2 x^4 \right) \\ & + \sqrt{p} x (C_1 t^2 + C_2 p t x^2 + C_3 p^2 x^4) \\ & + \sqrt{p} x (C_4 t^2 + C_5 p t x^2 + C_6 p^2 x^4) \cdot \text{Erf}[\sqrt{2}\sqrt{\frac{p}{t}}x] \\ & FUNC = \frac{\quad}{2\sqrt{p}\sqrt{2\pi}} \end{aligned}$$

值得注意的是，式中的 Erf 函数和 Erfc 分别表示高斯误差函数和误差互补函数，具有如下关系：

$$\begin{aligned} & \text{Erf} + \text{Erfc} = 1 \\ & \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\eta^2} d\eta. \end{aligned}$$

式中 GR 表示 GammaRegularized 函数，表示正则不完全伽玛函数。

代入具体数据后，常数项如下：

$$\begin{cases} C_1 = 0.050132565492620004 \\ C_2 = 0.13368684131365333 \\ C_3 = 0.053474736525461336 \\ C_4 = -0.050132565492620004 \\ C_5 = -0.13368684131365333 \\ C_6 = -0.053474736525461336 \\ C_7 = 186.575 \\ C_8 = 3383.391115 \end{cases}$$

通过电脑绘制函数图像，我们可以发现，超声波通过材料所需的时间和材料密度的函数关系具有单调性。应此我们可以通过二分法求解。由于每一时刻对应的密度可能有所不同，我们对结果与平均值，作为我们的答案。

2)结果分析

通过上述方法，我们求得了在每一个时刻对应的密度。

时刻	对应密度 (kg/m ³)
1	1.1284450E-09
2	3.5365000E-08
3	2.6320000E-07
4	1.0876000E-06
5	3.2550000E-06
6	7.9450000E-06
7	1.6849000E-05
8	3.2237000E-05
9	5.7015000E-05
10	9.4786000E-05
平均	2.13474E-05

求得该材料的密度为 $2.1347 \times 10^{-5} \text{kg/m}^3$ 。分析得出，该答案与模型实际情况有较大差距。

进一步分析数据，可以发现数据本身有一定问题。以第一秒内的情况为例。我们用下式计算出整根材料获得的热量：

$$Q = \int_0^1 0.01t^2 dt = \frac{1}{300} J$$

而根据

$$\tau(t) = \int_0^L \frac{dx}{v(T(x,t))}$$

为了让 $\tau = 31.3867821492946 \mu\text{s}$ ，我们需要让材料的平均温度提升到约 438K 左右，提升约 70K 。如果用如此少的热量就可以将材料的平均温度大幅度提高，显然，解出的材料密度会非常小。

3)问题重构

由于上述结果与实际相违背，我们将热传导时间进行扩展，在经过多次拟合后，发现在 $[300\text{s}, 600\text{s}]$ 的加热时间区间内，存在既符合题给数据，又符合客观现实的介质材料密度。我们将以 $[400\text{s}, 410\text{s}]$ 和 $[500\text{s}, 510\text{s}]$ 的加热区间为例，给出解。

4)我们的结果

六、模型的综合评价和推广

6.1 模型的综合评价

6.1.1 模型的优点

本文在

本文在。

本文在。

6.1.2 模型的缺点分析

本文。

。

6.2 模型的推广

该题目中的模型在实际的传热学问题中实用性较强，考虑的因素全面，能够应对多元复杂的环境。

热传导模型中，考虑的因素很具体、丰富，在考虑第二热边界条件的影响下，求出一维介质中的热传导方程中的解析解。并且该模型具有很强的应用性，能够直接将最终的数据代入考虑约束条件下的函数解析表达式，计算实际热力学中需要的指标或者影响因素。

超声波运动学模型中，考虑到的因素也符合现代物理学和实际生活的需求、限制。在众多影响因素下，建立了满足物理学逻辑和数学逻辑的函数表达式解析解。并且此模型也具有很强的应用能力，考虑到了很多现实中存在的影响因素。此外，该模型的适应能力也很好。适用范围广泛，最终得到的函数表达式十分简洁、明了，可以直接将其带入热传导模型等具有现实意义的物理模型公式中，研究超声波运动学和整个热力学之间的关系。

整个数学模型和代码计算过程体系完整、严密，逻辑性强，对于热力学领域的研究具有重要的作用和参考价值，满足当今社会的发展趋势，能够为工业生产的高效发展和进步贡献一定的力量。

八、参考文献

附录

附录 1： 模拟温度序列.cpp

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
double f(double x){
    return x*x*x-6*x*x+9*x;
}
int main(){
    double a;
    freopen("T.txt","w",stdout);
    for(a=0;a<5;a+=0.01){
        printf("%.3lf\t",f(a));
    }
```



```

    return 0;
}

```

附录 2: 差分法.cpp

```

#include<bits/stdc++.h>
using namespace std;
typedef long long ll;

double p,c=400,k=50;
double Dt=0.001,Dx=0.0001;    //长度和时间的拆分间隔
double limx=0.05,limt=10;    //长度和时间的限制条件
double T[5000];                //记录当前的温度
double q[5000];                //记录当前的热流
double DT[5000];                //记录当前温度变化量

double T0=373.15;

int main(){

    freopen("div1.txt","r",stdin);
    freopen("div2.txt","w",stdout);

    int xI=limx/Dx;            // 500
    int tI=limt/Dt;            // 10000

    //初始化
    for(int i=0;i<=xI;i++){
        T[i]=T0;
        q[i]=0;
        DT[i]=0;
    }

    while(~scanf("%lf",&p)){

        for(double t=0;t<=limt;t+=Dt){
            // 计算热流
            q[0]= (1/300)*( pow(t+Dt,3)-pow(t,3) );
            for(int i=1;i<xI;i++){    //1-499
                q[i]=( T[i-1]-T[i] )/Dx*k;
            }
            q[500]=0;

            //计算温度变化量
            for(int i=0;i<xI;i++){
                DT[i]=( q[i]-q[i+1] )/Dx/p/c*Dt;
            }

            //计算迭代温度
            for(int i=0;i<xI;i++){

```

```

        T[i]=T[i]+DT[i];
    }
}
double sumT=0;
for(int i=0;i<xI;i++){
    double v=(-0.4521)*T[i]+3383.391115;
    sumT+=Dx/v;
}
sumT*=2000000;
printf("%.10lf\t\t%.5lf\n",p,sumT);

}

return 0;
}

```

附录 3： 问题 1 制图.py

```

import matplotlib.pyplot as plt
import numpy as np

x =
np.array([300.00000000, 301.00000000, 302.00000000, 303.00000000, 304.00000000,
305.00000000, 306.00000000, 307.00000000, 308.00000000, 309.00000000, 310.000000
00, 311.00000000, 312.00000000, 313.00000000, 314.00000000, 315.00000000, 316.000
00000, 317.00000000, 318.00000000, 319.00000000, 320.00000000, 321.00000000, 322.
00000000, 323.00000000, 324.00000000, 325.00000000, 326.00000000, 327.00000000, 3
28.00000000, 329.00000000, 330.00000000, 331.00000000, 332.00000000, 333.0000000
0, 334.00000000, 335.00000000, 336.00000000, 337.00000000, 338.00000000, 339.0000
0000, 340.00000000, 341.00000000, 342.00000000, 343.00000000, 344.00000000, 345.0
0000000, 346.00000000, 347.00000000, 348.00000000, 349.00000000, 350.00000000, 35
1.00000000, 352.00000000, 353.00000000, 354.00000000, 355.00000000, 356.00000000,
357.00000000, 358.00000000, 359.00000000, 360.00000000, 361.00000000, 362.000000
00, 363.00000000, 364.00000000, 365.00000000, 366.00000000, 367.00000000, 368.000
00000, 369.00000000, 370.00000000, 371.00000000, 372.00000000, 373.00000000, 374.
00000000, 375.00000000, 376.00000000, 377.00000000, 378.00000000, 379.00000000, 3
80.00000000, 381.00000000, 382.00000000, 383.00000000, 384.00000000, 385.0000000
0, 386.00000000, 387.00000000, 388.00000000, 389.00000000, 390.00000000, 391.0000
0000, 392.00000000, 393.00000000, 394.00000000, 395.00000000, 396.00000000, 397.0
0000000, 398.00000000, 399.00000000, 400.00000000, 401.00000000, 402.00000000, 40
3.00000000, 404.00000000, 405.00000000, 406.00000000, 407.00000000, 408.00000000,
409.00000000, 410.00000000, 411.00000000, 412.00000000, 413.00000000, 414.000000
00, 415.00000000, 416.00000000, 417.00000000, 418.00000000, 419.00000000, 420.000
00000, 421.00000000, 422.00000000, 423.00000000, 424.00000000, 425.00000000, 426.
00000000, 427.00000000, 428.00000000, 429.00000000, 430.00000000, 431.00000000, 4
32.00000000, 433.00000000, 434.00000000, 435.00000000, 436.00000000, 437.00000000

```

0, 438. 00000000, 439. 00000000, 440. 00000000, 441. 00000000, 442. 00000000, 443. 0000
0000, 444. 00000000, 445. 00000000, 446. 00000000, 447. 00000000, 448. 00000000, 449. 0
0000000, 450. 00000000, 451. 00000000, 452. 00000000, 453. 00000000, 454. 00000000, 45
5. 00000000, 456. 00000000, 457. 00000000, 458. 00000000, 459. 00000000, 460. 00000000,
461. 00000000, 462. 00000000, 463. 00000000, 464. 00000000, 465. 00000000, 466. 000000
00, 467. 00000000, 468. 00000000, 469. 00000000, 470. 00000000, 471. 00000000, 472. 000
00000, 473. 00000000, 474. 00000000, 475. 00000000, 476. 00000000, 477. 00000000, 478.
00000000, 479. 00000000, 480. 00000000, 481. 00000000, 482. 00000000, 483. 00000000, 4
84. 00000000, 485. 00000000, 486. 00000000, 487. 00000000, 488. 00000000, 489. 00000000
0, 490. 00000000, 491. 00000000, 492. 00000000, 493. 00000000, 494. 00000000, 495. 0000
0000, 496. 00000000, 497. 00000000, 498. 00000000, 499. 00000000, 500. 00000000]])

y =

np. array([900. 00000000, 906. 01000000, 912. 04000000, 918. 09000000, 924. 16000000,
930. 25000000, 936. 36000000, 942. 49000000, 948. 64000000, 954. 81000000, 961. 000000
00, 967. 21000000, 973. 44000000, 979. 69000000, 985. 96000000, 992. 25000000, 998. 560
00000, 1004. 89000000, 1011. 24000000, 1017. 61000000, 1024. 00000000, 1030. 41000000,
1036. 84000000, 1043. 29000000, 1049. 76000000, 1056. 25000000, 1062. 76000000, 1069.
29000000, 1075. 84000000, 1082. 41000000, 1089. 00000000, 1095. 61000000, 1102. 24000
000, 1108. 89000000, 1115. 56000000, 1122. 25000000, 1128. 96000000, 1135. 69000000, 1
142. 44000000, 1149. 21000000, 1156. 00000000, 1162. 81000000, 1169. 64000000, 1176. 4
9000000, 1183. 36000000, 1190. 25000000, 1197. 16000000, 1204. 09000000, 1211. 040000
00, 1218. 01000000, 1225. 00000000, 1232. 01000000, 1239. 04000000, 1246. 09000000, 12
53. 16000000, 1260. 25000000, 1267. 36000000, 1274. 49000000, 1281. 64000000, 1288. 81
000000, 1296. 00000000, 1303. 21000000, 1310. 44000000, 1317. 69000000, 1324. 96000000
0, 1332. 25000000, 1339. 56000000, 1346. 89000000, 1354. 24000000, 1361. 61000000, 136
9. 00000000, 1376. 41000000, 1383. 84000000, 1391. 29000000, 1398. 76000000, 1406. 250
00000, 1413. 76000000, 1421. 29000000, 1428. 84000000, 1436. 41000000, 1444. 00000000,
1451. 61000000, 1459. 24000000, 1466. 89000000, 1474. 56000000, 1482. 25000000, 1489.
96000000, 1497. 69000000, 1505. 44000000, 1513. 21000000, 1521. 00000000, 1528. 81000
000, 1536. 64000000, 1544. 49000000, 1552. 36000000, 1560. 25000000, 1568. 16000000, 1
576. 09000000, 1584. 04000000, 1592. 01000000, 1600. 00000000, 1608. 01000000, 1616. 0
4000000, 1624. 09000000, 1632. 16000000, 1640. 25000000, 1648. 36000000, 1656. 490000
00, 1664. 64000000, 1672. 81000000, 1681. 00000000, 1689. 21000000, 1697. 44000000, 17
05. 69000000, 1713. 96000000, 1722. 25000000, 1730. 56000000, 1738. 89000000, 1747. 24
000000, 1755. 61000000, 1764. 00000000, 1772. 41000000, 1780. 84000000, 1789. 29000000
0, 1797. 76000000, 1806. 25000000, 1814. 76000000, 1823. 29000000, 1831. 84000000, 184
0. 41000000, 1849. 00000000, 1857. 61000000, 1866. 24000000, 1874. 89000000, 1883. 560
00000, 1892. 25000000, 1900. 96000000, 1909. 69000000, 1918. 44000000, 1927. 21000000,
1936. 00000000, 1944. 81000000, 1953. 64000000, 1962. 49000000, 1971. 36000000, 1980.
25000000, 1989. 16000000, 1998. 09000000, 2007. 04000000, 2016. 01000000, 2025. 00000
000, 2034. 01000000, 2043. 04000000, 2052. 09000000, 2061. 16000000, 2070. 25000000, 2
079. 36000000, 2088. 49000000, 2097. 64000000, 2106. 81000000, 2116. 00000000, 2125. 2
1000000, 2134. 44000000, 2143. 69000000, 2152. 96000000, 2162. 25000000, 2171. 560000
00, 2180. 89000000, 2190. 24000000, 2199. 61000000, 2209. 00000000, 2218. 41000000, 22

```
27. 84000000, 2237. 29000000, 2246. 76000000, 2256. 25000000, 2265. 76000000, 2275. 29
000000, 2284. 84000000, 2294. 41000000, 2304. 00000000, 2313. 61000000, 2323. 2400000
0, 2332. 89000000, 2342. 56000000, 2352. 25000000, 2361. 96000000, 2371. 69000000, 238
1. 44000000, 2391. 21000000, 2401. 00000000, 2410. 81000000, 2420. 64000000, 2430. 490
00000, 2440. 36000000, 2450. 25000000, 2460. 16000000, 2470. 09000000, 2480. 04000000,
2490. 01000000, 2500. 00000000])
```

```
plt.plot(x, y, ls="--", lw=2, label="Heat flow")
plt.legend()
plt.show()
```

附录 4: 红外温度图-Copy1.ipynb

```
{
  "cells": [
    {
      "cell_type": "code",
      "execution_count": 1,
      "metadata": {},
      "outputs": [
        {
          "data": {
            "text/plain": [
              "array([[ 0,  0, 255],\n",
              "       [ 0,  0, 255],\n",
              "       [ 0,  1, 255],\n",
              "       ..., \n",
              "       [255,  1,  0],\n",
              "       [255,  0,  0],\n",
              "       [255,  0,  0]])"

```

```

]
},
{
  "cell_type": "code",
  "execution_count": 2,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "array([255,    0,    0])"
        ]
      },
      "execution_count": 2,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "data1[796]"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {},
  "outputs": [],
  "source": []
},
{
  "cell_type": "markdown",
  "metadata": {},
  "source": [
    "0-796"
  ]
},
{
  "cell_type": "code",
  "execution_count": 50,
  "metadata": {},
  "outputs": [],
  "source": [
    "# 读入温度数据（一维）\n",
    "map_data = np.loadtxt(\"200s.txt\")\n",

```

```

    "map_data = map_data.tolist()\n"
]
},
{
    "cell_type": "code",
    "execution_count": 51,
    "metadata": {},
    "outputs": [
        {
            "data": {
                "text/plain": [
                    "501"
                ]
            },
            "execution_count": 51,
            "metadata": {},
            "output_type": "execute_result"
        }
    ],
    "source": [
        "n=np. size(map_data,0)\n",
        "n"
    ]
},
{
    "cell_type": "code",
    "execution_count": 52,
    "metadata": {},
    "outputs": [],
    "source": [
        "for ii in range(n):\n",
        "    map_data[ii]=( map_data[ii]-26 )/135"
    ]
},
{
    "cell_type": "code",
    "execution_count": 53,
    "metadata": {},
    "outputs": [],
    "source": [
        "w=50\n"
    ]
},
{

```

```

"cell_type": "code",
"execution_count": 54,
"metadata": {},
"outputs": [],
"source": [
    "im = Image.open('type1.png') #打开图片\n",
    "pix = im.load() #导入像素\n",
    "width = im.size[0] #获取宽度\n",
    "height = im.size[1] #获取长度\n"
]
},
{
"cell_type": "code",
"execution_count": 55,
"metadata": {},
"outputs": [],
"source": [
    "tmp1=np.zeros(n)\n",
    "for j in range(n):\n",
    "    value = map_data[j]*796\n",
    "    v1=int(value)\n",
    "    tmp1[j]=v1"
]
},
{
"cell_type": "code",
"execution_count": 56,
"metadata": {},
"outputs": [],
"source": [
    "tmp1=tmp1.astype(np.int16) "
]
},
{
"cell_type": "code",
"execution_count": 57,
"metadata": {},
"outputs": [],
"source": [
    "for j in range(n):\n",
    "    r=data1[tmp1[j],0]\n",
    "    g=data1[tmp1[j],1]\n",
    "    b=data1[tmp1[j],2]\n",
    "    for i in range(w):\n",

```

```

        "        im.putpixel((j,i),(r,g,b))\n",
        "        # im.putpixel(i,j)第 i 列第 j 行"
    ]
},
{
    "cell_type": "code",
    "execution_count": 58,
    "metadata": {},
    "outputs": [],
    "source": [
        "im.save('200s-.png')"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {},
    "outputs": [],
    "source": []
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {},
    "outputs": [],
    "source": []
}
],
"metadata": {
    "kernelspec": {
        "display_name": "Python 3",
        "language": "python",
        "name": "python3"
    },
    "language_info": {
        "codemirror_mode": {
            "name": "ipython",
            "version": 3
        },
        "file_extension": ".py",
        "mimetype": "text/x-python",
        "name": "python",
        "nbconvert_exporter": "python",
        "pygments_lexer": "ipython3",

```



```
    "version": "3.8.5"  
  }  
},  
  "nbformat": 4,  
  "nbformat_minor": 4  
}
```