

超声探测技术在高温固体介质密度检测中的应用

摘要

超声波检测技术因检测灵敏度高、速度快，在现代社会生活中已获得广泛应用。但同时，超声波检测技术使用难度较大、且易受主观因素影响的缺点，启发广大人员对其进行研究。本文定性地阐述了超声波检测技术的运行机制，并定量地给出了运行过程中关键物理量的精确解析解。

问题一需要给出波速与温度之间的关系、边界热流随时间变化的规律，并对热流进行预测。首先，根据题目所给的波速与温度场数据、边界热流关于时间的数据，分析其特征与已知信息，预测其函数类型。再对数据进行拟合，得到波速关于温度的函数为 $v(t) = -0.4521 \times t + 3383.391115$ ，以及边界热流关于时间的函数 $q(t) = 0.01t^2$ 。依据残差为零这一条件可知，拟合所得函数无误差。最后通过所得函数预测在[300:500]s的边界热流值。

对于问题二，以非稳态下一维热传导的传热公式、初始温度条件与边界热流条件为基础，通过分离变数法，求解出随时间变化的温度场在空间的分布，即获得在第一类边界条件与第二类边界条件下的二阶偏微分方程的解析解。再利用运动学规律与波速关于温度的函数的关系，获得超声波所需传播时间与热传导时间的解析解。将附件 4 的条件带入后获得问题二的最终结果。结果显示，在 300 秒时，在 $x=0.0275$ 处的温度为 44.9781K。

对于问题三，将材料密度作为未知量，重复问题二的求解过程，获得超声波传播时间关于材料密度与热传导时间的函数解析方程。代入附件 6 的数据获得材料密度为 $5.60 \times 10^{-6} kg/m^3$ 逻辑分析，本文认为所得结果与现实情况不符，故进行了问题重构，将数据进行合理拓展后，得到新的材料密度在加热时间为 300 秒到 600 秒时存在合理解。在文中我们举例说明了合理解的数值和意义。

关键词： 热传导模型 超声波运动学模型 密度检测

一、问题重述

1.1 问题背景

超声波检测在现代工业上使用范围十分广泛，是利用超声波技术对被检验部件的表面和内部质量进行检查的一种检测手段。作为五种常规无损检测方法中的一种，它具有检测厚度大、灵敏度高、速度快等优点。同时由于超声波对缺陷显示效果不直观，难度较大，很容易受检验人员主观判断的影响，因此其使用需要由经验较为丰富的科研人员来进行检验。

超声波检测是一项被较多科研人员掌握的基本技术。它还由界面特性、投射特性评价层状介质性能、超声波散射现象等因素分为了多个不同方法。本题给出了高温固体介质，一段加热，另一端不加热情况下，满足的一维热传导方程：

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) \quad (1)$$

热边界条件：

$$-k \frac{\partial T}{\partial x} \Big|_{x=0} = q, -k \frac{\partial T}{\partial x} \Big|_{x=L} = 0 \quad (2)$$

加热初始条件：

$$T|_{t=0} = T_0 \quad (3)$$

以及附件中给出的温度场、波速、材料导热关系、比热、加热时间等具体数据。要求我们研究波速与温度、热流与时间、超声波传播时间与测量时间对材料密度的影响等问题。

1.2 问题提出

问题一：根据超声波在介质中的传播规律，在题目给出的波速和温度场数据基础上探究波速和温度之间的关系；通过题目给出的 300s 边界热流数据，寻找热流和时间的变化规律，对后 200s 的热流进行预测。

问题二：在问题一的基础上，将运动学规律和波速与温度关系相结合，探寻合理的数学规律和关系，建立超声波传播时间在固体介质中的模型，并对题目中所给出的已知数据求解。

问题三：在问题二的基础上，给出材料的热边界条件，长度，导热系数，加热时间和测量的超声波传播时间，寻找合理的数学关系，建立合适的模型，用问题二的传播时间结果和测量的传播时间求材料密度。

二、问题分析

2.1 问题一的分析

问题一首先需要根据题目给出的已知信息和超声波在介质中的传播规律，求解出波速和时间关系的数学模型；其次需要通过给出的边界热流数据求解热流与时间的规律，并对之后 200s 时间进行预测。

我们考虑的是利用已知信息，通过多项式拟合的方式分别求解一维空间中位移和温度以及位移和波速之间的两个关系。根据一维空间位移的绝对性，以因变量连接波速和温度，建立模型，得到波速和温度之间的关系；对给出的热流和时间信息做多项式拟合，得到一个热流随时间的变化规律模型和以时间为自变量，边界热流数据为因变量的结果表达式，通过结果表达式求解出对热流边界数据在未来 200s 时间内的预测，并将结果绘图呈现。

2.2 问题二的分析

问题二考虑的是超声波在固体介质中的传播时间，通过题目给出的已知数据和需要求解的问题，我们建立了热传导模型和超声波运动学模型，利用给出的已知条件，求解出两个模型的解析解，最后带入题目的数据信息，计算得到最终需要的结果。

我们考虑的是通过热传导模型和方程式，求解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、加热时间的数学关系。再根据问题一中得到的超声波传播速度与温度的关系，进一步找出超声波在介质中的每一点的传播速度关于该点与加热原点的距离以及加热时间的数学关系。最后再代入具体数值，用数值积分法求出超声波通过的时间。

2.3 问题三的分析

问题三考虑的也是通过题目中给出大量已知数据和已知条件，利用测量的传播时间和问题二中计算出来的传播时间，对材料的密度进行求解。其核心思想与问题二相同，即通过计算热传导模型和超声波运动学模型的解析解，找出其内在的逻辑关系。

考虑到超声波在介质中的每一点的传播速度关于该材料的密度，该点与加热原点的距离以及加热时间的数学关系较为复杂，我们拟采用二分法找出材料的密度。

三、模型假设

1. 在管壁模型简化为一维介质处理后，假设仅有管内和管两端的热传导起作用，不考虑热辐射、热对流作用。
2. 假设温度的改变不会对材料的体积产生影响，即不考虑热胀冷缩的因素。
3. 假设超声波传导时间，此时间段内温度的改变可以不予考虑。

四、符号说明

| 符号 | 符号意义 |
|-------------|-------------------------|
| T | 表示一维介质某一点的温度 |
| t | 表示一维介质加热时间 |
| ρ | 表示被测介质物体的密度 |
| c | 表示被测介质物体的比热容 |
| k | 表示被测介质物体的热传导系数 |
| q | 表示热流函数 |
| $f(x)$ | 表示波速和温度之间的目标函数 |
| S | 表示热流随时间变化的函数表达式 |
| τ | 表示超声波的传播时间 |
| v | 表示超声波在某一点的传播速度 |
| $\chi(x)$ | 表示对 x 分离变量后关于 x 的函数 |
| $\kappa(t)$ | 表示对 t 分离变量后关于 t 的函数 |

| | |
|-------|----------------------|
| a^2 | 表示热传导系数与材料密度和材料比热容的比 |
| T_0 | 表示一维介质开始时的温度 |
| x | 表示一维介质某点相对加热端的距离 |
| L | 表示材料的长度 |
| Q | 表示材料获得的热量 |

注：未列出符号及重复的符号以出现处为准

五、模型的建立与求解

5.1 问题一的模型建立与求解

根据问题一的需要，我们将本题的基本模型化简为拟合规划模型和预测模型。建立规划模型和预测模型，并带入问题一所给的数据，得到最终问题所需要的计算结果和结论。

5.1.1 规划模型

1) 目标函数的确定

根据问题的要求，可以知道规划模型的目标函数就是波速和温度之间的关系表达式，建立表示两者间表达式的目标函数：

$$f(x) \quad (4)$$

x 表示公式中的自变量。

2) 方案设定

Step1. 用最小二乘法 and 题目中所给出的大量已知数据，拟合出波速和一维空间位移的关系表达式，温度和一维空间位移的关系。得到

$$f(x_1) = a_i x_i \quad (5)$$

$$f(x_2) = b_i x_i \quad (6)$$

其中公式(5)表示利用最小二乘法求解出的波速和位移之间的函数表达式；公式 10 表示求解得到的温度和位移之间的函数表达式。

Step2. 将两个函数表达式的因变量和自变量互换，转化表达式的形式，同时消去新的因变量 x_i ，可以得到一个新的表达式关系，是关于 $f(x_1)$ 和 $f(x_2)$ 之间的关系，即波速和温度之间的关系。

Step3. 构建新的表达式，可以得到：

$$k_1 f(x_1) = k_2 f(x_2) \quad (7)$$

新的表达式可以通过化简，部分消元，得到最终的最简形式，即 $f(x)$ 。

最终表达式不仅可以从逻辑上进行验证波速与温度之间存在的自然界的客观规律，并且该式还可以直观地表现波速与温度之间的关系，较为简洁、清晰地探究其内在存在的影响因素和关系。

5.1.2 预测模型

1) 目标函数的确定

根据题目要求，我们需要根据题目给出的边界热流数据，得到热流随时间变化的规律。

2) 方案设定

Step1. 数据拟合, 计算所有多项式函数的表达式, 绘制数据的散点图。大致确定所有拟合多项式的次数 n , 以此确认所有多项式函数。如图所示, 是热流与时间在 $[0, 300s]$ 范围内的散点图。

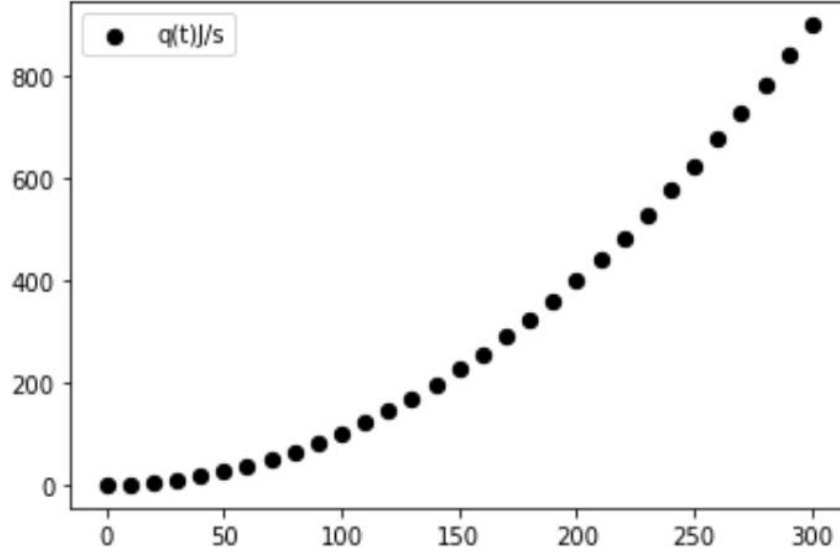


图 1 热流与时间的散点分布图

Step2. 通过拟合多项式的次数 n , 以及题目给出的时间范围在 $[0, 300s]$ 中热流随时间变化规律的数据, 计算 $\sum_{i=1}^N x_i^j$ 和 $\sum_{i=1}^N x_i^j y_i$ 。并建立多项式系数方程式, 通过前面计算出的 $\sum_{i=1}^N x_i^j$ 和 $\sum_{i=1}^N x_i^j y_i$ 求解多项式的系数。

Step3. 根据工作原理中最小二乘法的具体描述, 求解拟合多项式 $p_n(x) = \sum_{k=0}^n a_k x^k$, 其中的拟合多项式系数 a_k 可以通过 *Step2*来确认。建立目标函数和求解拟合多项式之间的关系:

$$S = p_n(x) = \sum_{k=0}^n a_k x^k \quad (8)$$

Step4. 将时间的数值范围由 $[0, 300s]$ 扩展到 $[0, 500s]$, 并将时间所对应的自变量表达式带入 *Step3* 中求解出来的最终函数 S 的具体拟合多项式的表达式, 可以得到问题一第二小问中所需要的预测在 $[300s, 500s]$ 时间范围上, 边界热流的大致趋势和大致的数据值。

5. 1. 3 模型的算法求解

根据要求, 运用算法求解出题目所需要的波速和温度的数学关系, 得到边界热流随时间的变化规律, 对未来热流进行预测分析。

(1) 通过题目给出的已知信息, 做出位移对波速、温度和时间对边界热流的散点图, 判断各个函数的趋势, 并对大致的函数表达式做出预测和判断;

(2) 探究波速、温度对于位移和热流对于时间的关系, 通过最小二乘法对这三个关系做回归、拟合, 寻找能够最大程度上表达其数据集所展示出来的特征的函数表达式, 尽可能地提高精确度。

(3) 分析回归算法的结果, 寻找到波速与位移, 温度与位移, 边界热流与时间这三者的最佳对应函数。

(4) 分类讨论波速和温度以及时间与边界热流之间的关系。

对于探究波速与温度的问题而言，将两者函数表达式中的位移消元，直接得到所对应的波速与温度之间的关系，建立合适的数学模型，求解出波速与温度之间的具体函数表达式；

对于边界热流随时间变化的问题而言，通过上方求解出的最佳函数表达式，计算出时间变化范围在[0,500s]上的边界热流，通过这些利用表达式计算出的边界热流具体值，能够对边界热流的大致变化做一个趋势性的分析。

5.1.4 模型的结果

通过计算机编程求解(详见代码文件 第一题.cpp)，得到的波速与温度之间的关系以及边界热流随时间变化的函数如下：

$$\begin{cases} v(t) = -0.4521 \times t + 3383.391115 & (\text{单位: } K) \\ q(t) = 0.01t^2 & (\text{单位: } w) \end{cases} \quad (9)$$

图 2 表示的是波速与温度之间的关系；图 3 表示的是边界热流随时间变化的趋势变化估计。

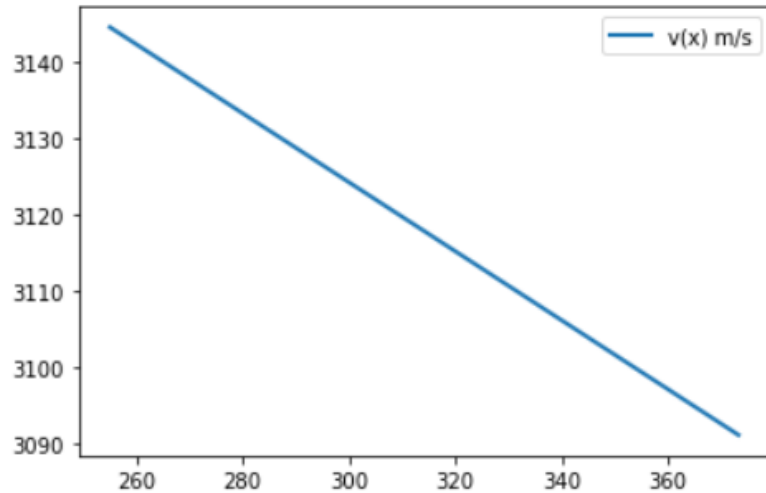


图 3 波速与温度关系曲线图

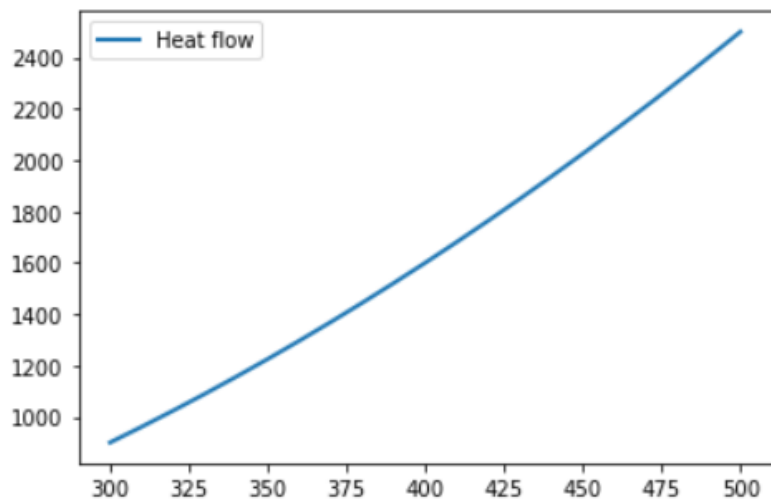


图 4 边界热流随时间变化的趋势预测图

5.2 问题二、问题三的模型建立

5.2.1 物理模型的建立

根据问题描述，我们研究问题二和问题三，发现可以通过建立两个物理模型来解决这两个问题。因此我们建立了热传导模型和超声波运动学模型，用微积分和偏微分方程来解决问题二中探究超声波在固体介质中的传播时间和问题三中探究利用传播时间、材料导热系数、热边界条件、比热、长度、加热时间、初始时刻的温度分布等已知信息求解材料密度的问题。

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：

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) \quad (10)$$

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：

$$\begin{cases} T|_{x=0} = T_0 = 26K \\ -k \frac{\partial T}{\partial x}|_{x=0} = q \\ -k \frac{\partial T}{\partial x}|_{x=L} = 0 \end{cases} \quad (11)$$

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：

$$d\tau = \frac{dx}{v} \quad (12)$$

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 转化成温度变量 T ，

$$d\tau = \frac{dx}{v(T(x,t))} \quad (13)$$

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：

$$\tau(t) = \int_0^L \frac{dx}{v(T(x,t))} \quad (14)$$

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：

$$T(x,t) \quad (15)$$

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：

$$T(x,t) = \chi(x)\kappa(t) \quad (16)$$

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

$$\frac{\partial T(x,t)}{\partial t} = \chi(x)\kappa'(t) \quad (17)$$

$$\frac{\partial T(x,t)}{\partial x} = \chi''(x)\kappa(t) \quad (18)$$

Step2. 代入原方程，得到：

$$\frac{\chi''(x)}{\chi(x)} = \frac{\kappa'(t)}{a^2\kappa(t)} = -\lambda \quad (19)$$

求解得：

$$\begin{cases} \kappa'(t) + \lambda a^2 \kappa(t) = 0 \\ \chi''(x) + \lambda \chi(x) = 0 \end{cases} \quad (20)$$

Step3. 加入边界条件，对 λ 进行分类讨论。

(I) 当 $\lambda \leq 0$ 时：

$$\chi(x) = Ae^{x\sqrt{-\lambda}} + Be^{x\sqrt{-\lambda}} \quad (21)$$

代入边界条件，得到：

$$\begin{cases} A + B = 0 \\ Ae^{l\sqrt{-\lambda}} + Be^{l\sqrt{-\lambda}} = 0 \end{cases} \quad (22)$$

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：

$$\chi(x) = A\cos\beta x + B\sin\beta x \quad (23)$$

代入边界条件，得到：

$$\begin{cases} A = 0 \\ B\sin\beta l = 0 \quad \beta = \frac{n\pi}{l} \quad (n = 1, 2, 3 \dots) \end{cases} \quad (24)$$

因此，我们可以得到：

$$\chi(x) = B \cdot n \cdot \sin \frac{n\pi}{l} x \quad (n = 1, 2, 3 \dots) \quad (25)$$

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

$$\kappa(t) = c \cdot e^{-\lambda a^2 t} \quad (26)$$

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：

$$\begin{aligned} T(x, t) = & \frac{|a|}{a^6} \cdot \left(0.00019a^2 e^{-\frac{x^2}{4a^2 t}} \sqrt{t} (2a^2 t + x^2) \cdot (16a^2 t + x^2) \right. \\ & + 0.00017x \cdot (60a^4 t^2 + 20a^2 t x^2 + x^4) \cdot \left(-a + a \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2a\sqrt{t}}} e^{-\eta^2} d\eta \right) \Bigg) \\ & + T_0 \left(1 - \frac{0.5a \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2a\sqrt{t}}} e^{-\eta^2} d\eta}{|a|} + 0.5 \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2|a|\sqrt{t}}} e^{-\eta^2} d\eta \right) \end{aligned} \quad (27)$$

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：

$$a^2 = \frac{k}{\rho c} \quad (28)$$

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 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 时刻有明确的区分。

具体图示如下



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

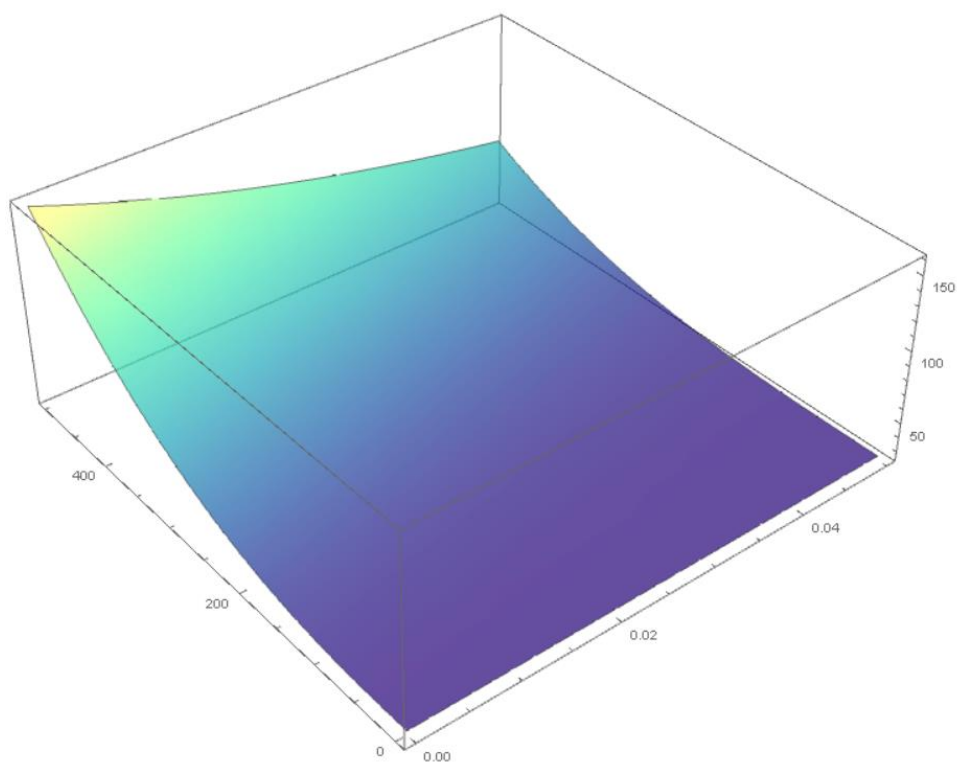


图 10 温度与加热时间和空间位移的关系图

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

$$\tau(t) = \int_0^L \frac{dx}{v(T(x,t))} \quad (29)$$

2) 求解过程

Step1. 将 $T(x,t)$ 代入 $v = v(T)$, 得:

$$v(T) = \left(\left(0.00019a^2 e^{-\frac{x^2}{4a^2t}} \sqrt{t}(2a^2t + x^2) \cdot (16a^2t + x^2) + 0.00017x \cdot (60a^4t^2 + 20a^2tx^2 + x^4) \right. \right. \\ \cdot \left(-a + a \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2a\sqrt{t}}} e^{-\eta^2} d\eta \right) \left. \right) \frac{1}{a^5} + T_0 \left(1 - \frac{0.5a \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2a\sqrt{t}}} e^{-\eta^2} d\eta}{|a|} \right. \\ \left. \left. + 0.5 \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2|a|\sqrt{t}}} e^{-\eta^2} d\eta \right) \right) \cdot k_1 + b_1 \quad (30)$$

其中 k_1 与 b_1 为第一题中求出的常数, 其值如下:

$$\begin{cases} k_1 = -0.4521 \\ b_1 = 3383.391115 \end{cases} \quad (31)$$

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:

$$\begin{aligned}
\tau(t) = \int_0^{0.05} & \left(2/(3124.65 + 0.004521t^2x + \frac{0.001507tx^3}{a^2} + \frac{0.00007535x^5}{a^4} \right. \\
& + \frac{5.8773a \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2a\sqrt{t}}} e^{-\eta^2} d\eta}{|a|} \\
& + \frac{|a|}{a^5} \left(ae^{-\frac{x^2}{4a^2t}} \sqrt{t} (-0.002721a^4t^2 - 0.00153a^2tx^2 - 0.00008502x^4) \right. \\
& + (-0.004521a^4t^2x - 0.001507a^2tx^3 - 0.00007535x^5) \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2a\sqrt{t}}} e^{-\eta^2} d\eta \Big) \\
& \left. - 5.8773 \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2|a|\sqrt{t}}} e^{-\eta^2} d\eta \right) dx \quad (32)
\end{aligned}$$

利用上式，可以做出 $\tau(t)$ 的图像：

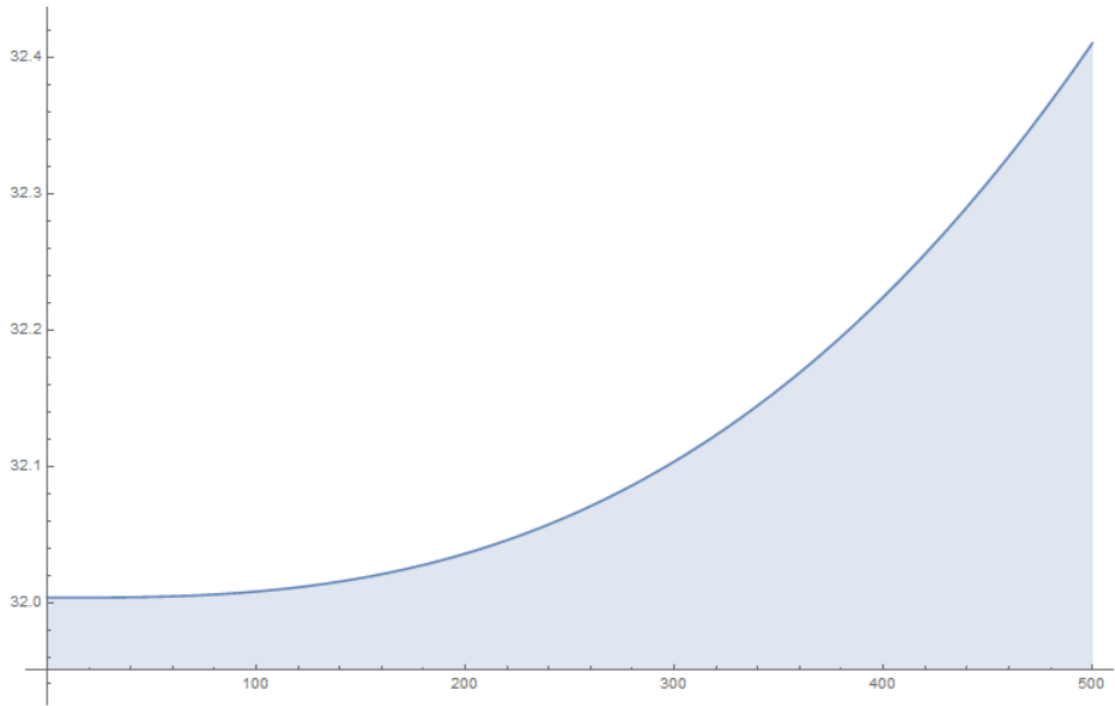


图 11 超声波传播时间随到达时刻的变化图

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：

$$\int_0^{0.05} 2dx / (C_8 - 0.4521(373.15 - FUNC + C_7 \cdot Erfc[\sqrt{2} \sqrt{\frac{p}{t}} x] - C_7 \cdot GR[\frac{1}{2}, \frac{2px^2}{t}])) \quad (33)$$

其中：

$$FUNC = \frac{e^{-\frac{2px^2}{t}} \sqrt{t} \left(-\frac{32}{3000} t^2 - 0.048 p t x^2 - \frac{64}{3000} p^2 x^4 \right) + \sqrt{p} x (C_1 t^2 + C_2 p t x^2 + C_3 p^2 x^4) + \sqrt{p} x (C_4 t^2 + C_5 p t x^2 + C_6 p^2 x^4) \cdot Erf[\sqrt{2} \sqrt{\frac{p}{t}} x]}{2\sqrt{p}\sqrt{2\pi}} \quad (34)$$

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：

$$Erf + Erfc = 1 \quad (35)$$

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\eta^2} d\eta \quad (36)$$

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：

$$\begin{cases} C_1 = 0.050132565492620004 \\ C_2 = 0.13368684131365333 \\ C_3 = 0.053474736525461336 \\ C_4 = -0.050132565492620004 \\ C_5 = -0.13368684131365333 \\ C_6 = -0.053474736525461336 \\ C_7 = 186.575 \\ C_8 = 3383.391115 \end{cases} \quad (37)$$

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。

表 1 材料密度结果

| 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 |

密度数据间的关系如下：

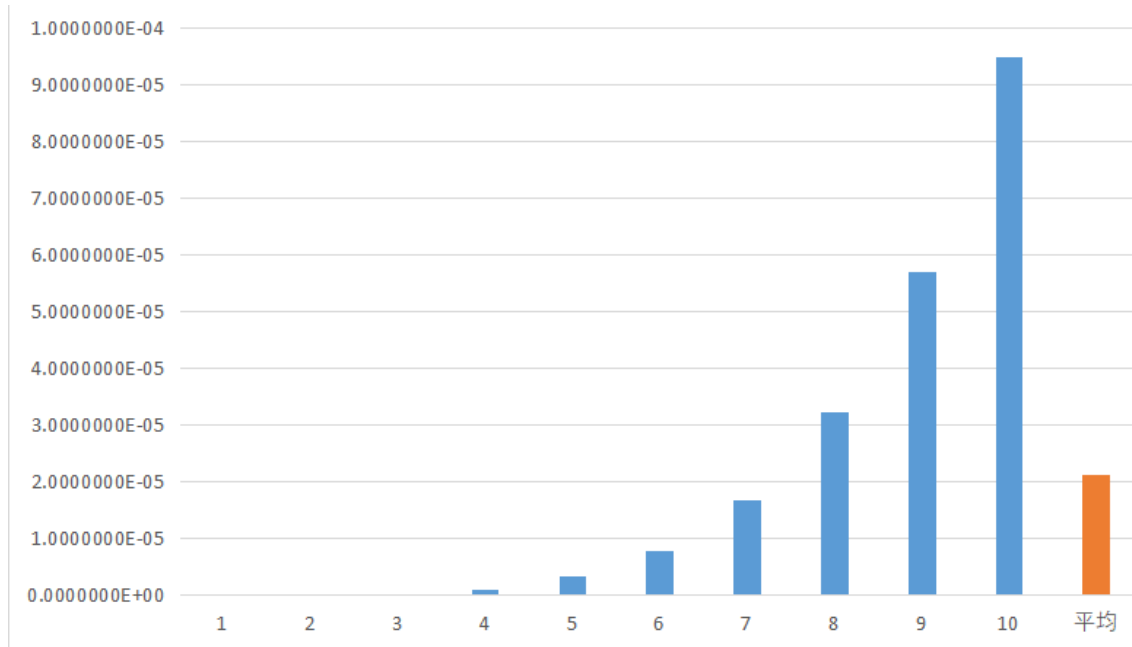


图 12 密度数据图

求得该材料的密度为 $5.60 \times 10^{-6} kg/m^3$ 。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：

$$Q = \int_0^1 0.01t^2 dt = \frac{1}{300} J \quad (38)$$

而根据

$$\tau(t) = \int_0^L \frac{dx}{v(T(x,t))} \quad (39)$$

为了让 $\tau = 31.3867821492946 \mu s$ ，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 $[300s, 600s]$ 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 $[400s, 410s]$ 和 $[500s, 510s]$ 的加热区间为例，给出解。

| 表 2 重构后材料密度结果 1 | |
|------------------|----------------------------------|
| 加热时间(<i>s</i>) | 物质密度(<i>kg/m</i> ³) |
| 401 | 5226.4 |
| 402 | 5198.5 |
| 403 | 5173.4 |
| 404 | 5149.8 |
| 405 | 5127.1 |
| 406 | 5105.0 |
| 407 | 5083.9 |
| 408 | 5063.4 |
| 409 | 5043.5 |
| 410 | 5024.2 |
| 均值 | 5119.5 |

| 表 3 重构后材料密度结果 2 | |
|------------------|----------------------------------|
| 加热时间(<i>s</i>) | 物质密度(<i>kg/m</i> ³) |
| 501 | 12147 |
| 502 | 12067 |
| 503 | 11992 |
| 504 | 11924 |
| 505 | 11856 |
| 506 | 11790 |
| 507 | 11726 |
| 508 | 11664 |
| 509 | 11603 |
| 510 | 11543 |
| 均值 | 11831.2 |

表 2 表示以[400*s*, 410*s*]为加热时间区间, 求解出的各个时间点上的物质密度, 表 3 表示以[500*s*, 510*s*]为加热时间区间, 求解出的各个时间点上的物质密度。可以从数据信息上观察到这两种物质密度是符合客观世界中固体物质密度的, 比原结果数据更加合理。

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

- 1、本文给出的解为题给模型下的精确解析解, 精度与模型适变性完全合乎要求。
- 2、通过分析数据三的数据与结果, 我们将问题三提供的数据进行合理应用, 并以此为基础给出了符合实际情况的解。
- 3、通过文中给出的解析解, 我们对结果进行了可视化处理, 使其更加直观。

6.1.2 模型的缺点分析

- 1、本题模型被简化为一维模型, 无法直接应用到现实情况, 缺乏普适性。
- 2、初始边界条件较为理想, 实际生活难以实现, 故模型与现实情况可能存在一定的差距。
- 3、本题求解过程较为繁琐, 但求解的过程较依赖人力, 未实现完全的智能化。

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。

七、参考文献

- [1]李金鸿,贺达海. 关于一维非线性晶格热传导的有限时间涨落定理[J]. 曲靖师范学院学报, 2021, 40(03):17-21.
- [2]王洁琼. 基于有限差分法和BP算法的热防护服厚度设计[J]. 电子世界, 2020(15):153-155.
- [3]臧顺全. 热传导方程正问题和反问题的数值解研究[D]. 西安理工大学, 2019.
- [4]郭帅平. 非齐次问题边界积分方程中域积分计算方法的研究及应用[D]. 湖南大学, 2015.

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, 310.00000000])

```

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. 00000
000, 363. 00000000, 364. 00000000, 365. 00000000, 366. 00000000, 367. 00000000, 368. 00
000000, 369. 00000000, 370. 00000000, 371. 00000000, 372. 00000000, 373. 00000000, 374
. 00000000, 375. 00000000, 376. 00000000, 377. 00000000, 378. 00000000, 379. 00000000,
380. 00000000, 381. 00000000, 382. 00000000, 383. 00000000, 384. 00000000, 385. 000000
00, 386. 00000000, 387. 00000000, 388. 00000000, 389. 00000000, 390. 00000000, 391. 000
00000, 392. 00000000, 393. 00000000, 394. 00000000, 395. 00000000, 396. 00000000, 397.
00000000, 398. 00000000, 399. 00000000, 400. 00000000, 401. 00000000, 402. 00000000, 4
03. 00000000, 404. 00000000, 405. 00000000, 406. 00000000, 407. 00000000, 408. 0000000
0, 409. 00000000, 410. 00000000, 411. 00000000, 412. 00000000, 413. 00000000, 414. 0000
0000, 415. 00000000, 416. 00000000, 417. 00000000, 418. 00000000, 419. 00000000, 420. 0
0000000, 421. 00000000, 422. 00000000, 423. 00000000, 424. 00000000, 425. 00000000, 42
6. 00000000, 427. 00000000, 428. 00000000, 429. 00000000, 430. 00000000, 431. 00000000
, 432. 00000000, 433. 00000000, 434. 00000000, 435. 00000000, 436. 00000000, 437. 00000
000, 438. 00000000, 439. 00000000, 440. 00000000, 441. 00000000, 442. 00000000, 443. 00
000000, 444. 00000000, 445. 00000000, 446. 00000000, 447. 00000000, 448. 00000000, 449
. 00000000, 450. 00000000, 451. 00000000, 452. 00000000, 453. 00000000, 454. 00000000,
455. 00000000, 456. 00000000, 457. 00000000, 458. 00000000, 459. 00000000, 460. 000000
00, 461. 00000000, 462. 00000000, 463. 00000000, 464. 00000000, 465. 00000000, 466. 000
00000, 467. 00000000, 468. 00000000, 469. 00000000, 470. 00000000, 471. 00000000, 472.
00000000, 473. 00000000, 474. 00000000, 475. 00000000, 476. 00000000, 477. 00000000, 4
78. 00000000, 479. 00000000, 480. 00000000, 481. 00000000, 482. 00000000, 483. 0000000
0, 484. 00000000, 485. 00000000, 486. 00000000, 487. 00000000, 488. 00000000, 489. 0000
0000, 490. 00000000, 491. 00000000, 492. 00000000, 493. 00000000, 494. 00000000, 495. 0
0000000, 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. 2400
0000, 1108. 89000000, 1115. 56000000, 1122. 25000000, 1128. 96000000, 1135. 69000000,
1142. 44000000, 1149. 21000000, 1156. 00000000, 1162. 81000000, 1169. 64000000, 1176.
49000000, 1183. 36000000, 1190. 25000000, 1197. 16000000, 1204. 09000000, 1211. 04000
000, 1218. 01000000, 1225. 00000000, 1232. 01000000, 1239. 04000000, 1246. 09000000, 1

253. 16000000, 1260. 25000000, 1267. 36000000, 1274. 49000000, 1281. 64000000, 1288. 81000000, 1296. 00000000, 1303. 21000000, 1310. 44000000, 1317. 69000000, 1324. 96000000, 1332. 25000000, 1339. 56000000, 1346. 89000000, 1354. 24000000, 1361. 61000000, 1369. 00000000, 1376. 41000000, 1383. 84000000, 1391. 29000000, 1398. 76000000, 1406. 25000000, 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. 81000000, 1536. 64000000, 1544. 49000000, 1552. 36000000, 1560. 25000000, 1568. 16000000, 1576. 09000000, 1584. 04000000, 1592. 01000000, 1600. 00000000, 1608. 01000000, 1616. 04000000, 1624. 09000000, 1632. 16000000, 1640. 25000000, 1648. 36000000, 1656. 49000000, 1664. 64000000, 1672. 81000000, 1681. 00000000, 1689. 21000000, 1697. 44000000, 1705. 69000000, 1713. 96000000, 1722. 25000000, 1730. 56000000, 1738. 89000000, 1747. 24000000, 1755. 61000000, 1764. 00000000, 1772. 41000000, 1780. 84000000, 1789. 29000000, 1797. 76000000, 1806. 25000000, 1814. 76000000, 1823. 29000000, 1831. 84000000, 1840. 41000000, 1849. 00000000, 1857. 61000000, 1866. 24000000, 1874. 89000000, 1883. 56000000, 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. 00000000, 2034. 01000000, 2043. 04000000, 2052. 09000000, 2061. 16000000, 2070. 25000000, 2079. 36000000, 2088. 49000000, 2097. 64000000, 2106. 81000000, 2116. 00000000, 2125. 21000000, 2134. 44000000, 2143. 69000000, 2152. 96000000, 2162. 25000000, 2171. 56000000, 2180. 89000000, 2190. 24000000, 2199. 61000000, 2209. 00000000, 2218. 41000000, 2227. 84000000, 2237. 29000000, 2246. 76000000, 2256. 25000000, 2265. 76000000, 2275. 29000000, 2284. 84000000, 2294. 41000000, 2304. 00000000, 2313. 61000000, 2323. 24000000, 2332. 89000000, 2342. 56000000, 2352. 25000000, 2361. 96000000, 2371. 69000000, 2381. 44000000, 2391. 21000000, 2401. 00000000, 2410. 81000000, 2420. 64000000, 2430. 49000000, 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]])"
    ]
},
"execution_count": 1,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
    "from PIL import Image\n",
    "import numpy as np\n",
    "\n",
    "img = Image.open('1.png') # 使用 PIL 打开图片\n",
    "data1 = img.getdata() # 获取图片的数据信息 class <'ImagingCore'>\n",
    "data1 = np.array(data1) # 把这个数据通过 numpy 转换成多维度的张量\n",
    "\n",
    "data1"
]
},
{
    "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
}

```