高温作业专用服装设计

摘要

本文以传热学思想与原理及工作九十分钟实验数据为依据，研究高温作业服装设计问题。采用一维非稳态及稳态下的数值与解析解法对不同厚度、不同外界温度及不同工作时间条件下服装内部温度分布和变化情况进行模拟。较为准确的分析出其温度随时间和厚度的变化过程，并解决了在符合规定约束条件下以降低成本为导向的服装最优结构设计问题。

第一问，因服装表面积远大于其厚度且其材料温度随时间和厚度不断变化，可将其简化为无限大平壁，建立一维非稳态导热模型。根据实验数据，确定由非稳态向稳态转变的时刻，各自加以分析。在研究过程中，我们忽略热辐射导热现象，在确定其材料无内热源且常物性的条件下，对一维非稳态通用导热控制微分方程加以简化。运用区域离散化理论以有限个离散点代替连续空间，在此基础上运用控制容积积分法，借助阶梯型分布型线对控制微分方程进行离散化处理，对于加权因子 f 取不同值可得到显式、隐式和克兰克-尼克尔森格式的差分方程组。通过 von Neumann 分析方法可得出，显式格式和克兰克-尼克尔森格式需要满足显式稳定性准则对步长有所限制，全隐式格式为普适性的无条件稳定格式。我们运用显式格式的递推公式法和隐式格式三对角阵算法分别通过 matlab 编程进行求解，结果相似性较高。要特别指出的是，我们将本题传热材料抽象为导热系数随厚度呈分段函数变化的模型，因此在交界面的确定和交界面导热系数的确定时选用物性阶跃面作为节点和调和平均法解决问题；初值与边界条件的选择应用第一类边界条件并拟合实验曲线用于此处。

第二问，在第一问模型的基础上，改变初值和边界条件及各个参数值，在II层材料厚度在确定范围内的均匀变化下提取内空气层内表面温度和时间可形成一个三维函数模型。在加以确定限制条件，会得出在满足条件下的II层材料厚度变化范围，以经济学原理下低成本导向为选择依据，可求得其范围内的最小值即为最优情况。

第三问，在第二问的基础上，利用双重循环可得符合限制条件的定Ⅱ层厚度下Ⅳ层厚度范围，进而使II层厚度均匀变化，可得两者对应范围关系。因为材料成本远大于内空气层成本，因此优先选择II层最小厚度为最优解，在此前提下选择其对应IV层厚度的最优值。

最后对模型进行评价，对不足提出改进方向和建议。

关键词：

传热学、一维非稳态模型、数值解法、matlab、非线性最优求解

1 问题重述

2015 年 8 月 12 日发生在天津滨海新区的火灾爆炸事故给我国的人民生命财产和国民经济造成了极大的损失，过程中共有 99 名消防人员遇难。对于消防人员来说，在高温环境下工作时，穿着特殊的服装以避免灼伤尤为重要。

专业服装一般由三层织物材料组成，记为 I、II、III 层，其中 III 层和皮肤之间还存在空隙，将此空隙记为 IV 层。I 层与 III 层厚度一定，分别为 0.6mm，3.6mm。

我们需要建立数学模型和算法，解决以下问题：

1. 根据题目给出的专业服装材料的特定参数值（下同），以及各层的厚度及环境温度，计算工作从 0-90 分钟之间服装各层的温度分布。
2. 根据题目给出的 IV 层的厚度及环境温度，在保证 60 分钟时假人皮肤外侧温度不超过 47℃，且超过 44℃的时间不超过 5 分钟的情况下，确定 II 层的最优厚度。
3. 根据给出的环境温度，在保证 30 分钟时假人皮肤外侧温度不超过 47℃，且超过 44℃的时间不超过 5 分钟的情况下，确定 II 层和 IV 层的最优厚度。

2 问题分析

第一问中，首先需要确定导热过程是否进入稳态，观察附件二，通过假人皮肤温度随时间的变化关系，在稳态和非稳态情况下分别建立导热微分方程：稳态时，运用第一类边界条件下多层平壁模型求解得到；非稳态时，通过控制容积积分法将一维非稳态导热微分方程转化为相应的离散方程，再运用三对角阵算法求出材料温度与距离和时间的三维抽象函数关系。带入具体数据即可得到三维具象函数关系。

第二问中，首先根据第一问中在非稳态情况下得到的材料温度与距离和时间的三维抽象函数关系，运用循环法，选取最佳步长得到 II 层厚度与空气内表面温度（即皮肤外侧温度，下同）和时间在有限范围内的全体三维具象函数关系，通过选取符合空气内表面温度和时间条件的三维具象函数关系，求得 II 层的最佳厚度（即最小厚度，下同）。

第三问中，首先根据第一问中在非稳态情况下得到的材料温度与距离和时间的三维抽象函数关系，运用迭代法，选取最佳步长得到 II 层、IV 层厚度与空气内表面温度和时间在有限范围内的全体四维具象函数关系，通过选取符合空气内表面温度和时间条件的四维具象函数关系，求得 II、IV 两层的最佳厚度。

3 基本假设

1， 服装各层分别各向同性，是连续均匀介质，且不随时间变化
2， 工作开始时防护服各材料层温度均为 37℃
3， 各材料层之间紧密接触，无空隙
4， 工作状态下，外环境和防护服之间只存在热对流和热传导，不存在热辐射。
5， 所有传热接触面温度分布均匀。

4 符号列表

Φ_x、Φ_y、Φ_z：沿 x、y、z 方向流入的热流；
T：温度；T_i^j：长度坐标为 i 的端面在 j 时刻的温度；
λ：导热系数；
τ：时间；
q_v：内热源温度
ρ：密度；
c：比热容；
x：长度坐标；
δ：厚度；
f：加权因子；
q：热流密度；
k：热导率；
h：对流换热系数；
a：导温系数
A:表面积；
Bio：毕渥数
Fo：傅里叶数；
S:内热源
Δt：时间间隔；
Δx：步长；

注：上表为本文所用基本变量，如有变式，已在相应位置说明。

5 问题一的求解

在问题一中，我们需要根据已知的环境温度和服装的各项指标求解温度分布。我们对服装温度进行建模。在这一问题中，我们需要得到的量为服装各层的温度随时间的变化关系。

5.1 理论基础

5.1.1 热量的传递方式

热量的传递有三种基本方式，分别是：热传导、热对流和热辐射。（P3A）热传导发生于物体内部或相互接触的物体表面之间，其依赖于两个基本条件：一是必须有温差，二是必须直接接触或是在物体内部传递；热对流是指由于流体的宏观运动致使不同温度的流体相对位移而产生的热量传递现象；热辐射是由于物体内部微观粒子的热运动而使物体向外发射辐射能的现象。

5.1.2 导热微分方程

导热微分定律的推导以能量守恒定律和傅里叶定律为基础。（P27B）

根据能量守恒定律可知：在单位时间内，从 x,y,z 三个方向通过导热进入微元体的净热量，加上微元体自身内热源的生成热，应该等于微元体内能增量。即

$$\boxed{\text{导入微元体净热量}}+\boxed{\text{微元体内热源生热}}=\boxed{\text{微元体内能增量}} \qquad \text{（a）}$$

单位时间内，x 方向进入微元体的净热流为

$$\Delta\Phi_x=\Phi_x-\Phi_{x+dx}$$

单位时间内，x 方向进入微元体的净热流为

$$\Delta\Phi_y=\Phi_y-\Phi_{y+dy}$$

单位时间内，x 方向进入微元体的净热流为

$$\Delta\Phi_z=\Phi_z-\Phi_{z+dz}$$

根据傅里叶定律并按泰勒级数展开，略去二阶导数以后的各项，可得单位时间内 x 方向进入微元体的净热流量为

$$\Delta\Phi_x=-\frac{\partial\Phi_x}{\partial x}dx,\Phi_x=-\lambda\frac{\partial T}{\partial x}dydz$$

在某一时间间隔（$d\tau$）内从 x 方向进入微元体的净热量为

$$\Delta\Phi_x d\tau=\frac{\partial}{\partial x}(\lambda\frac{\partial T}{\partial x})dxdydzd\tau$$

由此可以类推出在某一时间间隔内分别从 y 和 z 方向进入微元体的净热量。最终得到通过热传导方式进入微元体的净热量:

$$\text{导入微元体净热量}=[\frac{\partial}{\partial x}(\lambda\frac{\partial T}{\partial x})+\frac{\partial}{\partial y}(\lambda\frac{\partial T}{\partial y})+\frac{\partial}{\partial z}(\lambda\frac{\partial T}{\partial z})]dxdydzd\tau \qquad \text{(b)}$$

物体由于通电或化学反应等原因自行生成热量称为内热源。单位体积的物体在单位时间内产生的热量记作 q_v（W/m^3），称为内热源温度。则在 $d\tau$ 时间间隔内微元体自身内热源生成的能量为

$$\text{微元体内热源生成热}=q_v dzdydzd\tau \qquad \text{(c)}$$

考虑一般在导热问题中物体的密度和比热容近似不变，所以在$d\tau$时间间隔内，微元体内内能的增量可表示为

$$\text{微元体内能增量}=\rho c\frac{\partial T}{\partial \tau}dxdydzd\tau \tag{d}$$

将式(b)-(d)代入式(a)，经整理得

$$\rho c\frac{\partial T}{\partial \tau}=\frac{\partial}{\partial x}(\lambda\frac{\partial T}{\partial x})+\frac{\partial}{\partial y}(\lambda\frac{\partial T}{\partial y})+\frac{\partial}{\partial z}(\lambda\frac{\partial T}{\partial z})+q_v$$

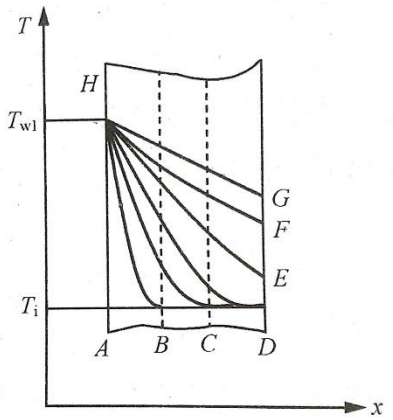
此式即为笛卡尔坐标系中三维非稳态导热微分方程的一般形式。

5.1.3 维度

当平壁的表面温度均匀不变，且平壁的厚度远远小于它的长度和宽带时，可认为它的导热只沿厚度方向进行，即一维导热。(P25A)

5.1.4 稳态与非稳态导热过程

在导热过程中，如果温度不随时间发生变化，则认为是稳态导热，否则为非稳态导热。简言之，非稳态导热的过程中导热体本身温度也在变化，但经过足够长的时间（理论上为无限长时间），其内层温度分布将成为一条直线 HG（图一）。此时，非稳态导热过程结束，进入稳态导热过程。



图一

5.1.5 一维非稳态导热过程离散解法

5.1.5.1 区域离散化

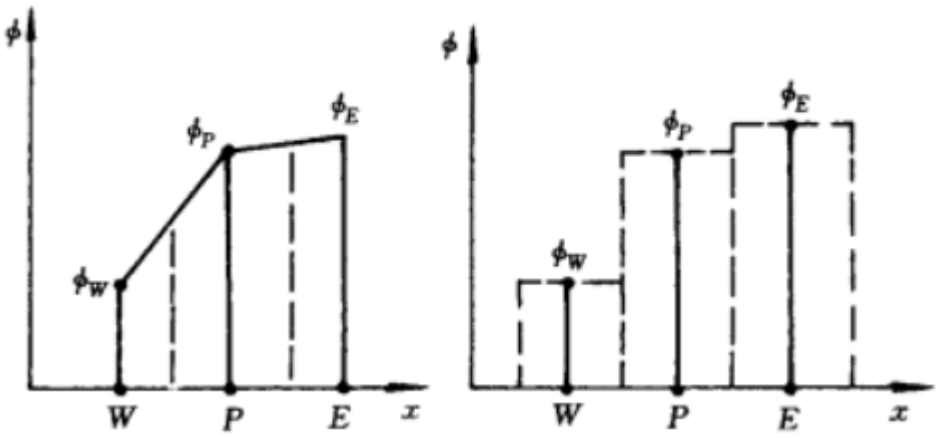
所谓区域离散化实质上就是用一组有限个离散的点来代替原来的连续空间。一般的实施过程是：把所计算的区域划分成许多个互不重叠的子区域，确定每个子区域中的节点位置及该节点所代表的控制容积。我们把节点看成是控制容积的代表。控制容积与子区域并不总是重合的。在区域离散化过程开始时，由一系列与坐标轴相应的直线或曲线簇所划分出来的小区域成为子区域。视节点在子区域中位置的不同，可以把 FDM 及 FVM 中的区域离散化方法分成两大类：外节点法和内节点法。

5.1.5.2 控制容积积分法

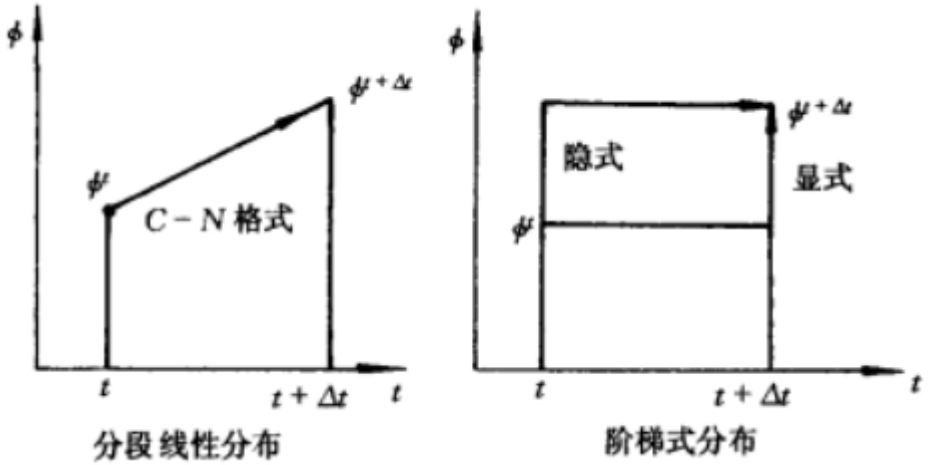
控制容积积分法是有限容积法中建立离散方程的主要方法，其导出离散方程的主要步骤如下：

1．将守恒型的控制方程在任一控制容积及时间间隔内对空间与时间作积分。
2．选定未知函数及其导数对时间及空间的局部分布曲线，即型线，也就是如何从相邻节点的函数值来确定控制容积界面上被求函数值的插值方式。
3．对各个项按选定的型线作出积分，并整理成关于节点上未知值 9 的代数方程。

在实施控制容积积分法时常用的型线有两种，即分段线性分布及阶梯式分布。在图二画出了函数 随空间坐标而变化的这两种型线，而图三则是随时间而变化的几种情形。



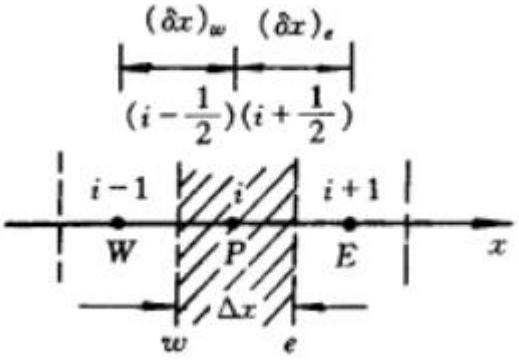
图二



图三

将一维模型方程的守恒形式对图四所示的控制容积 P 在 时间间隔内作积分，把可积的部分积出后得：

$$\rho\int_{\omega}^{e}\left(\phi^{t+\Delta t}-\phi^{t}\right)dx+\rho\int_{t}^{t+\Delta t}\left[(u\phi)_{e}-(u\phi)_{\omega}\right]dt$$
$$=\Gamma\int_{t}^{t+\Delta t}\left[\left(\frac{\partial\phi}{\partial x}\right)_{e}-\left(\frac{\partial\phi}{\partial x}\right)_{\omega}\right]dt+\int_{\omega}^{e}Sdxdt$$



图四

为了最终完成各项积分以获得节点上未知值间的代数方程，需要对各项中变量 的型线作出抉择。正是在这一步中，引入了对被求量的近似处理方法。

非稳态项 需要选定 随 x 而变化的型线，这里取为阶梯式，即同一控制容积中各处的 值相同，等于节点上之值 ，于是有：

$$\int_{\omega}^{e}\left(\phi^{t+\Delta t}-\phi^{t}\right)dx=\left(\phi_{P}^{t+\Delta t}-\phi_{P}^{t}\right)\Delta x$$

5.1.5.3 三对角阵算法

显式格式的优点在于下一时层之值可以直接由上一时层的值计算而得，不必解联立代数方程。但是其稳定性条件限制了所能采用的最大时间步长。对于所有的隐式格式（f>0 的格式），由于每一个节点方程中都包含了同一时层上相邻三点的值，必须通过解联立方程组才能获得该时层上的值。因此，无论是对一维稳态导热问题，还是一位非稳态的隐式格式，我们都必须联立求解形如下式的代数方程：

$$a_P T_P = a_E T_E + a_W T_W + b$$

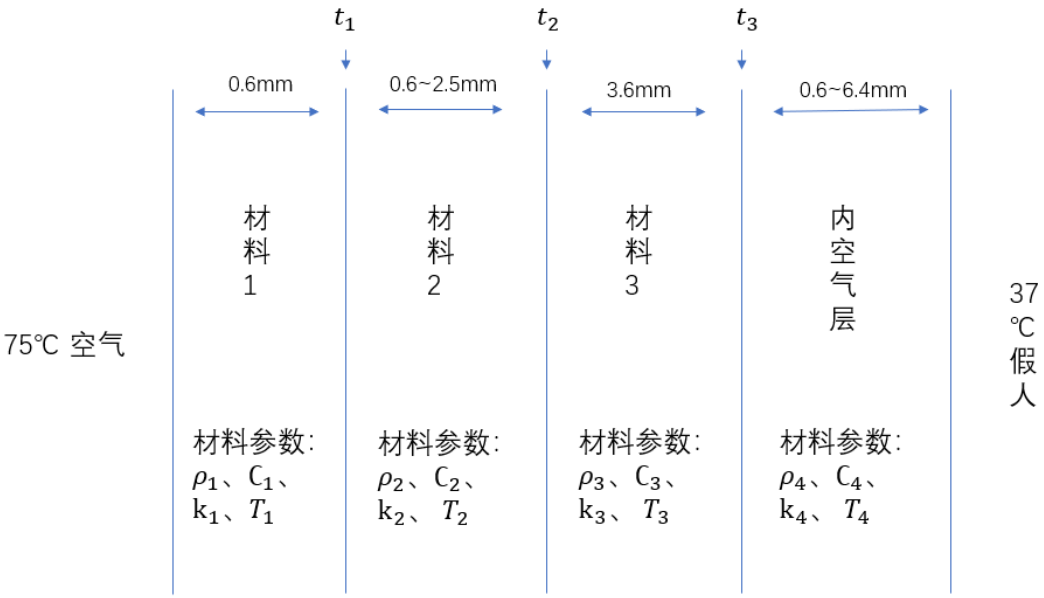
上式表明，每个节点的代数方程中最多只含三个节点的未知值，因此，一维导热问题的离散方程组的系数阵为一个三对角阵。

对于三对角阵的代数方程组，基于 Gauss 消元法的 Thomas 算法应用最广。

Thomas 的求解过程分为消元与回代两步。消元时，从系数矩阵的第二行起，逐一把每行中的非零元素消去一个，使原来的三元方程化为二元方程。消元进行到最后一行时，该二元方程就化为一元，可立即得出该未知量的值。然后逐一往前回代，由各二元方程解出其它未知值。

5.2 建立模型

题中所述防护服结构可抽象如下（图五）：



注： ρ(密度)、C(比热容)、k(导热系数)、t(交界面温度)、T(材料温度)

图五

由题意知：

1）由于防护服厚度远小于其表面积，符合前文所述一维导热概念，故将其理想化为一维导热过程。

2）在附件二中我们发现在 1645s 后，皮肤外侧温度稳定在 48.08℃，符合稳定导热过程的特征；而在该时间点之前，皮肤外侧温度随着热传导过程的进行而升高，符合非稳定导热过程。

因此，我们认为此过程分为两部分：一维非稳定导热过程和一维稳定导热过程。对于一维非稳态导热过程，我们建立非稳态通用微分方程：

$$\rho c\frac{\partial T}{\partial \tau}=\frac{\partial}{\partial x}(\lambda\frac{\partial T}{\partial x})+\frac{\partial}{\partial y}(\lambda\frac{\partial T}{\partial y})+\frac{\partial}{\partial z}(\lambda\frac{\partial T}{\partial z})+q_v$$

因为该模型无内热源且为一维，所以上式可简化为

$$\rho c\frac{\partial T}{\partial t}=\frac{\partial\left[\lambda\frac{\partial T}{\partial x}\right]}{\partial x}$$

在研究一位非稳态导热微分方程时，常用实验测定、理论分析和数值模拟三种方法，在这里我们采用了数值模拟的方法。

具体方法为运用控制容积积分法将微分方程转换为离散方程。为了建立其离散形式，在$[t,t+\Delta t]$时间间隔对控制体 P 作积分，ρc取 t 时刻之值，记为$(\rho c)_P$，则可得：

$$(\rho c)_P\,\Delta x\left(T_P^{t+\Delta t}-T_P^t\right)=\int_t^{t+\Delta t}\left[\frac{\lambda_e\left(T_e-T_p\right)}{(\delta x)_e}-\frac{\lambda_\omega\left(T_P-Tw\right)}{(\delta x)_\omega}\right]dt$$

为了把积分进行到底，需要选择上式右端项中 T 如何随时间而变化的型线。常用的型线有三种，它们都可以用以下的关系式来表示：

$$\int_t^{t+\Delta t}Tdt=\left[\,fT^{t+\Delta t}+(1-f)T^t\,\right]\Delta t=\left[\,fT+(1-f)T^0\,\right]\Delta t$$

这里为书写方便，上角标（$t+\Delta t$）也已删去，而上角标 t 则以 0 代替。f 是在 0 与 1 之间的加权因子。据此，上述积分式最后可化为：

$$(\rho c)_p\,\tfrac{\Delta x}{\Delta t}\left(T_p-T_p^0\right)=f\left[\frac{\lambda_e\left(T_E-T_P\right)}{(\delta x)_e}-\frac{\lambda_\omega\left(T_p-T_w\right)}{(\delta x)_\omega}\right]+(1-f)\left[\frac{\lambda_e\left(T_E^0-T_P^0\right)}{(\delta x)_e}-\frac{\lambda_\omega\left(T_P^0-T_W^0\right)}{(\delta x)_\omega}\right]$$

上式是一维非稳态导热两层格式（即离散方程中仅出现相邻两时层上的值）的一种通用形式，取 f=0，1 及 1/2 可依次得显式，隐式及 Crank-Nicolson 格式。在直角坐标系中当网格均分时，无内热源、常物性导热问题的这三种格式分别为：

显式 $$\frac{T_P-T_P^0}{\Delta t}=a\frac{T_E^0-2T_P^0-T_W^0}{\Delta x^2}$$

隐式 $$\frac{T_P-T_P^0}{\Delta t}=a\frac{T_E-2T_P-T_W}{\Delta x^2}$$

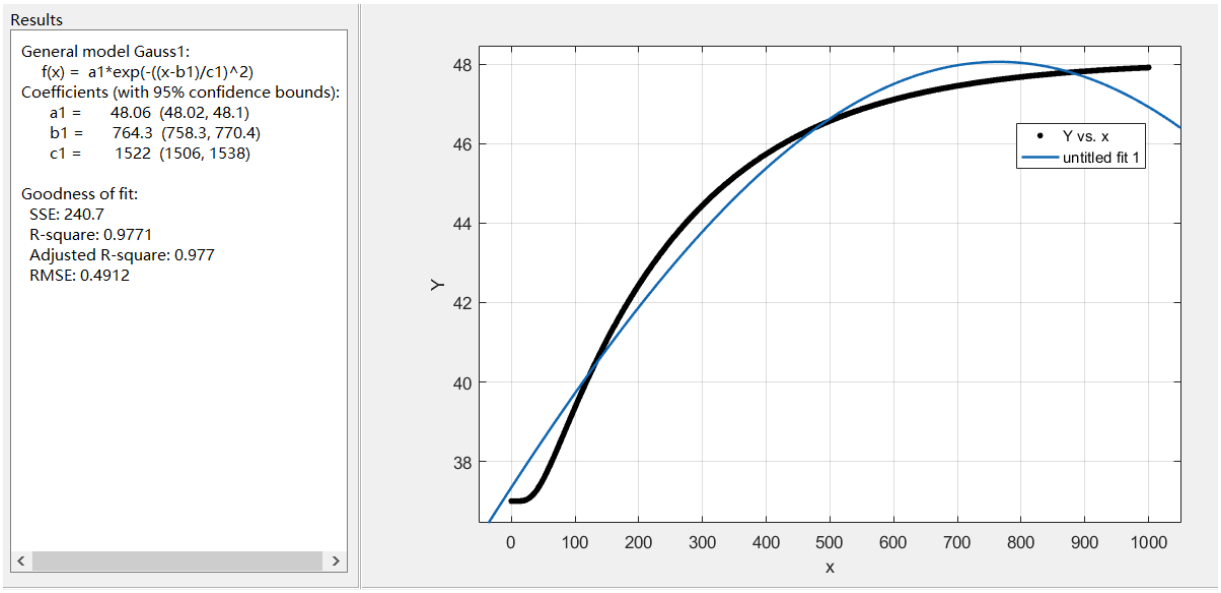
C-N 格式 $$\frac{T_P-T_P^0}{\Delta t}=\frac{a}{2}\left(\frac{T_E-2T_P-T_W}{\Delta x^2}+\frac{T_E^0-2T_P^0-T_W^0}{\Delta x^2}\right)$$

可以用分析方法证明对于源项不随时间而变的问题，积分式当 $\frac{1}{2} \le f \le 1$ 时是绝对稳定的，而当 $0 \le f < \frac{1}{2}$ 时，稳定的条件则为 $\frac{a\Delta t}{\Delta x^2} \le \frac{1}{2(1-2f)}$ 。

5.3 模型求解

5.3.1 显式格式求解

边界条件的拟合图



图六

显式差分方程如下：

$$t_i^{k+1} = Fo_\Delta(t_{i-1}^k + t_{i+1}^k) + (1-2Fo_\Delta)t_i^k$$

初始及边界条件如下：

X=0 时，T=75℃；

t=0 时，T=37℃；

x=15.2mm 时，T(t)= $a_1 * e^{-(\frac{x-b_1}{c_1^2})}$

稳定性限制：

$$Fo_\Delta \le \frac{1}{2Bi_\Delta + 2}$$

求解过程见附录一代码

5.3.2 隐式格式求解

隐式差分方程如下：

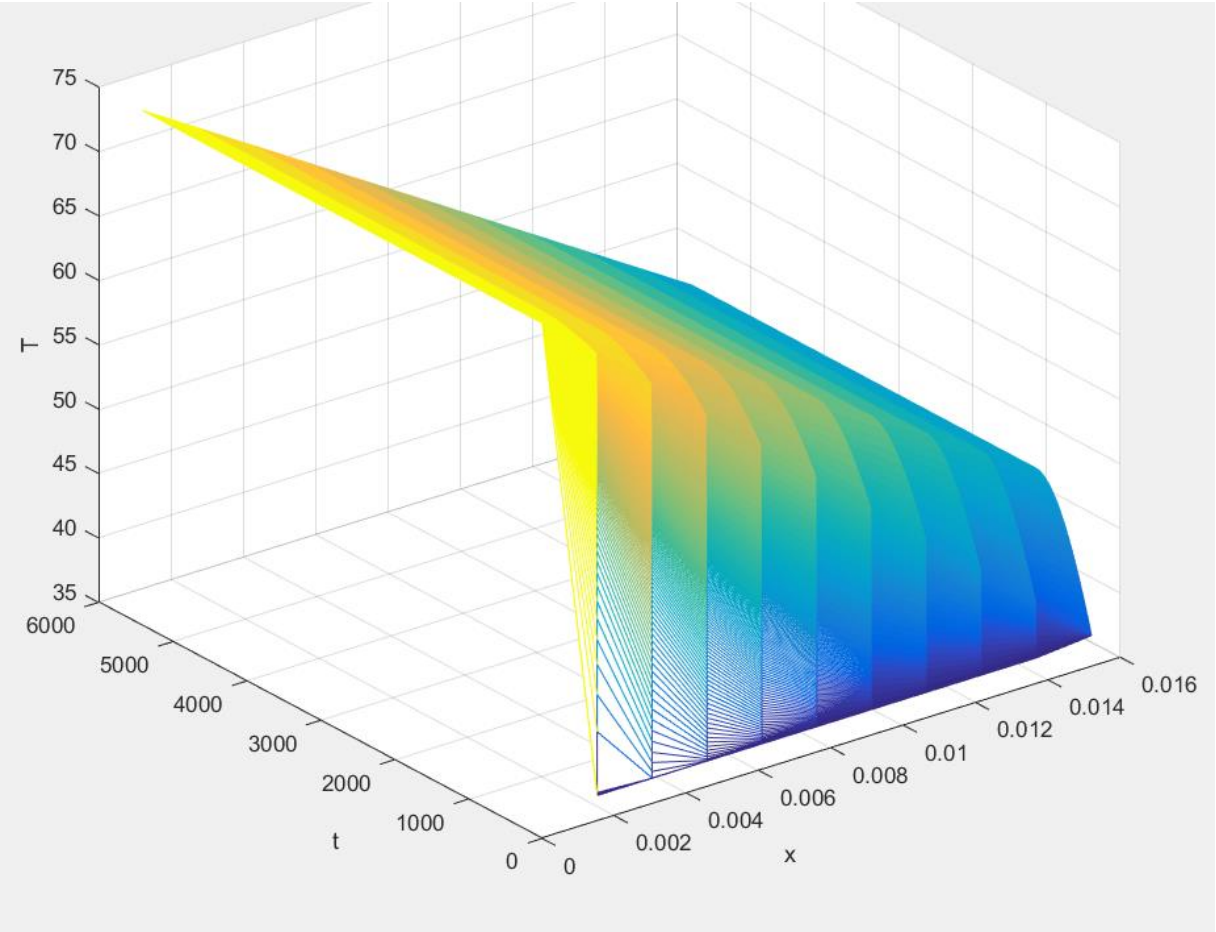
$$A_P^k T_P^k = A_E^k T_E^k + A_W^k T_W^k + A_P^k T_P^{k-1}$$

初始及边界条件同显式

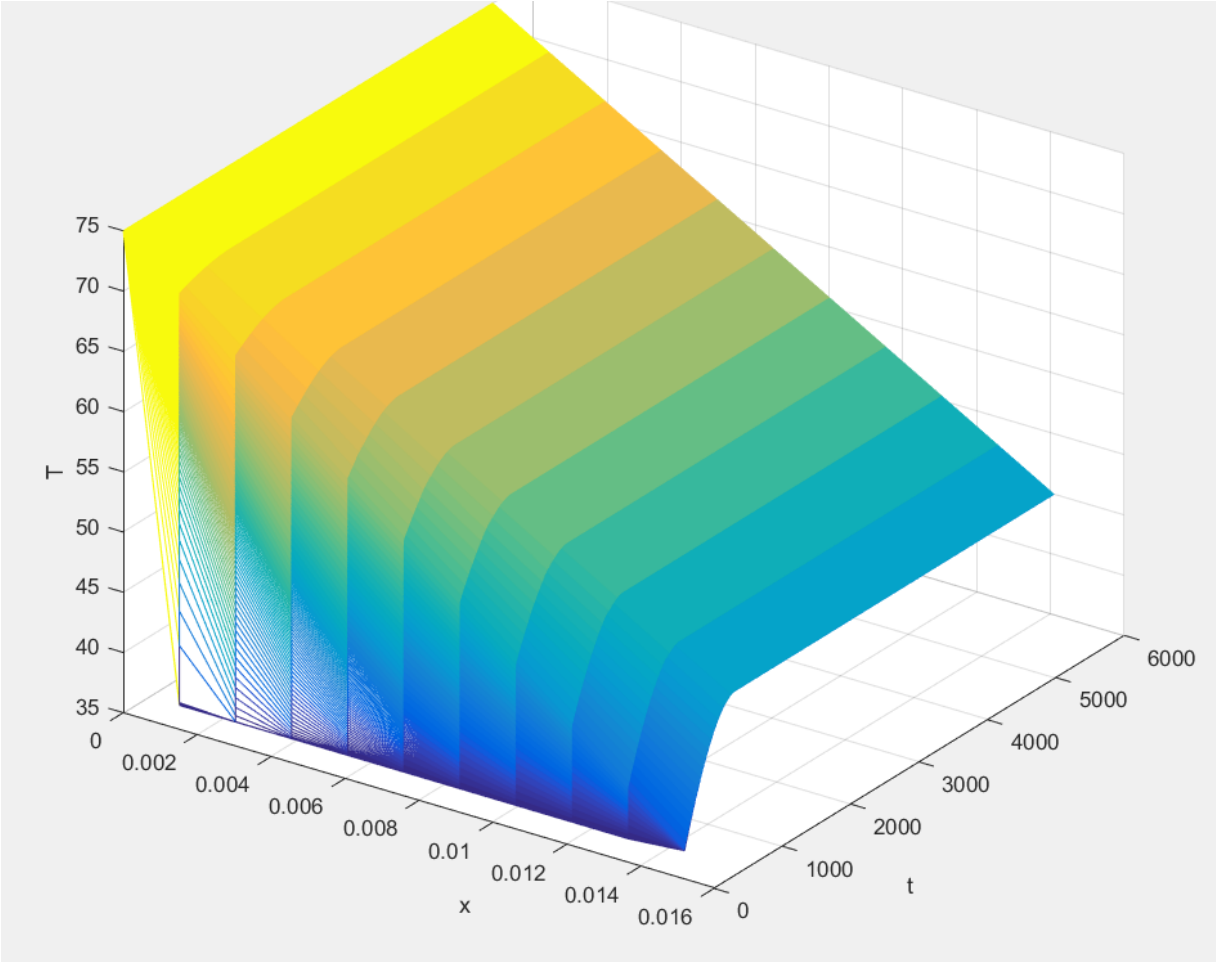
稳定性限制：恒稳

求解过程见附录二代码

5.4 求解结果



图七



图八

5.5 几点问题

5.5.1 交接面的导热系数

对于界面上当量导热系数的确定方法有两种，算术平均法和调和平均法。

当 f=0.5 时，

$$k_e = \frac{2k_j k_{j+1}}{k_j + k_{j+1}}$$

k_e 是 k_j 和 k_{j+1} 的调和平均。

调和平均法优于算术平均法，因为

（1） 取 $k_{j+1} \to 0$，此时算术平均法所得结果 $\to 0$，符合实际情况。但同时得 $k_e \to f_e k_i$ 给出了 e 面的热流量不等于零，这是不正确的。

（2） 取 k_j ？ k_{j+1}，得

$$k_e \to \frac{k_{j+1}}{f_e}$$

算术平均法仍保留了 k_j 对 k_e 的影响，因为 k_j ？ k_{j+1}，温差 $T_i - T_{i+1}$ 实际上发生在 h_e^+ 这段距离上，所以热流量公式应写为

$$q_e = \frac{k_{j+1}(T_j - T_{j+1})}{h_e^+}$$

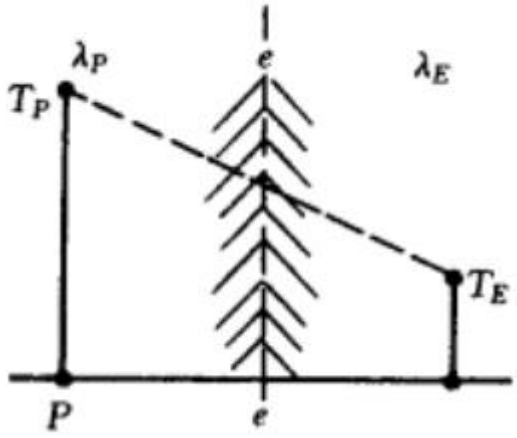
若用算术平均法，无法得到正确的热流量计算公式。

5.5.2 阶跃面与节点位置

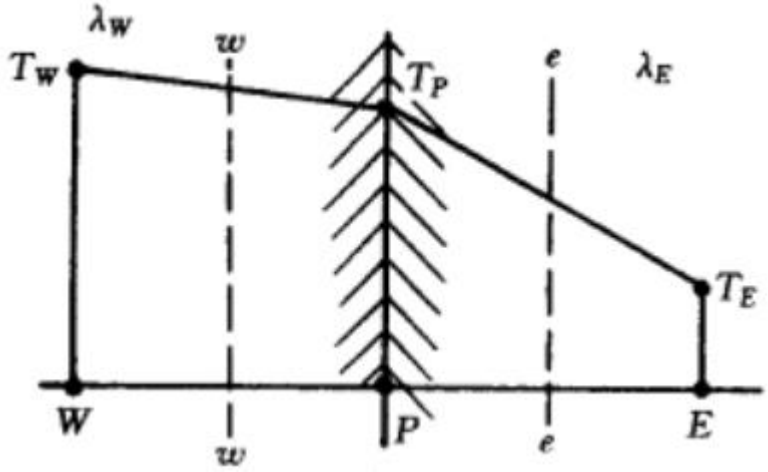
当计算区域中导热系数发生阶跃性变化时，对阶跃面有两种处理方法：

（1） 把物性阶跃面作为控制容积的分界面（图九）。这时采用调和平均计算的结果比算术平均时更准确。

（2） 把物性的阶跃面设置成一个节点的位置（图十）。数值计算的实例表明此时阶跃面上的热流密度的计算结果要比采用第一种布置方式更为准确。这是因为此时在阶跃面两侧的温度梯度是不同的。用第二种方法时，物性阶跃面两侧的五毒梯度单独计算，有利于提高计算精度。



图九



图十

6 问题二的求解

6.1 想法一

6.1.1 问题分析

就本题第二问而言，未知量应为材料 2 厚度。根据第一问模型可知，材料 2 的厚度同内空气层内表面温度与时间三个量存在三维函数关系。（1）60min 时，假人皮肤外侧温度不超过 47℃（2）55min 时，假人皮肤外侧温度不超过 44℃（3）0.6mm<d2<25mm，在此三个约束条件下，材料 2 厚度或产生一个满足条件的范围，考虑到控制成本的问题，取此范围中的 min 值，即为II层的最优厚度。

6.1.2 模型求解

差分方程与问题一显示格式相同，为

$$t_i^{k+1} = Fo_\Delta(t_{i-1}^k + t_{i+1}^k) + (1-2Fo_\Delta)t_i^k$$

初始及边界条件如下：

X=0 时，T=65℃；

t=0 时，T=37℃；

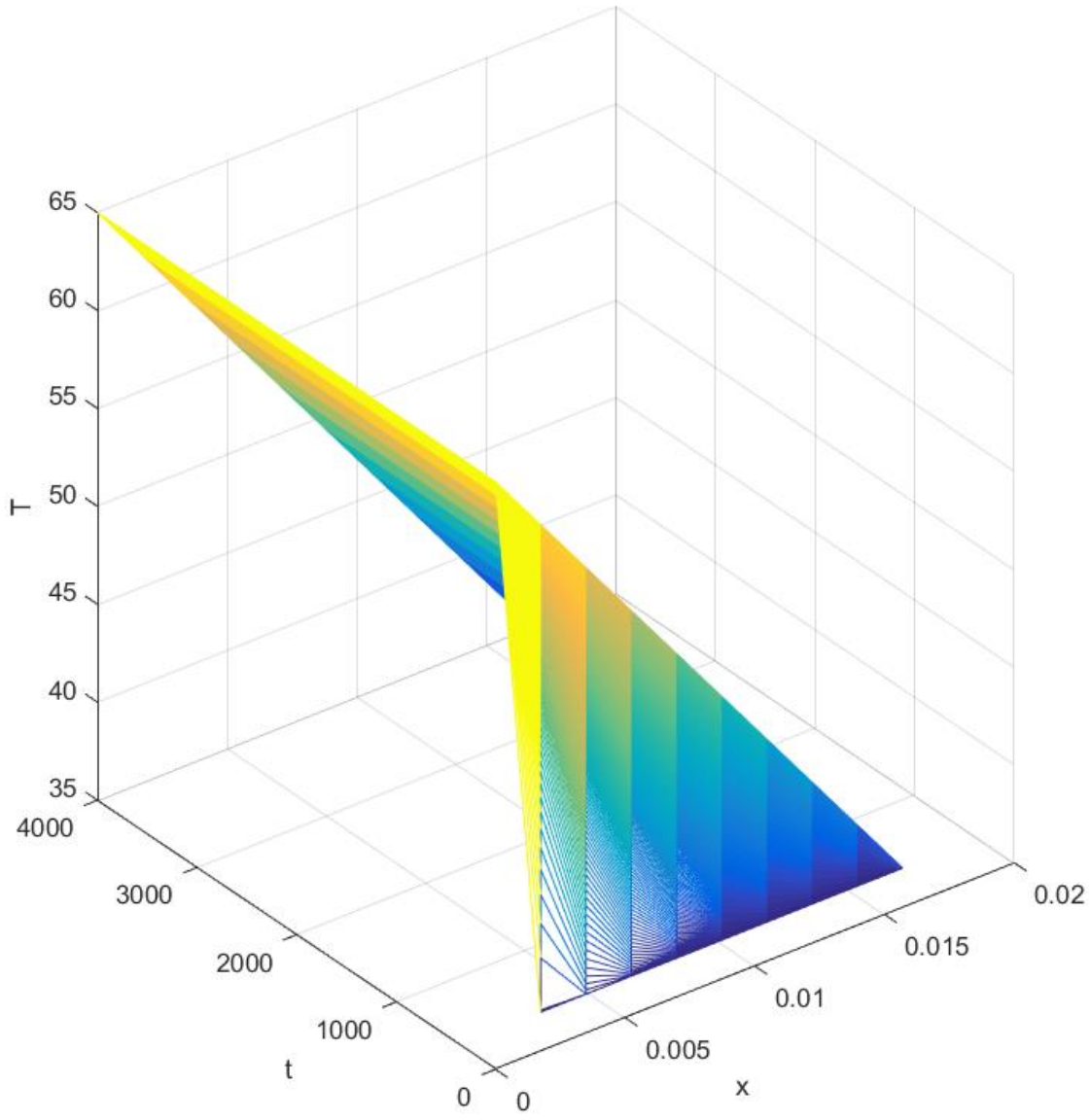
x=15.2mm 时，T = 37℃

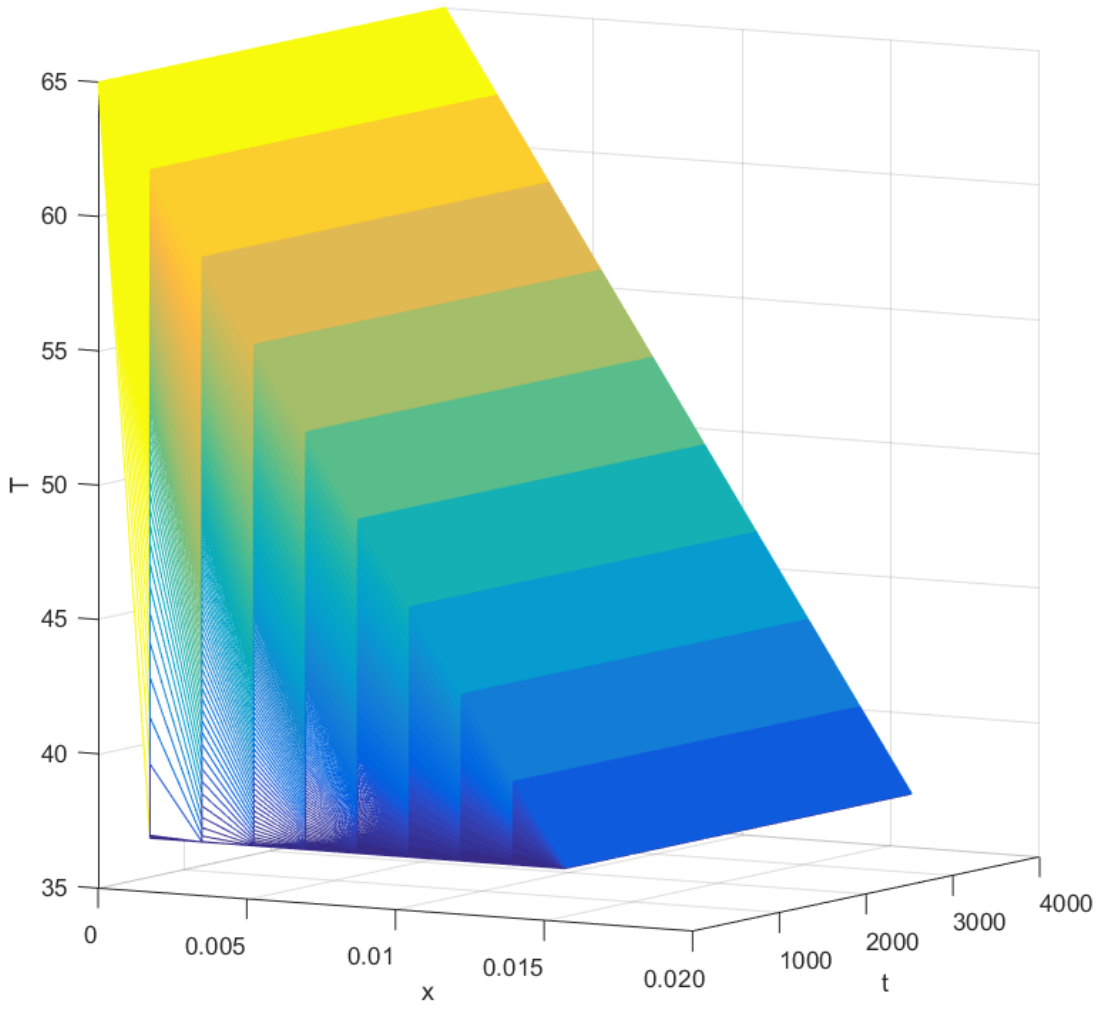
稳定性限制：

$$Fo_\Delta \le \frac{1}{2Bi_\Delta + 2}$$

求解过程见附录三代码

6.1.3 求解结果


65
60
55
50
45
40
35
T
4000
3000
2000
1000
0
t
0
0.005
0.01
0.015
0.02
x



6.2 想法二

6.1.1 问题分析

6.2.2 模型求解

6.2.3 求解结果

7 问题三的求解

7.1 问题分析

就本题第三问而言，我们在第二问的基础上利用双重循环，对每一个给定的 II 层厚度可得到符合条件的 IV 层厚度范围，取相应的步长使 II 层厚度均匀变化，得到两者对应关系。又因材料成本远大于内空气层成本，因此优先考虑选择 II 层最小厚度作为最优解，同时可求得对应 IV 层厚度的最优解。

7.2 模型求解

7.3 求解结果

8 模型评价与总结

模型优点：

1，该体系理论基础扎实，计算过程中的每一步都有相应的学科理论支持

2，过程中显式和隐式都用到了：对于隐式，其步长无限制，所以普适性更大

模型不足：

1，对于隐式，因其需要解多元线性方程组，所以实现过程更为复杂

References

[1] 张靖周、常海萍，传热学，北京：科学出版社，2009 年 1 月第一版

[2] 苏亚欣，传热学，武汉：华中科技大学出版社，2009 年 12 月第一版

[3] 陶文铨，数值传热学（第 2 版），西安：西安交通大学出版社，2001 年 5 月第一版

[4] 郭宽良，数值计算传热学，合肥：安徽科学技术出版社，1987 年 5 月第一版

[5] 陶文铨，传热学，西安：西北工业大学出版社，2006 年 12 月第一版

[6] 杨世铭，陶文铨，传热学第四版，北京：高等教育出版社，2006 年 8 月第四版

[7] Mingwei Tian、Sukang Zhu、Ning Pan、Lijun Qu、Guangting Han、Fukui Pan，Effects of layering sequence on thermal response of multilayer fibrous materials: Unsteady-state cases，Experimental Thermal and Fluid Science，Volume 41：Pages 143-148，September 2012

Appendices

在论文纸质版附录中，应给出参赛者实际使用的软件名称、命令和编写的全部计算机源程序（若有的话）。

附录主要包含：

（1）计算程序

（2）次要而繁杂的数据（不包括题目数据）

（3）繁琐的数值结果、图表等

（4）应用软件(如 Excel, Spss)无程序行，

可以用截图代替。

附录一，问题一求解代码（显式）：

主函数:

```
function Asolution
a1 =48.06;
b1 =764.3;
c1 =1522;
layer_d=[0.0006,0.006,0.0036,0.005];
xspan=[0 sum(layer_d)];
tspan=[0 5400];
ngrid=[540000 11];%t,x
k=[0.082,0.37,0.045,0.028];
k_transition=[0.134,0.08,0.0345];
C=[1377,2100,1726,1005];
ro=[300,862,74.2,1.18];
f=@(x)37;%IV 层内壁温度
g1=@(t)75;%I 层外壁温度
d1=ngrid(1)/range(tspan);
g2=@(t)a1*exp(-((t/d1-b1)/c1)^2)+0.05;%温度补偿，确保拟合函数可以达到定温 48.08
[T,x,t]=cal_fun(f,g1,g2,xspan,tspan,ngrid,layer_d,k,k_transition,C,ro);

[x,t]=meshgrid(x,t);
mesh(x,t,T);
xlabel('x')
ylabel('t')
zlabel('T')

output_matrix=zeros(range(tspan),ngrid(2));
output_count=1;
for i=1:ngrid(1)
   if(mod(i-1,100)==0)
      output_matrix(output_count,:)=T(i,:);
      output_count=output_count+1;
   end
end
 s=xlswrite('E:\同济\大三上\数模\program\A4（体温曲线分段）
\problem1.xlsx',output_matrix);
```

子函数:

```
function [U,x,t]=cal_fun(f,g1,g2,xspan,tspan,ngrid,layer_d,k,k_transition,C,ro)
n=ngrid(1);
m=ngrid(2);
h=range(xspan)/(m-1);
x=linspace(xspan(1),xspan(2),m);
t_step=range(tspan)/(n-1);
t=linspace(tspan(1),tspan(2),n);
x_step=h;
U=zeros(ngrid);
U(1,:)=f(x);
U(:,1)=g1(t);
temp_flag=0;
for i=1:n
  if(temp_flag==0)
   U(i,m)=g2(i);
  else
   U(i,m)=48.08;
  end

  if (U(i,m)>=48.08)
     temp_flag=1;
  end
end

for j=2:n
   for i=2:m-1
      if (j<(layer_d(1))/x_step-1)%不同材质的热传导率，比热容，密度
         para=[k(1),C(1),ro(1)];
      elseif  (j>((layer_d(1))/x_step-1)&&j<(layer_d(1)/x_step))
         para=[k_transition(1),C(1),ro(1)];
      elseif (j>((layer_d(1))/x_step)&&j<(layer_d(1)/x_step+1))
         para=[k_transition(1),C(2),ro(2)];
      elseif (j>(layer_d(1)/x_step+1)&&j<((layer_d(1)+layer_d(2))/x_step-1))
         para=[k(2),C(2),ro(2)];
      elseif (j>((layer_d(1)+layer_d(2))/x_step-1)&&j<((layer_d(1)+layer_d(2))/x_step))
         para=[k_transition(2),C(2),ro(2)];
      elseif (j>((layer_d(1)+layer_d(2))/x_step)&&j<((layer_d(1)+layer_d(2))/x_step+1))
         para=[k_transition(2),C(3),ro(3)];
      elseif
(j>((layer_d(1)+layer_d(2))/x_step+1)&&j<((layer_d(1)+layer_d(2)+layer_d(3))/x_step-1))
         para=[k(3),C(3),ro(3)];
      elseif (j>((layer_d(1)+layer_d(2)+layer_d(3))/x_step-
1)&&j<((layer_d(1)+layer_d(2)+layer_d(3))/x_step))
         para=[k_transition(3),C(3),ro(3)];
      elseif
(j>((layer_d(1)+layer_d(2)+layer_d(3))/x_step)&&j<((layer_d(1)+layer_d(2)+layer_d(3))/x_st
ep)+1)
         para=[k_transition(3),C(4),ro(4)];
      else
         para=[k(4),C(4),ro(4)];
      end
      a=para(1)/(para(2)*para(3));
```

```
        Fo=(a*t_step)/x_step^2;
        Bi=(h*x_step)/para(1);
        temp1=1/(2*Bi+2);
        if Fo>temp1
            fprintf('error')
        end
        U(j,i)=Fo*(U(j-1,i-1)+U(j-1,i+1))+(1-2*Fo)*U(j-1,i);
    end
end
```

附录二，问题一求解代码（隐式）：

主函数:

```
function Asolution
 layer_d=[0.0006,0.006,0.0036,0.005];
 k=[0.082,0.37,0.045,0.028];
 k_transition=[0.134,0.08,0.0345];%待定
 C=[1377,2100,1726,1005];
 ro=[300,862,74.2,1.18];

 x_step=0.0001;
 t_set=[0 100 1];%起始 终点 步长
 f=@(x)37;
 boundary_heat=@(t)75;
 boundary_body=@(t)37;
 [T,x,t]=cal_fun(layer_d,k,k_transition,C,ro,x_step,t_set,f,boundary_heat,boundary_body);
[x,t]=meshgrid(x,t);
mesh(x,t,T);
xlabel('x')
ylabel('t')
zlabel('T')
```

子函数:

```
function
[T,x,t]=cal_fun(layer_d,k,k_transition,C,ro,x_step,t_set,f,boundary_heat,boundary_body)
x_len=(layer_d(1)+layer_d(2)+layer_d(3)+layer_d(4))/x_step;
t_len=(t_set(2)-t_set(1))/t_set(3);
t=linspace(t_set(1),t_set(2),t_len);
x=linspace(0,(layer_d(1)+layer_d(2)+layer_d(3)+layer_d(4)),x_len);
T=zeros(t,x);

T(:,1)=boundary_heat(t);
T(:,x_len)=boundary_body(t);

T(1,:)=f(x);

for i=2:t_len
    U=zeros(x_len-2,x_len-2);
    constant_temp=zeros(x_len-2);
    for j=2:x_len-1
```

```
        if (j<(layer_d(1))/x_step-1)
            para=[k(1),C(1),ro(1)];
        elseif  (j>((layer_d(1))/x_step-1)&&j<(layer_d(1)/x_step))
            para=[k_transition(1),C(1),ro(1)];
        elseif (j>((layer_d(1))/x_step)&&j<(layer_d(1)/x_step+1))
            para=[k_transition(1),C(2),ro(2)];
        elseif (j>(layer_d(1)/x_step+1)&&j<((layer_d(1)+layer_d(2))/x_step-1))
            para=[k(2),C(2),ro(2)];
        elseif (j>((layer_d(1)+layer_d(2))/x_step-1)&&j<((layer_d(1)+layer_d(2))/x_step))
            para=[k_transition(2),C(2),ro(2)];
        elseif (j>((layer_d(1)+layer_d(2))/x_step)&&j<((layer_d(1)+layer_d(2))/x_step+1))
            para=[k_transition(2),C(3),ro(3)];
        elseif
(j>((layer_d(1)+layer_d(2))/x_step+1)&&j<((layer_d(1)+layer_d(2)+layer_d(3))/x_step-1))
            para=[k(3),C(3),ro(3)];
        elseif (j>((layer_d(1)+layer_d(2)+layer_d(3))/x_step-
1)&&j<((layer_d(1)+layer_d(2)+layer_d(3))/x_step))
            para=[k_transition(3),C(3),ro(3)];
        elseif
(j>((layer_d(1)+layer_d(2)+layer_d(3))/x_step)&&j<((layer_d(1)+layer_d(2)+layer_d(3))/x_st
ep)+1)
            para=[k_transition(3),C(4),ro(4)];
        else
            para=[k(4),C(4),ro(4)];
        end

        AE=para(1)/x_step;
        AW=para(1)/x_step;
        AP=(para(3)*para(2)*x_step)/t_set(3);

        if(j==2)
            U(j-1,j-1)=1;
            constant_temp(j-1)=T(i,j-1)*AE+T(i-1,j)*AP;
            U(j-1,j)=-AW;
        elseif(j==x_len-1)
            U(j-1,j-1)=1;
            constant_temp(j-1)=T(i,j+1)*AW+T(i-1,j);
            U(j-1,j-2)=-AE;
        else
            U(j-1,j-1)=1;
            constant_temp=T(i-1,j)*AP;
            U(j-1,j-2)=-AE;%矩阵对角线前一格
            U(j-1,j)=-AW;%矩阵对角线后一格
        end
    end
    y_temp=constant_temp*U^-1;
    for l=2:x_len-1
        T(i,l)=y_temp(l-1);
    end

end

end
```

附录三，问题二求解代码（显式）：

主程序:

```
function mypdesolution
c=1;
a1 =48.06;
b1 =764.3;
c1 =1522;
layer_d=[0.0006,0.006,0.0036,0.0055];
xspan=[0 sum(layer_d)];
tspan=[0 4000];
ngrid=[400000 10];%t,x
k=[0.082,0.37,0.045,0.028];
k_transition=[0.134,0.08,0.0345];
C=[1377,2100,1726,1005];
ro=[300,862,74.2,1.18];
f=@(x)37;
g1=@(t)65;
d1=ngrid(1)/range(tspan);
g2=@(t)37;%(t)a1*exp(-((t/d1-b1)/c1)^2);
[T,x,t]=rechuandao(c,f,g1,g2,xspan,tspan,ngrid,layer_d,k,k_transition,C,ro);
[x,t]=meshgrid(x,t);
fprintf('矩阵大小\%f,%f',size(T))
fprintf('x=%f',size(x))
fprintf('t=%f',size(t))
mesh(x,t,T);
xlabel('x')
ylabel('t')
zlabel('T')
output_matrix=zeros(range(tspan),ngrid(2));
output_count=1;
for i=1:ngrid(1)
   if(mod(i-1,100)==0)
      output_matrix(output_count,:)=T(i,:);
      output_count=output_count+1;
   end
end
% data_cell=mat2cell(output_matrix,ones(range(tspan),1),ones(ngrid(2),1));
%
title('time','x=1.52mm','x=3.04mm','x=4.56mm','x=6.08mm','x=7.6mm','x=9.12mm','x=10.64m
m','x=12.16mm','x=13.68mm','x=15.2mm');
 %result=[title];
 s=xlswrite('E:\同济\大三上\数模\program\A3(基础模型) - 副本
\problem2.xlsx',output_matrix);
```

子程序

```
function [U,x,t]=rechuandao(c,f,g1,g2,xspan,tspan,ngrid,layer_d,k,k_transition,C,ro)
n=ngrid(1);
m=ngrid(2);
h=range(xspan)/(m-1);
x=linspace(xspan(1),xspan(2),m);
t_step=range(tspan)/(n-1);
```

```
t=linspace(tspan(1),tspan(2),n);
x_step=h;
r=c^2*t_step/h^2;
% if r>0.5
%     error('jakfjl')
% end


U=zeros(ngrid);
U(1,:)=f(x);
U(:,1)=g1(t);
%U(:,m)=g2(t);
for i=1:n%range(tspan)

   U(i,m)=g2(i);
   fprintf('uim=%f',U(i,m))

end




for j=2:n
   for i=2:m-1
      if (j<(layer_d(1))/x_step-1)
         para=[k(1),C(1),ro(1)];
      elseif  (j>((layer_d(1))/x_step-1)&&j<(layer_d(1)/x_step))
         para=[k_transition(1),C(1),ro(1)];
      elseif (j>((layer_d(1))/x_step)&&j<(layer_d(1)/x_step+1))
         para=[k_transition(1),C(2),ro(2)];
      elseif (j>(layer_d(1)/x_step+1)&&j<((layer_d(1)+layer_d(2))/x_step-1))
         para=[k(2),C(2),ro(2)];
      elseif (j>((layer_d(1)+layer_d(2))/x_step-1)&&j<((layer_d(1)+layer_d(2))/x_step))
         para=[k_transition(2),C(2),ro(2)];
      elseif (j>((layer_d(1)+layer_d(2))/x_step)&&j<((layer_d(1)+layer_d(2))/x_step+1))
         para=[k_transition(2),C(3),ro(3)];
      elseif
(j>((layer_d(1)+layer_d(2))/x_step+1)&&j<((layer_d(1)+layer_d(2)+layer_d(3))/x_step-1))
         para=[k(3),C(3),ro(3)];
      elseif (j>((layer_d(1)+layer_d(2)+layer_d(3))/x_step-
1)&&j<((layer_d(1)+layer_d(2)+layer_d(3))/x_step))
         para=[k_transition(3),C(3),ro(3)];
      elseif
(j>((layer_d(1)+layer_d(2)+layer_d(3))/x_step)&&j<((layer_d(1)+layer_d(2)+layer_d(3))/x_st
ep)+1)
         para=[k_transition(3),C(4),ro(4)];
      else
         para=[k(4),C(4),ro(4)];
      end
      a=para(1)/(para(2)*para(3));
      Fo=(a*t_step)/x_step^2;
      Bi=(h*x_step)/para(1);
      temp1=1/(2*Bi+2);
```

```
        if Fo>temp1
            fprintf('error')
        end
        U(j,i)=Fo*(U(j-1,i-1)+U(j-1,i+1))+(1-2*Fo)*U(j-1,i);
    end
end
```