

基于多层织物热传递模型的高温作业专用服装设计

摘 要

工作人员在高温下进行作业时，若温度达到了某一阈值后，便会其身体内部及皮肤表面造成损伤。因此，热防护服便是保护其免受高温损伤的关键。热防护服往往由多层材料构成，本文主要通过数值方法对热防护服的传热性能进行了研究，对于热防护服的不同层设计有着理论指导的意义。

首先，本文通过模型的对称性，将模型视为一维问题，先建立了各层的一维热传导方程，并且对于空气层，我们综合考虑了热传导与热辐射。利用环境的恒定温度与所测得的初始温度，从而建立热传导方程的边界条件，再以古典隐式差分格式进行离散化后，将微分方程转换为差分方程，最终求得各个位置（点间隔 0.1mm）及各个时刻（时间间隔 1s）的温度数值解。

其次，我们考虑了不同层织物厚度对于皮肤表面层温度的影响，根据测试假人能够保持体温的特性，我们又建立了第 V 层——皮肤层的一维热传导方程，从而分别分析了第 II 层织物层与第 IV 层空气层的厚度变化带来的表皮温度的变化。随后，我们根据已知的发生烧伤的阈值，在 60 分钟的高温工作时间内确保人体不发生烧伤的情况下，基于上述建立的微分方程模型，通过人工蜂群算法（ABC），对第 II 层的织物厚度进行了数值优化，从而得出第 II 层的最优厚度为：4.683mm。

此外，我们又同时对第 II 层织物层与第 IV 层空气层的最优厚度进行了数值优化。我们在此同时考虑了服装厚度与服装质量两个因素，提出了厚度权重系数 α ，从而在已知的安全条件及 30 分钟的高温工作时间的条件下，对带权的质量厚度目标函数进行优化，得到第 II 层与第 IV 层的最优厚度分别为：5.2534mm，6.4mm。

最后，我们对于上述建立的微分方程模型与优化模型做了总结，并且对于该模型的优点与不足做了简要的分析。

关键词：热防护服的设计 热传导方程 微分方程数值解 数值最优化 人工蜂群算法

一 问题提出

1.1 题目背景与研究意义

防护织物是纺织产品向高性能发展的重要方向之一。各类防护织物中，热防护织物的应用最为广泛，其作用主要是保护人体不受各种热的伤害，如对流热、传导热、辐射热、熔融金属溅射以及热蒸汽或热气体等的伤害^[6]。防护服装通常由三层织物材料构成，一般为外壳、防水层和隔热层，其中外壳与环境接触，服装最内层与皮肤之间还存在空隙，称为空气层。

人的皮肤对热十分敏感，当人体皮肤的热流密度达到 2.68 J/cm^2 ，即皮肤温度达到 44°C 时，人就会有灼痛感^[5]。因此热防护服必须具有以下性能：

- ①阻燃性；
- ②完整性(遇热或熔融后服装保持完整)；
- ③隔热性(阻止热的转移)；
- ④拒液性(阻止油、溶剂、水及其它液体渗入织物)。

本问题主要对热防护服的隔热性能进行研究并建立模型，从科学的角度分析织物内部的热力学性质，进而结合设计目标决定织物的厚度、孔隙率和材质等物理学参数。

1.2 问题重述

为设计高温作业专用服装，将体内温度控制在 37°C 的假人放置在实验室的高温环境中，测量假人皮肤外侧的温度。为了降低研发成本、缩短研发周期，利用数学模型来确定假人皮肤外侧的温度变化情况，并解决以下问题：

(1) 进行实验测试，环境温度为 75°C ，工作时间为90分钟，专用服装各层织物材料的相关参数如下表：

参数 分层	密度 (kg/m^3)	比热 ($\text{J}/(\text{kg} \cdot ^\circ\text{C})$)	热传导率 ($\text{W}/(\text{m} \cdot ^\circ\text{C})$)	厚度 (mm)
I 层	300	1377	0.082	0.6
II 层	862	2100	0.37	6
III 层	74.2	1726	0.045	3.6
IV 层	1.18	1005	0.028	5

测量得到假人皮肤外侧的温度（见附件2）。

要求：建立数学模型，计算温度分布，并生成温度分布的Excel文件（文件名为problem1.xlsx）。

(2) 当环境温度为 65°C 、IV层的厚度为5.5 mm时，确定II层的最优厚度，满足

工作时间	假人皮肤外侧温度
55min	$\leq 44^\circ\text{C}$
60min	$\leq 47^\circ\text{C}$

(3) 当环境温度为 80°C 时，确定II层和IV层的最优厚度，满足

工作时间	假人皮肤外侧温度
25min	$\leq 44^\circ\text{C}$
30min	$\leq 47^\circ\text{C}$

二 问题分析

此部分，对本文要解决的问题进行建模前的初步分析准备，以求为后文建模部分提供逻辑结构。

2.1 问题一的分析

问题 1 要求建立多层热防护服的热传递模型。考虑到题目中实验环境的温度相对较低（相比于闪火环境下的 2000K 高温），且不存在高温热源，因此将防护服外壳表面的温度视为与环境保持一致，并向内发生热传导。

III 层与假人皮肤表面存在空隙，但由于此空气层厚度较小，故不考虑热对流问题，但由于热传递介质发生较大改变，此处可考虑热辐射和热传导同时存在的情况。因此，该题最终可以简化为一维多介质的热传导问题。对该正向问题的求解，在实际的隔热服设计过程中，为正问题。

2.2 问题二和问题三的分析

人的皮肤对热十分敏感，当人体皮肤的热流密度达到 2.68 J/cm^2 ，即皮肤温度达到 44°C 时，人就会有灼痛感。问题 2 和问题 3 均要求在高温作业时，确保人体表面体温在一定的时间内未达到危险值的最优厚度（单层/多层）。

问题 2、3 需要分析织物层厚度对皮肤表面温度的影响，为了使已建立的热传导模型具备定解条件（DP），考虑到实验过程中假人体内温度恒为 37°C ，进而可以考虑皮肤层（介于假人皮肤外侧和体内之间）的引入，作为皮肤内侧的边界条件。

优化模型主要基于问题 1 已经建立的热传导方程，从而对有约束下的目标函数（厚度/舒适度）进行优化，找到全局最优解后，得出本题所要求的最优厚度。在实际的隔热服设计过程中，该问题为反问题。

三 问题假设

1. 假设实验环境为密闭空间，在同一测试中环境温度处处相等且恒定不变，进而由温度场的连续性可知，热防护服表面温度与环境温度一致，且保持不变；
2. 由于衣物贴身穿戴，假人穿戴防护服进入实验环境，故可以认为初始时刻织物层温度与假人体温一致。
3. 热传递垂直于皮肤方向进行，因而可以视为一维的；
4. 热防护服的织物各项同性；
5. 由于测试环境温度相对较低，热传递过程中，织物的结构几乎不会发生改变，即认为织物的比热、热传导率以及厚度不变，织物不会发生热溶解；
6. 由于防护服外壳阻挡了大部分的辐射，因此在织物层及空气层传递过程中，热辐射可以忽略；
7. 测试中空气层的厚度值不超过 6.4 mm ，热对流影响小，因而不考虑热对流；
8. 由于采用假人实验，本模型仅考虑热传递，包括热辐射及热传导现象，不考虑湿传递，即忽略水汽、汗液的影响；
9. 假设暖体假人内部存在一维持体温的热源其位于假人皮肤表面 25cm 处。
10. 假设热传导和辐射热传递到织物的过程中是均匀的；
11. 织物之间，织物与空气层之间，空气层与皮肤之间的温度变化都是连续变化

的，但温度梯度是跳跃的。

四 符号说明

类型	符号	单位	符号意义
下标	sur	—	环境
	skin	—	皮肤表面
	int	—	假人体内
	x	mm	横坐标
	l	—	l 等于 I/II/III/IV/V 介质特性指标
	n	—	n 等于 1/2/3/4/5 介质交界序号
	i	—	i 等于 1、2…N 位置序号
	j	—	j 等于 1、2…M 时间序号
	M	—	空间划分格数
	N	—	时间划分格数
	I、II、III IV、V	—	I 层、II 层、III 层 IV层、V层
	III,IV	—	III 层对IV层作用量
物理量	D	mm	介质的厚度
	q	W/m^2	热通量
	c	$J/(Kg \cdot ^\circ C)$	各层介质的比热
	ρ	Kg/m^3	各层介质的密度
	k	$W/(m \cdot ^\circ C)$	各层介质的热传导率
	T	$^\circ C$	温度
	t	s	时间
	τ	s	差分格式中两时刻点间的时间差
	h	mm	差分格式中两格点间距离

五 模型的建立和求解

5.1 问题一的建模和求解

5.1.1 关于多层热防护服系统一维热传递模型的简化说明

为了简化模型，不考虑人体三维特征对热防护性能的影响，因而可以将假人考虑为圆柱体。（参考图 5.1）由于实验测试中环境温度相对较低（相比于闪火环境下 2000K 高温），仅有 350K 左右，故可以不考虑织物材料热属性的变化，建立一维径向传热模型^[4]。

在图 5.2 给出了由热防护服及热防护服一皮肤间的空气层组成的系统，其中，热防护服包括 I 层（外壳）、II 层（防水层）、III 层（隔热层），织物和皮肤之间的 IV 层（空气层）厚度小于 6.4mm。

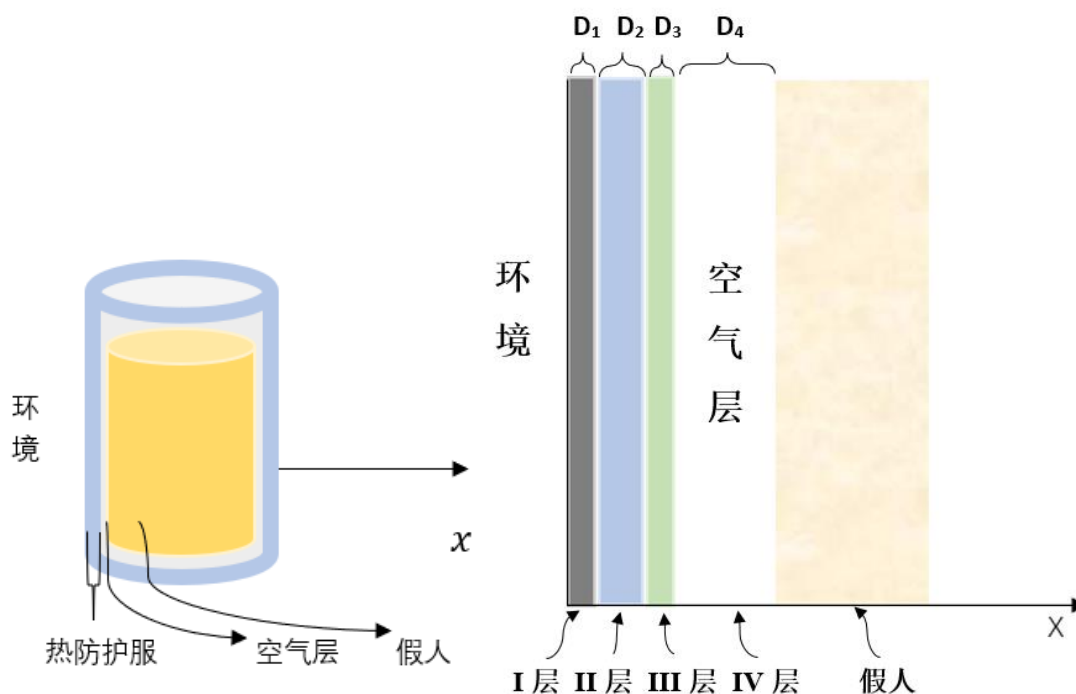


图 5.1

图 5.2

5.1.2 多层热防护服系统一维热传递模型的建立

（一）建立热传递方程

热防护服有三层性质不同的织物材料组成，由假设 6，可以不考虑织物层间的热辐射影响，认为在织物层中只有热传导。因此，三层的热防护服热传递模型可以写成：

$$\begin{cases} c_I \rho_I \frac{\partial T}{\partial t} = k_I \frac{\partial^2 T}{\partial x^2}, x \in (0, D_1) & (1) \\ c_{II} \rho_{II} \frac{\partial T}{\partial t} = k_{II} \frac{\partial^2 T}{\partial x^2}, x \in (D_1, D_1 + D_2) & (2) \end{cases}$$

$$c_{III}\rho_{III}\frac{\partial T}{\partial t} = k_{III}\frac{\partial^2 T}{\partial x^2}, x \in (D_1 + D_2, D_1 + D_2 + D_3) \quad (3)$$

考虑到 III 层对空气的热辐射，IV 层（空气层）中的热传递方程为

$$c_{IV}\rho_{IV}\frac{\partial T}{\partial t} = k_{IV}\frac{\partial^2 T}{\partial x^2} - \frac{\partial q_{III,IV}}{\partial x} \quad (4)$$

$$x \in (D_1 + D_2 + D_3, D_1 + D_2 + D_3 + D_4)$$

在该式中，根据假设 7，忽略空气层的对流，仅将空气层视为辐射介质，得到^[2]：

$$\frac{\partial q_{III,IV}}{\partial x} = A_{III} \left[\varepsilon_{III} T^4(x, t)|_{x=D_1+D_2+D_3} - \varepsilon_{IV} T^4(x, t) \right] \quad (5)$$

设实验时长为 t_{exp} ，则上面各式中 t 满足 $t \in (0, t_{exp})$

其中， c_I 、 c_{II} 、 c_{III} 、 c_{IV} 分别是 I 层、II 层、III 层和 IV 层的比热容；

ρ_I 、 ρ_{II} 、 ρ_{III} 、 ρ_{IV} 分别是 I 层、II 层、III 层和 IV 层的密度； D_1 、 D_2 、 D_3 、

D_4 分别是 I 层、II 层、III 层和 IV 层的厚度； k_I 、 k_{II} 、 k_{III} 、 k_{IV} 分别是 I 层、II 层、III 层和 IV 层的热传导率； T 是温度； t 是时间； x 是水平坐标；

$q_{III,IV}$ 是来自 III 层背面的热辐射； A_{III} 是 III 层和 IV 层的接触面积； ε_{III} 、 ε_{IV} 分别是 III 层和 IV 层的辐射系数。

（二）初始条件与边界条件

根据假设 1，织物层的初始温度与假人温度相同，作为初始温度的边界条件：

$$T(x, 0) = T_{skin}(0) \quad (6)$$

根据假设 2，织物层的左边界温度与环境温度相同，作为左边界条件为：

$$T(0, t) = T_{sur} \quad (7)$$

并且，I 层与 II 层之间的交界：

$$-k_I \frac{\partial T}{\partial x} \Big|_{x=D_1} = -k_{II} \frac{\partial T}{\partial x} \Big|_{x=D_1} \quad (8)$$

II 层和 III 层之间的交界：

$$-k_{II} \frac{\partial T}{\partial x} \Big|_{x=D_1+D_2} = -k_{III} \frac{\partial T}{\partial x} \Big|_{x=D_1+D_2} \quad (9)$$

III 层和 IV 层（空气层）之间的交界：

$$-k_{III} \frac{\partial T}{\partial x} \Big|_{x=D_1+D_2+D_3} = -k_{IV} \frac{\partial T}{\partial x} \Big|_{x=D_1+D_2+D_3} \quad (10)$$

IV 层（空气层）的右边界条件为：

$$T(D_1 + D_2 + D_3 + D_4, t) = T_{skin}(t) \quad (11)$$

其中, T_{skin} 表示假人皮肤表面的温度, 该函数仅与时间 t 有关; T_{sur} 表示环境温度, 在同一测试中为常数;

5.1.3 多层热防护服系统一维热传递模型的求解

对于该问题中的热传导方程, 本文采用古典隐式差分格式将连续的微分方程进行离散化, 从而计算得出的数值解近似为微分方程解。

(一) 微分方程及边界条件的离散化

原模型空气层中对热辐射的考虑, 使得模型方程由线性方程跃至四次方程, 模型求解难度陡增, 而在空气层中占据主导的热传递方式为热传导, 因此忽略空气层热辐射的影响, 将 (4) 式简化为

$$c_{IV}\rho_{IV} \frac{\partial T}{\partial t} = k_{IV} \frac{\partial^2 T}{\partial x^2}, x \in (D_1 + D_2 + D_3, D_1 + D_2 + D_3 + D_4) \quad (12)$$

1. 同介质中微分方程的离散:

$$c_l \rho_l \frac{\partial T}{\partial t} = k_l \frac{\partial^2 T}{\partial x^2} \quad (13)$$

其中下标 $l = I \text{ II III IV}$. 将同一介质中的厚度分为 $M_l + 1$ 个格点, 每个格点之间的距离为

$$h_l = \frac{D_l}{M_l} \quad (14)$$

将时间离散为 $N+1$ 个时刻, 每个时刻之间的时间差为

$$\tau = \frac{t_{exp}}{N} \quad (15)$$

经古典隐式差分后为:

$$\frac{T_i^{j+1} - T_i^j}{\tau} = \frac{k_l}{c_l \rho_l} \frac{T_{i+1}^{j+1} - 2T_i^{j+1} + T_{i-1}^{j+1}}{h_l^2} \quad (16)$$

其中 $i = 0, 1, 2, \dots, M_l$, 为位置指标, $j = 0, 1, 2, \dots, N$, 为时间指标。由于每层织物的 k_l 、 c_l 、 ρ_l 为常量, 为了简化方程形式, 定义

$$r_l = \frac{k_l \tau}{c_l \rho_l h_l^2} \quad (17)$$

则同介质内的微分方程离散化为 (过程示意图 5.3):

$$-r_l T_{i-1}^{j+1} + (1 + 2r_l) T_i^{j+1} - r_l T_{i+1}^{j+1} = T_i^j \quad (18)$$

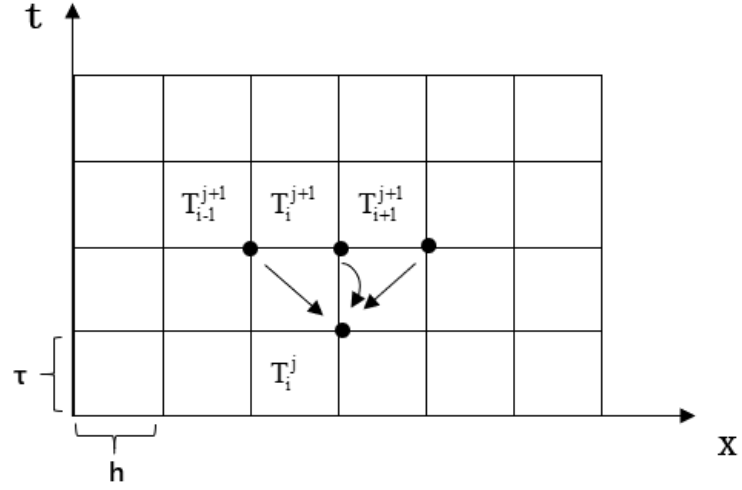


图 5.3 隐式差分方程的求解图示

2. 已知的边界条件的离散化

已知的边界条件为：

$$\left\{ \begin{array}{l} T(x, 0) = T_{skin}(0) \end{array} \right. \quad (19)$$

$$\left\{ \begin{array}{l} T(0, t) = T_{sur} \end{array} \right. \quad (20)$$

$$\left\{ \begin{array}{l} T(D_1 + D_2 + D_3 + D_4, t) = T_{skin}(t) \end{array} \right. \quad (21)$$

离散化后，得到：

$$\left\{ \begin{array}{l} T_i^0 = T_{skin}(0) \end{array} \right. \quad (22)$$

$$\left\{ \begin{array}{l} T_0^j = T_{sur} \end{array} \right. \quad (23)$$

$$\left\{ \begin{array}{l} T_{M_4}^j = T_{skin}(j\tau) \end{array} \right. \quad (24)$$

方便表达，在此定义 $M_1 = M_I$, $M_2 = M_I + M_{II}$, $M_3 = M_I + M_{II} + M_{III}$, $M_4 = M_I + M_{II} + M_{III} + M_{IV}$.

3. 介质层之间的交界离散化：

对于特定的两个介质交界上的格点，其差分后满足：

$$-k_l \frac{T_{M_n}^j - T_{M_n-1}^j}{h} = -k_{l+1} \frac{T_{M_n+1}^j - T_{M_n}^j}{h} \quad (25)$$

其中 $j=0, 1, 2 \cdots N$, $n=1, 2, 3, 4$. 整理后，可得对于介质交界上的格点，其可由同时刻左右两格点线性表出，即：

$$T_{M_n}^j = \frac{k_l/h_l}{k_l/h_l + k_{l+1}/h_{l+1}} T_{M_n-1}^j + \frac{k_{l+1}/h_{l+1}}{k_l/h_l + k_{l+1}/h_{l+1}} T_{M_n+1}^j \quad (26)$$

定义

$$u_l = \frac{k_l/h_l}{k_l/h_l + k_{l+1}/h_{l+1}}, \quad v_l = \frac{k_{l+1}/h_{l+1}}{k_l/h_l + k_{l+1}/h_{l+1}} \quad (27)$$

u_l 是交界 M_l 的左传导比, v_l 是交界 M_l 的右传导比。

由此, 古典隐式差分后, 其包含介质交界格点的离散化方程为 (过程示意图 5.4):

$$\left\{ \begin{aligned} -r_l T_{M_n-2}^{j+1} + (1 + 2r_l - u_l r_l) T_{M_n-1}^{j+1} - 0 \cdot T_{M_n}^{j+1} - v_l r_l T_{M_{l+1}}^{j+1} &= T_{M_n-1}^j \end{aligned} \right. \quad (28)$$

$$\left\{ \begin{aligned} -u_l T_{M_n-1}^{j+1} + T_{M_n}^{j+1} - v_l T_{M_{n+1}}^{j+1} &= 0 \end{aligned} \right. \quad (29)$$

$$\left\{ \begin{aligned} -u_l r_{l+1} T_{M_n-1}^{j+1} - 0 \cdot T_{M_n}^{j+1} + (1 + 2r_{l+1} - v_l r_{l+1}) T_{M_{n+1}}^{j+1} - r_{l+1} T_{M_{n+2}}^{j+1} &= T_{M_{n+1}}^j \end{aligned} \right. \quad (30)$$

由此, 便可根据以上离散化的差分方程求出其数值解。

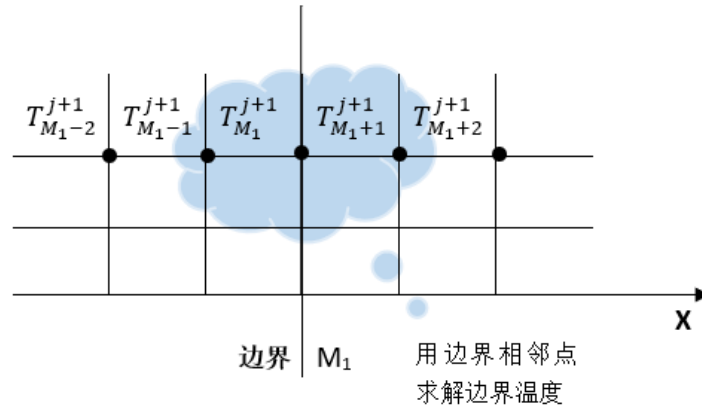


图 5.4 边界处的差分图示

(二) 模型的求解

1. 防护服各织物层的相关参数如下表:

参数 分层	密度 (kg/m ³)	比热 (J/(kg · °C))	热传导率 (W/(m · °C))
I 层	300	1377	0.082
II 层	862	2100	0.37
III 层	74.2	1726	0.045
IV 层	1.18	1005	0.028

表 1 各织物层的相关参数

2. 用隐式差分格式求解问题 1 的模型, 数值计算方法可以归纳为下式:

$$\mathbf{A} \begin{pmatrix} T_1^{j+1} \\ T_2^{j+1} \\ \vdots \\ T_{M-1}^{j+1} \end{pmatrix} = \mathbf{b} \quad (31)$$

其中 $j = 0, 1, 2, \dots, N-1$;

$$\mathbf{A} = \begin{bmatrix} 1+2r_1 & -r_1 & & & & \\ -r_1 & 1+r_1 & -r_1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -r_1 & 1+r_1 & -r_1 & \\ & & & \boxed{\begin{matrix} -r_1 & 1+2r_1-r_1u_1 & 0 & -r_1v_1 \\ & -u_1 & 1 & -r_1v_1 \\ & -r_2u_1 & 0 & 1+2r_2-r_2v_1 & -r_2 \end{matrix}} & & \\ & & & & -r_2 & 1+r_2 & -r_2 & \\ & & & & & \ddots & \ddots & \ddots \\ & & & & & & -r_2 & 1+r_2 & -r_2 \\ & & & & & & & -r_4 & 1+r_4 \end{bmatrix} \xrightarrow{\text{交界}}$$

$$\mathbf{b} = \begin{pmatrix} T_1^j + rT_0^{j+1} \\ T_2^j \\ T_3^j \\ \vdots \\ T_{M_1-1}^j \\ \boxed{0} \\ T_{M_1+1}^j \\ \vdots \\ T_{M_4-1}^j + rT_{M_4}^j \end{pmatrix} \xrightarrow{\text{I层与II层交界}}$$

用 MATLAB 求解后得到温度分布表（见文件 problem1.xlsx），并画出该温度场的三维图像及层间交界处的温度时间图（图 5.5）。

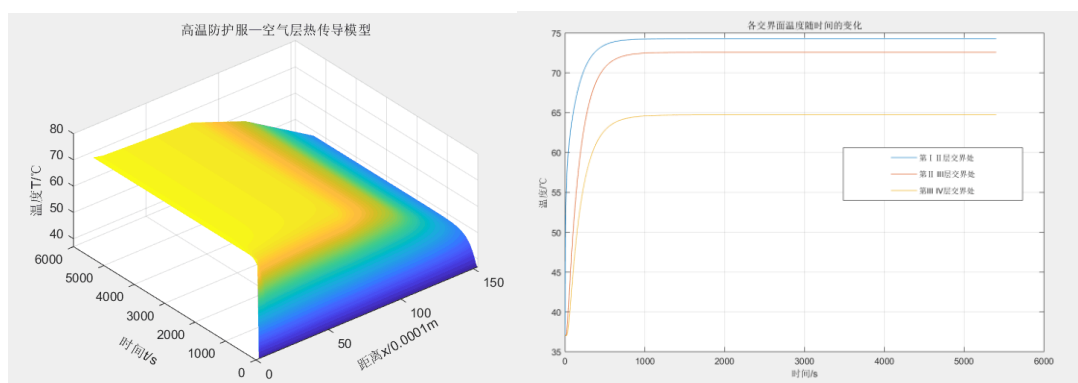


图 5.5 温度关于时间与位置的分布图（左），介质交界处温度随时间的变化图（右）

5.2 问题二的建模和求解

5.2.1 模型的改进

(一) 皮肤层（V层）的引入

问题2需要分析织物层厚度对皮肤表面温度的影响，为了使已建立模型具备定解条件（DP），考虑到实验过程中假人体内温度恒为 37°C ，因而可以将此温度视为右边界的边界条件，进而引入了皮肤层（介于假人皮肤外侧和体内之间），并记为V层。

查阅资料可知^[4]，皮肤表层的相关参数如下表（与织物层对比）：

参数 分层	密度 (kg/m^3)	比热 ($\text{J}/(\text{kg} \cdot ^{\circ}\text{C})$)	热传导率 ($\text{W}/(\text{m} \cdot ^{\circ}\text{C})$)
V层	1200	3598	0.255

表2 皮肤的相关参数

(二) 改进模型的建立和求解

1. 方程建立

V层的热传导方程为

$$c_V \rho_V \frac{\partial T}{\partial t} = k_V \frac{\partial^2 T}{\partial x^2}, \quad t \in (0, t_{exp}) \quad (32)$$

$$x \in (D_1 + D_2 + D_3 + D_4, D_1 + D_2 + D_3 + D_4 + D_5)$$

其中， D_5 为V层（皮肤层）的厚度。IV层（空气层）和V层（皮肤层）之间的交界处，满足：

$$-k_{IV} \frac{\partial T}{\partial x} \Big|_{x=D_1+D_2+D_3+D_4} = -k_V \frac{\partial T}{\partial x} \Big|_{x=D_1+D_2+D_3+D_4} \quad (33)$$

V层（空气层）的右边界条件为：

$$T(D_1 + D_2 + D_3 + D_4 + D_5, t) = T_{int} \quad (34)$$

其中 T_{int} 为假人体内的温度，恒为 37°C 。

2. 皮肤层厚度的确定

将皮肤层与空气层的交界处的温度视为假人皮肤外侧的温度，利用改进后的模型求解该点的温度分布函数，其与皮肤层，即V层的厚度相关。根据假设9，将皮肤层设为250cm，经由验证，在该边界条件下其皮肤最外层与问题一中测得的实际皮肤最外层温度接近，由此认为该假设合理，皮肤层的厚度 $D_5 = 250\text{mm}$ （示意图5.6）。

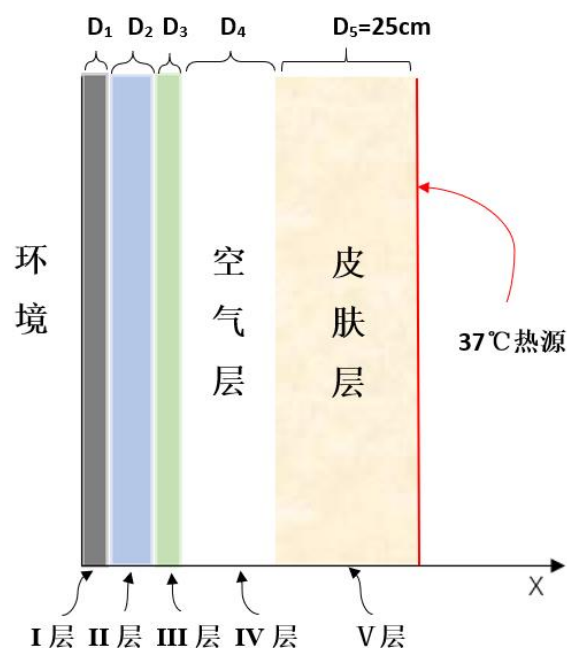


图 5.6

5.2.2 关于热防护服织物层厚度的优化模型

(一) 对于优化模型的说明

基于改进后“防护服-空气-皮肤”系统一维热传递模型，本文继续深入研究各层织物厚度对皮肤表面的温度的影响。

根据直观判断，织物越厚，热量传递至人体皮肤表面的时间就更长。这是由于织物厚度变大，热阻也会相应变大。当各个织物层的厚度增加时，隔热性能固然更佳，然而在实际设计时，防护服的厚度不可能依据要求无限制的加厚，过厚的防护服会导致工人无法在高温下进行精确快速地进行作业，而过薄的防护服又会致使工人的皮肤表面温度迅速升高，无法保障其作业时的人身安全。

由此，本节讨论了热防护服特定的织物层，其厚度的改变对于人体皮肤表面温度的影响，并且根据题目所给的安全要求，通过人工蜂群算法，找到了防护服 II 层的最优厚度。

(二) 织物及空气厚度的变化对于皮肤表面的影响：

首先，我们先基于上述模型，对其他条件相同时，第 II 层织物层的厚度以及第 IV 层空气层的厚度不同的情况下，其对应的 60min 及 55min 时皮肤表面的温度进行了计算，如下：

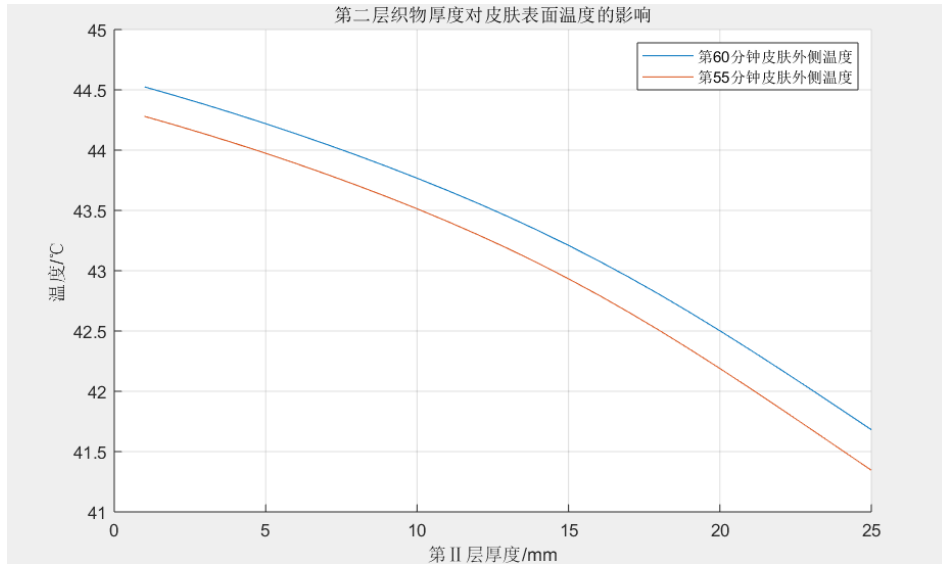


图 5.7 第 I 层厚度为 0.6mm 第 III 层厚度为 3.6mm 第 IV 层厚度为 5.5mm 皮肤层厚度为 25cm 时，皮肤在 60min 和 55min 时的温度与第 II 层温度的关系

由此分析，可以看出，第二层织物厚度增大时，皮肤外表面的温度会相应降低。一开始的变化趋势随织物厚度增大明显，随后其厚度对温度下降影响较小。

(三) 优化模型的建立

由于假人皮肤外侧的温度函数是单调递增的，因此问题 2 安全性要求可以用等价的数学语言描述如下：

工作 60 分钟时，假人皮肤外侧温度不超过 47°C，等价于

$$T_{skin}(3600s) \leq 47^{\circ}\text{C} \quad (35)$$

工作 60 分钟时，假人皮肤外侧温度超过 44°C 的时间不超过 5 分钟，等价于：

$$T_{skin}(3100s) \leq 44^{\circ}\text{C} \quad (36)$$

当工人处于高温环境下，在给定的作业时间内，满足人体安全性的条件下，以减薄厚度为目标，基于先前建立的微分方程模型，决定热防护服的最优厚度。

优化的目标函数与约束条件如下：

$$\min(D_2) \quad (37)$$

s. t.

$$\left\{ \begin{array}{l} D_2 \in (0.6, 25\text{mm}) \\ T_{sur} = 80^{\circ}\text{C} \end{array} \right. \quad (38)$$

$$\left\{ \begin{array}{l} T_{M_5}^j = 37^\circ C \quad \forall j = 0,1,\dots,1800 \quad (40) \\ T_{M_4}^{3600} \leq 47^\circ C \quad (41) \\ T_{M_4}^{3100} \leq 44^\circ C \quad (42) \\ r_l = \frac{k_l \tau}{c_l \rho_l h_l^2} \quad \forall l = I, II, III, IV, V \quad (43) \\ A \begin{pmatrix} T_1^{j+1} \\ T_2^{j+1} \\ \vdots \\ T_{M-1}^{j+1} \end{pmatrix} = b \quad \forall j = 0,1,2,\dots,N \quad (44) \end{array} \right.$$

(四) 优化模型的求解

对于以上的多约束下目标函数的优化问题，本文利用人工蜂群算法求解该优化问题。人工蜂群算法是模拟自然界中生物群在无控制情况下实现自组织、自适应行为而提出的一种智能算法。该算法最早是由土耳其学者 Karaboga^[1]在 2005 年首次提出，并将其成功应用于解决函数优化问题。

通过人工蜂群算法的求解，得到了满足题目所给的安全性要求，防护服 II 层的最优厚度为 4.683mm。

5.3 问题三的建模和求解

5.3.1 模型的建立

问题 3 要求综合考虑 II 层和 IV 层的厚度对假人皮肤外侧温度的影响，求解二维优化目标的问题。本文通过引入厚度权重系数 α ，实现了多优化目标的简化。

在防护服设计中，在降低薄度的同时，对于防护服的质量也有一定的要求，倘若防护服过重，则也会导致高温作业时的操作困难。由此，我们在本题的求解中，将同时考虑防护服的厚度以及质量两个因素，引入厚度权重系数 $\alpha \in [0,1]$ ，表示在防护服设计过程中对于厚度要求的重视程度。

对于防护服的质量，其 $m \propto \rho_I D_1 + \rho_{II} D_2 + \rho_{III} D_3 + \rho_{IV} D_4$ ，由于在该问题中， $\rho_I D_1 + \rho_{III} D_3$ 不变，因此其质量因素主要取决于 $\rho_{II} D_2 + \rho_{IV} D_4$ 。

由此，防护服的优化目标为：

$$f(D_2, D_4) = (1 - \alpha) \cdot \frac{\rho_{II} D_2 + \rho_{IV} D_4}{\rho_{II} D_{2max} + \rho_{IV} D_{4max}} + \alpha \cdot \frac{D_2 + D_4}{D_{2max} + D_{4max}} \quad (45)$$

其中 $\rho_{II} D_{2max} + \rho_{IV} D_{4max}$ 与 $D_{2max} + D_{4max}$ 为归一化因子。

由此，对于第三题的求解，其目标函数及约束问题可以写为：

$$\underset{D_2, D_4}{\operatorname{argmin}} f(D_2, D_4) \quad (46)$$

s. t.

$$T_{sur} = 80^\circ C \quad (47)$$

$$\begin{aligned}
D_2 &\in (0.6, 25\text{mm}) & (48) \\
D_4 &\in (0.6, 6.4\text{mm}) & (49) \\
T_{M_5}^j &= 37^\circ\text{C} \quad \forall j = 0, 1, \dots, 1800 & (50) \\
T_{M_4}^{1800} &\leq 47^\circ\text{C} & (51) \\
T_{M_4}^{1500} &\leq 44^\circ\text{C} & (52) \\
r_l &= \frac{k_l \tau}{c_l \rho_l h_l^2} \quad \forall l = I, II, III, IV, V & (53) \\
A \begin{pmatrix} T_1^{j+1} \\ T_2^{j+1} \\ \vdots \\ T_{M-1}^{j+1} \end{pmatrix} &= b \quad \forall j = 0, 1, 2, \dots, N & (54)
\end{aligned}$$

5.3.1 模型的求解

问题3的求解方法与问题2一致，都是采用人工蜂群算法，得到II层和IV层的最优厚度分别为5.2534mm和6.4mm。同时我们在此给出了第II层与第IV层厚度在问题3的条件下的厚度与边皮温度的关系图：

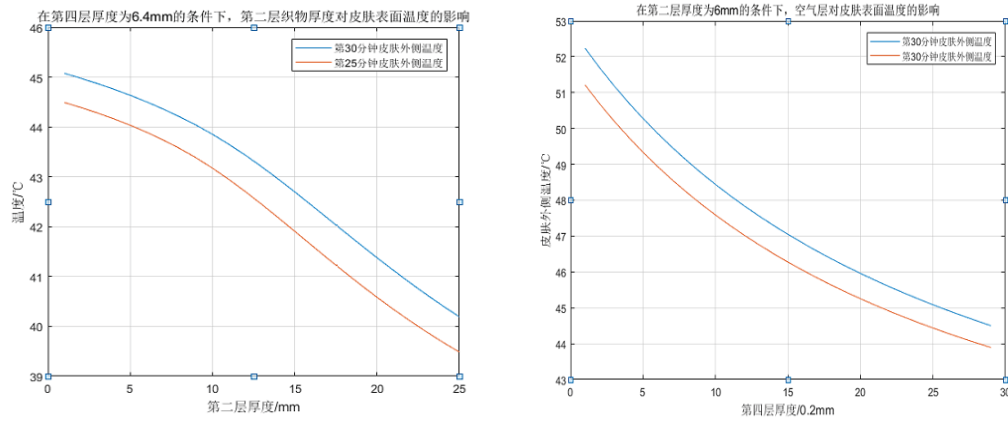


图 5.8 第 I 层厚度为 0.6mm 第 III 层厚度为 3.6mm 第 IV 层厚度为 6.4mm 皮肤层厚度为 25cm 时，皮肤在 60min 和 55min 时的温度与第 II 层温度的关系（左），第 I 层厚度为 0.6mm 第 II 层厚度为 6mm 第 III 层厚度为 3.6mm 皮肤层厚度为 25cm 时，皮肤在 60min 和 55min 时的温度与第 IV 层温度的关系（右）

六 模型的评价与推广

6.1 模型的评价

6.1.1 模型的优点

一方面，本模型充分考虑了热传递过程中的不同传热方式，在考虑严谨的情况下适当简化问题，适用于多层不同介质的热传导情况。另一方面，针对不同介质可其位置间隔的划分可不相同，因此在改变织物层厚度时，无需改变格点数目。本文建立的微分方程模型离散化后为线性方程，便于求解。

在优化方面，我们将人工蜂群算法（ABC）与本模型相结合，通过题目所给条件建立 ABC 算法的优化目标函数，对不等式约束条件，做罚函数处理，最终解得的参数对模型也有较高的契合度。由于人工蜂群算法不存在求导和函数连续化的限定，具有更好的全局优化能力，本文用人工蜂群算法求解设计最优化是创新之处。

6.1.2 模型的缺点

本文所建立的热防护服多介质热传导模型在各个细节都还有待完善之处，在分析由 III 层到 IV 层的热传递时，出于降低模型方程次数以简化计算的考虑，认为热辐射和热对流这两种热传递方式可以忽略不计。但在实际情况中，当空气层厚度较大时，热对流将不可再忽略，空气层的热传导模型还可以进行改进。

6.2 模型的推广

在热防护服装使用过程中，必然会有水汽产生（包括汗液和环境中的水汽），在本文中，没有考虑湿传递。同时，防护服本身也会吸收人体产生的汗液，这将会对第 III 层织物的物理特性产生影响，可进一步研究这之间的关系，用于模型的改进。

多层热防护服装热传递模型厚度优化方面，我们以减小织物层以及空气层的厚度进行优化，但没有考虑其他优化目标，如使人体皮肤表面温度尽可能保持较低温，可以在热防护服的人性化方面对优化函数进行进一步改进。

七 参考文献

- [1] Voelker C, Hoffmann S, Kornadt O, et al. Heat and moisture transfer through clothing[J]. Center for the Built Environment, 2010.
- [2] Karaboga D. An idea based on honey bee swarm for numerical optimization, tr06, [R]. Erciyes university, engineering faculty, computer engineering department, 2005.
- [3] Ghazy, Ahmed, Bergstrom, et al. Numerical simulation of the influence of fabric's motion on protective clothing performance during flash fire exposure[J]. Heat & Mass Transfer, 2013, 49(6):775-788.
- [4] 卢琳珍. 多层热防护服装的热传递模型及参数最优决定[D]. 浙江理工大学, 2018.
- [5] 潘斌. 热防护服装热传递数学建模及参数决定反问题[D]. 浙江理工大学, 2017.
- [6] 顾振亚, 宋国文, 陈克宁, 刘崇源, 颀小女, 崔振兴, 陈焕琪, 姜林太, 刘平峰. 热防护织物的研究[J]. 染整技术, 1996(06):18-19.

附录

```
%=====问题 1 代码=====
problem_1.m

clc
clear all
close all

dt=1;%时间间隔
dx=10^(-3)*0.1;%距离间隔
L=10^(-3)*[0.6 6 3.6 5];
p=[300 862 74.2 1.18];%密度
c=[1377 2100 1726 1005];%比热容
k=[0.082 0.37 0.045 0.028];%热传导率
r=(dt/dx^2)*k./(c.*p);%稳定性系数

T_environment=75;%环境温度
T_skin_obj=xlsread('C:\Users\16001\Desktop\ 数 学 建 模 \2018CUMCM\CUMCM-2018-
Problem-A-Chinese-Appendix.xlsx');
T_skin=T_skin_obj(:,2);%体表温度
T_skin=flipr(T_skin');
T_skin=T_skin';
T_initial=37;%初始温度

T=zeros(5400/dt+1,sum(L)/dx+1);%初始化温度分布
T(end,:)=T_initial;
T(:,1)=T_environment;
T(:,end)=T_skin;

%T(:,end)=37;

x_L=L/dx+1;%界面位置
x_L(1)=round(x_L(1));
x_L(2)=x_L(1)+x_L(2);
x_L(3)=x_L(3)+x_L(2);
x_L(4)=x_L(3)+x_L(4);

A=zeros(size(T,2)-2);
v_1=zeros(1,size(A,2));
v_2=zeros(1,size(A,2));
v_3=zeros(1,size(A,2));
```

```

for i=1:size(A,1)

    if i<x_L(1)-1
        v_1(i)=-r(1);
        v_2(i)=1+2*r(1);
        v_3(i)=-r(1);
    end

    if x_L(1)+1<i&& i<x_L(2)-1
        v_1(i)=-r(2);
        v_2(i)=1+2*r(2);
        v_3(i)=-r(2);
    end

    if x_L(2)+1<i&& i<x_L(3)-1
        v_1(i)=-r(3);
        v_2(i)=1+2*r(3);
        v_3(i)=-r(3);
    end

    if x_L(3)+1<i&& i<x_L(4)-1
        v_1(i)=-r(4);
        v_2(i)=1+2*r(4);
        v_3(i)=-r(4);
    end

    %交界 1 处理
    if x_L(1)-1<=i&& i<=x_L(1)+1
        if i==x_L(1)-1
            v_2(i)=1+2*r(1)-r(1)*(k(1)/(k(1)+k(2)));
            v_3(i)=-k(1)/(k(1)+k(2));
            v_1(i)=0;
            A(x_L(1)-1,x_L(1)+1)=-r(1)*k(2)/(k(1)+k(2));
        end

        if i==x_L(1)
            v_1(i)=-k(2)/(k(1)+k(2));
            v_2(i)=1;
            v_3(i)=0;
        end

        if i==x_L(1)+1
            v_1(i)=-r(2);
            v_2(i)=-k(2)/(k(1)+k(2))*r(2)+1+2*r(2);

```

```

v_3(i)=-r(2);

A(x_L(1)+1,x_L(1)-1)=-r(2)*k(1)/(k(1)+k(2));
end
end

```

```

%交界 2 处理
if x_L(2)-1<=i&&i<=x_L(2)+1
    if i==x_L(2)-1
        v_1(i)=0;
        v_2(i)=1+2*r(2)-r(2)*(k(2)/(k(2)+k(3)));
        v_3(i)=-k(2)/(k(2)+k(3));

        A(x_L(2)-1,x_L(2)+1)=-r(2)*k(3)/(k(2)+k(3));
        end

        if i==x_L(2)
            v_1(i)=-k(3)/(k(2)+k(3));
            v_2(i)=1;
            v_3(i)=0;
            end

            if i==x_L(2)+1
                v_1(i)=-r(3);
                v_2(i)=-k(3)/(k(2)+k(3))*r(3)+1+2*r(3);
                v_3(i)=-r(3);

                A(x_L(2)+1,x_L(2)-1)=-r(3)*k(2)/(k(2)+k(3));
                end
            end
end

```

```

%交界 3 处理
if x_L(3)-1<=i&&i<=x_L(3)+1
    if i==x_L(3)-1
        v_1(i)=0;
        v_2(i)=1+2*r(3)-r(3)*(k(3)/(k(3)+k(4)));
        v_3(i)=-k(3)/(k(3)+k(4));

        A(x_L(3)-1,x_L(3)+1)=-r(3)*k(4)/(k(3)+k(4));
        end

        if i==x_L(3)
            v_1(i)=-k(4)/(k(3)+k(4));
            v_2(i)=1;

```

```

        v_3(i)=0;
        end

        if i==x_L(3)+1
            v_1(i)=-r(4);
            v_2(i)=-k(4)/(k(3)+k(4))*r(4)+1+2*r(4);
            v_3(i)=-r(4);

            A(x_L(3)+1,x_L(3)-1)=-r(4)*k(3)/(k(3)+k(4));
        end
    end

    A(i,i)=v_2(i);
    if i<size(A,1)
        A(i,i+1)=v_1(i);
        A(i+1,i)=v_3(i);
    end

end

%A(end,end)=A(end,end)-r(4);

for k=1:5400
    b=T(end-(k-1),2:end-1);
    b=b';
    b(1)=b(1)+r(1)*T_environment;
    b(end)=b(end)+r(4)*T_skin(end-k);
    %b(end)=b(end)+r(4)*37;
    b(x_L(1:3))=0;

    T_obj=A\b;

    T(end-k,2:end-1)=T_obj';
    fprintf('The %d layers have been solved\n',k+1)
end

T=flipud(T);
filename='problem_1.xls';
xlswrite(filename,T,1);

[X Y]=meshgrid(1:size(T,2),1:size(T,1));
surf(X,Y,T)
shading interp

```

```

title('高温防护服—空气层热传导模型');
x1=xlabel(['距离 x/', num2str(dx),'m']);
x2=ylabel('时间 t/s');
x3=zlabel('温度 T/°C');
set(x1,'Rotation',30);
set(x2,'Rotation',-30);

%=====问题 2 代码=====
ABC_math.m%人工蜂群算法
clear all;close all;clc

%/*ABC 算法参数设置*/
NP=10;%采蜜蜂的个数或观察蜂的个数与蜜源的数量相等
maxCycle=100;%算法最大迭代次数
objfun='f_2';%目标函数
ub=[25*10^-3];%参数的取值上限[x1,x2,]
lb=[0.6*10^-3];%参数的取值下限[x1,x2,]
model=0;%模式设置： 0 代表实数规划， 1 代表整数规划
u=1;%领域搜索收敛因子，取值越小，收敛越慢，取值为(0,1]
du=200;%收敛梯度
%/*ABC 算法参数设置*/

FoodNumber=NP/2;%食物数量
D=size(ub,2);%参数个数
limit=NP*D;
runtime=1;%算法运行次数
GlobalMins=zeros(1,runtime);

for r=1:runtime
    Range=ub-lb;
    Lower=lb;
    Foods = rand(FoodNumber,D);%rand 生成 FoodNumber*D 大小的矩阵，元素为 0: 1
    for i=1:D
        if model==1
            Foods(:,i) = round(Foods(:,i) .* Range(i) + Lower(i));
        else
            Foods(:,i) = Foods(:,i) .* Range(i) + Lower(i);
        end
    end
end

ObjVal=feval(objfun,Foods);%feval 用来计算指定函数在某点的函数值
Fitness=calculateFitness(ObjVal);

```

```

trial=zeros(1,FoodNumber);%初始化每个食物的 trial 为 0

%/*The best food source is memorized*/
BestInd=find(ObjVal==min(ObjVal));%find 函数找到 ObjVal 数组中的最小值的位置
BestInd=BestInd(end);%取 ObjVal 数组中最后的最小值的位置
GlobalMin=ObjVal(BestInd);%记录函数的最小值
GlobalParams=Foods(BestInd,:);%记录函数最小值对应的参数 (x,y)

iter=1;
while (iter <= maxCycle)
    deta=u^(iter/du);%引入领域搜索收敛因子
    for i=1:(FoodNumber)

        %/*The parameter to be changed is determined randomly*/
        Param2Change=fix(rand*D)+1;%在[1:D]随机生成一个整数

        %/*A randomly chosen solution is used in producing a mutant
        %solution of the solution i*/
        neighbour=fix(rand*FoodNumber)+1;%即在所有食物中随机挑选食物
        %/*Randomly selected solution must be different from the solution i
        while(neighbour==i)
            neighbour=fix(rand*FoodNumber)+1;
        end

        sol=Foods(i,:);

        if model==0
            sol(Param2Change)=Foods(i,Param2Change)+...
                (Foods(i,Param2Change)-Foods(neighbour,Param2Change))...
                *(rand-0.5)*2*deta;%实数规划
        else
            sol(Param2Change)=Foods(i,Param2Change)+...
                round((Foods(i,Param2Change)-Foods(neighbour,Param2Change))...
                *(rand-0.5)*2*deta);%整数规划
        end
        % /*if generated parameter value is out of boundaries, it is
        %shifted onto the boundaries*/
        ind=find(sol<lb);
        sol(ind)=lb(ind);
        ind=find(sol>ub);
        sol(ind)=ub(ind);

        %evaluate new solution
        ObjValSol=feval(objfun,sol);

```



```

FitnessSol=calculateFitness(ObjValSol);

% /*a greedy selection is applied between the current solution i
%and its mutant
if (FitnessSol>Fitness(i)) /*If the mutant solution is better than
    %the current solution i, replace the solution with the mutant
    %and reset the trial counter of solution i*/
    Foods(i,:)=sol;
    Fitness(i)=FitnessSol;
    ObjVal(i)=ObjValSol;
    trial(i)=0;
else
    trial(i)=trial(i)+1;
end
end
prob=Fitness./sum(Fitness);

i=1;
t=0;
while(t<FoodNumber)
    if(rand<prob(i))
        t=t+1;
        /*The parameter to be changed is determined randomly*/
        Param2Change=fix(rand*D)+1;

        /*A randomly chosen solution is used in producing a mutant
        %solution of the solution i*/
        neighbour=fix(rand*(FoodNumber))+1;

        /*Randomly selected solution must be different from the solution i
        while(neighbour==i)
            neighbour=fix(rand*(FoodNumber))+1;
        end

        sol=Foods(i,:);

        if model==0
            sol(Param2Change)=Foods(i,Param2Change)+...
                (Foods(i,Param2Change)-Foods(neighbour,Param2Change))...
                *(rand-0.5)*2*deta;%实数规划
        else
            sol(Param2Change)=Foods(i,Param2Change)+...
                round((Foods(i,Param2Change)-Foods(neighbour,Param2Change))...
                *(rand-0.5)*2*deta);%整数规划

```

```

end

ind=find(sol<lb);
sol(ind)=lb(ind);
ind=find(sol>ub);
sol(ind)=ub(ind);

%evaluate new solution
ObjValSol=feval(objfun,sol);
FitnessSol=calculateFitness(ObjValSol);

% /*a greedy selection is applied between the current solution i and
%its mutant*/
if (FitnessSol>Fitness(i)) %/*If the mutant solution is better than
%the current solution i, replace the solution with the mutant and
%reset the trial counter of solution i*/
    Foods(i,:)=sol;
    Fitness(i)=FitnessSol;
    ObjVal(i)=ObjValSol;
    trial(i)=0;
else
    trial(i)=trial(i)+1;
end
end

i=i+1;
if (i==(FoodNumber)+1) %确保每个蜜源至少被观察 1 次
    i=1;
end
end

%/*The best food source is memorized*/
ind=find(ObjVal==min(ObjVal));
ind=ind(end);
if (ObjVal(ind)<GlobalMin)
    GlobalMin=ObjVal(ind);
    GlobalParams=Foods(ind,:);%储存最小函数值对应参数
end

ind=find(trial==max(trial));
ind=ind(end);
if (trial(ind)>limit)
    Bas(ind)=0;

    if model==0

```

```

sol=(ub-lb).*rand(1,D)+lb;%实数规划
else
sol=round((ub-lb).*rand(1,D))+lb;%整数规划
end

ObjValSol=feval(objfun,sol);
FitnessSol=calculateFitness(ObjValSol);
Foods(ind,:)=sol;
Fitness(ind)=FitnessSol;
ObjVal(ind)=ObjValSol;
end

fprintf('代数=%d GlobalMin=%g\n',iter,GlobalMin);
iter=iter+1;
X(iter)=iter;
Y(iter)=GlobalMin;
end% End of ABC
GlobalMins(r)=GlobalMin;
end %end of runs

```

```

figure(1);
plot(X(2:end),Y(2:end));
xlabel('迭代次数');
ylabel('函数取值');
GlobalParams %输出参数
save all

```

calculateFitness.m%人工蜂群算法适宜度函数

```

function fFitness=calculateFitness(fObjV)
fFitness=zeros(size(fObjV));
ind=find(fObjV>=0);
fFitness(ind)=1./(fObjV(ind)+1);
ind=find(fObjV<0);
fFitness(ind)=1+abs(fObjV(ind));

```

function ObjVal = f_2(Chrom)%问题 2 目标函数代码

```

for iter=1:size(Chrom,1)
dt=1;%时间间隔
dn=[3 3 18 25 250];%各层分割点数
L=10^(-3)*[0.6 Chrom(iter) 3.6 5.5 250];
dx=L./dn;%距离间隔

```

```

p=[300 862 74.2 1.18 1000];%密度
c=[1377 2100 1726 1005 2700];%比热容
k=[0.082 0.37 0.045 0.028 0.223];%热传导率
r=(dt./dx.^2).*k./(c.*p);%稳定性系数

T_environment=65;%环境温度
% T_skin_obj=xlsread('C:\Users\16001\Desktop\数学建模\2018CUMCM\CUMCM-2018-
Problem-A-Chinese-Appendix.xlsx');
% T_skin=T_skin_obj(:,2);%体表温度
% T_skin=fliplr(T_skin');
% T_skin=T_skin';
T_initial=37;%初始温度
T_skin=37;

T=zeros(3600/dt+1,sum(dn)+4);%初始化温度分布
T(end,:)=T_initial;
T(:,1)=T_environment;
T(:,end)=T_skin;

x_L=dn+1;%界面位置
x_L(2)=x_L(1)+x_L(2);
x_L(3)=x_L(3)+x_L(2);
x_L(4)=x_L(3)+x_L(4);
x_L(5)=x_L(5)+x_L(4);

A=zeros(size(T,2)-2);
v_1=zeros(1,size(A,2));
v_2=zeros(1,size(A,2));
v_3=zeros(1,size(A,2));

k=k./dx;%相对热传导率
for i=1:size(A,1)

    if i<x_L(1)-1
        v_1(i)=-r(1);
        v_2(i)=1+2*r(1);
        v_3(i)=-r(1);
    end

    if x_L(1)+1<i&& i<x_L(2)-1
        v_1(i)=-r(2);
        v_2(i)=1+2*r(2);
        v_3(i)=-r(2);
    end
end

```

```

if x_L(2)+1<i&&i<x_L(3)-1
v_1(i)=-r(3);
v_2(i)=1+2*r(3);
v_3(i)=-r(3);
end

```

```

if x_L(3)+1<i&&i<x_L(4)-1
v_1(i)=-r(4);
v_2(i)=1+2*r(4);
v_3(i)=-r(4);
end

```

```

if x_L(4)+1<i&&i<x_L(5)-1
v_1(i)=-r(5);
v_2(i)=1+2*r(5);
v_3(i)=-r(5);
end

```

%交界 1 处理

```

if x_L(1)-1<=i&&i<=x_L(1)+1
    if i==x_L(1)-1
        v_2(i)=1+2*r(1)-r(1)*(k(1)/(k(1)+k(2)));
        v_3(i)=-k(1)/(k(1)+k(2));
        v_1(i)=0;
        A(x_L(1)-1,x_L(1)+1)=-r(1)*k(2)/(k(1)+k(2));
    end

```

```

    if i==x_L(1)
        v_1(i)=-k(2)/(k(1)+k(2));
        v_2(i)=1;
        v_3(i)=0;
    end

```

```

    if i==x_L(1)+1
        v_1(i)=-r(2);
        v_2(i)=-k(2)/(k(1)+k(2))*r(2)+1+2*r(2);
        v_3(i)=-r(2);

```

```

        A(x_L(1)+1,x_L(1)-1)=-r(2)*k(1)/(k(1)+k(2));
    end

```

```

end

```

```

%交界 2 处理
if x_L(2)-1<=i&&i<=x_L(2)+1
    if i==x_L(2)-1
        v_1(i)=0;
        v_2(i)=1+2*r(2)-r(2)*(k(2)/(k(2)+k(3)));
        v_3(i)=-k(2)/(k(2)+k(3));

        A(x_L(2)-1,x_L(2)+1)=-r(2)*k(3)/(k(2)+k(3));
    end

    if i==x_L(2)
        v_1(i)=-k(3)/(k(2)+k(3));
        v_2(i)=1;
        v_3(i)=0;
    end

    if i==x_L(2)+1
        v_1(i)=-r(3);
        v_2(i)=-k(3)/(k(2)+k(3))*r(3)+1+2*r(3);
        v_3(i)=-r(3);

        A(x_L(2)+1,x_L(2)-1)=-r(3)*k(2)/(k(2)+k(3));
    end
end

```

```

%交界 3 处理
if x_L(3)-1<=i&&i<=x_L(3)+1
    if i==x_L(3)-1
        v_1(i)=0;
        v_2(i)=1+2*r(3)-r(3)*(k(3)/(k(3)+k(4)));
        v_3(i)=-k(3)/(k(3)+k(4));

        A(x_L(3)-1,x_L(3)+1)=-r(3)*k(4)/(k(3)+k(4));
    end

    if i==x_L(3)
        v_1(i)=-k(4)/(k(3)+k(4));
        v_2(i)=1;
        v_3(i)=0;
    end

    if i==x_L(3)+1
        v_1(i)=-r(4);
        v_2(i)=-k(4)/(k(3)+k(4))*r(4)+1+2*r(4);
    end
end

```

```

v_3(i)=-r(4);

A(x_L(3)+1,x_L(3)-1)=-r(4)*k(3)/(k(3)+k(4));
end
end

%交界 4 处理
if x_L(4)-1<=i&&i<=x_L(4)+1
    if i==x_L(4)-1
        v_1(i)=0;
        v_2(i)=1+2*r(4)-r(4)*(k(4)/(k(4)+k(5)));
        v_3(i)=-k(4)/(k(4)+k(5));

        A(x_L(4)-1,x_L(4)+1)=-r(4)*k(5)/(k(5)+k(4));
        end

        if i==x_L(4)
            v_1(i)=-k(5)/(k(5)+k(4));
            v_2(i)=1;
            v_3(i)=0;
            end

            if i==x_L(4)+1
                v_1(i)=-r(5);
                v_2(i)=-k(5)/(k(4)+k(5))*r(5)+1+2*r(5);
                v_3(i)=-r(5);

                A(x_L(4)+1,x_L(4)-1)=-r(5)*k(4)/(k(5)+k(4));
                end
            end

            A(i,i)=v_2(i);
            if i<size(A,1)
                A(i,i+1)=v_1(i);
                A(i+1,i)=v_3(i);
            end
        end

        end

        for m=1:3600
            b=T(end-(m-1),2:end-1);
            b=b';
            b(1)=b(1)+r(1)*T_environment;
            b(end)=b(end)+r(5)*T_skin;

```

```

% b(end-1)=b(end-1)-37*k(5)/(k(4)+k(5));
% b(end)=k(5)/(k(4)+k(5))*37;
b(x_L(1:4))=0;

T_obj=A\b;

if T_obj(x_L(4))>47
    T_obj(x_L(4))=T_environment;
    break;
end

if 1<=m&&m<=3300
    if T_obj(x_L(4))>44
        T_obj(x_L(4))=T_environment;
        break;
    end
end

T(end-m,2:end-1)=T_obj';
%fprintf('The %d layers have been solved\n',m+1)
end

ObjVal(iter)=0.5*T_obj(x_L(4))+0.5*Chrom(iter);

end

%=====问题 3 代码=====
function ObjVal = f_3(Chrom)%问题 3 目标函数代码

for iter=1:size(Chrom,1)

    dt=1;%时间间隔
    dn=[30,100,180,250,250];%各层分割点数
    L=10^(-3)*[0.6 Chrom(iter,1) 3.6 Chrom(iter,2) 250];
    dx=L./dn;%距离间隔
    p=[300 862 74.2 1.18 1000];%密度
    c=[1377 2100 1726 1005 2600];%比热容
    k=[0.082 0.37 0.045 0.028 0.190];%热传导率
    r=(dt./dx.^2).*k./(c.*p);%稳定性系数

    T_environment=80;%环境温度
    % T_skin_obj=xlsread('C:\Users\16001\Desktop\数学建模\2018CUMCM\CUMCM-2018-
    Problem-A-Chinese-Appendix.xlsx');

```



```

% T_skin=T_skin_obj(:,2);%体表温度
% T_skin=flipr(T_skin');
% T_skin=T_skin';
T_initial=37;%初始温度
T_skin=37;

T=zeros(1800/dt+1,sum(dn)+4);%初始化温度分布
T(end,:)=T_initial;
T(:,1)=T_environment;
T(:,end)=T_skin;

x_L=dn+1;%界面位置
x_L(2)=x_L(1)+x_L(2);
x_L(3)=x_L(3)+x_L(2);
x_L(4)=x_L(3)+x_L(4);
x_L(5)=x_L(5)+x_L(4);

A=zeros(size(T,2)-2);
v_1=zeros(1,size(A,2));
v_2=zeros(1,size(A,2));
v_3=zeros(1,size(A,2));

k=k./dx;%相对热传导率
for i=1:size(A,1)

    if i<x_L(1)-1
        v_1(i)=-r(1);
        v_2(i)=1+2*r(1);
        v_3(i)=-r(1);
    end

    if x_L(1)+1<i&&i<x_L(2)-1
        v_1(i)=-r(2);
        v_2(i)=1+2*r(2);
        v_3(i)=-r(2);
    end

    if x_L(2)+1<i&&i<x_L(3)-1
        v_1(i)=-r(3);
        v_2(i)=1+2*r(3);
        v_3(i)=-r(3);
    end

    if x_L(3)+1<i&&i<x_L(4)-1

```

```

v_1(i)=-r(4);
v_2(i)=1+2*r(4);
v_3(i)=-r(4);
end

if x_L(4)+1<i&& i<x_L(5)-1
v_1(i)=-r(5);
v_2(i)=1+2*r(5);
v_3(i)=-r(5);
end

%交界 1 处理
if x_L(1)-1<=i&&i<=x_L(1)+1
    if i==x_L(1)-1
        v_2(i)=1+2*r(1)-r(1)*(k(1)/(k(1)+k(2)));
        v_3(i)=-k(1)/(k(1)+k(2));
        v_1(i)=0;
        A(x_L(1)-1,x_L(1)+1)=-r(1)*k(2)/(k(1)+k(2));
    end

    if i==x_L(1)
        v_1(i)=-k(2)/(k(1)+k(2));
        v_2(i)=1;
        v_3(i)=0;
    end

    if i==x_L(1)+1
        v_1(i)=-r(2);
        v_2(i)=-k(2)/(k(1)+k(2))*r(2)+1+2*r(2);
        v_3(i)=-r(2);

        A(x_L(1)+1,x_L(1)-1)=-r(2)*k(1)/(k(1)+k(2));
    end
end

%交界 2 处理
if x_L(2)-1<=i&&i<=x_L(2)+1
    if i==x_L(2)-1
        v_1(i)=0;
        v_2(i)=1+2*r(2)-r(2)*(k(2)/(k(2)+k(3)));
        v_3(i)=-k(2)/(k(2)+k(3));

        A(x_L(2)-1,x_L(2)+1)=-r(2)*k(3)/(k(2)+k(3));

```

```

end

if i==x_L(2)
v_1(i)=-k(3)/(k(2)+k(3));
v_2(i)=1;
v_3(i)=0;
end

if i==x_L(2)+1
v_1(i)=-r(3);
v_2(i)=-k(3)/(k(2)+k(3))*r(3)+1+2*r(3);
v_3(i)=-r(3);

A(x_L(2)+1,x_L(2)-1)=-r(3)*k(2)/(k(2)+k(3));
end
end

%交界 3 处理
if x_L(3)-1<=i&&i<=x_L(3)+1
    if i==x_L(3)-1
        v_1(i)=0;
        v_2(i)=1+2*r(3)-r(3)*(k(3)/(k(3)+k(4)));
        v_3(i)=-k(3)/(k(3)+k(4));

        A(x_L(3)-1,x_L(3)+1)=-r(3)*k(4)/(k(3)+k(4));
    end

    if i==x_L(3)
        v_1(i)=-k(4)/(k(3)+k(4));
        v_2(i)=1;
        v_3(i)=0;
    end

    if i==x_L(3)+1
        v_1(i)=-r(4);
        v_2(i)=-k(4)/(k(3)+k(4))*r(4)+1+2*r(4);
        v_3(i)=-r(4);

        A(x_L(3)+1,x_L(3)-1)=-r(4)*k(3)/(k(3)+k(4));
    end
end

%交界 4 处理
if x_L(4)-1<=i&&i<=x_L(4)+1

```

```

    if i==x_L(4)-1
        v_1(i)=0;
        v_2(i)=1+2*r(4)-r(4)*(k(4)/(k(4)+k(5)));
        v_3(i)=-k(4)/(k(4)+k(5));

        A(x_L(4)-1,x_L(4)+1)=-r(4)*k(5)/(k(5)+k(4));
    end

    if i==x_L(4)
        v_1(i)=-k(5)/(k(5)+k(4));
        v_2(i)=1;
        v_3(i)=0;
    end

    if i==x_L(4)+1
        v_1(i)=-r(5);
        v_2(i)=-k(5)/(k(4)+k(5))*r(5)+1+2*r(5);
        v_3(i)=-r(5);

        A(x_L(4)+1,x_L(4)-1)=-r(5)*k(4)/(k(5)+k(4));
    end
end

A(i,i)=v_2(i);
if i<size(A,1)
    A(i,i+1)=v_1(i);
    A(i+1,i)=v_3(i);
end
end

for m=1:1800
    b=T(end-(m-1),2:end-1);
    b=b';
    b(1)=b(1)+r(1)*T_environment;
    b(end)=b(end)+r(5)*T_skin;

    % b(end-1)=b(end-1)-37*k(5)/(k(4)+k(5));
    % b(end)=k(5)/(k(4)+k(5))*37;
    b(x_L(1:4))=0;

    T_obj=A\b;

    if T_obj(x_L(4))>47
        T_obj(x_L(4))=T_environment;

```

```

        break;
    end

    if 1<=m&&m<=1500
        if T_obj(x_L(4))>44
            T_obj(x_L(4))=T_environment;
            break;
        end
    end
end

T(end-m,2:end-1)=T_obj';
fprintf('The %d layers have been solved\n',m+1)
end

ObjVal(iter)=
T_obj(x_L(4))+0.5*(Chrom(iter,2)*1.18+Chrom(iter,1)*862)/(1.18*6.4/1000+862*25/1000)+0.5
*(( Chrom(iter,2) +Chrom(iter,1))/ (1.18*6.4/1000+862*25/1000));

end

```