

基于偏微分方程和优化算法的防护服材料厚度及其反问题的研究

摘要

专用防护服可以保护人们免于在高温环境下受到灼伤。专用防护服常常由三层织物材料构成。称为 I、II、III 层。其中 I 层与外界接触，III 层与皮肤之间还有空气层。为了获得设计合理的防护服，我们既要求防护服达到满足功用的防护要求，又要实现制造企业利润的最大化。

通过数学建模可以对专用服装进行设计，确定三层织物材料的厚度。该建模涉及两方面的工作。

首先，确定合理的热传递模型。热传递主要存在三种基本形式：热传导、热辐射和热对流。只要在物体内部或物体间有温度差存在，热能就必然以以上三种方式中的一种或多种从高温到低温处传递。我们在第一问中建立偏微分方程模型，基础模型中我们假定在分界层处温度变化是连续的，得到的结果与事实相差较大。在进阶模型中我们由傅里叶定理，采用第一类与第三类边界条件，其中热通量正比于温度梯度，再模拟热量传递的过程。再与试验得到的 90 分钟后的皮肤层温度进行对比。得到了近乎一致的结果。可见我们建立的微分方程的进阶模型及其边界条件可以很好地模拟出热传递的过程。

接着，我们处理设计中的反问题。在第二问中，给定条件，要求单个变量（第 II 层厚度）变化时防护服仍能起到需要的防护作用。实则是对该变量进行优化以实现结果的最优化，即在允许的范围内获得最大的经济效益。通过对第 II 层厚度的多点取样及分析，我们获得了第 II 层厚度变化时皮肤层温度改变的函数。当第 II 层厚度同时满足两个条件时，我们取厚度所能取到的最小值以节省材料，期间我们进行了多项式拟合，选取最小二乘法得到的结果。最后计算出 II 层厚度最优值为 21.951mm。

在第三问中，给定条件，要求两个变量（第 II 层和第 IV 层厚度）都变化时防护服仍能起到防护作用。我们首先确定变量满足设计要求的范围，再使用遗传算法计算目标函数的最优解，从而数值模拟得到满足设计要求的最优解的组合（II 层和 IV 层）。这个问题本质上是一个非线性规划问题。

在模型的检验与改进方面我们之后对于多项式拟合进行了讨论，并且给出了第二问 II 层的最优厚度范围。之后我们假设人内部血液流动会对于外界介质有温度的影响，从而考虑出抑制系数和促进系数来优化模型，这里也放宽了假人恒定体内温度这一条件，使得模型更有实际意义。

本次建模的研究在之后的热量在多层介质中传递的情况也有较好的可移植性，体现了数学建模的价值。

关键词：热传导方程，差分法，傅里叶定理，线性与非线性优化，遗传算法

1 引言

1.1 问题背景

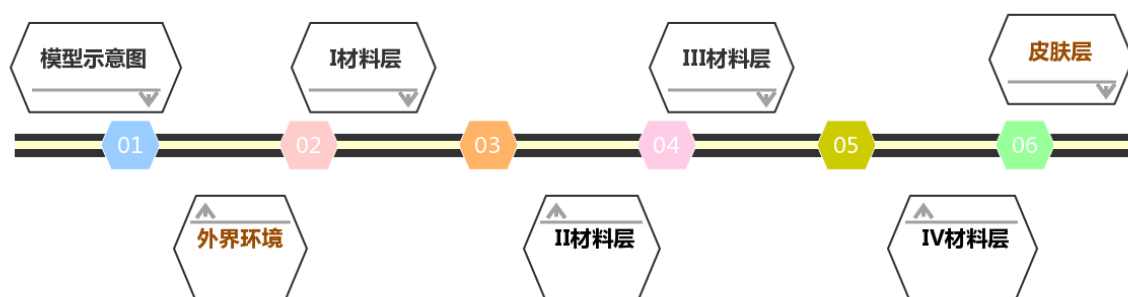
在高温环境下工作时，人们需要穿着专用服装以避免灼伤。如图所示，防护服的隔热性是指热防护服必须具备较好的减缓和阻止热量向服装内部转移的性能，避免热源对人体造成伤害。隔热性也是热防护服最重要的热防护性能之一。

在热防护服的实际使用中，大多数使用者并不直接接触火焰，外界热量以热对流、热辐射和热传导至人体，对人体造成伤害。因此，使用具有良好隔热性的热防护服，就可以在外界高温和人体之间为使用者提供一道保护屏障，使外界热量难以通过服装，从而保证使用者在高温环境中能安全工作。



1.2 问题重述

专用服装通常由三层织物材料构成，记为 I、II、III 层，其中 I 层与外界环境接触，III 层与皮肤之间还存在空隙，将此空隙记为 IV 层。如图所示：



图一、模型示意图

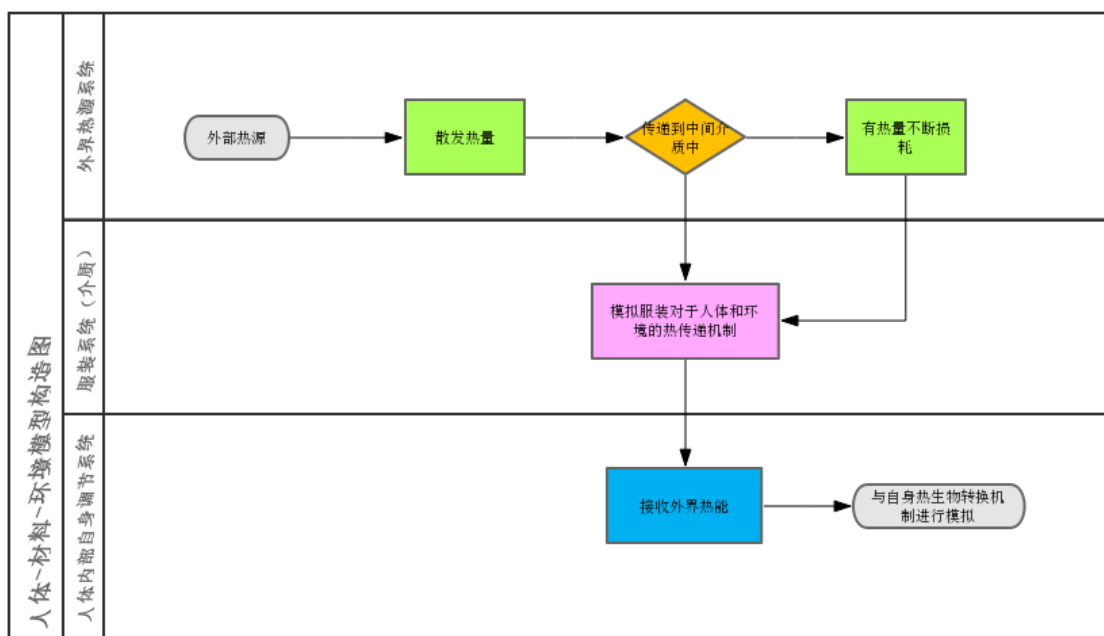
三个小问中每一个小问都环环相扣，步步衔接，前一步均为后一步做一定的铺垫和引导，下面我们简要分析三个问题与其中的关联。

在第一问中，我们通过建立微分方程模型，计算在环境温度为 75°C 、II 层厚度为 6 mm、IV 层厚度为 5 mm、工作时间为 90 分钟的情形下的温度分布，与试验得到的假人皮肤外侧温度进行对比，检验所求的模型的准确性。这一步是最为基础也是最为关键的，只有先熟悉了在两层(II 层和 IV 层)的参数，并且精确计算出相应结果后，后面几个小问的优化问题才会变得更加精确，所以第一小问的热传导模型需要考虑尽可能全面的模型影响因子，在后文我们会分为基础模型和进阶模型来逐步细致精确地求解该问题。

在第二问中，当环境温度为 65°C 、IV 层的厚度为 5.5 mm 时，我们根据已经建立的模型，确定 II 层的最优厚度。¹这一题承接了第一小问的求解，进一步放开了一个变量的设置，即第 II 层材料层，在限定第 IV 层为 5.5mm 时候第 II 层的最优厚度，并且加了一定的限定条件，转而使得偏微分方程的初始值有了改变，并且之后进行优化需要进一步调整参数，在第一小问的基础上更近了一步。

在第三问中，当环境温度为 80°C 时，根据已经建立的模型确定 II 层和 IV 层的最优厚度，确保工作 30 分钟时。²在第二问的基础上又使得第 II 层和第 IV 层的厚度都不确定，需要调参计算出最佳的厚度，在前几问的铺垫下更进了一步，需要考虑双变量调整参数的问题，并且结合限定条件，可以说每一题丝丝入扣，环环深入，引导我们把模型从简单的给出所有相应信息到逐步增加未知变量的个数从而形成进阶的参数最优化的模型。在后文的具体建模中我们也会先以简单的热传导模型入手，逐步考虑复杂的情况，进一步优化改善模型，进而逐步求解问题和做出相应结果。

当然，我们做了一个总体的模型构造图，主要分为三个模型板块：分别是人体内部自身调节系统，服装（介质）系统和外界热源系统，三个系统一起构成了完整的热力系统。（其中我们可以在计算过程中将人体内部自身调节系统因素暂时设定成常数，因为我们已经假设了假人的内部恒温 37°C ，之后再模型改进方面我们可以加以讨论这个要素）。



图二、整体模型构架图

¹ 条件：确保工作 60 分钟时，假人皮肤外侧温度不超过 47°C ，且超过 44°C 的时间不超过 5 分钟。

² 条件：假人皮肤外侧温度不超过 47°C ，且超过 44°C 的时间不超过 5 分钟。

2 问题分析

2.1 问题一分析

问题一的关键是要建立合适的微分方程模型来模拟热交换的过程。其关键是边界条件的确立。在基础模型中我们假设不同材料交接的边界处温度是连续变化的，得到的模拟结果不尽如人意。反思过后，在进阶模型中我们就采用了第一类和第三类边界条件获得了较好的模拟结果。

2.2 问题二分析

问题二的目标是找出在对应条件下温度的临界值。关键是要把握随着材料厚度变化，皮肤层的温度如何改变。

2.3 问题三分析

问题三的关键在于对优化目标的选取。问题三涉及两个变量，找出两个变量各自满足条件的自变量范围，并优化目标是第三问的要求。

3 模型假设

- 假设 1：热传递为垂直于皮肤方向进行，故可视为一维的。
- 假设 2：热防护服的织物具有各向同性
- 假设 3：热传递过程中，织物的结构几乎不变
- 假设 4：该系统仅考虑热传递，包括热辐射、热传导以及外界环境的热对流现象，不考虑湿传递，即忽略水汽、汗液的影响（题中给的是假人）
- 假设 5：热传导传递到防护材料的过程是均匀的，假定热防护材料没有发生热溶解
- 假设 6：防护材料之间，防护材料与空气层之间，空气层与皮肤之间温度梯度的变化是连续的。

4 符号说明

符号	意义
x	表示到外界环境与防护材料接触面的垂直距离
u	表示某时刻某点的温度

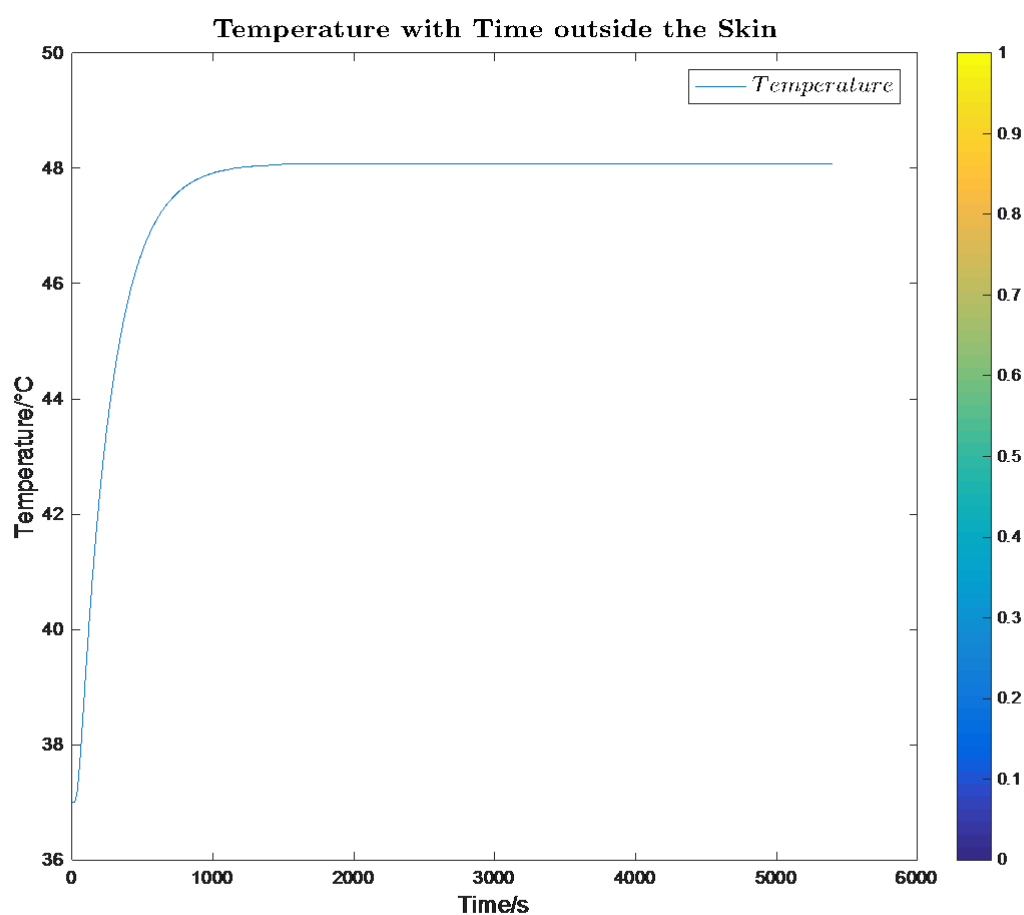
t	时间
α	热扩散率
k	热传导系数
γ	空气自然对流的热交换系数

5 模型建立与求解

5.1 问题一的模型建立与检验

5.1.1 附件数据处理和分析

首先对与 excel 中的附件二，我们用 MATLAB 拟合了人皮肤外侧温度随时间的走势图，如下图所示。



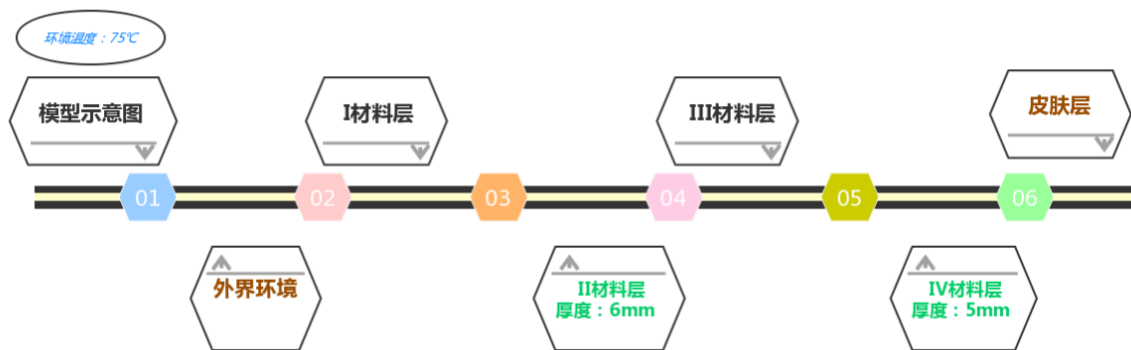
³ 具体一些符号如若在下文只出现少数次，即该位置会有说明

图三、假人皮肤外侧温度随时间变化图⁴

我们可以很清晰地看出在 1000s 的前后有明显的趋势性变化，在约 1000 秒之后假人皮肤外侧的温度有明显的收敛趋势，尤其在 1645 秒以及之后的假人皮肤外侧温度都在 48.08℃ 保持不变，说明的大体皮肤外侧的温度趋势是一个先以加速度不断减小而加速上升，然后到达 48℃ 附近趋于收敛趋势，在 800 秒前后有明显转折点。

5.1.2 问题一模型建立

我们之后的模型分为基础模型和进阶模型，基础模型主要是假设了介质与介质表面接触的温度是相等的，并且假设热量传递是处于理想状态，热量损失与耗散相对不计入考虑范畴。进阶模型主要会考虑两种材料的接触点温度并不一定相同(可以从偏微分方程中的第三类边界条件中所得)，进而考虑边界条件时着重考虑了所研究热量从一个介质到另一个介质中，其是与两者的温差成正比的。从介质内部来看热量传输由傅里叶定律确定，并且我们在进阶模型中完善了边界条件，承接基础模型做了更多限制，精确了结论，步步深入。

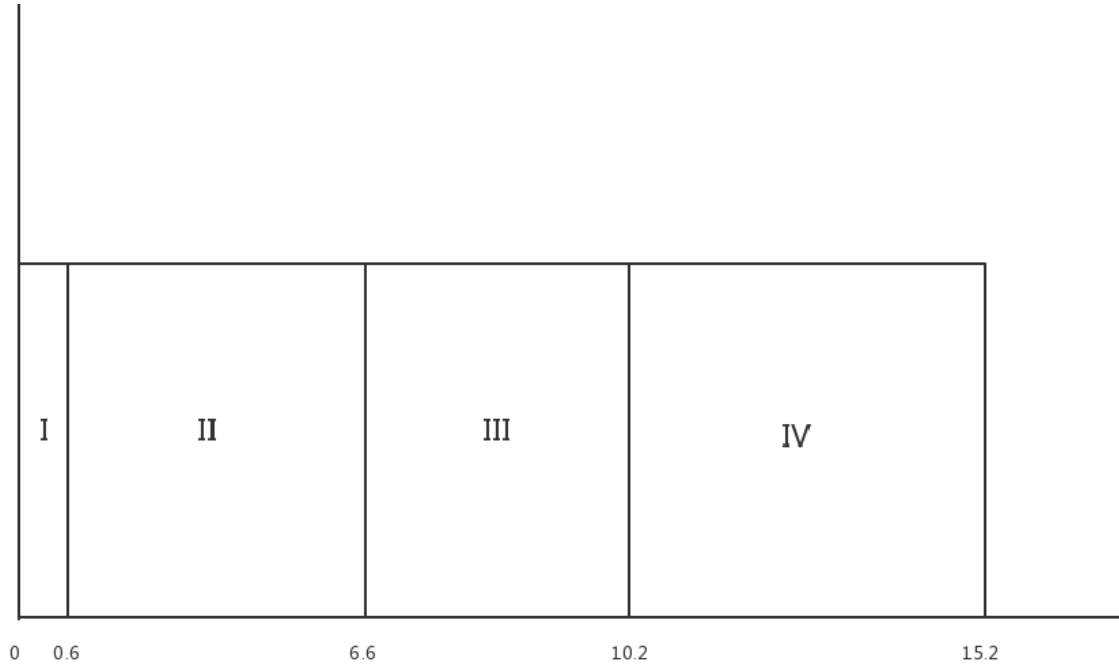


图四、第一小问模型概念图

基本模型

首先我们知道 $\alpha = \frac{k}{\rho c_p}$ ，其中 α 为热扩散率。

⁴ 程序见附录中的 skin_temperature.m



图五、材料分布示意图

上图中 I 为第一层材料，是与外部环境所接触的，II 与 III 层也是相应材料(依次向外)，其中 III 与皮肤之间还有空隙为 IV 层。

于是我们建立热传导方程(偏微分方程)，并且确定边界条件和相应初始条件。

$$u_t = \alpha_I \frac{\partial^2 u}{\partial x^2}, x \in (0, 0.6), t > 0 \quad (1)$$

$$u(0, 0) = 75, u(0.6^+, 0) = u(0.6^-, 0), u(x, 0) = 37$$

$$u_t = \alpha_{II} \frac{\partial^2 u}{\partial x^2}, x \in (0.6, 6.6), t > 0 \quad (2)$$

$$u(0.6^+, 0) = u(0.6^-, 0), u(6.6^+, 0) = u(6.6^-, 0), u(x, 0) = 37$$

$$u_t = \alpha_{III} \frac{\partial^2 u}{\partial x^2}, x \in (6.6, 10.2), t > 0 \quad (3)$$

$$u(10.2^+, 0) = u(10.2^-, 0), u(x, 0) = 37$$

$$u_t = \alpha_{IV} \frac{\partial^2 u}{\partial x^2}, x \in (10.2, 15.2), t > 0 \quad (4)$$

$$u(15.2, 0) = 37, u(x, 0) = 37$$

下面我们利用有限差分法

在各段上有

$$\frac{du_i(t)}{dt} = \frac{\alpha}{h^2} [u_{i+1}(t) + u_{i-1}(t) - 2u_i(t)] \quad (5)$$

对于第 I 层: $i = 1, \dots, 60, u_i(0) = 37$

$$u_0(t) = 75, u_{60}(t) = u_{61}(t)$$

$$\frac{d}{dt} \begin{bmatrix} u_1 \\ \vdots \\ u_{60} \end{bmatrix} = \frac{\alpha_I}{h^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 & -1 \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_{60} \end{bmatrix} + \frac{\alpha_I}{h^2} \begin{bmatrix} 75 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (6)$$

对于第 II 层: $i = 61, \dots, 660, u_i(0) = 37$

$$u_{60}(t) = u_{61}(t), u_{660}(t) = u_{661}(t)$$

$$\frac{d}{dt} \begin{bmatrix} u_{61} \\ \vdots \\ u_{660} \end{bmatrix} = \frac{\alpha_{II}}{h^2} \begin{bmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 & -1 \end{bmatrix} \begin{bmatrix} u_{61} \\ \vdots \\ u_{660} \end{bmatrix} \quad (7)$$

对于第 III 层: $i = 661, \dots, 1020, u_i(0) = 37$

$$u_{660}(t) = u_{661}(t), u_{1020}(t) = u_{1021}(t)$$

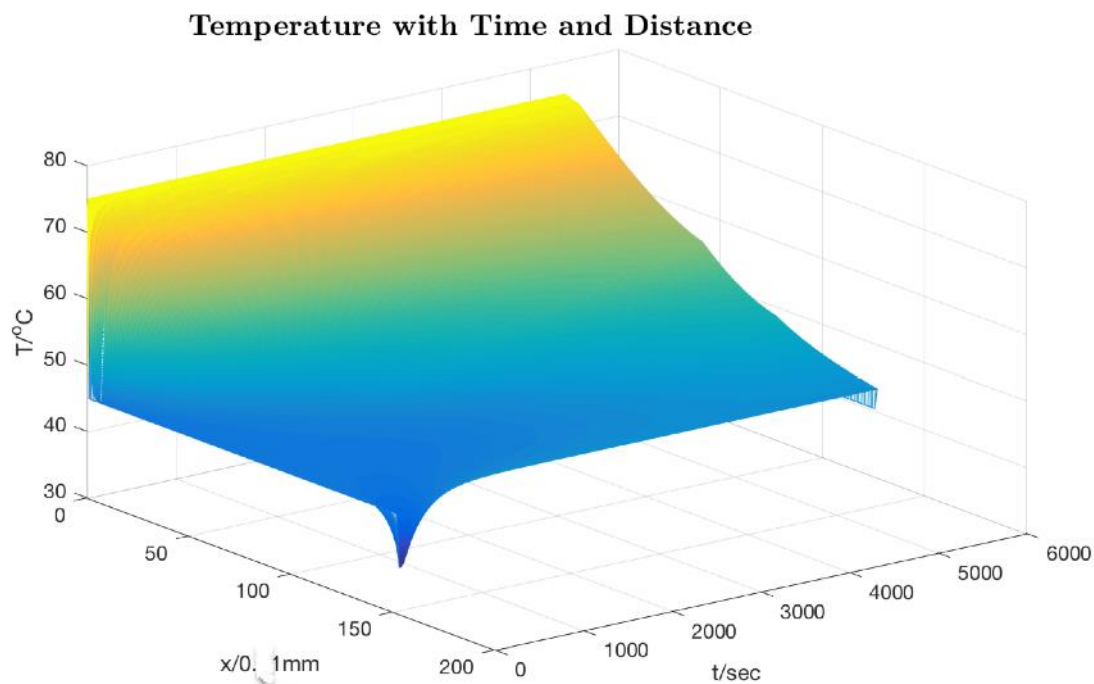
$$\frac{d}{dt} \begin{bmatrix} u_{661} \\ \vdots \\ u_{1020} \end{bmatrix} = \frac{\alpha_{III}}{h^2} \begin{bmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 & -1 \end{bmatrix} \begin{bmatrix} u_{661} \\ \vdots \\ u_{1021} \end{bmatrix} \quad (8)$$

对于第 IV 层: $i = 1021, \dots, 1520, u_i(0) = 37$

$$u_{1020}(t) = u_{1021}(t), u_{1521}(t) = 37$$

$$\frac{d}{dt} \begin{bmatrix} u_{1021} \\ \vdots \\ u_{1520} \end{bmatrix} = \frac{\alpha_{IV}}{h^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 & -2 \end{bmatrix} \begin{bmatrix} u_{1021} \\ \vdots \\ u_{1520} \end{bmatrix} + \frac{\alpha_{IV}}{h^2} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 37 \end{bmatrix} \quad (9)$$

然后我们进行 MATLAB 数值模拟, 得出结果如下图所示



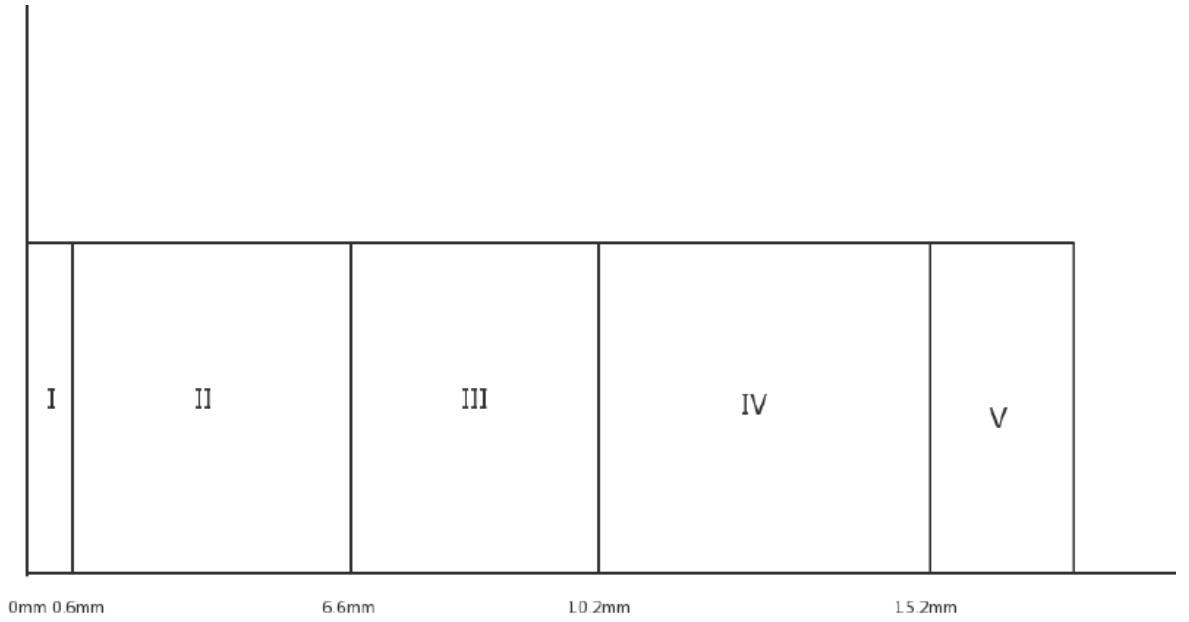
图六、温度随时间和距离的曲面示意图⁵

我们可以清晰地看到距离坐标轴上 0 是原点，其左侧为与外界接触，外界温度为 75°C，右侧 15.2mm 处即是四层介质的终点，也就是皮肤表层，假人体内温度控制在了恒温 37°C。不过可以从图中看出其实在三层材料与材料之间并没有很好的梯度效应，而是只有在两处边界有明显的衰减，主要原因还是我们的基础模型做了过多的理想化假设，并且没有考虑到介质与介质接触的地方其实温度是不相同的，进而，我们引入进阶模型来重构我们的数值计算过程。

进阶模型

在进阶模型中，我们对模型做了更加细致化的处理，将热扩散率进行了一定程度上的调整。

⁵ 代码如附录中 HeatTransfer.m 所示



图七、初始模型相对位置示意图

上图中 I 为第一层材料，是与外部环境所接触的，II 与 III 层也是相应材料(依次向外)，其中 III 与皮肤之间还有空隙为 IV 层，而 V 层则是表示皮肤层。也就是说在进阶模型中，我们把皮肤作为单独的一层分出来而非直接将右边界条件设定为体温(37℃)。

于是我们建立热传导方程(偏微分方程)，并且确定边界条件和相应初始条件。

$$u_t = \alpha_I \frac{\partial^2 u}{\partial x^2}, x \in (0, 0.6), t > 0 \quad (10)$$

$$k_1 \frac{\partial u}{\partial n} \Big|_{x=0} = \gamma_0 [u(0, t) - T_{\text{外}}], u(x) = 37 \quad (11)$$

其中 k 为热传导系数， γ 为空气自然对流热交换系数， $T_{\text{外}} = 75$

$$u_t = \alpha_{II} \frac{\partial^2 u}{\partial x^2}, x \in (0.6, 6.6), t > 0 \quad (12)$$

$$u(0.6^+, 0) = u(0.6^-, 0), u(6.6^+, 0) = u(6.6^-, 0), u(x) = 37$$

$$u_t = \alpha_{III} \frac{\partial^2 u}{\partial x^2}, x \in (6.6, 10.2), t > 0 \quad (14)$$

$$k_3 \frac{\partial u}{\partial n} \Big|_{x=10.2} = \gamma [u(10.2^+, t) - u(10.2^-, t)] \quad (15)$$

其中 k_3 为第三层热传导系数

$$u_t = \alpha_{IV} \frac{\partial^2 u}{\partial x^2}, x \in (10.2, 15.2), t > 0 \quad (16)$$

$$k_4 \frac{\partial u}{\partial n} \Big|_{x=15.2} = \gamma [u(15.2^+, t) - u(15.2^-, t)] \quad (17)$$

$$u_t = \alpha_v \frac{\partial^2 u}{\partial x^2}, x \in (15.2, 15.7), t > 0 \quad (18)$$

$$k_5 \frac{\partial u}{\partial n} |_{x=15.7} = \gamma_2 [u(15.7, t) - 37] \quad (19)$$

其中 k_5 为第五层热传导系数， γ_2 为血液对流热交换系数

下面我们利用有限差分法

作偏导离散化：

$$\frac{\partial u}{\partial t} = \frac{u_x^{k+\Delta t} - u_x^k}{\Delta t} \quad (20)$$

其中 u_x^k ，上标代表时间维度，下标代表空间维度

$$u(x + \Delta x, t) = u(x, t) + \Delta x \frac{\partial u}{\partial x} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 u}{\partial x^2} + \dots \quad (21)$$

$$u(x - \Delta x, t) = u(x, t) - \Delta x \frac{\partial u}{\partial x} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 u}{\partial x^2} + \dots \quad (22)$$

两式相加得

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{\Delta x^2} + O(\Delta x^2) \quad (23)$$

可以推出：

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{\Delta x^2} \quad (24)$$

由此 $\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$ 可推出

$$\frac{u_x^{k+\Delta t} - u_x^k}{\Delta t} = \alpha \frac{u_{x+\Delta x}^k - 2u_x^k + u_{x-\Delta x}^k}{\Delta x^2} \quad (25)$$

$$u_x^{k+\Delta t} = u_x^k + \frac{\alpha \Delta t}{\Delta x^2} [u_{x+\Delta x}^k - 2u_x^k + u_{x-\Delta x}^k] \quad (26)$$

令 $s = \frac{\alpha \Delta t}{\Delta x^2}$

$$u_x^{k+\Delta t} = (1 - 2s)u_x^k + s[u_{x+\Delta x}^k + u_{x-\Delta x}^k] \quad (27)$$

这里我们直接采用显示格式迭代，只需要

$$s \leq \frac{1}{2}$$

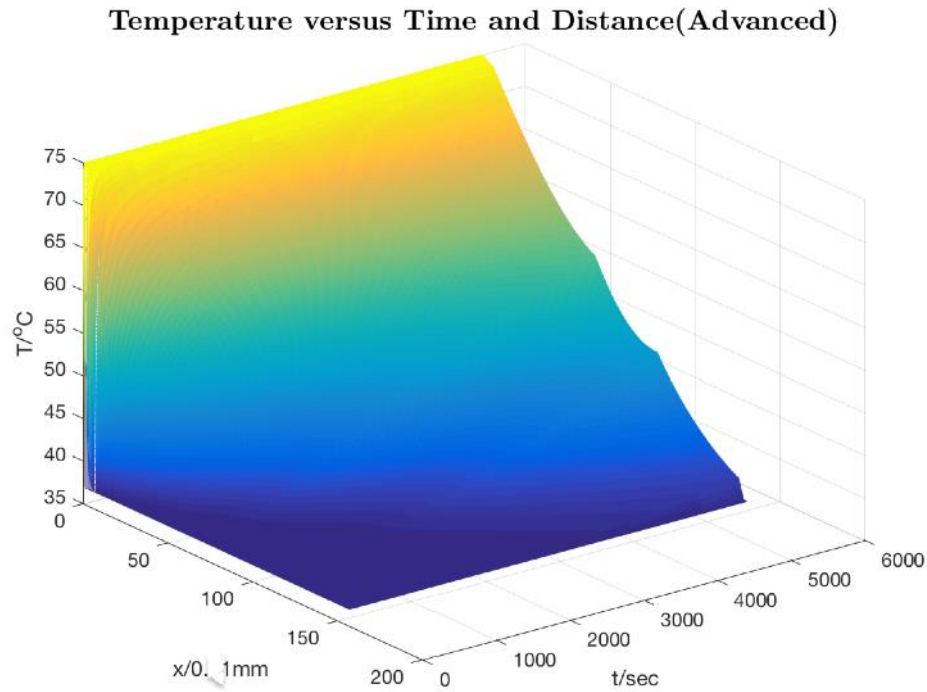
即可保证迭代收敛

所以与之前类似的，在各层上的迭代公式为：

$$u_x^{k+1} = (1 - 2s_i)u_x^k + s_i[u_{x+1}^k + u_{x-1}^k] \quad (28)$$

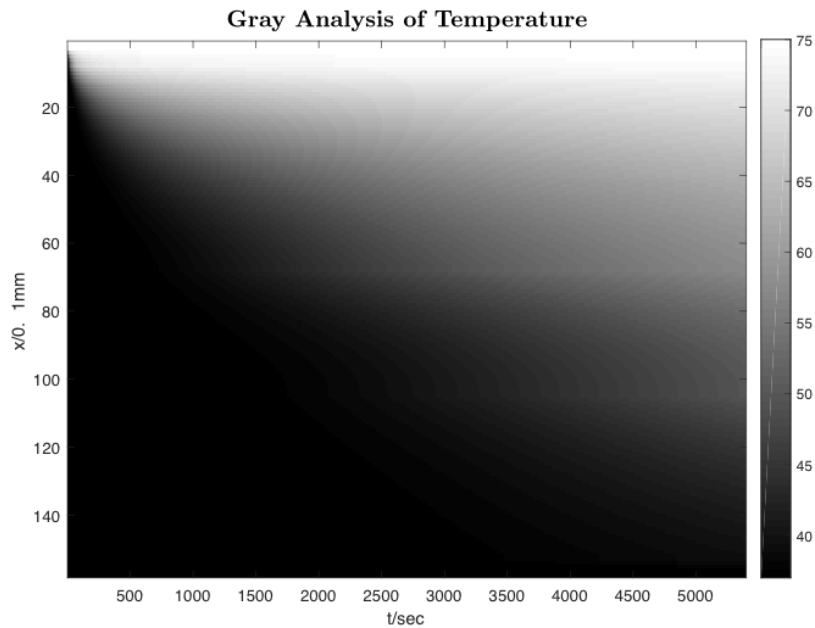
其中 s_i 为第*i*层的 $\frac{\alpha_i \Delta t}{\Delta x^2}$ ， α_i 为第*i*层的热扩散率

然后我们进行 MATLAB 数值模拟，得出结果如下图所示



图八、温度随时间和距离的曲面示意图(进阶版模型)⁶

可以看出经过模型的重构与限制条件的加入，温度分布有了明显的梯度性，较好地模拟出了包括空气间隙层的四层介质的温度分布图。

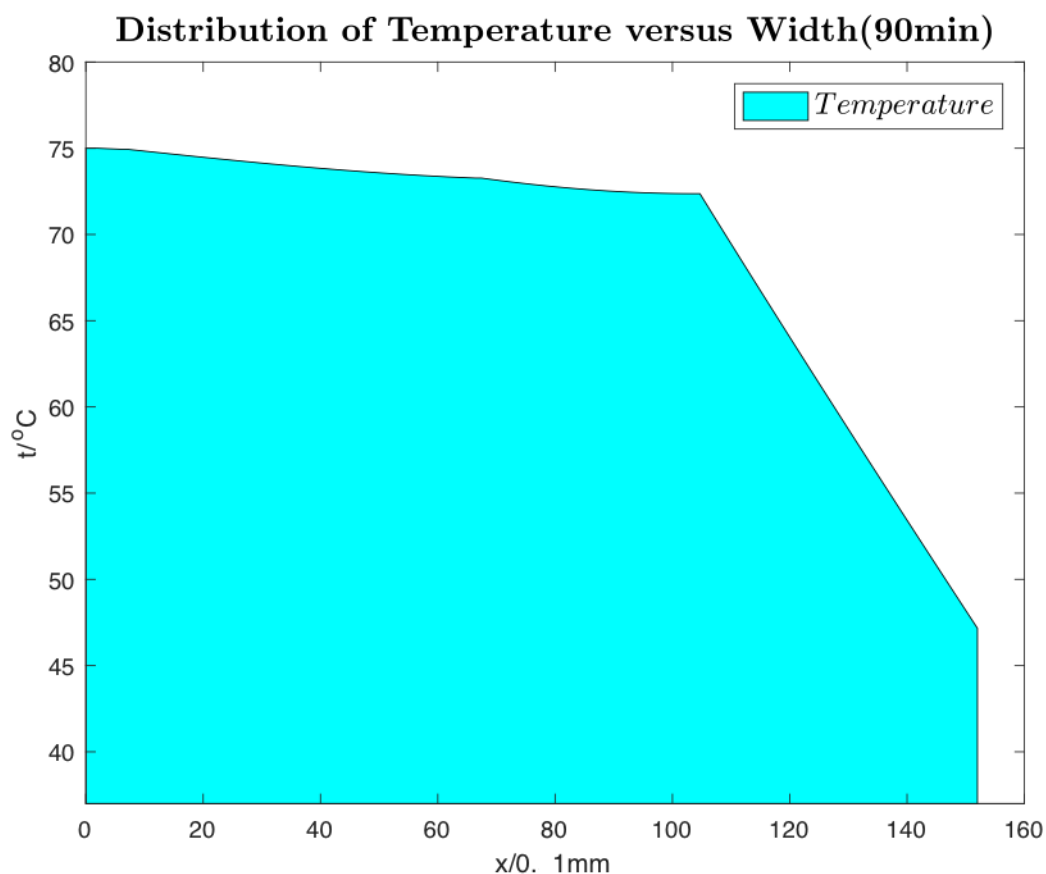


图九、温度矩阵灰度分析图

⁶代码如附录中 HT.m 所示

对于温度矩阵我们进行了灰度分析，可以看出随着时间的变化，各个厚度的温度都是有明显的变化趋势的，接下来，我们通过数值模拟出了第 90 分钟时的稳态温度随厚度的分布图，并且用了面积函数更好地体现出了整体的变化趋势。

根据上面的讨论，我们得到了最终达到稳态时的温度分布图：

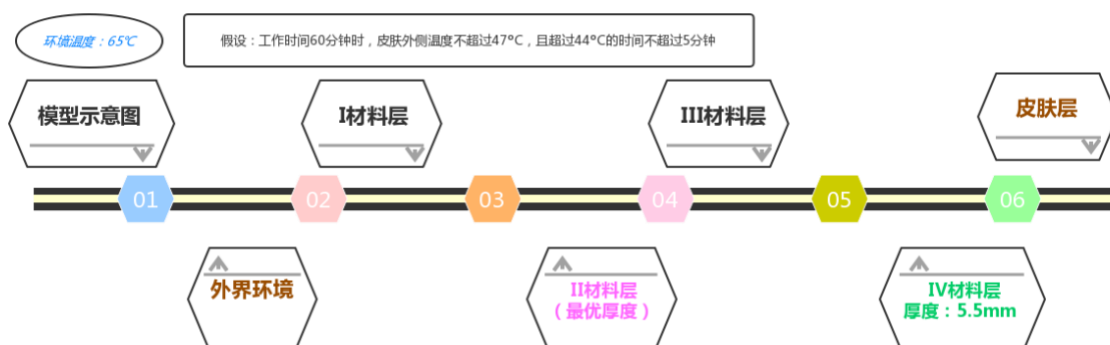


图十、稳态温度分布图⁷

我们可以看到在给定了四层介质相应的参数后，在规定的条件下 90 分钟后每一层的温度分布，在 1.05mm 处有一个明显的减缓趋势的变化，具体的数据可以参看附件中的 problem1.xlsx（在图中 0 的左侧部分是外界环境，15.2mm 的右侧部分是人体）

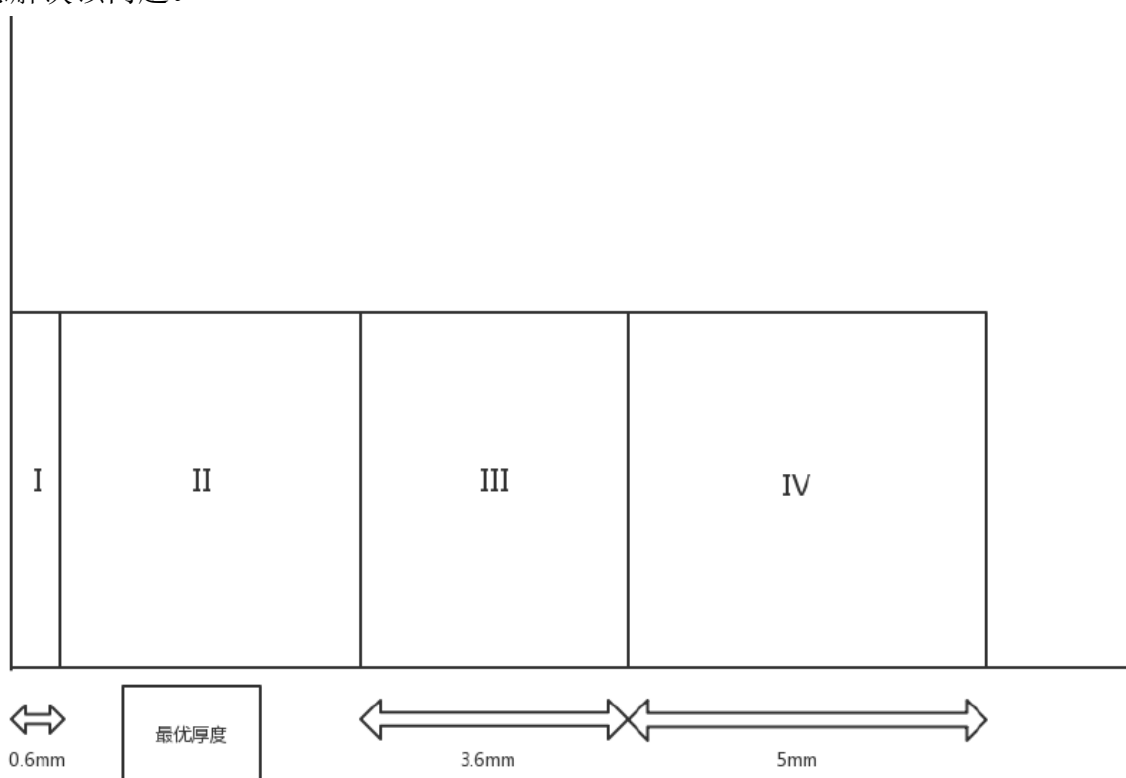
⁷ 代码见附录中 process_problem1.m

5.2 问题二的求解



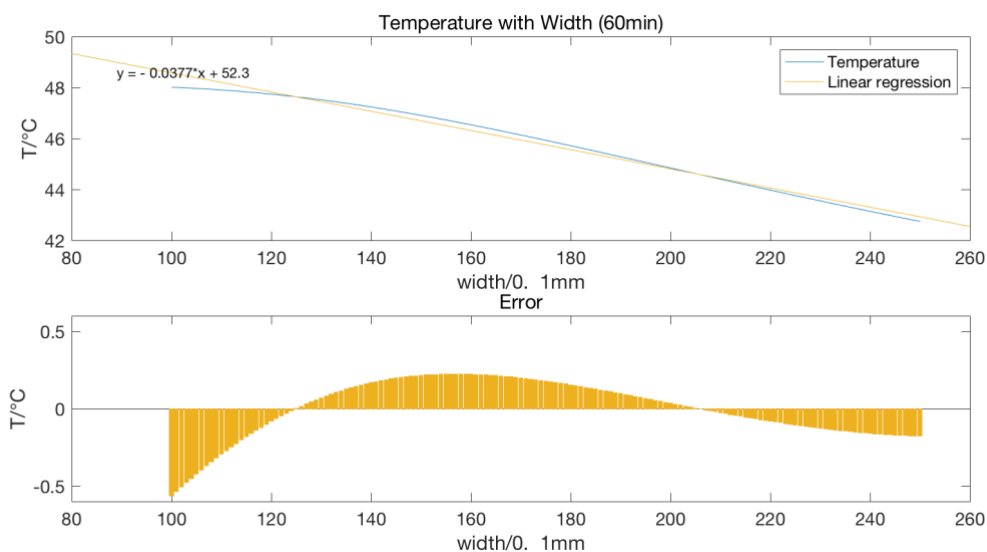
图十一、第二小问模型概念图

如图，我们要求解的是第 II 层的最优厚度，这个厚度在工作 60 分钟时需满足两个条件，亦即，假人皮肤外侧温度不超过 47°C ，并且同时超过 44°C 的时间不超过 5 分钟，其中第一个条件很容易满足，我们接下来着重考虑后一个条件，然后用一种对偶的思想解决该问题。



图十二、四种介质分布图（第二小问）

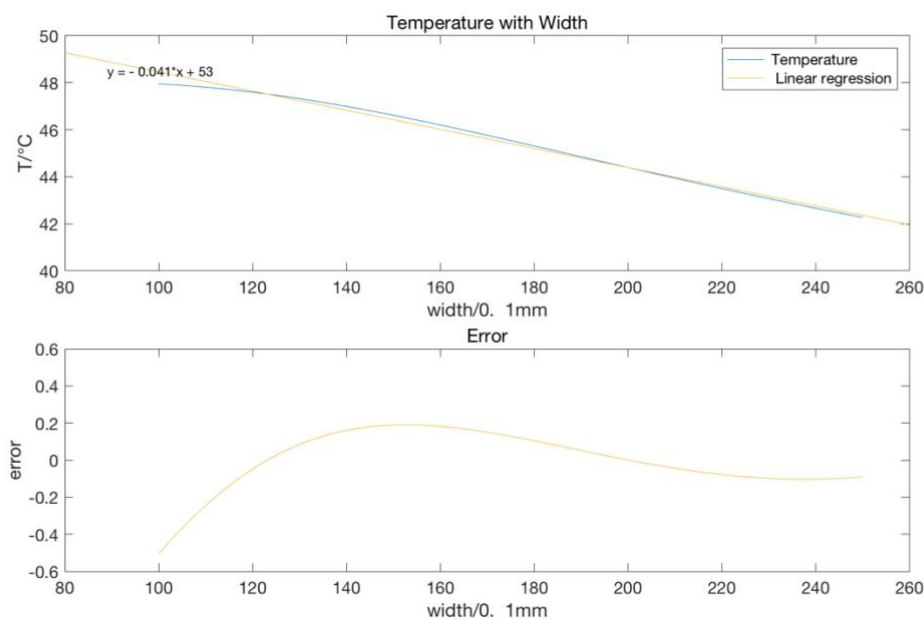
对于第一个条件，我们在环境温度为 65°C 、IV 层的厚度为 5.5 mm 且工作时间为 60 分钟时对第二层厚度进行遍历。得到在不同厚度情况下在 60 分钟后皮肤的表层温度情况，再对曲线作线性回归，获得当临界温度取 47°C 时的第 II 层厚度为 14.058mm 。如下图所示：



图十三、第二层厚度取不同值时 60 分钟后皮肤表层温度

从图中我们可以看出：当第 II 层的厚度超过 14.058mm 之后 60 分钟之后的皮肤层温度不超过 47°C。当第 II 层的厚度小于 14.058mm 之后 60 分钟之后的皮肤层温度大于 47°C。因此，设计要满足第一个要求，必须要求第 II 层的厚度超过 14.058mm。

对于第二个问题，要求超过 44°C 的时间不超过 5 分钟。由图像的单调性质，也就是说在第 55 分钟时，皮肤层的温度必须不超过 44 °C。我们在环境温度为 65°C、IV 层的厚度为 5.5 mm 且工作时间为 55 分钟时对第二层厚度进行遍历。得到在不同厚度情况下在 55 分钟后皮肤的表层温度情况，再对曲线作线性回归，获得当临界温度取 44°C 时的第 II 层厚度为 21.951mm。如图所示：



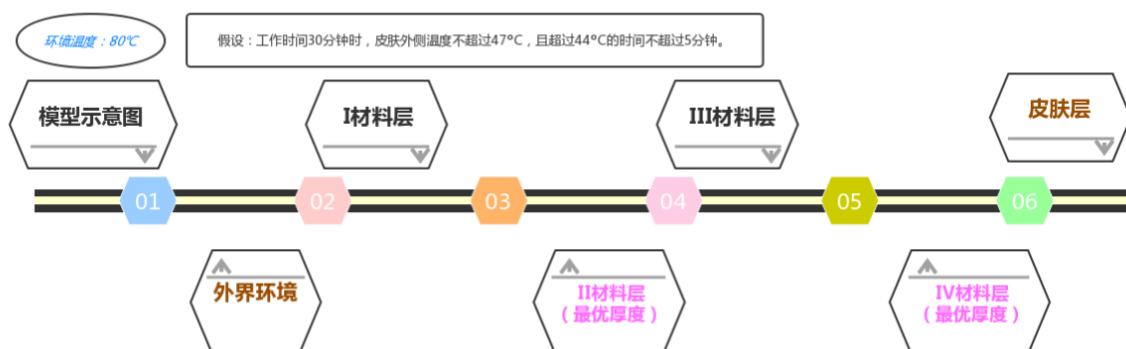
图十四、第二层介质第 55 分钟温度随厚度变化以及拟合残差图⁸

⁸ 代码参见附录 HT_second_problem.m 和 HT_plot.m

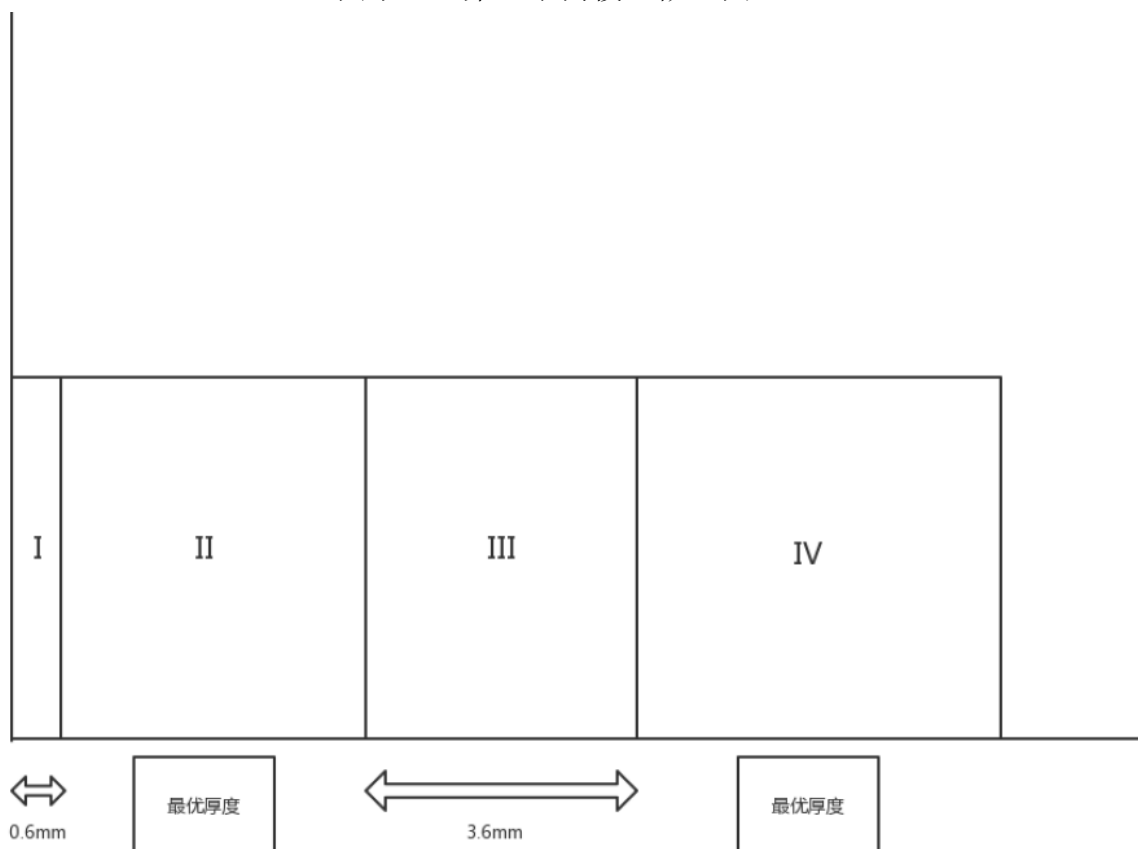
从图中我们可以看出：当第 II 层的厚度超过 21.951mm 之后 55 分钟之后的皮肤层温度不超过 44℃。当第 II 层的厚度小于 21.951mm 之后 55 分钟之后的皮肤层温度大于 44℃。因此，设计要满足第二个要求，必须要求第 II 层的厚度超过 21.951mm。

综上所述，第二个问题为满足材料最省的最优解为 21.951mm。

5.3 问题三的求解



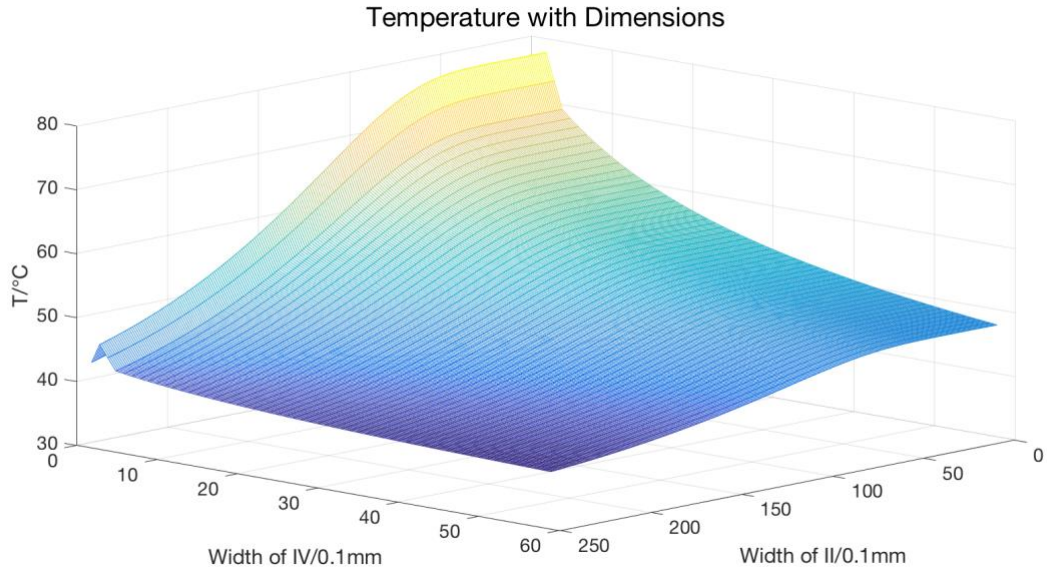
图十五、第三小问模型概念图



图十六、四种介质分布图（第三小问）

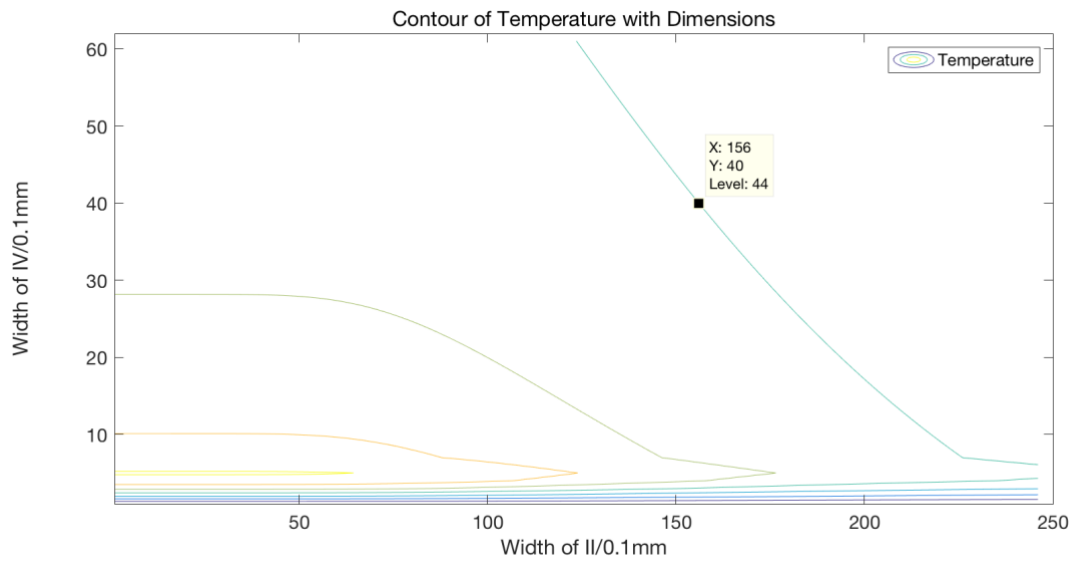
第三问中，我们要求解第二层和第四层的最优厚度，首先确定满足题目要求的范围。

受限于运算速度，我们先对不同的第二层第四层厚度等间距采样 25 分钟时皮肤表层温度。随后利用三次样条插值平滑数据，使曲面更加精细。



图十七、第 25 分钟时皮肤表层温度随第二第四层厚度变化

因为要使 30 分钟时，皮肤表层温度大于 44 摄氏度的时间不超过 5 分钟，故我们找到 25 分钟时不超过 44 摄氏度的区域即是我们要优化的范围。



图十八、第 25 分钟时皮肤表层温度随第二第四层厚度变化等高线

通过二次曲线拟合，我们找到了这个非线性的边界：

$$0.0021d_1^2 - 1.2681d_1 + 185.5530 - d_2 \leq 0 \quad (29)$$

其中 d_1 与 d_2 分别表示第二层与第四层厚度

接下来，我们讨论优化函数的设定。

我们知道第二层的厚度并不是越厚越好，材料的浪费也是我们的考虑因素。对于第

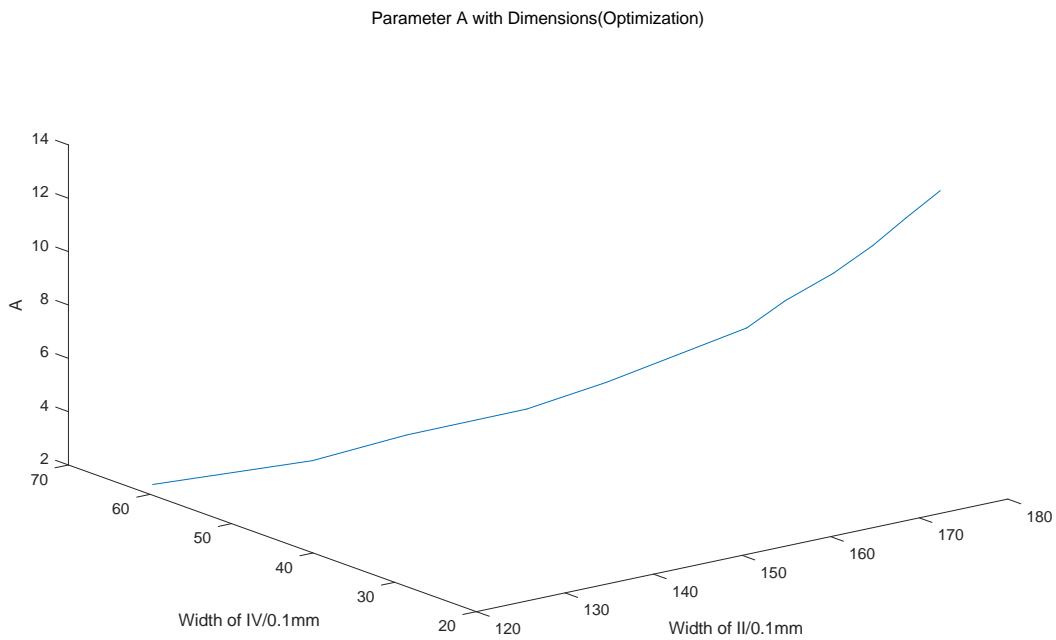
四层空气层来说，厚度越大有时会造成服装行动上的不便，所以在这里我们定义一个空气层的权重系数 A。

由此，我们的优化问题可以具体为：

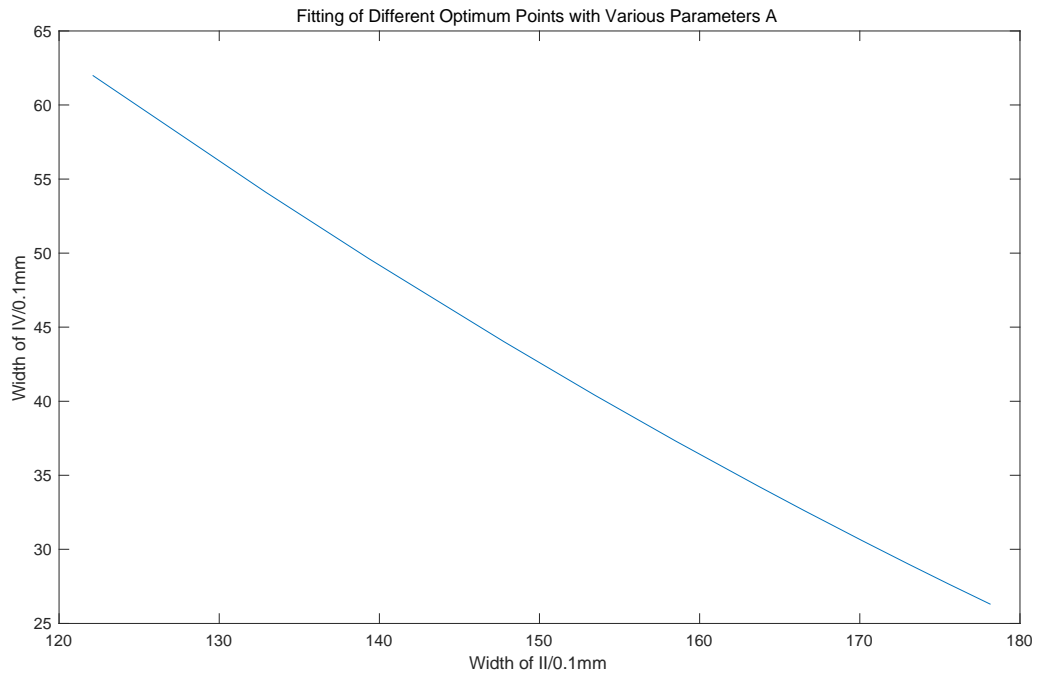
$$\begin{cases} \min f(d_1, d_2) = d_1^2 + A d_2^2 \\ d_1 \in (110, 250), d_2 \in (0, 62) \\ 0.0021d_1^2 - 1.2681d_1 + 185.5530 - d_2 \leq 0 \end{cases} \quad (30)$$

对于这个优化问题，我们采用遗传算法去寻找最优点，

经过 MATLAB 的数值模拟，我们找出了关于 II 层和 IV 层的最优解组合以及相应的拟合曲线，如下两图所示：



图十九、参数 A 随第 II 层、第 IV 层最优厚度的变化曲线

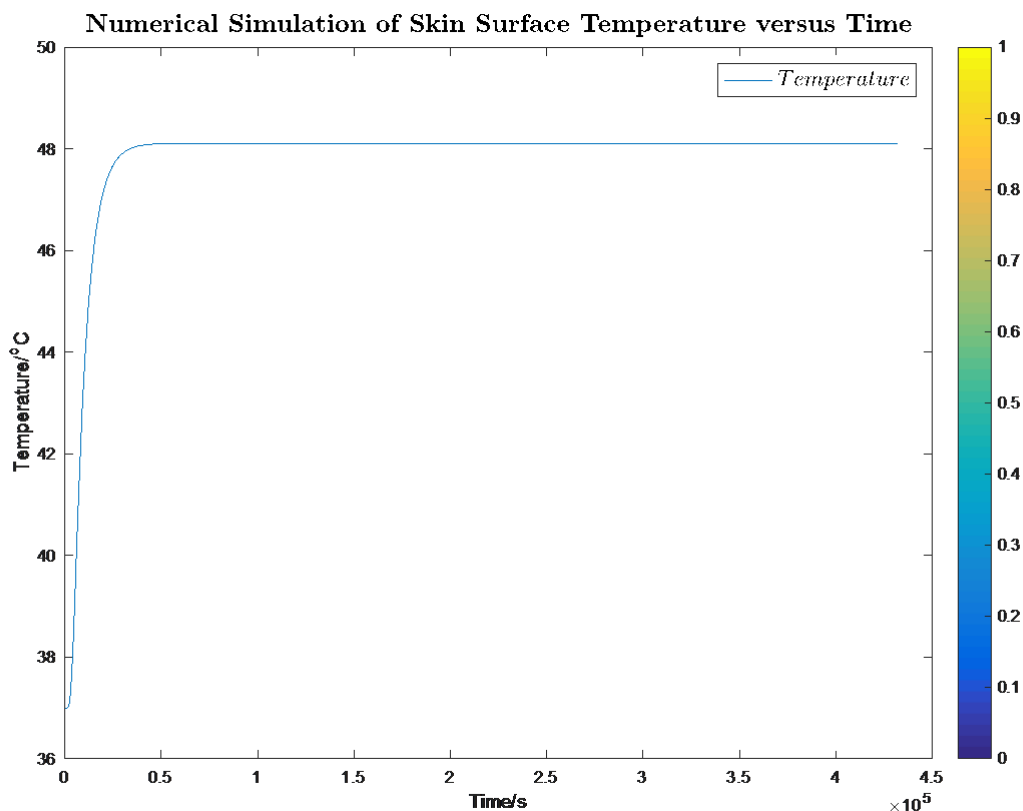


图二十、第 II 层、第 IV 层最优厚度点的拟合

6 模型的检验

6.1 模型的检验

根据我们在第一问建立的模型，我们可以画出数值模拟得到的皮肤表层温度随时间变化图，通过对比试验所得数据画出的图，可以知道我们所建立的进阶模型可以很好地拟合出温度的变化。



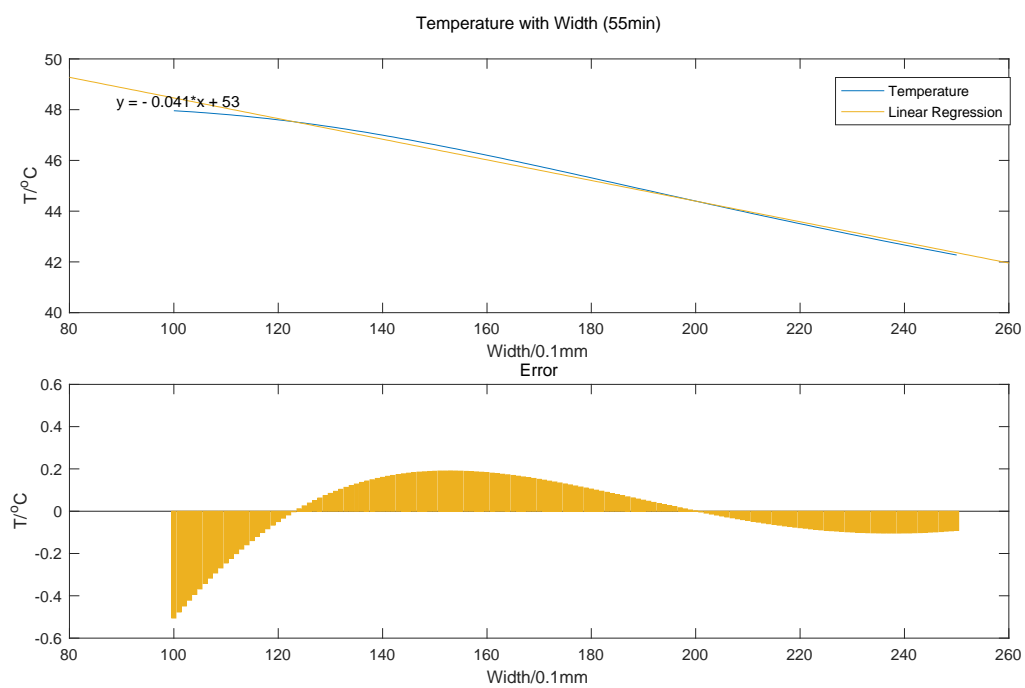
图二十一、数值模拟皮肤表层温度随时间变化图⁹

6.2 误差分析和改善

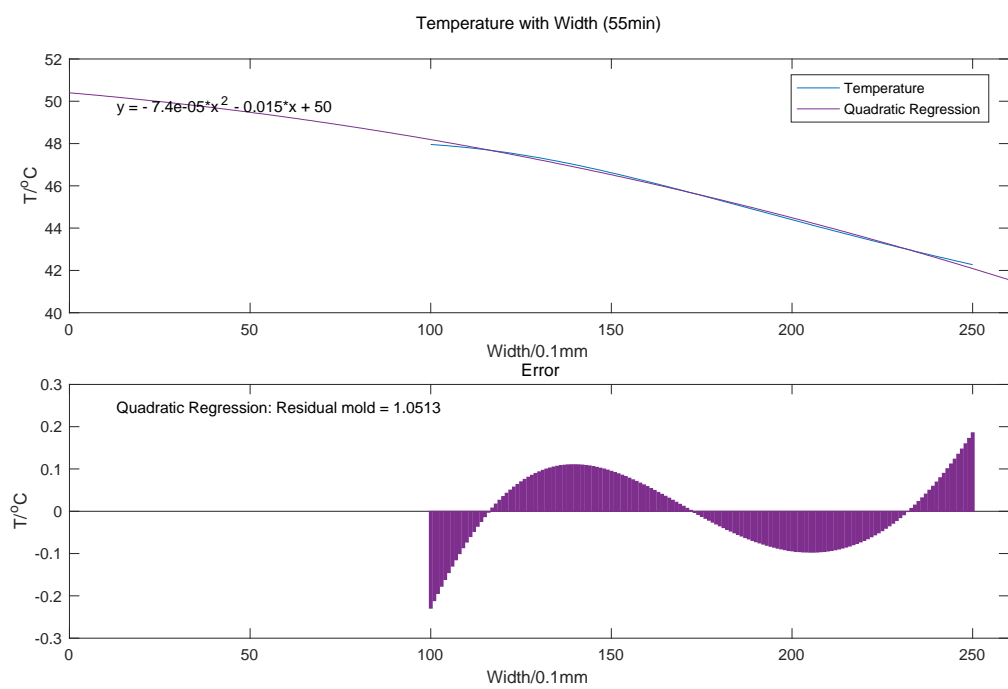
6.2.1 拟合分析

首先在解决第二题的时候，我们用了最小二乘法来拟合数据，主要因为最小二乘法有相当完备的统计性质并且拟合效果很满意，不过我们下面会对模型的不同拟合函数做敏感性分析，对比不同拟合函数算出的最优厚度。

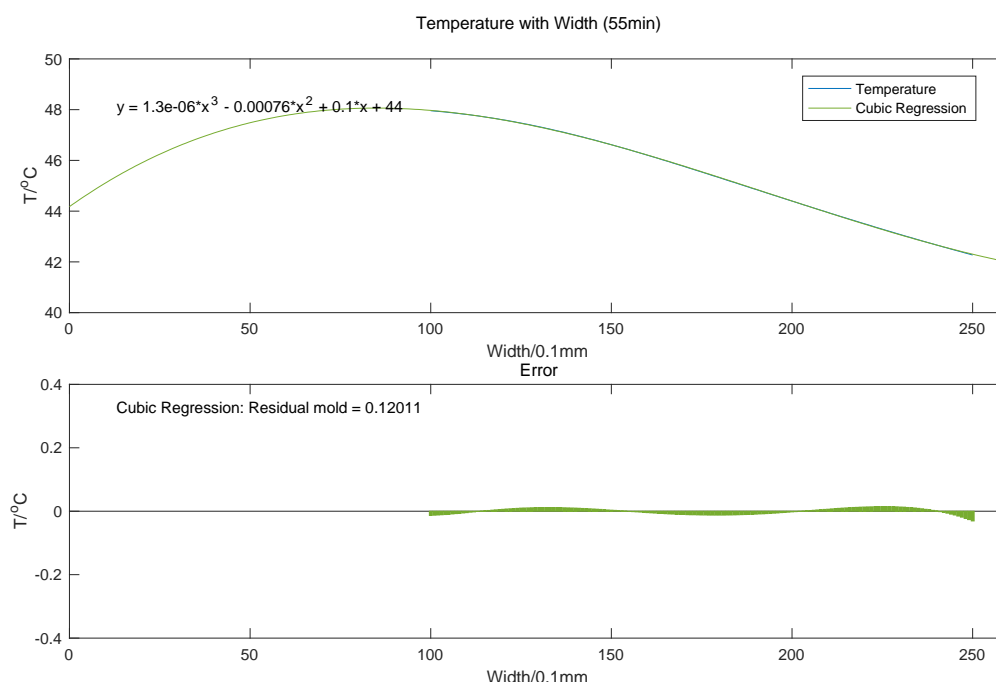
⁹ 代码见附录 HT_Advanced.m



图二十二、线性回归拟合与残差图



图二十三、二次多项式回归拟合与残差图



图二十四、三次多项式回归拟合与残差图

我们可以看出随着多项式的项数增加，残差的模也在逐渐降低，不过继续增加多项式拟合的项数，最终还会导致过拟合的现象，下面我们把三种拟合的相应数据整理在了下表中。

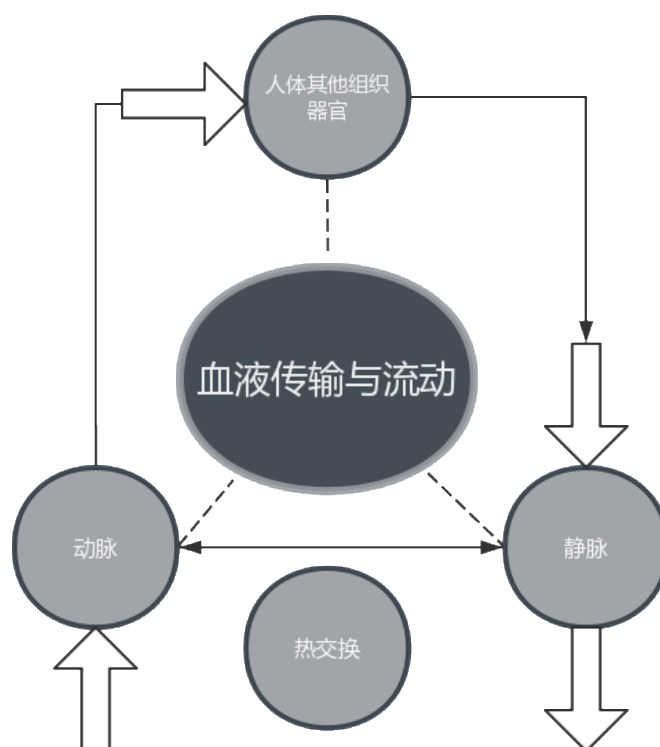
多项式拟合阶数	残差模	临界 44℃时第 II 层厚度(mm)
1	1.8706	21.9512
2	1.0513	20.8955
3	0.1201	20.0000

表一、多项式回归各相关参数对照表

容易看出随着阶数增加，其实过拟合问题是越来越严重的，三阶多项式极其以上在非拟合区域其实有明显的缺陷，并且很容易产生过拟合情况，故在表中不涉及三阶多项式以上拟合情况，对比结果可知用线性回归和二阶多项式回归的结果是较为精确的，也是符合趋势变化。

在上文第二问数值模拟中我们采用的是线性回归，其实可以看出二次回归也有一定的优势，比如从残差模可以看出二次回归的残差模略小于线性回归，并且临界温度时的厚度略小于线性拟合时的厚度，所以进一步优化模型时可以采用线性回归和二阶多项式回归结合的方法，具体选择需要看具体的参考标准与实际情况，最优厚度应该介于 20.8955mm 到 21.9512mm 之间。

6.2.2 人体内部温度优化分析



图二十五、人体内部热交换系统示意图

在人体内部我们都知道存在着较为复杂的热传导机制，题目中控制假人体内温度在 37°C 其实是有一定误差的，因为空气间隙这层介质也会与人的皮肤发生一定的热传递，所以我们对于模型的进一步改善是进一步考虑人体内部的结构对于影响外部四层介质的温度的关系。如上图所示，人体内部的热交换系统其实是非常复杂的，我们在此只是写出了一个大的框架图，把静脉，动脉与人体其他组织的相关关系通过概念图展示出来。

人体内体温会随环境而变，而不止是题目中理想化假人的恒定 37°C 。人体外侧温度在一开始升高时会有对温度的促进作用，而当外侧温度升高到一定的临界情况时，会对外侧的温度有一个抑制作用。这主要来源于人体内部其实是伴随着血液传输与流动，也间接影响了周围介质的温度。

从上文中我们数值模拟出的皮肤表层温度和本题给出的 excel 数据（附件 2）中的数据其实还是存在一些差异的，我们觉得把人体内部热交换考虑在模型内，会改善模型的精确性。

在这里我们就在一个临界温度，设为 T_0 ，在 $t < T_0$ 时对于温度有一个促进系数

$\xi \in (1, \infty)$, 在 $t \geq T_0$ 时对于皮肤表层温度有一个抑制系数 $\eta \in (0, 1)$

所以我们之后在对比数值模拟的结果和真实情况下, 其系数在不同的时间 (也即是不同温度) 情况下 (此时温度未收敛) 是不相同的, 调整相应乘数可以使结果更加精确。

7 模型的评价与推广

7.1 模型的评价

7.1.1 模型优点

- 模型容易移植;
- 模型易于修改参数, 输入和输出参数明显;
- 代码封装完善, 应用到其他问题时很容易;
- 考虑到的误差比较合理, 有较大的实际意义
- 模型使用简单, 不用理会复杂的数学关系。

7.1.2 模型缺点

- 算法带有一定的主观性
- 没有考虑湿传递带来的热量传递, 结果可能有一定的偏差。

7.2 模型的推广

模型可以移植到其他与热量在多层介质中传导的有关的实验或产品研发中, 具有较好的可移植性。

8 参考文献

- [1]. 陈才生. 数学物理方程[M]. 北京:陈才生, 2008.
- [2]. 徐元钟译. 数学物理中的偏微分方程[M]. 上海:Tyn Myint-U, 1983.
- [3]. 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:卢琳珍, 2017.
- [4]. 崔芑. 基于热湿传递稳态模型的纺织材料多参数决定反问题[D]. 浙江:崔芑, 2014.
- [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:潘斌, 2017.
- [6]. D. A. Torvi and T. G. Threlfall, Heat Transfer Model of Flame Resistant Fabrics during Cooling after Exposure to Fire, J. Fire Tech., vol. 42, pp. 27–48, 2006.
- [7]. F. P. Incropera and D. P. DeWitt, Fundamentals of Heat and Mass Transfer, 5th ed., pp. 821–826, Wiley, New York, 2002.

9 程序附录

9.1 HeatTransfer.m

```
function HeatTransfer
    clc; clear all; close all
    global NUM
    [NUM,TXT,RAW] = xlsread('temperature.xlsx');

    global A1 A2 A3 A4
    length = 152;      % 厚度为15.2 mm
    T = 5400;          % 时间总长
    global alpha1 alpha2 alpha3 alpha4

    alpha1 = 6.5e-3;
    alpha2 = 6e-3;
    alpha3 = 6e-3;
    alpha4 = 6e-3; % 热扩散率

    dt = 1;            % 时间步长
    global dx
    dx = 0.1;          % 公式中的h, 即分割的宽度
    n = length/dx;     % 内部的点数
    m = T/dt;          % 总时间步

    A1 = diag(-2*ones(1,60)) + diag(ones(1,59),1) + diag(ones(1,59),-1);
    A1(60,60) = -1;
    A2 = diag(-2*ones(1,600)) + diag(ones(1,600-1),1) + diag(ones(1,600-1),-1);
    A2(1,1) = -1; A2(600,600) = -1;
    A3 = diag(-2*ones(1,360)) + diag(ones(1,360-1),1) + diag(ones(1,360-1),-1);
    A3(1,1) = -1; A3(360,360) = -1;
    A4 = diag(-2*ones(1,500)) + diag(ones(1,500-1),1) + diag(ones(1,500-1),-1);
    A4(1,1) = -1;

    A1 = alpha1 /dx^2*A1;
    A2 = alpha2 /dx^2*A2;
    A3 = alpha3 /dx^2*A3;
    A4 = alpha4 /dx^2*A4;
```

```

t = zeros(1,m+1);
u = 37*ones(n+2,m+1);
u(1,:) = 75;
u(n+2,:) = NUM(1:end,2)';

global k
for k=1:m
    t(k+1) = t(k) + dt;
    u(2:(60+1),k+1) = RK4(t(k),u(2:(60+1),k),dt,@MyFun1);
    u(62,k) = u(61,k+1);
    u(62:(660+1),k+1) = RK4(t(k),u(62:(660+1),k),dt,@MyFun2);
    u(662,k) = u(661,k+1);
    u(662:(1020+1),k+1) = RK4(t(k),u(662:(1020+1),k),dt,@MyFun3);
    u(1022,k) = u(1021,k+1);
    u(1022:(n+1),k+1) = RK4(t(k),u(1022:(n+1),k),dt,@MyFun4);
end

[x,y] = meshgrid(0:1:1521, 0:dt:T); mesh(x,y,u');
xlabel('x/0.01mm'); ylabel('t/sec'); zlabel('T/^oC');


function z1 = MyFun1(x,y)
    z1 = A1*y;
    b1 = zeros(60,1);
    b1(1,1) = 75;
    z1 = z1 + alpha1 /dx^2 .* b1;
end

function z2 = MyFun2(x,y)
    z2 = A2*y;
end

function z3 = MyFun3(x,y)
    z3 = A3*y;
end

function z4 = MyFun4(x,y)
    z4 = A4*y;
    b4 = zeros(500,1);
    b4(500,1) = NUM(k,2);
    z4 = z4 + alpha4 /dx^2 .* b4;
end

```

```

display(1)

end

```

9.2 RK4.m

```

function y1 = RK4(x0,y0,h,f)
    K1 = f(x0,y0);
    K2 = f(x0+0.5*h,y0+0.5*h*K1);
    K3 = f(x0+0.5*h,y0+0.5*h*K2);
    K4 = f(x0+h,y0+h*K3);
    y1 = y0 + h*(K1+2*K2+2*K3+K4)/6;
end

```

9.3 Temperature.m

```

clc
clear
data = xlsread('skin.xlsx');
time = data(:,1);
temperature = data(:,2);
plot(time,temperature);
grid on
title('\bf{Temperature with Time outside the  
Skin}','FontSize',16,'Interpreter','Latex');
h81 = legend('$Temperature$', 'Interpreter','Latex' ) ;
set(h81,'FontSize',14,'Interpreter','Latex');
xlabel('Time(s)','FontSize' ,14 , 'Interpreter','Latex');
ylabel('Temperature(° C)','FontSize' ,14 , 'Interpreter','Latex');

```

9.4 HT.m

```

function HT
    clc; clear all; close all
    global NUM

```

```

[NUM,TXT,RAW] = xlsread('skin.xlsx');

global A1 A2 A3 A4
length = 15.2;    %厚度为15.2 mm
T = 5400;        %时间总长
global alpha1 alpha2 alpha3 alpha4

alpha1 = 1.98e-3*2;
alpha2 = 2.04e-3*2;
alpha3 = 3.51e-3;
alpha4 = 2.36e-3*2; %热扩散率

dt = 1;          %时间步长
global dx
dx = 0.1;        %公式中的h, 即分割的宽度
% n = length/dx; %内部的点数
n = 152;
m = T/dt;        %总时间步

s = dt/dx^2;

t = zeros(1,m+1);
u = 45*ones(n+2,m+1);
u(1,:) = 75;
u(n+2,:) = NUM(1:end,2)';

for k=1:m-1
    for x=2:7
        s1 = s * alpha1;
        u(x,k+1) = (1-2*s1)*u(x,k) + s1*(u(x-1,k)+u(x+1,k));
    end

    u(8,k) = u(7,k);

    for x=8:67
        s2 = s * alpha2;
        u(x,k+1) = (1-2*s2)*u(x,k) + s2*(u(x-1,k)+u(x+1,k));
    end

    u(68,k) = u(67,k);

```

```

    for x=68:103
        s3 = s * alpha3;
        u(x,k+1) = (1-2*s3)*u(x,k) + s3*(u(x-1,k)+u(x+1,k));
    end

    u(104,k) = u(103,k);

    for x=104:n+1
        s4 = s * alpha4;
        u(x,k+1) = (1-2*s4)*u(x,k) + s4*(u(x-1,k)+u(x+1,k));

    end
end

[x,y] = meshgrid(0:1:153,0:dt:T); mesh(y,x,u');
ylabel('x/0.01mm'); xlabel('t/sec'); zlabel('T/^oC');
title('\bf{Temperature with Time and  
Distance (Advanced) }', 'FontSize',16, 'Interpreter', 'Latex');

display(1)

end

```

9.5 HT_Advanced.m

```

clc; clear all; close all
global NUM
[NUM,TXT,RAW] = xlsread('skin.xlsx');

global A1 A2 A3 A4
length = 15.2;
skin_length = 1;
T = 5400*8;
global alpha1 alpha2 alpha3 alpha4 alpha5

alpha1 = 1.98e-1*5;
alpha2 = 2.04e-1*5;
alpha3 = 3.51e-1*2;
alpha4 = 2.36/2;
alpha5 = 1.98e-1*1.95;

```

```

dt = 0.1;
global dx
dx = 0.5;
% n = length/dx;
n = 156;
m = T/dt;
p = 0.1;

s = dt/dx^2;

t = zeros(1,m+1);
u = 37*ones(n+2,m+1);
u(1,:) = 75;
u(n+2,:) = 37;

for k=1:m-1

    for x=2:7
        s1 = s * alpha1;
        u(x,k+1) = (1-2*s1)*u(x,k) + s1*(u(x-1,k)+u(x+1,k));
    end

    u(3,k+1) = u(2,k+1) + p*(u(2,k+1)-75);
    u(8,k) = u(7,k);

    for x=8:67
        s2 = s * alpha2;
        u(x,k+1) = (1-2*s2)*u(x,k) + s2*(u(x-1,k)+u(x+1,k));
    end

    u(68,k) = u(67,k);

    for x=68:103
        s3 = s * alpha3;
        u(x,k+1) = (1-2*s3)*u(x,k) + s3*(u(x-1,k)+u(x+1,k));
    end

    for x=104:152
        s4 = s * alpha4;

```

```

        u(x,k+1) = (1-2*s4)*u(x,k) + s4*(u(x-1,k)+u(x+1,k));
    end

    u(105,k+1) = u(104,k+1) + p*(u(104,k+1)-u(103,k+1));
    u(153,k) = u(152,k);

    for x=153:157
        s5 = s * alpha5;
        u(x,k+1) = (1-2*s5)*u(x,k) + s5*(u(x-1,k)+u(x+1,k));
    end
    u(156,k+1) = u(157,k+1) - p*(u(157,k+1)-37);

end

% [x,y] = meshgrid(0:1:157, 0:dt:T); mesh(x,y,u');
% xlabel('x/0.01mm'); ylabel('t/sec'); zlabel('T/^oC');
plot(linspace(1,54000*8,54000*8),u(152,1:54000*8))

h81 = legend('$Temperature$', 'Interpreter','Latex' ) ;
set(h81,'FontSize',14,'Interpreter','Latex');

title('\bf{Numerical simulation of skin surface temperature versus
time}','FontSize',16,'Interpreter','Latex');

% display(1)

```

9.6 process_problem1.m

```

data = xlsread('problem1.xlsx');
width=linspace(0,152,152);
area(width,data,'FaceColor','c','EdgeColor','k')
title('\bf{Distribution of Temperature versus
Width(90min)}','FontSize',16,'Interpreter','Latex')
xlabel('x/0.01mm');
ylabel('t/^oC');
h81 = legend('$Temperature$', 'Interpreter','Latex' ) ;
set(h81,'FontSize',14,'Interpreter','Latex');
ylim([37, 80]);% 只设定x轴的绘制范围

```

9.7 HT_second_problem

```
function result = HT_second_problem(width)

% width = 6 ~ 250
% global NUM
% [NUM,TXT,RAW] = xlsread('temperature.xlsx');

% global A1 A2 A3 A4
length = 15.2;      %材料和空气层厚度为15.2 mm
skin_length = 1;    %皮肤层和内部恒温热源相聚1mm
T = 5400*2;         %时间总长
% T = 540;
global alpha1 alpha2 alpha3 alpha4 alpha5

alpha1 = 1.98e-1*5;
alpha2 = 2.04e-1*5;
alpha3 = 3.51e-1*2;
alpha4 = 2.36/2;
alpha5 = 1.98e-1*1.95; %热扩散率

dt = 0.1;           %时间步长
global dx
dx = 0.5;           %公式中的h, 即分割的宽度
% n = length/dx;    %内部的点数
n = width + 97;
m = T/dt;           % 总时间步
p = 0.1;

s = dt/dx^2;

t = zeros(1,m+1);
u = 37*ones(n+2,m+1);
u(1,:) = 75;
u(n+2,:) = 37;

for k=1:m-1
```



```

for x=2:7
    s1 = s * alpha1;
    u(x,k+1) = (1-2*s1)*u(x,k) + s1*(u(x-1,k)+u(x+1,k));
end

u(3,k+1) = u(2,k+1) + p*(u(2,k+1)-75);
u(8,k) = u(7,k);

for x=8 : 8 + width
    s2 = s * alpha2;
    u(x,k+1) = (1-2*s2)*u(x,k) + s2*(u(x-1,k)+u(x+1,k));
end

u(9 + width,k) = u(8 + width,k);

for x=9 + width: 9 + width + 35
    s3 = s * alpha3;
    u(x,k+1) = (1-2*s3)*u(x,k) + s3*(u(x-1,k)+u(x+1,k));
end

for x=9 + width + 36:9 + width + 36 + 48
    s4 = s * alpha4;
    u(x,k+1) = (1-2*s4)*u(x,k) + s4*(u(x-1,k)+u(x+1,k));
end

u(width + 46,k+1) = u(width + 45,k+1) + p*(u(width + 45,k+1)-
u(width + 44,k+1));
u(9 + width + 36 + 49,k) = u(9 + width + 36 + 48,k);

for x=width + 94 :width + 98
    s5 = s * alpha5;
    u(x,k+1) = (1-2*s5)*u(x,k) + s5*(u(x-1,k)+u(x+1,k));
end
u(width + 97,k+1) = u(width + 98,k+1) - p*(u(width + 98,k+1)-37);

end

% [x,y] = meshgrid(0:1:157, 0:1:T*10); mesh(x,y,u');
% xlabel('x/0.01mm'); ylabel('t/sec'); zlabel('T/^oC');

% plot(linspace(1,54000*2,54000*2),u(152,1:54000*2))

```

```

    % display(1)

    result = u(width+93,20*55*60); % 其中20*55*60中第二个参数修改即可画出不同时间
    间的图片

end

```

9.8 HT_plot.m

```

clc; clear all; close all
temp = zeros(1,250);
for i=100:250
    % temp(1,i) = HT3(i,49);
    temp(1,i) = HT_second_problem(i);
    if mod(i,15) == 0
        display('==== loading =====')
    end
end
plot(linspace(100,250,151),temp(1,100:250))

```

9.9 HT3.m

```

function outcome = HT3(width2,width4,time)

% width2 = 6 ~ 250
% width4 = 6 ~ 64
% global NUM
% [NUM,TXT,RAW] = xlsread('temperature.xlsx');

% global A1 A2 A3 A4
length = 15.2; %材料和空气层厚度为15.2 mm
skin_length = 1; %皮肤层和内部恒温热源相聚1mm
T = 3001; %时间总长
% T = 540;
global alpha1 alpha2 alpha3 alpha4 alpha5

alpha1 = 1.98e-1*5;

```

```

alpha2 = 2.04e-1*5;
alpha3 = 3.51e-1*2;
alpha4 = 2.36/2;
alpha5 = 1.98e-1*1.95; %热扩散率

dt = 0.1; %时间步长
global dx
dx = 0.5; %公式中的h, 即分割的宽度
% n = length/dx; %内部的点数
n = width2 + 48 + width4;
m = T/dt; %总时间步
p = 0.1;

s = dt/dx^2;

t = zeros(1,m+1);
u = 37*ones(n+2,m+1);
u(1,:) = 80;
u(n+2,:) = 37;

for k=1:m-1

    for x=2:7
        s1 = s * alpha1;
        u(x,k+1) = (1-2*s1)*u(x,k) + s1*(u(x-1,k)+u(x+1,k));
    end

    u(3,k+1) = u(2,k+1) + p*(u(2,k+1)-75);
    u(8,k) = u(7,k);

    for x=8 : 8 + width2
        s2 = s * alpha2;
        u(x,k+1) = (1-2*s2)*u(x,k) + s2*(u(x-1,k)+u(x+1,k));
    end

    u(9 + width2,k) = u(8 + width2,k);

    for x=9 + width2: 9 + width2 + 35
        s3 = s * alpha3;
        u(x,k+1) = (1-2*s3)*u(x,k) + s3*(u(x-1,k)+u(x+1,k));
    end
end

```

```

for x=9 + width2 + 36:9 + width2 + 36 + width4 -1
    s4 = s * alpha4;
    u(x,k+1) = (1-2*s4)*u(x,k) + s4*(u(x-1,k)+u(x+1,k));
end

u(width2 + 46,k+1) = u(width2 + 45,k+1) + p*(u(width2 + 45,k+1)-
u(width2 + 44,k+1));
u(width2 + 45 + width4,k) = u(width2 + 45 + width4 -1,k);

for x=width2 + 45 + width4 :width2 + 49 + width4
    s5 = s * alpha5;
    u(x,k+1) = (1-2*s5)*u(x,k) + s5*(u(x-1,k)+u(x+1,k));
end

u(width2 + 48 + width4,k+1) = u(width2 + 49 + width4,k+1) -
p*(u(width2 + 49 + width4,k+1)-37);

end

% [x,y] = meshgrid(0:1:157, 0:1:T*10); mesh(x,y,u');
% xlabel('x/0.01mm'); ylabel('t/sec'); zlabel('T/^oC');

% plot(linspace(1,54000*2,54000*2),u(152,1:54000*2))

% display(1)

outcome = u(width2 + 44 + width4,20*time*60);

end

```

9. 10 HT3_process.m

```
clc; close all
```

```

temp1 = zeros(50,19);
for i=1:5:250
    for j = 6:3:64

        temp1((i+4)/5,j/3) = HT3(i,j,25);

    end

    if mod(i,16) == 0
        display('==== loading ====')
    end
end

```

9. 11 fitness.m

```

function [y] = fitness( x )
y = x(1)^2 + 6*x(2)^2
end

```

9. 12 nlnlnl.m

```

function [c,ceq]=nlnlnl(x)
    c(1) = 0.0021*x(1)^2-1.2681*x(1)+185.5530 - x(2);
    ceq = [];
end

```

9. 13 gen_algorithm.m

```

x = ga(@fitness,2,[],[],[],[],[110;0],[250;62],@nlnlnl)

```

9. 14 Optimization_point.m

```

X = [122.1 132.9 139.35 147.79 153.47 158.50 163.66 166.50 170.10
173.02 175.52 178.16];
Y = [62 54.11 49.62 44.01 40.40 37.31 34.26 32.63 30.61 29.01 27.67

```

```
26.29];  
A = linspace(2,13,12);  
figure  
plot3(X,Y,A)  
figure  
plot(X,Y)
```