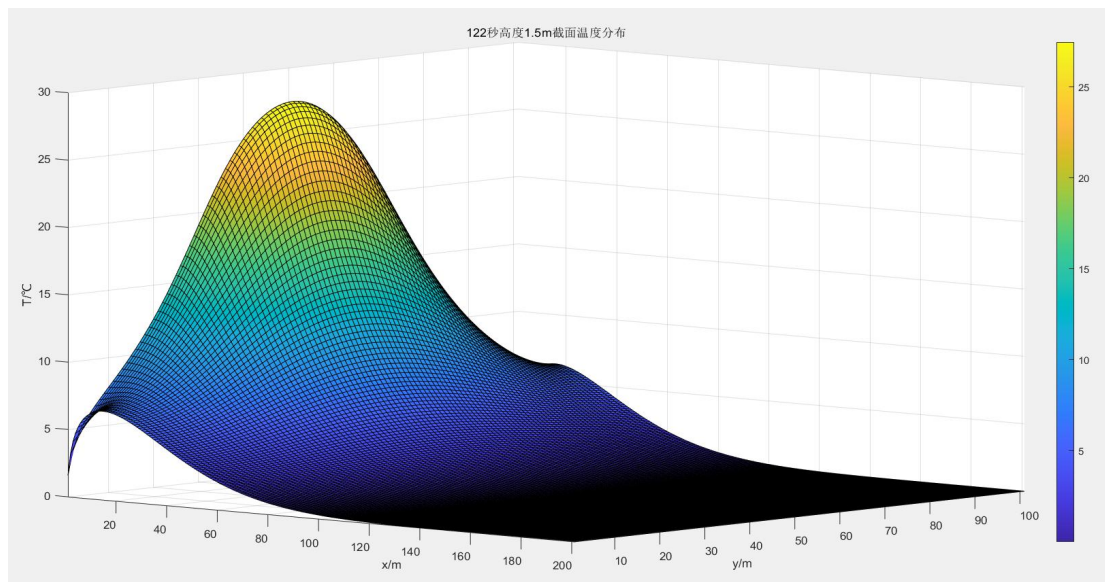


最优暖气温度自适性控制系统

在我国广大的北方地区，冬季温度十分寒冷，暖气作为我国北部地区最常见的供暖方式有着不可替代的作用。然而，大规模的持续供暖也消耗了大量的能源，为了提高暖气的供暖效率和用户使用的舒适性，同时减少无效能量的消耗，我们团队研究出一个暖气供暖最优策略算法，使用此算法可以为用户精确地提供其所需要的温度，同时大幅度降低暖气的能源消耗，并且随着使用时间的增加，调控效果会越来越好。

我们团队先是根据牛顿冷却定律、傅里叶导热定律和能量守恒等物理知识建立了理想化模型，确定了一般暖气房间的传热与温度模型。之后，我们将连续的物理过程和时间进行离散化，通过有限体积元的方法，依据最基本的物理原理模拟出整个房屋所有位置的温度随时间的变化。此模型可以帮助我们在输入为暖气的流量和温度时的得到房屋各点温度随时间的变化输出。因为我们的模型基于最基本的物理定律，离散过程的步长也十分精细，所以最终得到的模拟结果具有很高的参考价值。根据模拟结果我们发现温度的调控具有明显的滞后性。（下图为某特定房间某时刻一切面的温度分布）



根据我们建立的模拟模型，我们借助原始的 PID 控制算法，针对暖气传热的滞后性建立了一套新的控制系统。PID 的比例部分能够针对偏差立即产生控制作用，积分部分能够消除持续时间较长的微小偏差，微分部分能够进行早期修正。我们依据模型不同暖气流量下的模拟结果，通过变步长搜索得到了针对特定房屋的 PID 最优控制参数。同时，为了提高 PID 的控制效率，我们引入了分级调控，其可以在开启暖气后，以最快的速度达到用户所需要的温度，并使 PID 只负责较小范围的调控，提高参数的准确性和针对性。为了避免温度调控的不稳定性，我们使用最小调控时间间隔限定了调控的频率。

最后，我们根据长时间温度调试的输入、输出、室外温度变化的规律和主人设定温度的习惯，修正了分级 PID 控制，引入了“不同室外温度下达到目标温度稳态的暖气流量”，进而大幅度减少了暖气流量的调控次数，同时提高了结果的稳定性和控制的精确性。进一步提高了控制的节能性和用户使用的舒适性。

总结来说，我们的模型可以针对一个房间建立一个最优的暖气控制系统，通过长期的输入和输出修正模型，最终实现针对特定房间、特定室外温度变化规律和特定主人需求的暖气控制系统，达到节能与效能的统一。同时因为控制系统的建立是建立在最开始模拟模型的基础之上，而模拟模型因其可靠的物理机理和足够精细化的步长设置，具有很高的参考价值，所以我们的控制模型及研究结果也具有很强的可靠性。

基于 PID 自动控制的暖气房屋控温模型

摘要

本文针对冬季室内房间采暖问题，利用傅里叶传热定律等基本规律建立物理模型，并且采用有限元法，得出了任意时刻下房内任意位置的温度变化模型。在此基础上，使用分级 PID 自动控制方法，变步长搜索法以及二分法，计算出较优 PID 参数值，建立了基于历史数据的房屋温度调控模型，最终达到了控制均温在 22.90°C 的目标。

针对问题一，我们依据傅里叶传热定律，牛顿冷却定律，以及能量守恒定律，定量分析房屋内暖气片，内外墙，天花板和地面的传热性质。在这一基础上，利用有限元法，将空气作为基本单元进行分析，最终得出基于入水量的室内空气单元的热量传递公式和温度变化公式。在给定初始条件下，从多个角度展现了房屋温度传递过程。

针对问题二，我们调查前人的有关研究，使用分级 PID 控制方法对供热量进行调控。为了求取 PID 参数，采用变步长法进行求解，并且对于 PID 的调节频率做出限制。最终我们的模型在给定温度条件下，调节室内温度的均值为 21.86°C 方差为 1.753，在 24 小时内，系统总共调节 1454 次供暖水流。

针对问题三，我们利用二分查找法，确定了历史室外温度下的稳定供暖水流大小，对分级 PID 模型做出修改。修改后的室内温控模型结合了 PID 控制的灵活性，以及先验知识的稳定性，最终发挥出较原始模型更佳的效果，经过测试，屋内均值与设定值的误差减少到了 0.11°C ，方差也比原始模型减小 27%。

模型建立在较为严密的物理基础上，温度调节过程经过敏感性测试，对于外界温度变化仍能保持稳定，具有较好的推广性。

关键词: 有限元法 PID 自动控制 变步长搜索法

一、问题重述

暖气在我国北方地区被广泛使用，在寒冷的气候条件下，暖气可以调节室内温度，带来舒适的室内环境。在使用暖气的过程中，人们常常希望把温度控制在舒适的范围内。因此如何通过控制供热水流的大小来调节室内温度成为广大研究者关注的问题。

一种智能温控器被用于完成调节温度的任务。这一工具具有调节流量的作用，可以主动调节水流大小，以满足升高温度的需求，对于降低温度而言，只能关闭阀门留待暖气片中的水流自然冷却，温度回落到室温。该装置存在温度调节不智能，过度加热或冷却，温度波动变化大的缺点。

经过分析整理，我们需要解决以下问题：

1. 分析一个房间的采暖过程中的热量从供热管道到加热房间的过程，并建立模型来描述供热的水温与热水流量与房间内温度的关系。
2. 在给定波动的室外温度条件下，针对室内某个位置设定的目标温度，设计对供热水流的调节方法。以尽可能少的次数来调节水流大小，使得室内温度最快达到目标温度。并利用附件中的数据给出控制的结果，说明方法的合理性。
3. 假设温度控制器能记录下过去的调节结果，比如室外温度，用户设定温度，供热水阀门开闭程度，热水温度等。房间大小、房间中的属性、封闭程度、墙窗门的比例、墙壁保温程度等相对固定的。在上述条件下，对第二问中的控制方法进行优化。

二、问题分析

2.1 问题一的分析

问题一我们需要分析房间采暖过程的热量转移过程，要求我们就暖气这一供热装置的原理有所了解。为此需要查阅资料与相关参数，分析暖气片的热交换方式，以及分析进水温度 T_{wi} 和出水温度 T_{wo} 与热量传递 Q 之间的关系。为了建立模型，描述供水水温与热水流量同房间内温度的关系问题，需要就房间的整体结构进行分析。房间由墙体，窗户，以及天花板构成，部分墙体属于外墙与室外相接触，而有的墙体又属于内墙，内外墙之间的传热能量具有差异。此外，考虑空气导热性质不佳的特点，需要结合传热学知识，建立全屋的空气热量模型。

2.2 问题二的分析

针对问题二，我们需要在变化的环境温度这一条件下，动态控制热水的流入速率，使得室内温度尽可能达到要求的水平，且维持温度，使之波动保持在一定水平内。考虑到温度控制时知道当前时刻的室外温度，以及室内某处的实时温度和目标温度，我们需要控制暖水流来实时调节温度，以达到预期目标。

但是房屋温度的影响因素多样，于温度控制器而言相当于一个黑箱，并不知道其详细的工作原理。考虑使用 PID 自动控制的方法来自动调节温度，以达到在外界温度变化的情况下依然保持室内恒温的效果。可以利用在问题一中建立出的温度模型来模拟 PID 控制的情况，以判断某一组 PID 参数选取的优劣。在调节 PID 参数的过程中不可避免需要多次搜索确定最佳的参数组合，可以采取变步长法逐步减小参数的搜索范围，最终得到最优 PID 参数的组合，作为我们的调控结果。

2.3 问题三的分析

就问题三而言，此时温度控制不仅仅可以结合室内的当前状态，还可以就历史的数据进行温度控制调整。因此我们考虑结合问题 1 中的温度模型，记录每个室外温度下，室内保持 23 度的维持水流。并结合 PID 控制模型进行调整，最终得出优化后的温度调整模型。

三、模型假设

1. 暖气管内相邻两段水柱之间不发生热量交换。

原因：相邻两段水柱之间温度差较小，且表面积也小，暖气管内水流的热量散失主要是由向外界散发所导致。

2. 建筑内墙和地板认为是绝热体。

原因：这是对房屋的理想化假设，其原因为内墙两侧温差不大，因此认为没有热交换。由于地板之下可能是地基或下一层住户，也可认为温差不大不予考虑。

3. 室内窗户皆处于关闭状态。

原因：我国北方地区气候严寒，开窗的情况下难以保持室内温度，且开窗条件下室内外的空气流动，难以计算其热量交换数值。

4. 室内的气压保持稳定不变。

原因：虽然室内的窗户处于关闭状态，但是仍有缝隙保持气压平衡。

5. 分析全屋温度变化时，忽略暖气整体的宽度，认为其与墙壁融为一体。

原因：该做法有利于简化问题，利于分析屋内传热情况，且暖气片的厚度和房间进深相比较小，不构成太大影响。

四、符号说明

以下是本文使用的符号以及含义：

符号	说明	单位
t	时间	s
l	暖气中热水长度	m
v	供暖水流速度	m^3/h
S	暖气管横截面积	m^2
T	室内温度	$^{\circ}C$
T_n	供暖温度	$^{\circ}C$
T_{out}	室外温度	$^{\circ}C$
T_{aim}	目标温度	$^{\circ}C$
q	热流密度	$J/(^{\circ}C \cdot m^2 \cdot t)$
c	比热容	$J/(^{\circ}C \cdot kg)$
Q	热量	J

五、模型的建立与求解

以下将对提出的三个问题进行建模求解。

5.1 热水调控的房间温度模型

为了解决问题一，我们提出一种以房间热水流量和热水温度为输入，以最终房屋每处温度随时间的变化为输出的模型。为此，我们对暖气片的工作原理，房间的结构和传热特性进行分析。在得到房屋的热量传递模型后，结合牛顿冷却定律、傅里叶传热定律以及热量守恒定律，建立离散化的屋内温度模型。

5.1.1 暖气片供热模型

暖气片利用热水在管中流动的过程，加热其周围的空气，并传递热量到周围，是一种典型的间壁式散热器 [1]。在工作过程中，暖气片的热热水从一端流入，并且从另一端流出较冷的水，在内部为相连的若干 U 形管相连，如图（1）所示：

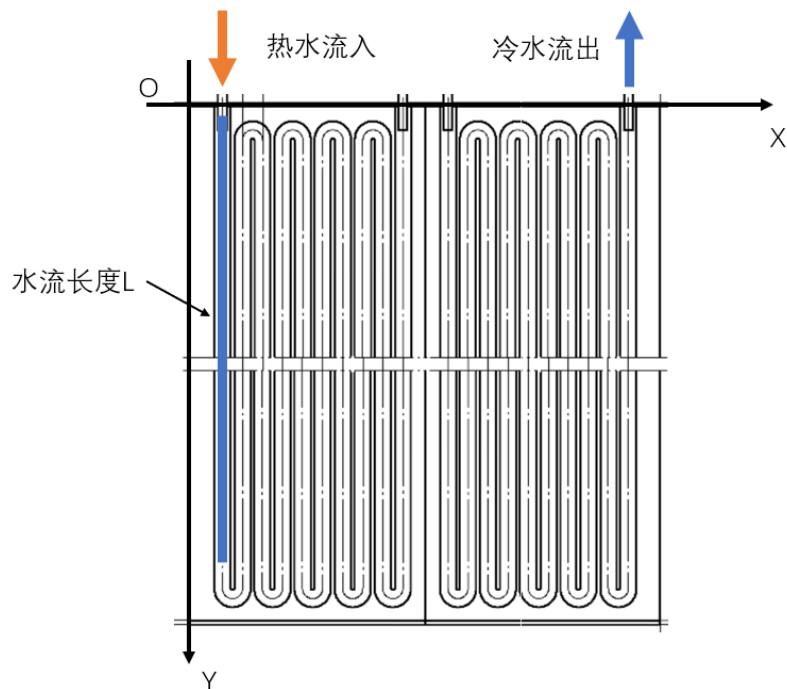
在 $t = 0$ 时，热水还未流入管内，此时管内水流温度等于室内温度 T ；

在经过时间 t_0 后，热水流入管中，此时管中热水长度 l 为：

$$l_0 = \int_0^{t_0} \frac{v(t)}{S} dt \quad (1)$$

其中 $v(t)$ 为热水流量，随调节过程随时间变化， S 为暖气管的横截面积。在水流长度为 l 的情况下，对应在图（1）中的直角坐标系，有唯一的坐标表示 $(x, y, z) = f(l)$ ，即：

$$\begin{cases} x = f_x(l) \\ y = f_y(l) \\ z = f_z(l) \end{cases} \quad (2)$$



说明：为展示方便， z 轴垂直纸面向内，隐去不画。

图 1: 暖气片内部水流示意图

取时间微元 $t \rightarrow t + dt$ 进行分析，此时长度变化为 dl ，使用水流速度 $v(t)$ ，横截面积 S 计算而出。

$$dl = \frac{v(t) * dt}{S} \quad (3)$$

考虑该段微元处的管道导热影响，我们画出其截面图（图??），对于该段柱体而言，可以假设管内的温度为 T_1 而管外温度为 T_2 ，观察暖气管的物理结构可知，满足长度远远大于其直径，可以看作是等温面共轴。

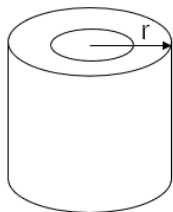


图 2: 水管截面图

在这一条件下，建立热传导微分方程：

$$\frac{d^2 T}{dr^2} + \frac{1}{r} \frac{dT}{dr} = 0 \quad (4)$$

其中 r 为水管半径，使用变量代换法，将上式变形：

$$u = \frac{dT}{dr} \Rightarrow \frac{du}{dr} = \frac{d^2T}{dr^2} \quad (5)$$

之后将原始微分方程变化为以下形式：

$$\frac{du}{dr} + \frac{u}{r} = 0 \quad (6)$$

可以通过分离变量法和两部积分法，可以求出对数形式的同届：

$$\begin{cases} \ln u + \ln r = \ln C_1 \\ T(r) = C_1 \ln r + C_2 \end{cases} \quad (7)$$

通过给定第一类边界条件 [2], 即表面温度分布，如下所示：

$$\begin{cases} r = r_1, & T = T_1 \\ r = r_2, & T = T_2 \end{cases} \quad (8)$$

可以计算出柱体内的最终温度场形式：

$$T(r) = T_1 - (T_1 - T_2) \frac{\ln r - \ln r_1}{\ln r_2 - \ln r_1} \quad (9)$$

利用傅里叶传热定律计算出表面的热流密度 ϕ ：

$$\phi = -\lambda \frac{dT}{dr} = \frac{\lambda}{r} \frac{T_1 - T_2}{\ln r_1 - \ln r_2} \quad (10)$$

经过上述分析，我们建立了水流经由暖气管向外散发热量的过程，在完成上述过程后分析一段微元 dl 处的热交换情况。由于微元处的长度很小，该段微元的温度可以由微元开始处的温度 $T_n(l, t_0)$ 所确定。我们利用牛顿冷却定律 [2]，在水柱微元和空气之间建立如下所示的传热关系：

$$\begin{aligned} q &= k_0 * (T(x, y, z, t_0) - T_n(l, t_0)) \\ &= k_0 * (T(f_x(l), f_y(l), f_z(l), t_0) - T_n(l, t_0)) \end{aligned} \quad (11)$$

其中 T, T_n 分别代表室内空气温度和暖气温度， q 代表热流密度，含义是单位时间内单位面积处发出或接受的热量。 k_0 是综合传热系数，将会在求解过程中给出。我们利用 q 可以求得一段水柱微元内的热量变化 dQ ，以及由于热量变化所导致的温度变化 dT_n ：

$$\begin{aligned} dQ &= q * C * dl * dt \\ &= q * C * v * dt^2 \end{aligned} \quad (12)$$

q 代表热流密度， C 为暖气管的截面周长， v 为热水管的流速，又依据比热容的计算公式，算出一段水柱的温度变化：

$$dT_n(l, t_0) = \frac{dQ}{c * s * dl * \rho} \quad (13)$$

在这里 c 代表水的比热容， S 代表管道截面积。由于水管中水流不断移动，相邻两段微元之间的温度差很小，近似认为不发生热交换，故在我们的模型中，该暖水微元的温度变化数值 dT 为后一时刻的同一段水柱温度变化，因此式 (13) 被解释为：

$$dT_n(l, t_0) = T_n(l + dl, t_0 + dt) - T_n(l, t_0) \quad (14)$$

也就是说在 dt 的时间间隔内，暖水微元移动了 dl 距离，但是在分析温差变化时，还是计算同一段水柱的变化数值。至此，我们已经阐述了流水在暖气管道中的热交换问题，在下面的内容中，我们将同样使用微分的方法，以能量守恒定律，牛顿冷却定律，傅里叶传热定律建立屋内空气变化模型。

5.1.2 房屋的传热学分析

对于一间房屋而言，其内部空气的温度受到墙壁种类，开窗的情况，以及天花板与外界接触情况影响，我们将对这些因素逐一进行分析。

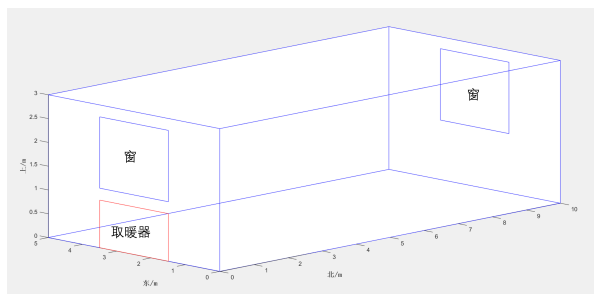


图 3: 房屋示意图

1. 对墙面的分析

注意到墙面依据所处的位置被分为内墙和外墙 [3]，划分的依据在于是否和室外空间相接触，我们将与室外空间接触的墙体视作为外墙，而不接触的称为内墙。对于外墙而言，外侧同室外的冷空气直接接触，造成温度较低，即使墙体具有部分阻隔冷气的作用，仍然会导致室内的墙体温度较低，带走室内的空气热量。对于内墙而言，内墙两侧都是室内空气，温度差值不大，可以近似认为内墙时绝热的，不会传导室内空气热量。

2. 对窗户的分析

外墙上常常装有窗户，与外界空气联通，因此会带来室内的空气交换与热量传递。即使是在没有打开窗户的情况下，热量交换也会透过玻璃进行。在分析房内温度的过程中，也应当对窗户加以考虑。

3. 对天花板和地面的分析

对于天花板而言，其与外界直接接触，本质同外墙没有区别，会向屋外散发热量。而分析可知地面温度随气温的变化程度较之于大气变化速度较慢，且有地基等因素的存在，确定地面的温度变化较为困难，因此将其认为绝热。

综合上述分析，如图（3）所示，我们对给定的矩形房屋建模，建立直角坐标系，标注其内外墙面的顶点坐标和对应类别，并以坐标形式呈现：

$$\begin{aligned}
&\text{内墙: } (x_{11}, y_{11}, z_{11}) \quad (x_{12}, y_{12}, z_{12}) \\
&\quad (x_{13}, y_{13}, z_{13}) \quad (x_{14}, y_{14}, z_{14}) \\
&\quad (x_{21}, y_{21}, z_{21}) \quad \dots \\
&\text{外墙: } (x_{11}, y_{11}, z_{11}) \quad (x_{12}, y_{12}, z_{12}) \\
&\quad (x_{13}, y_{13}, z_{13}) \quad (x_{14}, y_{14}, z_{14}) \\
&\quad (x_{21}, y_{21}, z_{21}) \quad \dots \\
&\text{窗户: } (x_{11}, y_{11}, z_{11}) \quad (x_{12}, y_{12}, z_{12}) \\
&\quad (x_{13}, y_{13}, z_{13}) \quad (x_{14}, y_{14}, z_{14}) \\
&\quad (x_{21}, y_{21}, z_{21}) \quad \dots \\
&\text{天花板: } (x_{11}, y_{11}, z_{11}) \quad (x_{12}, y_{12}, z_{12}) \\
&\quad (x_{13}, y_{13}, z_{13}) \quad (x_{14}, y_{14}, z_{14}) \\
&\text{地面: } (x_{11}, y_{11}, z_{11}) \quad (x_{12}, y_{12}, z_{12}) \\
&\quad (x_{13}, y_{13}, z_{13}) \quad (x_{14}, y_{14}, z_{14})
\end{aligned} \tag{15}$$

可以注意到，在房屋内的空气可以进行分析，对于那些直接同外壁接触的空气微元可能发生外部的热交换，而没有同外壁接触的空气微元则只进行空气微元的交换行为，这为我们分析空气传热问题提供了有利条件。

为了分析墙壁和玻璃窗的散热速率，我们计算其传热系数 K [2]，其计算公式如下：

$$K = \frac{1}{\frac{1}{h_1} + \frac{\sigma}{\lambda} + \frac{1}{h_2}} \tag{16}$$

其中 h_1 和 h_2 是材料的热交换系数， σ 是材料的厚度， λ 是材料的导热系数，依据这一公式可以求解出每平方米墙面或玻璃传递热量能力的大小。 K 的量纲为 $W/m^2 \cdot K \cdot s$ ，含义为单位时间内单位面积在内外相差 $1K$ （开尔文温度）时，传递热量的多少。利用上式可以推知，墙体（或窗户）和外界温度差为 ΔT 时，其热流密度为：

$$q = K * \Delta T \tag{17}$$

5.1.3 室内空气传热模型建立

在分析暖气片的工作原理以及房屋的传热分析后，我们就室内的空气进行传热分析，首先利用房屋中所建立的坐标系，分析在室内空气微元 dV 中的变化过程，其中 dV 表达如下，示意图如图（4）。

$$dV = [(x, y, z), (x + dx, y + dy, z + dz)] \tag{18}$$

在时间间隔 $[t, t + dt]$ 内，任何一个气体微元都满足能量守恒原理，该定理描述为微元内的能量改变只取决于传入传出能量的多少。在不考虑空气膨胀对外做功的情形，空气微元内能的变化直接反映在温度的变化上。综合以上分析，我们得出以下表达式：

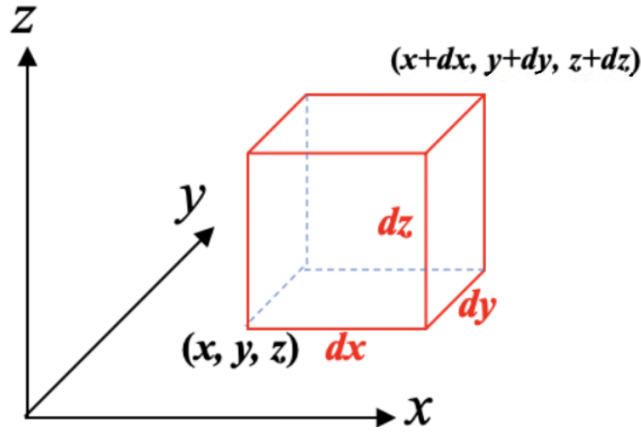


图 4: 气体微元示意图

$$Q_{in} - Q_{out} = \Delta Q \quad (19)$$

上式反映了气体微元内能变化，所带来的气体温度变化的关系，在此基础上，我们需要探究热量从高处向低处进行流动的过程，为此运用傅里叶传热定律 [1]，用于描述相邻热量的移动多少以及对应的传热率。

$$\vec{q} = -k\nabla T = \begin{cases} q_x = -k \frac{\partial T}{\partial x} \\ q_y = -k \frac{\partial T}{\partial y} \\ q_z = -k \frac{\partial T}{\partial z} \end{cases} \quad (20)$$

\vec{q} 为热流密度矢量，表示单位时间内沿单位面积的法向流出的热量。利用上式对微元左右、前后、上下三个方向的热量变化情况进行研究，得到下式：

$$\begin{aligned} Q_{左右} &= k \frac{\partial^2 T}{\partial x^2} dt dV \\ Q_{前后} &= k \frac{\partial^2 T}{\partial y^2} dt dV \\ Q_{上下} &= k \frac{\partial^2 T}{\partial z^2} dt dV \end{aligned} \quad (21)$$

考虑方向性，将式 (21) 中相加整理，即得到微元流入热量 Q ：

$$Q_{in} - Q_{out} = Q_{左右} + Q_{前后} + Q_{上下} = k \cdot \nabla^2 T dt dV \quad (22)$$

随后将 ΔQ 写为关于温度变化的形式：

$$\begin{aligned} \Delta Q &= c_p * \rho * dV * [T(t + dt, x, y, z) - T(t, z, y, z)] \\ &= c\rho \frac{\partial T}{\partial t} dt dV \end{aligned} \quad (23)$$

结合式 (19)，我们得到热量传递关于温度变化的公式：

$$k \cdot \nabla^2 T dt dV = c\rho \frac{\partial T}{\partial t} dt dV \quad (24)$$

在完成上述理论分析后，我们将对空气微元在房屋中的不同情况进行考虑，在空气微元处于房屋的内部，即微元相接触的仍然是空气，在这一情况可以认为微元在相邻的空气微元之间进行热交换，没有额外的热源进行吸放热。

而在房屋内侧边界处，也就是在式 (15) 中标注的坐标范围内，空气微元在同相邻微元进行热交换的同时，还受到边界的影响而从外部热源吸热和放热。这些热源包括放热的暖气片，吸热的外墙，窗户以及天花板，以及隔绝温度交换的内墙和地面。

我们利用 5.1.1 中的式 (11) 与式 (14) 分别描述了暖气片微元同空气之间的热流密度 q ，以及微元处温度变化同时间 t 和空间 l 的变化关系。根据 5.1.2 中吸热的外墙等进行分析，我们使用式 (16,17) 进行计算。

借鉴有限元 [4] 的分析思想，房屋空气任意时刻的温度分布函数 $T = f(x, y, z, t)$ 应当是一个连续的函数，但在热源房屋这一复杂情况下，难以求解其表达式。因此将室内空气划分为在房屋内部的单元，单元形状为长宽各为 $[dx, dy, dz]$ 的正六面体，如图 (??) 所示。

首先就房屋内部的微元进行分析，对于图 (5)，其热量转移方程 (式 23) 中的偏微分方程可以写作六个方向上的差分方程，在满足能量守恒定律的条件下，计算了空气单元处的温度变化：

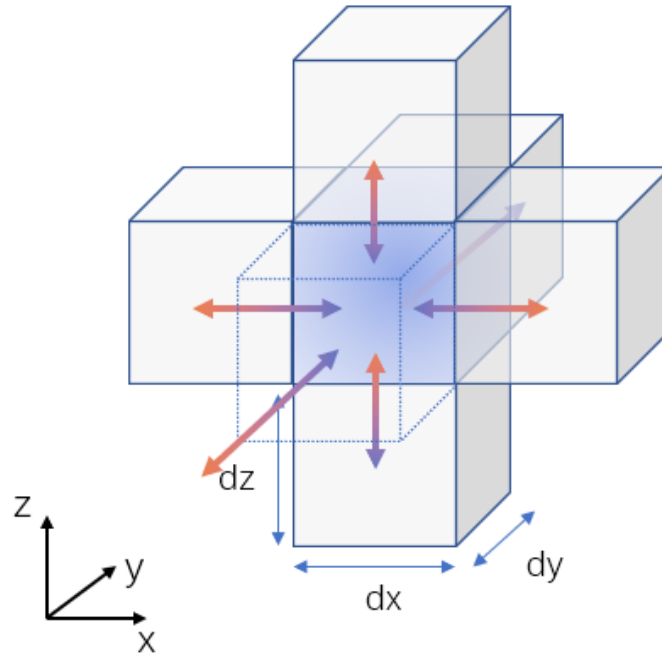


图 5: 室内微元分析

$$\begin{cases} Q_{x+} = k \cdot (T(x + dx, y, z, t) - T(z, y, z, t)) dydzdt \\ Q_{x-} = k \cdot (T(x - dx, y, z, t) - T(z, y, z, t)) dydzdt \\ Q_{y+} = k \cdot (T(x, y + dy, z, t) - T(z, y, z, t)) dx dz dt \\ Q_{y-} = k \cdot (T(x, y - dy, z, t) - T(z, y, z, t)) dx dz dt \\ Q_{z+} = k \cdot (T(x, y, z + dz, t) - T(z, y, z, t)) dx dy dt \\ Q_{z-} = k \cdot (T(x, y, z - dz, t) - T(z, y, z, t)) dx dy dt \end{cases} \quad (25)$$

对式 (25) 进行整理, 求解微元周围处热量变化综合求得任意一处微元在时间 $[t, t + dt]$ 下的热量变化:

$$\Delta Q(x, y, z, t) = k \cdot \left(\sum_{i \in c} T_i - 6 \cdot T(x, y, z, t) \right) \cdot a^2 \cdot dt \quad (26)$$

$$c = \{x+, x-, y+, y-, z+, z-\}$$

热量变化时, 微元对应的温度变化情况为:

$$\begin{aligned} \Delta T(x, y, z, t) &= \frac{\Delta Q(x, y, z, t)}{c_p \cdot \rho \cdot a^3} \\ &= k \cdot \frac{(\sum_{i \in c} T_i - 6 \cdot T(x, y, z, t)) \cdot dt}{c_p \cdot \rho \cdot a} \end{aligned} \quad (27)$$

综合上述过程, 我们已经得出内部单元的温度变化情况, 其热量传递和温度变化的过程又周围六个面的微元决定。

在分析完毕内部单元后，就和房屋边界处单元进行分析，房屋边界处单元示意图如下所示：

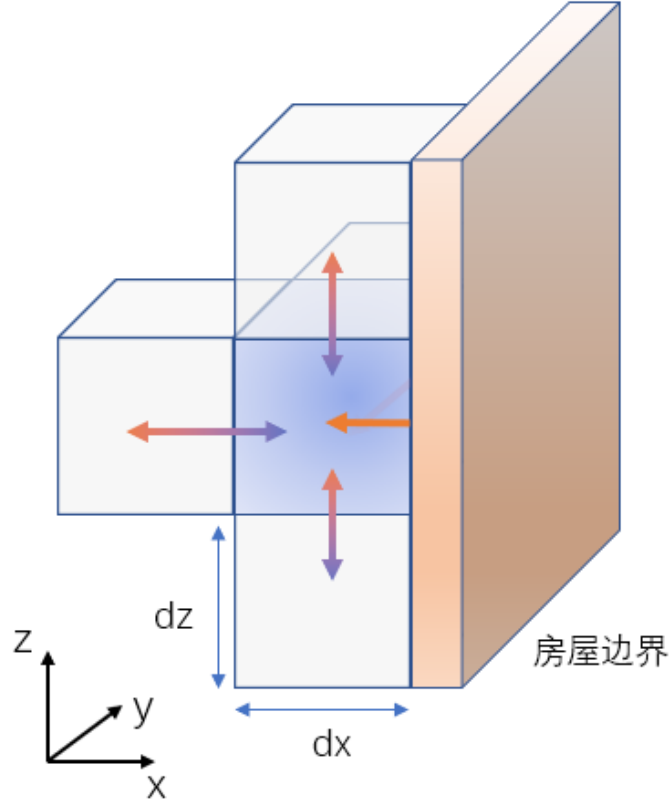


图 6: 房屋边界处空气单元示意图

外部单元在空间分布上处于式 (??) 划定的边界范围内，因此每个边界单元至少有一面不与空气接触，但是可能随着接触外壁种类的不同而发生不同的热交换行为，具体表现在接触面的热流密度 q 上，而在其余与空气接触的面中，热转移过程仍然满足式 (25) 中的部分规则。我们总结热量转移式如下：

$$\Delta Q(x, y, z, t) = [k \cdot (\sum_{i \in c-c'} T_i - |c'| \cdot T(x, y, z, t)) + \sum_{i \in c'} q_i] \cdot a^2 \cdot dt \quad (28)$$

$c = \{x+, x-, y+, y-, z+, z-\}$, c' 代表与房屋边界接触的单元面

其中 q 的取值根据边界分别为吸热边界（外墙，玻璃），绝热（地面，内墙），以及放热边界（暖气片）所决定，表述为下式：

$$q = \begin{cases} k_0 \cdot (T(f_x(l), f_y(l), f_z(l), t_0) - T_n(l, t_0)), & \text{放热边界} \\ 0, & \text{, 绝热边界} \\ \Delta T \cdot \frac{1}{\frac{1}{h_1} + \frac{\alpha}{\lambda} + \frac{1}{h_2}}, & \text{, 吸热边界} \end{cases} \quad (29)$$

边界单元的温度变化计算公式与内部相同，由式 (27) 给出。综合上述分析，我们已经建立了房屋内各个部分的空气单元的热量温度模型。要得出在任意时刻房间内任意地点的温度，需要在确定边界条件的情况下，模拟房间整体的传热过程，进行迭代计算。我们描述迭代计算的步骤如下，具体的计算结果，将在下一小节中给出。

步骤一： 给定房屋内外的初始条件，包括以下几点：

- (a) 初始室温，设定为同外部温度一致；
- (b) 进水管的流量以及入水温度；
- (c) 房屋暖气，窗户，墙壁等几何尺寸，对应的放热系数等必要参数；
- (d) 空气单元的尺寸和更新的时间步长。

步骤二： 依据初始条件遍历，利用式 (26,29) 计算每个单元处的热量转移量。

步骤三： 计算热量转移后利用式 (27) 来计算传递热量后空气单元的温度变化。

步骤四： 重复步骤二，直至计算处所需的时间长度。

5.1.4 屋内空气传热模型求解

我们在前面的工作中建立了屋内空气传热模型。给定初始条件以及边界条件，经过 matlab 求解后我们得到任意时刻内房屋各处的温度变化结果，展示结果如下：

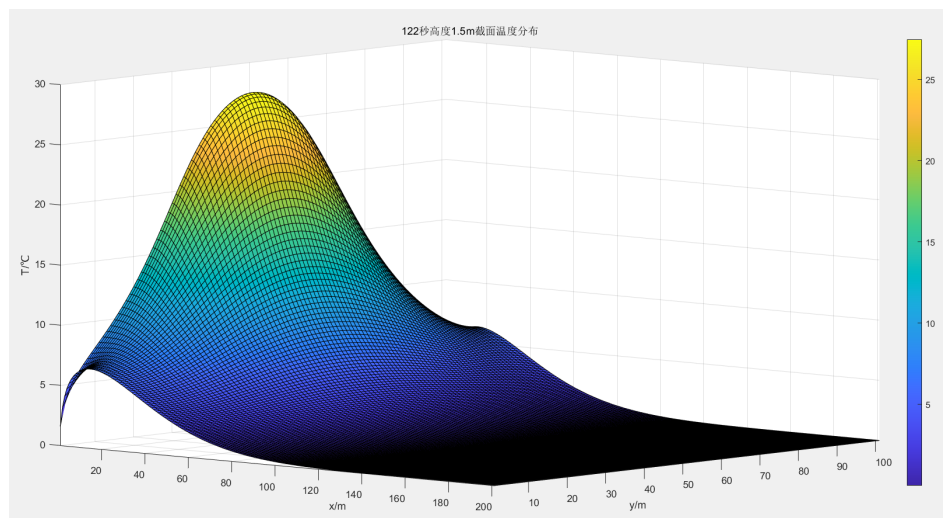


图 7: 122s 后高度为 1.5m 处的房屋截面温度分布图

我们在初始室温为 0°C ，室外温度为 0°C ，水流量为 $1\text{m}^3/\text{h}$ 的条件下，经过 122s 的模拟时间后，得出房屋在高度为 1.5m 处的截面温度。可见靠近暖气片一端房屋温度已经上升至 25 度以上，房屋北侧的空气此时还未受到传热，温度仍然不变。靠近南北墙壁处的温度皆不高，房屋此时最高温出现在房屋南北中轴靠南一侧，这与实际的情况相符合。

经过 500s 后，房屋温度整体上升，整体仍然呈现靠近暖气片处温度较高，四周温度低的特点。

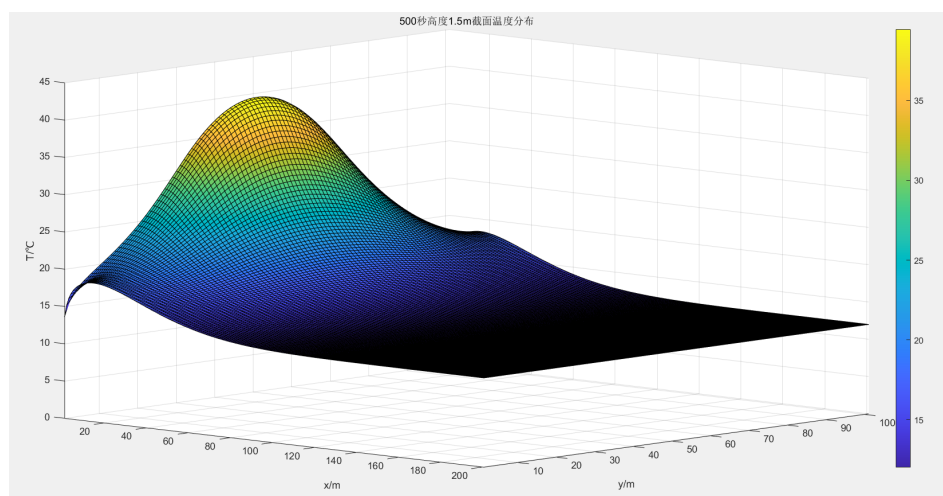


图 8: 500s 后高度为 1.5m 处的房屋截面温度分布图

为探究暖气片在竖直方向上的温度影响，绘制 x-z 方向上的房屋截面图（图 9），观察到其在竖直方向上呈现的温度关系，呈现为半马鞍形，有利于分析后续问题。

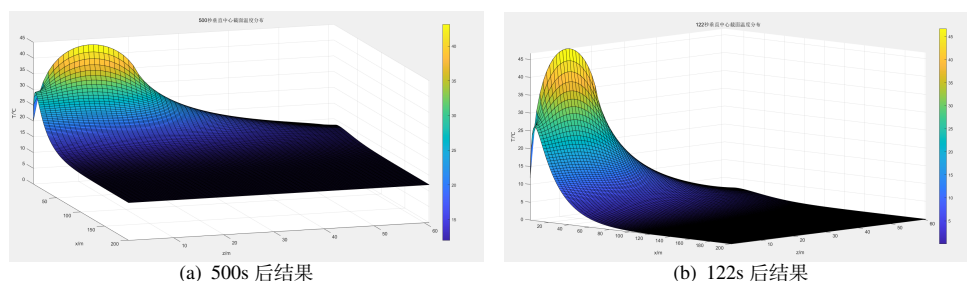


图 9: 房屋温度在 x-z 方向上的截面图

为了探究暖气片距离和温度变化的关系，我们观察打开暖气片后 1h 内距离暖气不同位置的温度变化，等距选取四个采样点，并绘制图（7）中相同条件下的温度变化曲线（图??）。

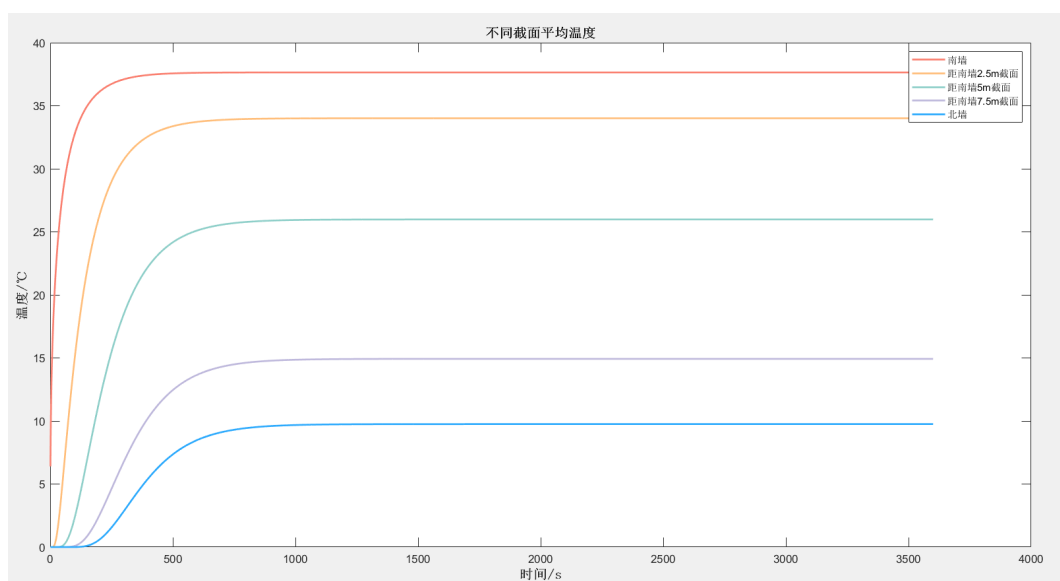


图 10: 沿南北方向上等距采样的温度变化结果

由图片结果可知，由于空气传热的特性，距离暖气不同距离的温度会保持在不同水平，距离暖气较近的空气温升快，且最终到达稳定时温度也较高，距离较远的地方速度则慢，最终也保持在较低的温度。针对这一调节过程延时较大的问题，我们将在下一章节设计自动控温模型，使得在外界温度变化时，内部温度也可以保持恒定。

5.2 温度动态控制模型

在这一部分，我们建立了室内温度动态调节模型，该模型使用 PID 调节方式对温度进行实时调节。在 PID 三个控制参数的确定过程中，建立 PID 调节模型的评价指标，使用变步长法进行参数搜索，最终得到较优的参数组合，并输出结果。

5.2.1 温度控制模型建立

在滞后效应明显的系统中，常常引入 PID 调节来控制输入，使得输出的结果保持稳定 [5]。分析暖气片加热室内温度的结果，观察到从室内外温度相同的时间开始，打开暖气热水，到室内温度趋于稳定不变时需要经历较长的时间。这是由于空气传热的滞后性，以及房屋边缘处存在散热现象。

PID[6] 的基本控制原理为反馈，主要由比例单元 (P)，积分单元 (I) 以及微分单元 (D) 构成，分别对应当前时刻的误差，历史累计的误差情况，未来可能产生的误差情况，一个典型的 PID 控制器如下图所示：

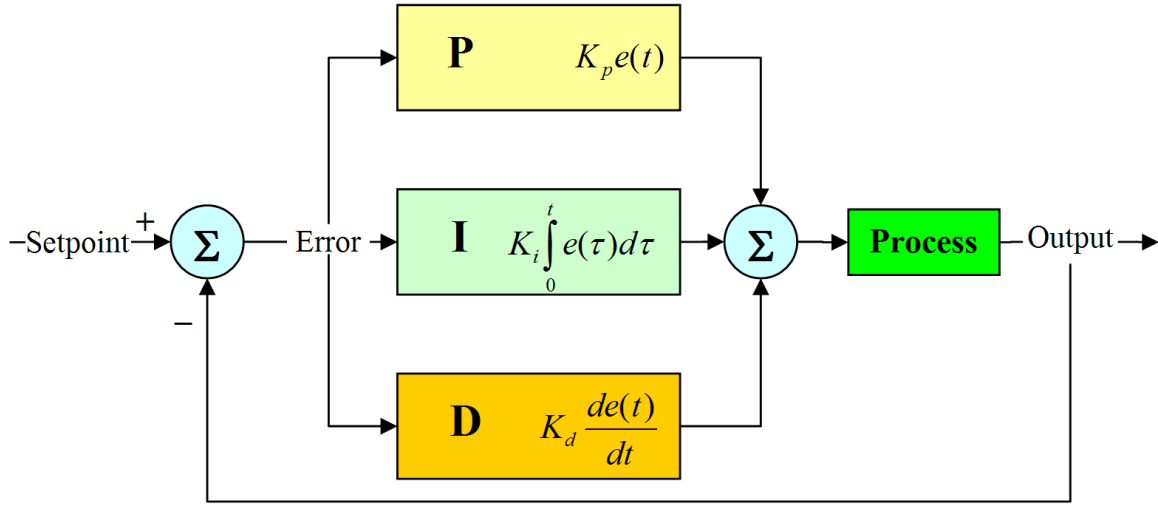


图 11: PID 控制器框图 [6]

PID 控制器的一般数学表达为：

$$u(t) = MV(t) = K_p(e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)) \quad (30)$$

其中 e 为观测值和预期值之间的误差，在本问题中，该误差 $e(t)$ 定义为当前室内温度和目标温度的差值，而输出被控量 $u(t)$ 对应暖气出水管处的流速，这与式 (12) 中热量变化计算过程相符。由于室内温度的采样具有间隔，因此不能当作连续函数那样进行积分和微分的计算，我们提出离散化的计算方法，给出如下计算公式：

$$v(t) = K_p(e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)) \quad (31)$$

$$\begin{cases} e(t) = T(x_0, y_0, z_0, t) - T_{aim} \\ \int_0^t e(\tau) d\tau = \sum_{i=0}^t e(i) \\ \frac{d}{dt} e(t) = e(t-1) - e(t) \end{cases}$$

其中 $T(x_0, y_0, z_0, t)$, T_{aim} 分别代表室内温度传感器的温度，以及设定的目标温度。依据附件信息传感器设置在东墙 1.5m 处的位置，设其坐标为 (x_0, y_0, z_0) ，东墙是内墙可以认为温度代表了室内温度。

考虑到题目所给要求，我们需要在尽可能快的时间内将温度调至设定的范围内，因此采用分级控制的思想规定在温度差处于 $[-\sigma, \sigma]$ 之间时才使用 pid 进行自动控制，而其余情况下将热水伐控制为最大打开或是关闭状态，给出如下计算式：

$$v = \begin{cases} v_{max}, & T(x_0, y_0, z_0, t) < \eta_{aim} - \sigma_{aim} \\ v(t), & \eta_{aim} - \sigma_{aim} < T(x_0, y_0, z_0, t) < \eta_{aim} + \sigma_{aim} \\ v_{min}, & \eta_{aim} + \sigma_{aim} < T(x_0, y_0, z_0, t) \end{cases} \quad (32)$$

其中 $v(t)$ 是 PID 的输出参数，已经由式 (31) 给出。这样调节的好处在于可以使温度在脱离范围时迅速回到设定的温度值。其中 v_{min}, v_{max} 由附件中数据给出，分别是 $0m^3/h$ 和 $2m^3/h$ ； η_{aim}, σ_{aim} 分别为目标调节温度的均值与方差，也由附件所给出。

在规定 PID 对水温的控制方式，给出误差项等观测指标的计算方法后，我们需要确定 K_p, K_i, K_d 三个参数的取值。在温度控制方面的 PID 算法参数设定方面，业界已经有较多研究。张夏夏 [6] 等人提出使用粒子群算法迭代求解最优 PID 参数，将三个参数独立处理，建立四个指标的评价规则，在轮船中央空调控温方面发挥了较大作用。金晶等人 [5] 提出一种结合衰减曲线法和循环遍历法的参数选取方法，在加热炉温度控制方面取得了较好成果。

综合前人的经验，我们利用附件中 24 小时内的温度变化以及 5.1 中建立的房屋温度控制模型，以实时室外温度作为模型中房屋的外部温度，使用式 (32) 对房屋暖气进水量进行控制。对一组 PID 参数 $[K_p, K_i, K_d]$ 进行 24 小时模拟，以温度保持在 $[\eta_{aim} - \sigma_{aim}, \eta_{aim} + \sigma_{aim}]$ 的时间 T_{remain} 作为调节效果的衡量指标。

为了减少单位时间内的参数调节次数，我们规定了单位时间内进行 PID 调节的频率上限为每分钟一次，这是为了避免连续多次调节，导致步进电机工作异常。

为了确定最优的参数组合，我们使用迭代搜索法确定各个参数，其基本思想为逐步搜索各个参数，逐步缩小搜索范围不断细化，直至满足要求。其流程图如下：

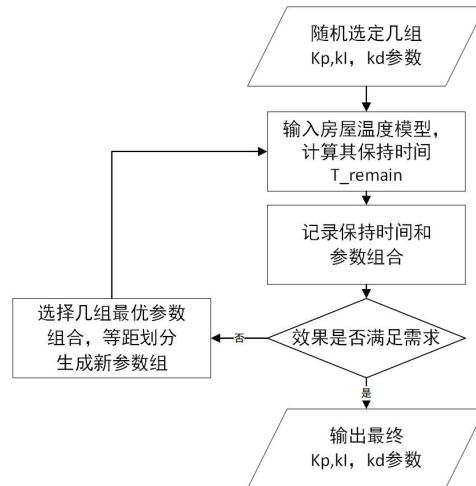


图 12: 计算最优 K_p, K_i, K_d 参数的流程

5.2.2 温度控制模型求解

经过多次测试，我们选取参数组合 $k_p = -0.01$ ， $k_i = 0.001$ 与 $k_d = 0.05$ 进行 24 小时模拟，最终获得如下模拟结果：

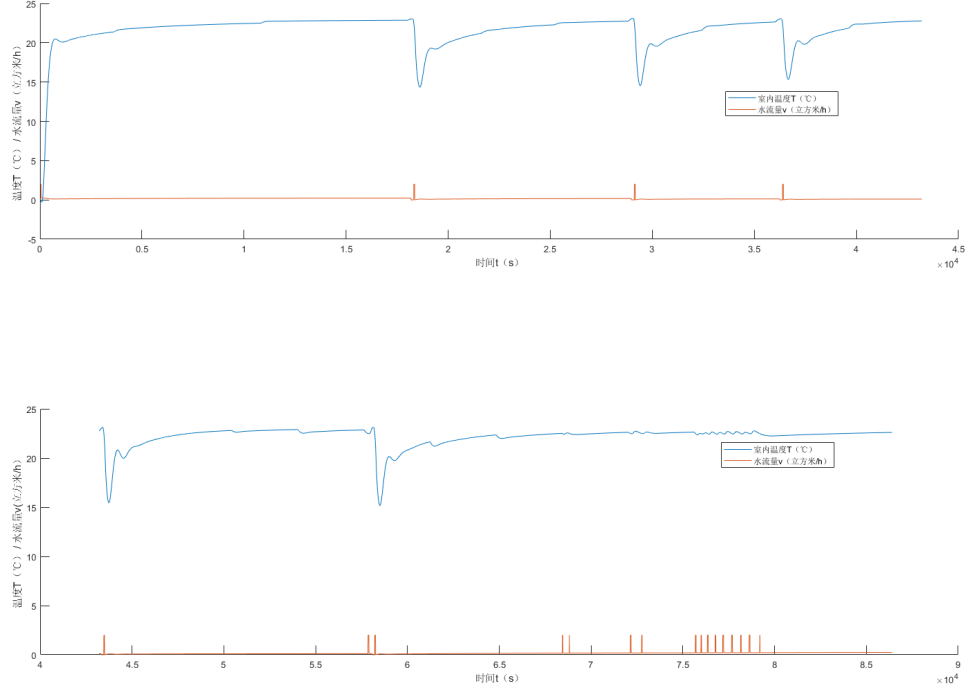


图 13: PID 控制下房间一日内的温度变化信息

可见在外界温度发生变化时，水流温度会做出调整，以维持室内温度保持一定范围之内，随后迅速恢复到设定的温度内。经过计算，在房屋温度从 0°C 开始，一直调控 24 小时的过程中，室内温度的均值为 21.86°C 方差为 1.753，在 24 小时内，系统总共调节 1454 次。所得结果能基本完成题目要求，在时间和算力允许的条件下，可以找出更佳的参数组合取得更好效果。

由于该温控模型由室内温度变化来控制水流，没有对于房屋属性以及过去信息的记录，存在部分时间点温度波动过大的行为，我们将在下一小节进行改进。

5.3 基于历史温度数据的温度控制模型

5.3.1 改进温度控制模型的建立

鉴于上一节中的 PID 控制模型效果依赖于参数的选择，最终温度控制的过程没有结合先前历史温度等信息进行修正，这导致在温度变化的时刻发生了较多的抖动。为了改善这一现象，我们使用问题一中的空气传热模型，计算出在不同室外温度 T_{out} 条件下室内稳定温度为 23°C 时对应的暖水流速，称为维持流速以 V 表示。

为了得到不同室外温度 i 下的流量 V_i 数值，使用二分法进行求解。我们固定室外温度分别为 $i^{\circ}\text{C}$ ，选取区间端点为 $[v_{min}, v_{max}]$ ，计算端点处的最终室内稳定温度，并不断减半区间，最终得到精度范围内的暖水流速。其计算步骤如下：

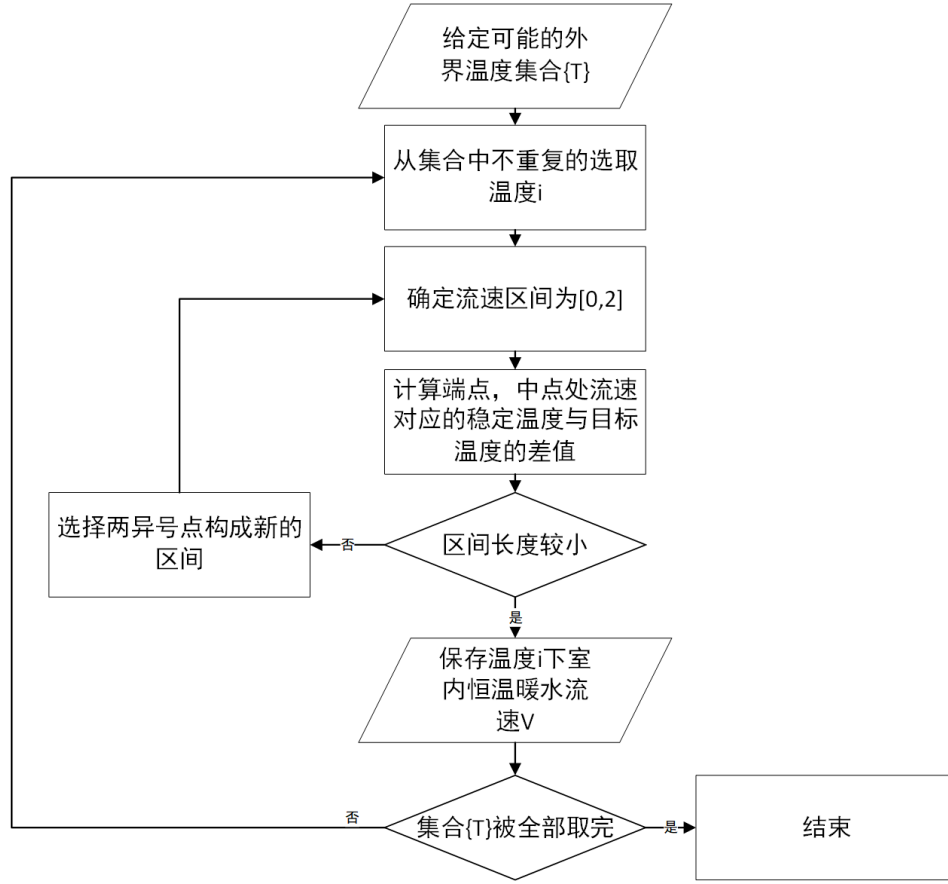


图 14: 二分查找维持流速流程图

随后我们对式 (32) 进行修改, 规定在温度差处于 $[-\sigma, \sigma]$ 之间时使用室外温度为 i 条件下的维持流速 V_i 进行自动控制, 而其余情况下利用式 (31) 中给出的自动控制模型进行控制。

$$v = \begin{cases} v(t), & T(x_0, y_0, z_0, t) < \eta_{aim} - \sigma_{aim} \\ V_i, & \eta_{aim} + \sigma_{aim} < T(x_0, y_0, z_0, t) < \eta_{aim} + \sigma_{aim} \\ v(t), & \eta_{aim} + \sigma_{aim} < T(x_0, y_0, z_0, t) \end{cases} \quad (33)$$

该模型兼顾了 PID 动态调整的特性与历史温度调控过程, 更全面的考虑先验知识。

5.3.2 改进温度控制模型的求解

我们利用二分法求得在不同室外温度下的维持流速 V_i ，如表（1）所示。

室外温度 $^{\circ}\text{C}$	维持流速 m^3/h	室外温度 $^{\circ}\text{C}$	维持流速 m^3/h
-5	0.2981	4	0.1579
-4	0.2655	5	0.1309
-3	0.2388	6	0.1238
-2	0.2165	7	0.1172
-1	0.1985	8	0.1115
0	0.1829	9	0.1060
2	0.1579	10	0.1012

随后通过利用该模型进行温度调节仿真，并且与初始调节模型进行比较。结果如下图所示：

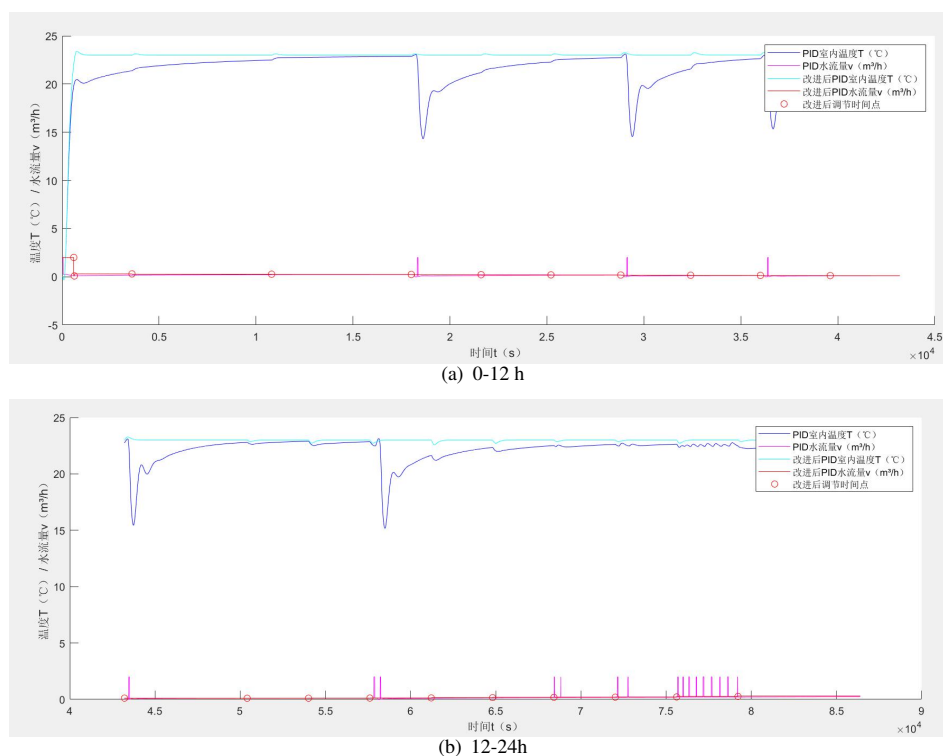


图 15: 修改前后的 PID 控制结果

可以直观看出来改进算法后温度控制更为稳定，对于温度降低时不易产生较大波动，且在稳定状态时更趋于目标温度。利用改进后的算法控制房间均温为 22.89°C 较之于原算法中的 21.8639°C 有很大提升，更接近于目标温度 23°C 。改进后的方差为 1.280，较之于改进前的方差减小了 27% 具有较大提升。可见结合历史数据，有助于更好的调节室内温度，适应外部温度变化。

六、敏感性分析

我们将附件二中的气温数据整体下调 1°C 随后使用改进后的温度控制模型进行敏感性分析，分析模型在原始气温和改变后气温的表现，得到以下结果：

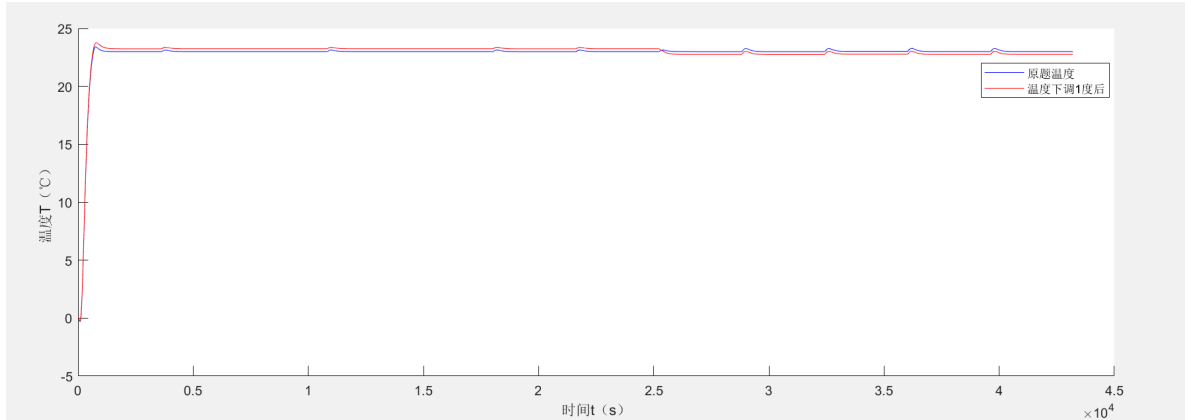


图 16: 改变温度前后的室内温度控制效果

可见温度降低，模型的控温效果基本没有影响，仍然保持较好的效果。在温度下降一度的情况下，室内平均温度只变化了 0.2°C ，具有较好的稳定性。

七、模型的评价

7.1 模型的优点

1. 利用傅里叶传热定律等基本规律建立物理模型，具有较高的可信度。
2. 求解房屋温度变化的过程中使用了有限体积元法，将微分方程的计算离散化，提高了计算效率。
3. 结合传统 PID 和先验知识进行温度调节，具有较好的调节效果。

7.2 模型的缺点

- 室内空气流动模型较为复杂，我们并未将其考虑周全，例如的某些复杂的热对流问题。
- 对于 PID 参数的调节限于时间和算力的要求未能得出满意的结果。

参考文献

- [1] 王补宣. 工程传热传质学下. 科学出版社, May 2015.
- [2] 王补宣. 工程传热传质学上. 科学出版社, May 2015.
- [3] 墙体.
- [4] 有限元素法, May 2022. Page Version ID: 71608413.
- [5] 晶金 and 琳郭. 新型 PID 参数选取方法在加热炉控制中的应用. 工业仪表与自动化装置, (1):92-93, 2010.
- [6] PID 控制器, May 2022. Page Version ID: 71658812.

附件

附件清单:

- get_heater.m

简化热水取暖器模型，输入外部薄层空气的平均温度，得到放热量的分布，取均值作为区域平均传热矢量

```
1 get_heater.m
2 %将加热器简化为一个输入三个参数，输出一个加热器的热流密度矢量的大小
3 %同时墙壁和玻璃也需要做此处理，求出热流密度矢量，提供给边界空气层
4 function q_heater = get_heater(T_air,v_water,T0_water)
5 %     T_air=0;
6 %     v_water=1;
7 %     T0_water=60;
8 %T_air 外部对应范围内空气边界薄层此时的均值温度
9 %v_water 水流量
10 %T0_water 水初始温度
11 %q_heater 热流密度矢量
12 %主要参数
13 C_w = 4200;           % 定义水的比热容
14 h_air = 30;           % 空气对流换热系数
15 k_iron = 12.5;        % 铁质取暖器的导热系数
16 rho_w = 1000;         % 定义水的密度
17 S = 0.0005;          % 散热水柱横截面积
18 L = 33;               % 取暖器等效水柱长度
19 dx = 0.01;           % 单位长度
20 %求出第一段的散热q，计算出下一个时刻的温度，计算出下一个时刻的散热
    q，以此递推
21 V_water = v_water/(S*3600);% 热水速度
22 T = zeros(1,L/dx);    % 存储热水微分段的温度
23 T(1) = T0_water;      % 第一分段水温是初始水温
24 q = zeros(1,L/dx);    % 存储热水微分段的热流密度
25 dt = dx/V_water;      % 流过一段单位长度所需要的时间
26 C = 4*(S^0.5);        % 界面圆周长
27 for i = 1:length(T)
28     q(i) = (h_air+k_iron) * (T(i) - T_air); %获得当前对象的热流密
        度
29     T(i+1) = T(i)-(q(i)*C*dx*dt)/(C_w*S*dx*rho_w); %获得下一个时间
        点的水温
30 end
```



```

31     q_heater = mean(q); %使用平均热流密度代表整体热流密度
32 end

```

- get_bound_mat.m

调用 get_heater.m, 求出六个边界面的热流密度矢量分布

```

1  %求解边界条件, 得到边界条件矩阵, 共6个 上 前 后
2  %边界空气薄层状态
3  %给出 三个面的q值分布矩阵
4  %长度微元 统一 0.05
5  function [q_top,q_front,q_back] = get_bound_mat(v_water,T0_water,Temp,
    T_out)
6      %参数
7      %    T_out = 0;
8      %    T0_water = 60;
9      %    v_water = 1;
10     %v_water 水流量
11     %T0_water 水初始温度
12     %x 长 10m
13     %y 宽 5m
14     %z 高 3m
15     %温度三维矩阵 Temp (x,y,z)
16     %室外温度 T_out
17     dl = 0.05;
18     h_air = 30;          % 空气对流换热系数
19     k_cement = 2.27;      % 水泥的导热系数
20     k_glass = 4.46;      % 玻璃的导热系数
21     X = 10/0.05 + 1;
22     Y = 5/0.05 + 1;
23     Z = 3/0.05 + 1;
24     %    Temp = zeros(X,Y,Z);
25     q_top = (h_air+k_cement)*(squeeze(Temp(:,:,end))-ones(X,Y)*T_out);
        %得到上边界热流密度矢量
26     win_martix = ones(Y,Z);
27     win_martix(1.5/dl+1:3.5/dl+1,1.25/dl+1:2.75/dl+1) = 0;
28     wall_martix = ones(Y,Z) - win_martix;
29     q_front = squeeze((h_air+k_cement)*(Temp(end,:,:)-ones(1,Y,Z)*
        T_out)).*win_martix + squeeze((h_air+k_glass)*(Temp(end,:,:)-
        ones(1,Y,Z)*T_out)).*wall_martix;
30     heater = zeros(Y,Z);

```

```

31     T_air = mean(Temp(1,1.5/dl+1:3.5/dl+1,0/dl+1:1/dl+1),'all'); %获取
        对应区域内的空气平均温度
32     heater(1.5/dl+1:3.5/dl+1,0/dl+1:1/dl+1) = get_heater(T_air,v_water
        ,T0_water);
33     q_back = -q_front + heater;
34 end

```

- main.m

解决第二类边界问题的三维傅里叶散热模型.

```

1 %房间温度求解主函数
2 %第二类边界条件下的有限元差分
3 clear;clc;
4 T_out = 0;           % 外部温度
5 T_in = 0;           % 房间内初始温度
6 T0_water = 60;       % 热水进入温度60度
7 v_water = 1;         % 热水每小时每立方米
8 dt = 0.01;          % 时间微元
9 time = 3600;         % 总仿真时间
10 a = 0.152;          % 定义散热系数
11 k_air = 30;          % 对流传热系数
12 C_air = 1005;        % 空气比热容
13 rho_air = 1.293;     % 空气密度
14 X = 10/0.05+1;
15 Y = 5/0.05+1;
16 Z = 3/0.05+1;
17 dx = 0.05;
18 dy = 0.05;
19 dz = 0.05;
20 Ax = (-2*eye(X)+diag(ones(1, X-1), 1)+diag(ones(1, X-1), -1)); % 定义
        X方向上的A矩阵
21 Ay = (-2*eye(Y)+diag(ones(1, Y-1), 1)+diag(ones(1, Y-1), -1)); % 定义
        Y方向上的A矩阵
22 Az = (-2*eye(Z)+diag(ones(1, Z-1), 1)+diag(ones(1, Z-1), -1)); % 定义
        Z方向上的A矩阵
23 Temp = ones(X,Y,Z)*T_in; %初始温度均匀分布
24 q_base = zeros(X,Y); %设置绝热条件
25 q_right = zeros(X,Z);
26 q_left = zeros(X,Z);
27 dTemp = zeros(201,101,61); %温度变换矩阵，需要结合最小单位时间才可以

```

```

28 %获取六个边界面的第二类边界条件
29 %有限元差分法
30 INDEX = 1;
31 for t = 1:time/dt
32     disp(t*dt)
33     %首先计算六个边界分布,八个拐角单独考虑
34     %对于内部空气热扩散
35     [q_top,q_front,q_back] = get_bound_mat(v_water,T0_water,Temp,T_out
        ); %获得边界热流密度矢量,每次都需要更新
36     dTempx = zeros(X,Y,Z);
37     dTempy = zeros(X,Y,Z);
38     dTempz = zeros(X,Y,Z);
39     for zzz = 1:Z %尝试求解三维差分矩阵
40         dTempx(:, :, zzz) = Ax*Temp(:, :, zzz);
41         dTempy(:, :, zzz) = Temp(:, :, zzz)*Ay;
42     end
43     for xxx = 1:X
44         dTempz(xxx, :, :) = squeeze(Temp(xxx, :, :))*Az;
45     end
46     dTemp = a^2*(dTempz+dTempy+dTempx)/(dx^2); %计算内部的空气导热情况
47     %对于六个边界面分析,一个方向有边界参与
48     dTemp(:, :, end) = ((k_air*dt*dx*dy*dz*(dTempx(:, :, end))/(dx^2))+
        (k_air*dt*dx*dy*dz*dTempy(:, :, end)/(dy^2))+((-q_top(:, :)+(-
        k_air*(Temp(:, :, end)-Temp(:, :, end-1)))/dz))*dt*dy*dx))/(C_air*
        rho_air*dt*dx*dy*dz); %上边界
49     dTemp(:, :, 1) = ((k_air*dt*dx*dy*dz*(dTempx(:, :, 1))/(dx^2))+
        (k_air*dt*dx*dy*dz*dTempy(:, :, 1)/(dy^2))+((q_base(:, :)-(-k_air*(Temp
        (:, :, 2)-Temp(:, :, 1))/dz))*dt*dy*dx))/(C_air*rho_air*dt*dx*dy*dz
        ); %下边界
50     dTemp(:, 1, :) = (squeeze(k_air*dt*dx*dy*dz*(dTempx(:, 1, :))/(dx^2))
        +((q_left(:, :)-(squeeze(-k_air*(Temp(:, 2, :)-Temp(:, 1, :))/dy)))*
        dt*dx*dz)+squeeze(k_air*dt*dx*dy*dz*dTempz(:, 1, :)/(dz^2)))/(
        C_air*rho_air*dt*dx*dy*dz); %左边界
51     dTemp(:, end, :) = (squeeze(k_air*dt*dx*dy*dz*(dTempx(:, end, :))/(dx
        ^2))+((-q_right(:, :)+squeeze(-k_air*(Temp(:, end, :)-Temp(:, end
        -1, :))/dy))*dt*dx*dz)+squeeze(k_air*dt*dx*dy*dz*dTempz(:, end, :)/
        (dz^2)))/(C_air*rho_air*dt*dx*dy*dz); %右边界
52     dTemp(end, :, :) = (((-q_front(:, :)+squeeze(-k_air*(Temp(end, :, :)-
        Temp(end-1, :, :))/dx))*dt*dy*dz)+squeeze(k_air*dt*dx*dy*dz*

```

```

    dTempy(end, :, :)/(dy^2))+squeeze(k_air*dt*dx*dy*dz*dTempz(end
    , :, :)/(dz^2)))/(C_air*rho_air*dt*dx*dy*dz); %前边界
53 dTemp(1, :, :) = (((q_back(:, :)-squeeze(-k_air*(Temp(2, :, :)-Temp
    (1, :, :))/dx))*dt*dy*dz)+squeeze(k_air*dt*dx*dy*dz*dTemp(1, :, :)/
    /(dy^2))+squeeze(k_air*dt*dx*dy*dz*dTempz(1, :, :)/(dz^2)))/(
    C_air*rho_air*dt*dx*dy*dz); %后边界
54 %对于十二条边，两个方向有边界参与
55
56 %    dTemp(1,1,2:Z-1) = (((q_back(1,2:Z-1)-(-k_air*(Temp(2,1,2:Z-1)-
    Temp(1,1,2:Z-1))/dx))*dt*dy*dz)+((q_left(1,2:Z-1)-(-k_air*(Temp
    (1,2,2:Z-1)-Temp(1,1,2:Z-1))/dy))*dt*dx*dz)+squeeze(k_air*dt*dx*dy*
    dz*(dTempz(1,1,2:Z-1))/(dz^2)))/(C_air*rho_air*dt*dx*dy*dz);
57 %    dTemp(1,end,2:Z-1) = (((q_back(end,2:Z-1)-(-k_air*(Temp(2,end,2:
    Z-1)-Temp(1,end,2:Z-1))/dx))*dt*dy*dz)+((-q_right(1,2:Z-1)+(-k_air
    *(Temp(1,end,2:Z-1)-Temp(1,end-1,2:Z-1))/dy))*dt*dx*dz)+squeeze(
    k_air*dt*dx*dy*dz*(dTempz(1,end,2:Z-1))/(dz^2)))/(C_air*rho_air*dt*
    dx*dy*dz);
58 %    dTemp(end,end,2:Z-1) = (((-q_front(end,2:Z-1)+(-k_air*(Temp(end,
    end,2:Z-1)-Temp(end-1,end,2:Z-1))/dx))*dt*dy*dz)+((-q_right(end,2:Z
    -1)+(-k_air*(Temp(end,end,2:Z-1)-Temp(end,end-1,2:Z-1))/dy))*dt*dx*
    dz)+squeeze(k_air*dt*dx*dy*dz*(dTempz(end,end,2:Z-1))/(dz^2)))/(
    C_air*rho_air*dt*dx*dy*dz);
59 %    dTemp(end,1,2:Z-1) = (((-q_front(1,2:Z-1)+(-k_air*(Temp(end,1,2:
    Z-1)-Temp(end-1,1,2:Z-1))/dx))*dt*dy*dz)+((q_left(end,2:Z-1)-(-
    k_air*(Temp(end,2,2:Z-1)-Temp(end,1,2:Z-1))/dy))*dt*dx*dz)+squeeze(
    k_air*dt*dx*dy*dz*(dTempz(end,1,2:Z-1))/(dz^2)))/(C_air*rho_air*dt*
    dx*dy*dz);
60 %
61 %
62 %    dTemp(2:X-1,1,end) = ((k_air*dt*dx*dy*dz*(dTempx(2:X-1,1,end))/(
    dx^2))+((q_left(2:X-1,end)-(-k_air*(Temp(2:X-1,2,end)-Temp(2:X-1,1,
    end))/dy))*dt*dx*dz)+((-q_top(2:X-1,1)+(-k_air*(Temp(2:X-1,1,end)-
    Temp(2:X-1,1,end-1))/dz))*dt*dy*dx))/(C_air*rho_air*dt*dx*dy*dz);
63 %    dTemp(2:X-1,end,end) = ((k_air*dt*dx*dy*dz*(dTempx(2:X-1,end,end)
    ))/(dx^2))+((-q_right(2:X-1,end)+(-k_air*(Temp(2:X-1,end,end)-Temp
    (2:X-1,end-1,end))/dy))*dt*dx*dz)+((-q_top(2:X-1,end)+(-k_air*(Temp
    (2:X-1,end,end)-Temp(2:X-1,end,end-1))/dz))*dt*dy*dx))/(C_air*
    rho_air*dt*dx*dy*dz);
64 %    dTemp(2:X-1,end,1) = ((k_air*dt*dx*dy*dz*(dTempx(2:X-1,end,1))/(

```

```

dx^2))+((-q_right(2:X-1,1)+(-k_air*(Temp(2:X-1,end,1)-Temp(2:X-1,
end-1,1))/dy))*dt*dx*dz)+((q_base(2:X-1,end)-(-k_air*(Temp(2:X-1,
end,2)-Temp(2:X-1,end,1))/dz))*dt*dy*dx))/(C_air*rho_air*dt*dx*dy*
dz);
65 %      dTemp(2:X-1,1,1) = (((k_air*dt*dx*dy*dz*(dTempx(2:X-1,1,1))/(dx
^2))+((q_left(2:X-1,1)-(-k_air*(Temp(2:X-1,2,1)-Temp(2:X-1,1,1))/dy
))*dt*dx*dz)+((q_base(2:X-1,1)-(-k_air*(Temp(2:X-1,1,2)-Temp(2:X
-1,1,1))/dz))*dt*dy*dx))/(C_air*rho_air*dt*dx*dy*dz);
66 %
67 %
68 %      dTemp(1,2:Y-1,end) = (((q_back(2:Y-1,end)-(-k_air*(Temp(2,2:Y-1,
end)-Temp(1,2:Y-1,end))/dx))*dt*dy*dz)+(k_air*dt*dx*dy*dz*(dTempy
(1,2:Y-1,end))/(dx^2))+((-q_top(1,2:Y-1)+(-k_air*(Temp(1,2:Y-1,end)
-Temp(1,2:Y-1,end-1))/dz))*dt*dy*dx))/(C_air*rho_air*dt*dx*dy*dz);
69 %      dTemp(1,2:Y-1,1) = (((q_back(2:Y-1,1)-(-k_air*(Temp(2,2:Y-1,1)-
Temp(1,2:Y-1,1))/dx))*dt*dy*dz)+(k_air*dt*dx*dy*dz*(dTempy(1,2:Y
-1,1))/(dx^2))+((q_base(1,2:Y-1)-(-k_air*(Temp(1,2:Y-1,2)-Temp(1,2:
Y-1,1))/dz))*dt*dy*dx))/(C_air*rho_air*dt*dx*dy*dz);
70 %      dTemp(end,2:Y-1,1) = (((-q_front(2:Y-1,1)+(-k_air*(Temp(end,2:Y
-1,1)-Temp(end-1,2:Y-1,1))/dx))*dt*dy*dz)+(k_air*dt*dx*dy*dz*(
dTempy(end,2:Y-1,1))/(dx^2))+((q_base(end,2:Y-1)-(-k_air*(Temp(end
,2:Y-1,2)-Temp(end,2:Y-1,1))/dz))*dt*dy*dx))/(C_air*rho_air*dt*dx*
dy*dz);
71 %      dTemp(end,2:Y-1,end) = (((-q_front(2:Y-1,end)+(-k_air*(Temp(end
,2:Y-1,end)-Temp(end-1,2:Y-1,end))/dx))*dt*dy*dz)+(k_air*dt*dx*dy*
dz*(dTempy(end,2:Y-1,end))/(dx^2))+((-q_top(end,2:Y-1)+(-k_air*(
Temp(end,2:Y-1,end)-Temp(end,2:Y-1,end-1))/dz))*dt*dy*dx))/(C_air*
rho_air*dt*dx*dy*dz);
72
73     for zz = 2:Z-1
74         dTemp(1,1,zz) = (((q_back(1,zz)-(-k_air*(Temp(2,1,zz)-Temp
(1,1,zz))/dx))*dt*dy*dz)+((q_left(1,zz)-(-k_air*(Temp(1,2,
zz)-Temp(1,1,zz))/dy))*dt*dx*dz)+(k_air*dt*dx*dy*dz*(Temp
(1,1,zz+1)-2*Temp(1,1,zz)-Temp(1,1,zz-1))/(dz^2)))/(C_air*
rho_air*dt*dx*dy*dz);
75         dTemp(1,end,zz) = (((q_back(end,zz)-(-k_air*(Temp(2,end,zz)-
Temp(1,end,zz))/dx))*dt*dy*dz)+((-q_right(1,zz)+(-k_air*(
Temp(1,end,zz)-Temp(1,end-1,zz))/dy))*dt*dx*dz)+(k_air*dt*
dx*dy*dz*(Temp(1,end,zz+1)-2*Temp(1,end,zz)-Temp(1,end,zz

```

```

76      -1))/(dz^2)))/(C_air*rho_air*dt*dx*dy*dz);
dTemp(end,end,zz) = (((-q_front(end,zz)+(-k_air*(Temp(end,end,
zz)-Temp(end-1,end,zz))/dx))*dt*dy*dz)+((-q_right(end,zz)
+(-k_air*(Temp(end,end,zz)-Temp(end,end-1,zz))/dy))*dt*dx*
dz)+(k_air*dt*dx*dy*dz*(Temp(end,end,zz+1)-2*Temp(end,end,
zz)-Temp(end,end,zz-1))/(dz^2)))/(C_air*rho_air*dt*dx*dy*dz
);
77 dTemp(end,1,zz) = (((-q_front(1,zz)+(-k_air*(Temp(end,1,zz)-
Temp(end-1,1,zz))/dx))*dt*dy*dz)+((q_left(end,zz)-(-k_air*(
Temp(end,2,zz)-Temp(end,1,zz))/dy))*dt*dx*dz)+(k_air*dt*dx*
dy*dz*(Temp(end,1,zz+1)-2*Temp(end,1,zz)-Temp(end,1,zz-1))
/(dz^2)))/(C_air*rho_air*dt*dx*dy*dz);
78 end
79 for xx = 2:X-1
80     dTemp(xx,1,end) = ((k_air*dt*dx*dy*dz*(Temp(xx+1,1,end)-2*Temp
(xx,1,end)-Temp(xx-1,1,end))/(dx^2))+((q_left(xx,end)-(-
k_air*(Temp(xx,2,end)-Temp(xx,1,end))/dy))*dt*dx*dz)+((-
q_top(xx,1)+(-k_air*(Temp(xx,1,end)-Temp(xx,1,end-1))/dz))*
dt*dy*dx))/(C_air*rho_air*dt*dx*dy*dz);
81 dTemp(xx,end,end) = ((k_air*dt*dx*dy*dz*(Temp(xx+1,end,end)-2*
Temp(xx,end,end)-Temp(xx-1,end,end))/(dx^2))+((-q_right(xx,
end)+(-k_air*(Temp(xx,end,end)-Temp(xx,end-1,end))/dy))*dt*
dx*dz)+((-q_top(xx,end)+(-k_air*(Temp(xx,end,end)-Temp(xx,
end,end-1))/dz))*dt*dy*dx))/(C_air*rho_air*dt*dx*dy*dz);
82 dTemp(xx,end,1) = ((k_air*dt*dx*dy*dz*(Temp(xx+1,end,1)-2*Temp
(xx,end,1)-Temp(xx-1,end,1))/(dx^2))+((-q_right(xx,1)+(-
k_air*(Temp(xx,end,1)-Temp(xx,end-1,1))/dy))*dt*dx*dz)+((
q_base(xx,end)-(-k_air*(Temp(xx,end,2)-Temp(xx,end,1))/dz))*
dt*dy*dx))/(C_air*rho_air*dt*dx*dy*dz);
83 dTemp(xx,1,1) = ((k_air*dt*dx*dy*dz*(Temp(xx+1,1,1)-2*Temp(xx
,1,1)-Temp(xx-1,1,1))/(dx^2))+((q_left(xx,1)-(-k_air*(Temp(
xx,2,1)-Temp(xx,1,1))/dy))*dt*dx*dz)+((q_base(xx,1)-(-k_air
*(Temp(xx,1,2)-Temp(xx,1,1))/dz))*dt*dy*dx))/(C_air*rho_air
*dt*dx*dy*dz);
84 end
85 for yy = 2:Y-1
86     dTemp(1,yy,end) = (((q_back(yy,end)-(-k_air*(Temp(2,yy,end)-
Temp(1,yy,end))/dx))*dt*dy*dz)+(k_air*dt*dx*dy*dz*(Temp(1,
yy+1,end)-2*Temp(1,yy,end)-Temp(1,yy-1,end))/(dx^2))+((-

```

```

      q_top(1,yy)+(-k_air*(Temp(1,yy,end)-Temp(1,yy,end-1))/dz))*
      dt*dy*dx))/(C_air*rho_air*dt*dx*dy*dz);
87 dTemp(1,yy,1) = (((q_back(yy,1)-(-k_air*(Temp(2,yy,1)-Temp(1,
      yy,1))/dx))*dt*dy*dz)+(k_air*dt*dx*dy*dz*(Temp(1,yy+1,1)-2*
      Temp(1,yy,1)-Temp(1,yy-1,1))/(dx^2))+((q_base(1,yy)-(-k_air
      *(Temp(1,yy,2)-Temp(1,yy,1))/dz))*dt*dy*dx))/(C_air*rho_air
      *dt*dx*dy*dz);
88 dTemp(end,yy,1) = (((-q_front(yy,1)+(-k_air*(Temp(end,yy,1)-
      Temp(end-1,yy,1))/dx))*dt*dy*dz)+(k_air*dt*dx*dy*dz*(Temp(
      end,yy+1,1)-2*Temp(end,yy,1)-Temp(end,yy-1,1))/(dx^2))+((
      q_base(end,yy)-(-k_air*(Temp(end,yy,2)-Temp(end,yy,1))/dz))
      *dt*dy*dx))/(C_air*rho_air*dt*dx*dy*dz);
89 dTemp(end,yy,end) = (((-q_front(yy,end)+(-k_air*(Temp(end,yy,
      end)-Temp(end-1,yy,end))/dx))*dt*dy*dz)+(k_air*dt*dx*dy*dz
      *(Temp(end,yy+1,end)-2*Temp(end,yy,end)-Temp(end,yy-1,end))
      /(dx^2))+((-q_top(end,yy)+(-k_air*(Temp(end,yy,end)-Temp(
      end,yy,end-1))/dz))*dt*dy*dx))/(C_air*rho_air*dt*dx*dy*dz);
90 end
91
92 %对于八个拐点，三个方向有边界参与
93 dTemp(1,1,1) = (((q_back(1,1)-(-k_air*(Temp(2,1,1)-Temp(1,1,1))/dx
      ))*dt*dy*dz)+((q_left(1,1)-(-k_air*(Temp(1,2,1)-Temp(1,1,1))/dy
      ))*dt*dx*dz)+((q_base(1,1)-(-k_air*(Temp(1,1,2)-Temp(1,1,1))/dz
      ))*dt*dy*dx))/(C_air*rho_air*dt*dx*dy*dz);
94 dTemp(1,1,end) = (((q_back(1,end)-(-k_air*(Temp(2,1,end)-Temp(1,1,
      end))/dx))*dt*dy*dz)+((q_left(1,end)-(-k_air*(Temp(1,2,end)-
      Temp(1,1,end))/dy))*dt*dx*dz)+((-q_top(1,1)+(-k_air*(Temp(1,1,
      end)-Temp(1,1,end-1))/dz))*dt*dy*dx))/(C_air*rho_air*dt*dx*dy*
      dz);
95 dTemp(1,end,end) = (((q_back(end,end)-(-k_air*(Temp(2,end,end)-
      Temp(1,end,end))/dx))*dt*dy*dz)+((-q_right(1,end)+(-k_air*(Temp
      (1,end,end)-Temp(1,end-1,end))/dy))*dt*dx*dz)+((-q_top(1,end)
      +(-k_air*(Temp(1,end,end)-Temp(1,end,end-1))/dz))*dt*dy*dx))/(
      C_air*rho_air*dt*dx*dy*dz);
96 dTemp(1,end,1) = (((q_back(end,1)-(-k_air*(Temp(2,end,1)-Temp(1,
      end,1))/dx))*dt*dy*dz)+((-q_right(1,1)+(-k_air*(Temp(1,end,1)-
      Temp(1,end-1,1))/dy))*dt*dx*dz)+((q_base(1,end)-(-k_air*(Temp
      (1,end,2)-Temp(1,end,1))/dz))*dt*dy*dx))/(C_air*rho_air*dt*dx*
      dy*dz);

```

```

97     dTemp(end,1,1) = (((-q_front(1,1)+(-k_air*(Temp(end,1,1)-Temp(end
    -1,1,1))/dx))*dt*dy*dz)+((q_left(end,1)-(-k_air*(Temp(end,2,1)-
    Temp(end,1,1))/dy))*dt*dx*dz)+((q_base(end,1)-(-k_air*(Temp(end
    ,1,2)-Temp(end,1,1))/dz))*dt*dy*dx))/(C_air*rho_air*dt*dx*dy*dz
    );
98     dTemp(end,1,end) = (((-q_front(1,end)+(-k_air*(Temp(end,1,end)-
    Temp(end-1,1,end))/dx))*dt*dy*dz)+((q_left(end,end)-(-k_air*(
    Temp(end,2,end)-Temp(end,1,end))/dy))*dt*dx*dz)+((-q_top(end,1)
    +(-k_air*(Temp(end,1,end)-Temp(end,1,end-1))/dz))*dt*dy*dx))/(
    C_air*rho_air*dt*dx*dy*dz);
99     dTemp(end,end,end) = (((-q_front(end,end)+(-k_air*(Temp(end,end,
    end)-Temp(end-1,end,end))/dx))*dt*dy*dz)+((-q_right(end,end)+(-
    k_air*(Temp(end,end,end)-Temp(end,end-1,end))/dy))*dt*dx*dz)
    +((-q_top(end,end)+(-k_air*(Temp(end,end,end)-Temp(end,end,end
    -1))/dz))*dt*dy*dx))/(C_air*rho_air*dt*dx*dy*dz);
100    dTemp(end,end,1) = (((-q_front(end,1)+(-k_air*(Temp(end,end,1)-
    Temp(end-1,end,1))/dx))*dt*dy*dz)+((-q_right(end,1)+(-k_air*(
    Temp(end,end,1)-Temp(end,end-1,1))/dy))*dt*dx*dz)+((q_base(end,
    end)-(-k_air*(Temp(end,end,2)-Temp(end,end,1))/dz))*dt*dy*dx))
    /(C_air*rho_air*dt*dx*dy*dz);
101    Temp = dTemp*dt + Temp; %更新温度
102    end

```

- PID.m

标注了 PID 参数以及函数。

```

1  function u = PID(e,Ie,de)
2  kp=-0.01;
3  ki=0.001;
4  kd=0;
5  u = kp*(e + ki*Ie + kd*de);
6
7  function [e,Ie,de] = PID_error(r,T,T_aim)
8  e = T(end) - r;
9  de = (T_aim(end) - r) - (T_aim(end-1) - r);
10 Ie = sum(T) - r*length(T);

```