

热传导方程两类参数的统计反演方法

胥文博

兰州交通大学数理学院, 甘肃 兰州

收稿日期: 2025年2月13日; 录用日期: 2025年3月6日; 发布日期: 2025年3月14日

摘要

本文以热传导方程为背景, 分别研究了不同位置上的常数型未知参数与函数型未知参数, 并给出统计反演算法, 完成数值反演试验。两类方法均能高效完成反演任务, 且在面对高噪声观测信息时, 具有很强的鲁棒性。

关键词

参数反演问题, 贝叶斯反演, 物理信息神经网络

Statistical Inversion Methods for Two Types of Parameters in the Heat Conduction Equation

Wenbo Xu

Department of Mathematics, Lanzhou Jiaotong University, Lanzhou Gansu

Received: Feb. 13th, 2025; accepted: Mar. 6th, 2025; published: Mar. 14th, 2025

Abstract

Focusing on the heat conduction equation, this paper investigates constant-type unknown parameters and function-type unknown parameters at different positions, and provides statistical inversion algorithms to complete numerical inversion experiments. Both methods can efficiently accomplish the inversion tasks and exhibit strong robustness in the presence of high-noise observational information.

Keywords

Parameter Inverse Problem, Bayesian Inversion Method, Physics-Informed Neural Networks

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

作为一类常见的描述实验现象与生产细节的数学物理方程，热传导方程被广泛研究并加以应用。其参数反演问题也因逆向研究视角的挑战性和高价值性，被学界广泛关注。本文将借用两类统计学中的数值方法，研究如下的两组方程的参数反演问题。

问题 P1：对于一个长为 L ，宽为 H 的二维薄板来说，其热传导方程具有如下形式：

$$\begin{cases} u_t - a(u_{xx} + u_{yy}) = 0, & (x, y, t) \in \Omega \times [0, 1], \\ u(x, y, 0) = \sin(\pi x)\sin(\pi y), & (x, y) \in \Omega, \\ u(0, y, t) = u(L, y, t) = 0, & y \in [0, H], t \in [0, 1], \\ u(x, 0, t) = u(x, H, t) = 0, & x \in [0, L], t \in [0, 1]. \end{cases} \quad (1)$$

对于方程(1)，其反问题是已知终端时刻，全局温度 $g = u(x, y, T)$ 的附加条件下，估计出合适的常扩散系数 a 。

问题 P2：该问题则聚焦于复杂边界条件下的一维导热杆热传递过程，其具有如下形式的表达：

$$\begin{cases} u_t - u_{xx} = e^{-t} \sin(\pi x/L), & (x, t) \in [0, L] \times [0, T], \\ u(x, 0) = x(1.75 - x), & x \in [0, L], \\ u(0, t) = 0, & t \in [0, T], \\ u_x(L, t) + \sigma(t) \cdot u(L, t) = 0, & t \in [0, T]. \end{cases} \quad (2)$$

该反问题将利用 x_0 处不同时刻的温度记录 $h = u(x_0, t)$ ，反演右边界上的热交换系数。

在反问题的发展过程中，形成了一些较为高效的参数反演方法。第一类是变分泛函为代表的优化解法，诸如 Landweber、共轭梯度方法等各种梯度型迭代算法。这类方法通过构建一个结合了未知参数和观测数据的代价泛函，并推导代价泛函的梯度表达式，然后选择适当的迭代算法来寻找泛函的极值，寻找到的局部或全局最优解作为参数的反演结果。例如，研究者 Yang 和 Deng 等人在文献[1]中，基于 Landweber 正则化迭代方案建立了一个最优控制框架，成功重建了热传导方程中的热源参数，不仅证实了该反问题解的全局唯一性和稳定性，也为后续研究提供了理论基础。此外，Wei 采用广义 Tikhonov 正则化的方法，处理了分数阶扩散方程中的时间依赖源项，并通过引入广义交叉验证策略来选择最优正则化参数，从而获取稳定的数值解[2]。

另一类重要方法是基于贝叶斯推断的统计抽样技术，该方法的一般步骤为推导未知参数的后验概率分布，再结合高效的统计抽样算法估计参数的值。Zabaras 等人便是利用这一方法，有效识别了一维及二维热传导方程 Neumann 边界条件下的热流密度 q ，同时证明了贝叶斯方法与最优控制方法在线性反问题求解中的等价性[3]。

近年来，各类基于数值驱动策略的反演方法也逐渐进入学者们的视野。吴自库利用最小二乘支持向量机(LS-SVM)解决热传导方程中的热源反演问题，将源项 $f(t)$ 的推断转变为一个二次优化问题，进而通过求解线性方程组获得最终答案[4]。Raissi 则提出了一种名为物理信息神经网络(Physics Informed Neural Networks, PINN)的新框架，它能够在小规模观测数据的条件下，高效地解决偏微分方程的正向和反向问题[5]。PINNs 不仅为数据驱动方法与物理模型之间的融合提供了一条新路径，而且在解决具体科学和工程问题上展现出了巨大潜力。文中分别求解了一维薛定谔方程与 2 维 N-S 方程组数值解，并重构了后者的流体密度和动力粘度参数。

本文将采取后两种统计反演方法，分别给出未知参数 a 和 $\sigma(t)$ 的数值反演结果。并验证不同观测误差下的算法稳定性。

2. 统计反演方法及其算法搭建

2.1. 正问题数值解法

在介绍参数反演问题前，还需构建精度高，计算速度快的数值计算方法，以便在后续的反演算法中频繁调用。本文将采用有限差分算法来完成该任务，其核心原理是将网格剖分节点处的差分算子近似为微分算子，以代替方程及其边界上的导数运算。

对于一维热传导方程，常用显格式、隐格式以及 C-N 差分三种差分格式，下面给出前两种格式的计算公式[6]：

① 显格式：

$$\frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t} - a(x_i, t_j) \cdot \left[\frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j)}{\Delta x^2} \right] = f(x_i, t_j)$$

② 隐格式：

$$\frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t} - a(x_i, t_j) \cdot \left[\frac{u(x_{i+1}, t_{j+1}) - 2u(x_i, t_{j+1}) + u(x_{i-1}, t_{j+1})}{\Delta x^2} \right] = f(x_i, t_{j+1})$$

可以看见前两种差分的主要区别便是空间项导数上对于时间节点选择不同，显格式是正向计算的，隐格式则是逆向求解线性方程组。至于 C-N 格式则是前两种的线性组合，其拥有更高的计算精度，这里将不再展示。

但对于二维方程组而言，上面两种格式不总是适用，会存在剖分要求高、非对角线方程组求解成本大等问题。因此迫切需要引入一种新的差分格式，来提高二维热传导方程的计算速度与精度。

Peaceman 和 Rachford 提出这样一种算法[6][7]，他们在计算高维方程时引入了一个中间层 $u_{i,j}^*$ ，将原本关于 $u_{i,j}^{n+1}$ 和 $u_{i,j}^n$ 的方程组，拆分成两步进行计算：首先，在 x 方向上使用隐格式，在 y 方向上则使用显格式，并求解 $u_{i,j}^n$ 和中间层 $u_{i,j}^*$ 构成的方程组，得到 $u_{i,j}^*$ ；第二步则交替隐格式的顺序，在 y 方向上使用隐格式，在 x 方向上使用显格式，之后求解 $u_{i,j}^{n+1}$ 关于 $u_{i,j}^*$ 的方程组，得到 $u_{i,j}^{n+1}$ 。以时间变量 t 为顺序，重复上述两个步骤，直至遍历完所有时间剖分节点，得到该高维方程组的数值解。下面给出上述两个步骤的计算过程：

第一步： $\frac{u_{i,j}^* - u_{i,j}^n}{\Delta t/2} = a_{i,j} \cdot \frac{1}{h^2} (u_{i+1,j}^* - 2u_{i,j}^* + u_{i-1,j}^* + u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n)$ ，此处令 $\Delta x = \Delta y = h$ ，经化简有，

$$\begin{aligned}
& \left[\begin{array}{ccc} 1+a_{2,j}r & -a_{2,j}r/2 & \\ -a_{3,j}r/2 & 1+a_{3,j}r & -a_{3,j}r/2 \\ & \ddots & \\ & -a_{Nx-2,j}r/2 & 1+a_{Nx-2,j}r & -a_{Nx-2,j}r/2 \\ & & -a_{Nx-1,j}r/2 & 1+a_{Nx-1,j}r \end{array} \right] \times \begin{bmatrix} u_{2,j}^* \\ u_{3,j}^* \\ \vdots \\ u_{Nx-2,j}^* \\ u_{Nx-1,j}^* \\ u^* \end{bmatrix} \\
& = \begin{bmatrix} u_{2,j}^n + \frac{a_{2,j} \cdot r}{2} (u_{2,j+1}^n - 2u_{2,j}^n + u_{2,j-1}^n) \\ u_{3,j}^n + \frac{a_{3,j} \cdot r}{2} (u_{3,j+1}^n - 2u_{3,j}^n + u_{3,j-1}^n) \\ \vdots \\ u_{Nx-2,j}^n + \frac{a_{Nx-2,j} \cdot r}{2} (u_{Nx-2,j+1}^n - 2u_{Nx-2,j}^n + u_{Nx-2,j-1}^n) \\ u_{Nx-1,j}^n + \frac{a_{Nx-1,j} \cdot r}{2} (u_{Nx-1,j+1}^n - 2u_{Nx-1,j}^n + u_{Nx-1,j-1}^n) \\ b_n \end{bmatrix} \tag{3}
\end{aligned}$$

记等号左边分别为矩阵 A 和 u^* , 右边为 b_n , 通过求解 $Au^* = b^n$, 得到中间虚拟层的 u^* 。

第二步: $\frac{u_{i,j}^{n+1} - u_{i,j}^*}{\Delta t / 2} = a \cdot \frac{1}{h^2} (u_{i+1,j}^* - 2u_{i,j}^* + u_{i-1,j}^* + u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1})$, 经化简,

$$\begin{aligned}
& \left[\begin{array}{ccc} 1+a_{i,2}r & -a_{i,2}r/2 & \\ -a_{i,3}r/2 & 1+a_{i,3}r & -a_{i,3}r/2 \\ & \ddots & \\ & -a_{i,Ny-2}r/2 & 1+a_{i,Ny-2}r & -a_{i,Ny-2}r/2 \\ & & -a_{i,Ny-1}r/2 & 1+a_{i,Ny-1}r \end{array} \right] \times \begin{bmatrix} u_{i,2}^{n+1} \\ u_{i,3}^{n+1} \\ \vdots \\ u_{i,Ny-2}^{n+1} \\ u_{i,Ny-1}^{n+1} \\ u^{n+1} \end{bmatrix} \\
& = \begin{bmatrix} u_{i,2}^* + \frac{a_{i,2} \cdot r}{2} (u_{i+1,2}^* - 2u_{i,2}^* + u_{i-1,2}^*) \\ u_{i,3}^* + \frac{a_{i,3} \cdot r}{2} (u_{i+1,3}^* - 2u_{i,3}^* + u_{i-1,3}^*) \\ \vdots \\ u_{i,Ny-2}^* + \frac{a_{i,Ny-2} \cdot r}{2} (u_{i+1,Ny-2}^* - 2u_{i,Ny-2}^* + u_{i-1,Ny-2}^*) \\ u_{i,Ny-1}^* + \frac{a_{i,Ny-1} \cdot r}{2} (u_{i+1,Ny-1}^* - 2u_{i,Ny-1}^* + u_{i-1,Ny-1}^*) \\ b^* \end{bmatrix} \tag{4}
\end{aligned}$$

2.2. 贝叶斯反演方法

贝叶斯反演的核心原理是在现实观测下, 对未知参数的后验概率密度进行估计, 并采用一些统计抽样技术, 将该后验密度转化为参数的实际取值[8]。其思想可以利用如下的公式概括。对于偏微分方程参数反演模型:

$$Y = f(X, E) \quad (5)$$

其中, X 为待估参数, E 为观测误差, Y 为观测值。这里并没有限制方程和参数的形式。该框架为统计反问题中更一般的, 普适性更强的反演框架, 能有效面对线性与非线性的参数反演问题。

$$p(x|y) = \frac{p(x) \cdot p(y|x)}{p(y)} \propto p(x) \cdot p(y|x) \quad (6)$$

式(6)通过贝叶斯定理, 利用参数的先验信息 $p(x)$ 和样本观测 $p(y|x)$ 的似然信息, 得到参数关于观测数据的后验分布信息。在得到参数后验概率 $p(x|y)$ 的稳定分布后, 可使用下面这两种方式表达反演结果。

① 最大后验估计: $x_{\text{MAP}} = \arg \max_{x \in X} p(x|y)$ 。

② 后验均值估计: $x_{\text{CM}} = E(x|y) = \int_{\Omega} x \cdot p(x|y) dx$ 。

对于最大后验估计, 反问题的求解被同样转换成一个优化问题, 可以通过极大似然或者拉格朗日方法求解。此外, 一些基于马尔科夫链的抽样估计方法, 例如 Gibbs 和 Metropolis-Hastings, 在这里同样适用。本文也将采取这种方法进行参数反演的可视化表达。

在贝叶斯反演框架中, 先验分布的选择至关重要, 它反映着研究者对于真实参数的预期分布。对于预期附近的参数, 常赋予较高的概率取值; 反之对于偏离预期较远的参数, 则赋予较低的概率密度。因此将不同种类的参数先验融入反演模型, 其参数反演结果也有所差异, 或是收敛速度的快慢, 又或者是收敛位置的偏差。

常见的先验分布有高斯分布、均匀分布、Markov 随机场、脉冲先验和共轭先验等分布[8], 它们的适用范围各不相同, 但均能反映出前文所提及的“远小近大”先验选取理念。高斯分布易于构造、通用性高, 因此适合绝大多数参数反演场景; 均匀分布相对更适合已知参数大致范围的情形, 在后验分布统计抽样中可以减少运算量; 后几种先验信息则被广泛应用在图像重建领域。

不论选择何种先验信息, 它们都是对已有经验的总结, 本文也将在之前各类相关问题的研究基础上, 给定先验细节。问题 P1 中的单参数反演问题, 在真参范围不明了的前提下, 选择高斯分布是较为稳妥的, 在文献[9] [10]中高斯先验的选取也保证了参数反演结果。本文将定义如下的先验信息:

$$p_{\text{prior}}(a) = \frac{1}{\sqrt{2\pi \cdot \sigma_p^2}} \cdot \exp \left\{ -\lambda \frac{(a - \bar{a})^2}{2\sigma_p^2} \right\} \quad (7)$$

其中 \bar{a} 和 σ_p^2 分别为算法的先验均值与先验方差。在已有研究工作的基础上, 经过多次数值实验后发现, 较小的先验均值和方差即可使待估参数迅速收敛到真实位置, 且后续似然函数对已有观测的拟合情况也更佳。因此, 后续实验中 \bar{a} 和 σ_p^2 分别取 0.5 和 0.1²。 λ 为正则化参数, 引入它可提升算法稳定性。

在推导 a 的似然函数之前, 还需假定模型噪声 ε 为符合正态分布 $N(0, \sigma_{\varepsilon}^2)$ 的加性噪声, 这个噪声是由微分方程数值求解误差与观测误差共同造成的, 此时附加的观测条件为 $g^{\delta}(x, y) = u(x, y, T; a) + \varepsilon$ 。则似然密度为,

$$p_{\text{like}}(g|a) = \frac{1}{(2\pi\sigma_{\varepsilon})^n} \cdot \exp \left\{ -\frac{\left(u^a \Big|_{t=1} - g \right)^T \left(u^a \Big|_{t=1} - g \right)}{2\sigma_{\varepsilon}^2} \right\} \quad (8)$$

式(8)中 n 的为样本数量, u^a 为数值解 $u_N(x, y, t; a)$ 在 $t=1$ 时的温度, 之后利用式(6), 有,

$$\begin{aligned}
 p_{\text{post}}(a|g) &= \frac{p_{\text{prior}}(a) \cdot p_{\text{like}}(g|a)}{p(g)} \\
 &\propto \exp \left\{ -\lambda \frac{(a - \bar{a})^2}{2\sigma_p^2} \right\} \cdot \exp \left\{ -\frac{\left(u^a \Big|_{t=1} - g \right)^T \left(u^a \Big|_{t=1} - g \right)}{2\sigma_\varepsilon^2} \right\}
 \end{aligned} \tag{9}$$

此外，上式中的后验表达采用了正比的形式，这是考虑到相比起较大的系数项 $(2\pi\sigma_\varepsilon)^{-\frac{n}{2}}$ ，在各类统计抽样技术里，指数函数中的均值与方差是更被关注的。图 1 绘制了两种误差下的后验密度，其均值集中在 0.1 附近，且越小的加性噪声对应越紧凑的密度函数。

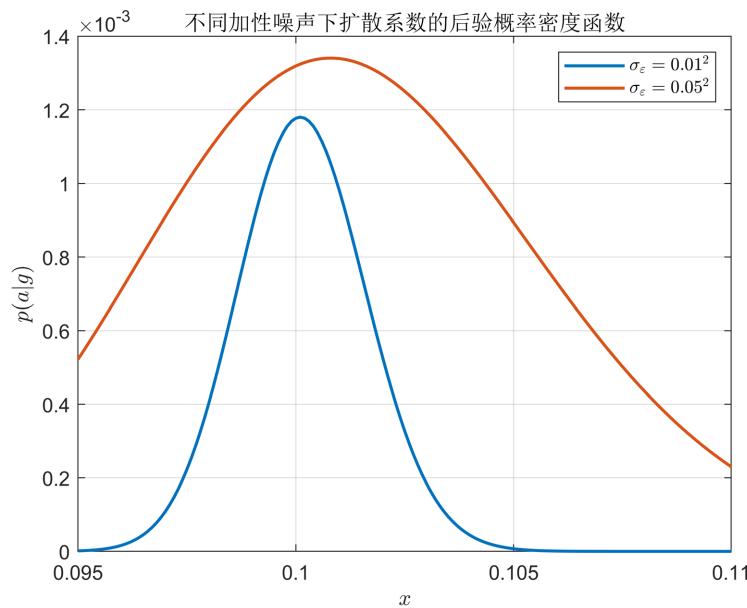


Figure 1. Posterior probability density function of diffusion coefficient for inversion

图 1. 扩散系数的后验概率密度函数

针对问题 P2，此时待估参数从常系数变化为了时间依赖的函数型参数，此时仅依靠贝叶斯定理与统计抽样技术，将很难反演出符合要求的热交换系数。于是文章借用径向基函数拟合的思想，将函数 $\sigma(t)$ 的推断转变为权重系数 ω_i 和基函数 ϕ_i 的推断，即

$$\sigma(t) = \sum_{i=1}^d \omega_i \cdot \phi_i$$

d 为基函数数量。且该拟合的基函数往往是固定的，一般为 Gauss 核或者多项式核，只需选择合适的系数即可。在上述技巧的支持下，P2 的反演目标变成了一组系数 $\omega = (\omega_1, \omega_2, \dots, \omega_d)$ 的估计。对于多参数的 Bayes 反演而言，其整体流程仍与 P1 中的单参数反演一致，只不过先验与后验密度变换为了高维的联合密度。在定义这些密度之前，还需提出系数 ω_i 的独立性假设，则整体的先验可拆分成各分量 ω_i 的先验乘积。需要注意的是，这里选择高斯先验将不再合适，因为多个独立分量构造出的 d 维高斯联合密度函数取值往往较小，是由多个小概率密度连乘取得，在后续的 M-H 抽样中，会导致接受概率极小，Markov 链较难更新，参数反演结果无法收敛。且探索 d 个不同分量的先验均值与方差的超参数组合，难度太高，很难找到合适的取值。又因为在较小区间上的径向基函数拟合，不同基函数对应的权重系数不会太大，

故不妨对称性地假定 ω_i 上下界，将参数的预期取值限定在范围内。这样可大幅基提升算法反演速率，因此取 ω_i 的分布为 $[-r, r]$ 上的均匀分布，则联合先验为，

$$p_{\text{prior}}(\omega) = (2r)^{-d} \cdot \lambda \|\omega\|_2^2 \quad (10)$$

h 的似然概率密度仍采用高斯分布，即，

$$p_{\text{like}}(h|\omega) = (2\pi\sigma_e)^{-n} \cdot \exp \left\{ -\frac{\left(u^\omega \Big|_{x=x_0} - h \right)^T \left(u^\omega \Big|_{x=x_0} - h \right)}{2\sigma_e^2} \right\} \quad (11)$$

则其联合后验概率密度函数可被表述为，

$$p_{\text{post}}(\omega|h) = c_\lambda \cdot \exp \left\{ -\frac{\left(u^\omega \Big|_{x=x_0} - h \right)^T \left(u^\omega \Big|_{x=x_0} - h \right)}{2\sigma_e^2} \right\} \quad (12)$$

其中 c_λ 为一包含正则算子和先验信息的常数。

最后给出该框架下，两种参数均能使用的 Metropolis-Hastings 反演算法，详细步骤见表 1。

Table 1. M-H inversion algorithm for heat conduction equation
表 1. 热传导方程 M-H 反演算法

Step1. 根据建议分布 $q(a^*|a_k)$ 生成随机数 a^* ，并计算随机数 a^* 与参数当前状态 a_k 的建议分布函数值 $q(a^*|a_k)$ 和 $q(a_k|a^*)$ ，以及目标分布函数值 $p(a^*)$ 和 $p(a_k)$ ，假设 a_0 为一固定常数，在算法开始前给定。上述密度函数有如下定义，

$$q(a^*|a_k) = \frac{1}{\sqrt{2\pi \cdot \sigma_k}} \cdot \exp \left\{ -\frac{(a^* - a_k)^2}{2\sigma_k^2} \right\}, \quad \sigma_k = 0.05a_k; \\ p(a^*) = p_{\text{post}}(a^*|g), \quad p(a_k) = p_{\text{post}}(a_k|g).$$

Step2. 计算接受概率

$$A(a^*, a_k) = \min \left\{ 1, \frac{p(a^*) \cdot q(a_k|a^*)}{p(a_k) \cdot q(a^*|a_k)} \right\};$$

Step3. 生成[0,1]上的均匀分布随机数 r ；

Step4. 如果有 $r \leq A(a^*, a_k)$ ，则令 Markov 链上的下一个状态 a_{k+1} 为 a^* ，反之则保持上一个的状态不变，即 $a_{k+1} = a_k$ ；

Step5. 令 $k = k + 1$ ，重复上述步骤，直至收集到足够长的 Markov 链，此时算法达到最大迭代次数 N_m 。

问题 P1 和 P2 的目标分布分别为后验分布 $p(a|g)$ 与 $p(\omega|h)$ ，建议分布则与各自先验保持一致，具体参数将在下一章数值实验中给出。

2.3. PINN 反演方法

几乎所有的机器学习问题都可以被看做一个优化过程，即定义标签数据与预测数据间合适的误差，并在参数的可行集内使损失达到最小。故本节基于数值驱动的 PINN 反演方法与最优控制框架和 Bayes 框架相似，均需给出误差最小化的表达。这里仿照前两种框架，定义如下平方项损失，

$$\mathcal{L}_{data} = \frac{1}{M} \sum_{j=1}^M \left(u_N(x_0, y_0, t_j) - u_{obs}(x_0, y_0, t_j) \right)^2 \quad (13)$$

其中 u_N 为网络计算出的对应位置的数值解，为附加的观测信息。其形式与 2.2 节中观测信息的似然密度函数，以及最优控制框架中参数在 $L^2(\Omega)$ 空间上的范数定义类似，但不同的是 PINN 并不需要反复计算正问题，而是将数值解和未知参数看做一个整体，同时给出反演结果。因此还需要正问题中涉及到的定解条件包括控制方程本身，和 \mathcal{L}_{data} 一起共同限制 PINN 网络的损失，并使用自动微分技术，替代方程中的各项微分算子。则网络总损失 \mathcal{L}_{total} 为，

$$\mathcal{L}_{total} = \omega_e \cdot \mathcal{L}_{equation} + \omega_i \cdot \mathcal{L}_{initial} + \omega_b \cdot \mathcal{L}_{bound} + \omega_d \cdot \mathcal{L}_{data} \quad (14)$$

其中前三项损失分别为控制方程损失、初值损失以及边界损失，最后一项为观测损失，这里采用的线性加权汇总总损失，各损失前的系数作为网络的超参数，可在实验中调整。

接下来给出问题 P1 各项损失的具体定义，

$$\begin{aligned} \text{MSE}_{\mathbf{P1}} &= \omega_b \cdot \text{MSE}_B + \omega_i \cdot \text{MSE}_I + \omega_f \cdot \text{MSE}_f + \omega_d \cdot \text{MSE}_d \\ &= \frac{2\omega_b}{N_x} \sum_{n=1}^M \sum_{j=1}^{N_y} \left(u_N(x, y_j, t_n) \Big|_{x=0, L} \right)^2 + \frac{2\omega_b}{N_y} \sum_{n=1}^M \sum_{i=1}^{N_x} \left(u_N(x_i, y, t_n) \Big|_{y=0, H} \right)^2 \\ &\quad + \frac{\omega_i}{N_x \cdot N_y} \sum_{j=1}^{N_y} \sum_{i=1}^{N_x} \left(u_N(x_i, y_j, 0) - \sin(\pi x_i) \sin(\pi y_j) \right)^2 \\ &\quad + \frac{\omega_f}{N_f} \sum_{i=1}^{N_f} \left(u_t^N(x_i, y_i, t_i) - a(u_{xx}^N(x_i, y_i, t_i) + u_{yy}^N(x_i, y_i, t_i)) \right)^2 \\ &\quad + \frac{\omega_d}{N_x \cdot N_y} \sum_{j=1}^{N_y} \sum_{i=1}^{N_x} \left(u_N(x_i, y_j, 1) - g(x_i, y_j) \right)^2 \end{aligned} \quad (15)$$

在给定损失之后，仍需给出具体的网络结构作为该反演任务的模型。对于问题 P1 中较为简单的输入特征(一些随机离散点)而言，在奥卡姆剃刀原理的支持下，使用灵活性高且计算效率更快的前馈型全连接网络具有更大的优势。此外，在以往学者的研究基础上发现[5] [11] [12]，得益于 PINN 中嵌入的物理信息，仅需少量的隐藏层即可达成反演目标。待估参数 a 则可作为神经网络的固定参数，同神经元的权值与偏置一起训练。

激活函数在训练中将直接影响网络的输出结果和参数更新梯度，如若选择不当，容易造成梯度消失或者爆炸的现象。不同的激活函数，其值域分布，梯度特性略有区别，需要根据输出需求来具体选择。在偏微分方程数值计算领域中，文献[13] [14]引入了一些自适应激活函数，以提升 PINN 方法的鲁棒性；文献[12]则从数值实验的角度给出 Tanh、Sigmoid、ISRU 和 Soft Plus 四类函数的反演表现，其中 Tanh 函数表现最佳。其 $[-1, 1]$ 的值域分布，也不会使后续神经元发生输入偏移或输出为 0 的情况，进而降低收敛速度与精度，因此本文将选择 Tanh 函数作为激活函数。

随机梯度下降方法，作为最常见的参数训练算法，其实现简单，计算效率高，但其固定的学习率使得整体训练速度较慢，且会出现陷入局部最优的情况。因此在后续的研究中，学者们陆续提出一些学习率自适应调整算法以及梯度动量加速算法，来避免上述情况。Adam 则结合了两种参数优化的优点，其训练时间更短，且训练结果的鲁棒性更强，是一种常见优化方法。本文的 PINN 反演算法也将采取该优化方式。

对于问题 P2，将定义如下损失，

$$\begin{aligned}
\text{MSE}_{P2} &= \omega_b \cdot \text{MSE}_B + \omega_i \cdot \text{MSE}_I + \omega_f \cdot \text{MSE}_f + \omega_d \cdot \text{MSE}_d \\
&= \frac{2\omega_b}{N_0} \sum_{n=1}^{N_0} (u_N(0, t_n))^2 + \frac{2\omega_b}{N_1} \sum_{n=1}^{N_1} (u_x^N(1, t_n) + \sigma_N(t_n) \cdot u_N(1, t_n))^2 \\
&\quad + \frac{\omega_i}{N_i} \sum_{i=1}^{N_i} (u_N(x_i, 0) - \sin(\pi x_i / L))^2 \\
&\quad + \frac{\omega_f}{N_f} \sum_{i=1}^{N_f} (u_t^N(x_i, t_i) - u_{xx}^N(x_i, t_i) - e^{-t_i} \sin(\pi x_i / L))^2 \\
&\quad + \frac{\omega_d}{N_{x0}} \sum_{n=1}^{N_{x0}} (u_N(x_0, t_n) - h(t_n))^2
\end{aligned} \tag{16}$$

注意到，这里面对着 2.2 节中相同的问题，即如何表达函数型参数 $\sigma(t)$ 。在上一节中，选取了基函数模拟等价替换的思路，这里理论上也可使用该方法，将 $\sigma(t)$ 的反演转换为一组 d 维系数的反演。除此之外，PINN 这类深度网络具其余反演理论不具备的优势，即其高度灵活的模型结构，使它在面对复杂任务或者特殊输出需求时，仅调整网络结构即可。这里仅需在最终的输出层上在添加一维输出作为 $\sigma(t)$ 的反演结果，且该维度的输出仍然与(16)中的损失保持一致，只不过要限制它的输入仅包含时间变量，故该问题对应的网络结构被修改为图 2 中的形式。

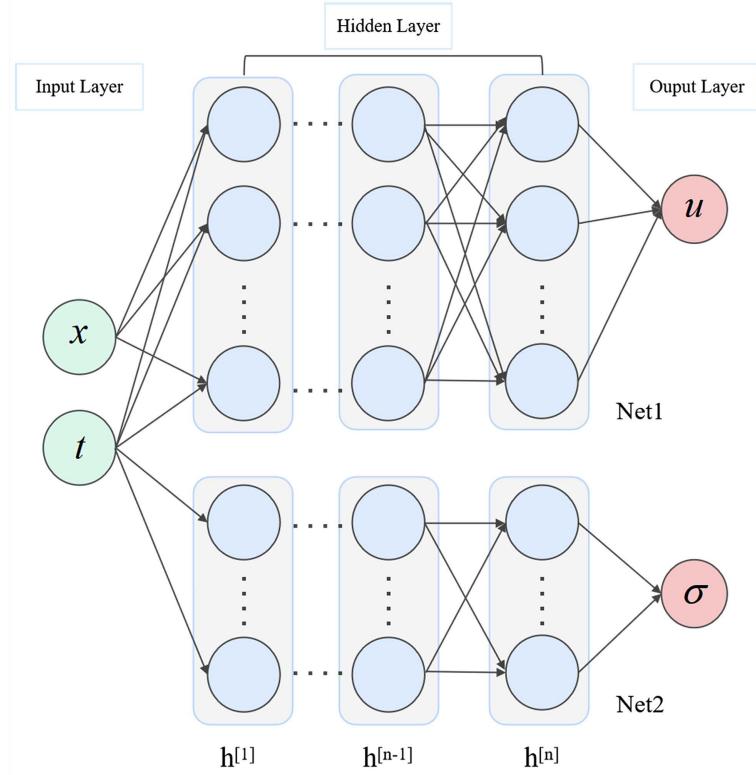


Figure 2. PINN network structure corresponding to problem P2
图 2. 问题 P2 对应的 PINN 网络结构

问题 P2 的激活函数与参数优化算法将与 P1 保持一致，对于学习率、隐藏层数、神经元个数等超参数的选取，将在文献[11] [12]的基础上，结合多次实验结果给出反演表现相对最佳的取值。

至此已经搭建好两类统计反演算法，下一章将给出这两种方法的数值反演表现。

3. 数值反演实验

针对问题 P1，设定扩散系数 a 的真实取值为 0.1，先验均值与方程分别为 0.5 和 0.1，算法初值设定为 0.5，在第三章中，已经给出了两种观测误差下的后验均值分布范围，图 3 给出 M-H 抽样的可视化进程。

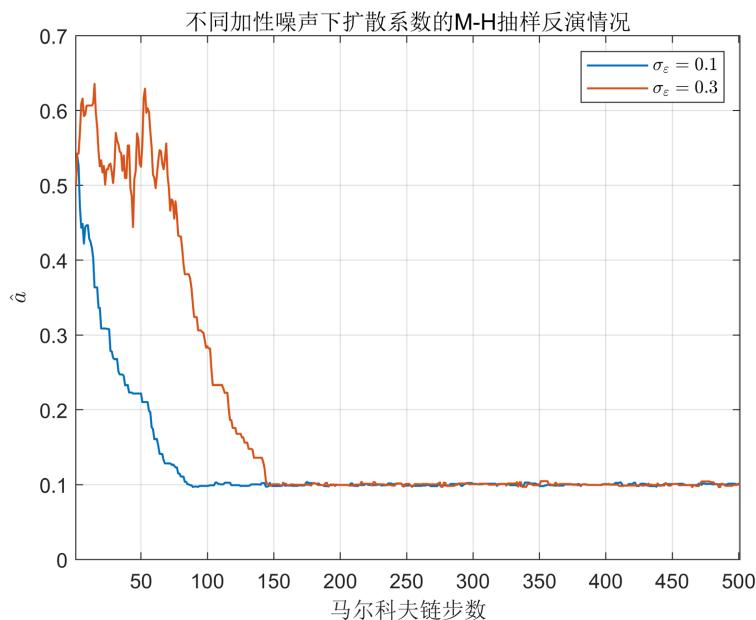


Figure 3. Parameter update process of M-H inversion algorithm for problem P1
图 3. 问题 P1 的 M-H 反演算法参数更新过程

其反演结果与相对精度见表 2。

Table 2. M-H algorithm inversion results for problem P1

表 2. 问题 P1 的 M-H 算法反演结果

观测误差	反演结果	相对精度	首次平稳步长
$\sigma_\varepsilon = 0.01$	0.1001	0.10%	85
$\sigma_\varepsilon = 0.05$	0.1003	0.30%	145

问题 P1 对应的 PINN 算法超参数取值情况如下，前馈网络具有 5 层隐藏层，每层 32 个神经元，dropout 取 0.05，训练时每个批次均重新随机抽样，初值与四条边界上各选取 2000 个采样点，控制方程选取 10,000 个采样点，为保证算法收敛，需提高观测位置采样点数量，这里取 5000 个，学习率取 10^{-3} ，激活函数为 \tanh 。其反演结果见表 3。

Table 3. PINN algorithm inversion results for problem P1

表 3. 问题 P1 的 PINN 算法反演结果

观测误差	反演结果	相对精度	收敛所需批次
$\delta = 0$	0.1001	0.10%	7042
$\delta = 3\%$	0.1010	1.00%	15,269
$\delta = 10\%$	0.1007	0.70%	9704

对于问题 P2 中的热交换系数 $\sigma(t)$ ，这里取真值为 $4\sin(\pi t)/3 + \cos(\pi t)/3$ ，若使用径向基函数模拟，当 t 被限制在较小区间时， d 取 3 便可满足模拟需求，M-H 反演算法中的先验均匀分布为 $[-10, 10]$ ，算法对初值不敏感，任意较小的初值在经历多次迭代后，皆可使结果收敛在真实值附近。建议分布的抽样变化步长为 1，是先验范围的 5%。反演结果如图 4 所示。

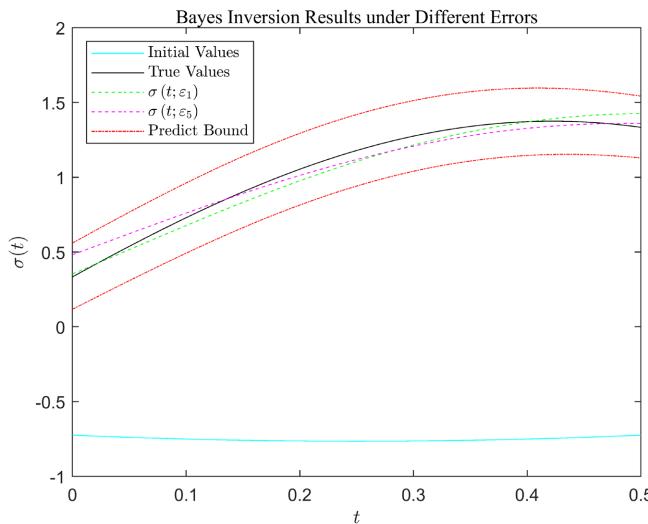


Figure 4. M-H algorithm inversion results for problem P2

图 4. 问题 P2 的 M-H 算法反演结果

其中红色的虚线为真实 $\sigma(t)$ 对应的基函数权值， $\omega_{true} = (-5.4462, 8.5574, -2.9136)$ 上下波动 1% 后，绘制出来的。两种误差下的反演结果，均落在预测区间内。

对于问题 P2 的变体 PINN，隐层设定为 6 层，每层 32 个神经元，为加快收敛进度，学习率取 0.005，dropout 则适当调大成 0.1。又因该问题的反演目标在右边界上，因此取右边界采样点为 5000，左边界与初值设为 2000 个，观测采样点仍与 P1 保持一致。其反演效果见图 5。

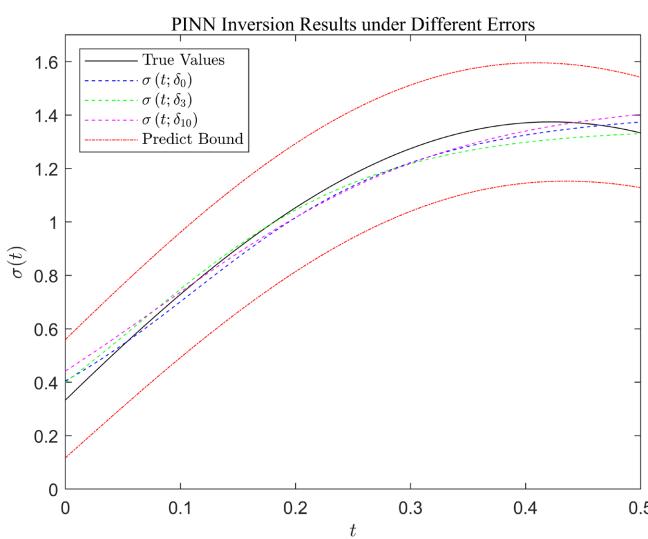


Figure 5. Inversion results of variant PINN algorithm for problem P2

图 5. 问题 P2 的变体 PINN 算法反演结果

Table 4. Inversion errors of two algorithms for problem P2**表 4.** 问题 P2 的两类算法的反演误差

观测误差	反演结果	相对精度
M-H 抽样反演		
$\sigma_\varepsilon = 0.01$	0.0030	4.55%
$\sigma_\varepsilon = 0.05$	0.0034	4.62%
变体 PINN 反演		
$\delta = 0$	0.0016	3.42%
$\delta = 3\%$	0.0021	3.63%
$\delta = 10\%$	0.0019	3.62%

在文章的最后，表 4 还给出了问题 P2 中两类方法的均方误差与相对误差表现，整体看来两种方法均能较好地反演出参数实际情况，在高噪声的情况下，反演表现也会优于一般的最优控制框架。横向对比两类方法，后者基于变体 PINN 在精度上更为出色，但训练时长会略久于 M-H 抽样反演。但这种情况在高维空间中却有所反转。

4. 总结

本文分别研究了热方程不同位置上两类参数的反演问题，给出两类统计反演算法及其反演表现，相比传统最优控制框架反演，这两种算法不需频繁地推导伴随方程与泛函梯度信息，均能高效地完成未知参数的估计。此外，这两种方法的鲁棒性更强，在高噪情形下的反演结果相对光滑[15]。

参考文献

- [1] Yang, L., Deng, Z., Yu, J. and Luo, G. (2009) Optimization Method for the Inverse Problem of Reconstructing the Source Term in a Parabolic Equation. *Mathematics and Computers in Simulation*, **80**, 314-326. <https://doi.org/10.1016/j.matcom.2009.06.031>
- [2] Wei, T. and Zhang, Z.Q. (2013) Reconstruction of a Time-Dependent Source Term in a Time-Fractional Diffusion Equation. *Engineering Analysis with Boundary Elements*, **37**, 23-31. <https://doi.org/10.1016/j.enganabound.2012.08.003>
- [3] Wang, J. and Zabaras, N. (2004) A Bayesian Inference Approach to the Inverse Heat Conduction Problem. *International Journal of Heat and Mass Transfer*, **47**, 3927-3941. <https://doi.org/10.1016/j.ijheatmasstransfer.2004.02.028>
- [4] 吴自库, 李福乐, Kwak, D.Y. 一维热传导方程热源反问题基于最小二乘法的正则化方法[J]. 计算物理, 2016, 33(1): 49-55.
- [5] Raissi, M., Perdikaris, P. and Karniadakis, G.E. (2019) Physics-informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *Journal of Computational Physics*, **378**, 686-707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- [6] 陆金甫, 关治. 偏微分方程数值解法[M]. 第 2 版. 北京: 清华大学出版社, 2004: 82-120.
- [7] Peaceman, D.W. and Rachford Jr., H.H. (1955) The Numerical Solution of Parabolic and Elliptic Differential Equations. *Journal of the Society for Industrial and Applied Mathematics*, **3**, 28-41. <https://doi.org/10.1137/0103003>
- [8] Kaipo, J., Somersalo, E. 统计与计算反问题[M]. 刘逸侃, 译. 北京: 科学出版社, 2018: 13-43.
- [9] Yan, L., Yang, F. and Fu, C. (2009) A Bayesian Inference Approach to Identify a Robin Coefficient in One-Dimensional Parabolic Problems. *Journal of Computational and Applied Mathematics*, **231**, 840-850. <https://doi.org/10.1016/j.cam.2009.05.007>
- [10] 刘艳杰. 参数反演的贝叶斯方法及其应用研究[D]: [硕士学位论文]. 淄博: 山东理工大学, 2020.
- [11] 张国安. 一类二维多层对流扩散方程参数反演的神经网络方法研究[D]: [硕士学位论文]. 上海: 上海财经大学, 2022.
- [12] 刘云美, 史正梅. PINNs 在反演计算中影响因素的数值比较分析[J]. 新乡学院学报, 2024, 41(9): 14-21.
- [13] Jagtap, A.D., Kawaguchi, K. and Karniadakis, G.E. (2020) Adaptive Activation Functions Accelerate Convergence in

- Deep and Physics-Informed Neural Networks. *Journal of Computational Physics*, **404**, Article ID: 109136. <https://doi.org/10.1016/j.jcp.2019.109136>
- [14] Jagtap, A.D., Kawaguchi, K. and Em Karniadakis, G. (2020) Locally Adaptive Activation Functions with Slope Recovery for Deep and Physics-Informed Neural Networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **476**, Article ID: 20200334. <https://doi.org/10.1098/rspa.2020.0334>
- [15] Deng, Z., Yang, L., Yu, J. and Luo, G. (2013) Identifying the Diffusion Coefficient by Optimization from the Final Observation. *Applied Mathematics and Computation*, **219**, 4410-4422. <https://doi.org/10.1016/j.amc.2012.10.045>