arXiv:2603.01141v1 [math.OC] 1 Mar 2026

迈向无网格概率形状优化的第一步

Stephan Schmidt* Maximilian W¨urschmidt*

我们提出了一种概率偏微分方程约束形状优化算法的初步实现。该方法基于一种新颖的形状导数概率表示，并通过蒙特卡罗抽样进行评估；同时不依赖网格。底层状态通过基于神经网络的偏微分方程求解器在点云上表示。本文将该方法应用于一个基准跟踪问题。

数学主题分类（2020年）：49Q10, 68T07, 65C05

关键词：形状优化、无网格方法、蒙特卡罗抽样、物理信息神经网络

1 引言

由椭圆型偏微分方程支配的跟踪型形状优化，是一大类数学反问题的典型代表。这类问题与众多应用相关，尤其是电阻抗层析成像和地质阻抗层析成像。在这类问题中，人们试图通过观测电势间接重建夹杂物。夹杂物会影响基体材料的电导率，优化过程需要确定材料电导率参数，使模拟电势（状态）与测量电势（跟踪数据）相一致，参见例如 [7]。夹杂物与基体材料之间的锐界面可通过两个子区域来表示。问题在于优化两者之间界面的形状。如果夹杂物是一个真正的空洞，通常只需考虑单个区域。在最一般的形式下，参见例如 [27, 第 2 节。6], 这将导出以下问题

$$\min_{\Omega\subset\mathcal{H}} \int_\Omega \tfrac{1}{2}\big(u(x)-z_d(x)\big)^2\,\lambda^2(\mathrm{d}x) \quad \text{s.t.} \begin{cases} -\Delta u &= 1 \quad \text{on } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega. \end{cases} \tag{1.1}$$

w这里 $\mathcal{H}\subset\mathbb{R}^2$ 是一个矩形包含域 区域，而 $z_d\colon\ \mathcal{H}\to\mathbb{R}$ 是跟踪数据 映射。 [1]

*特里尔大学，第四系——数学。M. W¨urschmidt 谨此感谢德国研究联合会（DFG）通过研究生培训组 2126：算法优化提供的资助。[1]在此背景下，形状泛函为 $J\colon(u,\ \Omega)\to\frac{1}{2}\|u-z_d\|^2_{L^2(\Omega)}$ 其中，$(u,\ \Omega)$ 应理解为一一对应关系，即我们假设对于每个可容许形状 Ω，偏微分方程约束都有唯一的经典解 u。例如，满足外球条件时即属此情形，参见 [9, 定理 6.13]。

当数据含有噪声时，此类反问题通常需要额外的正则化项。通常，该泛函是区域的体积或周长，以促进重建出光滑的区域。然而，某些应用（例如地质阻抗问题）要求重建非光滑区域，参见例如 [4, 5]。

在基于梯度的形状优化中，可以通过算法1近似得到（局部）最优形状。随后，我们将介绍一种新方法：使用神经偏微分方程求解器计算状态，并采用形状导数的概率表示来计算形状梯度。

算法1： 形状梯度下降

初始化 选择初始区域 Ω 和容差 $\varepsilon_{\mathrm{TOL}}$

当 $\|\nabla J(\Omega)\| > \varepsilon_{TOL}$ **时**

1. 计算 Ω 上的状态 u
2. 计算形状梯度 $\nabla J(\Omega)$
3. 执行线搜索以获得 α [可选]
4. 对 Ω 应用 $\alpha \nabla J(\Omega)$ 的几何流

end

近年来，人们提出并分析了多种基于深度学习的算法，例如所谓的物理信息神经网络（PINNs），以及利用随机框架的不同变体——深度 BSDE 求解器；这些算法确实能够逼近偏微分方程解。有关综述，请参阅 [3, 17, 22, 30] 和 [13, 第四部分]。此外，这些算法有望克服维数灾难，参见例如 [11]。

形状导数的概率表示针对一般状态空间维数在 [24] 中得到发展。本文考虑一个二维基准问题，但我们强调，概率公式以及下文讨论的（无网格）蒙特卡洛模拟均不受维数灾难的影响，参见例如 [6, 8]。

因此，该方法的一项关键优势在于，当前形状仅通过点云或等效的水平集描述来表示，使这一框架能够扩展到高维情形。

在此阶段，我们的实现需要边界网格，主要用于计算形状梯度（算法1中的步骤2），并且为了方便起见，也用于概率形状导数计算的部分环节。[2]

2 通过物理信息神经网络计算状态

我们使用物理信息神经网络（物理信息神经网络）来近似状态，见 [22, 26]。这是一种无监督、亦即自包含的学习算法——全面介绍请参见 [13, 第 16] 章——应用于具有光滑激活函数的前馈神经网络 $\mathrm{NN} \triangleq \mathrm{NN}^{\theta}\colon\ \mathbb{R}^d \to \mathbb{R}$。这里 θ 表示网络的权重，由以下元组给出

[2]代码已公开发布，地址为https://github.com/max-wuer/ProbabilisticShape

：权重矩阵和偏置向量——深入介绍请参见 [21, 第 2] 章。需要指出的是，一旦指定了激活函数，每个网络便由其权重唯一确定。

在给定的有界区域 $\Omega \subset \mathcal{H}$ 上，我们根据预先设定的分布抽取配点 $\mathbf{X}_\Omega \times \mathbf{X}_\Gamma \subset \Omega \times \partial\Omega$。训练过程中的目标函数是计算 PINN 损失泛函的极小化器。对于与问题 (1.1) 对应的泊松方程，该泛函为

$$\mathfrak{L}^{\text{PINN}}\colon \text{NN} \mapsto \tfrac{1}{|\mathbf{X}_\Gamma|+|\mathbf{X}_\Omega|}\Big(2|\mathbf{X}_\Gamma| \sum_{x\in\mathbf{X}_\Gamma} \big|\text{NN}(x)\big|^2 + |\mathbf{X}_\Omega| \sum_{x\in\mathbf{X}_\Omega} \big|\Delta\text{NN}(x)+1\big|^2\Big) \tag{2.1}$$

N注意，PINN 泛函取决于配点样本 $\mathbf{X}_\Omega \times \mathbf{X}_\Gamma$。 us

$$\mathfrak{L}^{\text{PINN}}(\text{NN}) = \mathfrak{L}^{\text{PINN}}(\text{NN};\, \mathbf{X}_\Omega \times \mathbf{X}_\Gamma) = \mathfrak{L}^{\text{PINN}}(\text{NN}^\theta;\, \mathbf{X}_\Omega \times \mathbf{X}_\Gamma) = \mathfrak{L}^{\text{PINN}}(\theta;\, \mathbf{X}_\Omega \times \mathbf{X}_\Gamma)$$

假设配点是从 Ω 上具有全支撑的连续分布中独立抽取的，根据大数定律，由抽样过程引起的任何依赖性都会渐近地变得可以忽略。

$\mathfrak{L}^{\text{PINN}}$ 的极小化通过梯度下降完成，参见例如 [21, 第 10]章。这里，拉普拉斯算子 Δ NN 和 PINN 损失梯度 $\partial_\theta \mathfrak{L}^{\text{PINN}}$ 的计算均通过自动微分完成。

3 通过概率形状导数计算形状梯度 e

本节讨论算法1的第二步，即计算形状更新，也就是每次下降步骤中的有效形状变形。与第2节一样，我们考虑一个一般的当前区域$\Omega \subset \mathcal{H}$，并说明如何计算形状更新。

本讨论分为两部分：首先，我们介绍形状导数的概率表示 [24], ，它基于对底层状态的边界灵敏度进行 Feynman-Kac展开，并用可在无网格情况下模拟的随机变量的期望来表示导数；其次，我们讨论形状梯度的计算，而这一步需要引入边界网格。

注3.1。 对于具有形状优化背景的读者，我们强调，概率表示 $\mathbb{D}J$（见下文(3.1)）基于恒等映射的原像／拉回扰动。因此，概率方法得到的是经典意义上考虑的方向形状导数 $\mathcal{D}J$，它基于恒等映射的推前扰动，但符号相反，即

$$\mathbb{D}J[V] = \mathcal{D}J[-V] \qquad \textit{for all admissible perturbations } V \qquad \diamond$$

关于形状优化的全面介绍，请参阅近期的综述 [1]。关于经典的推前形状导数概念，尤其请参阅 [1, Definition4.1]。

注3.2. 我们用符号 $\partial\Omega$表示无网格边界，用 $\partial\Omega_{FEM}$ 表示相应的边界网格。此外，当使用状态的 *PINN* 近似对当前区域进行数值描述时，我们记作Ω_{NN}。$\diamond$

概率形状导数 对于一个可容许扰动 $V\colon \partial\Omega \to \mathbb{R}^2$，将 [24, 定理 3.7] 中的公式应用于我们的跟踪型问题，可得

$$\mathbb{D}J[V] = m^+ \,\mathbb{E}\Big[\big\langle V, \nabla u\big\rangle\big(\widehat{X}^{X_0^+}\big)\Big] - m^- \,\mathbb{E}\Big[\big\langle V, \nabla u\big\rangle\big(\widehat{X}^{X_0^-}\big)\Big] \\ - \int_{\partial\Omega} \tfrac{1}{2} z_d(y)^2 \,\big\langle V, n\big\rangle(y)\; \mathcal{S}^1(\mathrm{d}y) \quad (3.1)$$

其中，$\widehat{X}^{X_0^+}$, $\widehat{X}^{X_0^-}$ 是所谓的随机起始退出-杀死随机变量，取值于 $\partial\Omega$和 m^+，$m^- \geq 0$ 是常数，稍后将予以介绍。

(3.1)中的两个期望具有相似的结构，区别仅在于随机起始分布的支撑集不同。对于跟踪型问题，给定当前状态 u，考虑区域[3] 的划分

$$\Omega^+ \triangleq \big\{x \in \Omega \,\big|\, u(x) \geq z_d(x)\big\} \qquad \text{and} \qquad \Omega^- \triangleq \Omega \setminus \Omega^+$$

我们用 $\pm$ 作为占位符，表示其依赖于所选子区域。因此

$$m^\pm = \pm \int_{\Omega^\pm} u(x) - z_d(x) \lambda^2(\mathrm{d}x)$$

随机起点退出-终止变量由一个随机起点 $X_0^\pm$构成，其服从分布$\rho^\pm \mathrm{d}\lambda^2$，且具有显式可得的密度函数

$$\rho^\pm \colon \mathbb{R}^d \to [0,\infty);\; x \mapsto \tfrac{1}{\pm m^\pm}\big(u(x) - z_d(x)\big)\; \mathbb{1}_{\Omega^\pm}(x) \quad (3.2)$$

这些密度函数取决于Ω的划分。对于跟踪问题，不存在杀死，因此退出由布朗运动 W的动力学决定，即停时 $\tau \triangleq \inf\{t \geq 0 \mid X_0^\pm + \sqrt{2}W_t \notin \Omega\}$。于是，(3.1)中的随机变量由上述过程[4] 在首次退出时的取值给出；

$$\widehat{X}^{X_0^\pm} = X_0^\pm + \sqrt{2}W_\tau \quad (3.3)$$

本段余下部分将介绍形状导数概率表示的无网格评估，见 (3.1)。

全文中，我们假设 NN $\approx u$ 是针对当前区域 Ω 上状态训练得到的物理信息神经网络，即 NN 是物理信息神经网络损失函数 (2.1) 的一个（局部）极小化器的良好近似。我们使用该物理信息神经网络近似 $\Omega \approx \Omega_{\mathrm{NN}} \triangleq \{\mathrm{NN} > 0\}$，这分别由极大值原理和 Hopf 引理所支持，见 [12]。类似地，我们将区域划分近似为

$$\Omega^+_{\mathrm{NN}} \triangleq \{\mathrm{NN} \geq z_d\} \cap \{\mathrm{NN} \geq 0\} \qquad \text{and} \qquad \Omega^-_{\mathrm{NN}} \triangleq \{\mathrm{NN} \leq z_d\} \cap \{\mathrm{NN} \geq 0\}$$

随机起点的模拟以及常数的计算均通过接受-拒绝抽样获得，见算法 2。更多细节请参见例如 [10, 第 2.2 节。2] 在所有应用中，我们均采用包容全域上的均匀分布作为相应的参考密度，即 $\rho_{\mathrm{ref}} = \lambda^2(\mathcal{H})^{-1}\mathbb{1}_{\mathcal{H}\circ}$

[3]注意，根据目标函数 (1.1)，该划分对应于映射 $\phi\colon (x,\ y) \mapsto \frac{1}{2}(y - z_d(x))^2$ 在水平集 $\Omega^+ = \{x \in \Omega \mid \partial_y \phi(x,\ u(x)) \geq 0\}$ 上。[4]请注意，这是一个在离开开集时停止的连续时间过程，因此其取值属于 $\partial\Omega$。

算法 2: （批量）接受拒绝采样

要求 参考密度 ρ_{ref}: $\mathbb{R}^d \to [0,\infty)$，我们可以从中抽样，并且存在满足 $\rho(x)$ $\leq C\rho_{\text{ref}}(x)$ 对所有 $x \in \mathbb{R}^d$ 均成立的 $C > 0$。**初始化** 选择从目标密度 ρ 中独立抽取的样本数量 $M \in \mathbb{N}$。设置$\mathbf{X}^\rho = \emptyset$。**当** $|\boldsymbol{X}^\rho| < M$ **时1.** 抽取 $X \sim \mathbb{P}_{\text{ref}} = \rho_{\text{ref}}\mathrm{d}\lambda^d$**2.** 抽取 $U \sim$ Uniform(0,1)**如果** $U \leq \ ^{\rho}{}_{(X)\, C\rho_{ref}(X)}$ **则** 接受样本：$\mathbf{X}^\rho \leftarrow \mathbf{X}^\rho \cup \{X\}$**结束 返回**样本集合 $\mathbf{X}^\rho$。

注意，对于$\Omega^\pm$ 上均匀分布的$\mathcal{U}^\pm$，成立

$$m^\pm \approx \pm \int_{\Omega^\pm} \mathrm{NN}(x) - z_d(x)\lambda^2(\mathrm{d}x) = \pm\lambda^2(\Omega^\pm)\,\mathbb{E}\big[\mathrm{NN}(\mathcal{U}^\pm) - z_d(\mathcal{U}^\pm)\big] \tag{3.4}$$

我们使用算法 2 分别近似每个因子。对两者，我们均使用上文的参考密度 ρ_{ref}、常数 $C = 1$，并以密度$\rho = {}^{1}{}_{\lambda^2 \ (\Omega^\pm \ \mathrm{NN})}\mathbb{1}_{\Omega^\pm \ \mathrm{NN}}$作为目标，以生成一组样本$\mathbf{X}^{\text{uniform}\ \Omega^\pm \ \mathrm{NN}}$。

注意 $\lambda^2(\Omega^\pm) = \lambda^2(\mathcal{H})\ \mathbb{P}_{\text{ref}}[\Omega^\pm]$。因此，勒贝格测度作为特例得到，此时蒙特卡洛估计退化为接受样本数与试验总数之比，即

$$\mathbb{P}_{\text{ref}}[\Omega^\pm] \approx \mathbb{P}_{\text{ref}}[\Omega^\pm_{\text{NN}}] \approx \frac{\#\text{ successful while loops}}{\#\text{ total while loops}} \tag{3.5}$$

特别地，此处并未使用样本$\mathbf{X}^{\text{uniform}\ \Omega^\pm \ \mathrm{NN}}$ 的具体坐标。接下来，利用同样的样本，我们计算第二个因子，即期望

$$\mathbb{E}\big[\mathrm{NN}(\mathcal{U}^\pm) - z_d(\mathcal{U}^\pm)\big] \approx {}^{1}{}_{|\mathbf{X}^{\text{uniform}}_{\Omega^\pm_{\text{NN}}}|} \sum_{x \in \mathbf{X}^{\text{uniform}}_{\Omega^\pm_{\text{NN}}}} \mathrm{NN}(x) - z_d(x).$$

有了这些近似之后，我们令 $m^\pm_{\text{NN}}$ 表示(3.4)各自相应的数值计算版本。

随机起点退出-终止变量（3.3）按顺序进行采样；具体而言，算法 2 生成随机起点集合 $\mathbf{X}^\pm_0$，然后使用欧拉-丸山格式计算这些起点在区域边界上的退出坐标，参见算法3。

为此，我们采用接受-拒绝抽样（见算法 2），并以包含域上的均匀分布作为参考密度 ρ_{ref}。目标密度函数由显式映射给出 ρ^+，ρ^- 如式 (3.2) 所示，并使用 PINN水平集进行近似，即:

$$\rho^\pm_{\text{NN}} \triangleq {}^{1}{}_{m^\pm_{\text{NN}}} \big|\mathrm{NN} - z_d\big|\ \mathbb{1}_{\Omega^\pm_{\text{NN}}}$$

并将其作为接受条件中的常数（我们使用与上文相同的样本 $\mathbf{X}^{\text{uniform}\ \Omega^{\pm}\ \text{NN}}$）；

$$C^{\pm} = \lambda^2(\mathcal{H})\ \text{SF}\ \max\left\{\rho^{\pm}_{\text{NN}}(x)\ \middle|\ x \in \mathbf{X}^{\text{uniform}}_{\Omega^{\pm}_{\text{NN}}}\right\} \tag{3.6}$$

其中，SF $>$ 1表示安全系数，以确保算法中的估计值能够从均匀采样的点云合理地外推至整个水平集 $\Omega^{\pm}_{\text{NN}}$。由此，我们得到随机起点 $\mathbf{X}^{\pm}_0$。第二步，按照算法3计算相应的出口点 $\mathbf{X}^{\mathbf{X}^{\pm}_0}$ NN,FEM。这里，只有投影到 $\partial\Omega_{\text{FEM}}$ 时才需要底层边界网格。原因有二：一方面，使用网格法向很方便；另一方面，由于有限元在其底层网格之外进行评估时不够稳健，因此我们要求以机器精度命中边界。[5]

算法3： 随机起点出口（终止）变量采样

初始化 采样起点 $x_0 \leftarrow X^{\pm}_0$。设置离散时间步长 $\Delta \in (0,1)$。
当 $x_k \in \Omega_{NN}$ **时执行**
  欧拉-丸山更新：$x_k \leftarrow x_{k-1} + \sqrt{2}\sqrt{\Delta}\xi$ 对于 $\xi \sim \mathcal{N}(0,\mathcal{I})$
end
返回 将 x_k 正交投影到边界 $\partial\Omega_{\text{FEM}}$

(3.1) 中边界积分的近似使用边界网格 $\partial\Omega_{\text{FEM}}$ 及其对应的面法向。在此处，为方便起见采用 $\partial\Omega_{\text{FEM}}$，因为它已经可用。另一种做法是在 $\partial\Omega$上采用均匀分布，并如上所述进行蒙特卡洛模拟。在这种情况下，参考测度是限制在 $\partial\Omega$上的一维豪斯多夫测度，即 $\mathcal{S}^1(\cdot \cap \partial\Omega)$。相应的近似如下

$$\int_{\partial\Omega_{\text{FEM}}} \tfrac{1}{2}\big[z_d^2\langle V, n_{\text{FEM}}\rangle\big](y)\mathcal{S}^1(\mathrm{d}y) \approx \int_{\partial\Omega} \tfrac{1}{2}\big[z_d^2\langle V, n\rangle\big](y)\mathcal{S}^1(\mathrm{d}y) = \mathbb{P}_{\mathcal{S}^1}(\partial\Omega)\ \mathbb{E}\Big[\tfrac{1}{2}\big[z_d^2\langle V, n\rangle\big](\mathcal{U})\Big]$$

其中 $\mathcal{U}$ 相对于测度 $\mathbb{P}_{\mathcal{S}^1} \triangleq {}^{1}_{\mathcal{S}^1\,\partial(\Omega)}\mathcal{S}^1(\cdot \cap \partial\Omega)$ 服从均匀分布 ——— .

变形方程 概率形状导数 (3.1) 为我们提供了一个方向导数，我们通过里斯映射将其转化为梯度

$$\langle W, V\rangle_{H^1(\partial\Omega)} \overset{!}{=} \mathbb{D}J[V] \quad \text{for all admissible directions } V.$$

使用 H^1 内积有几个优点。一方面，它能确保未知区域边界 $\partial\Omega$ 保持在所需的正则类中。另一方面，这一过程可以解释为一种近似牛顿格式，其中 H^1内积

[5]为完整起见，我们提及，在 $\partial\Omega_{\text{FEM}}$ 的每个顶点处，都存在一个没有明确定义的投影法向量的锥。若首次退出发生在该锥内，我们就投影到相应的顶点上。总之，我们始终投影到 $\partial\Omega_{\text{FEM}}$中距离最近的点上。具体而言，若 $x \in \mathbb{R} \setminus \Omega_{\text{NN}}$ 是一次欧拉–丸山首次退出的实现，令 v 表示距离最近的顶点，w 表示距离 x 最近的第二个顶点，其中 x 位于 $\partial\Omega_{\text{FEM}}$中。相应地，令 n_{FEM} 为边(v,w)的外法向量，然后设定 $x_p \leftarrow (\mathcal{I} - n_{\text{FEM}}n_{\text{FEM}}^{\top})x + n_{\text{FEM}}n_{\text{FEM}}^{\top}v$，以得到 $\partial\Omega_{\text{FEM}}$上的最近点。

是问题海森矩阵的一种近似。更多详情请参见 [25, 第 3],章，那里使用了实际的海森矩阵。虽然存在无网格的拉普拉斯–贝尔特拉米算子实现，也就是 H^1内积对应的偏微分方程，参见例如 [28, 29], ，但这些方法通常并不简单，且超出了本文的讨论范围。

现在我们引入边界网格 $\partial\Omega_{\text{FEM}} = (\{x_j\}_{j\le k},\ \mathcal{E})$，即二维中的一条闭合多边形链，其顶点为 $\{x_j\}_{j\le k}$、边为 $\mathcal{E}$，且外法向量 n_{FEM} 在每条边上均为常量，另见注3.2。在此网格上，我们定义一阶连续拉格朗日有限元空间；

$$\mathcal{V}_{\text{FEM}} \triangleq \left\{ v \in \mathcal{C}(\partial\Omega_{\text{FEM}}, \mathbb{R}) \,\middle|\, v|_e \in P_1(e)\ \forall\, e \in \mathcal{E} \right\}.$$

这里， P_1 是所有一阶多项式的集合， $\mathcal{E}$ 是网格边的集合。我们为该空间 $\mathcal{V}_{\text{FEM}}$ 配备基 $\{v_j\}_{j\le k}$，其具有如下性质

$$v_j(x_i) = \delta_{ij} \triangleq \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

每个 v_j 的支集恰好位于两条相邻边 e^ℓ、 e^r 上，它们对应的常量外法向量分别为 n^ℓ_{FEM}、n^r_{FEM}。因此，我们可以将可容许变形集合定义为以下函数的张成空间

$$V_j \colon \partial\Omega_{\text{FEM}} \setminus \left(\partial e^\ell \cup \partial e^r \right) \to \mathbb{R}^2; \quad x \mapsto \begin{cases} v_j(x) n^\ell_{\text{FEM}}, & x \in e^{\ell,\circ} \\ v_j(x) n^r_{\text{FEM}}, & x \in e^{r,\circ} \\ 0, & \text{otherwise} \end{cases}$$

其中 e° 表示相应边的内部。无需在（三个）顶点 $\partial e^\ell \cup \partial e^r$ 上作定义，因为这些点不会成为概率形状导数中的出口点。事实上，现在可以利用经典有限元确定概率形状导数（3.1）的 $H^1(\partial\Omega_{\text{FEM}})$ 梯度，即求出满足下式的 $w \in \mathcal{V}_{\text{FEM}}$

$$\sum_{e\in\mathcal{E}} \sum_{s=1}^{k_e} \omega_j^e \left(\frac{1}{2} \left\langle \nabla_\Gamma v_j|_e(\tilde{x}_s^e), \nabla_\Gamma w|_e(\tilde{x}_s^e) \right\rangle + \left\langle v_j|_e(\tilde{x}_s^e), w|_e(\tilde{x}_s^e) \right\rangle \right) \overset{!}{=} \mathbb{D}J[V_i] \quad \forall i = 1, \ldots, k.$$

这里， ∇_Γ 表示切向梯度[6]，而 ω_j^e 是边e 上 $\tilde{x}_s^e$ 处的求积权重，该边具有 k_e 个求积点。为了利用该 H^1（$\partial\Omega_{\text{FEM}}$）梯度更新形状，我们随后计算 $w\, n_{\text{FEM}}$ 到空间 $\mathcal{V}_{\text{FEM}} \times \mathcal{V}_{\text{FEM}}$ 的伽辽金投影 W，并将下一个区域设为

$$\partial\Omega^{\ell+1}_{\text{FEM}} = \left\{ x_j + \tau W(x_j) \mid x_j \text{ vertex in } \partial\Omega^\ell_{\text{FEM}} \right\},$$

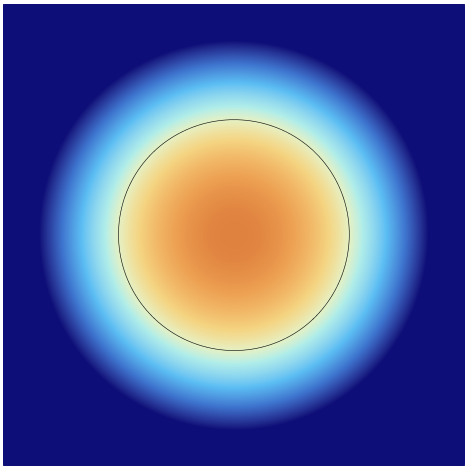
其中 $\alpha > 0$ 是合适的步长。关于符号翻转，请回顾注3.1。

4 数值结果

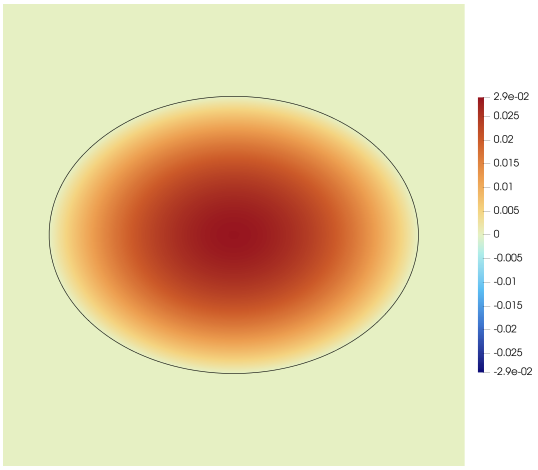
在模拟中（见图1），我们考虑一个最优形状明确可得的情形。具体而言，我们令 z_d 为偏微分方程约束的解，其区域 Ω^* 为一个椭圆

[6]注意，这是 $\mathcal{V}_{\text{FEM}}$中函数的自然梯度概念

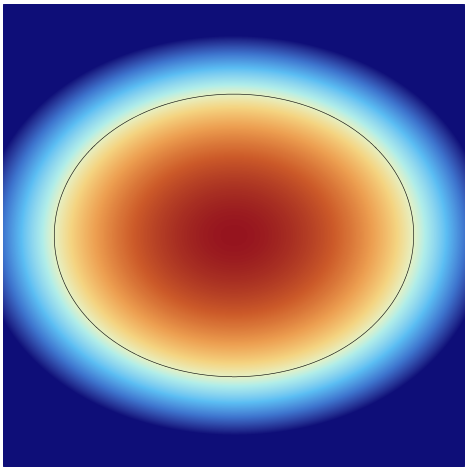
其中心位于 $c = (\frac{1}{2}, \frac{1}{2})$，长半轴为 $a = 0.4$，短半轴为 $b = 0.3$；并将其以常值零延拓至包含域 $\mathcal{H}$。[7] 在这种情况下，椭圆 Ω^* 满足式 (1.1)。



(a) 初始形状。为便于可视化，对物理信息神经网络进行点求值，并采用线性插值。
零水平集 $\{\mathrm{NN} = 0\}$ 以黑色表示。



(b) 包含域 z_d 上的跟踪数据 $\mathcal{H}$。狄利克雷条件在内界面 $\partial\Omega$上成立，内部区域为 Ω^*上形状优化问题的解析解。



(c)重构形状 Ω^*_{NN}。物理信息神经网络的评估；为便于可视化，采用线性插值。
水平集 {黑色显示 $\mathrm{NN} = 0$} 。

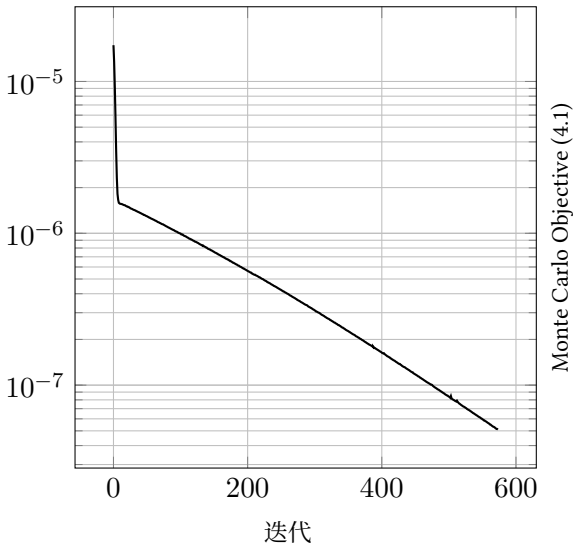


图1：优化结果。

我们从一个球 $B_R(c)$开始执行形状梯度下降方案（见算法1）。该球与椭圆具有相同的中心，半径为 $R = 0.25$。此框架允许将包含域设为 $\mathcal{H} = [0,1]^2$，因此因子 $\lambda^2(\mathcal{H}) = 1$ 可以忽略不计。

我们采用一个全连接前馈神经网络，其中包含两个隐藏层，每层有 20 个神经元，并使用双曲正切作为激活函数。具体而言，我们在包含域上近似偏微分方程

[7]请注意，该映射是明确可用的：$z_d : \mathcal{H} \to \mathbb{R}$；$(x_1, x_2) \mapsto 2\frac{a^2b^2}{(a^2+b^2)}\left(1 - \frac{(x_1-c_1)^2}{a^2} - \frac{(x_2-c_2)^2}{b^2}\right)\mathbb{1}_{\Omega^*}(x_1, x_2)$。

区域 $\mathcal{H}$ 在 $\partial\Omega$处具有内部狄利克雷界面。主要原因是，我们使用物理信息神经网络来通过水平集 $\{\mathrm{NN} \geq 0\}$描述当前区域。此外，这种方法显著提高了状态梯度的近似质量。

底层配点$\mathbf{X}_\Omega \times\mathbf{X}_\Gamma$ 由独立采样的点云组成。内部点$\mathbf{X}_\Omega$ 按照 $\mathcal{H}$ 上的均匀分布抽取。狄利克雷点$\mathbf{X}_\Gamma$ 通过边界网格 $\partial\Omega_{\mathrm{FEM}}$ 生成，具体做法是从每条边上分别进行均匀采样。具体而言，我们使用 $|\mathbf{X}_\Omega| = 45000$ 个样本和 $|\mathbf{X}_\Gamma| = 22500$ 个样本。对于狄利克雷点，无论边的几何长度如何，每条边抽取的样本数都相同。物理信息神经网络的优化使用 Adam优化器进行，参见 [14], ，并采用分段常数学习率。实现使用 PyTorch 库完成，参见 [19]。我们强调，优化步数存在巨大差异：初始训练耗时较长，需要进行 50000 – 100000 步梯度下降，并且多次重新启动训练，以找到非常好的局部最优解。在算法1的每次后续迭代中，区域变形后，我们会进行迁移学习，并在最多 200 步后中断训练。

概率公式各组成部分的蒙特卡洛模拟采用以下样本量：我们抽取 $|\mathbf{X}^{\mathrm{uniform}}{}_{\Omega^\pm{}_{\mathrm{NN}}}| = 10^7$ 个样本，以近似$\lambda^2(\Omega^\pm)$以及随机起点接受-拒绝抽样所需的常数 (3.6)。此外，我们使用 $\mathbf{X}^{\mathrm{uniform}}{}_{\Omega^\pm{}_{\mathrm{NN}}}$ 样本中的 10^6 来近似 $m^\pm$中的期望；安全系数为 $\mathrm{SF} = 1.1$。我们仅抽取 10^2 个随机起点样本$\mathbf{X}_0^\pm$，因此得到相同数量的退出样本$\mathbf{X}^{\mathbf{X}_0^\pm}$ NN，FEM。由于样本量较小，所得蒙特卡洛估计的方差相对较高。不过，对于我们的基准示例，精度已足够，能够在精度和计算时间之间取得良好平衡。

我们使用 FEniCS 库求解变形方程，参见 [16], ；该库原生支持曲面（即我们的多边形链）上的有限元函数空间和微分算子，参见[23]。对于图1所示的结果，我们在该多边形链上使用了 160 个顶点。我们还使用 FEniCS 库，通过在点云上计算 PINN 的线性插值来可视化图中的 PINN 解。

目标函数使用蒙特卡洛方法，通过先前生成的$\mathbf{X}^{\mathrm{uniform}}{}_{\Omega^\pm{}_{\mathrm{NN}}}$ 进行评估，即:

$$\begin{aligned}\int_\Omega \tfrac{1}{2}(u - z_d)^2 \,\mathrm{d}\lambda^2 &\approx \lambda^2(\Omega_{\mathrm{NN}})\, \mathbb{E}\Big[\tfrac{1}{2}\big(\mathrm{NN}(\mathcal{U}) - z_d(\mathcal{U})\big)^2\Big] \\ &\approx \big(\bar{\mathbb{P}}_{\mathrm{ref}}[\Omega^+_{\mathrm{NN}}] + \bar{\mathbb{P}}_{\mathrm{ref}}[\Omega^-_{\mathrm{NN}}]\big) \sum_{x \in \mathbf{X}^{\mathrm{uniform}}_{\Omega^+_{\mathrm{NN}}} \cup \mathbf{X}^{\mathrm{uniform}}_{\Omega^-_{\mathrm{NN}}}} \tfrac{1}{2}\big(\mathrm{NN}(x) - z_d(x)\big)^2, \end{aligned} \tag{4.1}$$

其中，$\mathcal{U}$ 表示 Ω 上均匀分布的随机变量，而 $\bar{\mathbb{P}}_{\mathrm{ref}}[\Omega^\pm_{\mathrm{NN}}]$ 表示对子区域测度的数值评估，即式 (3.5) 中的分数。

5 全无网格概率框架的展望

本研究旨在展示形状导数随机公式的适用性。我们特别强调，概率形状导数

独立于给定的形状离散化，因此尤其适用于无网格方法。数值实现主要基于蒙特卡罗采样和物理信息神经网络，因此，所提出的方法有望规避空间维度较多的问题中的维数灾难。我们使用概率形状导数求解拉普拉斯型跟踪问题，并仅用一条多边形链来跟踪形状。这样做是为了方便计算 $H^1(\partial\Omega_{\mathrm{FEM}})$ 下降方向；此外，也存在无网格替代方法。

我们强调，只要能够定义外法向量 n，并且可以从边界进行均匀采样，概率形状导数就能用于区域迭代的任何表示方式。常见的选择包括网格或水平集表述。

在我们的应用中，我们训练物理信息神经网络来求解包含域 $\mathcal{H}$ 上的偏微分方程，并在内部界面 $\partial\Omega$ 上施加（常数）狄利克雷条件。具体而言，训练好的物理信息神经网络可作为区域的水平集表述，我们可以计算 $n \approx \nabla\mathrm{NN}/\|\nabla\mathrm{NN}\|$。此外，借助物理信息神经网络的水平集表述，可以生成所需样本，以对（1.1）中的边界积分进行蒙特卡洛模拟，参见例如 [15, 31, 32]。

求解变形方程的一种概率方法，是通过非参数密度估计计算 $L^2(\partial\Omega)$ 的黎斯表示元。为此，令 $\nu^{\pm}$ 表示随机起点退出-终止变量的分布。假设存在密度函数 $\ell^{\pm}\colon \partial\Omega \to [0,\infty)$，使得 $\mathrm{d}\nu^{\pm} = \ell^{\pm}\mathrm{d}\mathcal{S}^1$ 映射 $\nabla J \triangleq z_d^2 n + \nabla u\big(m^+\,\ell^+ - m^-\,\ell^-\big)$ 满足

$$\langle \nabla J, V\rangle_{L^2(\partial\Omega)} \overset{!}{=} \mathbb{D}J[V] \quad \text{for all admissible directions } V$$

因此，∇J 确实是 L^2 形状梯度。可以基于边界上的某种聚合方式构建直方图，进行朴素估计。为了提高估计精度，可以使用流形上的核密度估计器，例如参见 [18, 20]。

最后，为了便于读者了解，我们说明形状优化循环采用固定步长运行，见图1。请注意，线搜索策略通常会检验 Armijo–Goldstein条件，即当步长满足以下条件时接受该步：

$$f(x+\alpha p) \le f(x) + \alpha c\,\langle \nabla f(x), p\rangle\,.$$

这里，α 是步长，p 是搜索方向，$0 < c < 1$ 是阻尼因子。因此，是否接受该步取决于实际下降量与目标函数预期下降量的比较。然而，在形状优化中，网格质量也是一个限制因素。尽管我们只需使用一条多边形线来计算 H^1下降方向，但步长过大仍会导致边反转，而 Armijo–Goldstein规则无法检测到这种情况。基于此，我们使用人工确定的固定步长，远小于更严格的 Armijo–Goldstein条件所允许的步长。否则，就必须基于一个包含网格质量的 merit函数来制定 Armijo准则，例如参见 [2, 第 5]节。

参考文献

[1] Gr´egoire Allaire、Charles Dapogny和Franc¸ois Jouve。第1章——形状与拓扑优化。载于Andrea Bonito和Ricardo H. Nochetto编，几何偏微分

微分方程——第二部分，数值分析手册第22卷，第1–132页。爱思唯尔，2021年。[2] Lukas Baumg¨artner、Ronny Bergmann、Roland Herzog、Stephan Schmidt、Jos´e Vidal-N´u ˜nez和Manuel Weiß。利用法向量的全变差与形状牛顿方法进行网格去噪和图像修复。《SIAM科学计算杂志》，47(1):A300–A324，2025年。[3] Christian Beck、Martin Hutzenthaler、Arnulf Jentzen和Benno Kuckuck。基于深度学习的偏微分方程逼近方法综述。《离散与连续动力系统：B辑》，第3697–3746页，2023年。[4] Ronny Bergmann、Marc Herrmann、Roland Herzog、Stephan Schmidt和Jos´e Vidal N´u ˜nez。以法向量场的离散全变差作为形状先验及其在几何反问题中的应用。《反问题》，36(5):054003，2020年。[5] Ronny Bergmann、Marc Herrmann、Roland Herzog、Stephan Schmidt和Jos´e Vidal N´u ˜nez。以法向量场的全变差作为形状先验。《反问题》，36(5):054004，2020年。[6] Russel E. Caflisch。蒙特卡洛方法与拟蒙特卡洛方法。《数值学报》，7:1–49，1998年。[7] Margaret Cheney、David Isaacson和Jonathan C. Newell。电阻抗断层成像。《SIAM评论》，41(1):85–101，1999年。[8] Andrew Gelman、Walter R. Gilks和Gareth O. Roberts。随机游走Metropolis算法的弱收敛与最优尺度调整。《应用概率年刊》，7(1):110–120，1997年。[9] David Gilbarg和Neil S. Trudinger。《二阶椭圆偏微分方程》。施普林格，2001年。[10] Paul Glasserman。《金融工程中的蒙特卡洛方法》。施普林格，2004年。[11] Philipp Grohs和Lukas Herrmann。带边界条件的高维椭圆型偏微分方程的深度神经网络逼近。《IMA数值分析杂志》，42(3):2055–2082，2022年。[12] Eberhard Hopf。关于二阶线性椭圆微分方程的一个注记。《美国数学会会刊》，3(5):791–793，1952年。[13] Arnulf Jentzen、Benno Kuckuck和Philippe von Wurstemberger。深度学习的数学导论：方法、实现与理论。arXiv预印本arXiv:2310.20360，2023年。[14] Diederik P. Kingma和Jimmy Ba。Adam：一种随机优化方法。载于国际学习表征会议（ICLR）论文集，2015年。[15] Selena Ling、Abhishek Madan、Nicholas Sharp和Alec Jacobson。通过投射射线进行曲面均匀采样。《计算机图形学论坛》，44(5):e70202，2025年。[16] Anders Logg、Kent-Andre Mardal和Garth N. Wells编。用有限元方法自动求解微分方程，计算科学与工程讲义第84卷。施普林格，2012年。

[17] Nikolas Nusken 和 Lorenz Richter。倒向随机微分方程与物理信息神经网络之间的插值：用于椭圆和抛物边值问题的深度学习。机器学习杂志，2:31–64，2023年。[18] Arkadas Ozakin 和 Alexander Gray。子流形密度估计。载于神经信息处理系统进展：*NeurIPS* 论文集（*2009*），第22卷。Curran Associates, Inc.，2009年。[19] Adam Paszke、Sam Gross、Francisco Massa、Adam Lerer、James Bradbury、Gregory Chanan、Trevor Killeen、Zeming Lin、Natalia Gimelshein、Luca Antiga 等。PyTorch：一种命令式风格的高性能深度学习库。神经信息处理系统进展，32，2019年。[20] Bruno Pelletier。黎曼流形上的核密度估计。统计与概率快报，73(3):297–304，2005年。[21] Philipp Petersen 和 Jakob Zech。深度学习的数学理论。*arXiv:2407.18384*，2024年。[22] Maziar Raissi、Paris Perdikaris 和 George E Karniadakis。物理信息神经网络：用于求解涉及非线性偏微分方程的正问题和逆问题的深度学习框架。计算物理学杂志，378:686–707，2019年。[23] Marie E. Rognes、David. A. Ham、Colin J. Cotter 和 Andrew T. T. McRae。在 FEniCS 1.2 中自动求解球面及其他流形上的偏微分方程。地球科学模型开发，6(6):2099–2119，2013年。[24] Luka Schlegel、Volker Schulz、Frank T Seifried 和 Maximilian W¨urschmidt。形状导数的概率方法。*arXiv:2409.15967*，2024年。[25] Stephan Schmidt 和 Volker H. Schulz。形状优化的线性视角。*SIAM* 控制与优化杂志，61(4):2358–2378，2023年。[26] Justin Sirignano 和 Konstantinos Spiliopoulos。深度Galerkin方法：一种求解偏微分方程的深度学习算法。计算物理学杂志，375:1339–1364，2018年。[27] Jan Sokolowski 和 Jean-Paul Zolesio。形状优化导论：形状灵敏度分析。施普林格，2012年。[28] Pratik Suchde 和 J¨org Kuhnert。用于曲面偏微分方程的无网格广义有限差分法。计算机与应用数学，78(8):2789–2805，2019年。[29] Tasuku Tamai、Kouhei Murotani 和 Seiichi Koshizuka。关于基于粒子的无网格离散格式对拉普拉斯算子的一致性与收敛性。*Comput-ers &Fluids*，142:79–85，2017年。[30] Derick Nganyu Tanyu、Jianfeng Ning、Tom Freudenberg、Nick Heilenk¨otter、Andreas Rademacher、Uwe Iben 和 Peter Maass。偏微分方程及相关参数识别问题的深度学习方法。《反问题》，39(10)，2023年。[31] Kerun Xu 和 Miranda Holmes-Cerfon。高维流形上的蒙特卡洛。计算物理学杂志，506:112939，2024年。

[32] Emilio Zappa、Miranda Holmes-Cerfon 和 Jonathan Goodman。《流形上的蒙特卡洛：采样密度和函数积分》。《纯粹与应用数学通讯》，71(12):2609–2647，2018年。