StruMNet: 一种融合神经网络和偏微分方程的结构网格生成方法

彭嘉明[1,2)], 陈新海[1,2)*], 刘杰[1,2)]

[1)] (高端装备数字化软件湖南省重点实验室 长沙 410073)
[2)] (国防科技大学并行与分布计算全国重点实验室 长沙 410073)
(chenxinhai16@nudt.edu.cn)

摘 要: 网格生成是计算流体力学前处理环节的关键, 在整个计算流体力学数值模拟中扮演着十分重要的角色. 为了进一步提升网格生成效率及智能化水平, 提出了一种融合神经网络和偏微分方程的结构网格生成方法 StruMNet. 该方法以计算域的边界以及内部的采样点为输入, 建立结合偏微分方程的神经网络模型用于网格生成, 控制方程和几何边界约束嵌入在损失函数中作为模型的优化目标; 引入损失函数自适应权重策略和点加权机制来提升网格质量及在复杂边界处的拟合能力; 训练后的 StruMNet 能够根据给定的边界形状预测网格点. 分别采用 3 种形状进行实验, 使用平均测试集损失、平均最大夹角、平均最小夹角并角波动最大值和角波动最小值引入判断生成的网格质量. 实验结果表明, StruMNet 在不同几何上的网格生成质量优于物理信息神经网络方法和 MGNet 方法, 在网格生成效率方面相较于传统方法提高 3 个数量级, 最终实现结构网格的快速生成.

关键词: 网格生成; 计算流体力学; 神经网络; 偏微分方程; 结构网格
中图法分类号: TP391 **DOI:** 10.3724/SP.J.1089.2023-00696

StruMNet: A Structural Mesh Generation Method Integrating Neural Networks and Partial Differential Equations

Peng Jiaming[1,2)], Chen Xinhai[1,2)*] , and Liu Jie[1,2)]

[1)] (*Hunan Laboratory of Digitizing Software for Frontier Equipment, Changsha* 410073)
[2)] (*Science and Technology on Parallel and Distributed Processing Laboratory, National University of Defense Technology, Changsha* 410073)

Abstract: Mesh generation is a crucial step in the preprocessing phase of computational fluid dynamics (CFD) and plays a vital role in the overall CFD numerical simulation process. To further enhance the efficiency and intelligence of mesh generation, we propose StruMNet, a structured mesh generation method that integrates neural networks with partial differential equations. This method takes the boundaries of the computational domain and internal sampling points as inputs to establish a neural network model combined with partial differential equations for mesh generation. The loss function incorporates both control equation loss and geometric boundary constraint loss as optimization objectives. Additionally, the method introduces an adaptive weighting strategy for the loss function and a point-weighting mechanism to enhance mesh quality and fitting capabilities, especially at complex boundaries. Once trained, StruMNet can predict mesh points based on given boundary shapes. Experiments are conducted on three different shapes, and the quality of the generated meshes is evaluated using the average test loss, the average maximum angle, the average minimum angle, as well as the maximum and minimum angle fluctuations. Experimental results show that StruMNet outperforms the phys-

收稿日期: 2023-10-24; 修回日期: 2024-09-01. **基金项目**: 国家自然科学基金(12402349); 湖南省自然科学基金(2024JJ6468); 国防科技大学青年自主创新科学基金(ZK2023-11); 国家重点研发计划(2021YFB0300101). **彭嘉明**(2000—), 女, 博士研究生, 主要研究方向为智能网格生成; **陈新海**(1995—), 男, 博士, 助理研究员, 论文通信作者, 主要研究方向为智能科学计算; **刘杰**(1969—) , 男, 博士, 教授, 博士生导师, 主要研究方向为高性能计算和数值分析.

ics-informed neural networks method and the MGNet method in terms of mesh quality across various geometries and improves mesh generation efficiency by three orders of magnitude compared to traditional methods, ultimately enabling the rapid generation of structured meshes.

Key words: mesh generation; computational fluid dynamics; neural network; partial differential equation; structured mesh

计算流体力学(computational fluid dynamics, CFD)是研究物理现象和探索流体流动规律的重要方法, 在航空航天、汽车工程、能源和环境科学等科学与工程领域都发挥着十分重要的作用[1-2]. CFD 数值模拟是研究物理现象和规律的重要手段, 数值模拟过程通常可以分为前处理、计算和后处理[3] 3 个步骤. 其中, 网格生成是 CFD 数值模拟得以进行的必要“前处理”工作. NASA 的研究报告表明, 生成高质量网格大约要占据整个数值模拟时间的 60%[4]. 因此, 提高网格生成效率是加速 CFD 数值模拟的重要环节. 结构网格计算时具有良好的稳定性和精度, 不需要额外存储网格单元实体的相互连接关系, 数据的访存效率较高. CFD 研究中常用的结构网格生成方法有代数网格生成方法[5]和微分方程法[6-8]. 代数网格生成方法采用代数关系式实现贴体坐标转换, 生成贴体网格; 该方法具有计算量小、效率高等优点, 但难以适应复杂的物理区域形状, 在几何形状较为复杂的区域中得到的坐标面可能会发生退化. 针对上述问题, Xiao 等[9]提出一种基于边界元法的自动域划分方法, 首先根据交叉场信息对外形进行域分解, 然后在分解后的各个域分别使用无限插值法(transfinite interpolation, TFI)[10]生成网格, 有助于进一步提升网格质量. 微分方程法将物理区域的网格边界作为方程的边界条件, 然后求解偏微分方程(partial differential equation, PDE), 得到的结果就是内部网格点坐标[6]; 该方法生成的网格正交性和光滑性较好, 但计算开销较大.

为了解决传统网格生成方法自动化、智能化水平不足的问题, 一些学者将人工智能技术引入网格生成领域. Zhang 等[11]提出 MeshingNet, 该模型可以学习初始(粗糙)均匀网格上的后验误差估计, 并预测非均匀网格密度以进行细化; Lu 等[12]将人工神经网络应用于适合黏性流动模拟的混合网格生成, 通过训练 2 个神经网络, 分别预测新点的前进方向和控制生成网格大小; ISpliter[13]采用神经网络选择最佳分割线并不断地划分几何形状, 直到所有区域都变成三角形: 然而上述方法依赖于大量数据集, 训练成本高. IGA-Reuse-NET[14]结合同构几何分析, 可以在拓扑一致的几何模型上快速进行数值模拟, 并在复杂几何体上求解 Poisson 方程, 其本质上是有监督、数据驱动的方法, 与本文工作具有一定的相似性.

Chen 等[15]首次将物理信息神经网络(physics-informed neural networks, PINNs)[16]应用于结构网格生成任务中, 提出智能结构化网格生成的方法 MGNet, 其是一种无监督的网格生成方法, 虽然减少了对训练数据的依赖, 但也存在着网格质量不佳、训练开销较大的问题. 为了进一步提高使用人工智能方法生成结构网格的质量并减少训练中的开销, 本文提出了一种融合神经网络和 PDE 的结构网格生成方法 StruMNet, 其核心是在神经网络的损失函数中嵌入与网格生成相关的 PDE, 将网格生成问题转化为无监督的优化问题. 该方法是一种无监督的网格生成方法, 除了给定的边界信息, 无需预先准备其他数据; 引入损失函数自适应权重策略, 根据每个任务的损失大小动态更新权重, 以避免单一任务主导网络参数更新; 针对模型边界拟合不佳的情况, 引入边界点加权机制, 提升神经模型的学习能力. 在 3 种外形上的实验结果表明, StruMNet 可以快速地生成高质量网格.

1 StruMNet

随着人工智能技术和方法的发展, 其在物理领域的应用也日益广泛, PINN 方法的出现为解决 PDE 问题提供了新的思路. PINN 的核心是将控制方程以及初始/边界条件作为惩罚项嵌入到神经网络的损失函数中, 利用各项残差之和构造损失函数; 其本质上是一种无网格技术, 通过将直接求解控制方程的问题转化为损失函数优化问题来逼近 PDE 的解. PINN 的损失函数由 PDE 的控制方程、

网络输出与初始条件和边界条件的标签数据的残差 3 部分组成, 即

$$L=L_{\mathrm{u}}+L_{\mathrm{B}}+L_{\mathrm{I}} \tag{1}$$

其中, L_{u} 是控制方程的残差; L_{B} 是边界条件的残差; L_{I} 是初始条件的残差.

MGNet 利用 PINN 在选定的变换域求解椭圆型、抛物型或双曲型微分方程, 方程求解的结果即是生成的网格点. 该方法首先通过边界拟合获得物理边界曲线的连续表达形式, 然后将得到的曲线函数和计算域内的采样点作为输入, 输入到 MGNet 的网络结构中; 其中, 边界点遵循类似于 PINN 中的边界项损失, 计算域内的采样点遵循用于指导网格生成的 PDE 的损失. 在训练过程中, MGNet 学习高维空间中的网格划分规则, 最终得到计算域与物理域之间有效的映射关系. 然而, 该方法使用 2 个子网络进行训练和网格生成, 训练开销较大, 且对于复杂边界处的信息捕捉能力不强, 生成的网格质量有待提高. 针对上述缺陷, StruMNet 通过使用单个神经网络进行训练和网格生成, 降低了训练开销; 其次, 引入损失函数自适应权重策略动态地平衡不同损失项和边界点加权机制, 增强模型的学习能力, 以更好地捕捉复杂边界处的信息, 提高边界处的网格质量.

图 1 所示为 StruMNet 的结构. 其中, 输入是从计算域的内部和边界上采样的训练数据, 输入层是数据点对应的二维点序列 $P=\left\{\left(\xi_n,\eta_n\right)\right\}_{n=1}^{n=N}$, N 表示样本点的个数; 隐藏层由多层全连接的神经元组成; 输出层有 2 个神经元, 负责输出预测的网格的 x, y 坐标; 自动微分层负责计算下一步控制方程中需要使用到的偏微分量; 损失函数由控制方程的残差和边界拟合误差分别乘以自适应权重 $\frac{1}{2a_1^2}$ 和 $\frac{1}{2a_2^2}$ (图中加权损失项的 $T=2$), 并加上正则化项得到. 随后, 损失函数会通过反向传播更新网络的参数 θ.

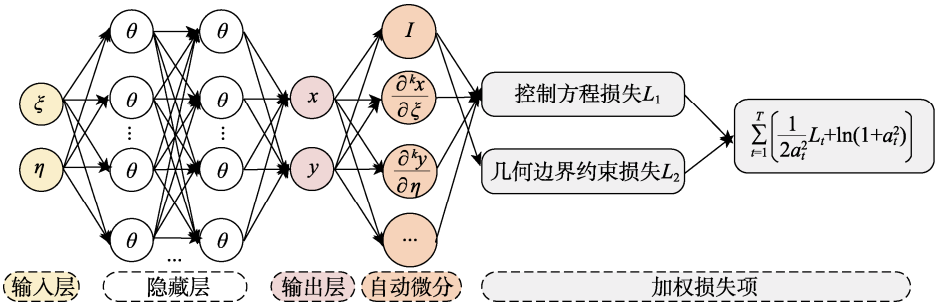


图 1　StruMNet 的结构

构造损失函数时, 本文将用于网格生成的椭圆型 PDE 的方程式嵌入到神经网络的损失函数中, 指导网格生成. 椭圆型 PDE 的计算公式为

$$\begin{cases}\alpha x_{\xi\xi}-2\beta x_{\xi\eta}+\gamma x_{\eta\eta}=0\\ \alpha y_{\xi\xi}-2\beta y_{\xi\eta}+\gamma y_{\eta\eta}=0\\ \alpha=x_\eta^2+y_\eta^2,\ \beta=x_\xi x_\eta+y_\xi y_\eta,\ \gamma=x_\xi^2+y_\xi^2\end{cases} \tag{2}$$

其中, x_ξ 表示 x 对 ξ 求一阶偏导, x_η 表示 x 对 η 求一阶偏导, $x_{\xi\xi}$ 表示 x 对 ξ 求二阶偏导, $x_{\xi\eta}$ 表示 x 先对 ξ 求一阶偏导再对 η 求一阶偏导, 以此类推; α 表示 $x_\eta^2+y_\eta^2$, β 表示 $x_\xi x_\eta+y_\xi y_\eta$, γ 表示 $x_\xi^2+y_\xi^2$. 最终构建的损失函数 L 由控制方程损失项 L_1 和几何边界约束项 L_2 组成, 即

$$L(x,y)=L_1+L_2 \tag{3}$$

其中,

$$L_1=\frac{1}{N_{\mathrm{r}}}\sum_{n=1}^{N_{\mathrm{r}}}\left(\left|\alpha x_{\xi\xi}-2\beta x_{\xi\eta}+\gamma x_{\eta\eta}\right|^2+\left|\alpha x_{\xi\xi}-2\beta x_{\xi\eta}+\gamma x_{\eta\eta}\right|^2\right);$$

$$L_2=\frac{1}{N_{\mathrm{b}}}\sum_{n=1}^{N_{\mathrm{b}}}w_n\left(\left|(x_n,y_n)-f(\xi_n,\eta_n)\right|^2\right).$$

N_{r} 和 N_{b} 分别表示控制方程训练点和几何边界训练点的数量; $\left(x_n,y_n\right)$ 表示第 n 个真实边界点坐标; $f(\xi_n,\eta_n)$ 表示输入为边界点 (ξ_n,η_n) 对应的预测值.

为了解决生成网格边界处拟合不佳的问题, StruMNet 引入边界点权重机制 w_n, 将预测点和真实点之间的距离作为数据点的权重, 通过对边界处拟合较差的点赋予更大的权重, 使得该方法能够更准确地学习和拟合边界条件, 改善边界处网格的质量. 边界点的权重的计算公式为

$$w_n=\begin{cases}d_n, & \text{if } e\leqslant 3\,000\\ d_n+1, & \text{otherwise}\end{cases} \tag{4}$$

其中，$d_n=\sqrt{\left(y_n^{\text{true}}-y_n^{\text{predict}}\right)^2+\left(x_n^{\text{true}}-x_n^{\text{predict}}\right)^2}$. x_n^{predict} 和 y_n^{predict} 表示第 n 个边界点的预测输出结果，x_n^{true} 和 y_n^{true} 表示第 n 个真实边界点的坐标，e 为当前训练轮数. 式(4)中，当 $e\leqslant 3\,000$ 时，为了确保模型的稳定收敛，使用较小的权重；当 $e>3\,000$ 时，模型的收敛已经较为稳定，学习也达到了瓶颈，此时，为了让模型能够更精准地学习边界信息，增大了对边界点的权重，通过将权重增加 1，进一步提高 StruMNet 对边界形状的学习能力，从而提升了边界处网格的质量. 但是，如果提前使用较大的权重，可能会使得 StruMNet 无法较好地学习到椭圆型 PDE 的物理信息，反而无法加速收敛.

此外在训练过程中，StruMNet 几何边界约束项的损失远大于控制方程项的损失，存在一些极端情况：当边界约束任务的损失非常大、控制方程的损失非常小时，多目标优化会退化为单目标优化，使得网络参数的更新被边界约束损失项所主导，忽略了对控制方程任务的优化. 针对上述问题，StruMNet 引入损失函数自适应权重策略，以动态地平衡控制方程和边界拟合 2 部分的损失；同时为它们分配不同的权重，平衡它们对总损失函数的影响. 任务权重可以根据每个任务的损失大小进行动态调整和更新，避免单一任务主导网络参数更新的情况. 损失函数的计算公式为

$$L(\theta)=\sum_{t=1}^{T}\left(\frac{1}{2a_t^2}L_t+\ln\left(1+a_t^2\right)\right) \tag{5}$$

其中，L_t 表示与任务 t 相关的损失；a_t 表示每个任务的噪声参数(任务权重). 式(5)中使用系数 $1/2a_t^2$ 平衡各项 L_t. 例如，如果一个任务很容易优化，则可以学习到的 a 就较小，使模型不会忽略这个简单的任务；$\ln\left(1+a_t^2\right)$ 可以被视为一个正则化项，防止学习到的 a 过大.

引入损失函数自适应权重策略以及边界点加权机制后，式(3)可改写为

$$L(x,y)=\left(\frac{1}{2a_1^2}L_1+\ln\left(1+a_1^2\right)\right)+\left(\frac{1}{2a_2^2}L_2+\ln\left(1+a_2^2\right)\right) \tag{6}$$

其中，L_1 和 L_2 同式(3)中的定义. StruMNet 最终采用式(6)作为损失函数指导 StruMNet 的训练.

2 实验及结果分析

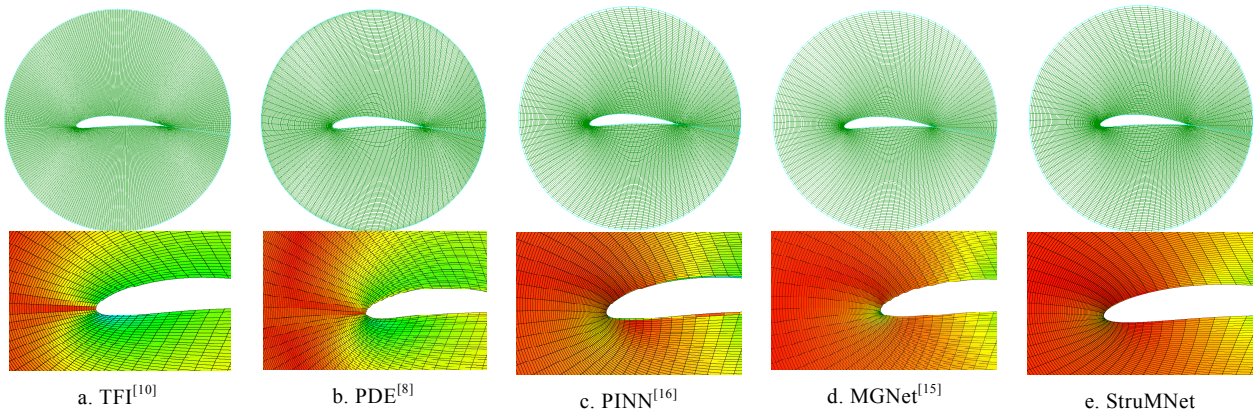
为了验证 StruMNet 生成网格的质量，在 3 种形状上进行实验，将 3 种形状称为外形 1~外形 3. StruMNet 的隐藏层设置为 4 层，每层包含 45 个神经元，除输出层外均使用 tanh 激活函数. 将 StruMNet 与 PDE[8], TFI[10], PINN[16]和 MGNet[15]方法进行对比实验，为了方便起见，将 PINN[16]和 MGNet[15]方法的参数配置与 StruMNet 相同. 在网络训练的过程中，StruMNet 使用 Xavier 正态分布进行初始化，学习率设置为 0.001；任务权重 a_i 的初始值设为 1，学习率为 0.000 1. 训练平台为 RTX 2080Ti GPU, 16 GB 内存.

除了使用平均测试集损失、平均最大夹角、平均最小夹角判断生成的网格质量外，本文引入角波动最大值和角波动最小值这 2 种新的评价指标辅助判断. 角波动最大值和角波动最小值是分别重复 10 次实验，选择在测试集上损失最小的模型进行网格生成，并将对比生成的网格之间的最大夹角的角度差记为角波动最大值；同理，可以得到角波动最小值. 这 2 个指标在一定程度上反映了神经网络预测的稳定性和鲁棒性. Jacobian 指标度量生成的网格是否能够准确地捕捉输入域的几何形状，在质量良好的网格中，该指标应均大于 0.

图 2 所示为在外形 1 上 5 种方法生成结构网格质量可视化对比结果. 其中，每个分图中下方的子图展示了最小夹角的可视化效果，红色区域表示网格正交性良好，蓝色(或绿色)区域表示正交性较差(下文同).

表 1 所示为在外形 1 上 5 种方法生成结构网格的数值结果对比，并通过平均测试集损失来辅助判断生成网格的质量. 可以看出，使用 StruMNet 生成网格的损失值为 3.41×10^{-7}，比 PINN[16]方法低了一个数量级；StruMNet 的网格夹角波动为 1°和 2°，均小于 PINN[16]方法的 5°和 8°，略低于 MGNet[15]方法，表明 StruMNet 的整体收敛比较平稳；StruMNet 的平均最大夹角为 103.616 75°，低于 PINN[16], PDE[8]和 TFI[10]方法，平均最小夹角为 76.317 13°，高于 PINN[16], PDE[8]和 TFI[10]方法，表明 StruMNet 生成网格的正交性更好.

图 3 所示为在外形 2 上 5 种方法生成结构网格结果对比. 表 2 所示为在外形 2 上 5 种方法生成结构网格的数值结果对比，并通过平均测试集损失来辅助判断生成网格的质量. 使用 StruMNet 生成



a. TFI[10]　b. PDE[8]　c. PINN[16]　d. MGNet[15]　e. StruMNet

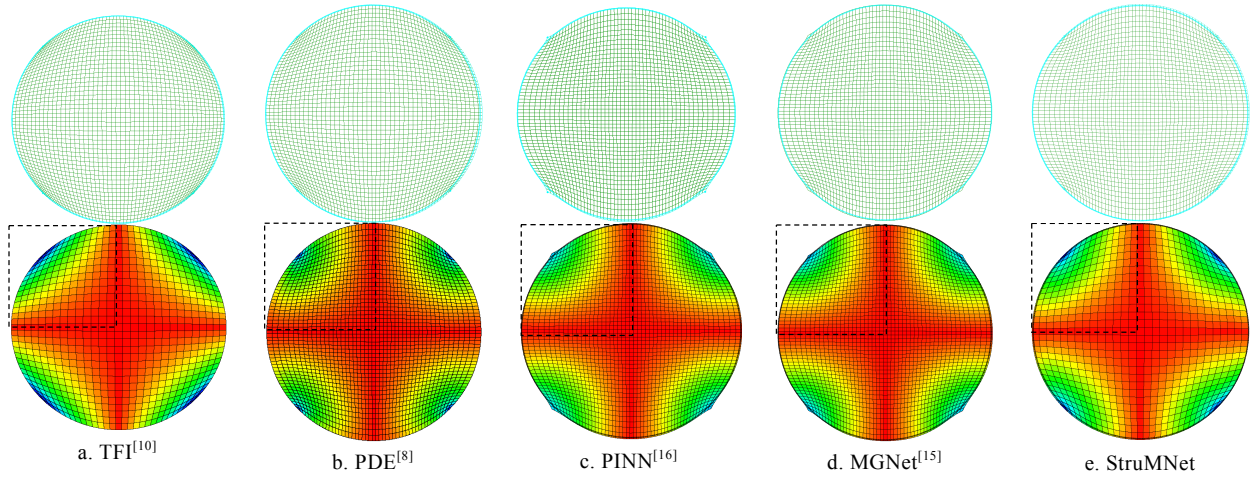
图 2　在外形 1 上 5 种方法生成结构网格结果对比

表 1　在外形 1 上 5 种方法生成结构网格数值结果对比

方法	平均测试集损失	平均夹角/(°)		角波动/(°)	
		最大	最小	最大	最小
TFI[10]		105.532 26	75.714 81		
PDE[8]		104.743 72	75.414 25		
PINN[16]	5.07×10^{-6}	104.109 44	75.867 99	5.0	8.0
MGNet[15]	8.02×10^{-7}	103.911 82	76.035 97	1.4	2.0
StruMNet	$\mathbf{3.41\times10^{-7}}$	**103.616 75**	**76.317 13**	**1.0**	**2.0**

注. 粗体表示最优值.

网格的损失值为 1.3×10^{-5}, 与 PINN[16]和 MGNet[15]方法相差不大; Jacobian 用于检测生成的网格是否出现跨越边界负面积(即 Jacobian<0)的情况, StruMNet 中有 4 处 Jacobian<0, 小于 PINN[16]方法中的 15 处和 MGNet[15]方法中的 12 处; StruMNet 生成网格的平均最大夹角为 103.024°, 与 PINN[16]方法生成网格的平均最大夹角 103.022°接近, StruMNet 生成网格的平均最小夹角为 77.32°, 也与 MGNet[15]方法生成网格的平均最小夹角 77.35°接近, 表明 StruMNet 可以在保证网格的正交性的同时减小 Jacobian<0 的网格数量, 提高网格质量; 在平均最大夹角和平均最小夹角上, StruMNet 生成的网格正交性优于 TFI[10]和 PDE[8]方法生成网格的正交性.



a. TFI[10]　b. PDE[8]　c. PINN[16]　d. MGNet[15]　e. StruMNet

图 3　在外形 2 上 5 种方法生成结构网格结果对比

表 2　在外形 2 上 5 种方法生成结构网格数值结果对比

方法	平均测试集损失	平均夹角/(°)		Jacobian<0
		最大	最小	
TFI[10]		110.375 75	69.984 34	0
PDE[8]		107.565 87	72.684 01	0
PINN[16]	8.87×10^{-5}	**103.022 75**	77.339 43	15
MGNet[15]	5.26×10^{-5}	103.996 67	**77.353 09**	12
StruMNet	$\mathbf{1.32\times10^{-5}}$	103.024 07	77.328 03	**4**

注. 粗体表示最优值.

为了验证 StruMNet 适用于复杂几何的结构网格生成, 首先使用区域分解(网格分块)方法[9]将外形分解为不同区域, 然后在分解后的不同区域内生成结构网格.

图 4 所示为在复杂外形(即外形 3)上, 使用 5 种方法生成结构网格的结果对比, 其中, 图 4a 用数字代表不同的区域. 其中, 与 TFI[10]和 PDE[8]方法相比, StruMNet 在区域 2 和区域 6 上的网格质量

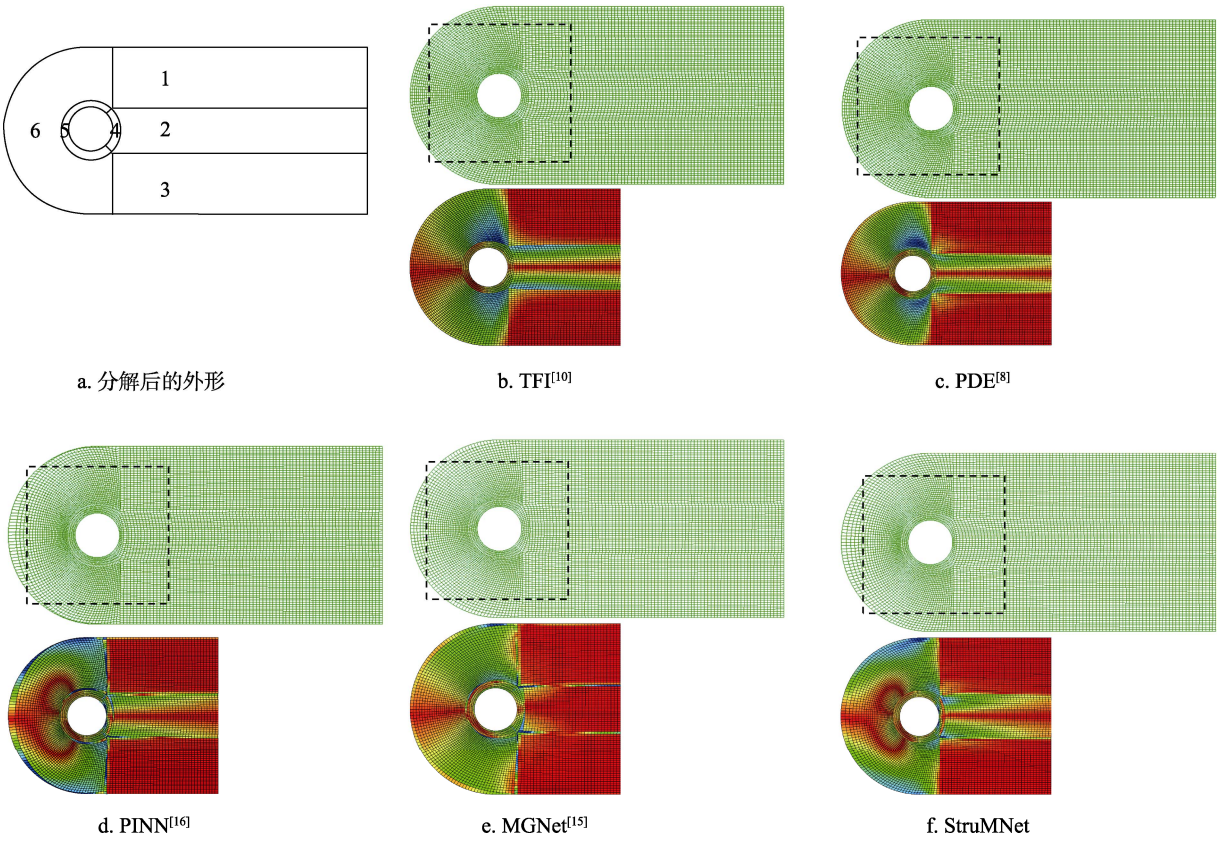


图 4 在区域分解后的外形(外形 3)上 5 种生成结构网格结果对比

有所提高; PINN[16]和 MGNet[15]方法在区域分解后的外形边界处(如区域 2 与区域 1 和区域 3 的相接处, 以及区域 5 和区域 6 的相接处)生成的网格质量不佳, 难以有效地控制边界处的网格密度和质量; StruMNet 不仅能够实现网格的均匀过渡, 且可以提升网格质量.

在区域分解后的外形上, 5 种方法生成结构网格的数值结果对比如表 3 所示. 可以看出, TFI[10]和 PDE[8]方法的正交性略差于 StruMNet 方法; PINN[16]和 MGNet[15]方法生成的网格在靠近边界处的网格质量控制存在不足; 相比之下, StruMNet 平均最大夹角和平均最小夹角结果最好, 网格正交性优于其他方法, 且该方法生成的网格没有出现 Jacobian<0 的情况, 也没有生成超出边界的网格单元, 表明 StruMNet 能够准确地捕捉输入区域的几何形状, 生成高质量的网格.

表 4 所示为在 3 种不同外形上 5 种方法生成结构网格的时间开销对比. 可以看出, StruMNet 在网格质量和生成时间方面取得了有效的平衡(这里是串行计算时间). 需要注意的是, StruMNet 的推理过程可以并行化, 与 PDE[8]方法相比, 可以实现 3 个数量级的速度提升. 实验结果表明, StruMNet 能够在保持与 PDE[8]方法生成相当的网格质量的同时, 显著地缩短了网格生成时间.

表 3 在外形 3 上 5 种方法生成结构网格数值结果对比

方法	平均测试集损失	平均夹角/(°)		Jacobian<0
		最大	最小	
TFI[10]		95.834 92	84.289 12	0
PDE[8]		95.102 73	84.920 96	0
PINN[16]	1.03×10^{-2}	94.990 16	85.076 80	2
MGNet[15]	7.09×10^{-3}	94.491 68	85.565 73	0
StruMNet	**1.17×10^{-3}**	**94.384 09**	**85.638 24**	**0**

注. 粗体表示最优值.

表 4 在 3 种外形上 5 种方法生成结构网格时间开销对比 s

方法	外形 1	外形 2	外形 3
TFI[10]	1.445 63	0.391 46	0.194 38
PDE[8]	441.761 09	120.670 30	141.756 00
PINN[16]	0.018 59	0.006 89	0.032 02
MGNet[15]	0.016 55	0.006 61	0.025 96
StruMNet	**0.014 86**	**0.005 92**	**0.021 99**

注. 粗体表示最优值.

3 结　语

本文提出一种基于神经网络的结构网格生成方法 StruMNet, 利用物理信息神经网络学习高质量网格划分规则, 为智能网格生成提供了一种全新的思路. 该方法引入自适应权重策略, 动态地平衡控制方程和边界项之间的损失; 为了提升网络在边界处的拟合能力, 引入边界点加权机制. 在 3 种外形上的实验结果表明, StruMNet 可以快速地生成正交性优于 TFI[10]和 PDE[8]方法的网格; 与通用的 PINN[16]方法的生成网格相比, StruMNet 收敛稳定, 鲁棒性更强. 然而, 本文方法尚未扩展至三维结构网格生成, 这将是下一步的工作, 以验证其在更复杂三维几何下的适用性.

参考文献(References):

[1] Chen X H, Liu J, Li S G, *et al*. TAMM: a new topology-aware mapping method for parallel applications on the Tianhe-2A supercomputer[C] //Proceedings of the 18th International Conference on Algorithms and Architectures for Parallel Processing. Heidelberg: Springer, 2018: 242-256

[2] Chang-Hoi A, Sang-Soo L, Hyuek-Jae L, *et al*. A self-organizing neural network approach for automatic mesh generation[J]. IEEE Transactions on Magnetics, 1991, 27(5): 4201-4204

[3] Jeong W, Seong J. Comparison of effects on technical variances of computational fluid dynamics (CFD) software based on finite element and finite volume methods[J]. International Journal of Mechanical Sciences, 2014, 78: 19-26

[4] Slotnick J P, Khodadoust A, Alonso J, *et al*. CFD vision 2030 study: a path to revolutionary computational aerosciences[R]. 2014

[5] Bern M W, Plassmann P E. Mesh Generation[M] //Sack J R, Urrutia J. Handbook of Computational Geometry. Amsterdam: North Holland, 2000: 291-332

[6] Liseikin V D. A computational differential geometry approach to grid generation[M]. Berlin: Springer, 2007

[7] Cortis C M, Friesner R A. An automatic three‐dimensional finite element mesh generation system for the Poisson-Boltzmann equation[J]. Journal of Computational Chemistry, 1997, 18(13): 1570-1590

[8] Thompson J F, Warsi Z U A, Mastin C W. Numerical grid generation: foundations and applications[M]. New York: Elsevier Science Ltd, 1985

[9] Xiao Z F, He S P, Xu G, *et al*. A boundary element-based automatic domain partitioning approach for semi-structured quad mesh generation[J]. Engineering Analysis with Boundary Elements, 2020, 113: 133-144

[10] Gordon W J, Hall C A. Construction of curvilinear co-ordinate systems and applications to mesh generation[J]. International Journal for Numerical Methods in Engineering, 1973, 7(4): 461-477

[11] Zhang Z Y, Wang Y X, Jimack P K, *et al*. MeshingNet: a new mesh generation method based on deep learning[C] //Proceedings of the 20th International Conference on Computational Science. Heidelberg: Springer, 2020: 186-198

[12] Lu P, Wang N H, Chang X H, *et al*. An automatic isotropic/anisotropic hybrid grid generation technique for viscous flow simulations based on an artificial neural network[J]. Chinese Journal of Aeronautics, 2022, 35(4): 102-117

[13] Liu Z S, Chen S Z, Gao X, *et al*. ISpliter: an intelligent and automatic surface mesh generator using neural networks and splitting lines[J]. Advances in Aerodynamics, 2023, 5(1): Article No.18

[14] Wang D D, Xu J L, Gao F, *et al*. IGA-Reuse-NET: a deep-learning-based isogeometric analysis-reuse approach with topology-consistent parameterization[J]. Computer Aided Geometric Design, 2022, 95: Article No.102087

[15] Chen X H, Li T J, Wan Q, *et al*. MGNet: a novel differential mesh generation method based on unsupervised neural networks[J]. Engineering with Computers, 2022, 38(5): 4409-4421

[16] Raissi M, Perdikaris P, Karniadakis G E. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations[J]. Journal of Computational Physics, 2019, 378: 686-707