Nitsche 稳定化对有限单元法几何多重网格的影响

*

S. Saberi † G. Meschke ‡ A. Vogel †

摘要

浸入式有限元方法旨在避免传统有限元分析中成本高昂的网格生成。然而，这类方法所得线性方程组的数值病态性给迭代求解器带来了挑战。本文聚焦于采用自适应求积、自适应网格细化（AMR）以及 Nitsche方法弱施加边界条件的有限胞元法（FCM）。针对离散化问题，本文采用自适应几何多重网格求解器。我们研究了 Nitsche方法中与网格相关的稳定化参数对几何多重网格求解器性能的影响，以及它对整体多层级设置的影响。我们使用基于广义特征值问题的全局估计和局部估计来选择稳定化参数。研究发现，求解器的收敛率显著受稳定化参数、估计方法的选择以及稳定化参数在多层配置中的处理方式影响。研究还发现，在每个网格上计算的局部估计是一种稳健的方法，并能使几何多重网格求解器快速收敛。

关键词 自适应几何多重网格 · Nitsche方法 · 浸入式有限元 · 有限胞元法 · 弱边界条件

1 引言

在复杂区域上对偏微分方程进行数值逼近是一项耗费大量人力的工作，其中，生成适当的边界拟合计算网格可能占据模拟工作流程中的显著部分。这促使人们开发了一类先进的离散化方法，统称为浸入式或非贴体有限元方法；扩展有限元法（XFEM）[1]、有限单元法（FCM）[2,3]和 CutFEM[4]方法均属于此类。每种方法的分类取决于区域积分和边界条件施加所采用的具体技术。拉格朗日乘子、[5–10]罚方法[11,12]、Nitsche方法[13–20]、最小二乘 Nitsche 方法[21,22]等，都是常用的边界条件施加技术。罚方法和 Nitsche方法都以弱形式施加本质边界条件，即修改变分形式以纳入本质边界条件，而非对状态变量引入显式约束。此外，这两种方法都不引入额外的未知量，并能保持矩阵的对称性、正定性和带状结构。与 Nitsche方法相比，罚方法在变分意义上并不一致[5,18]，还可能导致系统严重病态。有限

*本研究得到德国研究联合会（Deutsche Forschungsgemeinschaft，DFG）资助，资助项目为协作研究中心 SFB 837《机械化隧道施工中的交互建模》，项目编号 77309832。

†波鸿鲁尔大学高性能计算中心，Universit¨atsstr. 150，44801 波鸿，德国
‡波鸿鲁尔大学结构力学研究所，Universit¨atsstr. 150，44801 波鸿，德国

单元法采用自适应求积进行区域积分，并采用弱施加方法施加本质边界条件，[2,3,23]而在有限单元法（FCM）中，通常更倾向于采用 Nitsche方法。[24–26]

从计算角度来看，采用这类数值方法的主要障碍之一是求解离散化问题，这往往限制了它们在大规模问题中的应用。虽然直接求解器具有鲁棒性，且能以机器精度求解系统，但其复杂度通常并非最优，因此在求解大型系统时成本极高[27]。另一方面，迭代求解器具有更优的复杂度，并且能在并行计算机上实现更高的并发性；然而，它们的收敛通常高度依赖系统矩阵的谱，因此存在网格依赖性。[27] 多重网格方法已成功用于消除不同类别偏微分方程有限元离散中的迭代求解器的网格依赖性。[28] 在非拟合有限元方法中，众所周知，较小的切割比例会导致系统矩阵严重病态，从而限制迭代求解器的适用性。近期已有研究探讨了采用Nitsche方法的几何多重网格求解器在三种CutFEM形式[29] 和XFEM[30] 中的应用。近期也有研究探讨了采用罚函数形式[31] 以及Nitsche方法的有限胞元法中的几何多重网格。[32,33]

本文采用泊松问题的有限元胞形式，其中分别使用自适应求积和 Nitsche方法进行区域积分和边界条件的施加。我们为求解由此得到的系统构建了几何多重网格求解器，并研究 Nitsche方法中的稳定化参数对有限元胞形式几何多重网格求解器性能的影响。我们考察了求解器对稳定参数变化的敏感性，并探讨了估算适当稳定参数的可能方法。此外，我们还讨论了稳定参数依赖于网格这一特性对多层方法的影响。

本文其余部分安排如下。第 2 节详细介绍模型问题的数值形式、本质边界条件的弱施加方式、有限元胞形式以及几何多重网格求解器。第 3 节通过若干数值基准算例研究几何多重网格求解器，并讨论所得结果。最后，第 4 节给出结论。

2 数值方法

2.1 模型问题

作为本文的模型问题，我们采用泊松方程作为一种典型的椭圆型偏微分方程，其强形式可写为

$$\begin{aligned} -\nabla^2 u &= f \qquad \text{in } \Omega, \\ u &= g \qquad \text{on } \Gamma_D, \\ \nabla u \cdot \boldsymbol{n} &= t \qquad \text{on } \Gamma_N, \end{aligned} \tag{1}$$

其中，u 为标量解变量， Ω 为空间区域， $\Gamma = \Gamma_D \cup \Gamma_N$ 为区域的边界， Γ_D 和 Γ_N 分别为边界上的狄利克雷部分和诺伊曼部分，f 为体力函数，n 为边界的单位外法向量。g 和 t 分别是定义在 Γ_D 和 Γ_N上的给定函数。将强形式乘以适当的检验函数，在区域上积分并应用格林定理，可将边界贴合弱形式写为:

求 $u \in V$，使得对所有 $v \in V_0$

$$\int_\Omega \nabla v \cdot \nabla u \, d\boldsymbol{x} - \int_{\Gamma_D} v(\nabla u \cdot \boldsymbol{n}) \, d\boldsymbol{s} = \int_\Omega v f \, d\boldsymbol{x} + \int_{\Gamma_N} v t \, d\boldsymbol{s}, \tag{2}$$

其中 v 为检验函数，且

$$
\begin{aligned}
V &:= \{u \in H^1(\Omega) \mid u = g \text{ on } \Gamma_D\}, \\
V_0 &:= \{v \in H^1(\Omega) \mid v = 0 \text{ on } \Gamma_D\},
\end{aligned}
\tag{3}
$$

其中，H^1 是 Sobolov空间。我们注意到，在边界贴合情形下，方程（2）中 Γ_D 上的项为零，因为本质边界条件已包含在空间V中。

2.2 浸入式有限元方法

在经典有限元方法中，物理域 Ω 采用边界贴合剖分进行近似。然而，鉴于 Ω可能具有任意复杂的形状，而这一点直接反映在生成该域边界贴合剖分所需的工作量上，浸入式有限元方法将 Ω 嵌入嵌入域Ω_e中，如图1所示。因此，计算域扩展至包含虚构部分 $\Omega_e \setminus \Omega$。将 Ω 嵌入 Ω_e 需要对标准有限元弱形式作两项总体修改。一方面，弱形式必须还原物理域 Ω ，具体做法将在下文详细说明，即使用惩罚因子。另一方面，与空间V包含本质边界条件的边界贴合情形不同，由于物理边界 Ω不一定与计算域 Ω_e的边界贴合，浸入式方法必须修改方程2，以弱形式施加本质边界条件。我们先讨论后者，即本质边界条件的弱施加；本文采用对称 Nitsche 方法来实现。

应用Nitsche方法，方程（2）中的弱形式可转化为

$$
\begin{aligned}
\int_\Omega \nabla v \cdot \nabla u \, d\boldsymbol{x} - \int_{\Gamma_D} v(\nabla u \cdot \boldsymbol{n}) \, d\boldsymbol{s} - \int_{\Gamma_D} (u-g)(\nabla v \cdot \boldsymbol{n}) \, d\boldsymbol{s} + \int_{\Gamma_D} \lambda v(u-g) \, d\boldsymbol{s} = \\
\int_\Omega vf \, d\boldsymbol{x} + \int_{\Gamma_N} vt \, d\boldsymbol{s},
\end{aligned}
\tag{4}
$$

其中， λ是Nitsche方法中的标量稳定化参数。方程（4）左侧的第三项是对称一致性项，用于确保保留原弱形式的对称性。方程（4）左侧的第四项是稳定化项，用于确保满足边界条件，并在 λ足够大时保证该形式的稳定性。

下一步将方程（4）中的弱形式扩展到嵌入域，如下所示：

$$
\begin{aligned}
\int_{\Omega_e} \alpha \nabla v \cdot \nabla u \, d\boldsymbol{x} - \int_{\Gamma_D} v(\nabla u \cdot \boldsymbol{n}) \, d\boldsymbol{s} - \int_{\Gamma_D} (u-g)(\nabla v \cdot \boldsymbol{n}) \, d\boldsymbol{s} + \int_{\Gamma_D} \lambda v(u-g) \, d\boldsymbol{s} = \\
\int_{\Omega_e} \alpha vf \, d\boldsymbol{x} + \int_{\Gamma_N} vt \, d\boldsymbol{s},
\end{aligned}
\tag{5}
$$

其中

$$
\begin{cases}
\alpha = 1, & \text{in } \Omega, \\
\alpha = 0, & \text{in } \Omega_e \setminus \Omega,
\end{cases}
\tag{6}
$$

其中， α 是标量惩罚因子。通过对嵌入域中位于物理域之外的部分施加惩罚，可基本恢复物理域。需要注意的是，实际应用中，在虚构部分 $\Omega_e \setminus \Omega$中会使用一个很小的正非零值 $\alpha \ll 1$ ，而非零值，以避免严重的数值病态。

令 $\mathrm{T}_h := \{\mathrm{K}_i\}_{i=1}^{n_K}$ 为计算域 Ω_e 的剖分，将其划分为 n_K个具有非空内部的紧致连通的Lipschitz集合 K_i ，且 $\mathring{\mathrm{K}}_i \cap \mathring{\mathrm{K}}_j = \emptyset \ \forall \ \mathrm{i} \neq \mathrm{j}$。

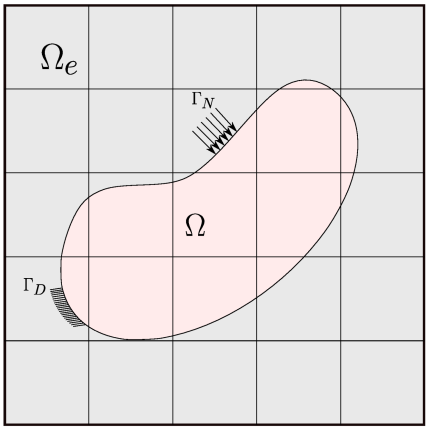

Figure 1: Illustration of the physical domain Ω along with Dirichlet (Γ_D) and Neumann (Γ_N) boundaries in a structured embedding domain Ω_e

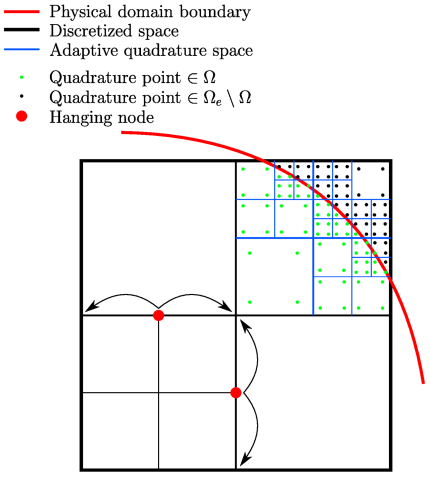

Figure 2: Adaptive quadrature integration on a cell cut by the physical domain and handling of hanging nodes as a result of adaptive mesh refinement

$\Omega_{e,\ h} := \cup^{n_K}_{i=1}\mathrm{K}_i$ 随后定义了 Ω_e的一个近似。引入有限维函数空间 $\mathrm{V}_h \subset \mathrm{H}^1(\Omega_e)$，可得到以下弱形式:

求解 $\mathrm{u}_h \in \mathrm{V}_h$ ，使得对所有 $\mathrm{v}_h \in \mathrm{V}_h$

$$a_h(u_h, v_h) = b_h(v_h), \tag{7}$$

with

$$a_h(u_h, v_h) := \int_{\Omega_e} \alpha \nabla v_h \cdot \nabla u_h \, d\boldsymbol{x} - \int_{\Gamma_D} v_h (\nabla u_h \cdot \boldsymbol{n}) \, d\boldsymbol{s} - \int_{\Gamma_D} u_h (\nabla v_h \cdot \boldsymbol{n}) \, d\boldsymbol{s} + \int_{\Gamma_D} \lambda v_h u_h \, d\boldsymbol{s},$$

$$b_h(v_h) := \int_{\Omega_e} \alpha v_h f \, d\boldsymbol{x} + \int_{\Gamma_N} v_h t \, d\boldsymbol{s} - \int_{\Gamma_D} g (\nabla v_h \cdot \boldsymbol{n}) \, d\boldsymbol{s} + \int_{\Gamma_D} \lambda v_h g \, d\boldsymbol{s}. \tag{8}$$

嵌入域通常是规则的，可以使用多种数据结构（如空间树）高效生成。[34] 在本研究中，我们采用自适应求积，并结合基于空间树的自适应网格细化（AMR）进行体积分，如图 2 所示。自适应求积积分是对空间树数据结构上的自适应网格细化的自然扩展。需要指出的是，自适应积分空间（图 2 中的蓝线）并不属于全局方程组。如图 2 所示，自适应网格细化会产生非协调离散，即悬挂节点；为获得一致的解，必须妥善处理这些节点。我们将悬挂节点约束到相应的非悬挂节点，并将其从全局方程组中移除。本研究始终使用传统的拉格朗日形函数。

只要恰当选择稳定化参数，对称 Nitsche 方法就具有鲁棒性和准确性；然而，参数过小会导致离散化失去稳定性，而参数过大则等同于罚方法。罚方法虽然稳定，却可能使矩阵严重病态。因此，稳定化参数的选择对离散化至关重要。若干用于估计稳定化参数的方法涉及求解一般特征值问题。[15,17,35] 应用杨氏不等式并取 ε，可对方程（8）中的双线性型进行估计:

$$\begin{aligned} a_h(v_h, v_h) \geq & \int_{\Omega_e} \alpha \nabla v_h \cdot \nabla v_h \, d\boldsymbol{x} - \frac{1}{\varepsilon} \int_{\Gamma_D} v_h v_h \, d\boldsymbol{s} \\ & - \varepsilon \int_{\Gamma_D} (\nabla v_h \cdot \boldsymbol{n})(\nabla v_h \cdot \boldsymbol{n}) \, d\boldsymbol{s} + \int_{\Gamma_D} \lambda v_h v_h \, d\boldsymbol{s}. \end{aligned} \tag{9}$$

假设存在常数 C，使得

$$C \int_{\Omega_e} \alpha \nabla v_h \cdot \nabla v_h \, d\boldsymbol{x} \geq \int_{\Gamma_D} (\nabla v_h \cdot \boldsymbol{n})(\nabla v_h \cdot \boldsymbol{n}) \, d\boldsymbol{s}, \tag{10}$$

并假设稳定化参数 λ 在整个积分区域内为常数 ，
可得以下关系

$$a_h(v_h, v_h) \geq (1 - \varepsilon C) \int_{\Omega_e} \alpha \nabla v_h \cdot \nabla v_h \, d\boldsymbol{x} + (\lambda - \frac{1}{\varepsilon}) \int_{\Gamma_D} v_h v_h \, d\boldsymbol{s}. \tag{11}$$

为了保证双线性型的强制性，要求 $(1 - \varepsilon C)$ 和 $(\lambda - \frac{1}{\varepsilon})$ 均为正值。由此可以很容易地证明，必然有 $\lambda > C$。在这一确保稳定性的约束条件下，为避免数值病态，应尽可能选择较小的稳定化参数 λ 。因此，为了确定 λ 的合适取值，需要对下界 C 进行良好的估计。这可以通过公式（10）实现，该公式允许通过求解广义特征值问题来计算 C，具体如下:

$$\boldsymbol{K}\boldsymbol{v} = \Lambda \boldsymbol{M}\boldsymbol{v}, \tag{12}$$

其中，\(v\) 和 Λ 分别为特征向量和特征值，使用基 $\{\phi_i\} \subset V_h$，矩阵 K 和 M 通过以下方式构造：

$$\begin{cases} \boldsymbol{K}_{ij} := \int_{\Gamma_D} (\nabla \phi_i \cdot \boldsymbol{n})(\nabla \phi_j \cdot \boldsymbol{n}) \, d\boldsymbol{s}, \\ \boldsymbol{M}_{ij} := \int_{\Omega_e} \alpha \nabla \phi_i \cdot \nabla \phi_j \, d\boldsymbol{x}, \end{cases} \tag{13}$$

对于该模型问题。现在，Nitsche方法中稳定化参数的下界 C 可取为最大特征值 $\max_k \Lambda_k$。请注意，在公式（13）中，M 是在整个区域上积分的。然而，为保守起见，相应的积分可以限制在与狄利克雷边界相交的区域部分上，即公式（10）中的部分，并因此也适用于公式（13）。

根据上述推理，可以提出两种稳定化参数估计方法：全局方法通过求解公式（12）给出的一个广义特征值问题来估计 C，其中 M 的积分仅限于与 Γ_D 相交的单元 $K \in T_h$。由此可得到一个适用于整个区域的稳定化参数 λ_g。另一种方法是局部方法，即将公式（10）中的估计以及类似地将公式（12）中的广义特征值问题限制在每个被狄利克雷边界切割的单元 $K \in T_h$ 所在的区域内。这样，通过求解一系列局部广义特征值问题，可计算出逐单元为常数的稳定化参数 λ_l^K，其中 $K \in T_h$ 且 $K \cap \Gamma_D \neq \emptyset$。为简化记法，我们将逐单元参数集合 $\{\lambda_l^K\}$ 记为 λ_l。

在本研究中，我们采用这两种方法，并在第 3 节比较这两种方法对迭代求解器性能的影响。从计算角度来看，主要区别在于求解一个大型特征值问题，还是求解多个较小的特征值问题。然而，由于矩阵存在秩亏，求解广义特征值问题 (12) 并非易事。一种可用于求解该问题的算法是奇异值分解，本研究采用了这种算法。对于全局方法，随着特征值问题规模增大，在非常细的网格上计算成本可能高得难以承受；而局部方法则仍然适用。

2.3 几何多重网格

第 2.2 节所述模型问题的离散形式可得以下方程组：

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{14}$$

其中，A 和 b 分别根据方程 (8) 中的双线性型和线性型定义，x 为解向量。

众所周知，当物理域与计算嵌入域中的单元相交部分较小时，会出现较小的切割比例，这可能导致浸入有限元方法出现数值病态，使系统矩阵的条件数变得任意大；因此，必须进行特殊处理，才能成功应用迭代法。本研究采用 Schwarz 型光滑器处理切割单元。[31,32] 多重网格方法的基本思想，即在细网格上平滑误差的振荡频率，并在粗网格上近似平滑频率，仍然保持不变。

2.3.1 组成部分

在最基本的形式下，几何多重网格需要以下组成部分：网格层次结构 τ_0，... τ_n，其中 τ_n 是原始细网格，τ_0 是最粗网格，

传递算子 R_l 和 $\mathrm{P}_l = \mathrm{R}_{l+1}^T$，分别将向量从第 l 层限制到 $\mathrm{l}-1$，以及从第 l 层延拓到 $\mathrm{l}+1$；一个平滑算子 S，用于除最粗层 τ_0 外每一层的若干预平滑和后平滑步骤；以及一个用于最粗层 τ_0的基求解器。

网格层次结构 本文着重介绍所采用的自适应网格细化中的几个重要方面。图 3 展示了一个示例网格细化过程，其中包含四层网格层次结构。所有网格均满足 2:1 平衡条件，即相邻象限的细化层级之差不超过两级。从细网格 τ_l 出发，为得到相邻的粗网格 τ_{l-1}，遍历数据结构，并对所有细化层级最高的象限进行粗化。递归应用此算法，即可得到所需数量的嵌套空间。

传递算子 给定层级 l 上的向量 v_l，可在粗层级 $\mathrm{l}-1$ 上得到受限向量 v_{l-1}，其表达式为

$$\boldsymbol{v}_{l-1} = \boldsymbol{R}_l \boldsymbol{v}_l, \tag{15}$$

其中，R_l 为限制算子。另一方面，双线性型的限制可通过两种常用方式实现。一种方式是使用限制算子和延拓算子，如下所示：

$$\boldsymbol{A}_{l-1} = \boldsymbol{R}_l \boldsymbol{A}_l \boldsymbol{R}_l^T, \tag{16}$$

我们将其称为限制-算子-延拓（RAP）法。另一种方式则通过在粗网格的函数空间中计算双线性型来构造粗矩阵，并结合基于网格的稳定化参数估计。我们将这种方式称为组装法。前文各节公式中网格相关稳定化参数的存在，使这两种方法在多层方法的背景下有了显著区别。第 3 节将探讨这一影响。

平滑器 我们在最细网格上使用乘法 Schwarz 型光滑器[31,32] 来处理切割单元。该平滑器通过求解一系列局部子域问题来实现：对于每个被物理边界切割的单元，我们创建一个基于单元的子域，其中包含与该单元相关联的所有自由度。对于其余未出现在基于单元的子域中的自由度，则分别创建一个单自由度子域。[32] 给定这 n_{sd} 个子域，平滑器算子可写为

$$\boldsymbol{S} = (\boldsymbol{R}_{s,n_{sd}}^T \boldsymbol{A}_{s,n_{sd}}^{-1} \boldsymbol{R}_{s,n_{sd}}) \dots (\boldsymbol{R}_{s,1}^T \boldsymbol{A}_{s,1}^{-1} \boldsymbol{R}_{s,1}) \quad \text{with} \quad \boldsymbol{A_{s,i}} = \boldsymbol{R_{s,i}} \boldsymbol{A} \boldsymbol{R_{s,i}^T}, \tag{17}$$

其中，$\mathrm{R}_{s,\ i}$ 是第 i 个子域的限制算子，而 $\mathrm{A}_{s,\ i}$ 是第 i 个子域的局部块。

对于较粗网格上的组装方法，可以采用相同的平滑策略。然而，对于 RAP 方法，粗网格矩阵的构造完全是代数的。因此，所采用的平滑器通常也选用代数方法。由于我们的研究主要旨在比较这两种方法，因此在两种情况下，我们都在粗网格上采用阻尼 Jacobi 平滑器。请注意，对组装后的粗矩阵使用 Schwarz 平滑器，很可能会获得比第 3 节中为便于比较而展示的结果更好的收敛率。

基准求解器 我们在基准层使用直接 LU 求解器。鉴于直接求解器[27], 的计算复杂度很高，我们注意到，正如第 3 节所述，将基准问题的规模保持在相对较小的范围内非常重要。

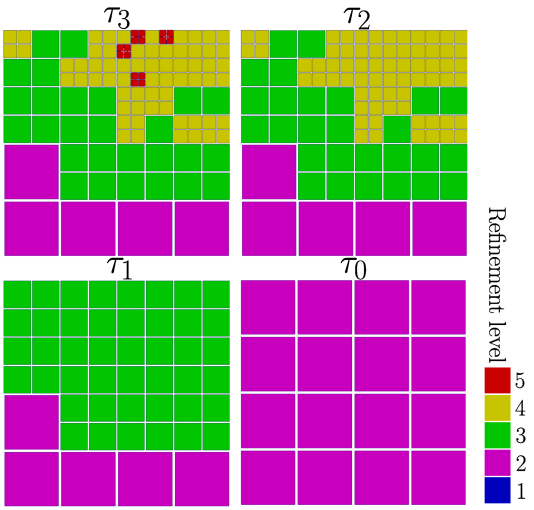

Figure 3: The hierarchy of grids constructed from a randomly-refined fine grid, τ_3. Cells are colored based on their refinement level

Table 1: Mesh hierarchy: the coarse mesh τ_0 is constructed by five levels of uniform refinement on the unit square. For $k \geq 1$, the meshes τ_k are constructed by adaptively refining its coarser parent τ_{k-1} one level towards the boundary of the physical domain. $n_{\texttt{DoF}}$ is the number of degrees of freedom in the mesh. λ_g and λ_l^{mean} are the global and mean local eigenvalues, respectively

Mesh	$n_{\texttt{DoF}}$	λ_g	λ_l^{mean}
τ_0	1089	-	-
τ_1	1209	1279.3	312.23
τ_2	1657	13343.2	912.02
τ_3	2637	22770.2	2457.70
τ_4	4661	310834.0	4188.30
τ_5	8665	136533.0	17873.99
τ_6	16953	-	30835.11
τ_7	33269	-	62312.00
τ_8	65689	-	161481.18

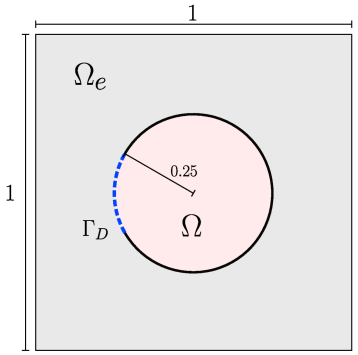

Figure 4: A circular physical domain is embedded in a unit square. A constant-valued inhomogeneous Dirichlet boundary is imposed on an arch to the left of the physical domain and the rest is left as homogeneous Neumann boundary

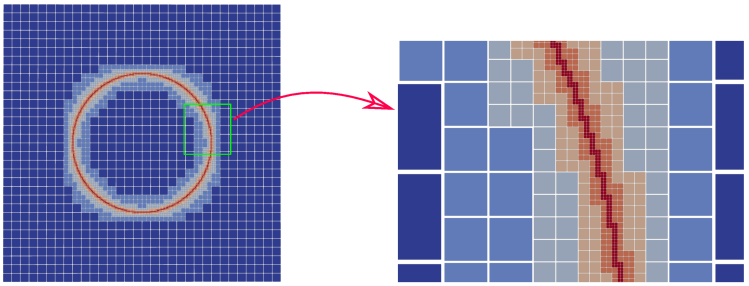

Figure 5: The discretization of the domain for τ_5, i.e., five levels of uniform refinement on the unit square followed by five levels of adaptive refinement towards the boundary of the physical domain

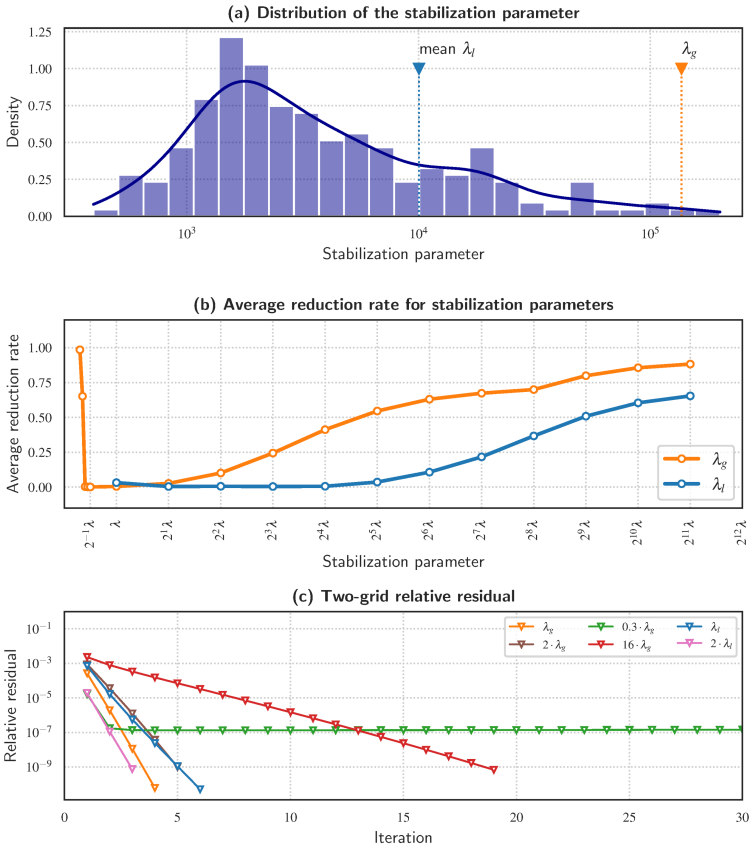

Figure 6: Two-grid solver on mesh τ_5. (a) The distribution of the local stabilization parameter computed from the generalized eigenvalue problem and the global stabilization parameter, (b) the average reduction rate for variations in the global and local stabilization parameters and (c) the convergence of the iterative solver in terms of relative residual for selected cases. λ_g and λ_l denote the computed stabilization parameters using the global and local approaches, respectively. In (b), λ denotes the element-wise constant stabilization parameter in the case of the local approach and the global stabilization parameter in the case of the global approach. Note that the solver does not converge for values smaller than the computed value in the local approach.

3 数值研究

在数值实验中，我们考虑一个嵌入正方形内的圆形物理域，如图4所示，并求解带有狄利克雷和诺伊曼边界条件的泊松问题。在域左侧的拱形部分施加值为 1.0 的非齐次狄利克雷边界条件，其余部分则采用齐次诺伊曼边界条件，如图4所示。从单位正方形开始，细化准则包括五层均匀细化，随后朝物理域边界进行若干层自适应细化，从而形成贯穿本节使用的嵌套网格层次结构。网格层次结构详见表 1。网格 τ_5 对该域进行离散化的结果如图5所示。此外，体积分采用八层自适应求积。

第2.3节所述的多重网格求解器用于求解离散化问题。采用一个包含三个预平滑步骤和三个后平滑步骤的V循环。将阈值为 10^{-9} 的相对残差作为收敛判据。惩罚参数 α 为10^{-10}。Nitsche公式中的稳定化参数通过第2.2节所述的局部逐单元方法或全局方法计算。由广义特征值问题计算得到的全局参数和局部参数分别记为 λ_g 和 λ_l。我们使用内部代码实现有限单元法和多重网格算法，并开展数值研究。p4est[34]和 PETSc [36]分别用于八叉树和线性代数功能。

3.1 稳定化参数的影响

为了研究迭代方法对各方案中稳定化参数的敏感性，本文考察了相对于计算所得参数的若干变化：将全局或局部稳定化参数乘以不同的因子，以分析其取值小于和大于通过求解第 2.2 节所述特征值问题得到的估计值时的行为。采用网格τ_5 上的双网格方法，以便更好地分离稳定化参数的影响。求解器的收敛性以及稳定化参数的分布如图 6 所示。请注意，全局稳定化参数不一定大于最大的局部参数，如图 6(a) 所示；也就是说，对于某些切割单元，计算得到的全局稳定化参数可能小于局部参数。

对于全局稳定化参数方案，如图 6(b) 所示，在计算值附近，解可以快速收敛。然而，当稳定化参数小于由广义特征值问题计算得到的参数时，可能会得到较低的下降率。这一趋势会持续到解的稳定性丧失为止。在此时，我们观察到解可能渐近地达到一个平台，或发生发散。图 6(c) 展示了这种行为的一个示例：稳定参数约为计算得到的全局值的三分之一时，解在大约 10^{-7} 处达到平台，并缓慢发散。进一步减小全局稳定化参数会导致求解器发散。由此可推断，对于该方案，解的最理想行为是在确保稳定性的前提下取尽可能小的全局参数值。多项研究[17,18] 所建议的 $2\lambda_g$经验选取方式，在全局估计的求解器收敛率方面似乎并无明显优势。

对于局部稳定化参数方案，图 6(b) 展示了稳定化参数变化时解的表现。可以观察到，最小下降率出现在计算得到的局部参数附近。虽然参数低于计算值时解会迅速变得不稳定，但与全局参数相比，解在更大的参数范围内仍能保持一致。由此可以推断，计算得到的局部稳定参数接近确保问题具有强制性所需的最小值。在这种情况下，经验值 $2\lambda_l$ 似乎是一个合理的选择。

尽管两种方法在计算得到的参数值处实现了几乎相同的收敛率，但在较大的相对参数下，局部方案的平均下降率明显低于其全局对应方案；在参数区间的大部分范围内，采用局部方案的迭代求解器表现出更好的收敛性。此外，在该方案中，迭代方法对稳定化参数变化的鲁棒性更强，而且在相当大的参数变化范围内，最小下降率都相对稳定。图 6(a) 中局部稳定化参数的分布有助于解释这一现象。不难看出，局部方案中的稳定化参数分布与计算得到的全局值之间存在巨大差异。直观而言，全局值可视为满足所有单元强制性条件所需最小值的估计，而局部值只需满足局部区域内的条件。这种灵活性使稳定参数能够实现更准确的分布，从而提升求解器的性能。

虽然求解器的其他配置（例如增加预/后平滑步）确实会改变平均缩减率等指标的绝对值，以及所需的迭代次数，但我们并未观察到这些变化会导致求解器的收敛行为出现根本性差异。

3.2 多重网格收敛

本节通过网格研究分析稳定化参数的网格依赖性对多重网格求解器的影响。采用表 1 中的网格层次结构。网格 τ_0 用作所有测试的基准问题，并使用直接求解器求解至机器精度。因此，规模更大的问题采用更深的网格层次结构。稳定化参数的局部和全局估计，以及计算粗网格矩阵的两种方法（即限制-算子-延拓（RAP）和组装），组合得到四种可能的情况。几何多重网格求解器在每一层上的配置与前一个示例相同。最细网格上采用 Schwarz 平滑器，所有较粗层上采用阻尼 Jacobi 平滑器。 $2\lambda_g$ 和 $2\lambda_l$ 分别用于全局方案和局部方案。在最细的三个问题上，仅采用局部方案，因为全局方案的计算成本高得难以承受。图 7(a) 给出了各情况下多重网格求解器的平均收敛率。

可以看出，对于获得系统矩阵的两种方法，即限制-算子-延拓（RAP）和组装，局部估计法都优于全局对应方案。随着层次结构的层数增加，两者之间的差距也随之扩大。无论采用局部估计还是全局估计，组装法的收敛率都低于其对应的 RAP 方法。在层次结构较高的层级上，各种组合之间的收敛率存在显著差距。

组装粗网格矩阵时，在最初几个层级中，全局估计方案的收敛率与局部估计方案相对接近，之后两者的收敛率开始出现偏差。这种现象背后的机制有两方面。在这种情况下，每一层都会估计全局稳定化参数，因此该参数适用于相应的离散化。这可以视为全局方案在这种配置下能够在一定层级范围内表现相对良好的主要原因。收敛率的恶化可能是由于多层配置中每一层略逊的收敛率逐渐累积所致。层次深度的增加会进一步加剧这一现象。

通过限制获得粗网格矩阵时，也观察到了类似的行为。通过细网格上的局部方案更准确地分布稳定化参数，会导致下降率降低，尤其是在层次结构的较高层级。

无论采用哪种稳定化参数估计方法，使用组装法构造粗矩阵都能获得更小且更具可扩展性的收敛率。这可归因于对稳定化参数更准确的估计

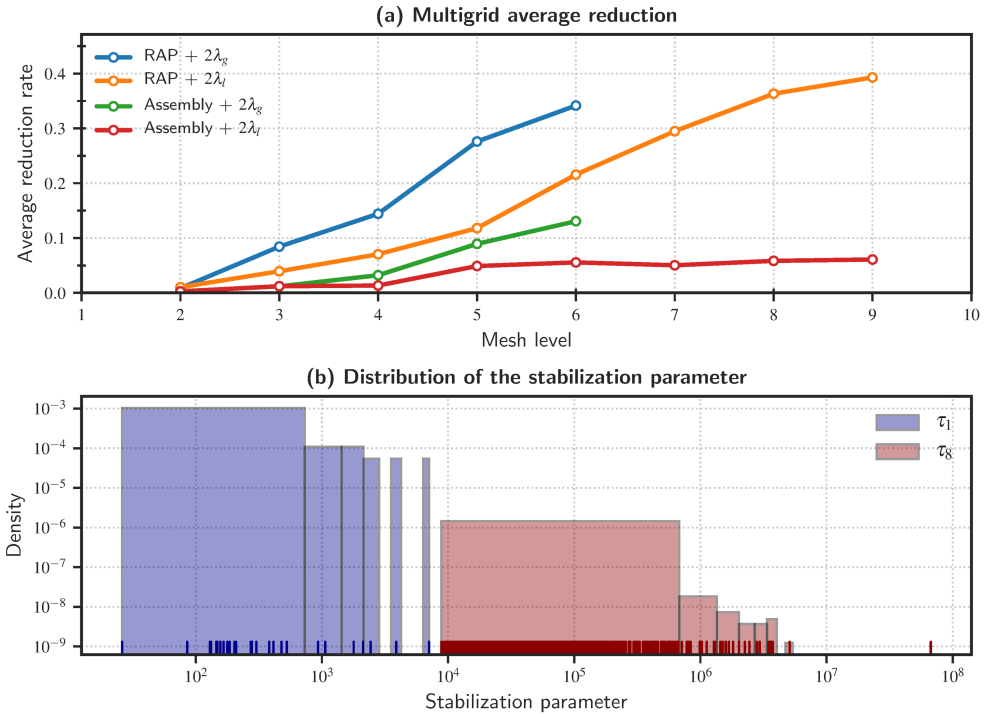

Figure 7: Study for multigrid hierarchy: (a) Average reduction rate for the geometric multigrid solver for global and local estimation of the stabilization parameter and two approaches of obtaining the coarse grid matrices, (b) the distribution of the local stabilization parameter on a coarse (τ_1) and a fine (τ_8) mesh

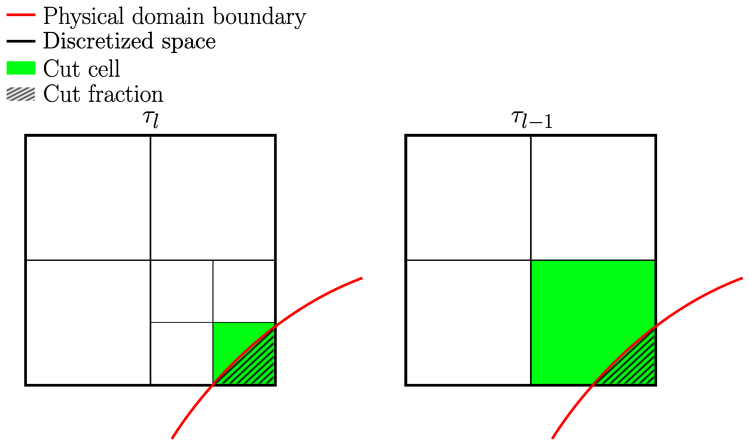

Figure 8: A possible scenario where the smaller cut fraction on the coarser grid could potentially lead to a larger stabilization parameter

在每个层次上都能获得更好的收敛率。在限制-算子-延拓（RAP）方法中，这种估计只在最细网格上进行，因此对于较粗网格而言可能并不合理。观察层次结构不同深度处稳定化参数的分布，可以更直观地理解这一点。图 7(b) 展示了两个样本网格上的稳定化参数分布。显然，稳定化参数的分布高度依赖于网格。若将一个网格上的分布用于另一个网格，根据上一节的结果，要么会导致收敛极其缓慢，要么会导致发散。由于粗网格上的单元大多较大，细网格得出的稳定化估计通常足以满足粗网格的需要。RAP 方法的收敛率及其相对稳健性，或许可以用这一机制来解释。不过，我们想指出一种可能出现的情况。稳定化估计不仅取决于网格尺寸，也取决于切割构型。具体而言，不利的切割单元——即物理边界仅与单元的一小部分相交的单元——会导致较大的稳定化参数。因此，粗网格上所需的稳定化参数可能大于细网格上的参数。例如，表 1 中 τ_4 与 τ_5 之间便体现了这一点：在全局估计中，较粗的网格 τ_4 所需的稳定化参数大于较细的网格 τ_5，但在局部估计中，τ_4 所需的稳定化参数平均而言更小。此外，图 8 展示了一种可能的情形：尽管粗网格的网格尺寸更大，其切割构型却有所恶化。总之，研究发现，通过组装法在每个网格层次上进行专门估计，可以显著提高求解器的收敛率。

4 结论

我们研究了沉浸式有限元公式的几何多重网格求解过程中，Nitsche方法中依赖于网格的稳定化参数。研究发现，稳定化参数不仅对于确保双线性算子的强制性，进而保证解的稳定性至关重要，而且还会显著影响迭代求解器的性能。

稳定化参数通常可以在积分过程中作为整个计算域统一的常数或逐单元常数引入。针对这两种方法，可以通过广义特征值问题分别构建合理的估计，从而得到全局估计和局部估计。全局估计通过求解一个广义特征值问题获得一个常数参数，而局部估计则通过一系列规模较小的广义特征值问题，得到稳定化参数的分布。对于多重网格框架，如果组装了粗网格矩阵，则可以针对每个网格层级分别计算稳定化参数。对于采用 RAP 方法构造粗网格矩阵的情况，稳定化参数仅在最细网格上进行估计，随后通过限制操作将其隐式包含在较粗层级的矩阵中。

对于稳定化参数方法，我们发现，偏离广义特征值问题所得到的数值会显著影响求解器的收敛率。具体而言，在全局估计中，当全局参数取能够保证解稳定性的最小值时，求解器表现最佳，而该值可能比通过广义特征值问题计算得到的值小数倍。对于局部估计，求解器往往在参数取值为 $2\lambda_l$-$4\lambda_l$ 时达到最小下降率。在这种方法中，当参数低于计算得到的分布时，解会迅速变得不稳定。然而，与全局估计相比，该方法对于较大的相对参数值具有更强的鲁棒性。在这两种方法中，当参数被高估时，求解器的下降率都会恶化。我们发现，采用局部估计方法时，求解器往往能够获得更好的收敛率，尤其是在层次结构较深的情况下。

稳定化参数的离散化依赖性会影响多层求解器的性能。通过对双线性形式进行评估来构建粗网格矩阵。

在粗函数空间上，可以获得层次结构中每个层级的单独稳定化参数估计；我们发现，这能够提供更好的收敛率。对于限制-算子-延拓（RAP）方法，最细层级的稳定化参数估计也会隐式地用于粗网格上，从而导致收敛率下降，尤其是在更深的网格层次结构中。

对于局部稳定化参数估计，结合层次结构每一层的组装矩阵，我们实现了最快且最具可扩展性的收敛率。考虑到采用广义特征值求解器计算全局估计的成本不断增加，而局部估计的性能更佳，局部估计似乎是获取稳定化参数的有利方案。通过在粗空间上评估双线性型，为多层次结构的每一层分别计算稳定化参数似乎也很有利，因为这会在粗矩阵中生成适配各层的稳定化参数，从而使我们的研究中的多重网格求解器获得更好的收敛率。

致谢

本研究获得了德国研究联合会（Deutsche Forschungsgemeinschaft，DFG）在协作研究中心 SFB 837《机械化隧道施工中的交互建模》子项目 C4 框架下提供的资助，资助编号为 77309832。我们谨此致谢。我们还衷心感谢使用 SFB 837 和波鸿鲁尔大学土木与环境工程系计算集群的计算时间，这些计算资源用于开展本文所述研究。

参考文献

[1] Ted Belytschko、Nicolas Mo¨es、Shuji Usui 和 Chandu Parimi。有限元中的任意不连续性。International Journal for Numerical Methods in Engineering，50(4):993–1013，2001年。[2] Jamshid Parvizian、Alexander D¨uster 和 Ernst Rank。有限胞元法。Computational Mechanics，41(1):121–133，2007年。[3] Alexander D¨uster、Jamshid Parvizian、Zhengxiong Yang 和 Ernst Rank。有限胞元法在三维固体力学问题中的应用。Computer Methods in Applied Mechanics and Engineering，197(45-48):3768–3782，2008年。[4] Erik Burman、Susanne Claus、Peter Hansbo、Mats G Larson 和 Andr´e Massing。CutFEM：几何与偏微分方程的离散化。International Journal for Numerical Methods in Engineering，104(7):472–501，2015年。[5] Sonia Fern´andez-M´endez 和 Antonio Huerta。无网格方法中本质边界条件的施加。Computer Methods in Applied Mechanics and Engineering，193(12-14):1257–1275，2004年。[6] Bernd Flemisch 和 Barbara I Wohlmuth。三维曲面界面四边形网格的稳定拉格朗日乘子。Computer Methods in Applied Mechanics and Engineering，196(8):1589–1602，2007年。[7] R Glowinski 和 Yu Kuznetsov。基于虚拟区域法求解二阶椭圆问题的分布式拉格朗日乘子。Computer Methods in Applied Mechanics and Engineering，196(8):1498–1506，2007年。[8] Axel Gerstenberger 和 Wolfgang A Wall。面向复杂流固耦合应用的固定网格方法改进。International Journal for Numerical Methods in Fluids，57(9):1227–1248，2008年。

[9] A Gerstenberger 和 WA Wall。三维连续体的嵌入式狄利克雷形式。International Journal for Numerical Methods in Engineering，82(5):537–563，2010年。[10] Erik Burman 和 Peter Hansbo。使用切割单元的虚拟区域有限元方法：I. 稳定化拉格朗日乘子法。Computer Methods in Applied Mechanics and Engineering，199(41-44):2680–2686，2010年。[11] Ivo Babuˇska。带罚项的有限元法。Mathematics of computation，27 (122):221–228，1973年。[12] T Zhu 和 SN Atluri。改进的配点法与罚函数形式：在无单元伽辽金法中施加本质边界条件。Computational Mechanics，21(3):211–222，1998年。.[13] Joachim Nitsche。关于一种变分原理：使用不满足边界条件的子空间求解狄利克雷问题。载于《汉堡大学数学研讨会论文集》，第36卷，第9–15页。Springer，1971年。[14] Anita Hansbo 和 Peter Hansbo。一种基于Nitsche方法的非拟合有限元法，用于椭圆界面问题。Computer Methods in Applied Mechanics and Engineering，191(47-48):5537–5552，2002年。[15] Michael Griebel 和 Marc Alexander Schweitzer。粒子分割统一法第五部分：边界条件。载于《几何分析与非线性偏微分方程》，第519–542页。Springer，2003年。[16] Yuri Bazilevs 和 Thomas JR Hughes。流体力学中狄利克雷边界条件的弱施加。Computers & Fluids，36(1):12–26，2007年。[17] John Dolbow 和 Isaac Harari。一种用于嵌入式界面问题的高效有限元法。International journal for numerical methods in engineering，78(2):229–252，2009年。[18] Anand Embar、John Dolbow 和 Isaac Harari。使用Nitsche方法和基于样条的有限元施加狄利克雷边界条件。International journal for numerical methods in engineering，83(7):877–898，2010年。[19] Erik Burman 和 Peter Hansbo。使用切割单元的虚拟区域有限元方法：II. 稳定化Nitsche方法。Applied Numerical Mathematics，62(4):328–341，2012年。[20] John A Evans 和 Thomas JR Hughes。用于稳态纳维–斯托克斯方程的等几何发散相容B样条。Mathematical Models and Methods in Applied Sciences，23(08):1421–1478，2013年。[21]Daniel Elfverson、Mats G. Larson 和 Karl Larsson。一种用于切割等几何分析的新型最小二乘稳定化Nitsche方法。Computer Methods in Applied Mechanics and Engineering，349:1–16，2019年。ISSN 0045-7825。doi：https://doi.org/10.1016/j.cma.2019.02.011。URL https://www.sciencedirect.com/science/article/pii/S0045782519300829。[22] Karl Larsson、Stefan Kollmannsberger、Ernst Rank 和 Mats G Larson。带最小二乘稳定化Nitsche边界条件的有限胞元法。Computer Methods in Applied Mechanics and Engineering，393:114792，2022年。[23] Dominik Schillinger 和 Martin Ruess。有限胞元法：高阶结构分析中CAD与基于图像的几何模型的综述。Archives of Computational Methods in Engineering，22(3):391–455，2015年。

[24] Dominik Schillinger, Martin Ruess, Nils Zander, Yuri Bazilevs, Alexander D¨uster 和 Ernst Rank。采用有限胞元法的 p 版和 B 样条版进行小变形与大变形分析。计算力学，50(4):445–478，2012年。[25] Martin Ruess, Dominik Schillinger, Y Bazilevs, Vasco Varduhn 和 E Rank。基于有限胞元法，对嵌入非均匀有理B样条及裁剪非均匀有理B样条几何中的本质边界条件进行弱施加。International Journal for Numerical Methods in Engineering，95(10):811–846，2013年。[26] Martin Ruess, Dominik Schillinger, Ali I Oezcan 和 Ernst Rank。非匹配和裁剪多片几何等几何分析的弱耦合。Computer Methods in Applied Mechanics and Engineering，269:46–71，2014年。[27] Yousef Saad。稀疏线性系统的迭代方法，第82卷。美国工业与应用数学学会，2003年。[28] Wolfgang Hackbusch。多重网格方法及其应用，第4卷。施普林格科学与商业媒体，2013年。[29] Thomas Ludescher, Sven Gross 和 Arnold Reusken。用于椭圆界面问题非拟合有限元离散的多重网格方法。《美国工业与应用数学学会科学计算杂志》，42(1):A318–A342，2020年。[30] Hardik Kothari 和 Rolf Krause。一种用于基于Nitsche方法的扩展有限元法的多重网格方法。计算机视觉科学，2021年。[31] Frits de Prenter, Clemens V Verhoosel, EH van Brummelen, JA Evans, Christian Messe, Joseph Benzaken 和 Kurt Maute。用于浸入式有限元方法和浸入式等几何分析的多重网格求解器。计算力学，65:1–32，2019年。[32] S. Saberi, A. Vogel 和 G. Meschke。采用自适应几何多重网格的并行有限胞元法。载于 Maciej Malawski 和 Krzysztof Rzadca 编，《Euro-Par 2020：并行处理》，第578–593页，施普林格出版社，2020年。施普林格国际出版。ISBN 978-3-030-57675-2。[33] Saberi, G. Meschke 和 A. Vogel。斯托克斯方程和纳维–斯托克斯方程混合有限胞元公式的自适应几何多重网格。International Journal for Numerical Methods in Fluids，n/a(n/a)，2023年。doi：https://doi.org/10.1002/fld.5180。URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.5180。[34] Carsten Burstedde, Lucas C Wilcox 和 Omar Ghattas。p4est：用于八叉树森林并行自适应网格细化的可扩展算法。《美国工业与应用数学学会科学计算杂志》，33(3):1103–1133，2011年。[35] Wadhah Garhuom, Khuldoon Usman 和 Alexander D¨uster。一种通过特征值稳定化技术提高有限胞元法在有限应变问题中鲁棒性的方法。计算力学，69(5):1225–1240，2022年。[36] Satish Balay, William D. Gropp, Lois Curfman McInnes 和 Barry F. Smith。面向对象数值软件库中的并行性高效管理。载于 E. Arge, A. M. Bruaset 和 H. P. Langtangen 编，《科学计算中的现代软件工具》，第163–202页。Birkh¨auser出版社，1997年。