

面向大规模并行计算的区域平衡 PDE 求解方法*

陈玉惠, 黄诗杰, 姚清河

中山大学航空航天学院, 广东 深圳 518107

摘要: 利用非重叠的区域分解方法 (DDM) 探讨了以 Poisson 方程、热传导方程和波动方程为代表的椭圆型、抛物型和双曲型偏微分方程 (PDEs) 的数值求解效率及内存消耗。针对由 DDM 产生的子区域间界面问题规模较大且奇异的特点, 采用了平衡区域分解 (BDD) 方法, 该方法结合了共轭梯度迭代法与预处理技术。所采用的并行算法基于对称多处理器 (SMP) 结构, 所有处理器单元地位平等且共享内存。首先, 介绍了基于 Poisson 方程的 DDM 和 BDD 实现方法。其次, 阐述了 3 种 PDEs 的有限元离散过程及其对应的离散矩阵形式。然后, 通过固定 H/h 、增加总自由度数量, 比较不同情况下迭代次数的变化; 并在 1000×1000 和 2000×2000 剖分下, 分析了 DDM 和 BDD 在求解这 3 类 PDEs 时的迭代效率与内存消耗量。最后, 通过扩散反应方程验证了 BDD 相较于 DDM 在数值求解方面具有更高的效率。

关键词: 区域平衡分解方法; 数值效率; 并行处理; 数值可扩展性

中图分类号: O241.7 **文献标志码:** A **文章编号:** 2097-0137 (2025) 05-0050-09

Balancing domain decomposition method of solving PDE for massively parallel computing

CHEN Yuhui, HUANG Shijie, YAO Qinghe

School of Aeronautics and Astronautics, Sun Yat-sen University, Shenzhen 518107, China

Abstract: This study investigates the numerical solution efficiency and memory consumption of Poisson equation, heat conduction equation, and wave equation, using a non-overlapping domain decomposition method (DDM). To address the large-scale and singular nature of interface problems between subdomains generated by DDM, the balanced domain decomposition (BDD) method was employed. This method integrates conjugate gradient iteration with preconditioning techniques. The parallel algorithm is based on a symmetric multiprocessing (SMP) architecture, where all processor units are equal in status and share memory. First, the implementation of DDM and BDD based on the Poisson equation is introduced. Next, the finite element discretization processes and corresponding discrete matrix forms for the three PDEs are presented. Then, by increasing the total degrees of freedom while maintaining H/h ratio, the variation in iteration counts under different conditions is compared. Additionally, the iterative efficiency and memory consumption of DDM and BDD when solving these three PDEs are analyzed and contrasted under 1000×1000 and 2000×2000 mesh partitions.

* 收稿日期: 2024-05-29 录用日期: 2024-06-13 网络首发日期: 2025-07-03

基金项目: 广东省基础与应用基础研究基金 (2021B1515310001, 2022B1515120009)

作者简介: 陈玉惠 (1991 年生), 女; 研究方向: 偏微分方程; E-mail: chenyh339@mail.sysu.edu.cn;
黄诗杰 (1997 年生), 男; 研究方向: 偏微分方程; E-mail: huangshj67@mail2.sysu.edu.cn
(陈玉惠、黄诗杰并列第一作者)

通信作者: 姚清河 (1980 年生), 男; 研究方向: 流体力学; E-mail: yaoqhe@mail.sysu.edu.cn



ZR20240178

Finally, the diffusion-reaction equation is used to verify that BDD is more efficient than DDM in numerical solutions.

Key words: balancing domain decomposition; numerical efficiency; parallel processing; numerical scalability

自然界中很多的物理过程可以用偏微分方程(PDE)描述。由于解析方法求解PDE具有很高的难度,因此通常采用数值逼近的方法得到PDE的近似解。随着PDE所求近似解精度要求的提高以及离散系统的规模不断增大,对于一些复杂问题,其大规模离散代数系统的性态较差,此时传统的求解算法难以使用。预条件 Krylov 子空间迭代法是一种求解PDE离散系统的重要方法,由于该方法的迭代收敛速度依赖于预处理系统的条件数,因此预条件子的构造是影响该算法效率的关键。

区域分解法(DDM)是求解复杂PDE大规模离散代数系统的一类非常流行的方法,其基本思想是将计算区域分解为若干个子区域,从而将整个计算区域上原问题的求解转化为各子区域上的求解。这种方法有利于采用并行算法加速,并且特别适合于含跳系数的情形。非重叠DDM是一类重要的DDM,其特点是各子区域间互不重叠。

经过几十年的发展,目前已经形成了多种类型的非重叠DDM预条件子,如:Dirichlet-Neumann型(Bjorstad et al., 1989)通过对区域的细致划分,可以实现计算求解过程的高效优化。然而,此方法对收敛性的要求较高。Neumann-Neumann型(Bourgat et al., 1988)在有限的子域范围内展示了其有效性。但由于粗网格存在的局限性,随着子域数量的增加,该方法的收敛性能会迅速下降。在FETI型方法(Farhat et al., 1991, 1994)内,浮动子域通过线性约束的方式被整合进求解框架内。这种方式的作用与基于约束的BDD方法中的粗网格校正机制相类似,旨在确保子域间的协调性和整体解的准确性。FETI-DP方法(Lee et al., 2021; Pacheco et al., 2022)通过引入对偶-原问题策略,在求解过程中明确地将选定的原问题变量视为未知量进行处理,该方法展现出了显著地计算效率提升。BDD型方法(Cowsar et al., 1995; 赵航, 2019; Kanayama et al., 2022)采用Neumann-Neumann预条件子结合粗网格修正技术以加速迭代收敛过程。BDDC方法(Dohrmann, 2003; Mandel et al., 2003)引入了一种基于约束能量最小化概念的子结构预条件技术。Miyamura et al. (2023)对一种将多点约束

(MPC)整合进BDD方法中的策略(即BDD-MPC方法)进行了深入研究与改进,这促进了BDD-MPC方法在处理复杂、大规模数值模拟任务中的效能和适用性。

在采用BDD方法求解大规模离散系统方面,国内外已有不少的研究,例如:热传导问题的研究中, Mukaddes et al. (2006)提出了一种高效且可扩展的BDD型预条件子,专门用于解决由三维传热问题产生的大规模线性系统。氢在部分开放空间中的分散行为研究(Yao et al., 2015)采用了一种与热对流问题相类比的方法。该研究运用压力稳定域分解方法,对N-S方程及对流扩散方程所生成的线性方程组进行了对称化处理。双圆柱绕流问题的研究(Zhu et al., 2016)利用域分解方法(DDM)进行并行计算,以探讨 $Re = 1000$ 时串列双圆柱绕流的特性。3D血流行为和壁面剪切应力(WSS)分布的研究(Yao et al., 2021)展示了一项基于实验数据的计算分析。锂离子电池降阶电化学模型的研究(Li et al., 2021)专注于开发一种兼具高保真度与低计算成本的降阶电化学模型(ROEM)。小振幅波与激波相互作用的研究(Vevak et al., 2021)提出了一种基于有限体积方法的区域分解技术,旨在处理线性化欧拉方程区域内小振幅声波和熵波的传播,并模拟这些波动与非线性区域内初始稳定的正激波之间的相互作用。心脏细胞逐细胞模型的研究(Huynh et al., 2023)构建并分析了一种BDDC预条件子,以求解在心脏细胞个体建模过程中出现的常微分方程和偏微分方程反应扩散系统的复合间断伽辽金(DG, discontinuous Galerkin)离散化问题。Bevilacqua et al. (2024)针对鞍点问题的虚单元离散化,提出并分析了一种BDDC预条件子。

针对高超声速关键部件的流固耦合、气热耦合分析以及流场振荡抑制,数值模拟提供了重要的研究手段(孙学功等, 2015; 秦启豪, 2020; 李佳伟, 2020; 张佳晨, 2020)。鉴于高超声速模型涉及大规模离散线性方程组的求解,本文探讨了在求解不同类型的PDEs时,BDD相较于传统DDM在迭代次数和内存消耗量方面的优势。研究涵盖了三类典型PDEs:椭圆型方程(以Poisson方程为代表)、抛物型

方程(以热传导方程为代表)及双曲型方程(以波动方程为代表)。通过这三类PDEs的数值实验验证了BDD方法的数值可扩展性,即经过适当预处理后,迭代次数不会随着子域数量的增加而显著增长。本研究采用的DDM和BDD均为并行算法,并且所使用的并行架构基于SMP,在这种架构下所有处理器单元地位平等、无主次之分,并共享内存资源。这种设置有利于最大化并行计算的效率与灵活性。最后,对比了DDM和BDD求解扩散反应方程的表现,进一步验证了BDD方法的高效性。

1 区域平衡分解方法

1.1 区域分解

基于Poisson方程介绍DDM的实现。假定 Ω 是一个光滑的二维平面区域,边界条件是Dirichlet边界条件且Lipschitz连续的,则Poisson方程为

$$\begin{cases} -\Delta u = f, & (x, y) \in \Omega, \\ u = \hat{u}, & (x, y) \in \Gamma, \quad \Gamma = \partial\Omega, \end{cases} \quad (1)$$

其中 f 和 \hat{u} 是已知函数。

通过三角单元 e_h 离散 Ω ,且网格尺寸大小为 h 。用 Λ_h 表示这些元素的集合,则 $\Omega = \bigcup_{e_h \in \Lambda_h} e_h$ 。然后,设 Ω 上的 N 个非重叠子区域剖分为 $\Omega = \bigcup_{i=1}^N \Omega^{(i)}$,另有限元空间定义为

$$S_h = \left\{ \psi_h \in C^0(\Omega) \mid \psi_h|_{e_h} \in P_1(e_h) \right\},$$

可得线性方程组 $\mathbf{K}\mathbf{u} = \mathbf{f}$,其中 \mathbf{K} 是对称正定矩阵, \mathbf{u} 表示 Ω 上的全局自由度, \mathbf{f} 是所有节点处函数值组成的源项向量。

令 $\mathbf{u}^{(i)}$ 是子区域 $\Omega^{(i)}$ 上的所有自由度组成的向量, $\mathbf{R}^{(i)\text{T}}$ 表示从局部自由度 $\mathbf{u}^{(i)}$ 到 Ω 上全局自由度 \mathbf{u} 的0-1矩阵,则有 $\mathbf{u}^{(i)} = \mathbf{R}^{(i)\text{T}}\mathbf{u}$, $i = 1, \dots, N$ 和 $\mathbf{K} = \sum_{i=1}^N \mathbf{R}^{(i)}\mathbf{K}^{(i)}\mathbf{R}^{(i)\text{T}}$,其中 $\mathbf{K}^{(i)}$ 是子区域 $\Omega^{(i)}$ 上形成的局部有限元矩阵。

定义浮动区域集合为

$$\mathcal{G} = \left\{ \Omega^{(i)} \mid \text{dist}(\Omega^{(i)} \cap \Gamma) > 0 \right\},$$

其中 $\text{dist}(Y)$ 表示集合 Y 中任意两点间的最大距离,令 $\mathbf{u}_I^{(i)}$ 表示 $\Omega^{(i)}/\partial\Omega^{(i)}$ 中的自由度向量, $\mathbf{u}_B^{(i)}$ 表示 $\partial\Omega^{(i)}/\Gamma$ 中的自由度向量, $\mathbf{u}_I^{(i)}$ 表示 $\partial\Omega^{(i)} \cap \Gamma$ 中的自由度向量。

当 $\Omega^{(i)} \in \mathcal{G}$,可得 $\mathbf{u}^{(i)}$ 的分解

$$\mathbf{u}^{(i)} = \begin{bmatrix} \mathbf{u}_I^{(i)} & \mathbf{u}_B^{(i)} \end{bmatrix}^{\text{T}}.$$

当 $\Omega^{(i)} \notin \mathcal{G}$,可得 $\mathbf{u}^{(i)}$ 的分解

$$\mathbf{u}^{(i)} = \begin{bmatrix} \mathbf{u}_I^{(i)} & \mathbf{u}_B^{(i)} & \mathbf{u}_I^{(i)} \end{bmatrix}^{\text{T}}.$$

同理,当 $\Omega^{(i)} \in \mathcal{G}$,可得 $\mathbf{K}^{(i)}, \mathbf{f}^{(i)}, \mathbf{R}^{(i)}$ 的分解

$$\mathbf{K}^{(i)} = \begin{pmatrix} \mathbf{K}_{II}^{(i)} & \mathbf{K}_{IB}^{(i)} \\ \mathbf{K}_{IB}^{(i)\text{T}} & \mathbf{K}_{BB}^{(i)} \end{pmatrix}, \quad \mathbf{f}^{(i)} = \begin{pmatrix} \mathbf{f}_I^{(i)} \\ \mathbf{f}_B^{(i)} \end{pmatrix}, \quad \mathbf{R}^{(i)} = \begin{pmatrix} \mathbf{R}_I^{(i)} & 0 \\ 0 & \mathbf{R}_B^{(i)} \end{pmatrix}.$$

当 $\Omega^{(i)} \notin \mathcal{G}$,可得 $\mathbf{K}^{(i)}, \mathbf{f}^{(i)}, \mathbf{R}^{(i)}$ 的分解

$$\mathbf{K}^{(i)} = \begin{pmatrix} \mathbf{K}_{II}^{(i)} & \mathbf{K}_{IB}^{(i)} & \mathbf{K}_{II}^{(i)} \\ \mathbf{K}_{IB}^{(i)\text{T}} & \mathbf{K}_{BB}^{(i)} & \mathbf{K}_{BI}^{(i)} \\ \mathbf{K}_{II}^{(i)\text{T}} & \mathbf{K}_{BI}^{(i)\text{T}} & \mathbf{K}_{II}^{(i)} \end{pmatrix}.$$

注意到 $\mathbf{u}_B^{(i)} = \mathbf{R}_B^{(i)\text{T}}\mathbf{u}_B$,将式(1)的整体自由度划分为区域内部自由度和区域间边界自由度,得

$$\begin{pmatrix} \mathbf{K}_{II}^{(1)} & 0 & \cdots & 0 & \mathbf{K}_{IB}^{(1)}\mathbf{R}_B^{(1)\text{T}} \\ & \mathbf{K}_{II}^{(2)} & \cdots & & \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & \mathbf{K}_{II}^{(N)} & \mathbf{K}_{IB}^{(N)}\mathbf{R}_B^{(N)\text{T}} \\ \mathbf{R}_B^{(1)}\mathbf{K}_{IB}^{(1)\text{T}} & \cdots & \mathbf{R}_B^{(N)}\mathbf{K}_{IB}^{(N)\text{T}} & \sum_{i=1}^N \mathbf{R}_B^{(i)}\mathbf{K}_{BB}^{(i)}\mathbf{R}_B^{(i)\text{T}} & \end{pmatrix} \begin{pmatrix} \mathbf{u}_I^{(1)} \\ \mathbf{u}_I^{(2)} \\ \vdots \\ \mathbf{u}_I^{(N)} \\ \mathbf{u}_B \end{pmatrix} = \begin{bmatrix} \tilde{\mathbf{f}}_I^{(1)} & \cdots & \tilde{\mathbf{f}}_I^{(N)} & \sum_{i=1}^N \mathbf{R}_B^{(i)}\tilde{\mathbf{f}}_B^{(i)} \end{bmatrix}^{\text{T}}, \quad (2)$$

其中

$$\tilde{\mathbf{f}}_I^{(i)} = \begin{cases} \mathbf{f}_I^{(i)}, & \Omega^{(i)} \in \mathcal{G}, \\ \mathbf{f}_I^{(i)} - \mathbf{K}_{II}^{(i)}\mathbf{u}_I^{(i)}, & \Omega^{(i)} \notin \mathcal{G}, \end{cases}$$

$$\tilde{\mathbf{f}}_B^{(i)} = \begin{cases} \mathbf{f}_B^{(i)}, & \Omega^{(i)} \in \mathcal{G}, \\ \mathbf{f}_B^{(i)} - \mathbf{K}_{BI}^{(i)}\mathbf{u}_I^{(i)}, & \Omega^{(i)} \notin \mathcal{G} \end{cases}$$

由式(2)可得子区域内部自由度的关系式

$$\mathbf{u}_I^{(i)} = \mathbf{K}_{II}^{(i)-1} \left(\tilde{\mathbf{f}}_I^{(i)} - \mathbf{K}_{IB}^{(i)}\mathbf{u}_B \right). \quad (3)$$

同理可得子区域间边界上自由度的关系式

$$\begin{aligned} & \left\{ \sum_{i=1}^N \mathbf{R}_B^{(i)} \left[\mathbf{K}_{BB}^{(i)} - \mathbf{K}_{IB}^{(i)\text{T}} \left(\mathbf{K}_{II}^{(i)} \right)^{-1} \mathbf{K}_{IB}^{(i)} \right] \mathbf{R}_B^{(i)\text{T}} \right\} \mathbf{u}_B \\ & = \sum_{i=1}^N \mathbf{R}_B^{(i)} \left[\tilde{\mathbf{f}}_B^{(i)} - \mathbf{K}_{IB}^{(i)\text{T}} \left(\mathbf{K}_{II}^{(i)} \right)^{-1} \tilde{\mathbf{f}}_I^{(i)} \right], \end{aligned} \quad (4)$$

式(4)可简写为

$$\mathbf{S}\mathbf{u}_B = \mathbf{g}, \quad (5)$$

式中 \mathbf{S} 是舒尔补的集合:

$$\mathbf{S} = \sum_{i=1}^N \mathbf{R}_B^{(i)}\mathbf{S}^{(i)}\mathbf{R}_B^{(i)\text{T}}, \quad (6)$$

这里

$$\mathbf{S}^{(i)} = \mathbf{K}_{BB}^{(i)} - \mathbf{K}_{IB}^{(i)\text{T}} \left(\mathbf{K}_{II}^{(i)} \right)^{-1} \mathbf{K}_{IB}^{(i)}. \quad (7)$$

当 $\Omega^{(i)} \in \mathcal{G}$, $\mathbf{S}^{(i)}$ 是奇异的。用 V 表示式(5)的全局自由度的解空间, $V^{(i)}$ 表示子区域界面上自由度构成的空间,则可将矩阵解释成映射

$$\mathbf{S}: V \rightarrow V, \quad \mathbf{S}^{(i)}: V^{(i)} \rightarrow V^{(i)}, \quad \mathbf{R}_B^{(i)}: V^{(i)} \rightarrow V.$$

根据式(2)和(4),当求解出 \mathbf{u}_B 之后,由式(3)可得到 \mathbf{u}_I 。此外,从式(5)~(7)观察到,由于每个子区域上的矩阵互不关联,因此DDM可并行计算。

1.2 预处理矩阵构造

本文中采用预处理共轭梯度方法迭代求解式(5), 此方法要求每一步都解决一个辅助问题 $\mathbf{M}\mathbf{z} = \mathbf{r}$, 其中 \mathbf{M} 是一个对称正定矩阵, 称为预条件子; \mathbf{r} 为预处理前的残差, \mathbf{z} 为预处理后的残差。

Neumann-Neumann 预条件子是 DDM 中的一种预处理方法, 它有如下的形式:

$$\mathbf{M}_{N-N}^{-1} = \sum_{i=1}^N \mathbf{R}_B^{(i)} \mathbf{D}^{(i)} \mathbf{S}^{(i)\dagger} \mathbf{D}^{(i)\top} \mathbf{R}_B^{(i)\top},$$

$$\mathbf{S}^{(i)\dagger} = \begin{cases} \mathbf{S}^{(i)-1}, & \Omega^{(i)} \notin \mathcal{G}, \\ \left(\mathbf{S}^{(i)} + \alpha \max[\text{diag}(\mathbf{S}^{(i)})] \cdot \mathbf{E} \right)^{-1}, & \Omega^{(i)} \in \mathcal{G}, \end{cases}$$

其中 \mathbf{E} 是单位矩阵, α 是一个很小的常数, 本文取 $\alpha = 1 \times 10^{-10}$ 。 $\mathbf{D}^{(i)}$ 是子区域的权重函数, 满足

$$\sum_{i=1}^N \mathbf{R}_B^{(i)} \mathbf{D}^{(i)} \mathbf{R}_B^{(i)\top} = \mathbf{E},$$

其中 $\mathbf{D}^{(i)}$ 可取为对角矩阵, 其对角元素等于与自由度相关的子域数量的倒数。

然而, Neumann-Neumann 预条件子缺乏子域间信息交换机制, 且存在浮动区域引起的奇异性问题。随着子域数量的增加, 矩阵终会变成病态, 这表明它不是一个大规模问题的高效预处理技术。为了阻止误差的传播, 在 Neumann-Neumann 预条件子的基础上, BDD 应用粗网格修正以减少对部分区域数量的依赖。

令 $\mathbf{Z}^{(i)}$ 是一个列满秩矩阵, 满足 $\text{Null } \mathbf{S}^{(i)} \subset \text{Range } \mathbf{Z}^{(i)}$, 全局粗空间定义为

$$\mathbf{W} = \left\{ v \in V \mid v = \sum_{i=1}^N \mathbf{R}_B^{(i)} \mathbf{D}^{(i)} v^{(i)}, v^{(i)} \in \text{Range } \mathbf{Z}^{(i)} \right\}.$$

则 BDD 算法的实现如下:

1) 通过求解未知向量 $\bar{\lambda}_w \in \mathbb{R}^N$ 的粗问题来平衡原始残差 \mathbf{r} , 有

$$\mathbf{S}_w \bar{\lambda}_w = \mathbf{R}_w^\top \mathbf{r}.$$

2) 使用 $\bar{\lambda}_w$ 对残差向量进行粗网格修正, 有

$$\tilde{\mathbf{r}} = \mathbf{r} - \mathbf{S} \mathbf{R}_w \bar{\lambda}_w.$$

3) 求解 Neumann-Neumann 问题以及这些结果的平均值

$$\tilde{\mathbf{u}} = \sum_{i=1}^N \mathbf{R}_B^{(i)} \mathbf{D}^{(i)} \mathbf{S}^{(i)\dagger} \mathbf{D}^{(i)\top} \mathbf{R}_B^{(i)\top} \tilde{\mathbf{r}}.$$

4) 计算新的残差向量 $\hat{\mathbf{r}} = \mathbf{r} - \mathbf{S} \tilde{\mathbf{u}}$ 。

5) 对于未知向量 $\bar{\mu}_w \in \mathbb{R}^N$, 再解粗问题 $\mathbf{S}_w \bar{\mu}_w = \mathbf{R}_w^\top \hat{\mathbf{r}}$ 。

6) 求得预处理向量 $\mathbf{z} = \tilde{\mathbf{u}} + \mathbf{R}_w \bar{\mu}_w$, 可得 BDD 的预处理矩阵为 $\mathbf{M}_{\text{BDD}}^{-1} = (\mathbf{P} + (\mathbf{E} - \mathbf{P}) \mathbf{T}_{N-N} (\mathbf{E} - \mathbf{P})) \mathbf{S}^{-1}$, 其中

$$\mathbf{P} = \mathbf{R}_w \mathbf{S}_w^{-1} \mathbf{R}_w^\top, \quad \mathbf{S}_w = \mathbf{R}_w^\top \mathbf{S} \mathbf{R}_w,$$

$$\mathbf{R}_w = (\mathbf{R}_B^{(1)} \mathbf{D}^{(1)} \mathbf{Z}^{(1)}, \dots, \mathbf{R}_B^{(N)} \mathbf{D}^{(N)} \mathbf{Z}^{(N)}),$$

$$\mathbf{T}_{N-N} = \sum_{i=1}^N \mathbf{R}_B^{(i)} \mathbf{D}^{(i)} \mathbf{S}^{(i)\dagger} \mathbf{D}^{(i)\top} \mathbf{R}_B^{(i)\top}.$$

上述预处理矩阵的构造过程参考文献 (Mandel, 1993; Mukaddes et al., 2006)。本文采用的并行算法是 SMP 结构的, 这有别于分层次区域分解算法 (Yagawa et al., 1993)。在数值模拟时, $\mathbf{Z}^{(i)} = (1, \dots, 1)^\top, i = 1, \dots, N$ 。

2 PDE 的有限元离散

本节首先讨论 BDD 数值可扩展性。根据文献 (Farhat et al., 1995; Toselli et al., 2004), 当线性系统的子域个数增加时, 其收敛速度不降低, 则称该方法具有数值可扩展性。这通常意味着当子域 H 变小时, 收敛性不会恶化, 这也保证了 BDD 的并行可扩展性。

对比运用 DDM 和 BDD 方法求解三类 PDEs 的迭代次数的效率和内存消耗量。椭圆型、抛物型和双曲型 PDE 的代表性方程为 Poisson 方程、热传导方程和波动方程。定义 $\text{ite}_{\text{ratio}} = \frac{\text{ite}_{\text{BDD}}}{\text{ite}_{\text{DDM}}}$, ite_{BDD} 和 ite_{DDM} 分别表示 BDD 和 DDM 的迭代次数。 $\text{ite}_{\text{ratio}}$ 越小, 说明 BDD 方法对此类 PDE 求解更高效。

下面, 介绍热传导方程和波动方程的有限元离散。热传导方程

$$\frac{\partial u}{\partial t} = \Delta u + f, (x, y) \in \Omega, t \in (0, T]. \quad (8)$$

波动方程

$$\frac{\partial^2 u}{\partial t^2} = \Delta u + f, (x, y) \in \Omega, t \in (0, T]. \quad (9)$$

给定边界条件和初始条件

$$u = \hat{u}, (x, y) \in \Gamma, \Gamma = \partial\Omega, t \in (0, T],$$

$$u = u^0, (x, y) \in \Omega, t = 0.$$

记空间

$$\bar{S} = \left\{ v; \iint_{\Omega} \left[v^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right] dx dy \text{ 有意义,} \right.$$

$$\left. \text{且在 } \partial\Omega \text{ 上 } v = \bar{u} \right\}.$$

则式(8)和(9)的 Galerkin 变分形式是找到 $u \in \bar{S}$ 使得

$$\iint_{\Omega} \frac{\partial u}{\partial t} v dx dy + \iint_{\Omega} \nabla u \cdot \nabla v dx dy = \iint_{\Omega} f v dx dy, \quad (10)$$

$$\iint_{\Omega} \frac{\partial^2 u}{\partial t^2} v dx dy + \iint_{\Omega} \nabla u \cdot \nabla v dx dy = \iint_{\Omega} f v dx dy. \quad (11)$$

经过有限元离散, 有限元空间定义为

$$\bar{S}_h = \left\{ \phi_h \in C^0(\Omega) \mid \phi_h|_{e_h} \in P_1(e_h) \right\}, e_h \in A_h.$$

令 N^* 是一个正整数, 有 $\Delta t = T/N^*$, $t^n = n\Delta t$, $n = 1, \dots, N^*$. 令 $\varphi_i (i = 1, \dots, \bar{N})$ 是有限元空间 \bar{S}_h 的基函数, 其中 \bar{N} 为节点个数. 则 u 的离散数值解 $u_h^n = \sum_{i=1}^{\bar{N}} U_i^n \varphi_i \in \bar{S}_h$, $\varphi_i^n = \varphi_i(t^n)$, 其中 U_i^n 是在 $t = t^n$ 时第 i 个节点的数值解.

使用向后 Euler 格式离散公式(10)和(11)为

$$\begin{aligned} & \int_{\Omega} \frac{u_h^{n+1} - u_h^n}{\Delta t} v_h dx dy + \int_{\Omega} \nabla u_h^{n+1} \cdot \nabla v_h dx dy \\ &= \int_{\Omega} f_h^{n+1} v_h dx dy, \quad v_h \in \bar{S}_h, \\ & \int_{\Omega} \frac{u_h^{n+1} - 2u_h^n + u_h^{n-1}}{\Delta t^2} v_h dx dy + \int_{\Omega} \nabla u_h^{n+1} \cdot \nabla v_h dx dy \\ &= \int_{\Omega} f_h^{n+1} v_h dx dy, \quad v_h \in \bar{S}_h. \end{aligned}$$

令

$$\mathcal{M} = (\mathcal{M}_{ij})_{\bar{N} \times \bar{N}}, \quad \mathcal{S} = (\mathcal{S}_{ij})_{\bar{N} \times \bar{N}}, \quad \mathcal{F}(t) = (\mathcal{F}_i(t))_{\bar{N} \times 1},$$

其中

$$\begin{aligned} \mathcal{M}_{ij} &= \int_{\Omega} \varphi_i \varphi_j dx dy, \\ \mathcal{S}_{ij} &= \int_{\Omega} \nabla \varphi_i \cdot \nabla \varphi_j dx dy, \\ \mathcal{F}_i(t) &= \int_{\Omega} f_h(t) \varphi_i dx dy. \end{aligned}$$

设 $\mathcal{F}^{n+1} = \mathcal{F}(t^{n+1})$. 可得最后的矩阵形式

$$\left[\frac{1}{\Delta t} \mathcal{M} + \mathcal{S} \right] U^{n+1} = \mathcal{F}^{n+1} + \frac{1}{\Delta t} \mathcal{M} U^n,$$

$$\left[\frac{1}{\Delta t^2} \mathcal{M} + \mathcal{S} \right] U^{n+1} = \mathcal{F}^{n+1} + \frac{1}{\Delta t^2} \mathcal{M} (2U^n - U^{n-1}).$$

然后用BDD求解。

3 数值结果

本章进行数值测试, 然后通过数值实验验证 BDD 的数值可扩展性, 比较 DDM 和 BDD 的迭代效率. 所有数值实验中, 共轭梯度方法的收敛阈值取为 1×10^{-6} .

3.1 数值测试

本节固定 H/h 为 100, 改变节点个数, 得到迭代次数的变化, 以验证程序的可靠性. 设置它们的源项和边界为

$$f = \pi^2(\sin \pi x + \sin \pi y), \quad \hat{u} = 1, \quad (x, y) \in \partial \Omega$$

且热传导方程和波动方程的初始条件为 $u^0 = 1, t = 0$.

在热传导方程中 $T = 0.1, \Delta t = 0.01$; 而在波动方程中 $T = 1, \Delta t = 0.1$. 计算区域为 $[0, 3] \times [0, 3]$, 在 x 和 y 方向的网格剖分数分别用 h_{N_x}, h_{N_y} 表示, 在 x 和 y 方向的子区域数分别用 H_{N_x}, H_{N_y} 表示. 表 1 为 BDD 计算 Poisson 方程、热传导方程和波动方程的迭代次数. 表 1 中, BDD 算法的 Poisson、热传导和波动方程的迭代次数趋于平稳, 这验证了 BDD 中预处理后矩阵的条件数只与 H/h 的界相关.

表 1 Poisson 方程、热传导方程和波动方程的迭代次数

Table 1 The number of iterations of the Poisson equation, the heat conduction equation, and the wave equation

网络剖分和子区域数	Poisson 方程迭代次数	热传导方程迭代次数	波动方程迭代次数
$h_{N_x} = h_{N_y} = 300, \quad H_{N_x} = H_{N_y} = 3$	2	30	27
$h_{N_x} = h_{N_y} = 500, \quad H_{N_x} = H_{N_y} = 5$	4	40	36
$h_{N_x} = h_{N_y} = 600, \quad H_{N_x} = H_{N_y} = 6$	6	50	45
$h_{N_x} = h_{N_y} = 1000, \quad H_{N_x} = H_{N_y} = 10$	7	60	54
$h_{N_x} = h_{N_y} = 1200, \quad H_{N_x} = H_{N_y} = 12$	7	60	54
$h_{N_x} = h_{N_y} = 1500, \quad H_{N_x} = H_{N_y} = 15$	6	60	54
$h_{N_x} = h_{N_y} = 2000, \quad H_{N_x} = H_{N_y} = 20$	6	63	56

3.2 BDD 的数值可扩展性及效率

本节用 Poisson、热传导和波动方程来验证 BDD 的数值可扩展性及效率. 子区域数量和迭代次数、内存消耗量的关系如图 1-2 所示. 小模型参数设置为: 计算区域为 $[0, 3] \times [0, 3]$, 在 x 和 y 方向使用 1000 剖分, 网格离散尺寸 $h = 0.003$, 全局节

点个数 $\bar{N} = 1002001$, 区域内边界上的自由度为 17901, 离散单元个数为 2000000; 大模型的计算区域为 $[0, 3] \times [0, 3]$, 在 x 和 y 方向使用 2000 剖分, 网格离散尺寸 $h = 0.0015$, 全局节点个数 $\bar{N} = 4004001$, 区域内边界上的自由度 75601, 离散单元个数为 8000000.

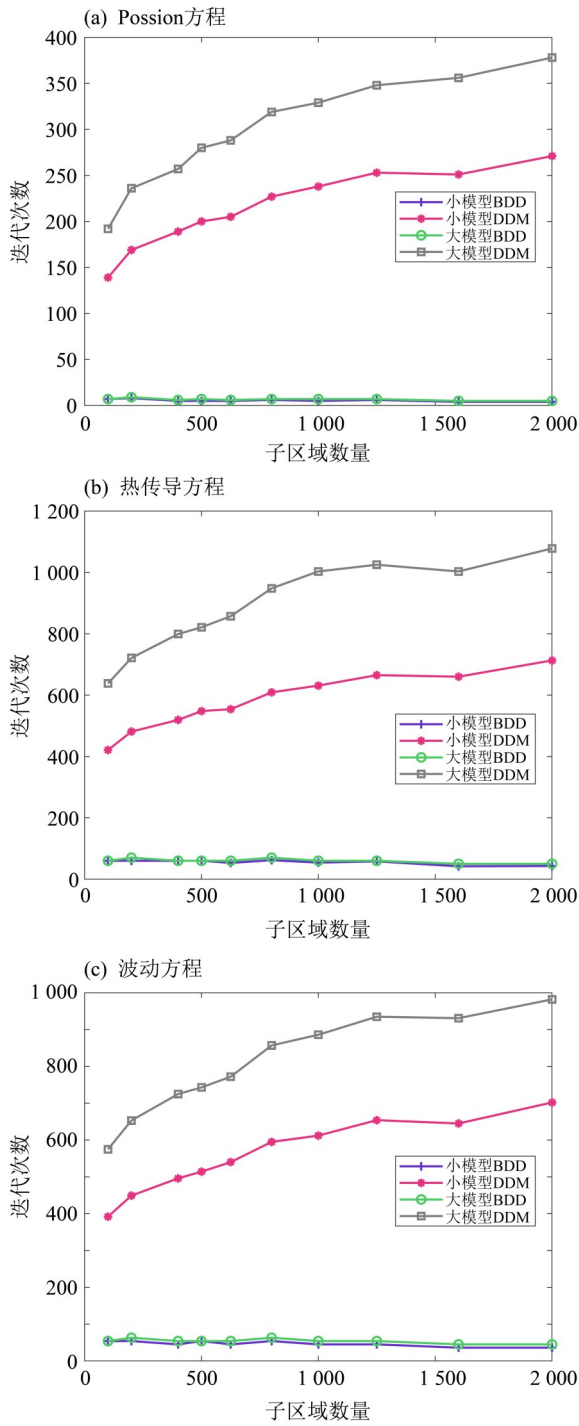


图1 子区域数量和迭代次数的关系

Fig. 1 The relationship between the number of subdomains and the number of iterations

从图 1 可知,随着子区域数量的增加,BDD 相比于 DDM 有很好的数值可扩展性。从图 2 可观察到,由于预处理矩阵构造的复杂性,导致 BDD 比 DDM 的内存消耗量大。另外,DDM 的内存消耗量比较稳定;而由于本文的并行算法是 SMP 结构的,随着子区域数的增加,子区域间的通讯信息量增加,因此 BDD 的内存消耗量逐渐增大。

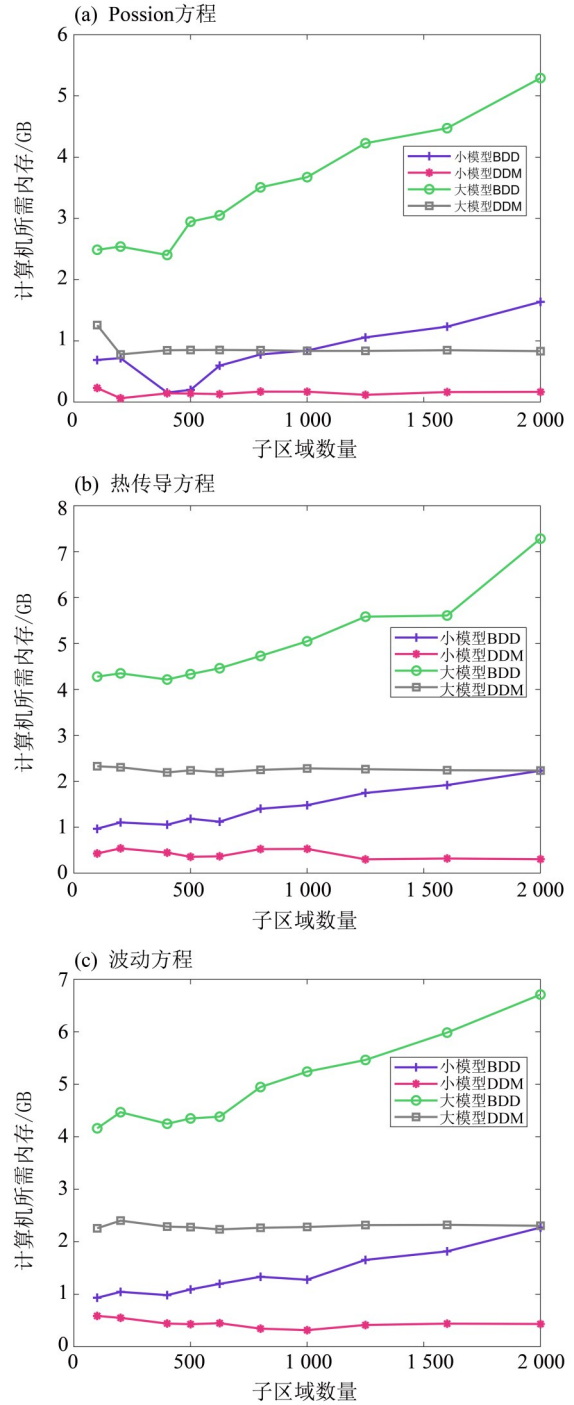


图2 子区域数量和内存消耗量的关系

Fig. 2 The relationship between the number of subdomains and memory consumption

DDM 和 BDD 的迭代次数比值如图 3 所示。从图中可以看出,在运用 BDD 提高迭代效率上, Poisson 方程比热传导和波动方程有更好的效果。这是由于在数值实验部分热传导方程中 $\Delta t = 0.01$, 波动方程中 $\Delta t = 0.1$, 热传导方程和波动方程的系数矩阵是对角占优的,所以使用 BDD 时 Poisson 方程的效率更好。

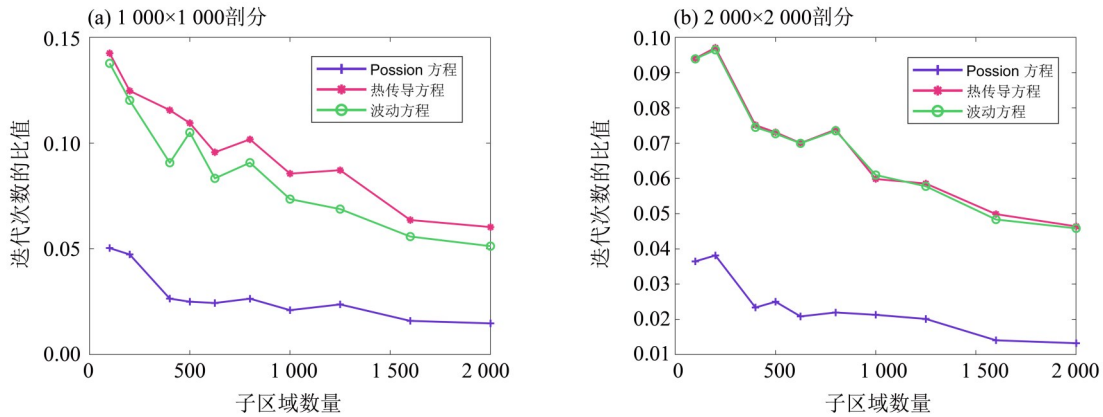


图3 DDM与BDD迭代次数的比值

Fig. 3 The ratio of the iterations of DDM and BDD

3.3 扩散反应方程

本节使用扩散反应方程来验证BDD的高效性。

考虑控制方程

$$\begin{cases} \frac{\partial u}{\partial t} - \epsilon \Delta u + \gamma u = f, & (x, y) \in \Omega, t \in (0, T], \\ u = u^0, & (x, y) \in \Omega, t = 0, \\ u = \bar{u}, & (x, y) \in \Gamma, \Gamma = \partial\Omega, t \in (0, T], \end{cases}$$

其中 ϵ 和 γ 是常数。空间离散使用有限元离散,时间离散使用向后差分稳定格式。

3.3.1 单个热源 设单个热源的扩散反应方程参

数 $T = 2, \Delta t = 0.01, \epsilon = 0.1, \gamma = 0.1$, 初始条件设置为

$$u^0 = \begin{cases} 1, & (x - 1.5)^2 + (y - 1.5)^2 < 1^2, \\ 0, & \text{其他.} \end{cases}$$

边界条件设置为

$$\bar{u}|_{x=0} = 0, \bar{u}|_{x=3} = 1, \bar{u}|_{y=0} = 0, \bar{u}|_{y=3} = 0.$$

单个热源的数值模拟结果如图4所示。其中,计算区域为 $[0, 3] \times [0, 3]$, 在 x 和 y 方向使用 2 000 剖分, 网格离散尺寸 $h = 0.0015$, 子区域个数为 $N = 400$, 全局节点个数 $\bar{N} = 4\,004\,001$, 区域内边界上的自由度 75 601, 离散单元个数为 8 000 000。

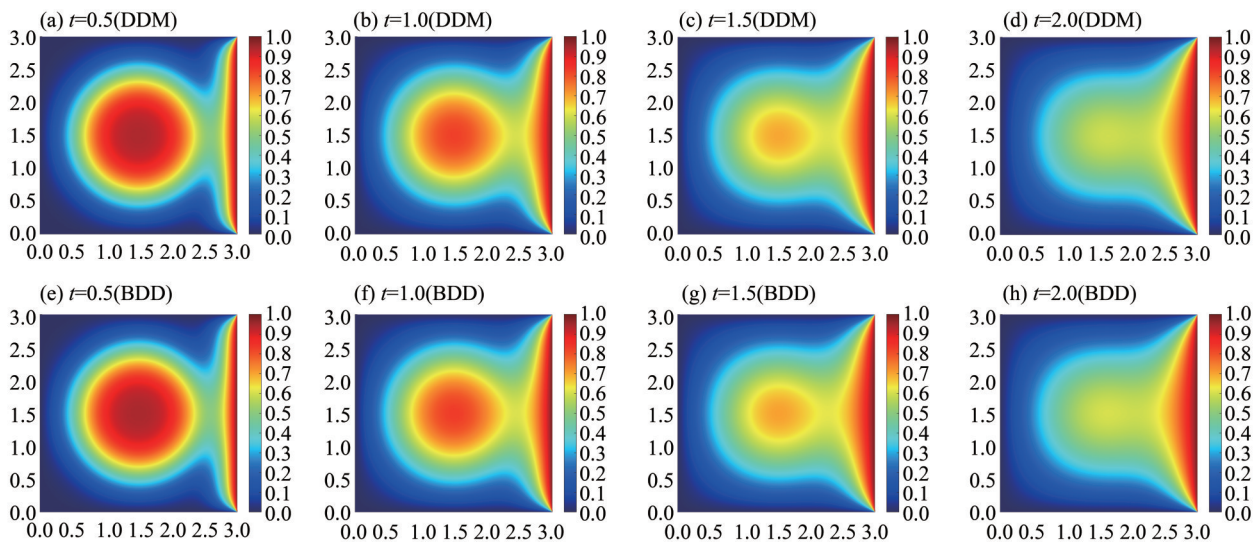


图4 单个热源的数值模拟结果

Fig. 4 Numerical simulation results for a single heat source

实验中,DDM的总迭代步数为 6 351, CPU 时间为 60 738 s; BDD的总迭代步数为 1 000, CPU 时间为 23 329 s; 与图4相结合可以看出, BDD 相比于 DDM 是一种更高效的数值方法。

3.3.2 多个热源 设多个热源的扩散反应方程参

数 $T = 2, \Delta t = 0.01, \epsilon = 0.05, \gamma = 0.1$, 初始条件设置为

$$u^0 = \begin{cases} 1, & (x - 0.75)^2 + (y - 1.5)^2 < 0.5^2, \\ 1, & (x - 2.25)^2 + (y - 1.5)^2 < 0.5^2, \\ 0, & \text{其他.} \end{cases}$$

边界条件设置为

$$\begin{aligned} \bar{u}|_{x=0} &= 1, \quad \bar{u}|_{x=3} = 2, \\ \bar{u}|_{y=0} &= 0, \quad \bar{u}|_{y=3} = 0, \quad t > 0. \end{aligned}$$

多个热源的数值模拟结果如图5所示。其

中, 计算区域为 $[0, 3] \times [0, 3]$, 在 x 和 y 方向都用 2 500 剖分, 网格离散尺寸 $h = 0.0012$, 子区域个数为 $N = 625$, 全局节点个数 $\bar{N} = 6\,255\,001$, 区域内边界上的自由度 119 424, 离散单元个数为 12 500 000。

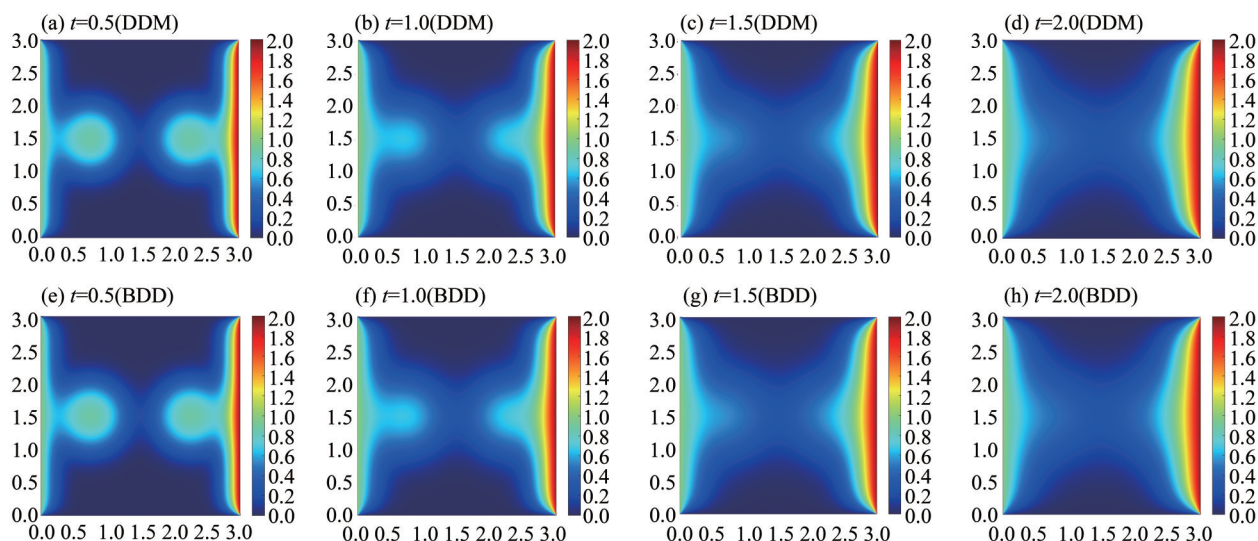


图5 多个热源的数值模拟结果

Fig. 5 Numerical simulation results of multiple heat sources

实验中, DDM的总迭代步数为 5 978, CPU时间为 68 998 s; BDD的总迭代步数为 1 000, CPU时间为 31 456 s。与图5相结合可以看出, 在多个热源的扩散反应方程模拟实验中, BDD相比于DDM是一种面向大规模的、可并行的、高效率的数值方法。

4 结论

利用非重叠的DDM和BDD, 探讨了以Poisson、热传导和波动方程为代表的PDEs数值求解效率与内存消耗量。结论如下: 在采用BDD提

升迭代效率方面, $1\,000 \times 1\,000$ 和 $2\,000 \times 2\,000$ 剖分条件下, Poisson方程相较于热传导和波动方程均表现出更为显著的效果。由于本研究采用基于SMP结构的并行算法, 其特点是所有处理器单元共享内存。因此, 随着子区域数量的增加, 子区域间的通信信息量也随之增大, 这导致BDD相比DDM在内存消耗上有所增加。扩散反应方程的数值模拟, 进一步验证了BDD是一种比DDM更为高效的数值方法, 为复杂PDEs的高效求解提供了重要的理论依据。

参考文献:

- 李佳伟, 2020. 基于格子 Boltzmann 通量求解器的高超声速流-固-热多场耦合数值模拟方法研究[D]. 南京: 南京航空航天大学.
- 秦启豪, 2020. 典型超/高超声速流场的多物理场耦合研究[D]. 南京: 南京航空航天大学.
- 孙学功, 龚春叶, 2015. 高超声速飞行器并行仿真方法研究[J]. 系统仿真学报, 27(5): 948-958+964.
- 张佳晨, 2020. 高超声速流场的辐射加热特性数值模拟研究[D]. 杭州: 浙江大学.
- 赵航, 2019. 直流电阻率平衡区域分解算法的计算效率影响因素分析[J]. 江西科学, 37(2): 170-174.

- BEVILACQUA T, 2024. Balancing domain decomposition by constraints preconditioners for virtual element discretizations of saddle-point problems [EB/OL]. <https://air.unimi.it/handle/2434/1022953>.
- BJORSTAD P E, WIDLUND O B, 1989. To overlap or not to overlap: A note on a domain decomposition method for elliptic problems[J]. SIAM J Sci Stat Comput, 10(5): 1053-1061.
- BOURGAT J F, GLOWINSKI R, LE TALLEC P, et al, 1988. Variational formulation and algorithm for trace operation in domain decomposition calculations[D]. Paris: INRIA

- (Institut national de recherche en informatique et en automatique).
- COWSAR L C, MANDEL J, WHEELER M F, 1995. Balancing domain decomposition for mixed finite elements[J]. *Math Comput*, 64(211):989–1015.
- DOHRMANN C R, 2003. A preconditioner for substructuring based on constrained energy minimization [J]. *SIAM J Sci Comput*, 25(1):246–258.
- FARHAT C, CHEN P S, MANDEL J, 1995. A scalable Lagrange multiplier based domain decomposition method for time-dependent problems[J]. *Int J Numer Meth Eng*, 38(22):3831–3853.
- FARHAT C, MANDEL J, ROUX F X, 1994. Optimal convergence properties of the FETI domain decomposition method [J]. *Comput Meth Appl Mech Eng*, 115(3/4):365–385.
- FARHAT C, ROUX F X, 1991. A method of finite element tearing and interconnecting and its parallel solution algorithm [J]. *Int J Numer Meth Eng*, 32(6):1205–1227.
- HUYNH N M M, CHEGINI F, PAVARINO L F, et al, 2023. Convergence analysis of BDDC preconditioners for composite DG discretizations of the cardiac cell-by-cell model [J]. *SIAM J Sci Comput*, 45(6):A2836–A2857.
- KANAYAMA H, OGINO M, SUGIMOTO S I, et al, 2022. A condition for positive-definiteness of the coarse matrix in BDD-DIAG of a perturbed magnetostatic problem[J]. *J Adv Simul Sci Eng*, 9(2):263–277.
- LEE C O, PARK J, 2021. A dual-primal finite element tearing and interconnecting method for nonlinear variational inequalities utilizing linear local problems [J]. *Int J Numer Meth Eng*, 122(22):6455–6475.
- LI C, CUI N, WANG C, et al, 2021. Reduced-order electrochemical model for lithium-ion battery with domain decomposition and polynomial approximation methods[J]. *Energy*, 221:119662.
- MANDEL J, 1993. Balancing domain decomposition [J]. *Commun Numer Meth Eng*, 9(3):233–241.
- MANDEL J, DOHRMANN C R, 2003. Convergence of a balancing domain decomposition by constraints and energy minimization [J]. *Numer Linear Algebra Appl*, 10(7):639–659.
- MIYAMURA T, YOSHIMURA S, 2023. Balancing domain decomposition method for large-scale analysis of an assembly structure having millions of multipoint constraints [J]. *Comput Meth Appl Mech Eng*, 405:115846.
- MUKADDES A M M, OGINO M, KANAYAMA H, et al, 2006. A scalable balancing domain decomposition based preconditioner for large scale heat transfer problems [J]. *JSME Int J Ser B*, 49(2):533–540.
- PACHECO D R Q, STEINBACH O, 2022. Space-time finite element tearing and interconnecting domain decomposition methods [M]. *Domain Decomposition Methods in Science and Engineering XXVI*. Cham, Springer:479–486.
- TOSELLI A, WIDLUND O, 2004. Domain decomposition methods—algorithms and theory [M]. Berlin, Heidelberg: Springer Science and Business Media.
- VEVEK U S, ELHADIDI B, CHAN W L, 2021. A domain decomposition technique for small amplitude wave interactions with shock waves [J]. *J Comput Phys*, 437:110326.
- YAGAWA G, SHIOYA R, 1993. Parallel finite elements on a massively parallel computer with domain decomposition [J]. *Comput Syst Eng*, 4(4/5/6):495–503.
- YAO Q, ZHU H, 2015. Numerical simulation of hydrogen dispersion behaviour in a partially open space by a stabilized balancing domain decomposition method [J]. *Comput Math Appl*, 69(10):1068–1079.
- YAO Q, ZHU H, 2021. Domain decomposition modeling of carotid artery stenosis based on 3D rotational angiography [J]. *Adv Mech Eng*, 13(5):16878140211018069.
- ZHU H, YAO Q, KANAYAMA H, 2016. Large-scale computations of flow around two cylinders by a domain decomposition method [J]. *Math Probl Eng*, 2016(1):4126123.

(责任编辑 王海蓉)