arXiv:2301.01426v1 [math.NA] 4 Jan 2023

求解非对称或不定椭圆问题的迭代两层算法

Ming Tang[a], Xiaoqing Xing[a], Ying Yang[b,*], Liuqiang Zhong[a]

[a]*School of Mathematical Sciences, South China Normal University, Guangzhou 510631, China*
[b]*School of Mathematics and Computational Science, Guangxi Colleges and Universities Key Laboratory of Data Analysis and Computation, Guilin University of Electronic Technology, Guilin 541004, China*

摘要

本文提出了一种新的迭代两层算法，用于求解非对称或不定椭圆问题的有限元离散系统。该迭代两层算法采用与传统双网格算法相同的粗空间，但其"细空间"是在粗网格上使用更高阶有限元空间。因此，迭代两层算法只需一个网格，计算成本远低于传统迭代双网格算法。最后，与传统双网格算法相比，数值实验表明，要达到相同的收敛阶，该算法的计算成本更低。

关键词： 迭代两层算法、有限元离散、非对称或不定椭圆问题

1. 引言

我们考虑一般二阶椭圆型偏微分方程的以下狄利克雷边值问题

$$-\mathrm{div}(\boldsymbol{\alpha}(x)\nabla u) + \boldsymbol{\beta}(x)\cdot\nabla u + \gamma(x)u = f \qquad \text{in } \Omega\ , \tag{1.1}$$

$$u = 0 \qquad \text{on } \partial\Omega\ , \tag{1.2}$$

其中，系数 $\boldsymbol{\alpha}(x) \in \mathbb{R}^d \times \mathbb{R}^d$ 是定义在 $\bar{\Omega}$ 上的光滑函数，并满足一致椭圆性，即存在最小和最大特征值 α_* 和 α^*，分别满足 $\alpha_*|\xi|^2 \leqslant \alpha(x)\xi\cdot\xi \leqslant \alpha^*|\xi|^2, \quad \forall \xi \in \mathbb{R}^d$。$\boldsymbol{\beta}(x) \in \mathbb{R}^d$ 和 $\gamma(x) \in \mathbb{R}^1$ 也都是定义在 $\bar{\Omega}$ 上的光滑函数，而$f \in L^2(\Omega)$ 是给定函数。注意，当 $\boldsymbol{\beta}(x) \neq 0$时，(1.1)-(1.2) 的连续变分问题（CVP）是非对称的；当 $\gamma(x) < 0$时，(1.1)-(1.2) 的 CVP 可能是不定的。

*通讯作者*Email addresses:* mingtang@m.scnu.edu.cn (Ming Tang)、 xingxq@scnu.edu.cn (Xiaoqing Xing)、yangying@lsec.cc.ac.cn (Ying Yang)、 zhong@scnu.edu.cn (Liuqiang Zhong)

双网格（TG）算法最初由 Xu [6] 提出，用于求解非对称或不定问题。该算法的主要思想是先在粗网格上求解原问题，以获得近似有限元解，然后利用该近似解在细网格上求解相应的线性对称正定问题。在过去二十年中，双网格（TG）算法已广泛用于求解许多问题，例如非线性椭圆问题 [1, 10], 半线性抛物方程 [5], 非线性抛物方程 [3, 2],泊松-能斯特-普朗克问题 [7],和麦克斯韦方程 [8, 9]。

本文的主要目标是提出一种新的迭代两层算法。与传统的迭代双网格算法相比，我们的算法只需使用一个网格，并且以更少的 CPU 时间达到相同的精度。

在本文中，我们首先介绍采用有限元方法离散的椭圆问题。然后，提出一种迭代两层算法。最后，给出一些数值结果，以说明所提算法的效率。

2. 有限元离散

给定 $S \subset \mathbb{R}^d (d = 2, 3)$，记 $W^{1,2}(S)$ 为带范数 $\|\cdot\|_{1,2,S}$ 的标准索伯列夫空间。为简化记号，记 $\|\cdot\|_1 = \|\cdot\|_{1,2,\Omega}$，并将 $H_0^1(\Omega) := \{u \in H^1(\Omega) : u|_{\partial\Omega} = 0\}$ 视为迹意义下的空间。定义以下两个双线性型

$$a(u,v) := \int_\Omega (\boldsymbol{\alpha}(x)\nabla u)\cdot\nabla v \mathrm{d}x, \tag{2.1}$$

$$\hat{a}(u,v) := a(u,v) + \int_\Omega (\boldsymbol{\beta}(x)\cdot\nabla u + \gamma(x)u)v\mathrm{d}x. \tag{2.2}$$

于是，我们得到 (1.1)-(1.2) 的 CVP：求 $u \in H_0^1(\Omega)$，使得

$$\hat{a}(u,v) = (f,v), \quad \forall v \in H_0^1(\Omega). \tag{2.3}$$

假设 Ω 被拟一致剖分 $\mathcal{T}_h = \{\tau_i\}$ 划分。这里指 τ_i' 是大小为 h 的单纯形，其中 $h \in \in (0, 1)$，且 $\Omega = \cup_i \tau_i$。对于给定的拟一致剖分 $\mathcal{T}_h$，协调有限元空间定义如下

$$V_h^l := \{v \in C(\Omega) : v|_\tau \in \mathbb{P}_l(\tau),\ \forall \tau \in \mathcal{T}_h,\ v|_{\partial\Omega} = 0\},$$

其中，$\mathbb{P}_l(\tau)$ 是次数不高于正整数 l 的多项式空间，定义在子剖分单元 τ 上。式 (2.3) 的离散变分问题是求 $u_h^l \in V_h^l$，使得

$$\hat{a}(u_h^l, v_h^l) = (f, v_h^l), \quad \forall v_h^l \in V_h^l. \tag{2.4}$$

3. 迭代两层算法

经典迭代双网格算法的基本机制是 Ω 的两个拟一致四面体嵌套网格，即细空间 $V^{k}{}_{h}{}^{l}$ 和粗空间 $V^{k}{}_{H}{}^{l}$，它们的网格尺寸分别为 h 和 H（$h < H$）。此外，在下文给出的应用中，我们始终假设，对于某个 $0 < \lambda < 1.$，有 $H = O\left(h^{\lambda}\right)$

算法 3.1 （[6]）。令 $u^{l}{}_{h}{}^{0} = 0$；假设已得到 $u^{l}{}_{h}{}^{k} \in V_h^l$（$k \geq 0$），$u^{l}{}_{h}{}^{k+1} \in V_h^l$ 定义如下：

1. 求 $e^{l}{}_{H}{}^{k} \in V_H^l$，使得 $\hat{a}\left(e^{l}{}_{H}{}^{k}, v_H^l\right) = \left(f, v_H^l\right) - \hat{a}\left(u^{l}{}_{h}{}^{k}, v_H^l\right) \quad \forall v_H^l \in V_H^l.$

2. 求 $u^{l}{}_{h}{}^{k+1} \in V_h^l$，使得 $a(u^{l}{}_{h}{}^{k+1}, v_h^l) = (f, v_h^l) - N(u^{l}{}_{h}{}^{k} + e^{l}{}_{H}{}^{k}, v_h^l) \quad \forall v_h^l \in V_h^l$ ，

$where\ N(u, v) := \hat{a}(u, v) - a(u, v), \quad \forall u, v \in H_0^1(\Omega).$

假设 $u^{l}{}_{h}{}^{k+1} \in V_h^l$ 是使用算法 3.1 在 $k \geq 1$ 条件下得到的解，则有（见 [6]中的定理 4.4）

$$\|u - u_h^{l,k+1}\|_1 \lesssim (h^l + H^{k+l})\|u\|_{l+1}, \tag{3.1}$$

这意味着，算法 3.1 给出的双网格解可以有效地逼近式 (2.4) 的有限元解 u_h^l。

注 3.1. 为了在(3.1)中获得最优收敛阶，我们应假设H 和 h 满足关系 $h = O(H^{\frac{l+k}{l}})$。然而，要实现满足$h = O(H^{\frac{l+k}{l}})$的精确嵌套网格非常困难。

—

接下来，我们考虑在算法 3.1 的第二步中用 $V_H^s(s \geq l+1)$ 替换 V_h^l，并得到以下算法。

算法 3.2. 设$\hat{u}^{l}{}_{H}{}^{0} = 0$；假设已得到$\hat{u}^{s}{}_{H}{}^{k} \in V_H^s(k \geq 0)$，则$\hat{u}^{s}{}_{H}{}^{k+1} \in V_H^s$ 定义如下：

1. 求 $e^{l}{}_{H}{}^{k} \in V_H^l$，使得$\hat{a}(e^{l}{}_{H}{}^{k}, v_H^l) = (f, v_H^l) - \hat{a}(\hat{u}^{s}{}_{H}{}^{k}, v_H^l) \quad \forall v_H^l \in V_H^l.$

2. 求 $\hat{u}^{s}{}_{H}{}^{k+1} \in V_H^s$，使得 $a(\hat{u}^{s}{}_{H}{}^{k+1}, v_H^s) = (f, v_H^s) - N(\hat{u}^{s}{}_{H}{}^{k} + e^{l}{}_{H}{}^{k}, v_H^s) \quad \forall v_H^s \in V_H^s.$

注3.2。比较算法 *3.1*和算法 *3.2*，尽管该算法在第二步求解的是*SPD*问题，但 V_H^s 的自由度少于 V_h^l（见表*1*），因此可以大幅缩短计算时间。

$\mathcal{T}_H$	$dof(V_H^3)$	$dof(V_h^3)$	$dof(V_H^4)$	$dof(V_H^5)$	$dof(V_H^6)$
$1/9$	784	59536	1369	2116	3025
$1/10$	961	90601	1681	2601	3721
$1/11$	1156	132496	2025	3136	4489
$1/12$	1369	187489	2401	3721	5329

表1：取 $h = H^2$、 V_H^4、 V_H^5、 V_H^6 和 V_h^3自由度进行比较

4. 数值结果

本节通过数值实验验证迭代两层算法的有效性。我们借助软件包 Fenics [4]，对迭代两层算法进行了所有实验。

例4.1. 我们考虑模型问题 (1.1)-(1.2)，其中计算区域为 $\Omega = (0, 1)^2$,，系数为 $\alpha = 1.0, \boldsymbol{\beta} = (0,0)^t$ 和 $\gamma = -10$，精确解为 $u = \sin(\pi x)\sin(\pi y)$，而 f 可通过将精确解代入方程 (1.1) 得到。

我们首先将区域 Ω 的 $x-$ 轴和 $y-$ 轴等分为 M 个子区间，然后用斜率为 -1 的直线将每个正方形分成两个三角形。因此，我们得到一系列嵌套的结构化网格及其对应的网格 $\mathcal{T}_H$，其中 $H = 1/M$，$M \geq 2$ 为整数，见图1。我们选择基于网格 $\mathcal{T}_H$ 的分片共形 l 阶有限元空间V_H^l。对于算法 3.1，我们取 $h = H^2$。

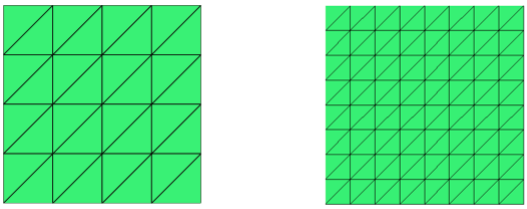


图1：带有 $\mathcal{T}_{1/4}$ （左）和 $\mathcal{T}_{1/8}$ （右）的结构化网格。

从表2可以看出，我们的算法和传统迭代网格都能达到最优收敛阶。并且，为了达到相同的精度，我们的算法所需的 CPU 时间更少。

从表2和表3可以看出，当粗空间多项式次数$l = 3$ 固定时，细空间多项式次数每增加一次，算法 3.2 解的 H^1-范数误差估计的收敛阶就提高一阶。可以看出，算法 3.2 解的 H^1-范数误差取决于 s 的值。

	算法 3.1, $l=3$, $k=3$			算法 3.2, $l=3$, $s=6$, $k=3$		
H	$\|u-u_h^{3,3}\|_1$	$\|u-u_h^{3,3}\|_1 * H^{-6}$	CPU	$\|u-\hat{u}_H^{6,3}\|_1$	$\|u-\hat{u}_H^{6,3}\|_1 * H^{-6}$	CPU
$1/9$	9.8925E-07	5.2573E-01	6.1544	5.7750E-08	3.0691E-02	0.2503
$1/10$	5.2609E-07	5.2609E-01	9.0338	3.0706E-08	3.0706E-02	0.3082
$1/11$	2.9711E-07	5.2635E-01	16.177	1.7339E-08	3.0717E-02	0.3644
$1/12$	1.7634E-07	5.2655E-01	24.047	1.0290E-08	3.0726E-02	0.4347

表2：经典迭代双网格算法 3.1 与迭代两层算法 3.2 之间的 H^1 误差估计比较。

	算法 3.2, $l=3$, $s=4$, $k=3$			算法 3.2, $l=3$, $s=5$, $k=3$		
H	$\|u-\hat{u}_H^{4,3}\|_1$	$\|u-\hat{u}_H^{4,3}\|_1 * H^{-4}$	CPU	$\|u-\hat{u}_H^{5,3}\|_1$	$\|u-\hat{u}_H^{5,3}\|_1 * H^{-5}$	CPU
$1/9$	3.6409E-05	2.3888E-01	0.1383	1.6093E-06	9.5028E-02	0.1825
$1/10$	2.3903E-05	2.3903E-01	0.1633	9.5141E-07	9.5141E-02	0.2172
$1/11$	1.6334E-05	2.3915E-01	0.2785	5.9129E-07	9.5228E-02	0.2571
$1/12$	1.1538E-05	2.3924E-01	0.2237	3.8298E-07	9.5299E-02	0.3020

表 3：迭代两层算法 3.2 的 H^1 误差估计。

例 4.2. 我们考虑模型问题 (1.1)-(1.2)，其中计算区域为 $\Omega=(0,1)^2$,，系数为 $\alpha=1.0$, $\boldsymbol{\beta}=(0,0)^t$ 和 $\gamma=-10$，精确解为 $u=x(1-x)^2y(1-y)^2$,，且 f 可通过将精确解代入方程 (1.1)得到。

	算法 3.1, $l=3$, $k=3$			算法 3.2, $l=3$, $s=6$, $k=3$		
H	$\|u-u_h^{3,3}\|_1$	$\|u-u_h^{3,3}\|_1 * H^{-6}$	CPU	$\|u-\hat{u}_H^{6,3}\|_1$	$\|u-\hat{u}_H^{6,3}\|_1 * H^{-6}$	CPU
$1/9$	6.0567E-08	3.2188E-02	6.0850	2.8919E-13	1.5369E-07	0.1901
$1/10$	3.2255E-08	3.2255E-02	8.9463	1.2153E-13	1.2153E-07	0.2335
$1/11$	1.8235E-08	3.2305E-02	16.0683	7.9992E-14	1.4171E-07	0.2751
$1/12$	1.0832E-08	3.2344E-02	23.8708	7.8801E-14	2.3530E-07	0.3291

表4：比较经典迭代双网格算法 3.1 与迭代两层算法 3.2 的 H^1 误差估计。

从表4和表5可以观察到与表2和表3相同的结论。

致谢

第一、第二和第四作者获得国家自然科学基金（编号：12071160）资助。第二作者还获得国家自然科学基金（编号：11901212）资助。第三作者获得国家自然科学基金（编号：

	算法 3.2, $l=3$, $s=4$, $k=3$			算法 3.2, $l=3$, $s=5$, $k=3$		
H	$\|u-\hat{u}^{4,3}_{H}\|_1$	$\|u-\hat{u}^{4,3}_{H}\|_1 * H^{-4}$	CPU	$\|u-\hat{u}^{5,3}_{H}\|_1$	$\|u-\hat{u}^{5,3}_{H}\|_1 * H^{-5}$	CPU
$1/9$	1.9981E-06	1.3110E-02	0.1035	5.2140E-08	3.0788E-03	0.1380
$1/10$	1.3129E-06	1.3129E-02	0.1214	3.0796E-08	3.0796E-03	0.1637
$1/11$	8.9783E-07	1.3145E-02	0.1420	1.9126E-08	3.0803E-03	0.1938
$1/12$	6.3456E-07	1.3158E-02	0.1661	1.2381E-08	3.0809E-03	0.2273

表 5：迭代两层算法 3.2 的 H^1 误差估计。

12161026）和广西自然科学基金（编号：2020GXNS- FAA159098）资助。

参考文献

参考文献

[1] C.J. Bi、C. Wang、Y.P. Lin，非线性椭圆问题双网格有限元方法的后验误差估计，*J. Sci. Comput.* 74 (1) (2018) 23–48.[2]C. Dawson、M. Wheeler、C. Woodward，非线性抛物方程的双网格有限差分格式，*SIAM J Numer. Anal.*35 (2) (1998) 435–452.[3] C.N. Dawson、M. F. Wheeler，非线性抛物方程混合有限元逼近的双网格方法，*Contemp. Math.*, 180 (1994) 191–203.[4] A. Logg、K.A. Mardal、G. Wells，通过有限元方法自动求解微分方程，*Springer*, 2012.[5] M. Marion、J.C. Xu，基于双网格有限元的新型非线性Galerkin方法的误差估计，*SIAM J. Numer. Anal.*32 (4) (1995) 1170–1184.[6] J.C. Xu，线性和非线性偏微分方程的双网格离散技术，*SIAM J. Numer.Anal.*33 (5) (1996) 1759–1777.[7] Y. Yang、B.Z. Lu、Y. Xie，稳态Poisson–Nernst–Planck方程的解耦双网格方法，*J. Comput. Math.* 37 (4) (2019) 556–578.[8] L.Q. Zhong、C.M. Liu、S. Shu，时谐Maxwell方程边元离散的两层加性预条件子，*Comput. Math. Appl.* 66 (4) (2013) 432–440.[9] L.Q. Zhong、S. Shu、J.X. Wang、J.C. Xu，时谐Maxwell方程的双网格方法，*Numer. Linear Algebra Appl.* 20 (1) (2013) 93–111.[10] L.Q. Zhong、L.L. Zhou、C.M. Liu、J. Peng，非线性椭圆偏微分方程的双网格IPDG离散格式，*Commun. Nonlinear Sci. Numer. Simul.*95 (2021) 105587.