

一种求解半线性椭圆问题的快速 多重网格法^{*1)}

谢和虎

(中国科学院数学与系统科学研究院, 计算数学研究所, 国家数学与交叉科学中心,
科学与工程计算国家重点实验室, 北京 100190;
中国科学院大学, 数学科学学院, 北京 100049)

谢满庭

(天津大学应用数学中心, 天津 300072)

张 宁

(中国科学院数学与系统科学研究院, 计算数学研究所, 国家数学与交叉科学中心,
科学与工程计算国家重点实验室, 北京 100190;
中国科学院大学, 数学科学学院, 北京 100049)

摘 要

本文介绍一种求解半线性问题的完全多重网格算法, 该算法是基于多重校正算法与线性边值问题的多重网格迭代结合而设计的. 多重校正算法将半线性问题的求解转化成线性边值问题的求解加上在一个低维空间上的半线性问题的求解. 利用并行计算技术, 这里所提出的多重网格算法可以明显地提高求解半线性椭圆问题的效率. 更进一步, 当非线性项是多项式函数的时候, 本文也设计了一种高效的完全多重网格算法, 并且通过分析可以知道该算法求解多项式形式的半线性椭圆问题的计算量具有渐近最优的性质. 最后用数值实验验证了本文算法的有效性.

关键词: 半线性椭圆问题; 有限元; 完全多重网格; 多水平校正; 张量计算技术; 渐近计算量绝对最优

MR (2010) 主题分类: 65N30, 65B99

AN EFFICIENT MULTIGRID METHOD FOR SEMILINEAR ELLIPTIC EQUATION

Xie Hehu

(LSEC, NCMIS, Institute of Computational Mathematics, Academy of Mathematics and Systems
Science, Chinese Academy of Sciences, Beijing 100190, China;
School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China)

Xie Manting

(Center for Applied Mathematics, Tianjin University, Tianjin 300072, China)

* 2019 年 3 月 18 日收到.

¹⁾ 基金项目: 国家自然科学基金 (91730302, 11771434, 91330202, 11371026, 11001259, 11031006) 资助.

Zhang Ning

(LSEC, NCMIS, Institute of Computational Mathematics, Academy of Mathematics and Systems
Science, Chinese Academy of Sciences, Beijing 100190, China;

School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract

A full multigrid method is proposed to solve the semilinear elliptic problem by the finite element method based on the combination of multilevel correction method and multigrid method for the linear elliptic problems. In the proposed method, solving the semilinear problem is decomposed into solutions of the linear elliptic problem by the multigrid method, and the semilinear problem which is defined in a very low dimension space. With the help of parallel computing technique, the overfull efficiency can be improved clearly. Furthermore, when the nonlinear term is a polynomial function, an efficient full multigrid method is designed such that the asymptotically computational work is absolutely optimal. One numerical example is provided to validate the efficiency of the proposed method in this paper.

Keywords: semilinear elliptic problem; finite element; full multigrid; multilevel correction; tensor computing technique; asymptotically optimal

2010 Mathematics Subject Classification: 65N30, 65B99

1. 引言

本文主要研究半线性椭圆问题的快速多重网格方法. 众所周知, 多重网格 (multigrid) 方法和多水平 (multilevel) 方法 (见文献 [3–6, 9, 15–17, 22]) 可以给出求解线性边值问题的最优收敛阶. 由这些算法计算得到近似解的误差界与有限元离散的理论误差界可以达到一致. 近十年来, 为了提高非线性椭圆问题的求解效率, 对非线性椭圆问题的多重网格方法进行了许多研究 [6, 17, 23, 24]. 在现有的多重网格方法中, 通常采用牛顿迭代法将非线性方程线性化, 而且需要非线性项的二阶导数有界. 有关更多信息, 请参阅 [10, 17, 23] 及其中所引用的参考文献.

近年来, 在文献 [12, 13, 18, 19, 21] 中提出了一种具有效率最优性质的求解特征值问题的多水平校正方法及其相应的多重网格方法. 这种方法可以将非线性的求解转化成标准线性方程的求解和在一个低维空间中的特征值问题的求解, 从而使得求解特征值问题的计算量与求解相应线性问题的计算量相当. 本文的目的在于提出一种基于多水平校正方法 [18, 19] 的求解半线性椭圆问题的完全多重网格方法 (full multigrid method). 多水平校正方法的主要思想是设计一个特殊的低维空间将半线性问题的求解转化标准线性边值问题的求解和这个特殊的低维空间上半线性问题的求解. 并且在每个校正步中, 对于其中的线性椭圆问题并不需要精确求解, 而只需要进行一定的多重网格迭代步. 利用这个新的多重网格方法, 求解半线性椭圆问题的计算量并不会比求解相应的线性椭圆边值问题的计算大很多. 与现有的半线性问题的多重网格方法相比, 这里的方法仅需要非线性项在一定的范数意义下满足 Lipschitz 连续条件. 最后提出一种求解多项式形式的半线性椭圆问题的高效多重网格方法. 利用所提出的高效方法, 可以使多重网格方法真正达到渐近最优的计算复杂度, 这几乎不依赖于方程的非线性.

本文的概要如下: 在第 2 节, 我们介绍半线性椭圆问题的有限元方法. 求解半线性椭圆问题的完全多重网格方法将在第 3 节中给出. 第 4 节将给出一种求解校正步中非线性问题的快

速迭代算法的实现方式. 第 5 节提供一个数值算例来验证数值方法的有效性. 最后一节给出了一些总结及可能的推广.

2. 有限元离散

在本文中, 字母 C (或者带有上下标) 表示一个与网格尺寸无关的常数, 并且在不同的地方可能表示不同的大小. 为了方便起见, 下文用表达式 $x_1 \lesssim y_1$, $x_2 \gtrsim y_2$ 和 $x_3 \approx y_3$ 来表示 $x_1 \leq C_1 y_1$, $x_2 \geq c_2 y_2$ 和 $c_3 x_3 \leq y_3 \leq C_3 x_3$. 符号 $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$) 表示一个具有 Lipschitz 连续边界 $\partial\Omega$ 的 d 维区域. 这里我们使用标准的 Sobolev 空间符号 $W^{s,p}(\Omega)$ 以及相应的范数 $\|\cdot\|_{s,p,\Omega}$ 和半范数 $|\cdot|_{s,p,\Omega}$ [1]. 当 $p = 2$, 定义 $H^s(\Omega) = W^{s,2}(\Omega)$ 以及 $H_0^1(\Omega) = \{v \in H^1(\Omega) : v|_{\partial\Omega} = 0\}$, 其中 $v|_{\partial\Omega} = 0$ 是在函数迹的意义下理解的. 简单起见, 本文中使用 $\|\cdot\|_{s,\Omega}$ 来表示 $\|\cdot\|_{s,2,\Omega}$, V 来表示 $H_0^1(\Omega)$.

本文考虑如下形式的半线性椭圆方程: 求 u 使得

$$\begin{cases} -\nabla \cdot (\mathcal{A} \nabla u) + f(x, u) = g, & \text{在 } \Omega \text{ 内,} \\ u = 0, & \text{在 } \partial\Omega \text{ 上,} \end{cases} \quad (2.1)$$

其中 $\mathcal{A} = (a_{i,j})_{d \times d}$ 是一个对称正定矩阵, 元素 $a_{i,j} \in W^{1,\infty}(\Omega)$ ($i, j = 1, 2, \dots, d$), $f(x, u)$ 是关于第二个参数的非线性函数.

半线性椭圆问题 (6.1) 的变分形式可以描述为: 求 $u \in V$, 使得

$$a(u, v) + (f(x, u), v) = (g, v), \quad \forall v \in V, \quad (2.2)$$

其中

$$a(u, v) = (\mathcal{A} \nabla u, \nabla v). \quad (2.3)$$

易知 $a(u, v)$ 在空间 V 上满足如下的有界性和强椭圆性, 即

$$a(u, v) \leq C_a \|u\|_{1,\Omega} \|v\|_{1,\Omega} \quad \text{以及} \quad c_a \|u\|_{1,\Omega}^2 \leq a(u, u), \quad \forall u, v \in V. \quad (2.4)$$

由上面的强制性可以定义范数 $\|w\|_{a,\Omega} = \sqrt{a(w, w)}$ 且满足 $\|w\|_{a,\Omega} \approx \|w\|_{1,\Omega}$. 如此我们在本文中用 $\|\cdot\|_a := \sqrt{a(\cdot, \cdot)}$ 来代替标准的 $H^1(\Omega)$ 上的范数 $\|\cdot\|_1$. 同时为了后面的理论分析, 由 Poincaré 不等式可以得到如下的不等式

$$\|w\|_0 \leq C_p \|w\|_a, \quad \forall w \in V, \quad (2.5)$$

其中常数 C_p 由 Poincaré 不等式确定.

为了保证问题 (2.2) 存在唯一性解, 我们假设非线性项 $f(\cdot, \cdot)$ 满足如下假设:

假设 A: 非线性函数 $f(x, \cdot)$ 满足如下的凸性和 Lipschitz 连续条件

$$\begin{cases} (f(x, w) - f(x, v), w - v) \geq 0, & \forall w \in V, \forall v \in V, \\ (f(x, w) - f(x, v), \phi) \leq C_f \|w - v\|_0 \|\phi\|_1, & \forall w \in V, \forall v \in V, \forall \phi \in V. \end{cases} \quad (2.6)$$

现在, 我们介绍半线性椭圆问题 (2.2) 的有限元方法. 首先在计算区域 $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$) 上产生一个正则剖分, 即把二维区域分解成正则的三角形或四边形 (三维区域分解成四面体

或六面体)^[7,8]. 用 h_K 表示单元 $K \in \mathcal{T}_h$ 的直径, h 表示网格 $\mathcal{T}_h$ 中所有单元的最大直径. 在所定义的网格 $\mathcal{T}_h$ 上构造相应的线性有限元空间 $V_h \subset V$. 简便起见, 我们用 V_h 表示线性有限元空间:

$$V_h = \{v_h \in C(\Omega) \mid v_h|_K \in \mathcal{P}_1, \forall K \in \mathcal{T}_h\} \cap H_0^1(\Omega), \quad (2.7)$$

这里 $\mathcal{P}_1$ 表示一次多项式空间.

求解半线性椭圆问题 (2.2) 的有限元方法为: 求 $\bar{u}_h \in V_h$, 使得

$$a(\bar{u}_h, v_h) + (f(x, \bar{u}_h), v_h) = (g, v_h), \quad \forall v_h \in V_h. \quad (2.8)$$

这里我们用 $\bar{u}_h$ 表示在有限元空间 V_h 上精确的有限元逼近.

为了后面的理论分析, 定义线性化算子 $L: H_0^1(\Omega) \rightarrow H^{-1}(\Omega)$:

$$(Lw, v) = (\mathcal{A}\nabla w, \nabla v), \quad \forall w \in V, \forall v \in V.$$

为了导出先验误差估计, 我们定义 $\eta_a(V_h)$ 如下:

$$\eta_a(V_h) = \sup_{f \in L^2(\Omega), \|f\|_0=1} \inf_{v_h \in V_h} \|L^{-1}f - v_h\|_a.$$

易知当 $h \rightarrow 0$ 时, $\eta_a(V_h) \rightarrow 0$ (参见文献 [7, 8]).

为了给出有限元方法的离散误差, 我们定义

$$\delta_h(u) = \inf_{v_h \in V_h} \|u - v_h\|_a.$$

从文献 [17] 中, 可以得到如下误差估计.

引理 1. 当假设 A 成立时, 问题 (2.2) 和 (2.8) 都存在唯一解, 且如下误差估计成立

$$\|u - \bar{u}_h\|_a \leq (1 + C_f \eta_a(V_h)) \delta_h(u), \quad (2.9)$$

$$\|u - \bar{u}_h\|_0 \leq (1 + C_f C_p) \eta_a(V_h) \|u - \bar{u}_h\|_a. \quad (2.10)$$

证明. 由文献 [17] 中定理 6.1 可知, 问题 (2.2) 和 (2.8) 都存在唯一解. 现在我们证明误差估计. 由下式定义有限元投影算子 P_h

$$a(P_h w, v_h) = a(w, v_h), \quad \forall w \in V, \forall v_h \in V_h.$$

易知 $\|u - P_h u\|_a = \delta_h(u)$ 和 $\|u - P_h u\|_0 \leq \eta_a(V_h) \|u - P_h u\|_a$ 成立. 定义 $w_h = P_h u - \bar{u}_h$. 由 (2.2), (2.6) 和 (2.8), 可以得到

$$\begin{aligned} a(P_h u - \bar{u}_h, w_h) &\leq a(P_h u - \bar{u}_h, w_h) + (f(x, P_h u) - f(x, \bar{u}_h), w_h) \\ &= a(P_h u, w_h) + (f(x, P_h u), w_h) - (g, w_h) \\ &= a(P_h u - u, w_h) + (f(x, P_h u) - f(x, u), w_h) \\ &= (f(x, P_h u) - f(x, u), w_h) \leq C_f \|u - P_h u\|_0 \|w_h\|_a. \end{aligned}$$

在上式中取 $w_h = P_h u - \bar{u}_h$ 可以得到如下不等式成立

$$\|P_h u - \bar{u}_h\|_a \leq C_f \|u - P_h u\|_0 \leq C_f \eta_a(V_h) \|u - P_h u\|_a. \quad (2.11)$$

结合 (2.11) 和三角不等式, 有

$$\begin{aligned}\|u - \bar{u}_h\|_a &\leq \|u - P_h u\|_a + \|P_h u - \bar{u}_h\|_a \leq \delta_h(u) + C_f \eta_a(V_h) \|u - P_h u\|_a \\ &\leq (1 + C_f \eta_a(V_h)) \delta_h(u).\end{aligned}$$

此即所要证明的结论 (2.9). 由 (2.11) 和三角不等式可得

$$\begin{aligned}\|u - \bar{u}_h\|_0 &\leq \|u - P_h u\|_0 + \|P_h u - \bar{u}_h\|_0 \leq \|u - P_h u\|_0 + C_p \|P_h u - \bar{u}_h\|_a \\ &\leq \eta_a(V_h) \|u - P_h u\|_a + C_f C_p \eta_a(V_h) \|u - P_h u\|_a \\ &\leq (1 + C_f C_p) \eta_a(V_h) \|u - P_h u\|_a \leq (1 + C_f C_p) \eta_a(V_h) \|u - \bar{u}_h\|_a.\end{aligned}$$

这就是需要证明的第二个结论 (2.10).

3. 半线性椭圆问题的完全多重网格法

本节介绍一种基于多水平校正方法^[18, 19]的求解半线性问题的完全多重网格方法. 这种方法的主要思想是把半线性椭圆问题的求解转化为有限元空间序列上的边值问题的求解以及在一个低维空间上半线性问题的求解. 同时对于边值问题, 我们不需要精确求解, 只需要利用多重网格迭代得到一个近似解.

为了实现这种完全多重网格法, 首先需要构造一系列嵌套有限元空间. 给定一个网格尺寸为 H 的粗网格 $\mathcal{T}_H$, 并在这个网格上定义相应的线性有限元空间 V_H . 然后利用如下规则建立嵌套网格序列: 对 $\mathcal{T}_H$ 进行几次一致加密得到一个初始网格 $\mathcal{T}_{h_1}$, 然后对 $\mathcal{T}_{h_{k-1}}$ 进行一次加密得到 $\mathcal{T}_{h_k}$ (每个单元产生 β^d 个新单元), 网格尺寸满足

$$h_k = \frac{1}{\beta} h_{k-1}, \quad k = 2, \dots, n, \quad (3.1)$$

其中 $\beta > 1$ 表示加密指数, 通常情况下 $\beta = 2$.

根据上述网格序列, 就可以构造相应的有限元空间:

$$V_H \subseteq V_{h_1} \subset V_{h_2} \subset \dots \subset V_{h_n}. \quad (3.2)$$

有限元空间 V_{h_k} ($k = 1, 2, \dots, n$) 以及 V_H 有如下的关系^[7, 8]:

$$\eta_a(V_{h_k}) \approx \frac{1}{\beta} \eta_a(V_{h_{k-1}}), \quad \delta_{h_k}(u) \approx \frac{1}{\beta} \delta_{h_{k-1}}(u), \quad k = 2, \dots, n. \quad (3.3)$$

3.1. 单步校正算法

为了设计完全多重网格算法, 需要先设计一个单步校正算法. 假设我们已经得到了近似解 $u_{h_k}^{(\ell)} \in V_{h_k}$. 现在介绍单步校正算法来提高给定的近似解 $u_{h_k}^{(\ell)}$ 的精度.

算法 1. 单步校正算法

1. 定义如下的线性边值问题: 求 $\hat{u}_{h_k}^{(\ell+1)} \in V_{h_k}$, 使得

$$a(\hat{u}_{h_k}^{(\ell+1)}, v_{h_k}) = -(f(x, u_{h_k}^{(\ell)}), v_{h_k}) + (g, v_{h_k}), \quad \forall v_{h_k} \in V_{h_k}. \quad (3.4)$$

以 $u_{h_k}^{(\ell)}$ 为初值, 执行 m 步多重网格迭代得到一个近似特征函数 $\tilde{u}_{h_k}^{(\ell+1)} \in V_{h_k}$, 相应的误差满足如下下降比率

$$\|\tilde{u}_{h_k}^{(\ell+1)} - \hat{u}_{h_k}^{(\ell+1)}\|_a \leq \theta \|u_{h_k}^{(\ell)} - \hat{u}_{h_k}^{(\ell+1)}\|_a, \quad (3.5)$$

其中 $\theta < 1$ 是与 h_k 独立的常数 (多重网格方法的一致收敛性质). 我们把上述过程表示为:

$$\tilde{u}_{h_k}^{(\ell+1)} = \text{MG}(V_{h_k}, -f(x, u_{h_k}^{(\ell)}) + g, u_{h_k}^{(\ell)}, m), \quad (3.6)$$

其中 V_{h_k} 表示执行多重网格迭代的空间, $-f(x, u_{h_k}^{(\ell)}) + g$ 是线性方程的右端项, $u_{h_k}^{(\ell)}$ 表示迭代初值, m 表示多重网格迭代次数.

2. 定义一个新的有限元空间 $V_{H, h_k} = V_H \oplus \text{span}\{\tilde{u}_{h_k}^{(\ell+1)}\}$, 求解如下的半线性椭圆问题: 求 $u_{h_k}^{(\ell+1)} \in V_{H, h_k}$, 使得

$$a(u_{h_k}^{(\ell+1)}, v_{H, h_k}) + (f(x, u_{h_k}^{(\ell+1)}), v_{H, h_k}) = (g, v_{H, h_k}), \quad \forall v_{H, h_k} \in V_{H, h_k}. \quad (3.7)$$

把上述问题的解 $u_{h_{k+1}}^{(\ell+1)}$ 作为输出.

为了简化符号, 把上面两步总结为:

$$u_{h_k}^{(\ell+1)} = \text{SemilinearMG}(V_H, u_{h_k}^{(\ell)}, V_{h_k}).$$

依据半线性问题的有限元方法离散误差估计可以给出算法 1 的误差估计如下.

定理 1. 假设存在常数 C_1 使得给定解 $u_{h_k}^{(\ell)}$ 有如下误差估计

$$\|\bar{u}_{h_k} - u_{h_k}^{(\ell)}\|_0 \leq C_1 \eta_a(V_H) \|\bar{u}_{h_k} - u_{h_k}^{(\ell)}\|_a. \quad (3.8)$$

则执行单步校正算法 1 之后, 所得近似解 $u_{h_k}^{(\ell+1)}$ 有如下误差估计

$$\|\bar{u}_{h_k} - u_{h_k}^{(\ell+1)}\|_a \leq \gamma \|\bar{u}_{h_k} - u_{h_k}^{(\ell)}\|_a, \quad (3.9)$$

$$\|\bar{u}_{h_k} - u_{h_k}^{(\ell+1)}\|_0 \leq (1 + C_p C_f) \eta_a(V_H) \|\bar{u}_{h_k} - u_{h_k}^{(\ell+1)}\|_a, \quad (3.10)$$

其中 $\gamma := (\theta + (1 + \theta)C_1 C_f \eta_a(V_H))(1 + C_f \eta_a(V_H))$.

证明. 由 (2.6), (2.8) 和 (3.4), 有

$$\begin{aligned} a(\bar{u}_{h_k} - \hat{u}_{h_k}^{(\ell+1)}, v_{h_k}) &= (f(x, u_{h_k}^{(\ell)}) - f(x, \bar{u}_{h_k}), v_{h_k}) \leq C_f \|\bar{u}_{h_k} - u_{h_k}^{(\ell)}\|_0 \|v_{h_k}\|_a \\ &\leq C_1 C_f \eta_a(V_H) \|\bar{u}_{h_k} - u_{h_k}^{(\ell)}\|_a \|v_{h_k}\|_a, \quad \forall v_{h_k} \in V_{h_k}. \end{aligned} \quad (3.11)$$

结合 (2.4) 和 (3.11), 有

$$\|\bar{u}_{h_k} - \hat{u}_{h_k}^{(\ell+1)}\|_a \leq C_1 C_f \eta_a(V_H) \|\bar{u}_{h_k} - u_{h_k}^{(\ell)}\|_a. \quad (3.12)$$

对方程 (3.4) 执行 m 步多重网格迭代得到解 $\tilde{u}_{h_k}^{(\ell+1)}$. 由 (3.5) 和 (3.12), 如下估计成立

$$\|\tilde{u}_{h_k}^{(\ell+1)} - \bar{u}_{h_k}\|_a \leq \|\tilde{u}_{h_k}^{(\ell+1)} - \hat{u}_{h_k}^{(\ell+1)}\|_a + \|\hat{u}_{h_k}^{(\ell+1)} - \bar{u}_{h_k}\|_a$$

$$\begin{aligned}
&\leq \theta \|u_{h_k}^{(\ell)} - \hat{u}_{h_k}^{(\ell+1)}\|_a + \|\hat{u}_{h_k}^{(\ell+1)} - \bar{u}_{h_k}\|_a \\
&\leq \theta \|u_{h_k}^{(\ell)} - \bar{u}_{h_k}\|_a + \theta \|\hat{u}_{h_k}^{(\ell+1)} - \bar{u}_{h_k}\|_a + \|\hat{u}_{h_k}^{(\ell+1)} - \bar{u}_{h_k}\|_a \\
&\leq (\theta + (1 + \theta)C_1 C_f \eta_a(V_H)) \|\bar{u}_{h_k} - u_{h_k}^{(\ell)}\|_a.
\end{aligned} \tag{3.13}$$

注意到半线性椭圆问题 (3.7) 可以看做半线性椭圆问题 (2.8) 的有限维逼近. 令 $P_{H,h_k} : V \rightarrow V_{H,h_k}$ 表示如下有限维投影算子

$$a(P_{H,h_k} w, v_{H,h_k}) = a(w, v_{H,h_k}), \quad \forall w \in V, \forall v_{H,h_k} \in V_{H,h_k}.$$

由于 $\hat{u}_{h_k}^{(\ell+1)} \in V_{H,h_k}$ 和 $V_H \subset V_{H,h_k}$, 容易知道 $\eta_a(V_{H,h_k}) \leq \eta_a(V_H)$ 和如下不等式成立

$$\|\bar{u}_{h_k} - P_{H,h_k} \bar{u}_{h_k}\|_a = \inf_{v_{H,h_k} \in V_{H,h_k}} \|\bar{u}_{h_k} - v_{H,h_k}\|_a \leq \|\bar{u}_{h_k} - \hat{u}_{h_k}^{(\ell+1)}\|_a, \tag{3.14}$$

$$\|\bar{u}_{h_k} - P_{H,h_k} \bar{u}_{h_k}\|_0 \leq \eta_a(V_{H,h_k}) \|\bar{u}_{h_k} - P_{H,h_k} \bar{u}_{h_k}\|_a \leq \eta_a(V_H) \|\bar{u}_{h_k} - P_{H,h_k} \bar{u}_{h_k}\|_a. \tag{3.15}$$

我们在证明中定义 $w_{h_k} = P_{H,h_k} \bar{u}_{h_k} - u_{h_k}^{(\ell+1)} \in V_{H,h_k}$. 由问题 (2.8) 和 (3.7), 则如下估计成立

$$\begin{aligned}
&a(P_{H,h_k} \bar{u}_{h_k} - u_{h_k}^{(\ell+1)}, w_{h_k}) \\
&\leq a(P_{H,h_k} \bar{u}_{h_k} - u_{h_k}^{(\ell+1)}, w_{h_k}) + (f(x, P_{H,h_k} \bar{u}_{h_k}) - f(x, u_{h_k}^{(\ell+1)}), w_{h_k}) \\
&= a(P_{H,h_k} \bar{u}_{h_k}, w_{h_k}) + (f(x, P_{H,h_k} \bar{u}_{h_k}), w_{h_k}) - (g, w_{h_k}) \\
&= a(P_{H,h_k} \bar{u}_{h_k} - \bar{u}_{h_k}, w_{h_k}) + (f(x, P_{H,h_k} \bar{u}_{h_k}) - f(x, \bar{u}_{h_k}), w_{h_k}) \\
&= (f(x, P_{H,h_k} \bar{u}_{h_k}) - f(x, \bar{u}_{h_k}), w_{h_k}) \leq C_f \|\bar{u}_{h_k} - P_{H,h_k} \bar{u}_{h_k}\|_0 \|w_{h_k}\|_a.
\end{aligned} \tag{3.16}$$

由 (3.15) 和 (3.16), 有

$$\|P_{H,h_k} \bar{u}_{h_k} - u_{h_k}^{(\ell+1)}\|_a \leq C_f \|\bar{u}_{h_k} - P_{H,h_k} \bar{u}_{h_k}\|_0 \leq C_f \eta_a(V_H) \|\bar{u}_{h_k} - P_{H,h_k} \bar{u}_{h_k}\|_a. \tag{3.17}$$

结合 (3.14), (3.17) 和三角不等式, 可得

$$\begin{aligned}
\|\bar{u}_{h_k} - u_{h_k}^{(\ell+1)}\|_a &\leq \|\bar{u}_{h_k} - P_{H,h_k} \bar{u}_{h_k}\|_a + \|P_{H,h_k} \bar{u}_{h_k} - u_{h_k}^{(\ell+1)}\|_a \\
&\leq (1 + C_f \eta_a(V_H)) \|\bar{u}_{h_k} - P_{H,h_k} \bar{u}_{h_k}\|_a \\
&\leq (1 + C_f \eta_a(V_H)) \|\bar{u}_{h_k} - \hat{u}_{h_k}^{(\ell+1)}\|_a.
\end{aligned} \tag{3.18}$$

结合 (3.13) 和 (3.18) 即可证得 (3.9). 由 (3.15), (3.17), 有限元投影的最优性, Poincaré 不等式和三角不等式, 我们有如下估计

$$\begin{aligned}
\|\bar{u}_{h_k} - u_{h_k}^{(\ell+1)}\|_0 &\leq \|\bar{u}_{h_k} - P_{H,h_k} \bar{u}_{h_k}\|_0 + \|P_{H,h_k} \bar{u}_{h_k} - u_{h_k}^{(\ell+1)}\|_0 \\
&\leq \|\bar{u}_{h_k} - P_{H,h_k} \bar{u}_{h_k}\|_0 + C_p \|P_{H,h_k} \bar{u}_{h_k} - u_{h_k}^{(\ell+1)}\|_a \\
&\leq (1 + C_p C_f) \eta_a(V_H) \|\bar{u}_{h_k} - P_{H,h_k} \bar{u}_{h_k}\|_a \\
&\leq (1 + C_p C_f) \eta_a(V_H) \|\bar{u}_{h_k} - u_{h_k}^{(\ell+1)}\|_a,
\end{aligned} \tag{3.19}$$

这就是所要证明的结论 (3.10), 证毕.

注 1. 条件 (3.8) 中的常数 C_1 将会在下面介绍完全多重网格算法及其分析的时候给出, 定理 1 给出的是进行单步误差估计的框架.

定理 1 的证明显示了低维空间 V_{H,h_k} 的结构在算法 1 所具有的重要作用. 这个特殊的空间使有限元投影 P_{H,h_k} 既有如 (3.14) 式中的精度又有 (3.15) 式中 L^2 -范数的对偶估计 (Aubin-Nitsch 技巧). 这也是多水平校正方法的关键之处.

3.2. 完全多重网格法

在本小节中, 基于算法 1 所定义的单步校正法, 我们构造一个求解半线性椭圆问题的完全多重网格方法. 该算法只需最优的计算复杂度就可以达到最优收敛速度.

算法 2. 完全多重网格法

1. 在 V_{h_1} 中求解如下半线性问题: 求 $u_{h_1} \in V_{h_1}$, 使得

$$a(u_{h_1}, v_{h_1}) + (f(x, u_{h_1}), v_{h_1}) = (g, v_{h_1}), \quad \forall v_{h_1} \in V_{h_1}.$$

2. 对 $k = 2, \dots, n$, 进行如下迭代:

- (a) 令 $u_{h_k}^{(0)} = u_{h_{k-1}}$.

- (b) 对 $\ell = 0, \dots, p-1$, 进行如下迭代

$$u_{h_k}^{(\ell+1)} = \text{SemilinearMG}(V_H, u_{h_k}^{(\ell)}, V_{h_k}).$$

- (c) 定义 $u_{h_k} = u_{h_k}^{(p)}$.

最后得到一个在最细层空间内的近似解 $u_{h_n} \in V_{h_n}$.

定理 2. 在粗网格尺寸 H 足够小使得条件 $\gamma^p \beta < 1$ 成立时, 执行由算法 2 定义的完全多重网格算法 2 所得到的近似解 u_{h_n} 有如下误差估计:

$$\|\bar{u}_{h_n} - u_{h_n}\|_a \leq \frac{2\gamma^p \beta}{1 - \gamma^p \beta} \delta_{h_n}(u), \quad (3.20)$$

$$\|\bar{u}_{h_n} - u_{h_n}\|_0 \leq (1 + C_p C_f) \eta_a(V_H) \|\bar{u}_{h_n} - u_{h_n}\|_a, \quad (3.21)$$

其中 $\gamma = (\theta + (1 + \theta)(1 + C_p C_f) C_f \eta_a(V_H))(1 + C_f \eta_a(V_H))$.

证明. 根据算法 2 的第 1 步, 我们可得 $u_{h_1} = \bar{u}_{h_1}$. 那么由引理 1 和定理 1 的证明, 可知 $u_{h_2}^{(0)}$ 满足如下的不等式

$$\begin{aligned} \|\bar{u}_{h_2} - u_{h_2}^{(0)}\|_0 &= \|\bar{u}_{h_2} - \bar{u}_{h_1}\|_0 \leq (1 + C_p C_f) \eta_a(V_{h_1}) \|\bar{u}_{h_2} - \bar{u}_{h_1}\|_a \\ &= (1 + C_p C_f) \eta_a(V_{h_1}) \|\bar{u}_{h_2} - u_{h_2}^{(0)}\|_a \leq (1 + C_p C_f) \eta_a(V_H) \|\bar{u}_{h_2} - u_{h_2}^{(0)}\|_a. \end{aligned} \quad (3.22)$$

由此可以知道误差估计 $\|\bar{u}_{h_2} - u_{h_2}^{(0)}\|_0$ 满足条件 (3.8) 且此时 $C_1 = (1 + C_p C_f)$. 则由定理 1 中的结论 (3.10) 及递归论证可以知道条件 (3.8) 对任意有限元空间 V_{h_k} 和任意迭代指标 ℓ 都成立, 且 $C_1 = (1 + C_p C_f)$ 和 $\gamma = (\theta + (1 + \theta)(1 + C_p C_f) C_f \eta_a(V_H))(1 + C_f \eta_a(V_H))$. 当 $\theta < 1$ 且 H 充分小的时候易知 $\gamma < 1$.

由算法 2 的定义, 定理 1 和递归论证可以得到最后的近似解 u_{h_n} 有以下误差估计

$$\|\bar{u}_{h_n} - u_{h_n}\|_a \leq \gamma^p \|\bar{u}_{h_n} - u_{h_n}^{(0)}\|_a = \gamma^p \|\bar{u}_{h_n} - u_{h_{n-1}}\|_a$$

$$\begin{aligned}
&\leq \gamma^p (\|\bar{u}_{h_n} - \bar{u}_{h_{n-1}}\|_a + \|\bar{u}_{h_{n-1}} - u_{h_{n-1}}\|_a) \\
&\leq \gamma^p \|\bar{u}_{h_n} - \bar{u}_{h_{n-1}}\|_a + \gamma^{2p} (\|\bar{u}_{h_{n-1}} - \bar{u}_{h_{n-2}}\|_a + \|\bar{u}_{h_{n-2}} - u_{h_{n-2}}\|_a) \\
&\leq \sum_{k=1}^{n-1} \gamma^{kp} \|\bar{u}_{h_{n-k+1}} - \bar{u}_{h_{n-k}}\|_a \\
&\leq \sum_{k=1}^{n-1} \gamma^{kp} (\|\bar{u}_{h_{n-k+1}} - u\|_a + \|u - \bar{u}_{h_{n-k}}\|_a) \\
&\leq 2 \sum_{k=1}^{n-1} \gamma^{kp} \delta_{h_{n-k}}(u) \leq 2 \sum_{k=1}^{n-1} \gamma^{kp} \beta^k \delta_{h_n}(u) \leq \frac{2\gamma^p \beta}{1 - \gamma^p \beta} \delta_{h_n}(u),
\end{aligned}$$

这就是要证明的结论 (3.20). 类似于定理 1 的证明同样可以得到 (3.21), 证毕.

注 2. 执行由算法 2 定义的完全多重网格算法所得到的近似解 u_{h_n} 有误差估计 (3.20). 从误差估计 (3.20) 可以知道我们可以增大 p (增加单步校正的次数) 来提高逼近解的代数精度 $\|\bar{u}_{h_n} - u_{h_n}\|_a$, 即这里的算法可以使得代数误差达到机器精度. 这与多重网格算法求解线性边值问题的性质一样.

推论 1. 由算法 2 得到的最后近似解 u_{h_n} , 有如下估计

$$\|u - u_{h_n}\|_a \leq \left(1 + C_f \eta_a(V_{h_n}) + \frac{2\gamma^p \beta}{1 - \gamma^p \beta}\right) \delta_{h_n}(u), \quad (3.23)$$

$$\|u - u_{h_n}\|_0 \leq \left((1 + C_f C_p) \eta_a(V_{h_n}) + (1 + C_p C_f) \frac{2\gamma^p \beta}{1 - \gamma^p \beta} \eta_a(V_H)\right) \delta_{h_n}(u). \quad (3.24)$$

证明. 这是引理 1 和定理 2 的直接结论.

3.3. 计算量的估计

在这一小节, 我们来分析完全多重网格算法 2 的计算量. 首先, 定义有限元空间 V_{h_k} 的维数为 $N_k := \dim V_{h_k}$, 那么有如下的关系

$$N_k \approx \left(\frac{1}{\beta}\right)^{d(n-k)} N_n, \quad k = 1, 2, \dots, n. \quad (3.25)$$

算法 1 的第二步需要求解半线性椭圆问题 (3.7), 通常采用一些非线性迭代方法 (如: 不动点迭代法或 Newton 类型迭代法) 来求解该低维半线性椭圆问题. 与多重校正算法求解特征值问题 [13, 18, 19] 不一样的是, 在每次非线性迭代步中都需要进行矩阵组装. 假设在每个非线性迭代步骤中, 有限元空间 V_{H, h_k} ($k = 2, \dots, n$) 上组装矩阵需要的工作量为 $\mathcal{O}(N_k)$. 幸运的是由于矩阵组装几乎不需要进行数据交换, 所以这一部分可以很容易地进行并行执行.

定理 3. 假设在算法 2 的求解过程中使用了 ϑ 个计算进程来组装矩阵, 在低维空间 V_{H, h_k} ($k = 2, \dots, n$) 以及 V_{h_1} 上求解半线性椭圆问题需要的工作量分别为 $\mathcal{O}(M_H)$ 和 $\mathcal{O}(M_{h_1})$, 在每一层有限元空间 V_{h_k} 上的多重网格迭代 $\text{MG}(V_{h_k}, -f(x, u_{h_k}^{(\ell)}) + g, u_{h_k}^{(\ell)}, m)$ 需要的计算量为 $\mathcal{O}(N_k)$, 其中 $k = 2, 3, \dots, n$. 用 ϖ 表示求解半线性椭圆问题 (3.7) 的非线性迭代次数. 则算法 2 在每个计算进程上需要的计算量为

$$\text{Total work} = \mathcal{O}\left(\left(1 + \frac{\varpi}{\vartheta}\right) N_n + M_H \log N_n + M_{h_1}\right). \quad (3.26)$$

证明. 令 W_k 表示在第 k 层有限元空间 V_{h_k} 上每个进程的计算量. 则由算法 2 和单步校正算法 1 的定义可得

$$W_k = \mathcal{O} \left(N_k + M_H + \varpi \frac{N_k}{\vartheta} \right). \quad (3.27)$$

递归地使用 (3.27), 同时利用 (3.25), 可以推出

$$\begin{aligned} \text{Total work} &= \sum_{k=1}^n W_k = \mathcal{O} \left(M_{h_1} + \sum_{k=2}^n \left(N_k + M_H + \varpi \frac{N_k}{\vartheta} \right) \right) \\ &= \mathcal{O} \left(\sum_{k=2}^n \left(1 + \frac{\varpi}{\vartheta} \right) N_k + (n-1)M_H + M_{h_1} \right) \\ &= \mathcal{O} \left(\sum_{k=2}^n \left(\frac{1}{\beta} \right)^{d(n-k)} \left(1 + \frac{\varpi}{\vartheta} \right) N_n + M_H \log N_n + M_{h_1} \right) \\ &= \mathcal{O} \left(\left(1 + \frac{\varpi}{\vartheta} \right) N_n + M_H \log N_n + M_{h_1} \right). \end{aligned} \quad (3.28)$$

这就是需要证明的结论, 证毕.

注 3. 由于在算法 1 的第 2 步中我们有一个精度足够好的初始解 $\tilde{u}_{h_k}^{(\ell+1)}$, 那么求解半线性椭圆问题 (3.7) 往往不需要很多次的非线性迭代. 在这种情况下, 如果再有条件 $M_H \ll N_n$ 和 $M_{h_1} \leq N_n$, 可得在每个计算进程上的计算量最多为 $\mathcal{O}(N_n)$. 事实上, 对于更困难的非线性问题, 采用充分多的并行计算进程时, 在每个计算进程的计算量往往也不会超过 $\mathcal{O}(N_n)$.

4. 多项式非线性问题的快速算法

为了达到最终的计算量与非线性强弱无关这一最终目标, 当非线性项 $f(x, u)$ 是一个关于 u 的多项式函数的时候, 这里我们介绍一种可以进行快速执行单步校正算法的方法. 为了更清晰地介绍算法构造的思想, 我们假设非线性函数是三次多项式的情形, 即 $f(x, u) = wu + \zeta u^3$, 其中 $w \geq 0$ 和 $\zeta > 0$ 为标量函数. 这意味着本节要求解如下的非线性方程: 求 u 满足如下的方程

$$\begin{cases} -\Delta u + wu + \zeta u^3 = g, & \text{在 } \Omega \text{ 内,} \\ u = 0, & \text{在 } \partial\Omega \text{ 上.} \end{cases} \quad (4.1)$$

这种非线性问题经常出现在求解相場方程中, 所以这里介绍的算法也可以应用于相場方程的求解.

求解算法 1 中第二步的半线性方程 (3.7) 即为求解如下的非线性方程: 求 $u_{h_k}^{(\ell+1)} \in V_{H,h_k}$ 使得

$$(\nabla u_{h_k}^{(\ell+1)}, \nabla v_{H,h_k}) + (wu_{h_k}^{(\ell+1)}, v_{H,h_k}) + (\zeta(u_{h_k}^{(\ell+1)})^3, v_{H,h_k}) = (g, v_{H,h_k}), \quad \forall v_{H,h_k} \in V_{H,h_k}. \quad (4.2)$$

为了简单起见, 本节我们用 h 表示 h_k , 如此就可以用 $V_h, \tilde{u}_h$ 和 $V_{H,h} = V_H + \text{span}\{\tilde{u}_h\}$ 表示 $V_{h_k}, \tilde{u}_{h_k}^{(\ell+1)}$ 和 $V_{H,h_k} = V_H + \text{span}\{\tilde{u}_{h_k}^{(\ell+1)}\}$. 定义 $N_H := \dim V_H$ 和 $N_h := \dim V_h$, 用 $\{\phi_{k,H}\}_{1 \leq k \leq N_H}$ 表示有限元空间 V_H 的 Lagrange 插值基函数的集合.

这里考虑用不动点迭代 (必要的时候采用阻尼技术) 方法来求解非线性方程 (4.2). 同样为了书写方便, 我们把非线性方程 (4.2) 简写成如下的形式: 求 $u_h \in V_{H,h}$ 使得

$$(\nabla u_h, \nabla v_{H,h}) + (w u_h, v_{H,h}) + (\zeta(u_h)^3, v_{H,h}) = (g, v_{H,h}), \quad \forall v_{H,h} \in V_{H,h}. \quad (4.3)$$

不动点迭代方法求解方程 (4.3) 就是每次迭代的时候求解如下的线性方程: 求 u_h^{j+1} 使得

$$(\nabla u_h^{j+1}, \nabla v_{H,h}) + (w u_h^{j+1}, v_{H,h}) + (\zeta(u_h^j)^2 u_h^{j+1}, v_{H,h}) = (g, v_{H,h}), \quad \forall v_{H,h} \in V_{H,h}, \quad (4.4)$$

其中上标 j 表示非线性迭代次数的指标, $u_h^j \in V_{H,h}$ 是已知的初值.

假设已有的初值 u_h^j 具有展开式: $u_h^j = u_H^j + \alpha_j \tilde{u}_h$, 需要求解的 u_h^{j+1} 具有展开式 $u_h^{j+1} = u_H^{j+1} + \alpha_{j+1} \tilde{u}_h$. 求解方程 (4.4) 就是得到一个新的函数 $u_H^{j+1} \in V_H$ 和一个新的数值 $\alpha_{j+1} \in \mathbb{R}$. 令 $u_H^{j+1} = \sum_{k=1}^{N_H} u_k^{j+1} \phi_{k,H}$ 且定义向量 $\mathbf{u}_H^{j+1}$ 为 $\mathbf{u}_H^{j+1} = [u_1^{j+1}, \dots, u_{N_H}^{j+1}]^T$. 根据空间 $V_{H,h}$ 的结构可以知道线性问题 (4.4) 具有如下的矩阵结构

$$\begin{pmatrix} A_H & b_{Hh} \\ b_{Hh}^T & \xi \end{pmatrix} \begin{pmatrix} \mathbf{u}_H^{j+1} \\ \alpha_{j+1} \end{pmatrix} = \begin{pmatrix} c_H \\ c_h \end{pmatrix}, \quad (4.5)$$

其中 $\mathbf{u}_H^{j+1} \in \mathbb{R}^{N_H}$ 和 $\alpha_{j+1} \in \mathbb{R}$.

由于在非线性迭代过程中 $\tilde{u}_h$ 不会改变, 所以向量 c_H 和 c_h 在非线性迭代过程中也不会改变. 但是矩阵 A_H , 向量 b_{Hh} 和标量 ξ 在非线性迭代过程中将会改变. 本节最主要的内容就是构造一个高效的算法在每次非线性迭代的时候来组装矩阵 A_H , 向量 b_{Hh} 和标量 ξ . 这里用到的最主要性质是在非线性迭代过程中, 定义在细网格 $\mathcal{T}_h$ 上的函数 $\tilde{u}_h$ 将保持不变. 同样是为了书写的方便, 我们去掉初值 u_h^j 的上标, 即 u_h^j 的展开式写成 $u_h^j = u_H + \alpha \tilde{u}_h^{(\ell)}$. 由有限维空间 $V_{H,h}$ 和非线性迭代中线性问题 (4.4) 的定义, 矩阵 A_H 有如下的展开式

$$\begin{aligned} (A_H)_{i,j} &= \int_{\Omega} \nabla \phi_{i,H} \nabla \phi_{j,H} d\Omega + \int_{\Omega} w \phi_{i,H} \phi_{j,H} d\Omega + \int_{\Omega} \zeta(u_H + \alpha_j \tilde{u}_h)^2 \phi_{i,H} \phi_{j,H} d\Omega \\ &:= (A_{H,1})_{i,j} + (A_{H,2})_{i,j}, \end{aligned} \quad (4.6)$$

其中

$$(A_{H,1})_{i,j} = \int_{\Omega} \nabla \phi_{i,H} \nabla \phi_{j,H} d\Omega + \int_{\Omega} w \phi_{i,H} \phi_{j,H} d\Omega \quad (4.7)$$

和

$$\begin{aligned} (A_{H,2})_{i,j} &= \int_{\Omega} \zeta(u_H + \alpha \tilde{u}_h)^2 \phi_{i,H} \phi_{j,H} d\Omega \\ &= \int_{\Omega} \zeta((u_H)^2 + 2\alpha u_H \tilde{u}_h + \alpha^2 (\tilde{u}_h)^2) \phi_{i,H} \phi_{j,H} d\Omega \\ &= \int_{\Omega} \zeta(u_H)^2 \phi_{i,H} \phi_{j,H} d\Omega + 2\alpha \int_{\Omega} \zeta \tilde{u}_h u_H \phi_{i,H} \phi_{j,H} d\Omega + \alpha^2 \int_{\Omega} \zeta(\tilde{u}_h)^2 \phi_{i,H} \phi_{j,H} d\Omega \\ &:= (A_{H,2,1})_{i,j} + 2\alpha (A_{H,2,2})_{i,j} + \alpha^2 (A_{H,2,3})_{i,j}. \end{aligned} \quad (4.8)$$

显然矩阵 $A_{H,2,1}$ 有如下的计算方式

$$(A_{H,2,1})_{i,j} = \int_{\Omega} \zeta(u_H)^2 \phi_{i,H} \phi_{j,H} d\Omega, \quad (4.9)$$

计算只需要在粗网格 $\mathcal{T}_H$ 上进行, 相应的计算量为 $\mathcal{O}(N_H)$. 并且矩阵 $A_{H,1}$ 和如下定义的矩阵 $A_{H,2,3}$

$$(A_{H,2,3})_{i,j} = \int_{\Omega} \zeta(\tilde{u}_h)^2 \phi_{i,H} \phi_{j,H} d\Omega \quad (4.10)$$

在非线性迭代过程中不会改变, 只需要进行一次组装即可.

矩阵 $A_{H,2,2}$ 有如下的展开式

$$(A_{H,2,2})_{i,j} = \sum_{k=1}^{N_H} u_k \int_{\Omega} \zeta \tilde{u}_h \phi_{k,H} \phi_{i,H} \phi_{j,H} d\Omega. \quad (4.11)$$

展开式 (4.11) 提示我们可以定义一个张量 T_H 如下

$$(T_H)_{i,j,k} = \int_{\Omega} \zeta \tilde{u}_h \phi_{k,H} \phi_{i,H} \phi_{j,H} d\Omega. \quad (4.12)$$

定义张量 T_H 之后, 矩阵 $A_{H,2,2}$ 可以有如下的计算方式

$$A_{H,2,2} = T_H \cdot \mathbf{u}_H, \quad (4.13)$$

其中 $T_H \cdot \mathbf{u}_H$ 表示张量 T_H 和向量 $\mathbf{u}_H$ 关于最后一个指标 k 的乘积. 由张量 T_H 的定义 (4.12) 可以知道张量 T_H 的维数为 $\mathbb{R}^{N_H \times N_H \times N_H}$, 并且非零元素的个数为 $\mathcal{O}(N_H)$. 由此可以知道张量 T_H 是一个稀疏张量, 并且张量与向量乘积 (4.13) 的计算量为 $\mathcal{O}(N_H)$.

现在来考虑向量 b_{Hh} 的计算. 同样由空间 $V_{H,h}$ 和问题 (4.4) 的定义, 向量 b_{Hh} 有如下的展开式

$$\begin{aligned} (b_{Hh})_i &= \int_{\Omega} \nabla \tilde{u}_h \nabla \phi_{i,H} d\Omega + \int_{\Omega} w \tilde{u}_h \phi_{i,H} d\Omega + \int_{\Omega} \zeta(u_H + \alpha \tilde{u}_h)^2 \tilde{u}_h \phi_{i,H} d\Omega \\ &:= (b_{Hh,1})_i + (b_{Hh,2})_i, \end{aligned} \quad (4.14)$$

其中

$$(b_{Hh,1})_i = \int_{\Omega} \nabla \tilde{u}_h \nabla \phi_{i,H} d\Omega + \int_{\Omega} w \tilde{u}_h \phi_{i,H} d\Omega, \quad (4.15)$$

和

$$\begin{aligned} (b_{Hh,2})_i &= \int_{\Omega} \zeta(u_H + \alpha \tilde{u}_h)^2 \tilde{u}_h \phi_{i,H} d\Omega = \int_{\Omega} \zeta((u_H)^2 + 2\alpha \tilde{u}_h u_H + \alpha^2 (\tilde{u}_h)^2) \tilde{u}_h \phi_{i,H} d\Omega \\ &= \int_{\Omega} \zeta(u_H)^2 \tilde{u}_h \phi_{i,H} d\Omega + 2\alpha \int_{\Omega} \zeta(\tilde{u}_h)^2 u_H \phi_{i,H} d\Omega + \alpha^2 \int_{\Omega} \zeta(\tilde{u}_h)^3 \phi_{i,H} d\Omega \\ &:= (b_{Hh,2,1})_i + 2\alpha (b_{Hh,2,2})_i + \alpha^2 (b_{Hh,2,3})_i. \end{aligned} \quad (4.16)$$

可以知道向量 $b_{Hh,1}$ 在非线性迭代的过程中不会改变, 那么只需要考虑向量 $b_{Hh,2}$ 的计算.

首先关于向量 $b_{Hh,2,1}$ 的计算可以采用如下的执行方式

$$(b_{Hh,2,1})_i = \int_{\Omega} \left(\sum_{j=1}^{N_H} u_j \phi_{j,H} \right)^2 \tilde{u}_h \phi_{i,H} d\Omega = \sum_{j=1}^{N_H} \sum_{k=1}^{N_H} u_j u_k \int_{\Omega} \tilde{u}_h \phi_{j,H} \phi_{k,H} \phi_{i,H} d\Omega. \quad (4.17)$$

利用张量 T_H , 向量 $b_{Hh,2,1}$ 可以采用如下的张量与向量的乘积来计算

$$b_{Hh,2,1} = (T_H \cdot \mathbf{u}_H) \cdot \mathbf{u}_H = A_{H,2,2} \mathbf{u}_H, \quad (4.18)$$

其中 $(T_H \cdot \mathbf{u}_H) \cdot \mathbf{u}_H$ 表示张量 T_H 与向量 u_H 关于最后两个指标 k 和 j 的乘积. 类似地, 张量与向量的乘积 (4.18) 也仅需 $\mathcal{O}(N_H)$ 的计算量.

然后可以用如下的方式来生成向量 $b_{Hh,2,2}$

$$(b_{Hh,2,2})_i = \sum_{j=1}^{N_H} u_j \int_{\Omega} \zeta(\tilde{u}_h)^2 \phi_{j,H} \phi_{i,H} d\Omega = (A_{H,2,3} \mathbf{u}_H)_i. \quad (4.19)$$

最后, 如下定义的向量 $b_{Hh,2,3}$

$$(b_{Hh,2,3})_i = \int_{\Omega} \zeta(\tilde{u}_h)^3 \phi_{i,H} d\Omega, \quad (4.20)$$

在非线性迭代的过程中也将不会改变.

现在我们来考虑标量 ξ 的计算. 显然标量 ξ 具有如下的展开形式

$$\begin{aligned} \xi &= \int_{\Omega} |\nabla \tilde{u}_h|^2 d\Omega + \int_{\Omega} w(\tilde{u}_h)^2 d\Omega + \int_{\Omega} \zeta(u_H + \alpha \tilde{u}_h)^2 (\tilde{u}_h)^2 d\Omega \\ &= \int_{\Omega} (|\nabla \tilde{u}_h|^2 + w(\tilde{u}_h)^2) d\Omega + \int_{\Omega} \zeta((u_H)^2 + 2\alpha u_H \tilde{u}_h + \alpha^2 (\tilde{u}_h)^2) (\tilde{u}_h)^2 d\Omega \\ &:= d_1 + d_2, \end{aligned} \quad (4.21)$$

其中

$$d_1 = \int_{\Omega} (|\nabla \tilde{u}_h|^2 + w(\tilde{u}_h)^2) d\Omega, \quad (4.22)$$

和

$$\begin{aligned} d_2 &= \sum_{i=1}^{N_H} \sum_{j=1}^{N_H} u_i u_j \int_{\Omega} \zeta(\tilde{u}_h)^2 \phi_{i,H} \phi_{j,H} d\Omega + 2\alpha \sum_{i=1}^{N_H} u_i \int_{\Omega} \zeta(\tilde{u}_h)^3 \phi_{i,H} d\Omega + \alpha^2 \int_{\Omega} \zeta(\tilde{u}_h)^4 d\Omega \\ &= \mathbf{u}_H^T A_{H,2,3} \mathbf{u}_H + 2\alpha \mathbf{u}_H^T b_{Hh,2,3} + \alpha^2 \xi_h. \end{aligned} \quad (4.23)$$

其中的标量 ξ_h 采用如下的计算方式

$$\xi_h = \int_{\Omega} \zeta(\tilde{u}_h)^4 d\Omega. \quad (4.24)$$

通过以上的讨论, 我们可以定义如下的非线性迭代算法来求解算法 1 中第二步的半线性椭圆方程 (3.7).

算法 3. 求解问题 (3.7) 的非线性迭代方法

1. 非线性迭代过程的准备: 首先按照 (4.12) 的方式计算张量 T_H , 然后由 (4.7) 和 (4.10) 计算向量 $A_{H,1}$ 和 $A_{H,2,3}$. 按照 (4.15) 和 (4.20) 的方式计算向量 $b_{Hh,1}$ 和 $b_{Hh,2,3}$, 同时按照 (4.22) 和 (4.24) 的方式计算标量 d_1 和 ξ_h .

2. 非线性迭代 (迭代指标为 j):

- 按照 (4.9) 和 (4.13) 的方式计算矩阵 $A_{H,2,1}$ 和 $A_{H,2,2}$, 然后计算 $A_H = A_{H,1} + A_{H,2,1} + 2\alpha A_{H,2,2} + \alpha^2 A_{H,2,3}$ 来得到矩阵 A_H .
- 依照 (4.18) 和 (4.19) 的方式来得到向量 $b_{Hh,2,1}$ 和 $b_{Hh,2,2}$, 然后计算向量 $b_{Hh} = b_{Hh,1} + b_{Hh,2,1} + 2\alpha b_{Hh,2,2} + \alpha^2 b_{Hh,2,3}$.
- 按照 (4.23) 的方式计算得到标量 d_2 , 然后计算标量 ξ 为 $\xi = d_1 + d_2$.
- 求解线性方程组 (4.5) 得到解 $(u_H^{j+1}, \alpha_{j+1})$.
- 如果非线性迭代满足精度要求, 则非线性迭代停止, 否则继续进行非线性迭代以达到精度要求.

3. 最后得到有限元函数 $u_h^{j+1} = u_H^{j+1} + \alpha_{j+1} \tilde{u}_h = \sum_{i=1}^{N_H} u_i^{j+1} \phi_{i,H} + \alpha_{j+1} \tilde{u}_h$ 作为输出.

注 4. 在算法 3 中的第一步组装张量、矩阵、向量和标量需要的计算量为 $\mathcal{O}(N_h)$. 但是在每一次非线性迭代中 (算法 3 中的第二步) 只需要 $\mathcal{O}(M_H)$ 的计算量. 这里 M_H 表示求解线性方程 (4.5) 的计算量且通常 $M_H \geq N_H$. 假设需要进行 ϖ 次的非线性迭代, 则算法 3 所需要的计算量为 $\mathcal{O}(N_h + \varpi M_H)$.

基于本节前面的讨论, 我们可以利用算法 3 执行算法 1 中的第二步, 从而可以定义一种高效的完全多重网格算法如下:

算法 4. 高效的完全多重网格算法

- 构造满足条件 (3.2) 和 (3.3) 的嵌套有限元空间序列 $V_H, V_{h_1}, V_{h_2}, \dots, V_{h_n}$.
- 在 V_{h_1} 中求解如下半线性问题: 求 $u_{h_1} \in V_{h_1}$, 使得

$$(\nabla u_{h_1}, \nabla v_{h_1}) + (w u_{h_1}, v_{h_1}) + (\zeta(u_{h_1})^3, v_{h_1}) = (g, v_{h_1}), \quad \forall v_{h_1} \in V_{h_1}.$$

- 对 $k = 2, \dots, n$ 进行如下的迭代

- 令 $u_{h_k}^{(0)} = u_{h_{k-1}}$.
- 对 $\ell = 0, \dots, p-1$, 利用算法 3 来求解算法 1 中的非线性问题 (3.7), 且进行如下的迭代:

$$u_{h_k}^{(\ell+1)} = \text{SemilinearMG}(V_H, u_{h_k}^{(\ell)}, V_{h_k}).$$

- 定义 $u_{h_k} = u_{h_k}^{(p)}$.

最后得到一个近似解 $u_{h_n} \in V_{h_n}$.

为了体现算法 4 的效率优势, 我们来考虑算法 4 的计算量.

定理 4. 假设在算法 4 的求解过程中只使用 1 个计算节点, 在低维空间 V_{H,h_k} ($k = 2, \dots, n$) 以及 V_{h_1} 上求解半线性椭圆问题需要的工作量分别为 $\mathcal{O}(M_H)$ 和 $\mathcal{O}(M_{h_1})$, 在每一层有限元空间 V_{h_k} 上的多重网格迭代 $\text{MG}(V_{h_k}, -f(x, u_{h_k}^{(\ell)}) + g, u_{h_k}^{(\ell)}, m)$ 需要的计算量为 $\mathcal{O}(N_k)$, 其中 $k = 2, 3, \dots, n$. 用 ϖ 表示求解半线性椭圆问题 (3.7) 的非线性迭代次数. 则算法 4 所需要的计算量为

$$\text{Total work} = \mathcal{O}(N_n + \varpi M_H \log N_n + \varpi M_{h_1}). \quad (4.25)$$

证明. 证明方法本质上与定理 3 的证明方法一样. 令 W_k 表示在第 k 层有限元空间 V_{h_k} 的计算量. 则由算法 4、单步校正算法 1 和算法 3 的定义可得

$$W_k = \mathcal{O}(N_k + \varpi M_H). \quad (4.26)$$

递归地使用 (4.26), 同时利用关系式 (3.25), 可以推出

$$\begin{aligned} \text{Total work} &= \sum_{k=1}^n W_k = \mathcal{O}\left(\varpi M_{h_1} + \sum_{k=2}^n (N_k + \varpi M_H)\right) \\ &= \mathcal{O}\left(\sum_{k=2}^n N_k + (n-1)\varpi M_H + \varpi M_{h_1}\right) \\ &= \mathcal{O}\left(\sum_{k=2}^n \left(\frac{1}{\beta}\right)^{d(n-k)} N_n + \varpi M_H \log N_n + \varpi M_{h_1}\right) \\ &= \mathcal{O}(N_n + \varpi M_H \log N_n + \varpi M_{h_1}). \end{aligned}$$

这就是所需要证明的结论 (3.26).

注 5. 利用算法 3 进行非线性迭代, 可以得到计算量的估计为 $\mathcal{O}(N_n + \varpi M_H \log N_n + \varpi M_{h_1})$. 也就是说非线性迭代次数 ϖ 对计算量的影响是通过与 M_H 和 M_{h_1} 相乘得到的. 我们知道在实际计算过程中往往 $M_H \ll N_{h_n}$ 和 $M_{h_1} \ll N_{h_n}$ 成立, 如此可以得到总的计算量几乎不受非线性迭代次数的影响, 或者说计算量具有绝对渐近最优的性质.

5. 数值算例

本节用一个数值算例来验证在算法 2 和 4 中定义的多重网格方法的有效性. 这里, 我们关心算法 2 和 4 对于不同 ζ 的多项式半线性问题 (4.1) 的计算时间 (以秒为单位). 值得注意的是 ζ 的大小表示了问题的非线性强弱. 为了验证算法 4 的计算量估计 (4.25) 的合理性, 这里只在串行机器上对算法 2 和 4 进行了测试.

例 1. 本例中求解多项式型半线性椭圆问题 (4.1), 计算区域 Ω 为单位正方形区域 $(0, 1) \times (0, 1)$, 其中 $w = 1$, 选取合适的右端项 g 使得精确解为 $u = \sin \pi x \sin \pi y$, 非线性系数 ζ 分别取 1, 10, 20 和 30, 并对计算时间进行对比.

从初始网格 $\mathcal{T}_H$ 出发, 采用 $\beta = 2$ (连接各边中点) 的一致加密构造一系列嵌套网格和有限元空间序列. 在本例中, 粗网格 $\mathcal{T}_H$ 如图 1 所示, 第一层网格 $\mathcal{T}_{h_1}$ 为从初始网格加密 3 次得到的网格. 在每层网格上求解线性方程 (3.4) 是使用了 1 次的多重网格迭代 (即 $m = 1$), 每次多重网格迭代的前后光滑各采用 1 次的共轭梯度步.

这里我们测试了算法 2 和 4 对不同的非线性系数 ζ 的 CPU 计算时间 (秒). 图 2 展示了两种算法的 CPU 计算时间. 从图 2 可以看出, 当计算规模足够大时, 算法 4 中的非线性迭代过程的计算量远小于原始算法 2 中非线性迭代过程的计算量. 由定理 3 可知, 算法 2 的总体计算量与离散规模也有线性关系, 但这种线性关系依赖于问题的非线性强度 (ζ 和 ϖ 的大小). 众所周知, 当 ζ 值越大时, 问题 4.1 的非线性越强, 所需要的非线性迭代次数越多 (ϖ 越大). 图 2 的结果表明, 当计算规模足够大时, 算法 4 的计算量几乎不依赖于问题 (4.1) 的非线性强弱, 也就是渐近计算量达到了绝对的最优 (与非线性迭代次数无关), 这也验证了定理 4 中关于计算量的估计式 (4.25).

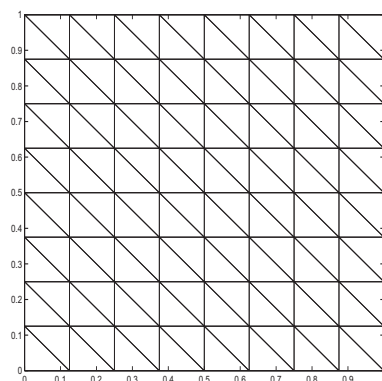


图 1 初始网格

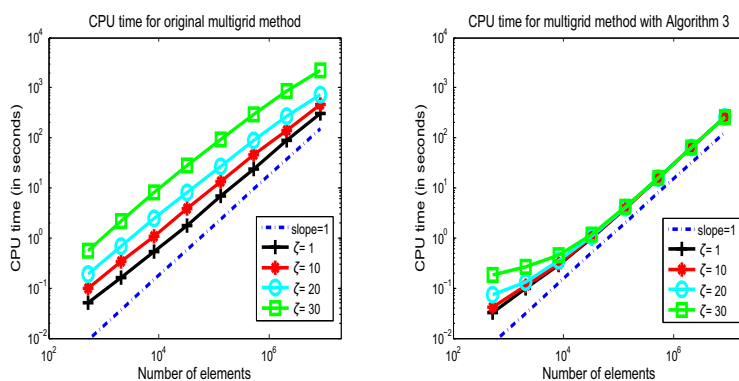


图 2 左图为原始多重网格方法的 CPU 时间, 右图为高效多重网格方法的 CPU 时间

6. 总 结

本文根据多水平校正的思想提出了一种求解对半线性椭圆问题的快速完全多重网格算法及其相应的高效实现方法. 基于非线性迭代新的实现方法, 求解非线性问题的渐近计算量与用多重网格方法求解相应的线性边值问题的渐近计算量基本相同, 几乎与非线性强弱无关. 同时也通过数值算例验证了该方法的有效性, 揭示了渐近计算量确实与非线性迭代次数无关.

本文的思想和方法也可以推广到其他具有多项式非线性问题, 如 Navier-Stokes 方程和一些相场模型等. 此外, 我们还可以利用这里的算法来设计一般非线性问题和非线性特征值问题的预优子. 比如求解如下一般形式的非线性问题:

$$\begin{cases} -\nabla \cdot (\mathcal{A} \nabla u) + f(x, u) = g, & \text{在 } \Omega \text{ 内,} \\ u = 0, & \text{在 } \partial\Omega \text{ 上,} \end{cases} \quad (6.1)$$

我们知道 Newton 迭代方法是基于对非线性问题的一阶多项式的近似, 具体形式如下

$$\begin{cases} -\nabla \cdot (\mathcal{A}\nabla u) + \frac{\partial f}{\partial u}(x, u_0)(u - u_0) = g - f(x, u_0), & \text{在 } \Omega \text{ 内,} \\ u = 0, & \text{在 } \partial\Omega \text{ 上.} \end{cases}$$

通过文中对算法 4 的计算量分析可以知道对于多项式形式的非线性问题也可以达到计算量渐近最优的程度, 即求解这种形式的非线性问题与求解线性问题的计算量相当. 自然地我们可以用高阶多项式的非线性问题来作为一般非线性问题的迭代格式, 或者说用作求解非线性问题的预条件方法. 比如三阶多项式预条件格式可以定义为如下的形式:

$$\begin{cases} -\nabla \cdot (\mathcal{A}\nabla u) + \frac{\partial f}{\partial u}(x, u_0)(u - u_0) + \frac{1}{2} \frac{\partial^2 f}{\partial u^2}(x, u_0)(u - u_0)^2 \\ + \frac{1}{6} \frac{\partial^3 f}{\partial u^3}(x, u_0)(u - u_0)^3 = g - f(x, u_0) & \text{在 } \Omega \text{ 内,} \\ u = 0, & \text{在 } \partial\Omega \text{ 上,} \end{cases}$$

其中 u_0 是已知的初始解. 利用本文所介绍的算法 4 求解上面多项式形式的非线性问题只需要渐近最优的计算量, 即与一步 Newton 迭代格式的计算量相当. 但由于使用的是高阶多项式的逼近, 这样的非线性迭代的收敛速度将会远快于 Newton 迭代从而使最终的计算量有大幅度的下降. 未来我们会将这种思想应用于更多的非线性问题的求解.

参 考 文 献

- [1] Adams R A. Sobolev Spaces. Academic Press, New York, 1975.
- [2] Bank R E, Dupont T. An optimal order process for solving finite element equations[J]. Math. Comp., 1981, 36: 35–51.
- [3] Bramble J H. Multigrid Methods. Pitman Research Notes in Mathematics, Vol. 294, John Wiley and Sons, 1993.
- [4] Bramble J H, Pasciak J E. New convergence estimates for multigrid algorithms[J]. Math. Comp., 1987, 49: 311–329.
- [5] Bramble J H, Zhang X. The Analysis of Multigrid Methods. Handbook of Numerical Analysis, 2000, 173–415.
- [6] Brandt A, McCormick S, Ruge J. Multigrid methods for differential eigenproblems[J]. SIAM J. Sci. Stat. Comput., 1983, 4(2): 244–260.
- [7] Brenner S, Scott L. The Mathematical Theory of Finite Element Methods. New York, Springer-Verlag, 1994.
- [8] Ciarlet P G. The Finite Element Method for Elliptic Problem. North-holland Amsterdam, 1978.
- [9] Hackbusch W. Multi-grid Methods and Applications. Springer-Verlag, Berlin, 1985.
- [10] Huang Y, Shi Z, Tang T, Xue W. A multilevel successive iteration method for nonlinear elliptic problem[J]. Math. Comp., 2004, 73: 525–539.
- [11] Jia S, Xie H, Xie M, Xu F. A full multigrid method for monlinear eigenvalue problems[J]. Sci. China Math., 2016, 59: 2037–2048.
- [12] 林群, 谢和虎. 有限元 Aubin-Nitsche 技巧新认识及其应用 [J]. 数学的实践与认识, 2011, 41(17): 247–258.
- [13] Lin Q, Xie H. A multi-level correction scheme for eigenvalue problems[J]. Math. Comp., 2015, 84(291): 71–88.

- [14] Lin Q, Xie H, Xu F. Multilevel correction adaptive finite element method for semilinear elliptic equation[J]. Appl. Math., 2015, 60(5): 527–550.
- [15] Scott L, Zhang S. Higher dimensional non-nested multigrid methods[J]. Math. Comp., 1992, 58: 457–466.
- [16] Toselli A, Widlund O. Domain Decomposition Methods: Algorithm and Theory. Springer-Verlag, Berlin Heidelberg, 2005.
- [17] Shaidurov V V. Multigrid Methods for Finite Elements. Kluwer Academic Publics, Netherlands, 1995.
- [18] Xie H. A type of multilevel method for the Steklov eigenvalue problem[J]. IMA J. Numer. Anal., 2014, 34: 592–608.
- [19] Xie H. A multigrid method for eigenvalue problem[J]. J. Comput. Phys., 2014, 274: 550–561.
- [20] Xie H. 非线性特征值问题的多重网格算法. 中国科学: 数学, 2015, 45(8): 1193–1204.
- [21] Xie H, Xie M. A multigrid method for the ground state solution of Bose-Einstein condensates[J]. Commun. Comput. Phys., 2016, 19(3): 648–662.
- [22] Xu J. Iterative methods by space decomposition and subspace correction[J]. SIAM Review, 1992, 34(4): 581–613.
- [23] Xu J. Two-grid discretization techniques for linear and nonlinear PDEs[J]. SIAM J. Numer. Anal., 1996, 33(5): 1759–1777.
- [24] Xu J. A novel two-grid method for semilinear elliptic equations[J]. SIAM J. Sci. Comput., 1994, 15(1): 231–237.