

A CASCADIC MULTIGRID METHOD FOR SEMILINEAR ELLIPTIC EQUATIONS *

Fei Xu

*Beijing Institute for Scientific and Engineering Computing, Beijing University of Technology,
Beijing 100124, China
Email: xufei@lsec.cc.ac.cn*

Fusheng Luo

*The Third Institute of Oceanography, SOA, Xiamen 361005, China
Email: luofusheng@tio.org.cn*

Abstract

This paper introduces a cascadic multigrid method for solving semilinear elliptic equations based on a multilevel correction method. Instead of the common costly way of directly solving semilinear elliptic equation on a very fine space, the new method contains some smoothing steps on a series of multilevel finite element spaces and some solving steps to semilinear elliptic equations on a very coarse space. To prove the efficiency of the new method, we derive two results, one of the optimal convergence rate by choosing the appropriate sequence of finite element spaces and the number of smoothing steps, and the other of the optimal computational work by applying the parallel computing technique. Moreover, the requirement of bounded second order derivatives of nonlinear term in the existing multigrid methods is reduced to a bounded first order derivative in the new method. Some numerical experiments are presented to validate our theoretical analysis.

Mathematics subject classification: 65N30, 65N25, 65L15, 65B99.

Key words: Semilinear elliptic equation, Parallel computing, Cascadic multigrid, Multilevel correction, Finite element method.

1. Introduction

The purpose of this paper is to study the multigrid finite element method for semilinear elliptic problems. As we know, the multigrid methods [4–6, 9, 13, 19, 24] provide optimal order algorithms for solving boundary value problems. The error bounds of the approximate solutions obtained from these efficient numerical algorithms are comparable to the theoretical bounds determined by the finite element discretization. In the past decades, the multigrid method is also applied to nonlinear elliptic problem to improve the efficiency of nonlinear elliptic problem solving, i.e. [19, 25, 26]. In these methods, the Newton iteration is adopted to linearize the nonlinear equations which require bounded second order derivatives of the nonlinear terms. For more information, please refer to [15, 19, 25] and the references cited therein.

Recently, a type of multigrid method for eigenvalue problems has been proposed in [17, 23]. And the corresponding idea can be found in [7, 11, 16]. The aim of this paper is to present a cascadic multigrid method for solving semilinear elliptic equations based on the combination of the multilevel correction method [17, 23] and the cascadic multigrid method for boundary value problems. Similarly to the cascadic multigrid method for the boundary value problem [2, 20],

* Received March 22, 2017 / Revised version received September 4, 2017 / Accepted October 31, 2017 /
Published online August 14, 2018 /

we only do the smoothing steps for the involved boundary value problems. Besides, we need to solve some semilinear elliptic equations on a low dimensional space. By organizing suitable numbers of smoothing iteration steps in different levels, the final approximate solution can have the same accuracy as the solution of standard finite element method. In this new version of multigrid method, solving semilinear elliptic problem will not be much more difficult than the multigrid scheme for the corresponding linear boundary value problems. Compared with the existing multigrid method for the semilinear problem, our new method only require a bounded first order derivative of the nonlinear term.

During the numerical calculation, computational complexity and memory consumption increase exponentially with the growth of the scale. As we know, distributed parallel computing can balance the load on each computing node, which will play an important role in the simulation of large scale systems. So as to improve the computational efficiency, we will use the parallel technique to design an algorithm with good scalability.

An outline of the paper goes as follows. In Section 2, we introduce the finite element method for the semilinear elliptic equation as well as some important properties. The Section 3 is the main part of the paper, where a type of cascadic multigrid algorithm for solving the semilinear elliptic equation and the corresponding error estimate are given. In Section 4, we add the parallel technique to the cascadic multigrid algorithm in Section 3 and estimate the computational work for the parallel algorithm. Four numerical examples are presented in Section 5 to validate our theoretical analysis. Some concluding remarks are given in the last section.

2. Finite Element Method for Semilinear Elliptic Equation

In this paper, the letter C (with or without subscripts) is used to denote a constant which may be different at different places. For convenience, the symbols $x_1 \lesssim y_1$, $x_2 \gtrsim y_2$ and $x_3 \approx y_3$ mean that $x_1 \leq C_1 y_1$, $x_2 \geq c_2 y_2$ and $c_3 x_3 \leq y_3 \leq C_3 x_3$. Let $\Omega \subset \mathcal{R}^d$ ($d = 2, 3$) denote a bounded convex domain with Lipschitz boundary $\partial\Omega$. We use the standard notation for Sobolev spaces $W^{s,p}(\Omega)$ and their associated norms $\|\cdot\|_{s,p,\Omega}$ and seminorms $|\cdot|_{s,p,\Omega}$ (see, e.g, [1]). For $p = 2$, we denote $H^s(\Omega) = W^{s,2}(\Omega)$ and $H_0^1(\Omega) = \{v \in H^1(\Omega) : v|_{\partial\Omega} = 0\}$, where $v|_{\partial\Omega} = 0$ is in the sense of trace. For simplicity, we use $\|\cdot\|_s$ to denote $\|\cdot\|_{s,2,\Omega}$ and V to denote $H_0^1(\Omega)$ in the rest of the paper.

We consider the following type of semilinear elliptic equation:

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

where $\mathcal{A} = (a_{i,j})_{d \times d}$ is a symmetric positive definite matrix with $a_{i,j} \in W^{1,\infty}$ ($i, j = 1, 2, \dots, d$), and $f(x, u)$ is a nonlinear function corresponding to the second variable and satisfies the following assumption.

Assumption A: The nonlinear function $f(x, \cdot)$ has a non-negative derivative in the second argument

$$0 \leq \frac{\partial f}{\partial v}(x, v) \leq C_f, \quad \forall x \in \Omega \quad \text{and} \quad \forall v \in V. \quad (2.2)$$

The weak form of the semilinear problem (2.1) can be described as: Find $u \in V$ such that

$$a(u, v) + (f(x, u), v) = (g, v), \quad \forall v \in V, \quad (2.3)$$

where

$$a(u, v) = (\mathcal{A}\nabla u, \nabla v). \quad (2.4)$$

Obviously, $a(u, v)$ is bounded and coercive on V , i.e.,

$$a(u, v) \leq C_a \|u\|_{1,\Omega} \|v\|_{1,\Omega} \quad \text{and} \quad c_a \|u\|_{1,\Omega}^2 \leq a(u, u), \quad \forall u, v \in V. \quad (2.5)$$

we take the norm $\|w\|_a := \sqrt{a(w, w)}$ for any $w \in V$ in this paper instead of the standard norm $\|\cdot\|_1$.

Now, we introduce the finite element method for semilinear elliptic problem (2.3). First we generate a shape regular decomposition of the computing domain $\Omega \subset \mathcal{R}^d$ ($d = 2, 3$) into triangles or rectangles for $d = 2$, tetrahedrons or hexahedrons for $d = 3$ (cf. [10, 12]). The mesh diameter h describes the maximum diameter of all cells $K \in \mathcal{T}_h$. Based on the mesh \mathcal{T}_h , we construct the finite element space $V_h \subset V$. For simplicity, we set V_h as the linear finite element space defined as follows

$$V_h = \left\{ v_h \in C(\Omega) \mid v_h|_K \in \mathcal{P}_1, \quad \forall K \in \mathcal{T}_h \right\} \cap H_0^1(\Omega), \quad (2.6)$$

where \mathcal{P}_1 denotes the linear function space.

The standard finite element scheme for semilinear equation (2.3) is: Find $\bar{u}_h \in V_h$ such that

$$a(\bar{u}_h, v_h) + (f(x, \bar{u}_h), v_h) = (g, v_h), \quad \forall v_h \in V_h. \quad (2.7)$$

Denote a linearized operator $L : H_0^1(\Omega) \rightarrow H^{-1}(\Omega)$ by:

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

In order to deduce the global prior error estimates, we introduce $\eta_a(V_h)$ as follows:

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

It is easy to know that $\eta_a(V_h) \rightarrow 0$ as $h \rightarrow 0$ (cf. [10, 12]).

Denote

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

From [19], we can give the following error estimates.

Lemma 2.1. *When Assumption A is satisfied, Eqs. (2.3) and (2.7) are uniquely solvable and the following estimates hold*

$$\|u - \bar{u}_h\|_a \lesssim \delta_h(u), \quad (2.8)$$

$$\|u - \bar{u}_h\|_0 \lesssim \eta_a(V_h) \|u - \bar{u}_h\|_a. \quad (2.9)$$

Furthermore, we have the following estimate

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

Proof. The desired results (2.8)-(2.9) can be derived from Lemmas 6.2.1 and 6.2.2 in [19]. Here we only give the proof for the estimate (2.10). For this aim, we define the finite element projection operator P_h by the following equation

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

It is easy to know that $\|u - P_h u\|_a = \delta_h(u)$. From (2.2) and (2.9), we have

$$\begin{aligned} a(P_h u - \bar{u}_h, v_h) &= a(u - \bar{u}_h, v_h) = (f(x, \bar{u}_h) - f(x, u), v_h) \\ &\leq C_f \|u - \bar{u}_h\|_0 \|v_h\|_0 \\ &\leq C\eta_a(V_h) \|u - \bar{u}_h\|_a \|v_h\|_a, \quad \forall v_h \in V_h. \end{aligned} \quad (2.11)$$

Combining (2.11) and the triangle inequality leads to the following estimates

$$\begin{aligned} \|u - \bar{u}_h\|_a &\leq \|u - P_h u\|_a + \|P_h u - \bar{u}_h\|_a \\ &\leq \delta_h(u) + C\eta_a(V_h) \|u - \bar{u}_h\|_a, \end{aligned} \quad (2.12)$$

which means that

$$\|u - \bar{u}_h\|_a \leq \frac{1}{1 - C\eta_a(V_h)} \delta_h(u) \leq (1 + C\eta_a(V_h)) \delta_h(u). \quad (2.13)$$

This is the desired result (2.10) and the proof is completed. \square

3. Cascadic Multigrid Method for Semilinear Elliptic Equations

In this section, a type of cascadic multigrid method for semilinear problems is proposed based on the multilevel correction scheme in [17, 23]. The main idea in this method is to approximate the underlying boundary value problems defined on each level by applying some simple smoothing iteration steps. In order to describe the cascadic multigrid method, we first introduce the sequence of finite element spaces and the properties of the concerned smoothers.

In order to design the multigrid scheme, we first generate a coarse mesh \mathcal{T}_H with the mesh size H and the coarse linear finite element space V_H is defined on it. Then we define a sequence of triangulations \mathcal{T}_{h_k} of $\Omega \subset \mathcal{R}^d$ as follows. Suppose \mathcal{T}_{h_1} (produced from \mathcal{T}_H by regular refinement) is given and let \mathcal{T}_{h_k} be obtained from $\mathcal{T}_{h_{k-1}}$ by a regular refinement step (produce β^d subelements) such that

$$h_k \approx \frac{1}{\beta} h_{k-1}, \quad (3.1)$$

where the positive number $\beta \geq 1$ (always equals 2) denotes the refinement index. Based on this sequence of meshes, we construct the corresponding nested linear finite element spaces such that

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

The nested spaces have the following relations of approximate accuracy

$$\eta_\alpha(V_H) \gtrsim \delta_{h_1}(u), \quad \delta_{h_k}(u) \approx \frac{1}{\beta} \delta_{h_{k-1}}(u), \quad k = 2, \dots, n. \quad (3.3)$$

In fact, since the computing domain is convex, we give the following assumptions

$$\eta_\alpha(V_H) \approx H, \quad \eta_\alpha(V_{h_k}) \approx h_k \quad \text{and} \quad \delta_{h_k}(u) \approx h_k, \quad k = 1, \dots, n. \quad (3.4)$$

Remark 3.1. The relation (3.3) is reasonable since we can choose $\delta_{h_k}(u) = h_k$ ($k = 1, \dots, n$). Always the upper bound of the estimate $\delta_{h_k}(u) \lesssim h_k$ holds. Recently, the lower bound result $\delta_{h_k}(u) \gtrsim h_k$ is also obtained (c.f. [18]).

For generality, we introduce a smoothing operator $S_h : V_h \rightarrow V_h$ which satisfies the following estimates

$$\begin{cases} \|S_h^m w_h\|_a \leq \frac{C}{m^\alpha} \frac{1}{h} \|w_h\|_0, \\ \|S_h^m w_h\|_a \leq \|w_h\|_a, \\ \|S_h^m(w_h + v_h)\|_a \leq \|S_h^m w_h\|_a + \|S_h^m v_h\|_a, \end{cases} \quad (3.5)$$

where C is a constant independent of h and α is some positive number depending on the choice of smoother. It is proved in [13, 19, 22] that the symmetric Gauss-Seidel, the SSOR, the damped Jacobi and the Richardson iteration are smoothers in the sense of (3.5) with parameter $\alpha = 1/2$ and the conjugate-gradient iteration is the smoother with $\alpha = 1$ (cf. [20, 21]).

Then we define the following notation

$$w_h = \text{Smooth}(V_h, f, \xi_h, m, S_h) \quad (3.6)$$

as the smoothing process for the following boundary value problem

$$a(u_h, v_h) = (f, v_h), \quad \forall v_h \in V_h, \quad (3.7)$$

where ξ_h denotes the initial value of the smoothing process, S_h denotes the chosen smoothing operator, m denotes the number of the iteration steps and w_h denotes the output of the smoothing process.

Now, we come to introduce the cascadic multigrid method for the semilinear elliptic equation (2.3). Assume we have obtained an approximate solution $u^{h_k} \in V_{h_k}$. We design the following cascadic type one correction step to improve the accuracy of the current approximation.

Algorithm 3.1. *Cascadic type of One Correction Step*

1. Define the following auxiliary source problem: Find $\hat{u}^{h_{k+1}} \in V_{h_{k+1}}$ such that

$$a(\hat{u}^{h_{k+1}}, v_{h_{k+1}}) = (g, v_{h_{k+1}}) - (f(x, u^{h_k}), v_{h_{k+1}}), \quad \forall v_{h_{k+1}} \in V_{h_{k+1}}. \quad (3.8)$$

Perform the smoothing process (3.6) to obtain a new approximate solution $\tilde{u}^{h_{k+1}} \in V_{h_{k+1}}$ by

$$\tilde{u}^{h_{k+1}} = \text{Smooth}(V_{h_{k+1}}, g - f(x, u^{h_k}), u^{h_k}, m_{k+1}, S_{h_{k+1}}). \quad (3.9)$$

2. Define a new finite element space $V_H^{h_{k+1}} = V_H + \text{span}\{\tilde{u}^{h_{k+1}}\}$ and solve the following semilinear elliptic equation: Find $u^{h_{k+1}} \in V_H^{h_{k+1}}$ such that

$$a(u^{h_{k+1}}, v_H^{h_{k+1}}) + (f(x, u^{h_{k+1}}), v_H^{h_{k+1}}) = (g, v_H^{h_{k+1}}), \quad \forall v_H^{h_{k+1}} \in V_H^{h_{k+1}}. \quad (3.10)$$

Summarize the above two steps by defining

$$u^{h_{k+1}} = \text{SmoothCorrection}(V_H, V_{h_{k+1}}, u^{h_k}, m_{k+1}, S_{h_{k+1}}).$$

Based on the above algorithm, i.e. the cascadic type of one correction step, we can construct a cascadic multigrid method for semilinear elliptic equation (2.3) as follows:

Algorithm 3.2. *Semilinear Elliptic Cascadic Multigrid Method*

1. Solve the following semilinear elliptic equation in the initial finite element space V_{h_1} :
Find $u^{h_1} \in V_{h_1}$ such that

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

2. For $k = 1, \dots, n-1$, do the following iteration

$$u^{h_{k+1}} = \text{SmoothCorrection}(V_H, V_{h_{k+1}}, u^{h_k}, m_{k+1}, S_{h_{k+1}}).$$

End Do

Finally, we obtain an approximate solution $u^{h_n} \in V_{h_n}$.

In order to analyze the convergence of Algorithm 3.2, we introduce an auxiliary algorithm and then show its superapproximate property. Similarly, assume we have obtained an approximate solution $\tilde{u}_{h_k} \in V_{h_k}$. The auxiliary one correction step is defined as follows.

Algorithm 3.3. *Auxiliary One Correction Step*

1. Solve the following auxiliary source problem: Find $\hat{u}_{h_{k+1}} \in V_{h_{k+1}}$ such that

$$a(\hat{u}_{h_{k+1}}, v_{h_{k+1}}) = (g, v_{h_{k+1}}) - (f(x, \tilde{u}_{h_k}), v_{h_{k+1}}), \quad \forall v_{h_{k+1}} \in V_{h_{k+1}}. \quad (3.11)$$

2. Define a new finite element space $\tilde{V}_{H, h_{k+1}} = V_H + \text{span}\{\hat{u}_{h_{k+1}}\} + \text{span}\{\tilde{u}^{h_{k+1}}\}$ and solve the following semilinear elliptic equation: Find $\tilde{u}_{h_{k+1}} \in \tilde{V}_{H, h_{k+1}}$ such that

$$a(\tilde{u}_{h_{k+1}}, \tilde{v}_{H, h_{k+1}}) + (f(x, \tilde{u}_{h_{k+1}}), \tilde{v}_{H, h_{k+1}}) = (g, \tilde{v}_{H, h_{k+1}}), \quad \forall \tilde{v}_{H, h_{k+1}} \in \tilde{V}_{H, h_{k+1}}. \quad (3.12)$$

Summarize the above two steps by defining

$$\tilde{u}_{h_{k+1}} = \text{AuxiliaryCorrection}(V_H, V_{h_{k+1}}, \tilde{u}_{h_k}, \tilde{u}^{h_{k+1}}).$$

Algorithm 3.4. *Semilinear Elliptic Auxiliary Multilevel Correction Method*

1. Solve the following semilinear elliptic equation in the initial finite element space V_{h_1} :
Find $\tilde{u}_{h_1} \in V_{h_1}$ such that

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

2. For $k = 1, \dots, n-1$, do the following iteration

$$\tilde{u}_{h_{k+1}} = \text{AuxiliaryCorrection}(V_H, V_{h_{k+1}}, \tilde{u}_{h_k}, \tilde{u}^{h_{k+1}}).$$

End Do

Finally, we obtain an approximate solution $\tilde{u}_{h_n} \in V_{h_n}$.

Before analyzing the convergence of Algorithm 3.2, we show a superapproximate property of \tilde{u}_{h_k} obtained by Algorithm 3.4.

Theorem 3.1. *Assume \tilde{u}_{h_k} ($k = 1, \dots, n$) are obtained by Algorithm 3.4 and \bar{u}_{h_k} ($k = 1, \dots, n$) denote the standard finite element solutions in V_{h_k} . If the sequence of finite element spaces V_{h_1}, \dots, V_{h_n} and the coarse finite element space V_H satisfy the following condition*

$$C\eta_a(V_H)\beta^2 < 1, \quad (3.13)$$

then the following estimates hold:

$$\|\bar{u}_{h_k} - \tilde{u}_{h_k}\|_a \leq C\eta_a(V_{h_k})\delta_{h_k}(u), \quad k = 1, \dots, n, \quad (3.14)$$

$$\|\bar{u}_{h_k} - \tilde{u}_{h_k}\|_0 \leq C\eta_a(V_{h_k})\delta_{h_k}(u), \quad k = 1, \dots, n. \quad (3.15)$$

Proof. From (2.7) and (3.11), we have

$$\begin{aligned} \hat{a}(\bar{u}_{h_{k+1}} - \hat{u}_{h_{k+1}}, v_{h_{k+1}}) &= (f(x, \tilde{u}_{h_k}) - f(x, \bar{u}_{h_{k+1}}), v_{h_{k+1}}) \\ &\lesssim \|\bar{u}_{h_{k+1}} - \tilde{u}_{h_k}\|_0 \|v_{h_{k+1}}\|_a \\ &\lesssim (\|\bar{u}_{h_{k+1}} - \bar{u}_{h_k}\|_0 + \|\bar{u}_{h_k} - \tilde{u}_{h_k}\|_0) \|v_{h_{k+1}}\|_a. \end{aligned}$$

It leads to the following estimate

$$\|\bar{u}_{h_{k+1}} - \hat{u}_{h_{k+1}}\|_a \leq C(\|\bar{u}_{h_{k+1}} - \bar{u}_{h_k}\|_0 + \|\bar{u}_{h_k} - \tilde{u}_{h_k}\|_0). \quad (3.16)$$

Note that the semilinear elliptic equation (3.12) can be regarded as a finite dimensional subspace approximation of the semilinear elliptic equation (2.7). From Lemma 2.1 and the second step in Algorithm 3.3, the following estimate holds:

$$\|\bar{u}_{h_{k+1}} - \tilde{u}_{h_{k+1}}\|_a \leq C \inf_{\tilde{v}_{H,h_{k+1}} \in \tilde{V}_{H,h_{k+1}}} \|\bar{u}_{h_{k+1}} - \tilde{v}_{H,h_{k+1}}\|_a \leq C\|\bar{u}_{h_{k+1}} - \hat{u}_{h_{k+1}}\|_a. \quad (3.17)$$

Then combining (3.16) and (3.17) leads to

$$\|\bar{u}_{h_{k+1}} - \tilde{u}_{h_{k+1}}\|_a \leq C(\|\bar{u}_{h_{k+1}} - \bar{u}_{h_k}\|_0 + \|\bar{u}_{h_k} - \tilde{u}_{h_k}\|_0). \quad (3.18)$$

From the properties of $V_{h_k} \subset V_{h_{k+1}}$, $\tilde{V}_{H,h_k} \subset V_{h_k}$, Lemma 2.1 and (3.3), we have

$$\begin{aligned} \|\bar{u}_{h_{k+1}} - \bar{u}_{h_k}\|_a &\leq C\delta_{h_k}(u), \quad \|\bar{u}_{h_{k+1}} - \bar{u}_{h_k}\|_0 \leq C\eta_a(V_{h_k})\|\bar{u}_{h_{k+1}} - \bar{u}_{h_k}\|_a, \\ \|\bar{u}_{h_k} - \tilde{u}_{h_k}\|_0 &\leq C\eta_a(V_H)\|\bar{u}_{h_k} - \tilde{u}_{h_k}\|_a. \end{aligned}$$

Substituting above inequalities into (3.18) leads to the following estimate

$$\|\bar{u}_{h_{k+1}} - \tilde{u}_{h_{k+1}}\|_a \leq C(\eta_a(V_{h_k})\delta_{h_k}(u) + \eta_a(V_H)\|\bar{u}_{h_k} - \tilde{u}_{h_k}\|_a). \quad (3.19)$$

When $k = 1$, since $\tilde{u}_{h_1} := \bar{u}_{h_1}$, we have

$$\|\bar{u}_{h_2} - \tilde{u}_{h_2}\|_a \leq C\eta_a(V_{h_1})\delta_{h_1}(u). \quad (3.20)$$

Based on (3.3), (3.19), (3.20) and recursive argument, we have the following estimates

$$\begin{aligned}
\|\bar{u}_{h_k} - \tilde{u}_{h_k}\|_a &\leq C \sum_{j=2}^k (C\eta_a(V_H))^{k-j} \eta_a(V_{h_{j-1}}) \delta_{h_{j-1}}(u) \\
&\leq C \sum_{j=2}^k (C\eta_a(V_H))^{k-j} \beta^{k-j+1} \eta_a(V_{h_k}) \beta^{k-j+1} \delta_{h_k}(u) \\
&\leq C\beta^2 \left(\sum_{j=2}^k (C\eta_a(V_H)\beta^2)^{k-j} \right) \eta_a(V_{h_k}) \delta_{h_k}(u) \\
&\leq \frac{C\beta^2}{1 - C\beta^2\eta_a(V_H)} \eta_a(V_{h_k}) \delta_{h_k}(u).
\end{aligned} \tag{3.21}$$

Therefore, the desired result (3.14) holds under the condition $C\eta_a(V_H)\beta^2 < 1$. Furthermore, (3.15) can be obtained directly from Lemma 2.1 and the property $\tilde{V}_{H,h_{k+1}} \subset V_{h_{k+1}}$. \square

Note that $V_H^{h_k} \subset \tilde{V}_{H,h_k}$, we can obtain the following estimates which will play an important role in our analysis.

Lemma 3.1. *Let u^{h_k} , $V_H^{h_k}$ and \tilde{u}_{h_k} , \tilde{V}_{H,h_k} be defined in Algorithms 3.1 and 3.3. Then the following estimates hold:*

$$\|u^{h_k} - \tilde{u}_{h_k}\|_a \leq C \|\hat{u}_{h_k} - \tilde{u}^{h_k}\|_a, \tag{3.22}$$

$$\|u^{h_k} - \tilde{u}_{h_k}\|_0 \leq C\eta_a(V_H) \|u^{h_k} - \tilde{u}_{h_k}\|_a. \tag{3.23}$$

Proof. Since $V_H^{h_k} \subset \tilde{V}_{H,h_k}$, according to (3.10) and (3.12), u^{h_k} can be viewed as a finite dimensional approximation of \tilde{u}_{h_k} . Then from Lemma 2.1 and the definitions of \tilde{V}_{H,h_k} and $V_H^{h_k}$, we have

$$\begin{aligned}
\|\tilde{u}_{h_k} - u^{h_k}\|_a &\leq C \inf_{v_H^{h_k} \in V_H^{h_k}} \|\tilde{u}_{h_k} - v_H^{h_k}\|_a \leq C \inf_{v_H^{h_k} \in V_H^{h_k}} \|\hat{u}_{h_k} - v_H^{h_k}\|_a \\
&\leq C \|\hat{u}_{h_k} - \tilde{u}^{h_k}\|_a,
\end{aligned} \tag{3.24}$$

which is the desired result (3.22).

Similarly, we also have (3.23) by the following argument

$$\|\tilde{u}_{h_k} - u^{h_k}\|_0 \leq C\eta_a(V_H^{h_k}) \|\tilde{u}_{h_k} - u^{h_k}\|_a \leq C\eta_a(V_H) \|\tilde{u}_{h_k} - u^{h_k}\|_a.$$

The proof is completed. \square

Remark 3.2. Since $V_H \subset V_H^{h_k}$ and $V_H \subset \tilde{V}_{H,h_k}$, from Lemma 2.1, we have

$$\|u^{h_k} - \tilde{u}_{h_k}\|_a \leq \|u^{h_k} - u\|_a + \|u - \tilde{u}_{h_k}\|_a \leq C\delta_H(u). \tag{3.25}$$

Now, we come to give the error estimates for Algorithm 3.2.

Theorem 3.2. *Assume the approximate solution u^{h_n} is obtained by Algorithm 3.2, \tilde{u}_{h_n} is obtained by Algorithm 3.4 and the smoother selected in each level V_{h_k} satisfy the smoothing*

property (3.5) for $k = 1, \dots, n$. Under the conditions of Theorem 3.1, we have the following estimates

$$\|\tilde{u}_{h_n} - u^{h_n}\|_a \leq C \sum_{k=2}^n \frac{(1 + C\eta_a(V_H))^{n-k}}{m_k^\alpha} \delta_{h_k}(u), \quad (3.26)$$

$$\|\tilde{u}_{h_n} - u^{h_n}\|_0 \leq C\eta_a(V_H) \sum_{k=2}^n \frac{(1 + C\eta_a(V_H))^{n-k}}{m_k^\alpha} \delta_{h_k}(u). \quad (3.27)$$

Proof. Define $e_{h_k} := u^{h_k} - \tilde{u}_{h_k}$ for $k = 1, \dots, n$. Then it is easy to see that $e_{h_1} = 0$.

From Lemma 3.1, the following inequalities hold

$$\begin{aligned} \|e_{h_{k+1}}\|_a &= \|u^{h_{k+1}} - \tilde{u}_{h_{k+1}}\|_a \leq C \|\hat{u}_{h_{k+1}} - \tilde{u}^{h_{k+1}}\|_a \\ &\leq C (\|\hat{u}_{h_{k+1}} - \hat{u}^{h_{k+1}}\|_a + \|\hat{u}^{h_{k+1}} - \tilde{u}^{h_{k+1}}\|_a). \end{aligned} \quad (3.28)$$

For the first term in (3.28), from (3.8), (3.11), Lemma 3.1 and (3.25), we have

$$\|\hat{u}_{h_{k+1}} - \hat{u}^{h_{k+1}}\|_a \leq C \|u^{h_k} - \tilde{u}_{h_k}\|_0 \leq C\eta_a(V_H) \|e_{h_k}\|_a. \quad (3.29)$$

For the second term in (3.28), due to (3.5) and (3.29), the following estimates hold

$$\begin{aligned} \|\hat{u}^{h_{k+1}} - \tilde{u}^{h_{k+1}}\|_a &= \|S_{h_{k+1}}^{m_{k+1}}(\hat{u}^{h_{k+1}} - u^{h_k})\|_a \\ &\leq \|S_{h_{k+1}}^{m_{k+1}}(\hat{u}^{h_{k+1}} - \tilde{u}_{h_k})\|_a + \|S_{h_{k+1}}^{m_{k+1}}(\tilde{u}_{h_k} - u^{h_k})\|_a \\ &\leq \|S_{h_{k+1}}^{m_{k+1}}(\hat{u}^{h_{k+1}} - \hat{u}_{h_{k+1}})\|_a + \|S_{h_{k+1}}^{m_{k+1}}(\hat{u}_{h_{k+1}} - \tilde{u}_{h_k})\|_a + \|\tilde{u}_{h_k} - u^{h_k}\|_a \\ &\leq \|\hat{u}_{h_{k+1}} - \hat{u}^{h_{k+1}}\|_a + \frac{C}{m_{k+1}^\alpha} \frac{1}{h_{k+1}} \|\hat{u}_{h_{k+1}} - \tilde{u}_{h_k}\|_0 + \|\tilde{u}_{h_k} - u^{h_k}\|_a \\ &\leq (1 + C\eta_a(V_H)) \|e_{h_k}\|_a + \frac{C}{m_{k+1}^\alpha} \frac{1}{h_{k+1}} \|\hat{u}_{h_{k+1}} - \tilde{u}_{h_k}\|_0. \end{aligned} \quad (3.30)$$

According to (3.3), (3.14), (3.16), Theorem 3.1 and its proof, we have

$$\begin{aligned} \|\hat{u}_{h_{k+1}} - \tilde{u}_{h_k}\|_0 &\leq \|\hat{u}_{h_{k+1}} - \bar{u}_{h_{k+1}}\|_0 + \|\bar{u}_{h_{k+1}} - \bar{u}_{h_k}\|_0 + \|\bar{u}_{h_k} - \tilde{u}_{h_k}\|_0 \\ &\leq C\eta_a(V_{h_{k+1}}) \delta_{h_{k+1}}(u). \end{aligned} \quad (3.31)$$

Combining (3.4) and (3.28)–(3.31), we derive

$$\|e_{h_{k+1}}\|_a \leq (1 + C\eta_a(V_H)) \|e_{h_k}\|_a + \frac{C}{m_{k+1}^\alpha} \delta_{h_{k+1}}(u), \quad k = 1, \dots, n-1. \quad (3.32)$$

Based on (3.32), the fact $e_{h_1} = 0$ and the recursive argument, the following estimates hold

$$\begin{aligned} \|e_{h_n}\|_a &\leq (1 + C\eta_a(V_H)) \|e_{h_{n-1}}\|_a + \frac{C}{m_n^\alpha} \delta_{h_n}(u) \\ &\leq (1 + C\eta_a(V_H))^2 \|e_{h_{n-2}}\|_a + (1 + C\eta_a(V_H)) \frac{C}{m_{n-1}^\alpha} \delta_{h_{n-1}}(u) + \frac{C}{m_n^\alpha} \delta_{h_n}(u) \\ &\leq C \sum_{k=2}^n (1 + C\eta_a(V_H))^{n-k} \frac{1}{m_k^\alpha} \delta_{h_k}(u). \end{aligned}$$

This is the desired result (3.26). The estimate (3.27) can be obtained from Lemma 2.1 and (3.26). \square

Corollary 3.1. *Under the conditions of Theorem 3.2, we have the following estimate*

$$\|\bar{u}_{h_n} - u^{h_n}\|_a \leq C \left(\eta_a(V_{h_n}) \delta_{h_n}(u) + \sum_{k=2}^n \frac{(1 + C\eta_a(V_H))^{n-k}}{m_k^\alpha} \delta_{h_k}(u) \right). \quad (3.33)$$

Define the dimension of each linear finite element space as

$$N_k := \dim V_{h_k}, \quad k = 1, \dots, n.$$

Then we have

$$N_k \approx \left(\frac{h_k}{h_n} \right)^{-d} N_n = \left(\frac{1}{\beta} \right)^{d(n-k)} N_n, \quad k = 1, \dots, n. \quad (3.34)$$

To give a precise analysis for the final error and complexity estimates, we assume the following inequality holds for the number of smoothing iterations in each level mesh:

$$\left(\frac{h_k}{h_n} \right)^\zeta \leq \frac{m_k^\alpha}{\bar{m}^\alpha} \leq \sigma \left(\frac{h_k}{h_n} \right)^\zeta, \quad k = 2, \dots, n-1, \quad (3.35)$$

where $\bar{m} = m_n$, $\sigma > 1$ and $\zeta > 1$ are some appropriate constants.

Now, we give the final error estimates for Algorithm 3.2.

Theorem 3.3. *Under the conditions (3.3), (3.35) and $\beta^{1-\zeta}(1 + CH) < 1$, for any given $\gamma \in (0, 1]$, the final error estimate*

$$\|u^{h_n} - \tilde{u}_{h_n}\|_a \leq \gamma h_n \quad (3.36)$$

holds if we take

$$\bar{m} > \left(\frac{CC_\zeta}{\gamma} \right)^{\frac{1}{\alpha}}, \quad (3.37)$$

where $C_\zeta = 1/(1 - \beta^{1-\zeta}(1 + CH))$.

Proof. By Theorem 3.2, together with (3.1), (3.4), (3.26), (3.35) and $\beta^{1-\zeta}(1 + CH) < 1$, we have the following estimates

$$\begin{aligned} \|u^{h_n} - \tilde{u}_{h_n}\|_a &\leq C \sum_{k=2}^n (1 + C\eta_a(V_H))^{n-k} \frac{1}{m_k^\alpha} \delta_{h_k}(u) \\ &\leq C \sum_{k=2}^n (1 + CH)^{n-k} \frac{1}{\bar{m}^\alpha} \left(\frac{h_k}{h_n} \right)^{-\zeta} h_k \\ &\leq C \sum_{k=2}^n (1 + CH)^{n-k} \beta^{(n-k)(1-\zeta)} \frac{h_n}{\bar{m}^\alpha} \\ &= C \frac{h_n}{\bar{m}^\alpha} \sum_{k=0}^{n-2} (\beta^{1-\zeta}(1 + CH))^k \\ &\leq C \frac{h_n}{\bar{m}^\alpha} \frac{1}{1 - \beta^{1-\zeta}(1 + CH)} \leq \frac{CC_\zeta}{\bar{m}^\alpha} h_n. \end{aligned} \quad (3.38)$$

Then it is obvious that $\|u^{h_n} - \tilde{u}_{h_n}\|_a \leq \gamma h_n$ when \bar{m} satisfies the condition (3.37). \square

Corollary 3.2. *Under the same conditions of Theorem 3.3 and (3.37), if $Ch_n \leq \gamma$, we have the following estimate*

$$\|u^{h_n} - \bar{u}_{h_n}\|_a \leq 2\gamma h_n. \quad (3.39)$$

4. Parallel Cascadic Multigrid Method and Computational Work

In this section, we will propose how to solve the semilinear elliptic equation (2.3) by parallel computing technique.

For the boundary value problem included in the first step of Algorithm 3.1, there exists a standard procedure to execute parallel computing. We have to assemble the matrix and storage the data on different nodes. to carry out the smooth process by parallel computing.

It is different from the linear elliptic equation case that, in the second step of Algorithm 3.1, we have to solve a semilinear elliptic problem on the newly constructed coarse space $V_H^{h_k}$. Always, some type of nonlinear iteration method is used to solve this semilinear elliptic equation. In each nonlinear iteration step, the computational work of assembling the stiffness matrix on the finite element space $V_H^{h_k}$ ($k = 2, \dots, n$) is $\mathcal{O}(N_k)$. Fortunately, the matrix assembling can be carried out by the parallel way easily in the finite element space since there is no data transfer. And the solving time can be ignored due to the low dimension.

So based on the parallel scheme above, we can rewritten Algorithm 3.1 and 3.2 in the following way.

Algorithm 4.1. *Parallel type of One Correction Step*

1. Define the following auxiliary source problem: Find $\hat{u}^{h_{k+1}} \in V_{h_{k+1}}$ such that

$$a(\hat{u}^{h_{k+1}}, v_{h_{k+1}}) = (g, v_{h_{k+1}}) - (f(x, u^{h_k}), v_{h_{k+1}}), \quad \forall v_{h_{k+1}} \in V_{h_{k+1}}. \quad (4.1)$$

Perform the smoothing process (3.6) by parallel computing to obtain a new approximate solution $\tilde{u}^{h_{k+1}} \in V_{h_{k+1}}$ by

$$\tilde{u}^{h_{k+1}} = \text{Smooth}(V_{h_{k+1}}, g - f(x, u^{h_k}), u^{h_k}, m_{k+1}, S_{h_{k+1}}). \quad (4.2)$$

2. Define a new finite element space $V_H^{h_{k+1}} = V_H + \text{span}\{\tilde{u}^{h_{k+1}}\}$ and solve the following semilinear elliptic equation: Find $u^{h_{k+1}} \in V_H^{h_{k+1}}$ such that

$$a(u^{h_{k+1}}, v_H^{h_{k+1}}) + (f(x, u^{h_{k+1}}), v_H^{h_{k+1}}) = (g, v_H^{h_{k+1}}), \quad \forall v_H^{h_{k+1}} \in V_H^{h_{k+1}}. \quad (4.3)$$

Assemble the stiffness matrix and right hand term on the finite element space $V_H^{h_{k+1}}$ by parallel computing in every nonlinear iteration step. And then solve these low dimensional matrix equations.

Summarize the above two steps by defining

$$u^{h_{k+1}} = \text{ParaSmoothCorrection}(V_H, V_{h_{k+1}}, u^{h_k}, m_{k+1}, S_{h_{k+1}}).$$

Based on the parallel type one correction step, we can construct a parallel cascadic multigrid method for semilinear elliptic equation as follows:

Algorithm 4.2. *Parallel Cascadic Multigrid Method*

1. Solve the following semilinear elliptic equation by parallel technique in the initial finite element space V_{h_1} : Find $u^{h_1} \in V_{h_1}$ such that

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

2. For $k = 1, \dots, n-1$, do the following iteration

$$u^{h_{k+1}} = \text{ParaSmoothCorrection}(V_H, V_{h_{k+1}}, u^{h_k}, m_{k+1}, S_{h_{k+1}}).$$

End Do

Now we come to estimate the computational work for Algorithm 4.2.

Theorem 4.1. *Assume the semilinear elliptic equation solving in the coarse spaces V_H and V_{h_1} need work M_H and M_{h_1} , respectively. We use p computing-nodes in Algorithm 4.2, and let ϖ denote the nonlinear iteration times when we solve the semilinear elliptic problem (3.10). If $\zeta/\alpha < d$, the total computational work of Algorithm 4.2 can be bounded by $\mathcal{O}\left(\frac{1+\varpi}{p}N_n + M_{h_1} + M_H \log(N_n)\right)$ and furthermore $\mathcal{O}((1+\varpi)N_n/p)$ provided M_H and M_{h_1} is small enough. While if $\zeta/\alpha = d$, the two bounds change to $\mathcal{O}\left(\frac{1+\varpi}{p}N_n \log(N_n) + M_{h_1} + M_H \log(N_n)\right)$ and $\mathcal{O}((1+\varpi)N_n \log(N_n)/p)$, respectively.*

Proof. Let W denote the whole computational work of Algorithm 4.2, w_k denote the work on the k -th level for $k = 1, \dots, n$. From the definition of Algorithms 4.1 and 4.2, (3.1), (3.34) and (3.35), it follows that

$$\begin{aligned} W &= \sum_{k=1}^n w_k \leq M_{h_1} + \sum_{k=2}^n \frac{m_k N_k}{p} + \sum_{k=2}^n \frac{\varpi}{p} N_k + M_H \log_{\beta}(N_n) \\ &\leq M_{h_1} + CM_H \log(N_n) + \frac{\bar{m}\sigma^{1/\alpha}N_n}{p} \sum_{k=2}^n \left(\frac{1}{\beta}\right)^{(n-k)(d-\zeta/\alpha)} \\ &\quad + \frac{\varpi}{p} N_n \sum_{k=2}^n \left(\frac{1}{\beta}\right)^{d(n-k)} \\ &\leq M_{h_1} + CM_H \log(N_n) + C\frac{\varpi}{p}N_n + \frac{\bar{m}\sigma^{1/\alpha}N_n}{p} \sum_{k=2}^n \left(\frac{1}{\beta}\right)^{(n-k)(d-\zeta/\alpha)}. \end{aligned}$$

Thus, the computational work W can be bounded by $\mathcal{O}(M_{h_1} + M_H \log(N_n) + (1+\varpi)N_n/p)$ when $d - \zeta/\alpha > 0$ and by $\mathcal{O}(M_{h_1} + M_H \log(N_n) + (1+\varpi)N_n \log(N_n)/p)$ when $d - \zeta/\alpha = 0$ respectively, and moreover, by $\mathcal{O}((1+\varpi)N_n/p)$ and $\mathcal{O}((1+\varpi)N_n \log(N_n)/p)$ if M_H and M_{h_1} are small enough. \square

Remark 4.1. Since we have a good enough initial solution $\tilde{u}^{h_{k+1}}$ in the second step of Algorithm 3.2, solving the semilinear elliptic equation (3.10) always does not need many nonlinear iteration times.

If we choose the conjugate gradient method as the smoothing operator, then $\alpha = 1$ and the computational work of Algorithm 3.2 can be bounded by $\mathcal{O}((1+\varpi)N_n/p + M_{h_1} + M_H \log(N_n))$ for both $d = 2$ and $d = 3$ when we choose $1 < \zeta < d$.

When the symmetric Gauss-Seidel, the SSOR, the damped Jacobi or the Richardson iteration acts as the smoothing operator, we know $\alpha = 1/2$. Then the computational work of Algorithm 3.2 can be bounded by $\mathcal{O}((1+\varpi)N_n/p + M_{h_1} + M_H \log(N_n))$ only for $d = 3$ when we choose $1 < \zeta < 3/2$. In the case of $\alpha = 1/2$ and $d = 2$, from Theorem 3.3 and its proof, we can only choose $\zeta = 1$ and then the final error has the estimate $\|u^{h_n} - \bar{u}_{h_n}\|_a \leq Ch_n(1+CH)^{|\log(h_n)|}$ and the computational work can only be bounded by $\mathcal{O}((1+\varpi)N_n \log(N_n)/p + M_{h_1} + M_H \log(N_n))$.

5. Numerical Example

In this section, four numerical experiments are presented to verify the theoretical analysis and efficiency of Algorithm 4.2. We will check different nonlinear terms including polynomial, exponential function and the function that only has a bounded first order derivative. Here, we choose the conjugate-gradient iteration as the smoothing operator ($\alpha = 1$) with the number of iteration steps

$$m_k = \lceil \bar{m}\sigma\beta^{\zeta(n-k)} \rceil \quad \text{for } k = 2, \dots, n$$

with $\bar{m} = 2$, $\sigma = 2$, $\beta = 2$, $\zeta = 2.8$ and $\lceil r \rceil$ denoting the smallest integer which is not less than r .

5.1. Example 1

We consider the following semilinear elliptic problem:

$$\begin{cases} -\Delta u + u^3 = g, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega, \end{cases} \quad (5.1)$$

where $\Omega = (0, 1)^3$. We choose the right hand side term g such that the exact solution is given by

$$u = \sin(\pi x) \sin(\pi y) \sin(\pi z). \quad (5.2)$$

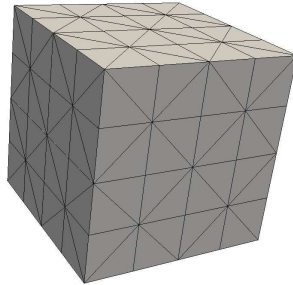


Fig. 5.1. The initial mesh for Example 1.

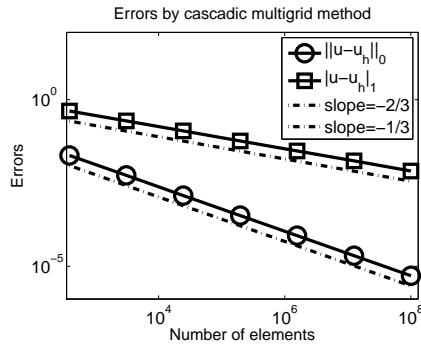


Fig. 5.2. Errors of Algorithm 4.2 for Example 1.

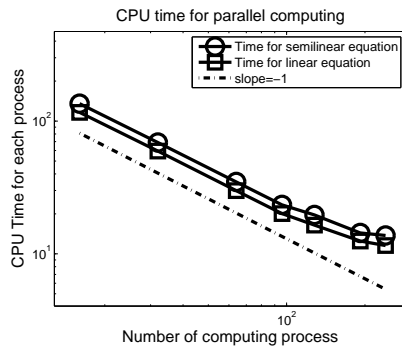


Fig. 5.3. CPU time (in seconds) of cascadic multigrid method in the parallel computing way for Example 1.

We give the numerical results for the approximate solutions obtained by Algorithm 4.2. Figure 5.1 shows the initial triangulation. Figure 5.2 shows the error estimates. It is shown in the Figure 5.2 that the approximate solution produced by Algorithm 4.2 has the optimal convergence order which coincides with the theoretical results in Corollary 3.2.

Here we present the parallel efficiency of Algorithm 4.2 for (5.1) and the corresponding boundary value problem in Figure 5.3. And the boundary value problem is solved by software package PHG. From these two tables we can find that the parallel cascadic multigrid method for semilinear elliptic problem can derive nearly the same parallel efficiency as parallel AFM for the boundary value problem.

5.2. Example 2

In the second example, we solve the following semilinear elliptic problem:

$$\begin{cases} -\Delta u - e^{-u} = 1, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega, \end{cases} \tag{5.3}$$

where $\Omega = (0,1)^3$. Since the exact solution is not known, we choose an adequate accurate approximate solution on a fine enough mesh as the exact one.

Algorithm 4.2 is applied to this example. Figure 5.1 shows the initial mesh. Figure 5.4 gives the corresponding numerical results which also show the optimal convergence rate of Algorithm 4.2. Figure 5.5 shows the parallel efficiency of Algorithm 4.2 for problem (5.3)

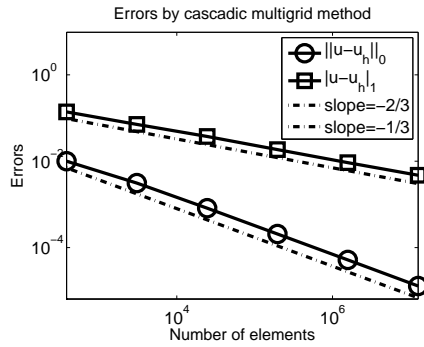


Fig. 5.4. Errors of Algorithm 4.2 for Example 2.

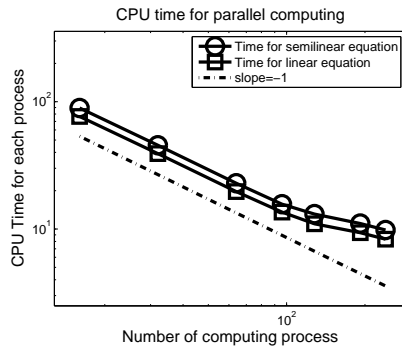


Fig. 5.5. CPU time (in seconds) of cascadic multigrid method in the parallel computing way for Example 2.

and the corresponding boundary value problem, which indicate the good scalability of parallel cascadic multigrid method.

5.3. Example 3

In the third example, we solve the following semilinear elliptic problem:

$$\begin{cases} -\Delta u + f(x, u) = g, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega, \end{cases} \quad (5.4)$$

with

$$f(x, u) = \begin{cases} u^{3/2}, & \text{if } u \geq 0, \\ -u^{3/2}, & \text{if } u < 0, \end{cases} \quad (5.5)$$

where $\Omega = (0, 1)^3$. We choose the right hand side term g such that the exact solution is given by

$$u = \sin(100\pi x) \sin(100\pi y) \sin(100\pi z). \quad (5.6)$$

In this example, the nonlinear term $f(x, v)$ has the bounded first order derivative $\partial f(x, v)/\partial v$ but unbounded second order derivative $\partial^2 f(x, v)/\partial^2 v$. Then the methods given in [15, 19] can not be used for this example.

Algorithm 4.2 is applied to this example. Figure 5.1 shows the initial mesh. Figure 5.6 gives the corresponding numerical results which also show the optimal convergence rate of Algorithm

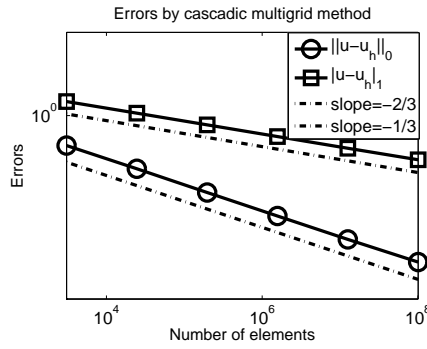


Fig. 5.6. Errors of Algorithm 4.2 for Example 3.

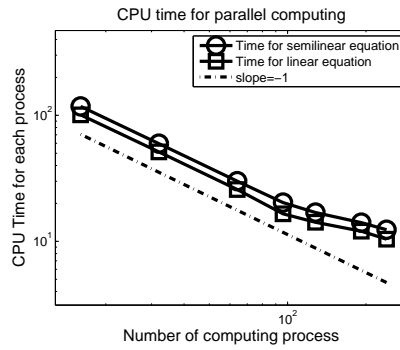


Fig. 5.7. CPU time (in seconds) of cascadic multigrid method in the parallel computing way for Example 3.

4.2. And Figure 5.7 shows the parallel efficiency of Algorithm 4.2 for problem (5.4) and the corresponding boundary value problem, which indicate the good scalability of Algorithm 4.2.

5.4. Example 4

In order to show the advantage of Algorithm 4.2 more clearly, we apply its serial version as well as the Full Approximation Scheme (FAS) [8], standard Newton multigrid method [25] and a multilevel successive iteration method [15] to Example 1-3 and compare their computational time.

In the test, we set the same number of degrees of freedom (2146689) for all examples. The computational time (in seconds) of Example 1 and 2 are provided in Table 5.1. From the result, we can find that they have very similar efficiency. But Algorithm 4.2 and multilevel successive iteration method work slightly better since we do not need to solve the linearized boundary value problem directly for these two schemes.

For the Example 3, the other methods fail to work due to the big truncation errors caused by the unbounded second order derivatives of the nonlinear term, but our method works, which shows the advantage of Algorithm 4.2.

Table 5.1: The computational time for Example 1 and 2.

	Algorithm 4.2	Newton MG	FAS	Multilevel successive iteration
Example 1	157.7136	168.9674	163.4874	154.5857
Example 2	110.1138	117.8412	114.3251	108.6365

6. Concluding Remarks

In this paper, we present a type of parallel cascadic multigrid method for semilinear elliptic equation based on the combination of the cascadic multigrid, multilevel correction scheme and parallel computing. The optimal convergence rate and optimal computational work have been demonstrated by theoretical analysis and numerical examples. Compared with the existing multigrid methods which require bounded second order derivatives of the nonlinear terms, the proposed method only requires a bounded first order derivative of the nonlinear term.

Acknowledgments. This work is supported in part by the National Science Foundation of China (NSFC 11401129).

References

- [1] R.A. Adams, Sobolev Spaces, Academic Press, New York, 1975.
- [2] F.A. Bornemann and P. Deuffhard, The cascadic multigrid method for elliptic problems, *Numer. Math.*, **75** (1996), 135–152.
- [3] R.E. Bank and T. Dupont, An optimal order process for solving finite element equations, *Math. Comp.*, **36** (1981), 35–51.
- [4] J.H. Bramble, Multigrid Methods, Pitman Research Notes in Mathematics, Vol. 294, John Wiley and Sons, 1993.
- [5] J.H. Bramble and J.E. Pasciak, New convergence estimates for multigrid algorithms, *Math. Comp.*, **49** (1987), 311–329.
- [6] J.H. Bramble and X. Zhang, The analysis of Multigrid Methods, Handbook of Numerical Analysis, 2000, 173–415.
- [7] A. Brandt, J. Brannick, K. Kahl, et al, Bootstrap AMG, *SIAM J. Sci. Comput.*, **33**(2) (2011), 612–632.
- [8] A. Brandt and O. Livne Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics, Revised Edition, Tech. Report 85, GMD-Studie, Sankt Augustin, Germany, 1984.
- [9] A. Brandt, S. McCormick and J. Ruge, Multigrid methods for differential eigenproblems, *SIAM J. Sci. Stat. Comput.*, **4**(2) (1983), 244–260.
- [10] S. Brenner, L. Scott, The Mathematical Theory of Finite Element Methods, New York, Springer-Verlag, 1994.
- [11] M. Brezina, C. Ketelsen, T. Manteuffel, et al, Relaxation-corrected bootstrap algebraic multigrid, *Numer. Linear Algebra Appl.*, **19**(2) (2012), 178–193.
- [12] P.G. Ciarlet, The finite Element Method for Elliptic Problem, North-holland Amsterdam, 1978.
- [13] W. Hackbusch, Multi-grid Methods and Applications, Springer-Verlag, Berlin, 1985.
- [14] X. Han, H. Xie and F. Xu, A cascadic multigrid method for eigenvalue problem, *J. Comput. Math.*, **35**(1) (2017), 74–90
- [15] Y. Huang, Z. Shi, T. Tang and W. Xue, A multilevel successive iteration method for nonlinear elliptic problem, *Math. Comp.*, **73** (2004), 525–539.
- [16] R. Kužel and P. Vaněk, Exact interpolation scheme with approximation vector used as a column of the prolongator, *Numer. Linear Algebra Appl.*, **22** (2015), 950–964.
- [17] Q. Lin and H. Xie, A multi-level correction scheme for eigenvalue problems, *Math. Comp.*, **84** (2015), 71–88.
- [18] Q. Lin, H. Xie, and J. Xu, Lower bound of the discretization error for piecewise polynomials, *Math. Comput.*, **83** (285) (2014), 1–13.
- [19] V. Shaidurov, Multigrid Methods For Finite Elements, Springer, 1995.
- [20] V. Shaidurov, Some estimates of the rate of convergence for the cascadic conjugate-gradient method, *Comput. Math. Appl.*, **31** (1996), 161–171.

- [21] V. Shaidurov and L. Tobiska, The convergence of the cascadic conjugate-gradient method applied to elliptic problems in domains with re-entrant corners, *Math. Comput.*, **69** (2000), 501–520.
- [22] L. Wang and X. Xu, The Basic Mathematical Theory of Finite Element Methods. Science Press(in Chinese), Beijing, 2004.
- [23] H. Xie, A multigrid method for eigenvalue problem. *J. Comput. Phys.*, **274** (2014), 550–561.
- [24] J. Xu, Iterative methods by space decomposition and subspace correction, *SIAM Review*, **34**(4) (1992), 581–613.
- [25] J. Xu, Two-Grid Discretization Techniques for Linear and Nonlinear PDEs, *SIAM J. Numer. Anal.*, **33**(5) (1996), 1759–1777.
- [26] J. Xu, A Novel Two-Grid Method for Semilinear Elliptic Equations, *SIAM J. Sci. Comput.*, **15**(1) (1994), 231–237.