INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 16, Number 5, Pages 789–803

© 2019 Institute for Scientific
Computing and Information

# AN EFFICIENT MULTIGRID METHOD FOR GROUND STATE SOLUTION OF BOSE-EINSTEIN CONDENSATES

NING ZHANG, FEI XU, AND HEHU XIE

**Abstract.** An efficient multigrid method is proposed to compute the ground state solution of Bose-Einstein condensations by the finite element method based on the combination of the multi-grid method for nonlinear eigenvalue problem and an efficient implementation for the nonlinear iteration. The proposed numerical method not only has the optimal convergence rate, but also has the asymptotically optimal computational efficiency which is independent from the nonlinearity of the problem. The independence from the nonlinearity means that the asymptotic estimate of the computational work can reach almost the same as that of solving the corresponding linear boundary value problem by the multigrid method. Some numerical experiments are provided to validate the efficiency of the proposed method.

**Key words.** BEC, GPE, nonlinear eigenvalue problem, multigrid, tensor, finite element method, asymptotically optimal efficiency.

## 1. Introduction

It is well known that Bose-Einstein condensation (BEC), which is a gas of bosons that are in the same quantum state, is an important and active field [2, 3, 4, 12, 19] in physics. The properties of the condensate at zero or very low temperature [13, 21] can be described by the well-known Gross-Pitaevskii equation (GPE) [15] which is a time-independent nonlinear Schrödinger equation [20].

Since this paper considers the numerical method for the nonlinear eigenvalue problem, we are concerned with the following non-dimensionalized GPE problem: Find $\lambda \in \mathbb{R}$ and a function $u$ such that

$$\left\{\begin{array}{rcll} -\Delta u + Wu + \zeta|u|^2u &=& \lambda u, & \text{in } \Omega,\\ u &=& 0, & \text{on } \partial\Omega,\\ \int_\Omega |u|^2 d\Omega &=& 1, & \end{array}\right. \tag{1}$$

where $\Omega \subset \mathbb{R}^d$ $(d = 1, 2, 3)$ denotes the computing domain which has the cone property [1], $\zeta$ is some positive constant and $W(x) = \gamma_1 x_1^2 + \ldots + \gamma_d x_d^2 \geq 0$ with $\gamma_1, \ldots, \gamma_d > 0$ [5, 29]. It is well known that the ground state solution for (1) is unique.

The convergence of the finite element method for GPEs is first proved in [29] and [8] gives prior error estimates which will be used in the analysis of our method. There also exist two-grid finite element methods for GPE in [9, 10, 17]. Recently, a type of multigrid method for eigenvalue problems has been proposed in [22, 24, 25, 26, 27]. Especially, [27] gives a multigrid method for GPE (1) and the corresponding error estimates. This type of multigrid method is designed based on the multilevel correction method in [22], and a sequence of nested finite element spaces with different levels of accuracy which can be built in the same way as the multilevel method for boundary value problems [28]. The corresponding error estimates have already been obtained in [27]. Furthermore, the estimate of computational work has also been given in [27]. The computational work of the multigrid in [27] is linear

Received by the editors May 8, 2018 and, in revised form, November 23, 2018.
2000 *Mathematics Subject Classification.* 65N30, 65N25, 65L15, 65B99.

scale but depends on the nonlinearity (i.e. the value of $\zeta$) in some sense. The aim of this paper is to improve the efficiency further with a special implementing method for the multigrid iteration by using the tensor tool [14] for the GPE. With the tensor tool, the nonlinear iteration can be implemented only in the coarsest mesh and needs very small computational work. By using the proposed implementing technique, the multigrid method can really arrive the asymptotically optimal computational complexity which is almost independent of the nonlinearity of the GPE.

An outline of the paper goes as follows. In Section 2, we introduce finite element method for the ground state solution of BEC, i.e. non-dimensionalized GPE (1). A type of one correction step is given in Sections 3. In Section 4, we propose an efficient implementing technique for the nonlinear eigenvalue problem included in the one correction step. A type of multigrid algorithm for solving the non-dimensionalized GPE by the finite element method will be stated in Section 5. Three numerical examples are provided in Section 6 to validate the efficiency of the proposed numerical method in this paper. Some concluding remarks are given in the last section.

## 2. Finite element method for GPE problem

This section is devoted to introducing some notation and finite element method for the GPE (1). The letter $C$ (with or without subscripts) denotes a generic positive constant which may be different at its different occurrences. For convenience, the symbols $\lesssim$, $\gtrsim$ and $\approx$ will be used in this paper. That $x_1 \lesssim y_1, x_2 \gtrsim y_2$ and $x_3 \approx y_3$, mean that $x_1 \leq C_1 y_1$, $x_2 \geq c_2 y_2$ and $c_3 x_3 \leq y_3 \leq C_3 x_3$ for some constants $C_1, c_2, c_3$ and $C_3$ that are independent of mesh sizes (see, e.g., [28]). The standard notation for the Sobolev spaces $W^{s,p}(\Omega)$ and their associated norms $\|\cdot\|_{s,p,\Omega}$ and seminorms $|\cdot|_{s,p,\Omega}$ (see, e.g., [1]) will be used. 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 and $\|\cdot\|_{s,\Omega} = \|\cdot\|_{s,2,\Omega}$. In this paper, we set $V = H_0^1(\Omega)$ and use $\|\cdot\|_s$ to denote $\|\cdot\|_{s,\Omega}$ for simplicity.

For the aim of finite element discretization, we define the corresponding weak form for (1) as follows: Find $(\lambda, u) \in \mathbb{R} \times V$ such that $b(u,u) = 1$ and

$$a(u,v) = \lambda b(u,v), \quad \forall v \in V, \tag{2}$$

where

$$a(u,v) := \int_\Omega \big(\nabla u \nabla v + W uv + \zeta |u|^2 uv\big) d\Omega, \qquad b(u,v) := \int_\Omega uv d\Omega.$$

Now, let us define the finite element method [7, 11] for the problem (2). First we generate a shape-regular decomposition of the computing domain $\Omega \subset \mathbb{R}^d$ $(d = 2,3)$ into triangles or rectangles for $d=2$ (tetrahedrons or hexahedrons for $d=3$). The diameter of a cell $K \in \mathcal{T}_h$ is denoted by $h_K$ and define $h$ as $h := \max_{K\in\mathcal{T}_h} h_K$. Then the corresponding linear finite element space $V_h \subset V$ can be built on the mesh $\mathcal{T}_h$. We assume that $V_h \subset V$ is a family of finite-dimensional spaces that satisfy the following assumption:

$$\lim_{h\to 0} \inf_{v_h \in V_h} \|w - v_h\|_1 = 0, \quad \forall w \in V. \tag{3}$$

The standard finite element method for (2) is to solve the following eigenvalue problem: Find $(\bar{\lambda}_h, \bar{u}_h) \in \mathbb{R} \times V_h$ such that $b(\bar{u}_h, \bar{u}_h) = 1$ and

$$a(\bar{u}_h, v_h) = \bar{\lambda}_h b(\bar{u}_h, v_h), \quad \forall v_h \in V_h. \tag{4}$$

Then we define

$$\delta_h(u) := \inf_{v_h \in V_h} \|u - v_h\|_1. \tag{5}$$

For understanding the multigrid method in this paper, we state the error estimates of the finite element method for GPE (1).

**Lemma 2.1.** *(*[8, Theorem 1],[29]*) There exists $h_0 > 0$, such that for all $0 < h < h_0$, the smallest eigenpair approximation $(\bar{\lambda}_h, \bar{u}_h)$ of (4) has following error estimates:*

$$\|u - \bar{u}_h\|_1 \lesssim \delta_h(u), \tag{6}$$

$$\|u - \bar{u}_h\|_0 \lesssim \eta_a(V_h)\|u - \bar{u}_h\|_1 \lesssim \eta_a(V_h)\delta_h(u), \tag{7}$$

$$|\lambda - \bar{\lambda}_h| \lesssim \|u - \bar{u}_h\|_1^2 + \|u - \bar{u}_h\|_0 \lesssim \eta_a(V_h)\delta_h(u), \tag{8}$$

*where $\eta_a(V_h)$ is defined as follows:*

$$\eta_a(V_h) = \|u - \bar{u}_h\|_1 + \sup_{f \in L^2(\Omega), \|f\|_0 = 1} \inf_{v_h \in V_h} \|Tf - v_h\|_1 \tag{9}$$

*with the operator $T$ being defined as follows: Find $Tf \in u^\perp$ such that*

$$a(Tf, v) + 2(\zeta|u|^2(Tf), v) - (\lambda(Tf), v) = (f, v), \quad \forall v \in u^\perp,$$

*where $u^\perp = \left\{v \in H_0^1(\Omega) | \int_\Omega uvd\Omega = 0\right\}$.*

## 3. One correction step

In this section, we recall the one correction step from [27] to improve the accuracy of the given eigenpair approximation. This correction step contains solving an auxiliary linear boundary value problem with multigrid method in the finer finite element space and a GPE on a very low dimensional finite element space which will be discussed in the next section.

In order to define the one correction step, we introduce a very coarse mesh $\mathcal{T}_H$ and the low dimensional linear finite element space $V_H$ defined on the mesh $\mathcal{T}_H$. Assume we have obtained an eigenpair approximation $(\lambda_{h_k}, u_{h_k}) \in \mathbb{R} \times V_{h_k}$ and the coarse space $V_H$ is a subset of $V_{h_k}$. Let $V_{h_{k+1}} \subset V$ be a finer finite element space such that $V_{h_k} \subset V_{h_{k+1}}$. Based on this finer finite element space, we define the following one correction step.

**Algorithm 3.1.** *One Correction Step*

(1) *Define the following auxiliary boundary value problem: Find $\widehat{u}_{h_{k+1}} \in V_{h_{k+1}}$ such that*

$$\begin{aligned}(\nabla\widehat{u}_{h_{k+1}}, \nabla v_{h_{k+1}}) + (W\widehat{u}_{h_{k+1}}, v_{h_{k+1}}) + (\zeta|u_{h_k}|^2\widehat{u}_{h_{k+1}}, v_{h_{k+1}})&\\ = \lambda_{h_k} b(u_{h_k}, v_{h_{k+1}}), \quad \forall v_{h_{k+1}} \in V_{h_{k+1}}.&\end{aligned} \tag{10}$$

*Solve this equation with multigrid method* [6, 7, 16, 23, 28] *to obtain an approximation $\widetilde{u}_{h_{k+1}} \in V_{h_{k+1}}$ with the error estimate $\|\widetilde{u}_{h_{k+1}} - \widehat{u}_{h_{k+1}}\|_1 \lesssim \varsigma_{h_{k+1}}$. Here $\varsigma_{h_{k+1}}$ is used to denote the accuracy for the multigrid iteration.*

(2) *Define a new finite element space $V_{H,h_{k+1}} = V_H + \text{span}\{\widetilde{u}_{h_{k+1}}\}$ and solve the following nonlinear eigenvalue problem: Find $(\lambda_{h_{k+1}}, u_{h_{k+1}}) \in \mathbb{R} \times V_{H,h_{k+1}}$ such that $b(u_{h_{k+1}}, u_{h_{k+1}}) = 1$ and*

$$a(u_{h_{k+1}}, v_{H,h_{k+1}}) = \lambda_{h_{k+1}} b(u_{h_{k+1}}, v_{H,h_{k+1}}), \quad \forall v_{H,h_{k+1}} \in V_{H,h_{k+1}}. \tag{11}$$

*Summarize above two steps into*

$$(\lambda_{h_{k+1}}, u_{h_{k+1}}) = \texttt{Correction}(V_H, \lambda_{h_k}, u_{h_k}, V_{h_{k+1}}, \varsigma_{h_{k+1}}).$$

Similarly, we also state the following error estimates from [27] for the one correction step defined in Algorithm 3.1.

**Theorem 3.1.** *([27, Theorem 3.1]) Assume $h_k < h_0$ (as in Lemma 2.1). Then after one correction step, the resultant approximation $(\lambda_{h_{k+1}}, u_{h_{k+1}}) \in \mathbb{R} \times V_{h_{k+1}}$ has the following error estimates:*

$$\|\bar{u}_{h_{k+1}} - u_{h_{k+1}}\|_1 \lesssim \varepsilon_{h_{k+1}}(u), \tag{12}$$

$$\|\bar{u}_{h_{k+1}} - u_{h_{k+1}}\|_0 \lesssim \eta_a(V_H)\|u - u_{h_{k+1}}\|_1, \tag{13}$$

$$|\bar{\lambda}_{h_{k+1}} - \lambda_{h_{k+1}}| \lesssim \eta_a(V_H)\varepsilon_{h_{k+1}}(u), \tag{14}$$

*where $\varepsilon_{h_{k+1}}(u) := \eta_a(V_{h_k})\delta_{h_k}(u) + \|\bar{u}_{h_k} - u_{h_k}\|_0 + |\bar{\lambda}_{h_k} - \lambda_{h_k}| + \varsigma_{h_{k+1}}$.*

## 4. Efficient implementation

In this section, we show an efficient implementing method for Step 2 of Algorithm 3.1, i.e., solving the nonlinear eigenvalue problem (11). For simplicity of notation, we use $h$ to denote $h_{k+1}$. Then $V_h$, $\widetilde{u}_h$ and $V_{H,h} = V_H + \text{span}\{\widetilde{u}_h\}$ denote $V_{h_{k+1}}$, $\widetilde{u}_{h_{k+1}}$ and $V_{H,h_{k+1}} = V_H + \text{span}\{\widetilde{u}_{h_{k+1}}\}$, respectively, in this section. Here we also define $N_H := \text{dim}V_H$ and $N_h := \text{dim}V_h$. Let $\{\phi_{k,H}\}_{1\leq k\leq N_H}$ denotes the Lagrange basis function for the coarse finite element space $V_H$.

For simplicity, the fixed point (self-consistent field) iteration method with dumping technique is adopted to solve the nonlinear eigenvalue problem (11). In each nonlinear iteration, the main content is to assemble the matrices for problem (11) which is defined on the special space $V_{H,h}$. The function in $V_{H,h}$ can be denoted by $u_{H,h} = u_H + \alpha\widetilde{u}_h$. Solving problem (11) is to obtain the function $u_H \in V_h$ and the value $\alpha \in \mathbb{R}$. Let $u_H = \sum_{k=1}^{N_H} u_k\phi_{k,H}$ and define the vector $\mathbf{u}_H$ as $\mathbf{u}_H = [u_1, \cdots, u_{N_H}]^T$.

Based on the structure of the space $V_{H,h}$, the matrix version of the eigenvalue problem (11) can be written as follows

$$\begin{pmatrix} A_H & b_{Hh} \\ b_{Hh}^T & \xi \end{pmatrix}\begin{pmatrix} \mathbf{u}_H \\ \alpha \end{pmatrix} = \lambda_h \begin{pmatrix} M_H & c_{Hh} \\ c_{Hh}^T & \gamma \end{pmatrix}\begin{pmatrix} \mathbf{u}_H \\ \alpha \end{pmatrix}, \tag{15}$$

where $\mathbf{u}_H \in \mathbb{R}^{N_H}$ and $\alpha \in \mathbb{R}$.

It is obvious that the matrix $M_H$, the vector $c_{Hh}$ and the scalar $\gamma$ will not change during the nonlinear iteration process as long as we have obtained the function $\widetilde{u}_h$. But the matrix $A_H$, the vector $b_{Hh}$ and the scalar $\xi$ will change during the nonlinear iteration process. It is required to consider the efficient implementation to update the the matrix $A_H$, the vector $b_{Hh}$ and the scalar $\xi$ since there is a function $\widetilde{u}_h$ which is defined on the fine mesh $\mathcal{T}_h$. The aim of this section is to propose an efficient method to update the matrix $A_H$, the vector $b_{Hh}$ and the scalar $\xi$ without computation on the fine mesh $\mathcal{T}_h$ during the nonlinear iteration process. Assume we have a given initial value $(u_H, \alpha) \in V_H \times \mathbb{R}$. Now, in order to carry out the nonlinear iteration for eigenvalue problem (15), we come to consider the computation for the matrix $A_H$, vector $b_{Hh}$ and the scalar $\xi$.

From the definitions of the space $V_{H,h}$ and the eigenvalue problem (11), the matrix $A_H$ has the following expansion

$$\begin{aligned}(A_H)_{i,j} &= \int_\Omega \nabla\phi_{i,H}\nabla\phi_{j,H}d\Omega + \int_\Omega w\phi_{i,H}\phi_{j,H}d\Omega + \int_\Omega \zeta(u_H + \alpha\widetilde{u}_h)^2\phi_{i,H}\phi_{j,H}d\Omega \\ &:= (A_{H,1})_{i,j} + (A_{H,2})_{i,j},\end{aligned} \tag{16}$$

where

$$(A_{H,1})_{i,j} = \int_\Omega \nabla\phi_{i,H}\nabla\phi_{j,H}d\Omega + \int_\Omega w\phi_{i,H}\phi_{j,H}d\Omega \tag{17}$$

and

$$\begin{aligned}(A_{H,2})_{i,j} &= \int_\Omega \zeta(u_H+\alpha\widetilde{u}_h)^2\phi_{i,H}\phi_{j,H}d\Omega\\ &= \int_\Omega \zeta\big((u_H)^2+2\alpha u_H\widetilde{u}_h+\alpha^2(\widetilde{u}_h)^2\big)\phi_{i,H}\phi_{j,H}d\Omega\\ &= \int_\Omega \zeta(u_H)^2\phi_{i,H}\phi_{j,H}d\Omega + 2\alpha\int_\Omega \zeta\widetilde{u}_h u_H\phi_{i,H}\phi_{j,H}d\Omega\\ &\qquad +\alpha^2\int_\Omega \zeta(\widetilde{u}_h)^2\phi_{i,H}\phi_{j,H}d\Omega\\ &:= (A_{H,2,1})_{i,j}+2\alpha(A_{H,2,2})_{i,j}+\alpha^2(A_{H,2,3})_{i,j}. \end{aligned}\tag{18}$$

It is obvious that the computational work for the matrix

$$(A_{H,2,1})_{i,j} = \int_\Omega \zeta(u_H)^2\phi_{i,H}\phi_{j,H}d\Omega \tag{19}$$

is $\mathcal{O}(N_H)$. The matrices $A_{H,1}$, and $A_{H,2,3}$ which is defined by

$$(A_{H,2,3})_{i,j} = \int_\Omega \zeta(\widetilde{u}_h)^2\phi_{i,H}\phi_{j,H}d\Omega \tag{20}$$

will not change during the nonlinear iteration process.

The matrix $A_{H,2,2}$ has the following expansion

$$(A_{H,2,2})_{i,j} = \sum_{k=1}^{N_H} u_k\int_\Omega \zeta\widetilde{u}_h\phi_{k,H}\phi_{i,H}\phi_{j,H}d\Omega. \tag{21}$$

The expansion (21) gives a hint to define a tensor $T_H$ as follows

$$(T_H)_{i,j,k} = \int_\Omega \zeta\widetilde{u}_h\phi_{k,H}\phi_{i,H}\phi_{j,H}d\Omega. \tag{22}$$

Then the matrix $A_{H,2,2}$ has the following computational scheme

$$A_{H,2,2} = T_H\cdot\mathbf{u}_H, \tag{23}$$

where $T_H\cdot\mathbf{u}_H$ denotes the multiplication of the tensor $T_H$ and the vector $\mathbf{u}_H$ corresponding to the last index $k$. From (22), it is easy to know that the dimension of the tensor $T_H$ is $\mathbb{R}^{N_H\times N_H\times N_H}$ and the number of nonzero elements is $\mathcal{O}(N_H)$. Thus $T_H$ is a sparse tensor and the computational work for the operation (23) is $\mathcal{O}(N_H)$.

Now, let us consider the computation for the vector $b_{Hh}$. From the definition of the space $V_{H,h}$ and the problem (11), the vector $b_{Hh}$ has the following expansion

$$\begin{aligned}(b_{Hh})_i &= \int_\Omega \nabla\widetilde{u}_h\nabla\phi_{i,H}d\Omega + \int_\Omega w\widetilde{u}_h\phi_{i,H}d\Omega + \int_\Omega \zeta(u_H+\alpha\widetilde{u}_h)^2\widetilde{u}_h\phi_{i,H}d\Omega\\ &:= (b_{Hh,1})_i+(b_{Hh,2})_i,\end{aligned}\tag{24}$$

where

$$(b_{Hh,1})_i = \int_\Omega \nabla\widetilde{u}_h\nabla\phi_{i,H}d\Omega + \int_\Omega w\widetilde{u}_h\phi_{i,H}d\Omega, \tag{25}$$

and

$$\begin{aligned}
(b_{Hh,2})_i &= \int_\Omega \zeta(u_H+\alpha\widetilde{u}_h)^2\widetilde{u}_h\phi_{i,H}d\Omega\\
&= \int_\Omega \zeta\big((u_H)^2+2\alpha\widetilde{u}_h u_H+\alpha^2(\widetilde{u}_h)^2\big)\widetilde{u}_h\phi_{i,H}d\Omega\\
&= \int_\Omega \zeta(u_H)^2\widetilde{u}_h\phi_{i,H}d\Omega+2\alpha\int_\Omega\zeta(\widetilde{u}_h)^2u_H\phi_{i,H}d\Omega+\alpha^2\int_\Omega\zeta(\widetilde{u}_h)^3\phi_{i,H}d\Omega\\
&:= (b_{Hh,2,1})_i+2\alpha(b_{Hh,2,2})_i+\alpha^2(b_{Hh,2,3})_i. \qquad (26)
\end{aligned}$$

It is obvious that the vector $b_{Hh,1}$ will not change during the nonlinear iteration process. Thus, we only need to consider the computation for the vector $b_{Hh,2}$.

First, the computation for the vector $b_{Hh,2,1}$ can be implemented as follows

$$\begin{aligned}
(b_{Hh,2,1})_i &= \int_\Omega\Big(\sum_{j=1}^{N_H}u_j\phi_{j,H}\Big)^2\widetilde{u}_h\phi_{i,H}d\Omega\\
&= \sum_{j=1}^{N_H}\sum_{k=1}^{N_H}u_ju_k\int_\Omega\widetilde{u}_h\phi_{j,H}\phi_{k,H}\phi_{i,H}d\Omega. \qquad (27)
\end{aligned}$$

Based on the tensor $T_H$, the vector $b_{Hh,2,1}$ can be calculated by the tensor multiplication

$$b_{Hh,2,1}=(T_H\cdot\mathbf{u}_H)\cdot\mathbf{u}_H=A_{H,2,2}\mathbf{u}_H, \qquad (28)$$

where $(T_H\cdot\mathbf{u}_H)\cdot\mathbf{u}_H$ denotes the multiplication of the tensor $T_H$ with the vector $\mathbf{u}_H$ corresponding to the last two indices $k$ and $j$. Similarly, the computational work for the operation (28) is also $\mathcal{O}(N_H)$.

Then the computation for $b_{Hh,2,2}$ can be done as follows

$$(b_{Hh,2,2})_i=\sum_{j=1}^{N_H}u_j\int_\Omega\zeta(\widetilde{u}_h)^2\phi_{j,H}\phi_{i,H}d\Omega=(A_{H,2,3}\mathbf{u}_H)_i. \qquad (29)$$

Finally, the vector $b_{Hh,2,3}$ which is defined as

$$(b_{Hh,2,3})_i=\int_\Omega\zeta(\widetilde{u}_h)^3\phi_{i,H}d\Omega, \qquad (30)$$

will not change neither during the nonlinear iteration process.

Now, let us come to consider the computation for the value $\xi$. It is obvious that $\xi$ has the following expansion

$$\begin{aligned}
\xi &= \int_\Omega|\nabla\widetilde{u}_h|^2d\Omega+\int_\Omega w(\widetilde{u}_h)^2d\Omega+\int_\Omega\zeta(u_H+\alpha\widetilde{u}_h)^2(\widetilde{u}_h)^2d\Omega\\
&= \int_\Omega\big(|\nabla\widetilde{u}_h|^2+w(\widetilde{u}_h)^2\big)d\Omega+\int_\Omega\zeta\big((u_H)^2+2\alpha u_H\widetilde{u}_h+\alpha^2(\widetilde{u}_h)^2\big)(\widetilde{u}_h)^2d\Omega\\
&:= d_1+d_2, \qquad (31)
\end{aligned}$$

where

$$d_1=\int_\Omega\big(|\nabla\widetilde{u}_h|^2+w(\widetilde{u}_h)^2\big)d\Omega, \qquad (32)$$

and

$$\begin{aligned} d_2 &= \sum_{i=1}^{N_H}\sum_{j=1}^{N_H} u_i u_j \int_\Omega \zeta(\widetilde{u}_h)^2 \phi_{i,H}\phi_{j,H} d\Omega + 2\alpha \sum_{i=1}^{N_H} u_i \int_\Omega \zeta(\widetilde{u}_h)^3 \phi_{i,H} d\Omega \\ &\quad + \alpha^2 \int_\Omega \zeta(\widetilde{u}_h)^4 d\Omega \\ &= \mathbf{u}_H^T A_{H,2,3}\mathbf{u}_H + 2\alpha \mathbf{u}_H^T b_{Hh,2,3} + \alpha^2 \xi_h, \end{aligned} \tag{33}$$

with the scalar $\xi_h$ being defined as follows

$$\xi_h = \int_\Omega \zeta(\widetilde{u}_h)^4 d\Omega. \tag{34}$$

Based on above discussion and preparation, we define the following algorithm for solving the nonlinear eigenvalue problem (11) in Step 2 of Algorithm 3.1.

**Algorithm 4.1.** *Nonlinear iteration method for eigenvalue problem (11)*

(1) *Preparation for the nonlinear iteration: Compute the tensor $T_H$ as in (22), the matrices $A_{H,1}$ and $A_{H,2,3}$ as in (17) and (20), vectors $b_{Hh,1}$ and $b_{Hh,2,3}$ as in (25) and (30), scalars $d_1$ and $\xi_h$ as in (32) and (34).*

(2) *Nonlinear iteration:*
   (a) *Produce the matrix $A_{H,2,1}$ and $A_{H,2,2}$ as in (19) and (23). Then compute the matrix $A_H = A_{H,1} + A_{H,2,1} + 2\alpha A_{H,2,2} + \alpha^2 A_{H,2,3}$.*
   (b) *Produce $b_{Hh,2,1}$ and $b_{Hh,2,2}$ as in (28) and (29). Then compute the vector $b_{Hh} = b_{Hh,1} + b_{Hh,2,1} + 2\alpha b_{Hh,2,2} + \alpha^2 b_{Hh,2,3}$.*
   (c) *Compute the scalar $d_2$ as in (33). Then compute the scalar $\xi = d_1 + d_2$.*
   (d) *Solve the eigenvalue problem (15) by some eigensolver to get a new eigenfunction $(u_H, \alpha)$ and the corresponding eigenvalue $\lambda_h$.*
   (e) *If the accuracy for nonlinear iteration is satisfied, stop the nonlinear iteration. Otherwise, continue the nonlinear iteration.*

(3) *Output the eigenfunction $u_h = u_H + \alpha\widetilde{u}_h = \sum_{i=1}^{N_H} u_i\phi_{i,H} + \alpha\widetilde{u}_h$ and the eigenvalue $\lambda_h$.*

**Remark 4.1.** *It is obvious that assembling the Tensor, matrices, vectors and scalar in Step 1 of Algorithm 4.1 needs computational work $\mathcal{O}(N_h)$. But, the computational work for each nonlinear iteration step (Step 2) of Algorithm 4.1 is only $\mathcal{O}(M_H)$, where $M_H$ denotes the computational work for solving the eigenvalue problem (15) and it holds that $M_H \geq N_H$. Assume there needs $\varpi$ nonlinear iteration times. Then the computational work for Algorithm 4.1 is only $\mathcal{O}(N_h + \varpi M_H)$.*

## 5. Multigrid method for GPE

Based on the preparation in previous sections, we introduce a type of multigrid method based on the *One Correction Step* defined in Algorithms 3.1 and the implementing technique defined in Algorithm 4.1. This type of multigrid method can obtain the same optimal error estimate as that for solving the GPE directly on the finest finite element space.

In order to develop multigrid scheme, we define a sequence of triangulations $\mathcal{T}_{h_k}$ of $\Omega$ as follows. Suppose $\mathcal{T}_{h_1}$ is produced from $\mathcal{T}_H$ by some regular refinements and let $\mathcal{T}_{h_k}$ be obtained from $\mathcal{T}_{h_{k-1}}$ via a regular refinement such that

$$h_k \approx \frac{1}{\beta} h_{k-1}, \qquad k = 2, \ldots, n, \tag{35}$$

where $\beta$ denotes the refinement index. Based on this sequence of meshes, we construct the corresponding linear finite element spaces $V_{h_1}, \ldots, V_{h_n}$ such that

$$V_H = V_{h_0} \subseteq V_{h_1} \subset V_{h_2} \subset \ldots \subset V_{h_n} \subset V. \tag{36}$$

In this paper, we assume the following relations of approximation errors hold

$$\eta_a(V_{h_k}) \approx \frac{1}{\beta}\eta_a(V_{h_{k-1}}), \quad \delta_{h_k}(u) \approx \frac{1}{\beta}\delta_{h_{k-1}}(u), \quad k = 2, \ldots, n. \tag{37}$$

**Algorithm 5.1.** *Multigrid Scheme for GPE*

```
(1) Construct a sequence of nested finite element spaces $V_H, V_{h_1}, V_{h_2}, \ldots, V_{h_n}$
    such that (36) and (37) hold.
(2) Solve the GPE on the initial finite element space $V_{h_1}$: Find $(\lambda_{h_1}, u_{h_1}) \in
    \mathbb{R} \times V_{h_1}$ such that $b(u_{h_1}, u_{h_1}) = 1$ and
        $a(u_{h_1}, v_{h_1}) = \lambda_{h_1} b(u_{h_1}, v_{h_1}), \quad \forall v_{h_1} \in V_{h_1}.$
(3) Do $k = 1, \ldots, n-1$
    Obtain a new eigenpair approximation $(\lambda_{h_{k+1}}, u_{h_{k+1}}) \in \mathbb{R} \times V_{h_{k+1}}$ with
    the one correction step being defined by Algorithm 3.1 and the nonlinear
    iteration being defined by Algorithm 4.1
        $(\lambda_{h_{k+1}}, u_{h_{k+1}}) = \texttt{Correction}(V_H, \lambda_{h_k}, u_{h_k}, V_{h_{k+1}}, \varsigma_{h_{k+1}}).$
    End Do
Finally, we obtain an eigenpair approximation $(\lambda_{h_n}, u_{h_n}) \in \mathbb{R} \times V_{h_n}$.
```

The error estimates for Algorithm 5.1 can be stated as follows.

**Theorem 5.1.** *(*[27, Theorem 4.1,Corollary 4.1]*) Assume $h_1 < h_0$ (as in Lemma 2.1) and the error $\varsigma_{h_{k+1}}$ of the linear solving by the multigrid method in the correction step on the $(k+1)$-th level mesh satisfies $\varsigma_{h_{k+1}} \leq \eta_a(V_{h_k})\delta_{h_k}(u)$ for $k = 1, \ldots, n-1$. After implementing Algorithm 5.1, the resultant eigenpair approximation $(\lambda_{h_n}, u_{h_n})$ has following error estimates*

$$\|\bar{u}_{h_n} - u_{h_n}\|_1 \lesssim \beta^2 \eta_a(V_{h_n})\delta_{h_n}(u), \tag{38}$$

$$\|\bar{u}_{h_n} - u_{h_n}\|_0 \lesssim \eta_a(V_{h_n})\delta_{h_n}(u), \tag{39}$$

$$|\bar{\lambda}_{h_n} - \lambda_{h_n}| \lesssim \eta_a(V_{h_n})\delta_{h_n}(u), \tag{40}$$

*under the condition $C\beta^2\eta_a(V_H) < 1$ for the concerned constant $C$.*

*Furthermore, we have following optimal error estimates*

$$\|u - u_{h_n}\|_1 \lesssim \delta_{h_n}(u), \tag{41}$$

$$\|u - u_{h_n}\|_0 \lesssim \eta_a(V_{h_n})\delta_{h_n}(u), \tag{42}$$

$$|\lambda - \lambda_{h_n}| \lesssim \eta_a(V_{h_n})\delta_{h_n}(u). \tag{43}$$

Now, we come to estimate the computational work for the multigrid scheme defined by Algorithm 5.1 with the nonlinear iteration method defined by Algorithm 4.1. Since the linear boundary value problem (10) in Algorithm 3.1 is solved by multigrid method, the computational work is asymptotically optimal.

First, we define the dimension of each level linear finite element space as

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

Then we have

$$N_k \approx \Big(\frac{1}{\beta}\Big)^{d(n-k)} N_n, \quad k = 1, \ldots, n. \tag{44}$$

Different from the method in [27], the computational work for the second step in Algorithm 3.1 with the nonlinear iteration method in Algorithm 4.1 is $\mathcal{O}(N_k + \varpi M_H)$ in each level space $V_{h_k}$.

**Theorem 5.2.** *Assume solving the linear eigenvalue problem (15) in the coarse spaces $V_{H,h_k}$ $(k = 1,\ldots,n)$ and $V_{h_1}$ need work $\mathcal{O}(M_H)$ and $\mathcal{O}(M_{h_1})$, respectively, and the work of the multigrid method for solving the source problem in $V_{h_k}$ is $\mathcal{O}(N_k)$ for $k = 2,3,\ldots,n$. Let $\varpi$ denote the nonlinear iteration times when we solve the nonlinear eigenvalue problem (11). Then the work involved in Algorithm 5.1 has the following estimate:*

$$\text{Total work} = \mathcal{O}\big(N_n + \varpi M_H \log N_n + \varpi M_{h_1}\big). \tag{45}$$

*Proof.* Let $W_k$ denote the work in the $k$-th finite element space $V_{h_k}$. Then with the correction definition in Algorithms 3.1, 4.1 and Remark 4.1, we have

$$W_k = \mathcal{O}\left(N_k + \varpi M_H\right). \tag{46}$$

Iterating (46) and using the fact (44), we obtain

$$\begin{aligned}
\text{Total work} &= \sum_{k=1}^{n} W_k = \mathcal{O}\left(\varpi M_{h_1} + \sum_{k=2}^{n}\Big(N_k + \varpi M_H\Big)\right)\\
&= \mathcal{O}\Big(\sum_{k=2}^{n} N_k + (n-1)\varpi M_H + \varpi M_{h_1}\Big)\\
&= \mathcal{O}\left(\sum_{k=2}^{n}\Big(\frac{1}{\beta}\Big)^{d(n-k)} N_n + \varpi M_H \log N_n + \varpi M_{h_1}\right)\\
&= \mathcal{O}\big(N_n + \varpi M_H \log N_n + \varpi M_{h_1}\big). \qquad (47)
\end{aligned}$$

This is the desired result (45) and we complete the proof. □

**Remark 5.1.** *The estimate of the computational work (45) is an essential improvement from the estimate in* [27]*. With the help of the implementing technique defined in Algorithm 4.1, the nonlinear iteration times affect the final computational work by $\varpi M_H$ and $\varpi M_{h_1}$ which is very small scale since $M_H \ll N_{h_n}$ and $M_{h_1} \ll N_{h_n}$. It means that the final computational work is asymptotically optimal and depends very weakly on the the nonlinearity of GPE.*

## 6. Numerical examples

In this section, we provided three numerical examples to validate the efficiency of the multigrid method stated in Algorithm 5.1 with the nonlinear iteration technique defined in Algorithm 4.1. About the convergence behavior of Algorithm 5.1, please refer to [18, 27] which give the corresponding numerical results. Here, we are only concerned with the computing time (in seconds) for Algorithm 5.1 for the eigenvalue problem (1) with different choices of $\zeta$.

**Example 6.1.** *In this example, we solve GPE (1) with the computing domain $\Omega$ being the unit square $\Omega = (0,1)\times(0,1)$, $W = x_1^2 + x_2^2$ and different choices of $\zeta$.*

The sequence of finite element spaces are constructed by using the linear finite element on the sequence of meshes which are produced by regular refinement with $\beta = 2$ (connecting the midpoints of each edge). In this example, we choose the coarse mesh $\mathcal{T}_H = \mathcal{T}_{h_1}$ which is shown in Figure 1 to investigate the CPU time (in seconds) for different $\zeta$.

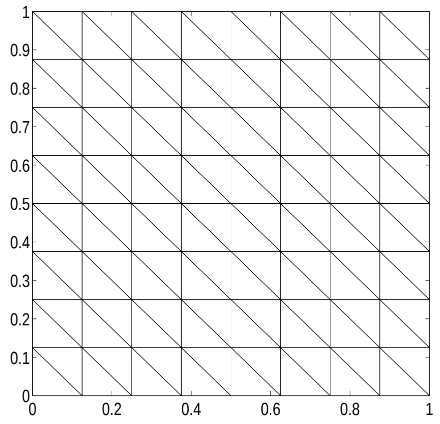

FIGURE 1. The coarse mesh $\mathcal{T}_H = \mathcal{T}_{h_1}$ for Example 6.1.

For comparison, we also present the CPU time of the original multigrid method which has been introduced in [27]. The CPU time results are shown in Figure 2. From Figure 2, we can find that the computational work of Algorithm 5.1 with the nonlinear iteration defined by Algorithm 4.1 is much smaller than that of the original multigrid method in [27]. The computational work of the original multigrid method in [27] has linear scale but depends on the nonlinearity of the problem. It is well known that bigger value of $\zeta$ means stronger nonlinearity of the problem (1). This is why the original multigrid method needs more CPU time for bigger $\zeta$. Figure 2 also shows that the asymptotic computational work for Algorithm 5.1 is almost independent from the nonlinearity (the choice of $\zeta$) of the eigenvalue problem (1) which consists with the estimate (45) in Theorem 5.2.

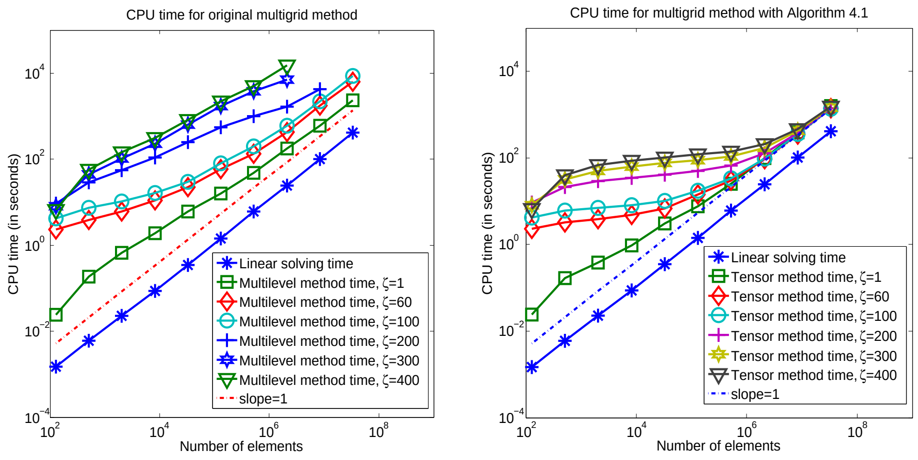

FIGURE 2. The CPU time (in seconds) for two dimensional eigenvalue problem (1). Here linear solving time denotes the CPU time for the linear elliptic boundary value problem by the multigrid method, multilevel method time denotes the CPU time for the original multigrid method in [27] and tensor method time denotes the CPU time for Algorithm 5.1.

**Example 6.2.** *In the second example, we solve GPE (1) with the computing domain $\Omega$ being the unit brick $\Omega = (0,1) \times (0,1) \times (0,1)$, $W = x_1^2 + x_2^2 + x_3^2$ and different choice of $\zeta$.*

The sequence of finite element spaces are constructed by using the linear finite element on the sequence of meshes which are produced by regular refinement with $\beta = 2$ from the coarse mesh $\mathcal{T}_H$ which is shown in Figure 3. In this example, we also use the initial mesh $\mathcal{T}_H = \mathcal{T}_{h_1}$ to investigate the CPU time (in seconds) for different $\zeta$.

In this example, we also present the CPU time for the original multigrid method introduced in [27] for comparison. Figure 4 shows the CPU time results where we can find the same behavior as in Example 6.1. The computational work of Algorithm 5.1 is much smaller than the original multigrid method in [27]. Figure 4 shows that the computational work of the the original multigrid method in [27] depends on the strength of the nonlinearity. Furthermore, the asymptotic computational work for Algorithm 5.1 is almost independent of the nonlinearity (the choice of $\zeta$) of the eigenvalue problem (1) which consists with the estimate (45) in Theorem 5.2.

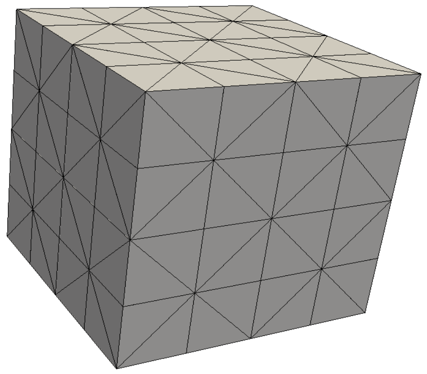

FIGURE 3. The coarse mesh $\mathcal{T}_H = \mathcal{T}_{h_1}$ for Example 6.2.

**Example 6.3.** *In this example, we also solve the GPE (1), where the computing domain $\Omega$ is the L-shape domain $\Omega = (0,2) \times (0,2) \backslash [1,2) \times [1,2)$, $W = x_1^2 + x_2^2$.*

Due to the reentrant corner of $\Omega$, the exact eigenfunction with singularities is expected. The convergence order for approximate eigenpair is less than the order predicted by the theory for regular eigenfunctions. Thus, the adaptive refinement is adopted to couple with the multigrid method described in Algorithm 5.1. Since the exact eigenvalue is not known, we also choose an adequately accurate approximation on a fine enough mesh as the exact one to check the error estimates. We give the numerical results of the multigrid method in which the sequence of meshes $\mathcal{T}_{h_1}, \cdots, \mathcal{T}_{h_n}$ is produced by the adaptive refinement with the following a posteriori error estimator

$$\eta^2(u_{h_k}, K) := h_K^2 \|\mathcal{R}_K(\lambda_{h_k}, u_{h_k})\|_{0,K}^2 + \sum_{e \in \mathcal{E}_I, e \subset \partial K} h_e \|\mathcal{J}_e(u_{h_k})\|_{0,e}^2, \tag{48}$$

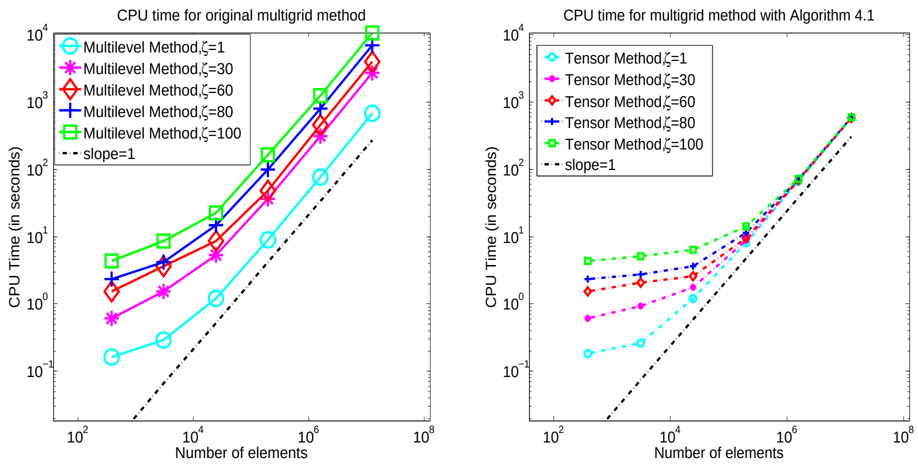

FIGURE 4. The CPU time (in seconds) for three dimensional eigenvalue problem (1). Here multilevel method time denotes the CPU time for the original multigrid method in [27] and tensor method time denotes the CPU time for Algorithm 5.1.

where the element residual $\mathcal{R}_K(u_{h_k})$ and the jump residual $\mathcal{J}_e(u_{h_k})$ are defined as follows:

$$\mathcal{R}_K(\lambda_{h_k}, u_{h_k}) := \lambda_{h_k} u_{h_k} + \Delta u_{h_k} - W u_{h_k} - \zeta |u_{h_k}|^2 u_{h_k}, \qquad \text{in } K \in \mathcal{T}_{h_k}, \tag{49}$$

$$\mathcal{J}_e(u_{h_k}) := -\nabla v^+ \cdot \nu^+ - \nabla v^- \cdot \nu^- := [\nabla v]_e \cdot \nu_e, \qquad \text{on } e \in \mathcal{E}_I. \tag{50}$$

Here $\mathcal{E}_I$ denotes the set of interior faces (edges or sides) of $\mathcal{T}_{h_k}$ and $e$ is the common side of elements $K^+$ and $K^-$ with the unit outward normals $\nu^+$ and $\nu^-$, respectively, and $\nu_e = \nu^-$.

Figure 5 shows the corresponding numerical results by Algorithm 5.1 coupled with the adaptive refinement. From the numerical experiment, it is also observed the errors by Algorithm 5.1 is the same as the original multigrid method in [27] since the difference between these two algorithms is only the implementing technique. From Figure 5, we can also find that Algorithm 5.1 can also work on the adaptive family of meshes and obtain the optimal accuracy.

In this example, for comparison, we also present the CPU time for the original multigrid method introduced in [27]. The CPU time results are shown in Figure 6 which shows the same behavior as in previous examples. The computational work of Algorithm 5.1 is much smaller than the original multigrid method in [27]. Figure 6 shows that the computational work of the the original multigrid method in [27] depends on the strength of the nonlinearity. Furthermore, the asymptotic computational work for Algorithm 5.1 is almost independent from the nonlinearity (the choice of $\zeta$) of the eigenvalue problem (1) even on the adaptive family of meshes.

## 7. Concluding remarks

In this paper, with the help of tensor, we propose an efficient implementing method for the multigrid method introduced in [27] to solve GPE. With the new implementing method for the nonlinear iteration, the asymptotical computational work for solving GPE is almost the same as solving the corresponding linear boundary value problem by the multigrid method, and almost independent of the

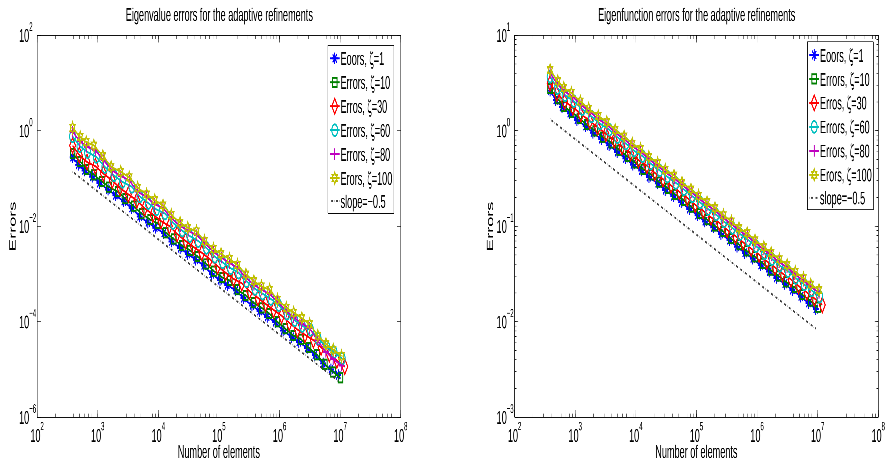

Figure 5. The errors for eigenvalue problem (1) which is solved by the multigrid method coupled with the adaptive refinement. The left subfigure shows the errors for the eigenvalue approximation and the right one shows the posteriori error estimates for the eigenfunction approximations.

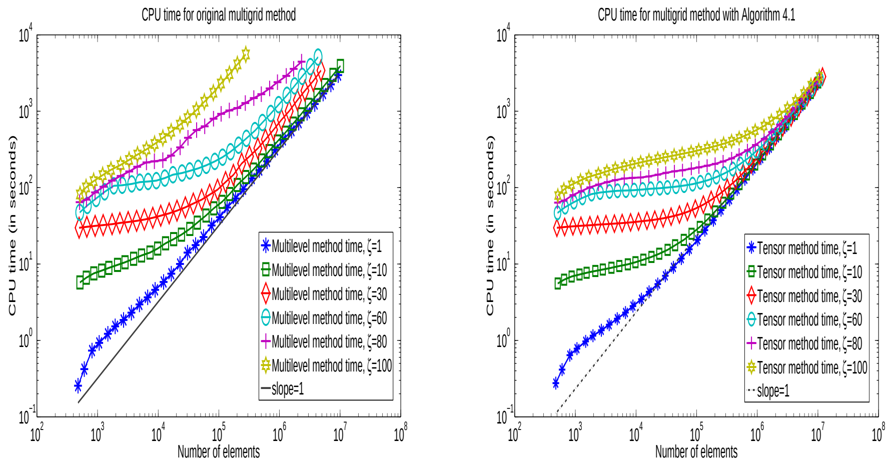

Figure 6. The CPU time (in seconds) for eigenvalue problem (1) which is solved by the multigrid method coupled with the adaptive refinement. Here multilevel method time denotes the CPU time for the original multigrid method in [27] and tensor method time denotes the CPU time for Algorithm 5.1.

nonlinearity of GPE. Three examples are provided to validate the efficiency of the proposed method.

The idea and method here can also be extended to other problems with polynomial or rational type of nonlinearity such as Navier-Stokes, Kohn-Sham equations and some phase models. Furthermore, we can use the algorithms here to design high order nonlinear iteration method for the general nonlinear problems and nonlinear eigenvalue problems.

## Acknowledgments

The authors thank Prof. Qingzhi Yang and Prof. Yimin Wei for theirs discussion and encouragement. We also thank Dr. Yunhui He for her careful checking on this paper. This research was supported in part by Science Challenge Project (No. TZ2016002), National Natural Science Foundations of China (NSFC 91730302, 11771434, 11801021, 91330202, 11371026, 11001259, 11031006, 2011CB309703), the National Center for Mathematics and Interdisciplinary Science, CAS.

## References

[1] R. A. Adams, Sobolev spaces, Academic Press, New York, 1975.

[2] M. H. Anderson, J. R. Ensher , M. R. Mattews , C. E. Wieman and E. A. Cornell, Observation of Bose-Einstein condensation in a dilute atomic vapor, Science, 269 (1995), 198-201.

[3] J. R. Anglin and W. Ketterle, Bose-Einstein condensation of atomic gasses, Nature, 416 (2002), 211-218.

[4] W. Bao and Y. Cai, Mathematical theory and numerical methods for Bose-Einstein condestion, Kinetic and Related Models, 6(1) (2013), 1-135.

[5] W. Bao and W. Tang, Ground-state solution of trapped interacting Bose-Einstein condensate by directly minimizing the energy functional, J. Comput. Phys., 187 (2003), 230-254.

[6] J. H. Bramble, Multigrid Methods, Pitman Research Notes in Mathematics, V. 294, John Wiley and Sons, 1993.

[7] S. Brenner and L. Scott, The Mathematical Theory of Finite Element Methods, New York: Springer-Verlag, 1994.

[8] E. Cancès, R. Chakir, Y. Maday, Numerical analysis of nonlinear eigenvalue problems, J. Sci. Comput., 45(1-3) (2010), 90-117.

[9] C.-S. Chien, H.-T. Huang, B.-W. Jeng and Z.-C. Li, Two-grid discretization schemes for nonlinear Schröinger equations, J. Comput. Appl. Math., 214 (2008), 549-571.

[10] C.-S. Chien, B.-W. Jeng, A two-grid discretization scheme for semilinear elliptic eigenvalue problems, SIAM J. Sci. Comput., 27(4) (2006), 1287-1304.

[11] P. G. Ciarlet, The Finite Element Method for Elliptic Problems, Amsterdam: North-Holland, 1978.

[12] E. A. Cornell and C. E. Wieman, Nobel Lecture: Bose-Einstein condensation in a dilute gas, the first 70 years and some recent experiments, Rev. Mod. Phys., 74 (2002), 875-893.

[13] F. Dalfovo, S. Giorgini, L. P. Pitaevskii and S. Stringari, Theory of Bose-Einstein condensation in trapped gases, Rev. Mod. Phys., 71 (1999), 463-512.

[14] S. Friedland, L. Qi, Y. Wei and Q. Yang, Tensor and hypergraph, Frontiers of Mathematics in China 12(6) (2017), 1277-1277.

[15] E. P. Gross, Nuovo, Cimento., 20 (1961), 454.

[16] W. Hackbush, Multi-grid Methods and Applications, Springer-Verlag, Berlin, 1985.

[17] P. Henning, A. Målqvist and D. Peterseim, Two-level discretization techniques for ground state computations of Bose-Eistein condensates, SIAM J. Numer. Anal., 52(4) (2014), 1525-1550.

[18] S. Jia, H. Xie, M. Xie and F. Xu, A full multigrid method for nonlinear eigenvalue problems, Sci China Math, 59 (2016), 2037-2048.

[19] W. Ketterle, Nobel lecture: When atoms behave as waves: Bose-Einstein condensation and the atom laser, Rev. Mod. Phys., 74 (2002), 1131-1151.

[20] L. Laudau and E. Lifschitz, Quantum Mechanics: non-relativistic theory, Pergamon Press, New York, 1977.

[21] E. H. Lieb, R. Seiringer and J. Yangvason, Bosons in a trap: a rigorous derivation of the Gross-Pitaevskii energy functional, Phys. Rev. A, 61 (2000), 043602.

[22] Q. Lin and H. Xie, A multi-level correction scheme for eigenvalue problems, Math. Comp., 84 (2015), 71-88.

[23] S. F. McCormick, ed., Multigrid Methods. SIAM Frontiers in Applied Matmematics 3. Society for Industrial and Applied Mathematics, Philadelphia, 1987.

[24] H. Xie, A type of multilevel method for the Steklov eigenvalue problem, IMA J. Numer. Anal.,34(2) (2014), 592-608.

[25] H. Xie, A type of multi-level correction scheme for eigenvalue problems by nonconforming finite element methods, BIT Numer. Math., 55 (2015), 1243-1266.

[26] H. Xie, A multigrid method for eigenvalue problem, J. Comput. Phys., 274 (2014), 550-561.

[27] H. Xie and M. Xie, A multigrid method for ground state solution of Bose-Einetein condensates, Commun. Comput. Phys., 19(3) (2016), 648-662.

[28] J. Xu, Iterative methods by space decomposition and subspace correction, SIAM Review, 34(4) (1992), 581-613.

[29] A. Zhou, An analysis of fnite-dimensional approximations for the ground state solution of Bose-Einstein condensates, Nonlinearity, 17 (2004), 541-550.

LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China, and School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, 100049, China

*E-mail*: `zhangning114@lsec.cc.ac.cn`

Beijing Institute for Scientific and Engineering Computing, Beijing University of Technology, Beijing 100124, China

*E-mail*: `xufei@lsec.cc.ac.cn`

LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China, and School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, 100049, China

*E-mail*: `hhxie@lsec.cc.ac.cn`

*URL*: `http://lsec.cc.ac.cn/∼hhxie/`