NUMERICAL SOLUTION OF THE KOHN-SHAM EQUATION BY FINITE ELEMENT METHODS WITH AN ADAPTIVE MESH REDISTRIBUTION TECHNIQUE

GANG BAO[†‡], GUANGHUI HU[†], AND DI LIU[†]

Abstract.
A finite element method with an adaptive mesh redistribution technique is presented for solving the Kohn-Sham equation. The mesh redistribution strategy is based on the harmonic mapping, and the movement of grid points is partially controlled by the monitor function that depends on the gradient of the electron density. Compared with fixed meshes, both efficiency and accuracy of the solution are improved significantly. Effectiveness and robustness of the solver are illustrated by numerical experiments.

Key words. adaptive mesh redistribution, harmonic map, finite element method, Kohn-Sham equation, density functional theory

AMS subject classifications. 65N30, 65N55, 81Q05

1. Introduction.

The Schrödinger equation is fundamental for describing the quantum mechanical electronic structure of matter, since it dose not require any empirical input[30]. The time-independent Schrödinger equation takes the following form

$$H\Psi = E\Psi, \tag{1.1}$$

where H is the Hamiltonian operator, and E and Ψ represent the energy eigenvalue and eigenstate (wave-function), respectively. Unfortunately, it is too expensive to solve Eq. (1.1) directly because of the high dimensionality. In fact, Ψ here is a multi-variable function which depends on the position of every electron in the system. To this day, it is impossible to simulate a small molecule which consists of only a dozen of electrons by solving (1.1) directly. Therefore it is imperative to reduce the computational cost of the solution of the Schrödinger equation.

Various methods have been proposed to simplify the numerical simulation of solutions of the Schrödinger equation. The density functional theory (DFT) is an important one among them. In 1964, Hohenberg and Kohn [16] discovered the one-to-one correspondence between the ground state electron density $n(\vec{\mathbf{x}})$ and the external potential. According to their theorem, $n(\vec{\mathbf{x}})$ can be treated as the fundamental unknown in a many-electron system. Because $n(\vec{\mathbf{x}})$ is a function which only depends on three spatial coordinates, numerical simulations of large molecules or even solids become possible.

Inspired by Hohenberg and Kohn's theoretical results, Kohn and Sham [19] presented an implementation of DFT. The idea is to replace the physical system of interacting electrons with a fictitious system (the Kohn-Sham system) of non-interacting electrons. These two systems have the same ground state electron density. In the Kohn-Sham system, an electron is supposed to move in an effective potential, consists of three terms: the external potential which is usually generated by the attraction between the electron and the nucleus, the Hartree potential which describes the repulsion between electron and electron, and the exchange-correlation potential which

[†]Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA (bao@math.msu.edu, ghhu@math.msu.edu, richardl@math.msu.edu)
[‡]Department of Mathematics, Zhejiang University, Hangzhou 310027, China

is caused by certain non-classical Coulomb interactions. Among these terms, the exchange-correlation potential can not be known analytically, which means that we need some approximate methods to mimic it. In [19], Kohn and Sham introduced a local density approximation (LDA) for the exchange-correlation potential, which is also a key contribution of their work [29]. Since then, many other approximations have been proposed such as the generalized gradient approximation (GGA) [25, 27], the meta-generalized gradient approximation (meta-GGA) [6, 7]. By numerically solving the Kohn-Sham equation, we can determine the ground state electron density of the originally physical system, and therefore other interested quantities.

There are many numerical methods for solving the Kohn-Sham equation. One popular way is to use a plane-wave basis [21], which is based on the plane-wave expansion of the Kohn-Sham orbitals. The plane-wave basis set is orthonormal and the convergence of the calculations increases systematically with the number of the plane-waves. Many software packages have emerged, such as ABINIT [3], VASP [2], and KSSOLV[36] based on the plane-wave basis method. However, there are some drawbacks for this method. For instances, the plane-wave basis method is not suitable for the simulation of the non-periodic system. Because the basis is non-local in real space, it results in dense matrices, which is ill-suited for iterative schemes. To overcome these disadvantages, real-space discretization methods such as finite difference methods, finite element methods and the wavelet methods have been proposed to solve the Kohn-Sham equation. All these real-space methods can adopt different boundary conditions flexibly, and the resulting matrices are sparse which can be saved or operated much more efficiently. For an overview of all these numerical schemes and corresponding software, we refer to [29, 32] and references therein.

We focus on the finite element methods in this paper, which have been widely used in many different fields such as structural mechanics, fluid dynamics, electromagnetics. Early works [10, 30, 37] have been done for the applications of finite element methods on solving the Kohn-Sham equation. One advantage of these methods is the flexibility. Different boundary conditions such as Dirichlet boundary condition, periodic boundary condition can be implemented by the finite element methods in a straightforward way. Compared with the finite difference methods, it is much more suitable to discretize the governing equation on the complicated domain by using the finite element methods. Furthermore, since the finite element methods are based on the variational principle, the ground state energy can be approximated monotonically by refining the mesh or by enriching the order of the local approximation polynomials. In the calculations of the finite element methods, the physical domain is divided into small elements. For example, the tetrahedral element in this paper. It is still a challenging task to generate a high-quality tetrahedral mesh. In this paper, we get the tetrahedral mesh by a software named GMSH[1], which is an open-source software and published under GPL license. For the Kohn-Sham equation, especially for the all-electron calculations, the external potential is singular in the vicinity of the nucleus. Therefore, there is no hope to get a high-quality approximation of the solutions if this singularity is not resolved effectively. On the other hand, the problem in this paper is non-periodic, which means that the physical domain should be sufficiently large to reduce the truncation error from the boundary.

Based on above discussion, using the radial mesh seems to be a good choice. The radial mesh has small mesh size in the vicinity of each nucleus, and large one in the region away from nucleus. For this configuration of mesh, a large amount of grid points locate at the vicinity of nucleus, which can help to resolve the singular

external potential. For the region away from nucleus, since there is almost no variation for the potentials, wave-functions and electron density, just a few of grid points are needed. Based on this idea, two well-designed non-uniform meshes are given in [30, 22] respectively, and excellent numerical results are demonstrated there. Besides generating well-designed mesh beforehand, the adaptive technique is another way to improve the mesh quality.

There exist three classical adaptive methods: the h-adaptive methods which locally refine and coarsen the mesh, the r-adaptive methods which redistribute the grid points while keeping the number of mesh grids unchanged, and the p-adaptive methods which locally enrich the order of the approximate polynomial. The r-adaptive method is also known as the moving mesh method. To solve the Kohn-Sham equation with the h-adaptive methods, there are already some results, see [37, 10]. Interesting discussions may be found on the application of the hp-adaptive methods to solving the Kohn-Sham equation in [32]. In particular, an interesting observation was made in [32] "*The question of the feasibility of molecular dynamics with an unstructured nonuniform FE-mesh is often raised. The FE-community provides the following suggestion, based on experience in other fields: each atom is attached to the mesh, and the mesh is considered to be made of 'rubber'. As the atom moves, the mesh moves along...*". Based on our numerical experience, r-adaptive methods can partially satisfy the above requirements. A motivation of our current work is to propose a framework for solving the Kohn-Sham equation by using the r-adaptive methods. For the early work on this topic, we refer to [33] and references therein.

In this paper, we present a finite element solver with a r-adaptive technique to simulate the Kohn-Sham equation with tetrahedral meshes. The solver consists of two main iterations. The first one is a SCF iteration to generate a self-consistent electron density on the current mesh, while the second iteration is adopted to adaptively optimize the distribution of mesh grids in terms of the electron density. The Kohn-Sham equation is discretized by the standard finite element method. The Hartree potential is obtained by solving the Poisson equation with an algebraic multi-grid (AMG) method. The exchange-correlation potential is approximated by the local density approximation (LDA). On the boundary of the domain, the zero Dirichlet boundary condition is adopted for the wave-function. However, it is not the case for solving the Poisson equation. For the boundary condition of the Poisson equation, because the Hartree potential decays linearly, the boundary value is supplied by the multipole expansion. The NSRQMCG method [15] is the solver for the final generalized eigenvalue problem. To improve the convergence of SCF iteration, a linear mixing scheme is adopted to update the electron density. After the self-consistent electron density is obtained, an adaptive technique for the redistribution of the mesh grids is introduced for improving the mesh quality. The redistribution technique we used in the paper is originally presented by Li. et al. in [23, 24], and has been widely applied in different application fields. A review of the method may be found in [31]. Theoretically, this technique is based on the harmonic mapping, which maps a uniform mesh in a logical domain to a non-uniform mesh in a physical domain. The map is controlled partially by a so-called monitor function.

Based on our numerical experience, the monitor function is always given according to certain posteriori error estimates of numerical solutions or information from the physical solution itself. In the domain for solving the Kohn-Sham equation, compared with the distant regions from the atom, the regions around the atomic nuclei and between atoms of chemical bonds are more important[37]. More grid points are

expected to be arranged in these regions. Therefore, the monitor function is given based on the gradient of the electron density, and it can be observed from the results that those important regions are resolved successfully with this monitor function.

To guarantee the numerical accuracy, it is important to keep the smoothness and orthogonality of mesh grids after the mesh movement. In the r-adaptive methods, these requirements are satisfied by smoothing the monitor function. As mentioned before, the physical domain should be sufficiently large, while the interested region is just the vicinity of each nucleus. Hence the smoothness method proposed in [23, 24] is no longer adequate. By employing a smoothness strategy similar to [35, 34, 17] based on a diffusive mechanism of the monitor function, the mesh quality may be significantly improved as shown in the numerical experiments.

The outline of this paper is as follows. In the next section, the DFT and the Kohn-Sham equation are introduced. In Section 3, the finite element discretization of the Kohn-Sham equation is described. A mesh redistribution technique is also introduced in Section 4. In Section 5, some details of numerical techniques are discussed on our algorithm. Numerical simulations are demonstrated in Section 6, which illustrate the reliability and the effectiveness of our method. Finally, the paper is concluded with some general remarks as well as discussions on future directions in Section 7.

2. The Kohn-Sham Equation. In [19], Kohn and Sham presented the following self-consistent equation to reach the minimization of the total energy of a many-electron system (we use the atomic units $e^2 = \hbar = m = 1$ hereafter)

$$\left(\hat{T} + v_{eff}(\vec{\mathbf{x}})\right)\psi_i(\vec{\mathbf{x}}) = \varepsilon_i \psi_i(\vec{\mathbf{x}}), \tag{2.1}$$

where $\hat{T} = -\frac{1}{2}\nabla^2$ denotes the kinetic energy, $v_{eff}(\vec{x})$ is the effective potential with the following expression

$$v_{eff}(\vec{\mathbf{x}}) = v_{ext}(\vec{\mathbf{x}}) + v_{Hartree}(\vec{\mathbf{x}}) + v_{xc}(\vec{\mathbf{x}}). \tag{2.2}$$

For a closed system of M nuclei and N electrons, the external potential

$$v_{ext}(\vec{\mathbf{x}}) = -\sum_{i}^{M} \frac{Z_i}{|\vec{\mathbf{x}} - \vec{\mathbf{x}}_i|}, \tag{2.3}$$

describes the attraction between the nucleus and electron. Here Z_i is the i-th nucleus charge, and $\vec{\mathbf{x}}_i$ is the position of the i-th nucleus. The Hartree potential $v_{Hartree}$ is the Coulomb interaction of electrons

$$v_{Hartree}(\vec{\mathbf{x}}) = \int \frac{n(\vec{\mathbf{x}}')}{|\vec{\mathbf{x}} - \vec{\mathbf{x}}'|} d\vec{\mathbf{x}}'. \tag{2.4}$$

In the simulation, direct evaluating the above integral will result in $\mathcal{O}(N^2)$ operations, which is very time-consuming. A popular way to address $v_{Hartree}$ is to solve the following Poisson equation

$$-\nabla^2 v_{Hartree}(\vec{\mathbf{x}}) = 4\pi n(\vec{\mathbf{x}}). \tag{2.5}$$

with a proper Dirichlet boundary condition.

The last term in (2.2) is the exchange-correlation potential $v_{xc}(\vec{x})$. This term is caused by the Pauli exclusion principle and other non-classical Coulomb interactions.

Unfortunately, the analytical expression of the exchange-correlation potential for a general system is unknown. Thus this term needs to be approximated. The most popular way is to use the local density approximation (LDA) originally proposed by Kohn and Sham in [19]. The exchange-correlation energy $E_{xc}^{LDA}(n)$ is given as

$$E_{xc}^{LDA}(n) = \int \epsilon_{xc}(n) n(\vec{\mathbf{x}}) d\vec{\mathbf{x}}. \tag{2.6}$$

where $\epsilon_{xc}(n)$ stands for the exchange-correlation energy per unit volume of the homogeneous electron gas of the density n. Then v_{xc} can be addressed as

$$v_{xc}(\vec{\mathbf{x}}) = \frac{\delta E_{xc}}{\delta n}. \tag{2.7}$$

The exchange-correlation energy E_{xc} can be separated as two parts $E_{xc} = E_x + E_c$. We adopt the expression of the exchange energy E_x proposed in [19]

$$E_x^{LDA}(n) = -\frac{3}{4}(\frac{3}{\pi})^{1/3} \int n(\vec{\mathbf{x}})^{4/3} d\vec{\mathbf{x}}. \tag{2.8}$$

Then $v_x = -(3n(\vec{\mathbf{x}})/\pi)^{1/3}$. The approximation of E_c is much more complicated than that of E_x. Even in the free electron gas, we only know the expressions of E_c of the high-density limit and the low-density limit. In the simulation, the parametrization of Perdew and Wang [28] fitted to accurate Monte Carlo simulations carried out by Ceperley and Alder [11] is adopted.

From (2.1), we can see that the wave-function $\psi_i(\vec{x})$ depends on the effective potential $v_{eff}(\vec{\mathbf{x}})$, while the electron density $n(\vec{x})$ which has the following relationship with $\psi_i(\vec{x})$

$$n(\vec{\mathbf{x}}) = \sum_{i}^{occ} |\psi_i(\vec{\mathbf{x}})|^2, \tag{2.9}$$

determines the effective potential v_{eff} at the meantime. That means the equation (2.1) is non-linear, and we need a self-consistent field (SCF) method to solve it.

Finally, we present the formula to calculate the ground state energy of the physical system from the solution of the Kohn-Sham equation:

$$E = \sum_{i}^{occ} \varepsilon_i - \int \left(\frac{1}{2} v_{Hartree} + v_{xc}\right) n(\vec{\mathbf{x}}) d\vec{\mathbf{x}} + E_{xc}. \tag{2.10}$$

Note that for the simulations of a molecule which contains several nuclei, the total energy should be added to a repulsive Coulomb term that accounts for the interactions between the nuclei

$$E_{nn} = \sum_{j,k} \frac{Z_j Z_k}{|\vec{x}_j - \vec{x}_k|}.$$

To solve (2.1), many numerical methods have been proposed. Here, we focus on the finite element method for discretizing the governing equation because of its flexibility on different geometries and boundary conditions. The finite element discretization of (2.1) will be introduced in the next section.

3. Numerical Discretization. Assume that the physical domain is denoted by $\Omega \in \mathcal{R}^3$, and Ω is divided into a set of tetrahedron $\mathcal{T}$ with $\mathcal{T}_i$ as its element and $\vec{\mathbf{X}}_i$ as its nodes. The piecewise linear finite element space V_h is built on $\mathcal{T}$. Note that for the linear finite element case, the interpolation points of the degrees of freedom (Dof) in each element locate on the vertexes of the tetrahedral element (3D case).

Based on the above assumptions, the wave-function $\psi(\vec{\mathbf{x}})$ in (2.1) can be approximated as

$$\psi_h = \sum_{i}^{N_{basis}} \psi_i \phi_i, \tag{3.1}$$

where N_{basis} stands for the dimension of space V_h, and ϕ_i is the i-th basis function and ψ_i is its coefficient, which is also the value of the wave-function itself on the corresponding node.

For the finite element method, the variational approach is used to find out the coefficients $\psi = (\psi_0, \psi_1, \psi_2, \cdots, \psi_{N_{basis}})^T$, which leads to the following linear system

$$A\psi = \varepsilon B\psi. \tag{3.2}$$

Here A and B are two matrices with the entries

$$A_{i,j} = \frac{1}{2}\int_{\Omega} \nabla\phi_i \cdot \nabla\phi_j d\vec{\mathbf{x}} + \int_{\Omega} v_{eff}\phi_i\phi_j d\vec{\mathbf{x}}, \tag{3.3}$$

$$B_{i,j} = \int_{\Omega} \phi_i\phi_j d\vec{\mathbf{x}}, \tag{3.4}$$

respectively.

The equation (3.2) is actually a generalized eigenvalue problem with good properties, because the matrices A and B are both Hermitian, and the mass matrix B is positive definite. There are many solvers in the literature to solve this kind of problem such as the Aronldi algorithm [5] (because A is symmetric, it is also known as the Lanczos algorithm) which can be implemented by a software ARPack [4], and preconditioned conjugate gradient (PCG) method. In our algorithm, we use the NSRQMCG method [15] to search for the leftmost eigenpairs $(\varepsilon_i, \psi_i), i = 1, 2, \cdots$ of the equation (3.2).

To discretize the Poisson equation (2.5), we use the same finite element space mentioned above. The Hartree potential $v_{Hartree}$ can be approximated as

$$v_{H,h} = \sum_{i}^{N_{basis}} v_{H,i}\phi_i, \tag{3.5}$$

where $v_{H,i}$ is the coefficient of the i-th basis function ϕ_i. Let $v_H = (v_{H,0}, v_{H,1}, \cdots, v_{H,N_{basis}})^T$, then the final system can be read as

$$Pv_H = f, \tag{3.6}$$

where P is the stiffness matrix with entry

$$P_{i,j} = \int_{\Omega} \nabla\phi_i \nabla\phi_j d\vec{\mathbf{x}}, \tag{3.7}$$

and the right hand side f is a vector with entry

$$f_i = \int_\Omega 4\pi n(\vec{\mathbf{x}})\phi_i d\vec{\mathbf{x}}. \tag{3.8}$$

Note that in the above integration, the evaluation of the electron density $n(\vec{x})$ on a quadrature point $\vec{x}_q$ is given by

$$n(\vec{x}_q) = \sum_{i}^{occ} \sum_{j,k}^{N_{basis}} \phi_j \psi_{i,j}(\vec{x}_q)\phi_k \psi_{i,k}(\vec{x}_q).$$

An efficient solver for the Poisson equation is important, because (3.6) needs to be solved in each SCF iteration. In our algorithm, the algebraic multi-grid method is adopted. The implementation issues of this AMG solver will be discussed in Section 5.

The electron density is obtained by a SCF iteration. That is, the wave-function ψ is solved from (3.2), and then the electron density $n(\vec{x})$ is obtained from (2.9). Then the effective potential in (2.1) is evaluated in terms of the new $n(\vec{x})$. To end up the SCF iteration, the following criterion is used

$$\|n^{N+1}(\vec{x}) - n^N(\vec{x})\|_2 < tol, \tag{3.9}$$

where $n^N(\vec{x})$ and $n^{N+1}(\vec{x})$ mean the electron densities obtained from two adjacent iterations, and tol is an user-defined tolerance.

Once the self-consistent electron density is obtained, an adaptive technique of the mesh redistribution will be introduced to optimize the mesh quality, hence to improve the accuracy of the numerical solution. In the next section, the mechanism of this technique is presented.

4. Mesh Redistribution. The mesh redistribution method in this paper is based on the harmonic mapping. Since this technique was originally proposed in [23, 24], it has been widely applied to a variety of computational problems. We first briefly summarize the mechanism of this technique. Since we mainly focus on 3D simulations of the Kohn-Sham equation in this paper, the following definitions and description are all given in the 3D Euclidean space.

4.1. Harmonic maps. Suppose there are two compact Riemannian manifolds Ω and Ω_c with the metric tensors d_{ij} and $r_{\alpha\beta}$ in certain local coordinates $\vec{x}$ and $\vec{\xi}$. Define a map $\vec{\xi} = \vec{\xi}(\vec{x})$ between Ω and Ω_c, then the energy of this map is given by

$$E(\vec{\xi}) = \frac{1}{2}\int \sqrt{d} d^{ij} \frac{\partial \xi^\alpha}{\partial x^i}\frac{\partial \xi^\beta}{\partial x^j} d\vec{\mathbf{x}}, \tag{4.1}$$

where $d = det(d_{ij})$, $(d_{ij}) = (d^{ij})^{-1}$, and the standard summation convection is assumed. If the map ξ is an extremum of (4.1), ξ is called a harmonic map. This map is the solution of the following Euler-Lagrange equation

$$\frac{\partial}{\partial x^i}(G^{ij}\frac{\partial \xi^k}{\partial x^j}) = 0, \tag{4.2}$$

where the inverse of $G^{ij} = \sqrt{d}d^{ij}$ is called the monitor function.

The harmonic function ξ gives a continuous, one-to-one mapping from Ω to Ω_c with continuous inverse, which is differentiable and has a non-zero Jacobian [31]. In practice, we always use Ω as the *physical* domain, and Ω_c as a reference domain, called the *logical* domain. With the help of harmonic maps, we can generate a non-uniform mesh in the physical domain Ω from an uniform mesh in the logical domain Ω_c by partially controlling the monitor function. To guarantee the existence and the uniqueness of the harmonic map, the Riemannian curvature of Ω_c should be non-positive and its boundary should be convex[23]. In the simulations, we can directly use the physical domain (a cube) as the logical domain, and such selection obviously satisfies the above requirements.

4.2. Mesh redistribution using harmonic maps. Suppose the SCF iteration has been finished on current mesh, and the self-consistent electron density and wave-functions have been obtained. What we do in the next step is to reasonably redistribute the mesh grids in terms of those self-consistent quantities, with the help of the harmonic map.

In the practice, the harmonic map is reached by an iterative procedure. Assume that an initial(fixed) mesh on the logical domain Ω_c is given, the general framework of generation of the harmonic map can be described as the following.

Flow Chart of the Algorithm 1

S. 1 obtain the difference between the current mesh which is obtained from solving (4.3) and the initial mesh in the logical domain Ω_c. If the L^∞ norm of the difference is smaller than a given tolerance, the mesh redistribution is done, otherwise, go to next step;

S. 2 obtain the directions and the magnitudes of the movement of mesh grids in the physical domain Ω in terms of the difference given in **S. 1**, and then update the mesh in the physical domain;

S. 3 update solutions on the updated mesh in the physical domain, then go to **S. 1**.

In [23], the system which is used to generate the mesh in the logical domain in **S. 1** is (4.2) together with a Dirichlet boundary condition. However, with this system, only the interior grids are redistributed, while the grid points on the boundary are unchanged. Under this situation, the redistribution of grids on the boundary of the physical domain should be handled separately, such as using homogeneous Neumann boundary conditions, extrapolating the interior mesh points to the boundary, and relocating the mesh points by solving a lower-dimensional moving-mesh PDE. However, all these strategies may affect the efficiency of the algorithm.

To redistribute the whole mesh grids in an uniform manner, Li et al. [24] proposed an optimization problem for the harmonic map in place of the boundary value problem in [23]. The optimization problem can be read as

$$\begin{cases} \min \sum_k \int_\Omega M \dfrac{\partial \xi^k}{\partial x^i} \dfrac{\partial \xi^k}{\partial x^j} d\vec{\mathbf{x}}, \\ \text{s. t.} \quad \xi|_{\partial\Omega} = \xi_b \in \mathbf{K}, \end{cases} \tag{4.3}$$

where $K = \{\xi_b \in C^0(\partial\Omega) | \xi_b : \partial\Omega \to \Omega_c; \xi_{b|\Lambda_i}$ is a linear segment and strictly increasing$\}$ denotes a mapping set from $\partial\Omega$ to $\partial\Omega_c$. By solving the above optimization problem,

the mesh points including those on the boundary may be redistributed simultaneously.

In the following, details in each step of Algorithm 1 will be briefly demonstrated. Assume that the tetrahedral mesh in the physical domain Ω is denoted by $\mathcal{T}$, with $\mathcal{T}_i$ as its element, and X_i as its node. For the logical domain Ω_c, we use $\mathcal{T}_c$ to denote the corresponding tetrahedral mesh, and $\mathcal{T}_{i,c}$ and $\mathcal{A}_i$ as its element and node, respectively.

To obtain the initial mesh in the logical domain Ω_c, we need to generate the initial mesh $\mathcal{T}_c^{ini}$ in the logical domain Ω_c by solving the following optimization problem

$$\begin{cases} \min \sum_k \int_\Omega \sum_i (\frac{\partial \xi^k}{\partial x^i})^2 d\vec{\mathbf{x}}, \\ \\ \text{s. t.} \quad \xi|_{\partial\Omega} = \xi_b \in \mathbf{K}. \end{cases} \tag{4.4}$$

During the whole procedure, this initial mesh is only a reference, and unchanged.

In the first step of the Algorithm 1, the optimization problem (4.3) needs to be solved. Note that at the current stage, we only try to improve the mesh quality i. e., not the physical solution, the efficiency of the solver for (4.3) is more important than its accuracy. However, the linear system which is obtained from (4.3) is neither Hermitian nor positive definite. To resolve this difficulty, it was suggested in [24] that the system be decoupled into two smaller systems: one is for the grid points on the boundary, and the other one is for the interior grid points. Although there is also no good property for the first system, it is much smaller than the original system, and a generalized minimal residual method (GMRES) or a biconjugate gradient (BiCG) method can be used to solve it. For the second one, a multi-grid method is adopted to solve it because it has good properties (Hermitian and positive definite).

To further improve the efficiency of the solver for (4.3), Di et al. [13] proposed a new approach to speed up the implementation. This new approach is based on an algebraic multi-grid method, together with a constraint for grid points on the boundary. In our simulations, this technique is adopted for solving (4.3). We refer to [13] and references therein for details.

After (4.3) is solved, we obtain a mesh $\mathcal{T}_c$ in the logical domain Ω_c. If the L^∞ norm of the difference between $\mathcal{T}_c$ and $\mathcal{T}_c^{ini}$ is small enough, say,

$$|A_i - A_i^{ini}|_\infty < tol,$$

the iteration is stop, otherwise we will use this difference to generate the movement of mesh grids in physical domain Ω.

In Step 2 of Algorithm 1, we use the following formula to generate the direction and magnitude of movement for each mesh grid in the physical domain,

$$\delta X_i = \frac{\sum_{\mathcal{T}} |\mathcal{T}| \frac{\partial \vec{x}}{\partial \xi}|_{\text{in } \mathcal{T}} \delta A_i}{\sum_{\mathcal{T}} |\mathcal{T}|},$$

where $\delta A_i = A_i^{ini} - A_i$, and $\mathcal{T}$ stands for the element in the physical domain with $|\mathcal{T}|$ as its volume. In the above formula, $\partial \vec{x}/\partial \xi$ is given by solving the following system

in each element,

$$\begin{pmatrix} A^1_{\mathcal{T}_{c,1}} - A^1_{\mathcal{T}_{c,0}} & A^1_{\mathcal{T}_{c,2}} - A^1_{\mathcal{T}_{c,0}} & A^1_{\mathcal{T}_{c,3}} - A^1_{\mathcal{T}_{c,0}} \\ A^2_{\mathcal{T}_{c,1}} - A^2_{\mathcal{T}_{c,0}} & A^2_{\mathcal{T}_{c,2}} - A^2_{\mathcal{T}_{c,0}} & A^2_{\mathcal{T}_{c,3}} - A^2_{\mathcal{T}_{c,0}} \\ A^3_{\mathcal{T}_{c,1}} - A^3_{\mathcal{T}_{c,0}} & A^3_{\mathcal{T}_{c,2}} - A^3_{\mathcal{T}_{c,0}} & A^3_{\mathcal{T}_{c,3}} - A^3_{\mathcal{T}_{c,0}} \end{pmatrix} \begin{pmatrix} \frac{\partial x^1}{\partial \xi^1} & \frac{\partial x^1}{\partial \xi^2} & \frac{\partial x^1}{\partial \xi^3} \\ \frac{\partial x^2}{\partial \xi^1} & \frac{\partial x^2}{\partial \xi^2} & \frac{\partial x^2}{\partial \xi^3} \\ \frac{\partial x^3}{\partial \xi^1} & \frac{\partial x^3}{\partial \xi^2} & \frac{\partial x^3}{\partial \xi^3} \end{pmatrix}$$

$$= \begin{pmatrix} X^1_{\mathcal{T}_1} - X^1_{\mathcal{T}_0} & X^1_{\mathcal{T}_2} - X^1_{\mathcal{T}_0} & X^1_{\mathcal{T}_3} - X^1_{\mathcal{T}_0} \\ X^2_{\mathcal{T}_1} - X^2_{\mathcal{T}_0} & X^2_{\mathcal{T}_2} - X^2_{\mathcal{T}_0} & X^2_{\mathcal{T}_3} - X^2_{\mathcal{T}_0} \\ X^3_{\mathcal{T}_1} - X^3_{\mathcal{T}_0} & X^3_{\mathcal{T}_2} - X^3_{\mathcal{T}_0} & X^3_{\mathcal{T}_3} - X^3_{\mathcal{T}_0} \end{pmatrix},$$

where $(X^1_{\mathcal{T}_i}, X^2_{\mathcal{T}_i}, X^3_{\mathcal{T}_i})$ means the i-th vertex of the element $\mathcal{T}$ in physical domain Ω, and $(A^1_{\mathcal{T}_i}, A^2_{\mathcal{T}_i}, A^3_{\mathcal{T}_i})$ means the i-th vertex of its corresponding element $\mathcal{T}_c$ in logical domain Ω_c. Because the topological structure of the mesh will not change during the iteration, the correspondence between the mesh in Ω and Ω_c will also not change.

After we get the movement of mesh grids in Ω, the mesh in the physical domain can be updated with

$$X^{new} = X + \tau \delta X,$$

where $\tau \in [0, 1]$. Here τ is used to prevent the mesh tangling.

In [24], the solution update is done under the assumption that the surface of the solution is unchanged with respect to the movement of the grid points of the physical mesh. The new solution is obtained by solving an ordinary differential equation derived from the above assumption. However, our problem is time-independent. Hence we may adopt a much simpler way to update the solution. In the implementation, the values of the degrees of freedom are actually unchanged during the iteration for mesh redistribution. This update strategy is reasonable because that when the system is close to an equilibrium state, the variation of electron density becomes small, so is the mesh grid. Therefore the updated wave-functions are good approximations to the exact wave-functions by using our strategy, which make good convergence of SCF iteration on the new mesh. From the numerical experiments in the last section, it is clear that such solution update strategy works well.

The monitor function M in Step 1 of Algorithm 1 is important. A good monitor function can help on resolving the important regions in the physical domain, while keeps the high quality of the mesh. In the following subsection, a monitor function is given, and a special smoothness strategy of the monitor function is also discussed.

4.3. Monitor Function. As mentioned in the introduction, the important regions in the simulation of the Kohn-Sham equation are the vicinity of nucleus and between atoms of chemical bonds. Based on our numerical experience, the variation of the electron density in those regions is much larger than that in other regions. Therefore, we choose the following monitor function

$$M = \sqrt{\epsilon + |\nabla n(\vec{\mathbf{x}})|^2}, \tag{4.5}$$

where $n(\vec{x})$ is the electron density. With (4.5), the grid points move towards the region with large gradient of the electron density, and the adaptivity is controlled by the parameter ϵ. The smaller ϵ results in more adaptivity. Note that the problem is

non-periodic, a sufficiently large domain is needed to reduce the truncation error. At the meantime, the large variation of the density only appears in the vicinity of the nucleus, which is a relatively very small region. Consequently, the monitor function (4.5) results in a dramatic change of the mesh size around the nucleus. Fig. 4.1 ((a) and (b)) demonstrates this phenomena. Two drawbacks are observed from the figure: (i), although the density of grid points in the vicinity of the nucleus becomes large after the movement of the mesh, the large distortion of the element is also observed nearby the nucleus, which may negatively affects on the quality of the numerical results; (ii), there are too many grid points in the outer layer of the domain. Because there is almost no variation of the electron density far away from the nucleus, the positions of grid points in that region are almost unchanged by using the monitor function (4.5). Based on the above two observations, moving the grid points in the region away from the nucleus towards to the important regions is a good idea to further improve the mesh quality.

A similar situation which is mentioned above has been discussed by Wang et al. in [35]. They avoid this problem by introducing a diffusive process in the monitor function. This strategy is also adopted by Hu and Zegeling in [17]. In the following, we will give the implementation of this technique in our algorithm.

Instead of using a constant parameter ϵ in (4.5), a function $\tilde{n}$ which is the solution of the following equation

$$\frac{\partial \tilde{n}}{\partial t} - \delta \Delta \tilde{n} = |\nabla n(\vec{x})|^2 \tag{4.6}$$

is adopted, where δ is a parameter which depends on the size of the physical domain. With the function $\tilde{n}$, the monitor function becomes the following form

$$M = \sqrt{\tilde{n} + |\nabla n(\vec{\mathbf{x}})|^2}. \tag{4.7}$$

Because we only want a rough solution of the above equation, the increment of the CPU time which is caused by solving (4.7) is not significant, with the help of the AMG method. The significant improvement of the mesh quality can be observed from Fig. 4.1 ((c) and (d)).

Although most of the applications of the r-adaptive method are time-dependent, it makes sense to use this method for the calculation of the ground state of a many-electron system, which is actually a time-independent problem. After the self-consistent electron density is obtained, the mesh is redistributed adaptively according to this physical quantity, and those important regions in the physical domain is resolve by the mesh grids much more reasonably. Consequently, a much more accurate numerical solution can be expected on the new mesh with the same number of mesh grids.

5. Computational Issues. We have stated the finite element discretization of the Kohn-Sham equation, and introduced Algorithm 1 to improve the quality of the numerical results. In this section, we present some technical details in the algorithm, and the flow chart of the whole algorithm at the end.

5.1. The Algebraic Multi-grid Method. To efficiently solve (3.6), an algebraic multi-grid (AMG) method which is proposed in [12] is adopted. The implementation of AMG method can be briefly summarized as the following.

There are two essential components in the multi-grid methods: The projection operator and the smoother. The projection operator is used to construct the system on the coarse mesh from the original dense mesh, and correct the results on the dense

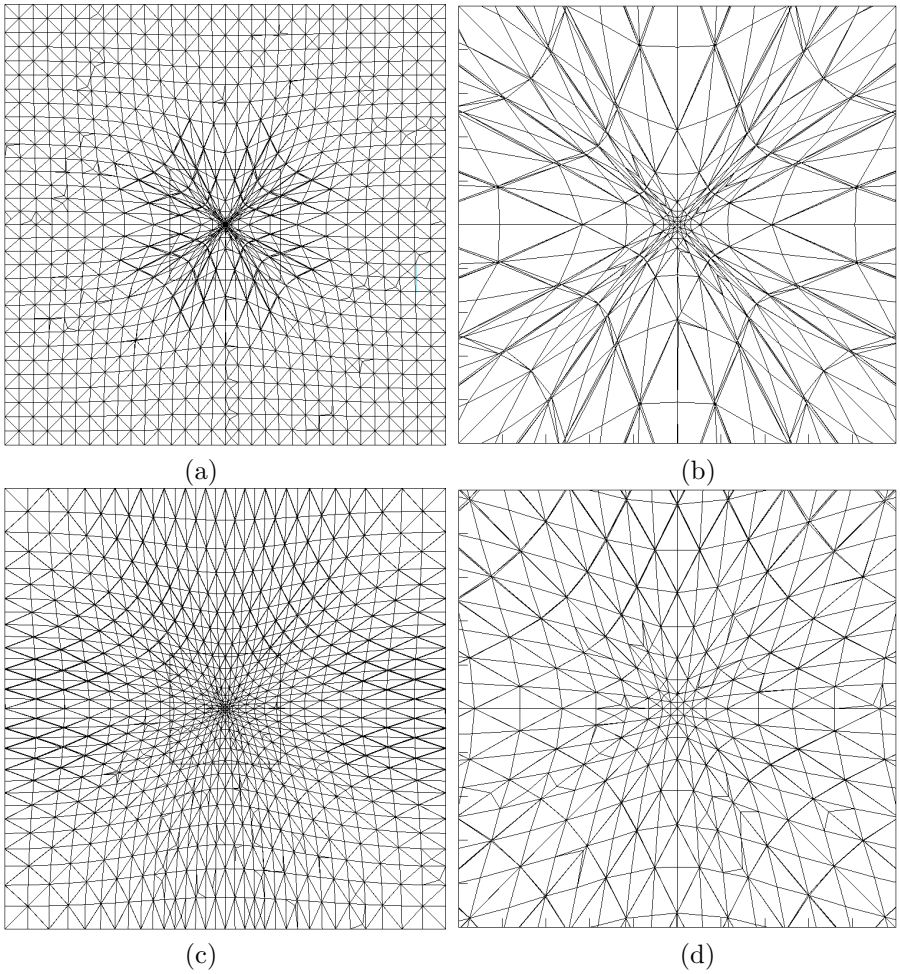

FIG. 4.1. *The demonstration of a slice of the 3-dimensional mesh in the plane* $(0, 0, 1)$ *after the mesh redistribution. (a) and (b) are generated with (4.5), while (c) and (d) are generated with (4.7). (b) and (d) show the detailed mesh around the nucleus of the whole mesh (a) and (c) respectively.*

mesh by the solution obtained from the coarse mesh. There are two different way to get the projection operator. One is the geometrical method which uses the information of the mesh to generate the projection operator. The other one is the algebraic method which only use the information of linear system itself. We follow [12] to use the algebraic method to construct the projection operator in the implementation.

The smoother in the multi-grid method is employed to damp out the relatively high frequency parts of the numerical error of the result, with respect to the current mesh level. In our implementation, the Gauss-Seidel iteration method is adopted. To cancel the error effectively, a lower-upper process is implemented.

For the detailed algorithm of AMG, we refer to [12] and references therein. Here we present a test which demonstrates that our implementation of AMG method works well for solving the Poisson problem. For the problem, a cubic domain $[-1, 1] \times [-1, 1] \times [-1, 1]$ is chosen. The equation we want to solve is

$$-\Delta u = f. \tag{5.1}$$

Since it is only a test, we use $u = sin(\pi x)cos(\pi y)sin(\pi z)$ as the solution to setup the boundary condition and f respectively. Table 5.1 demonstrates the reliability of our AMG solver.

Dof	63	365	2457	17969	137313
l_2 error	4.45e-01	2.37e-01	6.38e-02	1.63e-02	4.10e-03
Conv. Order		0.91	1.89	1.97	1.99

TABLE 5.1

The l_2 error and the corresponding convergence order of the the solution of (5.1). The finite element method is used to discretize the equation with piecewise linear approximation, and the linear system is solved by using the AMG method.

5.2. Boundary Conditions. To solve the equations (3.2) and (3.6), appropriate boundary conditions are needed. For (3.2), the zero Dirichlet boundary conditions are adopted naturally. However, for (3.6), the zero Dirichlet boundary condition is no longer suitable. As we know that, far from a nucleus, the Hartree potential decays as N/r, where N is the electron number. Consequently, the simple use of zero Dirichlet boundary condition will introduce large truncation error on the boundary. To give the the evaluation of the Hartree potential on the boundary, (2.4) can be used directly. However, it results in a $\mathcal{O}(N^{\frac{5}{3}})$ operations in the algorithm, which is not consistent with $\mathcal{O}(N)$ of the AMG solver. To reduce the cost, a multipole expansion approximation is employed for the boundary value. In the simulations, the following three terms, monopole, dipole and quadrupole, are used for the approximation,

$$V_{Hartree}(\vec{x})|_{\vec{x}\in\partial\Omega} \approx V_{mon}(\vec{x}) + V_{dip}(\vec{x}) + V_{quad}(\vec{x}),$$

where

$$\begin{aligned} V_{mon}(\vec{x}) &= \frac{1}{4\pi|\vec{x}|}\int_{\Omega'} n(\vec{x}')d\vec{x}', \\ V_{dip}(\vec{x}) &= -\frac{1}{4\pi|\vec{x}|^2}\int_{\Omega'} n(\vec{x}')(\hat{\vec{x}}\cdot\vec{x}')d\vec{x}', \\ V_{quad}(\vec{x}) &= \frac{1}{8\pi|\vec{x}|^3}\int_{\Omega'} n(\vec{x}')(3(\hat{\vec{x}}\cdot\vec{x}') - |\vec{x}'|^2)d\vec{x}', \end{aligned}$$

and in the above expressions, $\hat{\vec{x}} = \vec{x}/|\vec{x}|$.

Note that even the boundary value of Hartree potential is approximated by the multipole expansion, a sufficiently large physical domain is still needed, because the numerical accuracy of multipole expansion can only be guaranteed under the condition $\vec{x} >> \vec{x}'$.

5.3. Linear Mixing Scheme. In the SCF iteration, we use a mixing strategy for updating electron density to guarantee the stability of the iteration. It is known that in each SCF iteration, if the electron density which is obtained in the last iteration is adopted directly to calculate the effective potential v_{eff}, the SCF iteration could be unstable. To fix this problem, many mixing schemes are proposed. Suppose we want to calculate electron density of k-th step of SCF iteration, the idea of these mixing schemes is to use the electron densities of previous

S steps, $n_{P-S+1}(\vec{\mathbf{x}}), n_{P-S+2}(\vec{\mathbf{x}}), \cdots, n_{P-1}(\vec{\mathbf{x}})$, to generate an input electron density. This means we use

$$n_P^{in} = f(n_{P-S+1}(\vec{\mathbf{x}}), n_{P-S+2}(\vec{\mathbf{x}}), \cdots, n_{P-1}(\vec{\mathbf{x}})) \tag{5.2}$$

to calculate v_{eff}, then to find out new wave-functions and the new electron density. There are many mixing schemes such as linear mixing schemes which just linearly mixes the input and output electron density of the last step as the input of the current step, and the GR-Pauly mixing scheme [9]. In the implementation, the simple linear mixing scheme is used:

$$n_P^{in} = \gamma n_{P-1}^{out} + (1-\gamma) n_{P-1}^{in}, \tag{5.3}$$

where $0 < \gamma < 1$. In the simulations, we always use $\gamma = 0.7$.

Now we present the flow chart of the whole algorithm.

Flow Chart of the Algorithm 2

Input: The initial guess $n^{ini}(\vec{\mathbf{x}})$ for the electron density, and ψ_i^{ini} for the wave-function which is used in NSRQMCG method. Let $n^{in}(\vec{\mathbf{x}}) = n^{ini}(\vec{\mathbf{x}})$.

S. 1 : With $n^{in}(\vec{\mathbf{x}})$, get the evaluation of $v_{Hartree}$ by solving (3.6), and the evaluations of v_{ext} and v_{xc} with (2.3) and (2.7) respectively. Then the generalized eigenvalue problem (3.2) is obtained.

S. 2 : Using NSRQMCG method to solve (3.2) to obtain the ψ_i and $n^{out}(\vec{\mathbf{x}})$. Note that the wave-function which obtained in the last iteration is used as the initial guess for NSRQMCG method. If (3.9) is satisfied, goto **Output**; otherwise, goto **S.3**

S. 3 : Using Algorithm 1 to redistribute the grid points, then using (5.3) to get the new $n^{in}(\vec{\mathbf{x}})$, and goto **S. 1**.

Output: the approximation of the ground state density, the total energy, and wave-functions.

6. Numerical Examples. In this section, the convergence of the proposed solver is examined first. Then several atoms and molecules are simulated to show the reliability and effectiveness of our solver. All simulations are implemented on a server with Intel Xeon 2.66 GHz CPU and 64 GB of RAM.

Example 1: Solve the equation

$$-\frac{1}{2}\nabla^2 u - \frac{1}{|\vec{\mathbf{x}}|}u = \lambda u \tag{6.1}$$

with Algorithm 2. The lowest eigenvalue of this equation is -0.5.

The above equation models the hydrogen atom. The physical domain is chosen as $[-10, 10] \times [-10, 10] \times [-10, 10]$. The uniform mesh is refined successively from the coarsest one which contains 63 grid points. The results are demonstrated in Fig. 6.1, which shows that both the convergence rate and the accuracy obtained with the mesh redistribution are superior to that of the fixed uniform mesh. With the refinement of the mesh, the convergence order of solver with mesh redistribution reaches around 1.9, which shows that the eigenvalue of this equation converges at the expected rate

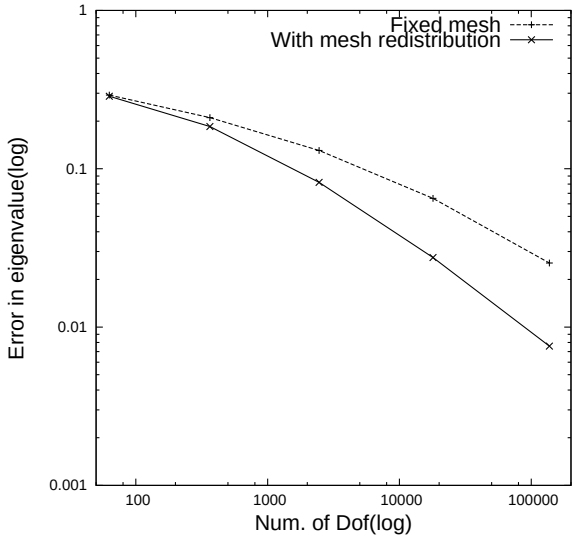

FIG. 6.1. *Convergence rate of the eigenvalue with fixed uniform mesh and with mesh redistribution for Example 1. .*

of the convergence for the linear finite element method.

Example 2: Calculate the total energy of the ground state of the helium and the lithium atoms by using the all-electron computation.

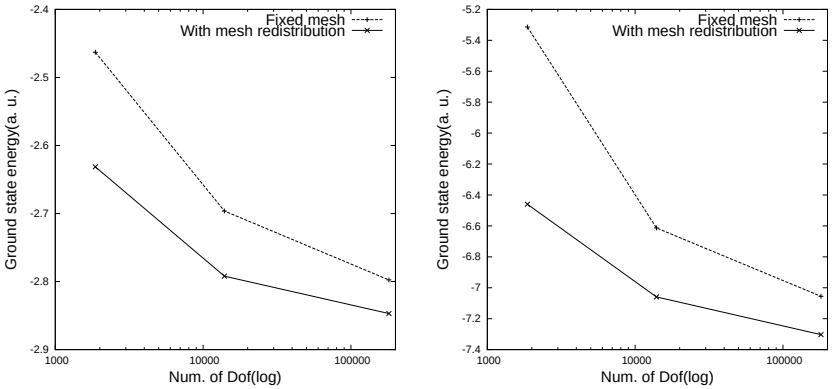

FIG. 6.2. *The energies of the ground state of the helium atom (left) and the lithium atom (right) obtained on successively refined meshes, with fixed mesh and with mesh redistribution, for Example 2.*

For the all-electron calculation, the external potential (2.3) is very singular around the nucleus. To resolve the external potential is very important for obtaining accurate ground state energy. Therefore, a rough radial mesh which is generated by the GMSH[1] is adopted. The coarsest mesh has 1867 nodes, and the mesh is successively refined two times.

The results are shown in Fig. 6.2. It can be observed from the figure that, with the successive refinement of the mesh, both results (with and without mesh redistribution) converges to the reference results, -2.835 for helium atom, and -7.335 for

lithium atom [20], respectively. It also can be observed that with the help of the mesh redistribution technique, the numerical results are much more accurate than that obtained from the fixed mesh with the same number of Dofs. It worth mentioning that for both simulations, the results from using the mesh redistribution with 13924 Dofs are almost the same to that from the fixed mesh with 180539 Dofs, which shows the significant improvement on the solution quality the mesh redistribution technique gives.

Example 3: Calculate the pseudo-atom energies of the lithium and the sodium atoms, and the cohesive energies of these two metal dimers by using the smooth local Evanescent Core pseudopotential [14].

The Evanescent Core pseudopotential has the following form

$$V_{ion}^{I}(\vec{x}, \mathbf{R}_I) = -\frac{Z}{R_c}\left[\frac{1}{y}(1-(1+\beta y)e^{-\alpha y}) - Ae^{-y}\right],$$

where Z is the number of valence electrons and $y = |x - \mathbf{R}_I|/Rc$. The element-dependent constants R_c and α can be found from [14]. In the above formula, the parameters β and A are given by

$$\beta = \frac{\alpha^3 - 2\alpha}{4(\alpha^2 - 1)}, \qquad A = 0.5\alpha^2 - \alpha\beta.$$

To calculate the pseudo-atom energy, the initial uniform mesh is adopted. The results are demonstrated in Table 6.1. It can be seen that our results give reasonable approximations to the reference results. Especially for the sodium atom, with the help of the mesh redistribution, the solver using 17969 Dofs gives more accurate solution than that using fixed uniform mesh with 137313 grid points. Furthermore, the CPU time is also studied for the simulation of the sodium atom, and the results are given in Table 6.2. It takes 207.20 CPU seconds to finish the calculating with the mesh redistribution technique (17969 Dofs), while it is 587.11 CPU seconds for the fixed uniform mesh case (137313 Dofs). It means that to achieve almost the same numerical accuracy, mesh redistribution technique significantly improves the efficiency of the algorithm.

	Dof: 17969	Dof: 137313	
Metal	Mesh redistribution	Fixed mesh	Ref. [26]
Li	-5.95	-5.95	-5.97
Na	-5.19	-5.18	-5.21

TABLE 6.1
The pseudo-atom energy (eV) using the Evanescent Core pseudopotential.

	17969 Dof (mesh redistribution)	137313 Dof (fixed mesh)
CPU time (second)	207.20	587.11

TABLE 6.2
The CPU seconds for the calculating the pseudo-atom energy of the sodium atom.

The cohesive energy of the lithium and the sodium dimers are also calculated, and the results are demonstrated in Table 6.3. A rough radial mesh is used as the initial mesh in the simulation. The bond lengths of these two dimers are given following [26]. From the results, we can see that the cohesive energies of both dimers are approximated reasonably. Again, compared with the fixed mesh case, the mesh redistribution technique improves the solution quality successfully. Fig. 6.3 shows the occupied valence molecular orbital of Na_2, and its corresponding sliced mesh around the orbital. It is obviously observed that the bond region is resolved by mesh grids, which shows that our mesh redistribution technique and the monitor function work very well.

	Mesh redistribution	Fixed Mesh	Ref. [26]
Li_2	-0.49	-0.45	-0.52
Na_2	-0.45	-0.43	-0.46

TABLE 6.3

The cohesive energy (eV) of the lithium and the sodium dimers obtained by using the Evanescent Core pseudopotential.

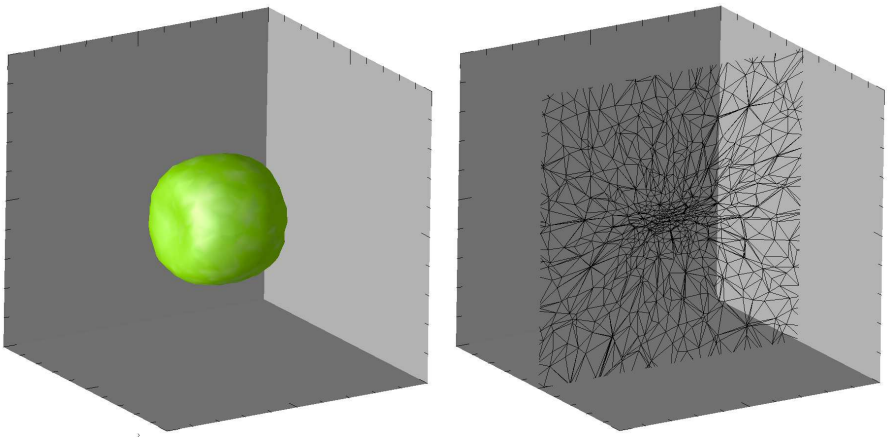

FIG. 6.3. *The occupied valence molecular orbital of Na_2 (top), and the corresponding sliced mesh on the X-Y plane when $Z = 0$(bottom). The mesh redistribution technique is used.*

Example 4: Calculate the ground state of the molecule BeH_2, and demonstrate the relationship of the bond length of Be-H and the ground state energy of the molecule BeH_2.

In the calculation, the smooth local Evanescent Core pseudopotential is adopted for the beryllium atom, while the all-electron is used for two hydrogen atoms. The selection of the parameters in the pseudopotential for the beryllium atom is again followed [14]. Fig. 6.4 shows the relationship between the bond length of Be-H and the total energy of the molecule BeH_2, with and without the mesh redistribution respectively. For the domain, a cube with the size $[-20, 20] \times [-20, 20] \times [-20, 20]$ is chosen. This domain is discretized uniformly by 17969 grid points, which means that the mesh size is around 1.25 a.u.. This mesh size is too large to get a reliable numerical results for the fixed mesh case. This can be demonstrated in Fig. 6.4 (left).

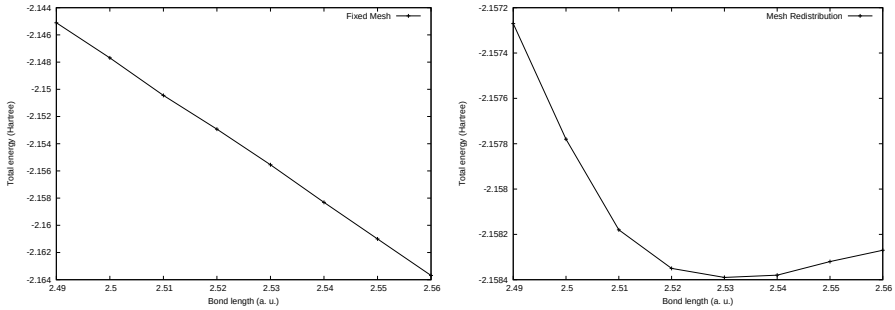

FIG. 6.4. *The relationship between the bond length of Be-H and the total energy of the molecule BeH_2. Left one is the results obtained with fixed uniform mesh, while the right one is the results with the mesh redistribution technique. The reference bond length of Be-H is 2.5204 a.u. [8].*

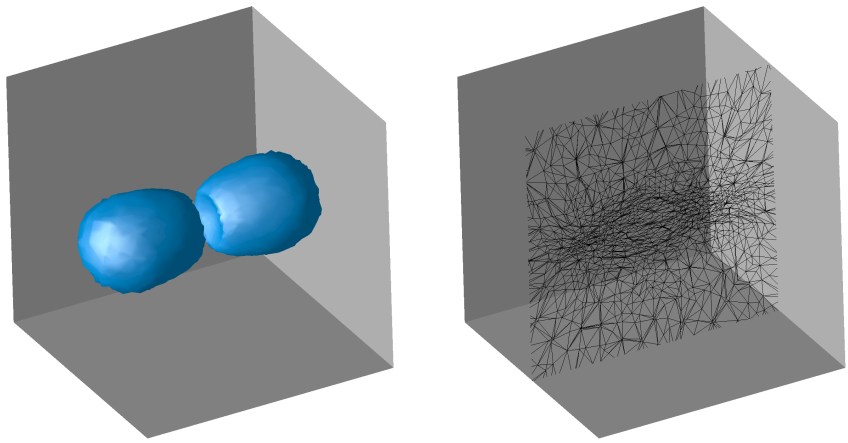

FIG. 6.5. *The isosurface of the electron density of the molecule BeH_2 (top) and the corresponding sliced mesh on the X-Y plane when $Z = 0$(bottom). The mesh redistribution technique is used.*

However, with our mesh redistribution technique, the grid points move towards to the important region like the vicinity of the nucleus and the chemical bond region. Hence the mesh density in these regions becomes large, which helps on obtaining the reliable bond length. From Fig. 6.4 (right), the bond length of Be-H can be read around 2.53 a.u., which agrees with the reference value very well.

The electron density of the molecule BeH_2 obtained with the mesh redistribution is shown in Fig. 6.5. Again, it can be easily observed that two Be-H bond regions are resolved successfully with our mesh redistribution technique.

Example 5: The all-electron simulation of the molecule LiH.

To simulate the molecule LiH, an initial radial mesh is adopted. The initial mesh around two nuclei is demonstrated in Fig. 6.6 (bottom right). From the structure of the molecule LiH, the variation of the electron density is large around the lithium atom and the hydrogen atom, and in the region between the lithium atom and the hydrogen atom. The result in Fig. 6.6 (bottom left) shows that our mesh redistribution technique adaptively resolves these important regions successfully. Fig. 6.6 (top left) shows the electron density in the range $[0.002, 1]$.

The relationship between the bond length of the Li-H and the total energy of the

ground state of the molecule LiH is also studied by using our solver. The result is shown in Fig. 6.6 (top right). It can be observed easily from the figure that the bond length of the Li-H is around 3.01 a.u.. This agrees with the experimental value 3.016 a.u. [18] very well.

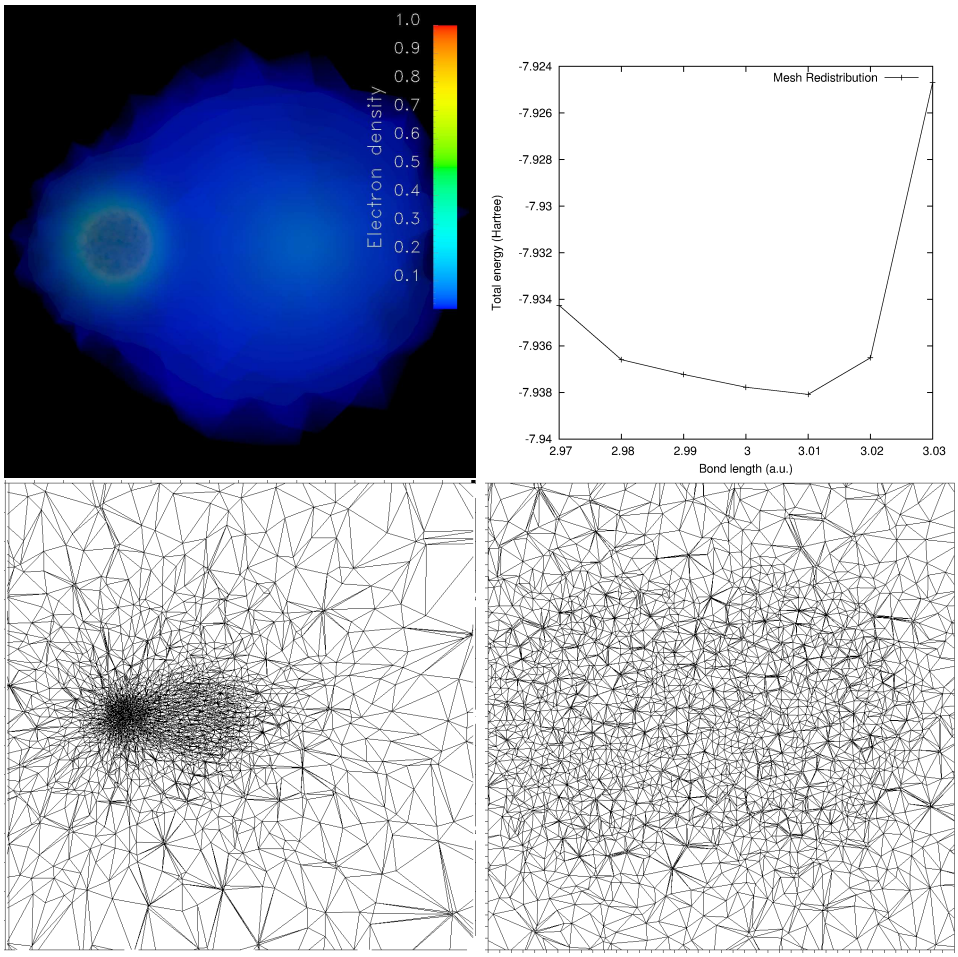

FIG. 6.6. *Top left: the electron density of the molecule LiH; Top right: the relationship between the bond length of the Li-H and the total energy of the ground state of the molecule LiH; Bottom left: the mesh with the mesh redistribution; Bottom right: initial mesh.*

7. Concluding Remarks. We present a finite element method with an adaptive mesh redistribution technique to solve the Kohn-Sham equation in this paper. The solver includes two main iterations. The first one is a SCF iteration which is for the calculations of the self-consistent electron density on the current mesh. In this iteration, the Kohn-Sham equation is discretized by the standard finite element method. In the Hamiltonian operator, the Hartree potential is obtained by solving the Poisson equation with AMG method. The LDA is used to give the exchange-correlation potential. For the external potential, both the all-electron and the local pseudopotential are considered. The generalized eigenvalue problem is solved using the NSRQMCG method.

After the self-consistent electron density is obtained, an adaptive mesh redistribution technique, which is based on the harmonic mapping, is proposed to optimize the mesh quality. The harmonic mapping is obtained iteratively with a given monitor function, which depends on the gradient of the electron density. To guarantee the mesh quality, the monitor function is smoothed by a method which based on the diffusive mechanism. The results show that the mesh quality is significantly improved with our mesh redistribution strategy.

The numerical experiments successfully demonstrate the convergence of our solver. Furthermore, the numerical accuracy and the CPU time are also studied. Results show that with the help of the mesh redistribution technique, both the numerical accuracy of the solution and the efficiency of the algorithm are significantly improved, compared with the fixed mesh case.

In this paper, we mainly focus on the ground state of atoms and molecules. To obtain the accurate ground state energy and the electron density, using a well-designed non-uniform mesh is proven a good idea by [30, 22]. It is expected that that our solver may be extended to time-dependent density functional theory (TDDFT) case. In the TDDFT, if the external potential is not strong, the system can be studied by the perturbative methods . Under this situation, the perturbed system is not far away from the ground state, which means that the well-designed, fixed mesh is still applicable. However, this may not be the case for a very strong external potential. For example, for the simulations of the high-order harmonic generations and the multi-photon ionizations. In these cases, the electronic structure may be dramatically changed. This may cause difficulty for just using a fixed mesh. Consequently, the mesh redistribution technique which is proposed in this paper may help on this issue. With our adaptive technique, the region with the large gradient of the electron density will always be resolved adaptively, which can effectively improve the numerical accuracy of solutions. Some preliminary numerical results have shown the advantages of our solver on the TDDFT calculations. The research findings will be reported on our forthcoming paper.

Acknowledgements. The authors would like to thank Prof. Chao Yang (Lawrence Berkeley National Laboratory) for his helpful comments, suggestions on this work. The research was supported in part by the NSF Focused Research Group grant DMS-0968360. This research of G. Bao was also supported in part by the NSF grants DMS-0908325, CCF-0830161, EAR-0724527, the ONR grant N00014-09-1-0384 and a special research grant from Zhejiang University.

REFERENCES

[1] *http://geuz.org/gmsh/.*
[2] *http://cms.mpi.univie.ac.at/vasp.*
[3] *http://www.abinit.org.*
[4] *http://www.caam.rice.edu/software/ARPACK.*
[5] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quarterly of Applied Mathematics, 9 (1951), pp. 17–29.
[6] A. D. Becke, *Hartree-Fock exchange energy of an inhomogeneous electron gas*, International Journal of Quantum Chemistry, 23 (1983), pp. 1915–1922.
[7] ———, *A new inhomogeneity parameter in density-functional theory*, The Journal of Chemical Physics, 109 (1998), pp. 2092–2098.
[8] P. F. Bernath, A. Shayesteh, K. Tereszchuk, and R. Colin, *The vibration-rotation emmision spectrum of free* beh_2, Science, 297 (2002), p. 1323.
[9] D. R. Bowler and M. J. Gillan, *An efficient and robust technique for achieving self consis-*

tency in electronic structure calculations, Chemical Physics Letters, 325 (2000), pp. 473–476.

[10] E. J. Bylaska, M. Holst, and J. H. Weare, *Adaptive finite element method for solving the exact Kohn-Sham equation of density functional theory*, Journal of Chemical Theory and Computation, 5 (2009), pp. 937–948.

[11] D. M. Ceperley and B. J. Alder, *Ground state of the electron gas by a stochastic method*, Physical Reivew Letters, 45 (1980), pp. 566–569.

[12] A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. Mccormick, G. N. Miranda, and J. W. Ruge, *Robustness and scalability of algebraic multigrid*, SIMA Journal on Scientific Computing, 21 (2000), pp. 1886–1908.

[13] Y. N. Di, R. Li, and T. Tang, *A general moving mesh framework in 3D and its application for simulating the mixture of multi-phase flows*, Communications in Computational Physics, 3 (2008), pp. 582–602.

[14] C. Fiolhais, J.P. Perdew, S. Q. Armster, J.M. Maclaren, and M. Brajczewska, *Dominant density parameters and local pseudopotentials for simple metals*, Physical Review B, 51 (1995), pp. 14001–14011.

[15] G. Gambolati, F. Sartoretto, and P. Florian, *An orthogonal accelerated deflation technique for large symmetric eigenproblems*, Computer Methods in Applied Mechanics and Engineering, 94 (1992), pp. 13–23.

[16] P. Hohenberg and W. Kohn, *Inhomogeneous electron gas*, Physical Review, 136 (1964), pp. B864–B871.

[17] G. H. Hu and P. A. Zegeling, *Simulating finger phenomenon in porous media with a moving finite element method*, Journal of Computational Physics, 230 (2011), pp. 3249–3263.

[18] K. Huber and G. Herzberg, *Molecular Spectra and Molecular Structure. IV. Constants of Diatomic Molecules*, Van nostrand Reinhold: New York, 1979.

[19] W. Kohn and L. J. Sham, *Self-consistent equations including exchange and correlation effects*, Physical Reivew, 140 (1965), pp. A1133–A1138.

[20] S. Kotochigova, Z. H. Levine, E. L. Shirley, M. D. Stiles, and C. W. Clark, *Local-density-functional calculations of the energy of atoms*, Physical Review A, 55 (1997), pp. 191–199.

[21] G. Kresse and J. Furthmüller, *Efficient iterative shcemes for ab initio total-energy calculations using a plane-wave basis set*, Physical Reivew B, 54 (1996), pp. 11169–11186.

[22] L. Lehtovaara, V. Havu, and M. Puska, *All-electron density functional theory and time-dependent density functional theory with high-order finite elements*, Journal of Chemical Physics, 131 (2009), p. 054103.

[23] R. Li, T. Tang, and P. W. Zhang, *Moving mesh methods in multiple dimensions based on harmonic maps*, Journal of Computational Physics, 170 (2001), pp. 562–588.

[24] ———, *A moving mesh finite element algorithm for singular problems in tow and three space dimensions*, Journal of Computational Physics, 177 (2002), pp. 365–393.

[25] S. K. Ma and K. A. Brueckner, *Correlation energy of an electron gas with a slowly varying high density*, Physical Review, 165 (1968), pp. 18–31.

[26] F. Nogueira, C. Fiolhais, J.S. He, and A. Rubio, *Transferability of a local pseudopotential based on solid-state electron density*, Journal of Physics: Condensed Matter, 8 (1996), pp. 287–302.

[27] J. P. Perdew and Y. Wang, *Accurate and simple density functional for the electronic exchange energy: Generalized gradient approximation*, Physical Reivew B, 33 (1986), pp. 8800–8802.

[28] ———, *Accurate and simple analytical representation of the electron-gas correlation energy*, Physical Review B, 45 (1992), pp. 13244–13249.

[29] Y. Saad, J. R. Chelikowsky, and S. M. Shontz, *Numerical methods for electronic structure calculations of materials*, SIAM Review, 52 (2010), pp. 3–54.

[30] P. Suryanarayana, V. Gavini, T. Blesgen, K. Bhattacharya, and M. Ortiz, *Non-periodic finite-element formulation of Kohn-Sham density functional theory*, Journal of the Mechanics and Physics of Solids, 58 (2010), pp. 256–280.

[31] T. Tang, *Moving mesh methods for computational fluid dynamics*, Contemporary Mathematics, 383 (2005).

[32] T. Torsti, T. Eirola, J. Enkovaara, T. Hakala, P. Havu, V. Havu, T. Höynälänmaa, J. Ignatius, M. Lyly, I. Makkonen, T. T. Rantala, J. Ruokolainen, K. Ruotsalainen, E. Räs"anen, H. Saarikoski, and M. J. Puska, *Three real-space discretization techniques in electronic structure calculations*, Physica Status Solidi (b), 243 (2006), pp. 1016–1053.

[33] E. Tsuchida and M. Tsukada, *Adaptive finite-element method for electronic-structure calculations*, Physical Reivew B, 54 (1996), pp. 7602–7605.

[34] H. Y. Wang and R. Li, *Mesh sensitivity for numerical solutions of phase-field equations using r-adaptive finite element methods*, Communications in Computational Physics, 3 (2008), pp. 357–375.

[35] H. Y. Wang, R. Li, and T. Tang, *Efficient computation of dendritic growth with r-adaptive finite element methods*, Journal of Computational Physics, 227 (2008), pp. 5984–6000.

[36] C. Yang, J. C. Meza, B. Lee, and L. W. Wang, *Kssolv-a matlab toolbox for solving the Kohn-Sham equations*, ACM Transactions on Mathematical Software, 36 (2009).

[37] D. E. Zhang, L. H. Shen, A. H. Zhou, and X. G. Gong, *Finite element method for solving Kohn-Sham equations based on self-adaptive tetrahedral mesh*, Physics Letter A, 372 (2008), pp. 5071–5076.