

Towards Translational Invariance of Total Energy with Finite Element Methods for Kohn-Sham Equation

Gang Bao¹, Guanghui Hu^{2,3,*} and Di Liu⁴

¹ Department of Mathematics, Zhejiang University, Hang Zhou, Zhejiang Province, China.

² Department of Mathematics, University of Macau, Macao S.A.R., China.

³ UM Zhuhai Research Institute, Zhuhai, Guangdong Province, China.

⁴ Department of Mathematics, Michigan State University, Michigan, USA.

Received 19 January 2015; Accepted (in revised version) 20 July 2015

Abstract. Numerical oscillation of the total energy can be observed when the Kohn-Sham equation is solved by real-space methods to simulate the translational move of an electronic system. Effectively remove or reduce the unphysical oscillation is crucial not only for the optimization of the geometry of the electronic structure, but also for the study of molecular dynamics. In this paper, we study such unphysical oscillation based on the numerical framework in [G. Bao, G. H. Hu, and D. Liu, *An h-adaptive finite element solver for the calculations of the electronic structures*, Journal of Computational Physics, Volume 231, Issue 14, Pages 4967-4979, 2012], and deliver some numerical methods to constrain such unphysical effect for both pseudopotential and all-electron calculations, including a stabilized cubature strategy for Hamiltonian operator, and an *a posteriori* error estimator of the finite element methods for Kohn-Sham equation. The numerical results demonstrate the effectiveness of our method on restraining unphysical oscillation of the total energies.

AMS subject classifications: 35Q55, 65N30

Key words: Translational invariance, adaptive finite element methods, Kohn-Sham equation, unstructured mesh.

1 Introduction

Density functional theory (DFT) [11] has been becoming one of the most important models for the electronic structure calculations. In DFT, the Kohn-Sham equation [14] plays a crucial role in both theoretical and numerical studies [19].

*Corresponding author. *Email addresses:* baog@zju.edu.cn (G. Bao), garyhu@umac.mo (G. H. Hu), richardl@math.msu.edu (D. Liu)

Lots of research has been done towards the numerical methods for Kohn-Sham equation. For example, the plane-wave expansion methods [16, 32], the real-space methods including the finite difference methods [7, 8], the finite element methods [23, 29], the discontinuous Galerkin methods [18], etc. Among these methods, the plane-wave pseudopotential (PWP) methods have been well developed and widely applied in the computational chemistry community. One of the main advantages of plane-wave expansion methods is that the plane-wave basis functions are independent of the ionic positions. Hence no Pulay corrections [25] is needed on the calculation of the ionic force. Although the real-space methods have a lot of advantages such as the flexibility on handling various boundary conditions and complex practical domains, all these methods suffer from the numerical oscillation of the total energy when simulating the translational or/and rotational move of the electronic systems in the domain.

To explain the numerical oscillation of the total energy, let us consider an electronic structure system with N_{nuc} atoms in the domain. By real-space methods, the Hamiltonian operator is discretized based on a mesh of the computational domain, i.e., the Hamiltonian operator needs to be evaluated pointwisely. It can be imagined that when a different mesh is used, or the molecule moves, the evaluation of the Hamiltonian would not be consistent with the previous one. It is acceptable in the calculations if the inconsistency is sufficiently small. This happens when all terms in the Hamiltonian are smooth and vary gently. However, once there is singularity in the Hamiltonian, and such singularity does not resolved well by the numerical methods, this kind of inconsistency could be large enough to qualitatively affect the numerical result such as the ground-state total energy of the system and the derived intermolecular force from Hellman-Feynman theorem [9]. Unfortunately, the Coulomb interaction between the electron and nucleus, which is called the external potential in the Hamiltonian, is quite singular. For example, the external potential in the Kohn-Sham equation for system we discussed can be given as

$$V_{ext}(\vec{x}) = - \sum_i^{N_{nuc}} \frac{Z_i}{|\vec{x} - \vec{R}_i|}, \quad i = 1, 2, \dots, N_{nuc},$$

where Z_i and \vec{R}_i stand for the i -th nucleus charge and position, respectively. Besides the external potential, the kinetic energy operator could also behave singular in the vicinity of the nucleus because the wavefunctions vary dramatically in the same area. Large numerical error will be introduced if we evaluate these terms inadequately. Furthermore, when we use a fixed uniform mesh to partition the domain, and let the electronic structure do a translational move in the domain, the numerical error from the inadequate discretization will appear periodically with the period the mesh size. Consequently, the total energy of the system obtained from the calculation will also oscillate periodically with the same period, which is called egg-box effect. This is not physical because nothing is changed in the system but the position of the electronic structure, so the total energy should be kept as a constant theoretically. This numerical oscillation is unacceptable since it will affect the evaluation of ground-state total energy, and the calculation of the ionic force

acting on each nucleus. In computation, the ionic force acting on the i -th nucleus along the x -coordinate direction, $F_{i,x}$, is calculated by

$$F_{i,x} = -\frac{\partial E_{tol}}{\partial X_i}, \quad (1.1)$$

where E_{tol} here denotes the total energy of the system. Hence, the break of the translational invariance of the total energy will obviously introduce large error on the computation of the ionic force by the above formula.

The use of the pseudopotential in the simulation can effectively remove the singularity mentioned above. Consequently, the numerical oscillation of the total energy can also be restrained significantly. In a chemical reaction process, one observes that only outer electrons (valence electrons) are chemically active, and inner electrons (core electrons) are relatively stable, in most cases. The idea of pseudopotential is simple. Since the core electrons are quite stable, they are frozen with the nucleus together. In this case, a more gentle effective potential is introduced in the simulation, instead of the singular Coulomb potential. Hence, in a similar simulation environment, the numerical oscillation of the total energy can be reduced significantly by using pseudopotential, compared with the all-electron simulation. To further pursue the translational invariance, a few numerical methods have been introduced such as the Fourier filter method [12], the grid-cell sampling method [10]. The Fourier filter method further removes the large variation introduced by wavefunctions, potential terms in Hamiltonian. This method has been applied in real-space methods [1, 22], and excellent results can be observed. For the grid-cell sampling method, it sufficiently takes advantage of the periodic property of the mesh grids to reduce the numerical error of the total energy by averaging the results on several sample points. Although the effect of the Fourier filter method and grid-cell sampling methods is impressive, both methods need some kind of uniform property of the mesh grids to guarantee the effectiveness, also there is requirement on the regularity of the computational domain. Compared with the uniform mesh, the nonuniform mesh has the potential to deliver more efficient simulations, and is more powerful for the practical problems. However, the good properties such as the periodic property of the mesh grids will lose once the nonuniform mesh is adopted.

To constrain the unphysical oscillation on the nonuniform mesh, one possible way is to raise the order of the cubature formula in the evaluation of the Hamiltonian operator. In this way, the numerical error introduced by numerical integration of singular potentials in Hamiltonian can be reduced effectively. However, since the dimension of the problem we consider is three, the growth of the cubature points is quite fast with the raise of the cubature order. Hence, using the high order cubature formula only for singular terms, or only in the "trouble" regions are reasonable choices to enhance the efficiency. The numerical tests in the paper will show the effectiveness of such strategy.

The pseudopotential method can remove the singularity in the Hamiltonian effectively, and has been widely applied in the simulations. However, more and more evidences [6, 26] show that it is not enough in some simulations if we only consider the

valence electrons. The core electrons also play important role in the simulations. As a result, how to efficiently obtain the translational invariance of the ground-state energy in an all-electron simulation becomes inevitable. In [2], the authors develop an adaptive finite element method to solve the Kohn-Sham equation, which provides a general framework of the implementation of h -adaptive methods to calculate the ground-state of a given electronic system. With the proposed hierarchy geometry tree (HGT) data structure [17], the refinement and coarsening process can be implemented very efficiently. Consequently, this framework provides an effective way to deliver translationally invariant simulations, as long as the error indicator used in the method can resolve the singularity effectively. In [2], only the gradient of the wavefunction is considered to design the indicator for h -adaptive method. Although this indicator works well in calculating the total energy, it can be shown that the mesh grids will be overrefined if an accurate result is desired. In the meantime, since the lack of the error information from the external potential term, its performance on preserving the translational invariance of the total energy is also not good.

In this paper, we will follow [30] to introduce a heuristic *a posteriori* error estimation of our finite element method for Kohn-Sham equation, and numerically study its performance on translational invariance property of the ground-state total energy. Basically, in each element, this local error estimation includes two parts. One is from the jump of the gradient of the wavefunctions on the edge of the element, while the other part is from the local residual. The numerical tests will show that the mesh adaptive method with this error indicator will resolve the above issue effectively.

The rest of this paper is arranged as follows. In next section, the Kohn-Sham equation and its finite element discretization are briefly reviewed. In Section 3, the translational variance phenomenon will be demonstrated, and our methods on resolving this issue will be introduced in detail. The numerical tests will be delivered in Section 4 to show the effectiveness of our methods on preserving the translational invariance of the total energy. Finally, the conclusion and future work will be given in Section 5.

2 Kohn-Sham equation and finite element discretization

In this section, we briefly review Kohn-Sham equation first. Then the h -adaptive finite element method for the Kohn-Sham equation in [2] follow.

2.1 Kohn-Sham equation

The Kohn-Sham equation is given by

$$H\psi_i(\vec{x}) = \epsilon_i\psi_i(\vec{x}), \quad i=1,2,\dots,N, \quad (2.1)$$

where H stands for the Hamiltonian operator of the system, N is the number of the electrons in the system, and ϵ_i and ψ_i denote the i -th eigenstate (wavefunction) and the i -th

eigenenergy, respectively. The Hamiltonian operator H includes four terms for a closed system,

$$H = -\frac{1}{2}\nabla^2 + V_{ext}(\vec{x}) + V_{Hartree}(\vec{x},\rho) + V_{xc}(\vec{x},\rho).$$

In the above expression, the first term is the kinetic energy operator. The second and the third terms describe the classical Coulombic interaction. They have the following analytical expressions

$$V_{ext}(\vec{x}) = -\sum_i^{N_{nuc}} \frac{Z_i}{|\vec{x} - \vec{R}_i|}, \quad \text{and} \quad V_{Hartree}(\vec{x},\rho) = \int \frac{\rho(\vec{x}')}{|\vec{x} - \vec{x}'|} d\vec{x}'. \quad (2.2)$$

where $\rho(\vec{x})$ denotes the charge density, and has the relation with the wavefunctions as the follows

$$\rho(\vec{x}) = \sum_i^N |\psi_i(\vec{x})|^2.$$

The last term in the Hamiltonian operator is the exchange-correlation potential. It explains all the manybody effect in the system, and no analytical expression is available. In the simulations in this paper, the local density approximation (LDA) [24] is employed. For a review of the exchange-correlation potential, we refer to [20] and references therein.

The task of the numerical methods for the Kohn-Sham equation is to solve (2.1) under the constraint

$$\int_{\Omega} \psi_i^*(\vec{x}) \psi_j(\vec{x}) d\vec{x} = \delta_{ij},$$

where ψ_i^* is the conjugate of ψ_i , and δ_{ij} is the Kronecker delta function. After the eigenpairs (ϵ_i, ψ_i) , $i=1,2,\dots,N$ are obtained, the total energy of the system is given by

$$E = \sum_i^N \epsilon_i - U(\rho) + E_{xc}(\rho) - \int V_{xc}(\vec{x},\rho) \rho(\vec{x}) d\vec{x}, \quad (2.3)$$

where $U(\rho(\vec{x})) = \frac{1}{2} \int \int \rho(\vec{x}) \rho(\vec{x}') / |\vec{x} - \vec{x}'| d\vec{x} d\vec{x}'$ is the Hartree electrostatic self-repulsion of the electron density, and E_{xc} denotes the exchange-correlation energy, and has the relation $V_{xc} = \delta E_{xc} / \delta \rho$ to the exchange-correlation potential V_{xc} .

2.2 An h -adaptive finite element method for Kohn-Sham equation

In [2], the authors proposed a framework of using h -adaptive finite element methods to numerically solve the Kohn-Sham equation (2.1). We briefly review the framework here, and present related numerical issues.

Basically, there are two components in the algorithm, i.e., a self-consistent field (SCF) iteration for solving Kohn-Sham equation on a fixed mesh, and a mesh adaption process. In the implementation, on a given mesh, the SCF iteration method is used to solve the Kohn-Sham equation till the stop criterion is satisfied, i.e., the SCF iteration converges

or the maximum iteration number is reached. Then the mesh adaptive method is implemented in terms of the numerical solutions. For detail of the algorithm, we refer to [2, 3]. The following is a brief flowchart of the algorithm.

Algorithm 1: The h -adaptive finite element algorithm for the Kohn-Sham equation.

Data: $\psi, E_{old}, E_{new}, MITER, TOL$

Result: (ϵ, ψ) .

```

1 while  $|E_{new} - E_{old}| > TOL$  and  $iter < MITER$  do
2   Let  $E_{old} = E_{new}$ ;
3   Implement SCF iteration to solve Eq. (2.1) on the current mesh till the stop
   criterion of SCF iteration is reached. Generate  $(\epsilon, \psi)$  and  $E_{new}$ ;
4   Implement mesh adaption process;
5    $iter++$ ;
6 end

```

In the third step in Algorithm 1, the nonlinear Kohn-Sham equation is linearized by the SCF iteration, i.e., the Hamiltonian is calculated with the old ψ and a generalized linear eigenvalue problem is solved to generate the new ψ . For detail of the SCF iteration algorithm in our method, we refer to [2, 3]. In our algorithm, the finite element method is adopted to derive the linear system. To make the description in the later section clearly, we briefly introduce the finite element discretization of Kohn-Sham equation as follows.

Let Ω be the computational domain, and $H^1(\Omega)$ denotes the standard Sobolev space, and $H_0^1 = \{\phi \in H^1(\Omega) : \phi = 0 \text{ on } \partial\Omega\}$. The variational form of the Kohn-Sham equation is as follows: Find $(\epsilon_i, \psi_i) \in \mathbb{R} \times H_0^1(\Omega)$, $i = 1, 2, \dots, N$ such that

$$\int_{\Omega} \left\{ \frac{1}{2} \nabla \psi_i \cdot \nabla \phi + (V_{ext} + V_{Hartree} + V_{xc}) \psi_i \phi \right\} d\vec{x} = \epsilon_i \int_{\Omega} \psi_i \phi d\vec{x}, \quad \forall \phi \in H_0^1(\Omega). \quad (2.4)$$

Now let $\mathcal{T} = \{\mathcal{T}_k, k = 1, 2, 3, \dots, N_{ele}\}$ denotes a tetrahedral partition of the domain Ω , and let $\mathbb{V}_h(\mathcal{T}) \subset H_0^1(\Omega)$ denotes a standard linear finite element space defined on \mathcal{T} . Here N_{ele} denotes the total number of tetrahedral element in the partition. Then the finite element discretization of the Kohn-Sham equation is as follows: Find $(\epsilon_i^h, \psi_i^h) \in \mathbb{R} \times \mathbb{V}_h(\mathcal{T})$, $i = 1, 2, \dots, N$ such that

$$\sum_{\mathcal{T}_k} \int_{\mathcal{T}_k} \left\{ \frac{1}{2} \nabla \psi_i^h \cdot \nabla \phi + (V_{ext} + V_{Hartree} + V_{xc}) \psi_i^h \phi \right\} d\vec{x} = \epsilon_i^h \sum_{\mathcal{T}_k} \int_{\mathcal{T}_k} \psi_i^h \phi d\vec{x}, \quad \forall \phi \in \mathbb{V}_h. \quad (2.5)$$

The analytical form of the Hartree potential $V_{Hartree}$ is given in (2.2). A direction evaluation of such term at each mesh grid results in the $\mathcal{O}(N_{mg}^2)$ computational complexity, which should be avoided in the simulations. Here N_{mg} means the total number of the mesh grids, and also the number of unknowns for a linear finite element discretization.

Instead, the Hartree potential is obtained by solving the Poisson equation

$$\begin{cases} -\nabla^2 V_{Hartree} = 4\pi\rho, & \text{for } x \in \mathbb{R}^3, \\ V_{Hartree} = 0, & \text{for } x \rightarrow \infty. \end{cases} \quad (2.6)$$

In the simulation, the Poisson equation is solved in the domain Ω , and the boundary values are given by the multipole expansion method. To solve the Poisson equation the same finite element space to (2.5) is used for the discretization, and the derived linear system is solved by an algebraic multigrid method. It is worth mentioning that, with solving Poisson equation by the multigrid methods, the computational complexity for evaluating $V_{Hartree}$ is reduced to $\mathcal{O}(N_{mg})$.

After the discretization, we arrive at the following generalized eigenvalue problem

$$A\psi^h = \epsilon^h B\psi^h,$$

and we need to find its N eigenpairs. It is noted that if the spin is not considered, only $N/2$ eigenpairs are needed. There have been a lot of works on solving the above generalized eigenvalue problem, and we choose the locally optimized blocked preconditioned conjugate gradient (LOBPCG) [13] in the simulation. For SCF iteration, it is worth mentioning that although there are several numerical tricks for improving the convergence, the theoretical results for the convergence are still desired. People may refer to [31, 33] and references therein for the related theoretical analysis.

In the fourth step in Algorithm 1, the mesh adaption is implemented. For an effective mesh adaptive method, two questions need to be answered well, i.e., where to locally refine and coarsen the mesh grids, and how to do it efficiently. In our implementation, the hierarchical geometry tree (HGT), which is based on the octree data structure, is used to guarantee the efficiency on mesh management. To answer the first question, an error indicator needs to be assigned for each tetrahedral element in terms of the numerical solutions on the current mesh. Then the mesh grids are locally refined or coarsened according to a given tolerance. In [2], the gradient of each wavefunction is used to design the indicator. In the vicinity of each nucleus, the variation of each wavefunction is large, hence such area can be effectively resolved with the indicator. In the meantime, when the all-electron calculations are considered, the external potential V_{ext} is also very singular in the same area. Consequently, this kind of indicator works very well in both all-electron and pseudopotential calculations, see [2]. However, when the all-electron calculations are considered, it is not adequate to resolve the singularity introduced by V_{ext} by using the gradient of the wavefunction. Although a smaller tolerance in the mesh adaption could improve the numerical solutions, it will result in the overrefinement of the mesh grids. Hence, a finer error indicator becomes necessary, and the error information from the external potential needs to be considered. In next section, a residual based a posteriori error estimation will be introduced to cover this issue.

Remark 2.1. The above numerical framework has been adopted in the previous work [2–5]. On the numerical convergence of the methods, we refer to those papers for detail. In the following, we focus on the eggbox effect to improve this numerical framework.

3 Translational variance of the total energy and restrained methods

In this section, the variance of the total energy caused by the translational move of the electronic structure will be demonstrated first. Then numerical methods will be introduced to restrain such phenomenon.

3.1 Eggbox effect

The eggbox effect is known as an unphysical oscillation when the real-space methods are used for simulating the translational move of an electronic system in the given domain with the uniform mesh. When the electronic structure does a translational move in the domain, the ground state total energy of the electronic structure should theoretically be a constant. However, it is not the case when the real-space methods are used. This can be seen from the result of Example 3.1 below.

Example 3.1. Find the ground state of a Helium atom system. The domain is $[-10,10] \times [-10,10] \times [-10,10]$, the initial ionic position is the origin $(0,0,0)$. A uniform tetrahedral mesh is used, the total number of the mesh grids is 17969, and the mesh size along the x -axis is around 1.2. After the ground state is reached at the current position, move the nucleus along the positive direction of the x axis with the length 0.03, and repeat. Atomic unit is used.

The finite element framework proposed in [2] is used without the mesh adaptive process in Example 3.1. The cross section of the mesh in the x - y plane is shown in Fig. 1. In

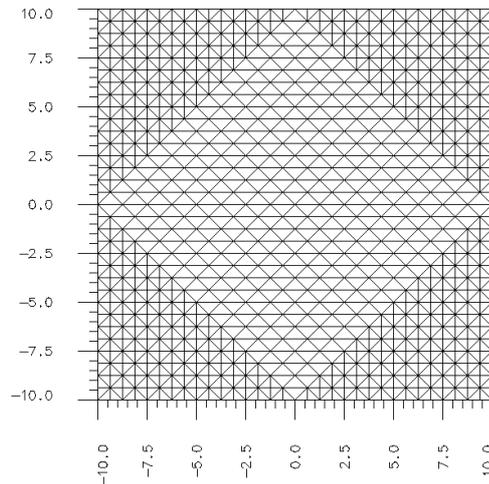


Figure 1: The cross section of the mesh on the x - y plane.

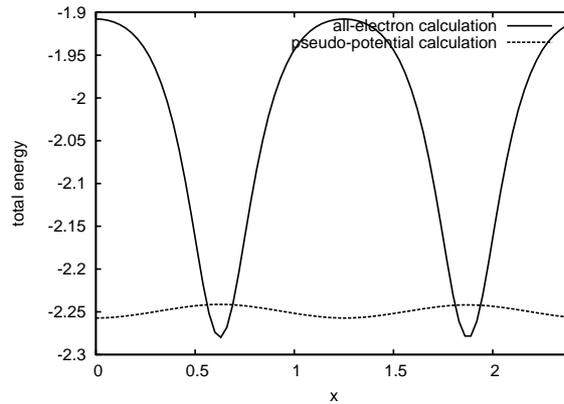


Figure 2: The curve (solid line for all-electron calculation, and dashed line for pseudopotential calculation) of the total energy versus the position of the nucleus of the Helium atom.

the simulation, the nucleus of the Helium atom moves to the right 0.03 au after the current ground state energy is obtained, and the curve of ionic position versus total energy is displayed in Fig. 2 (solid line).

Two observations can be made from the result. The first one is that the error of our numerical result is quite large, compared with the reference total energy (-2.83 au from [15]). The second one is the periodic oscillation of the total energy with the move of the nucleus, and the period is around 1.2 au , which is consistent with the edge length of the element along the x -axis.

The inadequate discretization of the Hamiltonian operator explains the inaccurate total energy in the above example. Figs. 3 and 4 show this clearly. As we can see from two figures, the mesh size 1.2 au is far from enough to resolve the variation of the all-electron wavefunction (Fig. 3, solid line) and full external potential (Fig. 4, solid line), which causes the large numerical error in evaluating total energy of the Helium atom. To explain the periodic oscillation of the total energy, let us study Fig. 5 in detail. The one dimensional function $f(x) = e^{-|x-a|}$ is shown in the figure with different a . The functional varies very fast around $x = a$, while decays when x is away from a . This is quite similar to the behavior of the external potential in the Hamiltonian. To form the stiff matrix in finite element methods, the numerical integration needs to be done in each element. If we use one-point Gauss quadrature to do the numerical integration, the point $x = 0.3$ would be the quadrature point for the interval $[0, 0.6]$. This is a fair choice for evaluating the function $f(x)$ when $a = 0$ and 0.6 . However, the integral will be highly overestimated for $a = 0.3$ case, since the function value in this case is given by the lower black circle point in the figure. With the increment of a , this kind of overestimation will appear periodically. This explains the eggbox effect shown in Fig. 2.

Based on the above discussion, both the inaccurate total energy and the nonphysical oscillation of the total energy are caused by the singularity introduced in the Hamiltonian operator. Hence, from the algorithm efficiency point of view, how to effectively remove

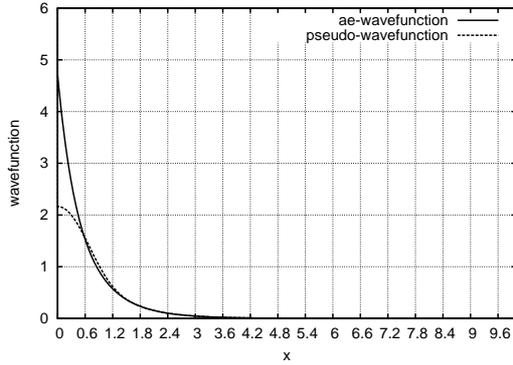


Figure 3: The all-electron wavefunction (solid line) and pseudo-wavefunction (dashed line) of a Helium atom. The results are obtained from APE [21].

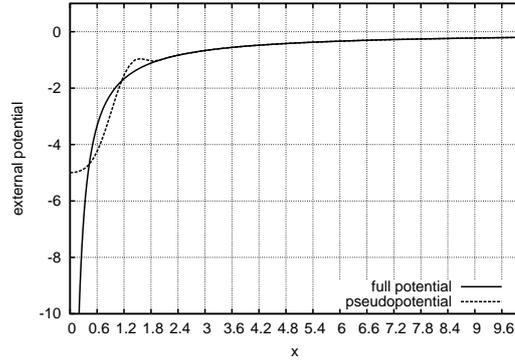


Figure 4: The all-electron potential $-2/|x|$ (solid line) and the pseudopotential (dashed line) of a Helium atom. The results are obtained from APE [21].

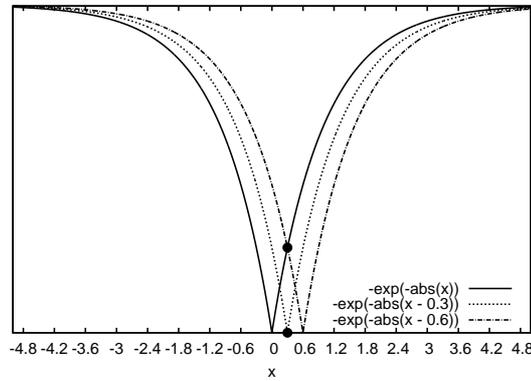


Figure 5: The function $e^{-|x-a|}$ for $a=0$ (left curve), $a=0.3$ (middle curve), and $a=0.6$ (right curve). The black circle points denote the value of each function at $x=0.3$.

the singularity becomes the main issue on restraining the translational oscillation of the total energy, and the use of the pseudopotential is an attractive answer towards this direction.

In next subsection, we first introduce the utilize of the pseudopotential and its performance on keeping the translational invariance of the total energy. Then, we will introduce a nonuniform cubature strategy for further restraining the translational oscillation of the total energy when some traditional methods such as Fourier filter method and grid cell sampling method, are no longer applicable.

3.2 Pseudopotential

An effective way to remove the singularity introduced by the external potential V_{ext} is to use the pseudopotential. The idea of pseudopotential is simple. Based on the observation

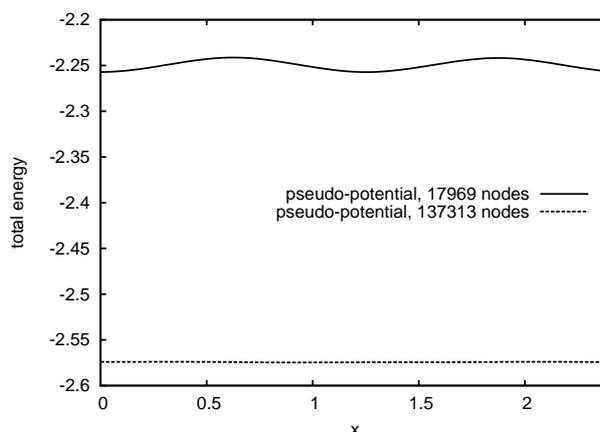


Figure 6: The curve of the total energy versus the position of the nucleus of the Helium atom. The results are obtained with pseudo-potential calculation. The solid line and dashed line show the results with 17969, and 137313 nodes in the mesh, respectively.

that only valence electrons are active in the chemical reaction in most cases, the core electrons can be frozen, being considered together with the nucleus as rigid non-polarizable ion cores. In this way, the core electrons are removed from the calculation, and the singular real Coulomb potential is replaced by a gentle pseudopotential $V_{pseudo}(\vec{x})$. Fig. 4 (dashed line) shows the effect of pseudopotential for a Helium atom. The pseudopotential is generated by Troullier-Martins norm-conserving method, and is provided by APE [21]. The cutoff radius we used in the simulation is $2 au$. Compared with the real Coulomb potential $-2/x$, the pseudopotential in Fig. 4 (dashed line) varies much more gently, which means that the variation of the external potential can be resolved well with a coarse mesh. With the use of the pseudopotential, the wavefunction also varies more gently than that from all-electron case, see Fig. 2 (dashed line) which shows the result for Example 3.1 with the use of the pseudopotential. It is noted that with the pseudopotential, the reference result for the total energy of the Helium atom is around $-2.7016 au$, which is obtained by using our mesh adaptive method with the tolerance $5e-05$.

From Fig. 2 (dashed line), it can be observed that the variation of the curve becomes much smaller than that of the all-electron calculation (solid line). However, the periodic behavior of the curve is still there, with the same period to that of the all-electron result. In the meantime, the numerical accuracy is still unacceptable. All these drawbacks of the numerical results can be overcome effectively by reducing the mesh size. For example, once we refine the mesh globally (137313 mesh grids), and redo the calculation, the numerical result is given in Fig. 6 (dashed line). It can be observed obviously that the numerical accuracy is improved dramatically, while the numerical oscillation of the total energy is also restrained significantly.

To further improve the numerical accuracy, a straightforward way is to keep refining the mesh. However, it will cause the implementation efficiency issue since refining the

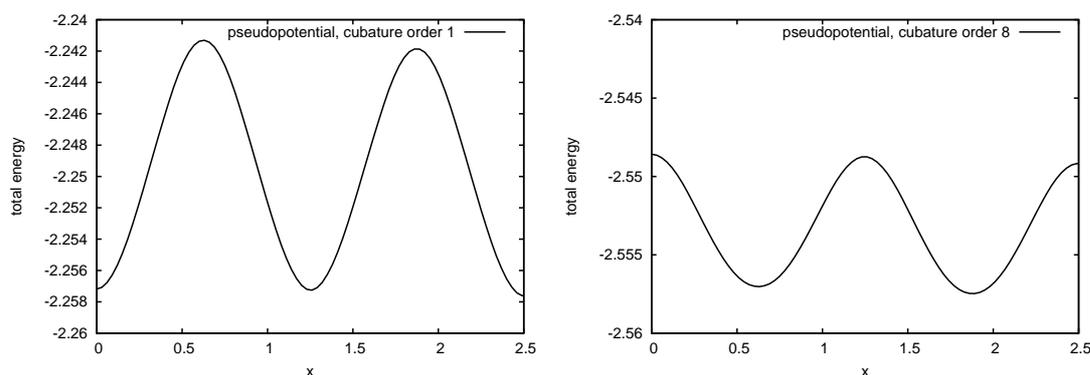


Figure 7: The result shown in left figure is same to Fig. 2 (dashed line), while the result shown in the right figure is from the same calculation except for the cubature order for the external potential.

mesh globally one more time will generate a new mesh with over one million nodes for the current case. It is noted that the inaccuracy in evaluating the integral with respect to the Hamiltonian operator causes the inaccuracy of the total energy and the numerical oscillation. Hence, raising the order of the Gauss cubature for such integral could be a reasonable alternative. To show the effect, we redo the calculation of Example 3.1 with the pseudopotential, and only raise the cubature order for the integral the external potential to 8 and keep other configurations unchanged. The reason that we only raise the cubature order for the external potential term is that, compared with the other potential terms, the variation of the external potential is larger. The result is shown in Fig. 7. It is shown obviously that with the raise of the cubature order for the external potential, both the numerical accuracy and the translational invariance of the total energy are improved, significantly and simultaneously. It is noted that with the uniform mesh, there are other numerical methods to restrain the eggbox effect such as the grid-cell sampling, Fourier filter. The excellent effect of those methods can be observed from [12] and [10]. However, it is quite challenging to use those methods for nonuniform mesh cases, and raising the cubature order could be an effective choice then.

Although pseudopotential method can remove the singularity introduced by the external potential well, the all-electron calculation is needed towards the ab-initio electronic structure calculations. To resolve very singular external potential in the all-electron calculations, a nonuniform mesh can be designed according to the structure of the given molecule in advance, see [27, 28]. However, when we want to study the geometry optimization, or the dynamics of a given molecule [4], the nucleus might move in the domain, and a well designed fixed mesh might not a good choice since the mesh might need to be regenerated to keep the accuracy, and it could be very CPU time demanding. The mesh adaptive methods can overcome the above issue effectively. Ideally, the mesh adaptive methods can always resolve the singular region with sufficient mesh grids dynamically, while keep the mesh coarse in the region with gentle solutions. In [2, 3, 5], the authors introduced two kinds of mesh adaptive methods. In the following section, we will fol-

low [2] to introduce the framework of the h -adaptive methods, and introduce a residual based *a posteriori* error estimator for the mesh adaptive process.

3.3 Mesh adaptive methods

In [2], the authors follow classical process

$$\cdots \rightarrow \text{Solve} \rightarrow \text{Estimate and Mark} \rightarrow \text{Refine} \rightarrow \text{Update} \rightarrow \cdots$$

to design the mesh adaptive algorithm. According to the above process, the Kohn-Sham equation is solved first. Then an indicator is designed in each element based on the numerical solutions. With a mark strategy, each element in the current mesh is marked to refine or coarsen. Finally, a new mesh is generated after implementing the mesh adaptation, and the numerical solutions are updated from the current mesh to the new mesh.

As we discussed in Section 2, although the indicator in [2] works well in calculating the ground state total energy, the overrefinement of the mesh grids can not be avoided when the all-electron calculation is considered. This can be explained by that the singularity introduced from the external potential is much stronger than that from the wavefunctions. Hence, to improve the performance of the mesh adaptive methods, the error information from the external potential needs to be considered, and the error indicator in this paper is obtained from the following *a posteriori* error estimation.

It is noted that since the Hamiltonian operator H in (2.1) depends on the electron density $\rho(\vec{x})$, while the electron density $\rho(\vec{x})$ depends on the unknown $\psi_i(\vec{x})$, $i = 1, 2, \dots, N$, Eq. (2.1) is actually nonlinear. It is difficult to analyze the numerical error directly from this nonlinear equation. Fortunately, the method we used to solve the Kohn-Sham equation is the so-called self-consistent field (SCF) iteration, i.e., the electron density obtained from the previous iteration is used to update the Hamiltonian, then the linearized eigenvalue problem is solved to update the wavefunctions and electron density. It means that a linear generalized eigenvalue problem is solved in each iteration, which also means that we can focus on the numerical analysis on the following system

$$\begin{cases} \left(-\frac{1}{2}\nabla^2 + V_{eff}(\vec{x})\right)\psi(\vec{x}) = \epsilon\psi(\vec{x}), & \text{for } \vec{x} \in \Omega, \\ \psi(\vec{x}) = 0, & \text{for } \vec{x} \in \partial\Omega, \end{cases} \quad (3.1)$$

where $V_{eff}(\vec{x}) = V_{ext}(\vec{x}) + V_{Hartree}(\vec{x}) + V_{xc}(\vec{x})$ and only depends on the spatial variable \vec{x} .

Based on Section 2, the variation form of (3.1) is as follows: Find $(\epsilon_i, \psi_i) \in \mathbb{R} \times H_0^1(\Omega)$, $i = 1, 2, \dots, N$ such that

$$\int_{\Omega} \left\{ \frac{1}{2} \nabla \psi_i \cdot \nabla \phi + V_{eff} \psi_i \phi \right\} d\vec{x} = \epsilon_i \int_{\Omega} \psi \phi d\vec{x}, \quad \forall \phi \in H_0^1(\Omega), \quad (3.2)$$

and the finite element discretization of (3.1) is as follows: Find $(\epsilon_i^h, \psi_i^h) \in \mathbb{R} \times \mathbb{V}_h(\mathcal{T})$, $i = 1, 2, 3, \dots, N$ such that

$$\sum_{\mathcal{T}_k} \int_{\mathcal{T}_k} \left\{ \frac{1}{2} \nabla \psi_i^h \cdot \nabla \phi + V_{eff} \psi_i^h \phi \right\} d\vec{x} = \epsilon_i^h \sum_{\mathcal{T}_k} \int_{\mathcal{T}_k} \psi_i^h \phi d\vec{x}, \quad \forall \phi \in \mathbb{V}_h. \quad (3.3)$$

The task of the *a posteriori* error estimation is to develop a computable quantity from (ϵ_i^h, ψ_i^h) , $i = 1, 2, \dots, N$, while the equivalence of this quantity and the distances between ψ_i and ψ_i^h , and ϵ_i and ϵ_i^h can also be proved. This can be done following the results in [30] by assuming that $V_{eff}(\vec{x})$ is strictly positive, which is unfortunately not satisfied by V_{eff} from Kohn-Sham equation. However, it is noted that the V_{eff} is bounded when the pseudopotential is employed in the simulations, and that the eigenvectors of (3.2) will not change if a real number is added in the Hamiltonian. It means that a sufficiently large constant M can be added on V_{eff} to make the generated $\tilde{V}_{eff} = V_{eff} + M$ be strictly positive, while the eigenpairs of the revised system become $(\epsilon + M, \psi)$. Hence, we could assume that V_{eff} is strictly positive here, and follow [30] to give the *a posteriori* error indicator in each element

$$\eta_{\mathcal{T}_k}^2 = \sum_i^N \left\{ \sum_e h_e \|\mathcal{J}_e(\psi_i^h)\|_{2,e}^2 + h_{\mathcal{T}_k}^2 \|\mathcal{R}_{\mathcal{T}_k}(\psi_i^h)\|_{2,\mathcal{T}_k}^2 \right\}. \quad (3.4)$$

There are two terms in the above estimator. In the first term, $\mathcal{J}_e(\psi_i^h) = \frac{1}{2}(\nabla \psi_i^h|_{\mathcal{T}_k} \cdot \vec{n}_{e,kj} + \nabla \psi_i^h|_{\mathcal{T}_j} \cdot \vec{n}_{e,jk})$ denotes the jump of the flux across the face e which is the common face of two tetrahedron elements \mathcal{T}_k and \mathcal{T}_j . $\vec{n}_{e,kj}$ and $\vec{n}_{e,jk}$ stand for the unit outward normals on the face e of \mathcal{T}_k and \mathcal{T}_j , respectively. The parameter h_e stand for the diameters of the face e . In the second one, $\mathcal{R}_{\mathcal{T}_k}(\psi_i^h)$ denotes the element residual of the equation, and has the following form

$$\mathcal{R}_{\mathcal{T}_k}(\psi_i^h) = \left(\epsilon_i \psi_i^h + \frac{1}{2} \nabla^2 \psi_i^h - V_{eff} \psi_i^h \right)_{\mathcal{T}_k},$$

where $h_{\mathcal{T}_k}$ stands for the diameter of the element \mathcal{T}_k . For the pseudopotential calculations, the effectiveness and reliability of the error indicator (3.4) can be proven theoretically, please refer to [30] for detail. For the all-electron calculations, since the boundedness of the Hamiltonian can not be obtained, the theoretical analysis in [30] can no longer be applied. However, the error indicator (3.4) still works very well for all-electron calculations, as shown by the numerical tests later.

In (3.4), the jump function $\mathcal{J}_e(\psi)$ describes the amplitude of the variation of the gradient of the wavefunction on each surface of a given tetrahedron element. Hence, as an indicator, its behavior is similar to the one in [2]. If we only use the first part in (3.4) in the simulation, it can be imagined that the numerical accuracy for the ground-state energy can still be improved significantly. For example, by using the h -adaptive framework in [2] and the indicator only with $\mathcal{J}_e(\psi)$ part, and with the tolerance $2.0e-04$, the numerical results for Example 3.1 are given by Figs. 8 and 9. Here the all-electron calculation

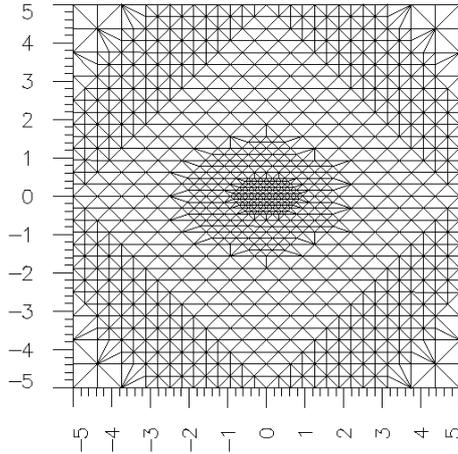


Figure 8: The cross section of the mesh on the x - y plane with the h -adaptive method.

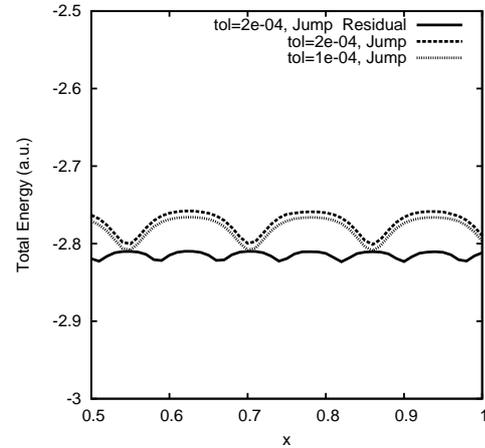


Figure 9: The curve of the total energy versus the position of the nucleus of the Helium atom. The h -Adaptive method is used.

is used. In Fig. 8, it shows the mesh obtained with Algorithm 1 when the nucleus at its original position. With the jump function $\mathcal{J}_e(\psi)$, the region with the large gradient jumps of the wavefunctions is resolved successfully. Consequently, the numerical accuracy of the ground-state energy in Fig. 9 (the upper dashed line) is much better than that in Fig. 2, and the amplitude of the nonphysical oscillation of the total energy with the move of the Helium atom is also significantly reduced.

To further improve the numerical behavior of the method, a straightforward way is to reduce the tolerance used in the mesh adaptive procedure, since the smaller tolerance gives more accurate description of the wavefunctions. However, with only jump function $\mathcal{J}_e(\psi)$ in the indicator, reducing the tolerance helps little on improving the numerical results. For instance, if we redo the Example 3.1 with the h -adaptive method with the tolerance $1.0e-04$, a slightly better result can be observed from Fig. 9 (lower dashed line). However, 40% more mesh grids are introduced in the simulation, which also increases the CPU time. Furthermore, almost nothing is improved on the nonphysical oscillation. The reason for this phenomenon is that the error information from evaluating the external potential V_{ext} is totally missed in the indicator.

In (3.4), the local residual $\mathcal{R}_{\mathcal{T}_k}(\psi)$ supply us a quite good estimation for the error introduced by V_{ext} . With a good approximation of the wavefunction, the error in the evaluation of the integral $\int V_{ext}\psi\phi d\vec{x}$ dominates the term $\|\mathcal{R}(\psi)\|_{\mathcal{T}_k}$. Hence, it can be imagined that the numerical results would be improved effectively with adding the local residual term in the error indicator. In Fig. 9, the result (solid line) is obtained with the tolerance $2e-04$ and the indicator (3.4). It can be observed clearly that the numerical result of the total energy is much closer to the reference value ($-2.83 au$). In the meantime, the amplitude of the nonphysical oscillation is also reduced significantly. In this simulation, only 20% more mesh grids are introduced in the simulation, compared with that in Fig. 8.

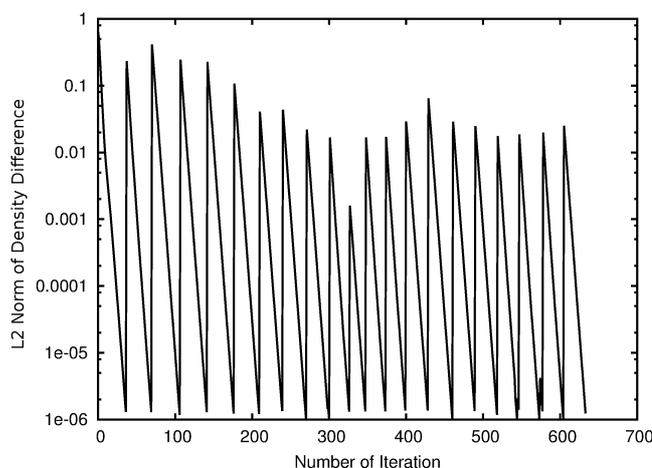


Figure 10: Convergence history of the simulation with h -adaptive methods for Helium atom.

It is worth mentioning that the Hierarchical Geometry Tree (HGT) [2] structure is used in our h -adaptive methods, which allows an efficient implementation of the interpolation of the solutions after an adaptive refinement of the mesh. In the meantime, the quality interpolation provides a quality starting point for the next SCF iteration, which makes the convergence of each SCF iteration quite stable. This can obviously be observed from Fig. 10. In the figure, the SCF iteration history with Algorithm 1 for Helium all-electron calculation is shown. In Algorithm 1, after the SCF convergent solution is obtained with the current mesh, the mesh adaptive process in Algorithm 1 will generate a new mesh if the stop criterion for the loop is not reached, and the solution will be updated from the current mesh to the new mesh. Then a new SCF iteration will be implemented. As we can see, when a new SCF iteration is started, the initial density difference is large. This can be explained by the interpolation error. However, the solution obtained from the interpolation still serve as a very good initial solution for the new SCF iteration, which makes the convergence quite stable, i.e., around 30 iteration steps are needed in each SCF iteration to make the L2 norm of the density difference below $1.0e-06$.

4 Numerical results

In this section, we present several numerical examples to demonstrate the ability of our methods on preserving the translational and rotational invariance of the total energy. The numerical results also show that the bond length of a given molecule, and ionic force acting on the nucleus can be calculated accurately with our methods.

All numerical experiments are implemented using the library AFEABIC [2, 3] on a Dell precision T5600 workstation (Intel(R) Xeon(R) CPU E5-2630 v2 @ 2.60GHz, 64G memory). OpenMP is used to accelerate the simulations.

4.1 A dihydrogen molecule with all-electron calculation

The example in this subsection is to predict the bond length of a H_2 molecule. All-electron calculation is used for this simulation. The size of the computational domain is $[-80,80]^3$, and the initial positions for two hydrogen atoms are $[-0.3,0,0]$ and $[0.3,0,0]$, respectively. The simulation process is as follows. After the ground state is reached at the current configuration, the distance between two nuclei will be enlarged by 0.04 au , i.e., the left nucleus will move towards left by 0.02 au , and the right nucleus will move towards right by 0.02 au . Fig. 11 shows the results of this simulation.

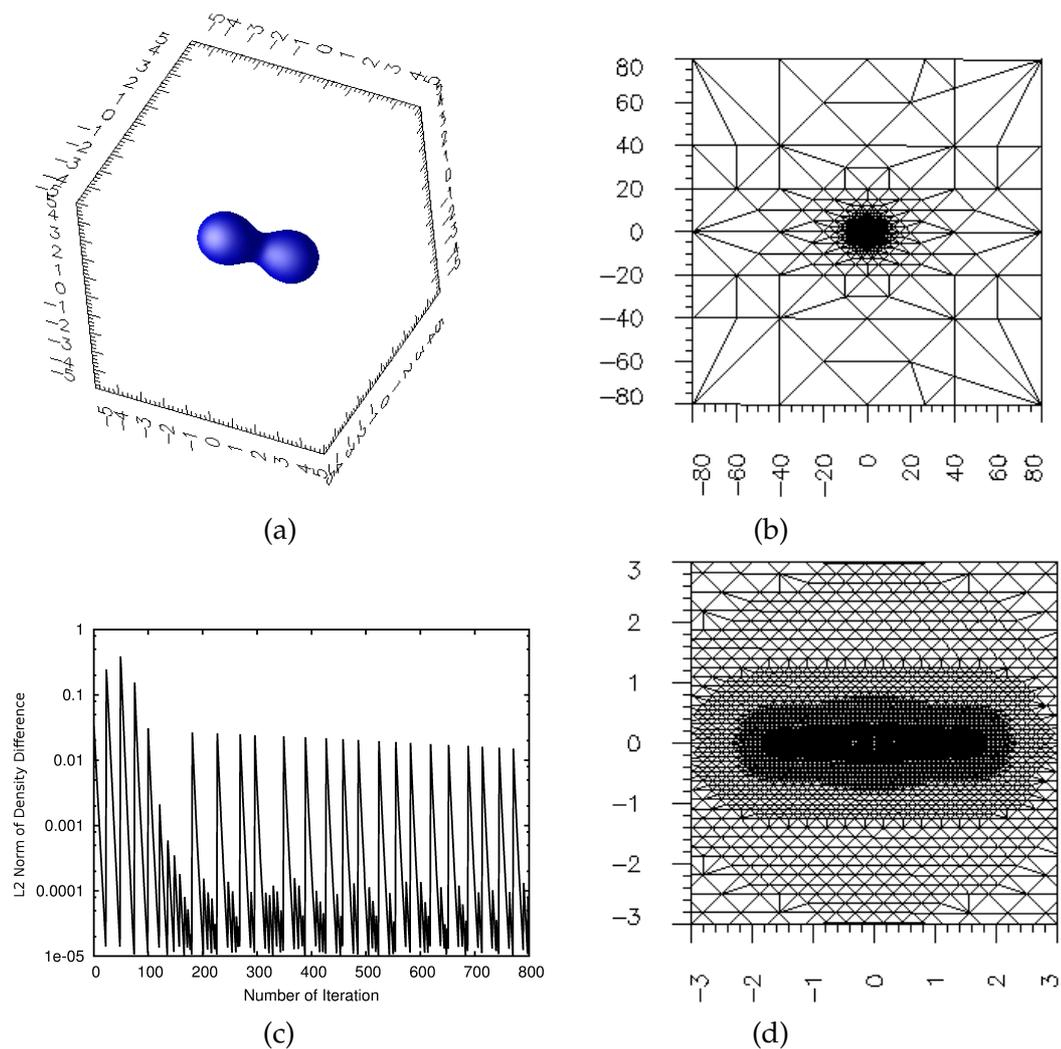


Figure 11: (a): the isosurface of H_2 ; (b): the mesh cross section on $x-y$ plane; (c): the convergence history; (d): local mesh of the one in (b).

Fig. 11(a) shows the isosurface of the H_2 molecule. With our mesh adaptive method, the isosurface is resolved smoothly. Fig. 11(b) and (d) shows that the error indicator (3.4) successfully and dynamically resolve the trouble region with the move of the nuclei, while keeps the mesh grids coarse in the region away from the trouble region. Fig. 11(c) shows the convergence history of the SCF iteration in the simulation. From the figure we can see that the SCF iteration converges very smoothly in each mesh generated by the mesh adaptive methods. It can also be observed that quality solution interpolation significantly reduces the initial L2 norm of the density difference, and also accelerates the convergence. Although around 10 times mesh adaptive refinement, and 18 SCF iteration steps by average in each mesh, are needed for implementing Algorithm 1 when the nuclei of H_2 are at their original positions, only around 5 times mesh adaptive refinement, and around 9 SCF iteration steps in each mesh, are needed after the nuclei move.

Fig. 12 shows the variation of the total energy and ionic force with respect to the change of the bond length of H_2 molecule. The ionic force here is calculated by using (1.1), i.e., the negative of the partial derivative of the total energy with respect to the ionic position. The numerical differentiation is employed for the calculation. Two observations can be made from the figure. First, with the help of the error indicator (3.4), our mesh adaptive method generates a very smooth curve for the variation of the total energy, no high frequency numerical oscillation can be observed from the curve. Our numerical result shows that the bond length of the H_2 molecule is around 1.420 au , which agrees with the experimental value (1.401 au) very well. Second, the ionic force acting on the nucleus is zero around 1.420 au , and the directions of the force are opposite when the bond length becomes small and large. Furthermore, the curve for the ionic force is monotone. All these observations agree with the theory very well, which shows that our mesh adaptive method can serve the force calculation very well.

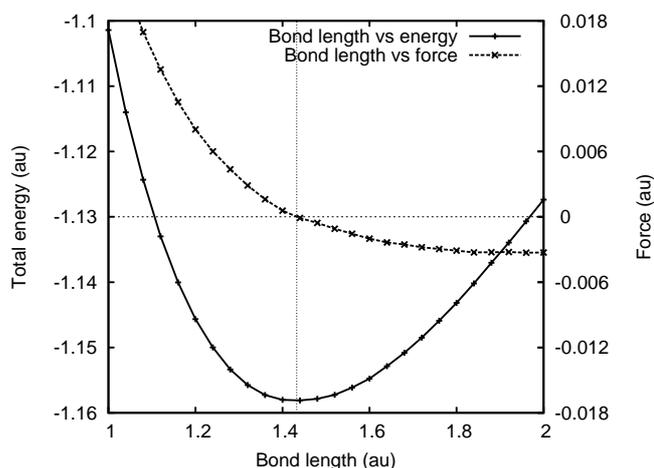


Figure 12: The curve of the bond length vs total energy (solid line), and the curve of the bond length vs ionic force (dashed line).

4.2 A body-centered cubic unit cell of sodium with pseudopotential calculation

In this example, the pseudopotential calculation for a sodium body-centered cubic (BCC) unit cell will be implemented. The purpose of this example is twofold. First, the lattice constant will be predicted by our method to compare with the experimental one. Second, the rotational invariance of the total energy of the sodium BCC unit will be demonstrated.

The first test in this example is for the prediction of the lattice constant of the BCC unit cell. The structure of the BCC unit cell of the sodium is shown in Fig. 13 (left one), where a there denotes the lattice constant. In the simulation, a cube with the dimension $[-80,80]^3$ is adopted as the computational domain, and the nucleus in the center of the BCC unit cell is put at the origin of the coordinate system. The simulation process is as follows. The initial lattice constant a is equal to 3 au . After the system is solved self-consistently, each nucleus except for the centered one moves 0.173 au along the line which connects the origin and the ionic position, and the direction is the one which is away from the origin. Then the Kohn-Sham equation is solved in the new configuration. We repeat the above process to get the results shown in Fig. 13 (right one). In the figure, we can see that the lattice constant is around 6.600 au . This value is far from the experimental one, i.e., 8.108 au . This is because only one BCC unit cell is considered in our simulation. By increasing the number of the BCC unit cell, the prediction should become more accurate, see [28]. Our point in this example is for the nonoscillatory curve of the total energy vs the lattice constant, and for the derived quality curve of the ionic force vs the lattice constant.

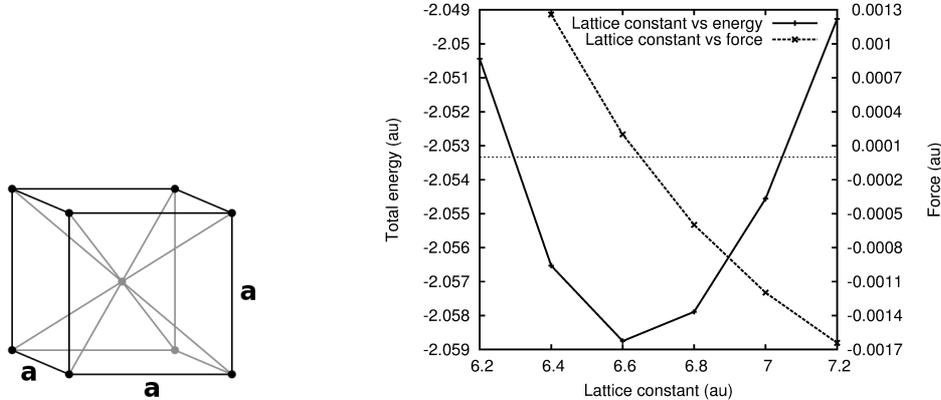


Figure 13: Left: a body-centered unit cell, the nuclei locate at eight corners and the center of the cube. a denotes the lattice constant. Right: the curve of the bond length vs total energy (solid line), and the curve of the bond length vs ionic force (dashed line)

The second test in this example will show the ability of our method on preserving the total energy when the BCC unit cell does a rotational move. Similar to the case of translational move, since nothing is changed but the positions of those nuclei, the total energy should be a constant theoretically. To show that our numerical method can preserve this

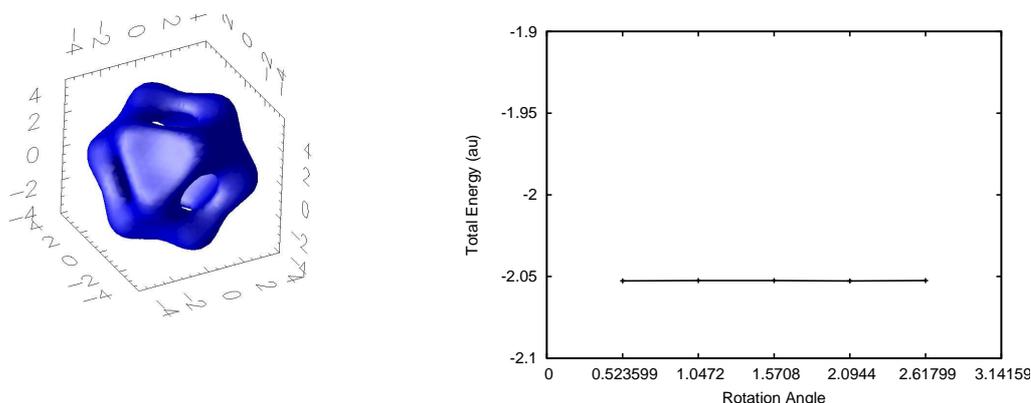


Figure 14: Left: the isosurface of a BCC unit cell of sodium from the initial configuration. Right: the total energy of BCC unit cell of sodium versus the rotation angle.

property well, we rotate the BCC unit cell by $\pi/6$ each time around the z -axis, then calculate the ground-state total energy. Fig. 14 shows the results of our simulation where the rotational invariance of the total energy can be observed obviously. In this simulation, the lattice constant 6.635 au from our above simulation is adopted.

5 Conclusion

In this paper, we numerically study the translational variance of the total energy of the Kohn-Sham equation when the finite element method is used. By using the pseudopotential, the singularity introduced by the external potential can be reduced effectively, while the eggbox effect can also be restrained significantly. We introduce a stabilized cubature strategy to further reduce the eggbox effect, which is applicable for both uniform and nonuniform meshes.

For the all-electron calculations, we follow [2] to use the h -adaptive finite element methods to reduce the numerical oscillation. Different from the error indicator in [2], we follow [30] to introduce a residual based *a posteriori* error estimator to serve as the indicator. The numerical results show that with such error indicator, the performance of our h -adaptive method is improved effectively, i.e., with the decreasing the tolerance in the h -adaptive process, both the numerical accuracy of the numerical solution, and the translational and rotational invariance of the total energy, are improved significantly.

Numerical tests show that our method works very well for both all-electron (a dihydrogen molecule) and pseudopotential (a BCC unit cell of sodium) calculations on the bond length and ionic force. The accurate evaluation on the ionic force is crucial for the geometry optimization of the electronic structure, and for the numerical study of the molecular dynamics. Although our method works well on the force calculation, it should be noted that using (1.1) to calculate the force is not an efficient way, since extra three SCF iterations are needed for the force evaluation for one single nucleus. This could

be very time consuming, especially for the large molecules. This issue can be resolved by using the Hellmann-Feynman theorem. The theorem indicates that under some conditions, the force can be calculated only based on the current Hamiltonian, which dramatically simplifies the ionic force calculation. Unfortunately, there are some issues on using Hellmann-Feynman force instead of the original ionic force, when the real-space methods are employed. In our forthcoming paper, we will deliver some results on calculating ionic force with the mesh adaptive finite element methods.

Acknowledgments

The work of G. Bao was supported in part by the NSF grants DMS-0968360, DMS-1211292, the ONR grant N00014-12-1-0319, a Key Project of the Major Research Plan of NSFC (No. 91130004), and a special research grant from Zhejiang University. The research of G. H. Hu was supported in part by MYRG2014-00111-FST and MRG/016/HGH/2013/FST from University of Macau, 085/2012/A3 from FDCT of Macao S.A.R., and National Natural Science Foundation of China (Grant No. 11401608). The research of D. Liu was supported by NSF grants DMS-0968360 and NSF-DMS 1418959.

References

- [1] E. Anglada and J. M. Soler. Filtering a distribution simultaneously in real and fourier space. *Phys. Rev. B*, 73:115122, Mar 2006.
- [2] G. Bao, G. H. Hu, and D. Liu. An h-adaptive finite element solver for the calculations of the electronic structures. *Journal of Computational Physics*, 231(14):4967–4979, 2012.
- [3] G. Bao, G. H. Hu, and D. Liu. Numerical solution of the Kohn-Sham equation by finite element methods with an adaptive mesh redistribution technique. *Journal of Scientific Computing*, 55(2):372–391, 2013.
- [4] G. Bao, G. H. Hu, and D. Liu. Real-time adaptive finite element solution of time-dependent Kohn-Sham equation. *Journal of Computational Physics*, 281(0):743–758, 2015.
- [5] G. Bao, G. H. Hu, D. Liu, and S.T. Luo. Multi-physical modeling and multi-scale computation of nano-optical responses. *Recent Advances in Scientific Computing and Applications, Contemporary Mathematics*, 586:43–55, 2013.
- [6] A. E. Boguslavskiy, J. Mikosch, A. Gijsbertsen, M. Spanner, S. Patchkovskii, N. Gador, M. J. J. Vrakking, and A. Stolow. The multielectron ionization dynamics underlying attosecond strong-field spectroscopies. *Science*, 335(6074):1336–1340, 2012.
- [7] A. Castro, H. Appel, M. Oliveira, C. A. Rozzi, X. Andrade, F. Lorenzen, M. A. L. Marques, E. K. U. Gross, and A. Rubio. octopus: a tool for the application of time-dependent density functional theory. *physica status solidi (b)*, 243(11):2465–2488, 2006.
- [8] J.-L. Fattebert and M. B. Nardelli. Finite difference methods for ab initio electronic structure and quantum transport calculations of nanostructures. In C. Le Bris, editor, *Special Volume, Computational Chemistry*, volume 10 of *Handbook of Numerical Analysis*, pages 571–612. Elsevier, 2003.
- [9] R. Feynman. Forces in molecules. *Phys. Rev.*, 56:340–343, Aug 1939.

- [10] J. D. Gale. *SIESTA: A Linear-Scaling Method for Density Functional Calculations*, pages 45–75. John Wiley & Sons, Inc., 2011.
- [11] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, Nov 1964.
- [12] R. D. King-Smith, M. C. Payne, and J. S. Lin. Real-space implementation of nonlocal pseudopotentials for first-principles total-energy calculations. *Phys. Rev. B*, 44:13063–13066, Dec 1991.
- [13] A. V. Knyazev, M. E. Argentati, I. Lashuk, and E. E. Ovtchinnikov. Block locally optimal preconditioned eigenvalue solvers (BLOPEX) in HYPRE and PETSC. *SIAM Journal on Scientific Computing*, 2007:2224–2239, 2007.
- [14] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133–A1138, Nov 1965.
- [15] S. Kotochigova, Z. H. Levine, E. L. Shirley, M. D. Stiles, and C. W. Clark. Local-density-functional calculations of the energy of atoms. *Phys. Rev. A*, 55:191–199, Jan 1997.
- [16] G. Kresse and J. Furthmüller. Efficient iterative schemes for *ab initio* total-energy calculations using a plane-wave basis set. *Phys. Rev. B*, 54:11169–11186, Oct 1996.
- [17] R. Li. On multi-mesh h-adaptive methods. *Journal of Scientific Computing*, 24(3):321–341, 2005.
- [18] L. Lin, J. Lu, L. Ying, and W. E. Adaptive local basis set for Kohn-Sham density functional theory in a discontinuous Galerkin framework I: Total energy calculation. *Journal of Computational Physics*, 231(4):2140 – 2154, 2012.
- [19] M. Marques, C. Fiolhais, and M. Marques, editors. *A Primer in Density Functional Theory*. Springer, 1 edition, 2003.
- [20] M. A. L. Marques, M. J. T. Oliveira, and T. Burnus. Libxc: A library of exchange and correlation functionals for density functional theory. *Computer Physics Communications*, 183(10):2272–2281, 2012.
- [21] M. J. T. Oliveira and F. Nogueira. Generating relativistic pseudo-potentials with explicit incorporation of semi-core states using APE, the Atomic Pseudo-potentials Engine. *Computer Physics Communications*, 178(7):524–534, 2008.
- [22] T. Ono, M. Heide, N. Atodiresei, P. Baumeister, S. Tsukamoto, and S. Blügel. Real-space electronic structure calculations with full-potential all-electron precision for transition metals. *Phys. Rev. B*, 82:205115, Nov 2010.
- [23] J. E. Pask and P. A. Sterne. Finite element methods in *ab initio* electronic structure calculations. *Modelling and Simulation in Materials Science and Engineering*, 13(3):R71, 2005.
- [24] J. P. Perdew and A. Zunger. Self-interaction correction to density-functional approximations for many-electron systems. *Phys. Rev. B*, 23:5048–5079, May 1981.
- [25] P. Pulay. *Ab initio* calculation of force constants and equilibrium geometries in polyatomic molecules. *Molecular Physics*, 17(2):197–204, 1969.
- [26] J. Rapp and D. Bauer. Effects of inner electrons on atomic strong-field-ionization dynamics. *Physical Review A*, 89:033401, Mar 2014.
- [27] V. Schauer and C. Linder. All-electron Kohn-Sham density functional theory on hierarchic finite element spaces. *Journal of Computational Physics*, 250(0):644–664, 2013.
- [28] 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(2):256–280, 2010.
- [29] E. Tsuchida and M. Tsukada. Electronic-structure calculations based on the finite-element method. *Phys. Rev. B*, 52:5573–5578, Aug 1995.

- [30] R. Verfürth. *A Posteriori Error Estimation Techniques for Finite Element Methods*. Numerical Mathematics and Scientific Computation. OUP Oxford, 2013.
- [31] C. Yang, W. Gao, and J. Meza. On the convergence of the self-consistent field iteration for a class of nonlinear eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 30(4):1773–1788, 2009.
- [32] C. Yang, J. C. Meza, B. Lee, and L.-W. Wang. Kssolv1a MATLAB toolbox for solving the Kohn-Sham equations. *ACM Transactions on Mathematical Software (TOMS)*, 36(2):10, 2009.
- [33] L.-H. Zhang. On a self-consistent-field-like iteration for maximizing the sum of the rayleigh quotients. *Journal of Computational and Applied Mathematics*, 257(0):14–28, 2014.