

A Linearized Structure-Preserving Numerical Scheme for a Gradient Flow Model of the Kohn-Sham Density Functional Theory

Guanghai Hu^{1,2,3}, Ting Wang¹ and Jie Zhou^{4,*}

¹Department of Mathematics, Faculty of Science and Technology, University of Macau, Macao SAR, China.

²Zhuhai UM Science & Technology Research Institute, Zhuhai, Guangdong Province, China.

³Guangdong-Hong Kong-Macao Joint Laboratory for Data-Driven Fluid Mechanics and Engineering Applications, University of Macau, Macao SAR, China.

⁴School of Mathematical and Computational Sciences, Xiangtan University, Xiangtan, Hunan Province, China.

Received 22 April 2022; Accepted (in revised version) 8 October 2022.

Abstract. Dai *et al.* [Multiscale Model. Simul. **18** (2020)] proposed a gradient flow model and a numerical scheme for ground state calculations in Kohn-Sham density functional theory. It is a feature that orthonormality of all wave functions can be preserved automatically during the simulation which makes such a method attractive towards simulations for large scale systems. In this paper, two extensions are proposed for further improving the efficiency of the method. The first one is a linearization of the original nonlinear scheme. It is shown analytically that both the orthonormality of wave functions and the decay of the total energy can be preserved well by this linear scheme, while a significant acceleration can be observed from the numerical experiments due to the removal of an iteration process in the nonlinear scheme. The second one is the introduction of the adaptivity in the algorithm both temporally and spatially — i.e. an h -adaptive mesh method is employed to control the total amount of mesh grids, and an adaptive stop criterion in time propagation process is designed based on an observation that total energy always decays much faster at the beginning. Plenty of numerical experiments successfully demonstrate effectiveness of our method.

AMS subject classifications: 65N30, 37M05

Key words: Kohn-Sham density functional theory, gradient flow model, structure-preserving, linear scheme, adaptive strategy.

1. Introduction

The Kohn-Sham density functional theory has been widely used in electronic structure

*Corresponding author. Email address: zhouj@xtu.edu.cn (J. Zhou)

calculations [22]. Due to the nonlinearity of the governing equation and the complexity of electronic structure system, numerical solution became a main research area in the density functional theory, which plays an important role in applications such as the design of functional materials and new energy development [10, 12].

The solution of the Kohn-Sham equation [2, 27] by self-consistent field (SCF) iteration is a popular approach to determine the ground state of electronic structure systems. It has been realized in various mature software, including DFT-FE[†], VASP[‡], and Quantum Espresso [14]. Several successful techniques have been developed for accelerating the simulation, such as adaptive mesh methods [1, 3, 23, 24], density mixing [20] and quality solvers for generalized eigenvalue problems [8] with effective preconditioners [16] in SCF iteration, Chebyshev filtering [29]. In spite of its popularity, the orthogonality constraint in solving Kohn-Sham equation brings difficulty on fully taking advantage of hardware when designing parallel algorithms. Another widely used approach is the imaginary time propagation (ITP) method [18, 21]. The Wick rotation allows to transfer the complex-valued time-dependent Kohn-Sham equation into a real-valued one. The ground state of a given system can be obtained when the imaginary time approaches infinity.

Besides solving the Kohn-Sham equation, minimizing total energy of the system attracts more and more attention recently. Several pioneer works are based on this idea. For example, in [5, 28], Zhou group focused on gradient type methods and conjugate gradient method for electronic structure calculations. Yang *et al.* [26] developed a new direct constrained optimization algorithm based on the projection of the total energy functional into a sequence of subspaces and finding the minimizer of the energy functional on every subspace. A new iteration method based on the gradient on the Stiefel manifold has been proposed in [28]. The main advantage of this approach is that the method avoids solving nonlinear eigenvalue problems, so that the main cost is the assembling of the total energy functional and operations on a manifold. However, it should be noted that [28] requires an explicit operation in order to preserve the orthogonality of the wave functions and this diminishes the efficiency of the simulations. To overcome this bottleneck, an infeasible method was proposed for minimizing the total energy of a Kohn-Sham system [13], so that the ground state can be gradually obtained, without explicit treatment of the orthogonality.

It is mentioned that recently an extended gradient flow based model for electronic structure calculations was proposed in [6], which is a time evolution problem and different from either nonlinear eigenvalue problem or energy minimization problem. It is shown [6] that this extended gradient exponentially decays over time t , and the equilibrium point of this model corresponds to a local minimizer of the Kohn-Sham energy functional. Even better, it has shown that the orthogonality of those wave functions can be automatically preserved during the simulation. Furthermore, the gradient flow based model was used in a general framework of orthogonality preserving schemes for electronic structure calculations — cf. [7]. All these features demonstrate this method potential in large system simulations.

An implicit midpoint scheme employed in [6] for temporal discretization showed the effectiveness of the algorithm. However, the room for further improvements of the algo-

[†]<https://github.com/dftfeDevelopers/dftfe>

[‡]<http://cms.mpi.univie.ac.at/vasp>

rithm efficiency can be noted from the following two aspects. First of all, at each time step, an iteration is needed to handle the problem nonlinearity. The evaluation of the Hartree potential in this case has to be implemented several times, so is the derived linear system. Secondly, due to the nonlinearity of the Kohn-Sham model and the complexity of molecular structure, an adaptive mesh method is much better to tailor the finite element space in terms of numerical solution. Along with an adaptive strategy for time propagation, a faster solution time can be expected.

Taking into account the above observation, we develop a linearized algorithm based on the work [6]. From theoretical aspects, we analyze the energy decay behavior for the proposed numerical scheme, such that the convergence of the total energy towards the ground state can be guaranteed. In addition, it is possible to show that the orthonormality of the wave functions can be preserved without explicit calculations. To further accelerate the simulation, the adaptive strategies are studied with respect to spatial and temporal discretizations. Due to the nonlinearity of the governing equation, it is nontrivial to construct a quality mesh and the corresponding finite element space to represent the solution at the initial time. Hence, a dynamical adjustment during the simulation becomes necessary, based on the numerical solutions obtained during the process. This can be handled very well by an adaptive mesh method. A classic procedure of adaptive mesh method includes following steps:

... \rightarrow solve \rightarrow estimation \rightarrow mark \rightarrow refinement \rightarrow

After a numerical solution is obtained in the current finite element space, an error distribution is derived by an error estimation method. Finally, with a certain marking strategy, a new mesh is generated by a local refinement of the current mesh, so it is a new finite element space. In this paper, we use the recovery-type a posteriori error estimator, which reconstructs an approximation of the second derivative of electron density ρ and computes the element-wise H^2 norm. We use the weighted averaging recovery scheme with a volume weight. In addition to spatial discretization, an adaptive strategy is also designed for temporal discretization, with the size of time step is adaptively adjusted according to the numerical solution. Besides, our numerical experiments show that the decay of the total energy is always drastic at the beginning of the simulation in a given finite element space, but it takes a longer time for a gentle decay of the total energy. Based on this observation, a stopping criterion of numerical scheme is introduced to take full advantage of the initial and fast decay of the energy, so that the final and fitting finite element space can be obtained efficiently. The effectiveness of the above strategies in the simulation acceleration can be observed from numerical experiments in Section 5.

The paper is organized as follows. In Section 2, we introduce notations and preliminaries of the Kohn-Sham density functional theory. In Section 3, a full discretized scheme is described. In Section 4, we introduce three accelerated strategies and propose a practical adaptive computing algorithm. In Section 5, numerical experiments are presented to validate the efficiency and convergence of the proposed method. Conclusion is given in the last section, as well as future works.

2. Gradient Flow Methods for Ground State Calculation

Let Ω be a bounded domain in \mathbb{R}^n . We denote by (\cdot, \cdot) the standard inner L^2 product, i.e.

$$(u, v) = \int_{\Omega} u(x)v(x)dx.$$

Besides, $\|u\| = (u, u)^{1/2}$ is the associate L^2 norm and $\|\cdot\|_1$ refers to the H^1 norm

$$\|u\|_1 = \sqrt{\|u\|^2 + \|\nabla u\|^2}.$$

For the Sobolev spaces

$$\begin{aligned} H^1(\Omega) &:= \{u \in L^2(\Omega) : \|u\|_1 < \infty\}, \\ H_0^1(\Omega) &:= \{u \in H^1(\Omega) : u|_{\partial\Omega} = 0\}, \end{aligned}$$

we consider the set $\mathcal{H} = (H_0^1(\Omega))^N$ of vectors $\Psi = (\psi_1, \psi_2, \dots, \psi_N)$, $\Phi = (\phi_1, \phi_2, \dots, \phi_N) \in \mathcal{H}$ and the inner product

$$\langle \Psi^T \Phi \rangle := (\psi_i, \phi_j)_{i,j=1}^N \in \mathbb{R}^{N \times N}.$$

Consequently, the norm of Ψ is $\|\Psi\| = (\Psi, \Psi)^{1/2}$, where $(\Psi, \Psi) = \text{tr} \langle \Psi^T \Psi \rangle$ and $\text{tr}(A)$ is the trace of matrix A . Then we introduce the Stiefel manifold

$$M^N := \{U \in \mathcal{H} : \langle U^T U \rangle = I_N\},$$

where I_N is the $N \times N$ identity matrix.

The Kohn-Sham density functional theory has two common methods to obtain the ground state of a molecular system — viz. solving Kohn-Sham equation and minimizing total energy of the Kohn-Sham system. For completeness, we briefly describe the methods here.

We consider a molecular system consisting of M nuclei with the charges $\{Z_1, Z_2, \dots, Z_M\}$, respective locations $\{R_1, R_2, \dots, R_M\}$, and N electron orbitals. The general Kohn-Sham energy functional has the form

$$\begin{aligned} E(U) &= \frac{1}{2} \sum_{i=1}^N f_i \int_{\Omega} |\nabla u_i(x)|^2 dx + \int_{\Omega} V_{ext}(x) \rho(x) dx \\ &\quad + \frac{1}{2} \int_{\Omega} \int_{\Omega} \frac{\rho(x)\rho(x')}{|x-x'|} dx dx' + E_{xc}(\rho) \end{aligned} \quad (2.1)$$

for $U = (u_1, u_2, \dots, u_N)$. Here u_i , $i = 1, 2, \dots, N$ are the Kohn-Sham orbitals, f_i is the occupation number of the i -th orbital, $\rho(x)$ the electron density defined by

$$\rho(x) = \sum_{i=1}^N f_i |u_i(x)|^2,$$

and $V_{ext}(x)$ the external potential defined by

$$V_{ext}(x) = -\sum_{k=1}^M \frac{Z_k}{|x - R_k|}.$$

The third term in (2.1) is the Hartree energy. In practice, the Hartree potential $V_{Har}(x)$ is defined by

$$V_{Har}(x) := \int_{\Omega} \left(\frac{\rho(x')}{|x - x'|} \right) dx'.$$

It can be obtained by solving the Poisson equation with properly chosen conditions. The fourth term in (2.1) is the exchange-correlation energy, to which some approximations, such as local density approximation (LDA), general gradient approximation (GGA) [11] and so on, should be applied. The functional derivative of $E_{xc}(\rho)$ gives the exchange-correlation potential $v_{xc}(x) = \delta E_{xc}(\rho) / \delta \rho(x)$. Following [6], we consider $f_i = 2$, $i = 1, \dots, N$. For the minimization of the total energy, a class of optimization methods can be applied — e.g. gradient type methods with various orthogonalization technologies are presented in [5,28]. Note that the minimization problem

$$\inf_{U \in M^N} E(U), \quad (2.2)$$

on the Stiefel manifold U has been solved by an orthogonalization free method in [6]. The approach there motivated our work here.

Another popular method for the ground state calculation consists in solving the Kohn-Sham equation. Thus one considers the following Lagrange function:

$$L(U, \Lambda) = E(U) - \frac{1}{2} \sum_{i,j} ((u_i, u_j) - \delta_{ij}) \lambda_{ij}, \quad (2.3)$$

where $\Lambda(\lambda_{ij})_{N \times N}$ is a symmetric matrix. The necessary condition of the problem (2.2) is

$$\begin{aligned} \nabla_U L &= 0, \\ \nabla_{\Lambda} L &= 0. \end{aligned} \quad (2.4)$$

Substituting (2.3) into (2.4) yields

$$\frac{\partial L}{\partial u_i} = \left(-\frac{1}{2} \Delta + V_{ext} + V_{Har} + v_{xc} \right) u_i - \sum_{j=1}^N u_j \lambda_{ij} = 0. \quad (2.5)$$

Assume that the eigenvector decomposition of Λ is

$$\Lambda = Q^T \text{diag}(\tilde{\lambda}_1, \tilde{\lambda}_2, \dots, \tilde{\lambda}_n) Q, \quad (2.6)$$

where $Q = (q_{ij})_{N \times N}$ is an orthogonal matrix. Then

$$\left(-\frac{1}{2} \Delta + V_{ext} + V_{Har} + v_{xc} \right) \sum_{i=1}^N u_i q_{ik} = \sum_{i,j=1}^N u_j \lambda_{ij} q_{ik}$$

$$= \sum_{j=1}^N u_j \sum_{i=1}^N \lambda_{ij} q_{ik} = \sum_{j=1}^N u_j q_{jk} \tilde{\lambda}_k = \tilde{u}_k \tilde{\lambda}_k, \quad (2.7)$$

where $\tilde{u}_k = \sum_{j=1}^N u_j q_{jk}$.

As before, we write (u_i, λ_i) to represent the wave functions and corresponding eigenvalues. Then the well-known Kohn-Sham equation is given as: Find $(\lambda_i, u_i) \in R^- \times H_0^1(\Omega)$, $i = 1, 2, \dots, N$ such that

$$\begin{aligned} \left(-\frac{1}{2} \Delta + V_{ext} + V_{Har} + v_{xc} \right) u_i &= \lambda_i u_i, \\ (u_i, u_j) &= \delta_{ij}. \end{aligned} \quad (2.8)$$

Among the two above approaches, solving Kohn-Sham equation for ground state is a classic method popular in the fields of condensed matter and computational chemistry. Most of mature software for ab-initio calculations are based on this approach. It should be pointed out that the one of challenges in such an approach is the solution of a nonlinear eigenvalue problem. For large systems, the orthogonality of those wave functions, as well as the possible Rayleigh-Ritz procedure employed in the method would cause the nontrivial issues to efficiency and scalability of the algorithm. To avoid the orthogonalization operation in the algorithm, various methods based on the optimization problem have been used. One of them is a gradient flow method proposed [6], which can be described as follows. Thus in order to solve the minimization problem (2.2), one tries to determine a function $U(t)$ such that the solution U^* of (2.2) can be obtained when $t \rightarrow \infty$. More exactly, starting from an initial value U_0 , the solution goes along a curve $U = U(t) \in M^N$ and till the minimization point U^* can be reached. Naturally, it is required that the values of the functional decrease at the fastest rate along this curve $U = U(t)$, i.e. at each point the direction of the curve coincides with the direction of the fastest decrease of the functional. Therefore, the curve $U = U(t)$ has to satisfy the initial value problem

$$\begin{aligned} \frac{dU}{dt} &= -\nabla E(U), \quad 0 < t < \infty, \\ U(0) &= U_0(x) \end{aligned} \quad (2.9)$$

with the constraint $U(t) \in M^N$.

A non-trivial challenge in solving the problem (2.9) is to always keep the solution $U(t)$ in M^N during the simulations. This problem has been resolved by defining a gradient in M^N , cf. [6, 9]. For convenience, we recall this definition here.

We write the stand gradient of $E(U)$ as

$$\nabla E(U) = (E_{u_1}, E_{u_2}, \dots, E_{u_N}) \in (H^{-1}(\Omega))^N,$$

where $E_{u_i} \in H^{-1}(\Omega)$ is defined by

$$E_{u_i} = \frac{\delta E(U)}{\delta u_i}.$$

On the Grassmann manifold G^N , the gradient of $E(U)$ is defined according to [9] by

$$\tilde{\nabla}_G E(U) = \nabla E(U) - U \langle \nabla E(U), U \rangle^T \quad \text{for all } U \in M^N.$$

In order to preserve the orthogonality [6], we extend the domain of $\tilde{\nabla}_G E(U)$ from M^N to $(H_0^1(\Omega))^N$ as follows:

$$\nabla_G E(U) = \nabla E(U) \langle U^T U \rangle - U \langle \nabla E(U), U \rangle^T \quad \text{for all } U \in (H_0^1(\Omega))^N.$$

Using the extended gradient ∇_G , we write the gradient flow model based on DFT as

$$\begin{aligned} \frac{dU}{dt} &= -\nabla_G E(U), \quad 0 < t < \infty, \\ U(0) &= U_0(x). \end{aligned}$$

By using the method in the density functional theory, Dai *et al.* [6] made the following conclusions:

- (i) Total energy decreases monotonically over time.
- (ii) Solution converges exponentially to a local minimizer.
- (iii) If the initial value belongs to the Stiefel manifold, then so is the solution.

Moreover, numerical experiments confirm the theoretical results. However, the method in [6] is an implicit middle point scheme. Because of the nonlinearity of the governing equation, an iteration method was used. As the result, the evaluation of Hartree potential and solving the derived linear system have to be implemented several times during every time propagation. In what follows, we employ a linear scheme, which allows to avoid the issue mentioned. We also prove that the desired properties are preserved by our method.

3. A Structure-Preserving Linear Scheme

In this section, we consider a fully discretized scheme for the model [6]. Note that we use a linear numerical scheme for temporal discretization and the standard linear finite element method for spatial discretization. This linear scheme preserves the orthonormality of wave functions and the decay of the total energy. Let us start with temporal discretization.

3.1. Temporal discretization

The gradient flow model studied in [6] has the form

$$\begin{aligned} \frac{dU}{dt} &= -\nabla_G E(U), \quad 0 < t < \infty, \\ U(0) &= U_0(x), \end{aligned} \tag{3.1}$$

where the initial condition $U_0(x) \in M^N$. The linear scheme used in our method is given as follows.

Let $\{t_n : n = 0, 1, 2, 3, \dots\} \subset [0, +\infty)$ be the discretization points such that

$$0 = t_0 < t_1 < t_2 < \dots < t_n < \dots.$$

Adopting the notation

$$U_{n+1/2} = \frac{U_n + U_{n+1}}{2}, \quad (3.2)$$

we write

$$\frac{U_{n+1} - U_n}{\Delta t_n} = -A_{U_n} U_{n+1/2}, \quad (3.3)$$

where

$$A_{U_n} = \{\nabla E(U_n), U_n\} := \nabla E(U_n) U_n^\top - U_n \nabla E(U_n)^\top.$$

Dai *et al.* [6] showed that if the initial condition is in M^N , then at any successive time steps the numerical solutions they obtained, also belong to M^N . In other words, the orthogonality of the wave functions should not be preserved explicitly. This is the most attractive feature of the method. We want to show that the linear scheme above also preserves the orthonormality. More exactly, we have the following result.

Proposition 3.1. *If $U_0 \in M^N$ and U_n is obtained from (3.3), then $U_n \in M^N$.*

Proof. The relations (3.2) and (3.3) give

$$\begin{aligned} \frac{U_{n+1} - U_n}{\Delta t_n} &= -\frac{A_{U_n} U_n + A_{U_n} U_{n+1}}{2}, \\ U_{n+1} - U_n &= -\frac{\Delta t_n}{2} (A_{U_n} U_n + A_{U_n} U_{n+1}), \\ w \left(I + \frac{\Delta t_n}{2} A_{U_n} \right) U_{n+1} &= \left(I - \frac{\Delta t_n}{2} A_{U_n} \right) U_n, \\ U_{n+1} &= \left(I + \frac{\Delta t_n}{2} A_{U_n} \right)^{-1} \left(I - \frac{\Delta t_n}{2} A_{U_n} \right) U_n. \end{aligned}$$

Moreover, since $A_{u_n} = -A_{u_n}^T$ and

$$\left(I - \frac{\Delta t_n}{2} A_{U_n} \right)^{-1} \left(I + \frac{\Delta t_n}{2} A_{U_n} \right)^{-1} = \left(I + \frac{\Delta t_n}{2} A_{U_n} \right)^{-1} \left(I - \frac{\Delta t_n}{2} A_{U_n} \right)^{-1},$$

we have

$$\begin{aligned} &\langle U_{n+1}^T U_{n+1} \rangle \\ &= \left\langle \left(\left(I + \frac{\Delta t_n}{2} A_{U_n} \right)^{-1} \left(I - \frac{\Delta t_n}{2} A_{U_n} \right) U_n \right)^T \left(\left(I + \frac{\Delta t_n}{2} A_{U_n} \right)^{-1} \left(I - \frac{\Delta t_n}{2} A_{U_n} \right) U_n \right) \right\rangle \\ &= \left\langle \left(U_n^T \left(I + \frac{\Delta t_n}{2} A_{U_n} \right) \left(I - \frac{\Delta t_n}{2} A_{U_n} \right)^{-1} \left(I + \frac{\Delta t_n}{2} A_{U_n} \right)^{-1} \left(I - \frac{\Delta t_n}{2} A_{U_n} \right) U_n \right) \right\rangle \\ &= \left\langle \left(U_n^T \left(I + \frac{\Delta t_n}{2} A_{U_n} \right) \left(I + \frac{\Delta t_n}{2} A_{U_n} \right)^{-1} \left(I - \frac{\Delta t_n}{2} A_{U_n} \right)^{-1} \left(I - \frac{\Delta t_n}{2} A_{U_n} \right) U_n \right) \right\rangle \\ &= \langle U_n^T U_n \rangle = I. \end{aligned} \quad \square$$

In addition, we consider the decay of the total energy. It is needed in order to guarantee the convergence of the solutions to the ground state of a given system.

Proposition 3.2. Assume that $U_0 \in M^N$, U_n is obtained from (3.3), Δt_n is extremely small, and $\nabla E(U)$ is Lipschitz continuous in the M^N , i.e.

$$\| \nabla E(U) - \nabla E(V) \| \leq L \| U - V \| \quad \text{for all } U, V \in M^N. \quad (3.4)$$

Then

$$E(U_{n+1}) \leq E(U_n). \quad (3.5)$$

Proof. Introduce the function

$$g(t) := tU_{n+1} + (1-t)U_n, \quad t \in [0, 1].$$

Assuming that $E(g(t))$ is differential in $(0, 1)$ implies the existence of a number $\xi \in (0, 1)$ such that

$$\begin{aligned} E(U_{n+1}) - E(U_n) &= E(g(1)) - E(g(0)) = \left. \frac{d(E(g(t)))}{dt} \right|_{t=\xi} \cdot (1-0) \\ &= \text{tr} \left\langle \nabla E(g(\xi))^T g'(\xi) \right\rangle = \text{tr} \left\langle \nabla E(g(\xi))^T (U_{n+1} - U_n) \right\rangle \\ &= -\Delta t_n \text{tr} \left\langle \nabla E(g(\xi))^T A_{U_n} U_{n+1/2} \right\rangle. \end{aligned} \quad (3.6)$$

Using (3.2) and (3.3) yields

$$U_{n+1/2} = \left(I + \frac{\Delta t_n}{2} A_{U_n} \right)^{-1} U_n. \quad (3.7)$$

Substituting (3.7) into (3.6) gives

$$\begin{aligned} &\text{tr} \left\langle \nabla E(g(\xi))^T A_{U_n} U_{n+1/2} \right\rangle \\ &= \text{tr} \left\langle \nabla E(g(\xi))^T A_{U_n} \left(I + \frac{\Delta t_n}{2} A_{U_n} \right)^{-1} U_n \right\rangle \\ &= \text{tr} \left\langle \nabla E(g(\xi))^T A_{U_n} U_n \right\rangle - \frac{\Delta t_n}{2} \text{tr} \left\langle \nabla E(g(\xi))^T A_{U_n} A_{U_n} U_n \right\rangle + o(\Delta t_n). \end{aligned} \quad (3.8)$$

Note that $(\Delta t_n/2) \text{tr} \left\langle \nabla E(g(\xi))^T A_{U_n} A_{U_n} U_n \right\rangle$ is of higher order than $\text{tr} \left\langle \nabla E(g(\xi))^T A_{U_n} U_n \right\rangle$. Hence, only the later should be estimated. Writing

$$\begin{aligned} &\text{tr} \left\langle \nabla E(g(\xi))^T A_{U_n} U_n \right\rangle \\ &= \text{tr} \left\langle \nabla E(g(0))^T A_{U_n} U_n \right\rangle + \text{tr} \left\langle \nabla E(g(\xi) - E(g(0)))^T A_{U_n} U_n \right\rangle. \end{aligned} \quad (3.9)$$

we consider the first item of (3.9), i.e.[§]

$$\text{tr} \left\langle \nabla E(g(0))^T A_{U_n} U_n \right\rangle = \text{tr} \left\langle \nabla E(g(0))^T A_{g(0)} g(0) \right\rangle$$

[§]See Remark 3.1 for more detail.

$$= -\frac{1}{2} \text{tr} \langle (A_{g(0)})^2 \rangle = \frac{1}{2} \text{tr} \langle A_{g(0)}^* A_{g(0)} \rangle = \frac{1}{2} \|A_{g(0)}\|^2. \quad (3.10)$$

We obtain

$$\| \| A_{g(0)} g(0) \| \| \leq \| A_{g(0)} \| \| \| g(0) \| \| \leq \sqrt{N} \| A_{g(0)} \|. \quad (3.11)$$

Thus,

$$\begin{aligned} \text{tr} \langle \nabla E(g(0))^T A_{U_n} U_n \rangle &= \text{tr} \langle \nabla E(g(0))^T A_{g(0)} g(0) \rangle \\ &\geq \frac{1}{2N} \| \| A_{g(0)} g(0) \| \|^2. \end{aligned} \quad (3.12)$$

Considering the second term in (3.9), we write

$$\begin{aligned} &\text{tr} \langle \nabla E(g(\xi) - E(g(0)))^T A_{U_n} U_n \rangle \\ &\leq L \| \| g(0) - g(\xi) \| \| \cdot \| \| A_{g(0)} g(0) \| \| \\ &\leq L \Delta t_n \| \| A_{U_n} U_{n+1/2} \| \| \cdot \| \| A_{g(0)} g(0) \| \| \\ &\leq L \Delta t_n \| \| A_{U_n} (I + (\Delta t_n/2) A_{U_n})^{-1} U_n \| \| \cdot \| \| A_{g(0)} g(0) \| \| \\ &\leq L \Delta t_n \| \| A_{U_n} U_n \| \| \cdot \| \| A_{g(0)} g(0) \| \| + o(\Delta t_n) \\ &\leq L \Delta t_n \| \| A_{g(0)} g(0) \| \|^2 + o(\Delta t_n). \end{aligned} \quad (3.13)$$

Finally, (3.5) follows from (3.6), (3.8), (3.9), (3.12) and (3.13). \square

Remark 3.1. The relation

$$\text{tr} \langle \nabla E(U)^T A_U U \rangle = -\frac{1}{2} \text{tr} \langle (A_U)^2 \rangle$$

has been used in the proof of (3.10). Now we want to prove it. Since

$$A_U = \nabla E(U) U^T - U \nabla E(U)^T,$$

we write

$$\begin{aligned} \text{tr} \langle (A_U)^2 \rangle &= \text{tr} \langle (\nabla E(U) U^T - U \nabla E(U)^T) (\nabla E(U) U^T - U \nabla E(U)^T) \rangle \\ &= \text{tr} \langle \nabla E(U) U^T \nabla E(U) U^T + U \nabla E(U)^T U \nabla E(U)^T \rangle \\ &\quad - \text{tr} \langle (\nabla E(U) U^T U \nabla E(U)^T + U \nabla E(U)^T \nabla E(U) U^T) \rangle \\ &= 2 \text{tr} \langle \nabla E(U)^T U \nabla E(U)^T U \rangle - 2 \text{tr} \langle \nabla E(U)^T \nabla E(U) U^T U \rangle. \end{aligned} \quad (3.14)$$

On the other hand,

$$\begin{aligned} \text{tr} \langle (\nabla E(U)^T A_U U) \rangle &= \text{tr} \langle \nabla E(U)^T (\nabla E(U) U^T - U \nabla E(U)^T) U \rangle \\ &= \text{tr} \langle \nabla E(U)^T \nabla E(U) U^T U \rangle - \text{tr} \langle \nabla E(U)^T U \nabla E(U)^T U \rangle \\ &= -\frac{1}{2} \text{tr} \langle (A_U)^2 \rangle. \end{aligned}$$

Comparing it with (3.14) gives the result.

It can be observed that with the linear scheme (3.3), an obvious advantage in the simulation is that, both the evaluation of the Hartree potential and the solving the linear system need to be implemented one time in each time propagation. Our numerical experiments would clearly show this advantage.

In order to get a fully discretized scheme, we next employ the finite element method for spatial discretization.

3.2. Spatial discretization

We assume that the computational domain $\Omega \subset \mathbb{R}^3$ is bounded. Let \mathcal{T}_h be a tetrahedral partition of Ω . Let τ refer to the elements of \mathcal{T}_h , and let S_h be the standard Lagrange finite element space on \mathcal{T}_h . More exactly, for an integer $r \geq 1$, we have

$$S_h := \{v_h \in C(\Omega) : v_h|_{\tau} \in P_r \text{ for all } \tau \in \mathcal{T}_h, v_h|_{\partial\Omega} = 0\},$$

where P_r is the space of polynomials of the degree at most r .

The fully-discrete problem of (3.3) consists in finding $U_h^{n+1} \in (S_h)^N$ such that

$$(U_h^{n+1}, V_h) - \frac{\Delta t_n}{2} a_n(U_h^{n+1}, V_h) = \frac{\Delta t_n}{2} a_n(U_h^n, V_h) + (U_h^n, V_h) \quad \text{for all } V_h \in (S_h)^N, \quad (3.15)$$

where U_h^n is given and

$$a_n(U, V) = (\nabla E(U_h^n) \langle U_h^n, U \rangle, V) - (U_h^n \langle \nabla E(U_h^n), U \rangle, V).$$

Remark 3.2. The coefficient matrix of the linear system (3.15) is full, which can influence the simulation efficiency. However, this matrix is the sum of a mass matrix and a number of full matrices which are the outer products of two vectors. Therefore, an efficient matrix vector multiplication can be implemented.

Remark 3.3. We employ a GMRES solver is in (3.15). In simulation the time step Δt_n is always small ranging from 1.0×10^{-6} to 1.0×10^{-3} . Therefore, the mass matrix M is a good preconditioner for the GMRES method. In addition, the previous value U_n is a good candidate for the initial value in GMRES. With these two strategies, a good performance of GMRES is observed in our simulations.

4. Acceleration Strategies

The improvement of the efficiency can be achieved from the numerical scheme above, based on our numerical experiments. Further improvements can be derived by adaptive mesh method, adaptive time step, and adaptive stop criterion towards generating an optimal finite element space during the simulations. Below we describe these strategies in more detail.

First we introduce an h -adaptive mesh method. In Kohn-Sham Hamiltonian, due to the nonlinearity and the existence of singularity, it is nontrivial to give a quality mesh and corresponding finite element space for representing the solution at the initial time. Therefore,

a dynamical adjustment based on the numerical solutions obtained in simulations becomes a necessity. To resolve these issues, a classical h -adaptive mesh method is employed. The general h -adaptive mesh method can be illustrated as follows:

$\cdots \rightarrow \text{solve} \rightarrow \text{estimation} \rightarrow \text{mark} \rightarrow \text{refinement} \rightarrow \cdots$

More precisely, to get \mathcal{T}_{k+1} from \mathcal{T}_k , we first solve the discrete equation (3.15) on the current mesh \mathcal{T}_k to obtain an approximate solution and compute an error indicator $\eta_{k,\tau}$. Using the error indicator and the Dörfler's marking strategy, some elements are marked and then refined. Finally, a new finite element space \mathcal{T}_{k+1} will be built, and the numerical solution will be updated from old space to the new space. In our work, we use recovery type error estimation to construct the error indicator

$$\eta_{h,\tau} = h_\tau \int_\tau |\nabla R(\nabla \rho)|^2 dx. \quad (4.1)$$

The definition of the recovery operator R can be found in [25].

In addition to the acceleration strategy above, we also adopt the following adaptive stopping criterion strategy. By Proposition 3.2, the energy decreases over time. However, as it found in our experiments, the total energy drastically decreases only during a few initial steps. After that, it will take a very long time to correct to the lowest energy — cf. Fig. 4 and the right part of Fig. 6. Based on such observation, it would be a reasonable strategy to just make stop after the drastic decay and use that solution to introduce the h -adaptive process. In our work, when the rate of decrease is very slow, we stop this propagation and refine the mesh to speed up the energy decay. In this way, the efficiency of the algorithm can be greatly improved. Furthermore, we have to point out that it is important to use certain standards to judge the energy decrease rate. We use the first derivative of total energy in time. When this quantity is smaller than ϵ , we think that the rate of energy decay is too slow. That is, when the following condition satisfies:

$$\frac{E(U^n) - E(U^{n+1})}{\Delta t^n} < \epsilon,$$

we stop calculation with the current mesh and call for the h -adaptive algorithm to prepare a new mesh for the next computing round. Note that this criterion ϵ should not be too small.

In addition to the h -adaptive and stop criterion in the time propagation, we also employ an adaptive time step strategy. Theoretically, the exact solution of the gradient flow problem can be reached when time t approaches infinity, which means a large size of time step is desired to speed up the simulation. Due to the nonlinearity of the governing equation, an adaptive strategy is needed to determine the size of the time step locally. The detailed process is as follows. First, we set $\Delta t = \min h_{\tau_i}^2$, where τ_i is the i -th element of the current mesh. Then if Δt is too large, namely $E(U^n) - E(U^{n+1}) < 0$, we reduce the time step — i.e. we set $\Delta t = \Delta t/2$. Finally, we can also try to increase the time step after every 200 steps by $\Delta t = 2\Delta t$.

Combining the above strategies, we present the following algorithm, and our experiment results show its effectiveness.

Algorithm 4.1 Adaptive computing.

-
- 1: Set $k = 0, n = 0$, given an adaptive initial grid \mathcal{T}_0 and initial value $U_0^0 \in (S_0)^N \cap M^N$.
Choose a positive integer *maxrefine* based on the computer memory.
 - 2: Choose $\Delta t = \min(h_{\tau_i})^2$, τ_i is the i th element of current mesh \mathcal{T}_k .
 - 3: Find $U_k^{n+1} \in (S_k)^N$ such that

$$(U_k^{n+1}, V_k) - \frac{\Delta t}{2} a_n(U_k^{n+1}, V_k) = \frac{\Delta t}{2} a_n(U_k^n, V_k) + (U_k^n, V_k), \quad \forall V_k \in (S_k)^N.$$

- 4: If $(E(U_k^{n+1}) \geq E(U_k^n))$, set $\Delta t = \Delta t/2$ and go back to line 3.
- 5: If $(\text{mod}(n, 200) == 0)$, set $\Delta t = 2\Delta t$.
- 6: If $((E(U_k^n) - E(U_k^{n+1}))/\Delta t_n < \epsilon)$, $U_k^{\text{end}} = U_k^{n+1}$ goto line 7; else set $n = n+1$ and go back to line 3.
- 7: If $(k < \text{maxrefine})$ goto line 8; else stop the whole simulation.
- 8: Estimate. Compute an error indicator $\eta_{k,\tau}$ on \mathcal{T}_k .
- 9: Mark. Select the minimal elements set $M_k \subset \mathcal{T}_k$ such that

$$\sum_{\tau \in M_k} \eta_{k,\tau}^2 \geq \theta \sum_{\tau \in \mathcal{T}_k} \eta_{k,\tau}^2 \quad \text{for some } \theta \in (0, 1).$$

- 10: Refine. Bisect the marked element M_k and get the mesh \mathcal{T}_{k+1} .
 - 11: Set $U_{k+1}^0 = U_k^{\text{end}}$, and orthogonalize $U_{k+1}^0, n = 0, k = k + 1$, go back to line 2.
-

5. Numerical Examples

In this section, we show the performance of the method. It is divided into three parts. In the first part, using the same mesh, we compare our scheme (3.3) and the nonlinear scheme in [6] in terms of the total energy, time propagation steps and the total number of linear systems that have to be solved in simulations. Moreover, the propositions established in Section 3 are verified numerically. In the second part, we show the numerical behavior of Algorithm 4.1, from which the advantages from three acceleration strategies can be observed clearly. In the last part, the robustness of our algorithm is demonstrated — i.e. we show that our algorithm is not sensitive to the selection of the initial values.

The hardware for the simulations is a Dell OptiPlex 7060 workstation with Inter(R) Core(TM) i7-8700 CPU @3.20 GHz and 8.00 GB of memory, while the code for the simulation is developed based on the iFEM package [4]. Some parameters, such as the time step Δt , computation domain Ω , and stop criterion ϵ , have no common principles in selection. Based on our numerical experience, we chose parameter values depending on the problem considered.

5.1. Comparison and properties testing

Following the ideas of [19], we construct a good quality mesh and compare the performance of our scheme (3.3) with the scheme [6]. In the next two examples, we set time step $\Delta t = 1.0e-3$ and choose a random initial value. The stop criterion for the ground state is

$$||| \nabla_G E(U) ||| < 1.0e-8.$$

Example 5.1. Consider the gradient flow model for a helium (He) atom with 1 electron orbital. The position of the nucleus is the origin. We set external potential $V_{ext}(x) = -2|x|^{-1}$ and electron density $\rho(x) = 2|u(x)|^2$. The Hartree potential is obtained by solving the Poisson equation

$$-\nabla^2 V_{Har}(x) = 4\pi\rho(x)$$

with the zero Dirichlet boundary conditions. The evaluation of exchange-correlation potential follows the work [15]. The computational domain is $\Omega_1 = (-10000, 10000)^3$.

In this example, we choose a computational domain large enough to simply calculate the Hartree potential. The left part of Fig. 1 shows the nonuniform mesh for simulation of a helium atom which includes 10381 nodes. Note that the singularity is well captured. The left part of Fig. 2 shows the energy curves obtained by our scheme (3.3) and the nonlinear scheme [6]. Both energies decay monotonically, consistent with Proposition 3.2. The time propagation steps and the total number of linear systems to be solved in the simulation are shown in Table 1. Although the time propagation steps required by our scheme (3.3) is a bit higher than for the nonlinear scheme, the number of linear systems to solve is much smaller. The reason is that with the help of linearization of scheme (3.3), both the evaluation of the Hartree potential and solving the derived linear system are implemented only once at each propagation step. Consequently, the acceleration of the simulations can be expected. The right part of Fig. 2 shows the normalization of the wave function, consistent with Proposition 3.1.

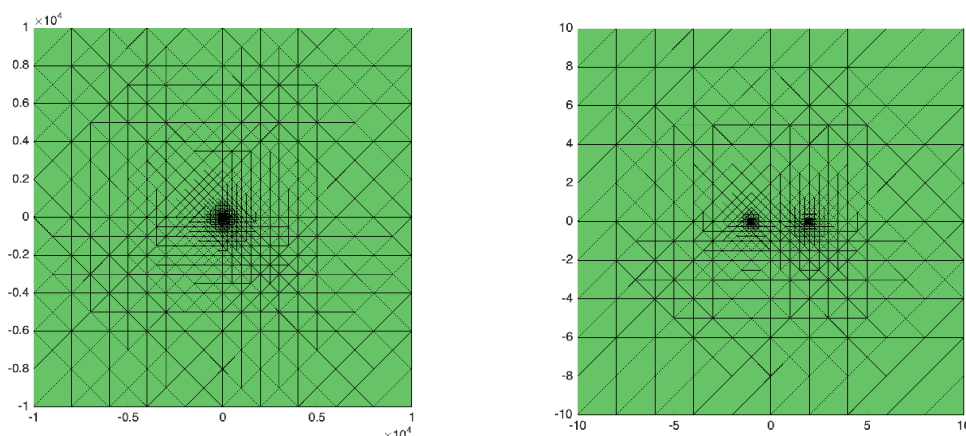


Figure 1: Left: Nonuniform mesh of 10381 nodes for simulation of a He atom. Right: Nonuniform mesh of 12351 nodes for simulation of a LiH molecule.

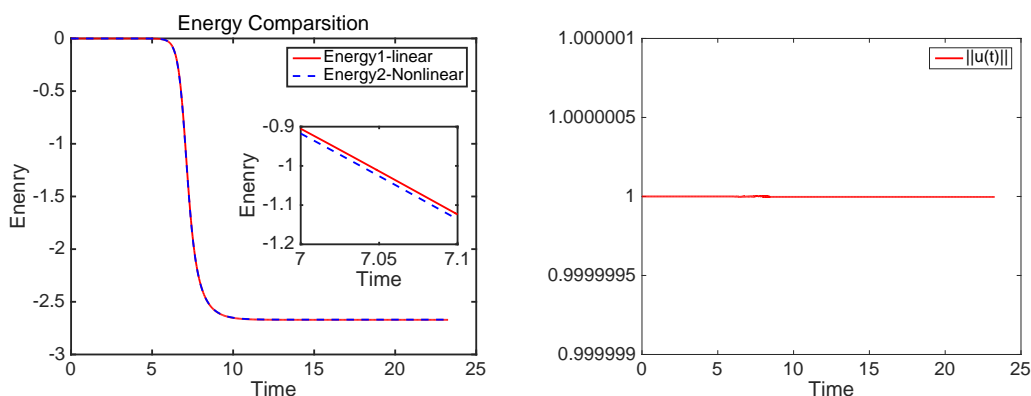


Figure 2: Left: Performance of linear scheme (3.3) and nonlinear scheme in [6]. Right: L_2 norm of u .

Table 1: A comparison between linear scheme (3.3) and nonlinear scheme in [6].

	Linear scheme (3.3)		Nonlinear scheme in [6]	
	No. of time steps	No. of linear systems	No. of time steps	No. of linear systems
rule He	31075	31075	28182	76312
LiH	4730	4730	4652	13415

Example 5.2. Consider the gradient flow model for a lithium hydride (LiH) molecule with 2 electron orbitals. The positions of nuclei are as follows: lithium atom $R_1 = (-1.0075, 0, 0)$, hydrogen atom $R_2 = (2.0075, 0, 0)$. Set the external potential $V_{ext}(x) = -3|x - R_1|^{-1} - |x - R_2|^{-1}$ and electron density $\rho(x) = 2(|u_1(x)|^2 + |u_2(x)|^2)$. The Hartree potential is obtained by solving the same Poisson equation with the boundary conditions given by the multipole expansion [1]. The evaluation of exchange-correlation potential is the same as in Example 5.1. We set the computational domain $\Omega_2 = (-10, 10)^3$.

The right part of Fig. 1 shows the nonuniform mesh for the simulation of a LiH molecule which includes 12351 nodes. We note that many more mesh points are located near the two singularities. The molecular structure of LiH and the contour of the electron density

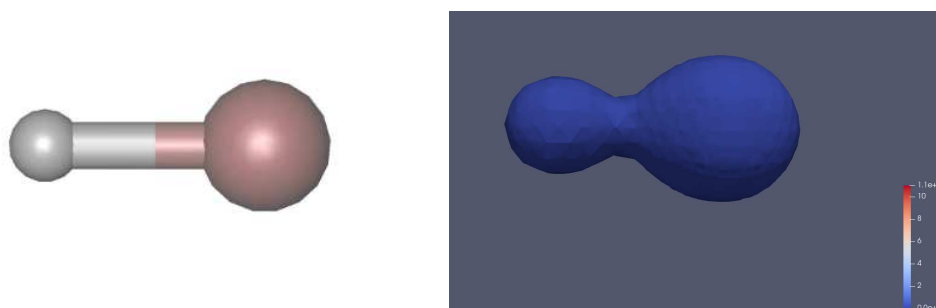


Figure 3: Left: Molecular structure of LiH [6]. Right: The contour plot of the electron density.

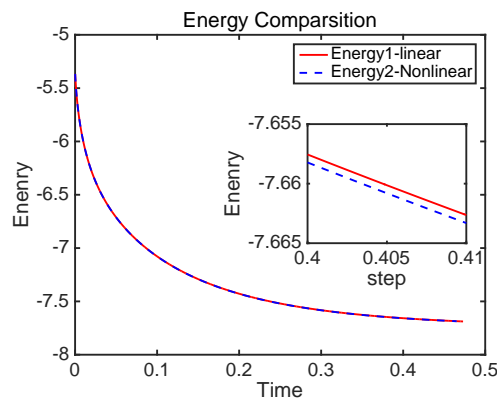


Figure 4: Performance of linear scheme (3.3) and nonlinear scheme in [6].

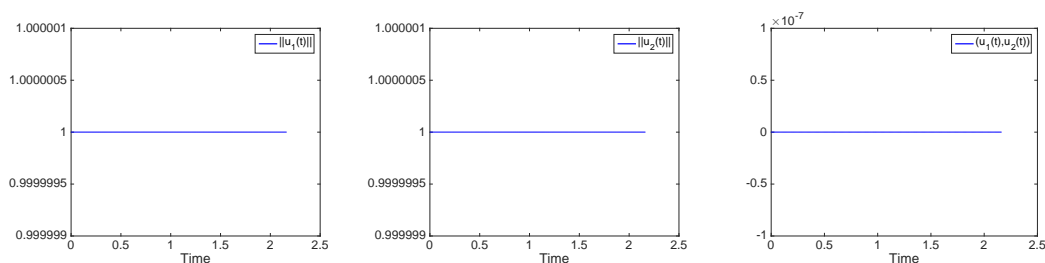


Figure 5: Left: L_2 norm of u_1 . Middle: L_2 norm of u_2 . Right: Inner product between u_1 and u_2 .

are shown in Fig. 3. Fig. 4 shows energy curves obtained by two different schemes. The results by our scheme (3.3) are in good qualitative agreement with the ones obtained by the nonlinear scheme [6]. Comparing the total number of linear systems to be solved — cf. Table 1, we see that our scheme (3.3) delivers the same accurate numerical solutions in a much smaller time. Fig. 5 shows the orthonormality of the wave functions, consistent with the Proposition 3.1.

5.2. Efficiency improvement

Examples 5.1-5.2 indicate that with a good initial mesh, our scheme (3.3) works well in the ground state calculations of small molecule structures. However, if the grid is fixed and time moves, the total energy decreases more and more slowly. Therefore, here we apply Algorithm 4.1 with acceleration strategies to calculate the ground state of the methane molecule. In the next example, the stop criterion is set to as $\epsilon = 0.75$ and the initial value is randomly chosen.

Example 5.3. Consider the gradient flow model for a methane (CH_4) molecule with 5 electron orbitals. The positions of nuclei are as follows: carbon atom $R_1 = (0, 0, 0)$, hydrogen atoms $R_2 = (1.3092, 1.3092, 1.3092)$, $R_3 = (-1.3092, -1.3092, 1.3092)$, $R_4 = (1.3092, -1.3092, -1.3092)$ and $R_5 = (-1.3092, 1.3092, -1.3092)$. We set the external

potential

$$V_{ext}(x) = -6|x - R_1|^{-1} - \sum_{i=2}^5 |x - R_i|^{-1}$$

and the electron density

$$\rho(x) = 2 \sum_{i=1}^5 |u_i(x)|^2.$$

The evaluations of the Hartree potential and the exchange-correlation potential are the same as in Example 5.2. The computational domain is $\Omega_2 = (-10, 10)^3$.

The left part of Fig. 6 shows the energy curve obtained by Algorithm 4.1. Note that the curve looks like a staircase, which is not smooth enough. This is because of the influence of the adaptive refine method. When the mesh is refined, we found that at the beginning the energy decreases drastically, and it takes a very long time to converge to the lowest energy. If we refine the mesh locally again, the energy decreases drastically again. We proceed in the same way till a satisfactory result is obtained. It is worth noting that the efficiency of simulations is greatly improved by using Algorithm 4.1 with an appropriate termination condition.

For comparison, we carry out another experiment. We use the finest grid generated by the adaptive Algorithm 4.1 as the initial computed grid, and the same initial value for calculation but do not refine the mesh in the computing process. The right part of Fig. 6 shows that the energy drops and converges to a solution of the similar accuracy, but it costs about three times as long as the former. The reason is that if choosing a suitable termination, these gradient flow methods always converge quickly at the beginning. Once it is judged that the convergence rate is too slow, we refine the grid to generate a new mesh, so that our calculation is always in a state of fast convergence. In the previous coarse grid calculation, the termination condition may be rough, but it is not necessary to calculate accurately, which can be understood as pre-optimization. The molecular structure of CH₄ and the contour of the electron density are shown in Fig. 7.

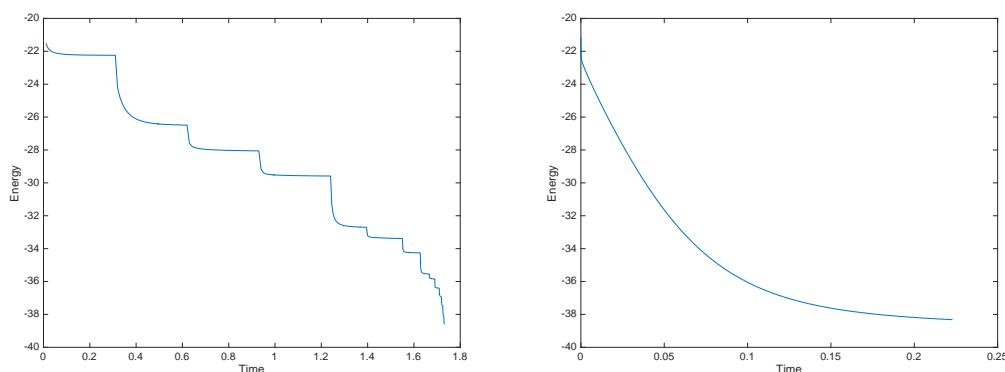


Figure 6: Left: Computed by Algorithm 4.1, cost CPU time 3520 seconds. Right: Computed by scheme (3.3) based on finest grid and time adaptive, cost CPU time 12530 seconds.

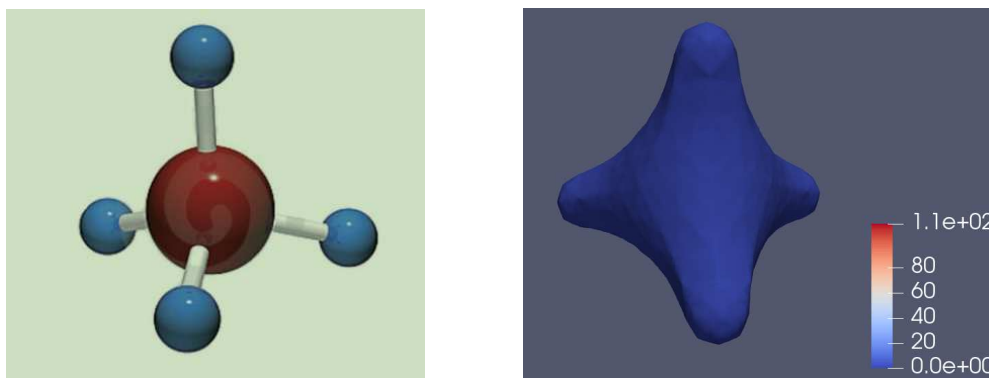


Figure 7: Left: Molecular structure of CH4 [15]. Right: The contour plot of the electron density.

5.3. Sensitivity to initial value choice

Robustness is a desired property for the numerical method. In our simulations, it is found that the convergence of the Algorithm 4.1 is not sensitive to the selection of the initial values, which shows the robustness of our method. We demonstrate this by an example. The stop criterion here is set to $\epsilon = 0.5$.

Example 5.4. Consider the gradient flow model for a water (H_2O) molecule with 5 electron orbitals. The positions of nuclei are as follows: oxygen atom $R_1 = (0, 0, 0.12)$, hydrogen atoms $R_2 = (0, 0.762, -0.479)$ and $R_3 = (0, -0.762, -0.479)$. We set external potential

$$V_{ext}(x) = -8|x - R_1|^{-1} - \sum_{i=2}^3 |x - R_i|^{-1}$$

and the electron density

$$\rho(x) = 2 \left(\sum_{i=1}^5 |u_i(x)|^2 \right).$$

The evaluations of the Hartree potential and the exchange-correlation potential are the same as in Example 5.2. The computational domain is the cube $\Omega_3 = (-100, 100)^3$.

Results shown in Fig. 8 are obtained by the same algorithm but with different initial values. In the left part of Fig. 8, the initial value is given by the interpolation of a convergent solution from a coarser mesh, while the initial value in the right part of Fig. 8 is set randomly. From the results, it can be seen clearly that the convergence behaviors of both results look quite similar to each other, and that both results converge to the reference value -75.11 [17]. In addition, around the end of the simulation, the solution has already close to the ground state total energy, based on this criterion ϵ . It can be expected that just a few iteration steps can give us one situation at the criterion satisfied and we can jump to the next h -adaptive process. We also note that the energy decreases faster at the end of process, and in all examples above, random initial guesses work well. This demonstrates the robustness of the method.

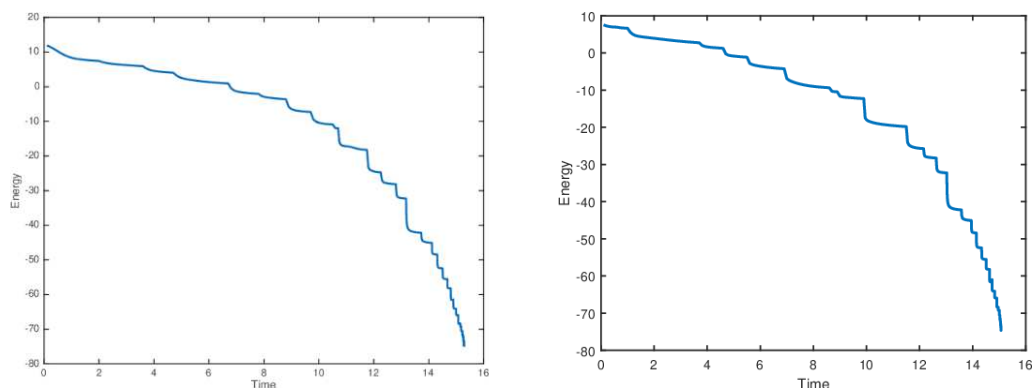


Figure 8: Left: Energy curve, coarse grid solution as the initial value. Right: Energy curve, a random initial value.

6. Conclusions

We propose a linear scheme to improve the computational efficiency of a numerical method for a gradient flow model [6] and study the energy decay and the preservation of the wave functions orthonormality. In order to handle the nonlinearity of the governing equation and the singularity of the external potential, we develop an algorithm combining various acceleration strategies based on the h -adaptive meshes, adaptive time steps and adaptive stop criterion. Numerical experiments show that these strategies substantially improve the simulation efficiency.

Experiments also show that our numerical approach preserves the orthonormality of the wave functions and the energy decay for fixed meshes. Besides, in the examples considered our method spent less time in order to obtain approximate solutions of the same accuracy as the non-linear scheme mentioned. Other examples demonstrate the improved efficiency of the algorithm in simulating a complex molecular system and its successful work even when it starts with a random value.

In the future, we will continue to study the algorithms related to this kind of problems, and further accelerate the speed of the current algorithm. It is known that the Algorithm 4.1 is not insensitive to the initial value, and the Newton iterative method has quadratic convergence. Hence, the combination of two methods will be studied to take the advantages from both.

Acknowledgements

The authors would like to thank Prof. Xiaoying Dai from the Chinese Academy of Sciences for the valuable discussions.

The first author would like to thank financial support from the National Natural Science Foundation of China (Grant Nos. 11922120, 11871489), FDCT of Macao SAR (Grant No. 0082/2020/A2), the University of Macau (Grant No. MYRG2020-00265-FST), and the Guang dong-Hong Kong-Macao Joint Laboratory for Data-Driven Fluid Mechanics and

Engineering Applications (Grant No. 2020B1212030001). The second author would like to thank the support from the UM-Funded PhD Assistantship from University of Macau. The third author was partially supported by the Macao Young Scholar Program (Grant No. AM201919), by the Excellent Youth Project of Hunan Education Department (Grant No. 19B543) and by the Hunan National Applied Mathematics Center of Hunan Provincial Science and Technology Department (Grant No. 2020ZYT003).

References

- [1] 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*, *J. Sci. Comput.* **55**, 372–391 (2013).
- [2] G. Bao, G.H. Hu and D. Liu, *Towards translational invariance of total energy with finite element methods for Kohn-Sham equation*, *Comput. Phys. Commun.* **19**, 1–23 (2016).
- [3] H.J. Chen, X.Y. Dai, X.G. Gong, L.H. He and A.H. Zhou, *Adaptive finite element approximations for Kohn-Sham models*, *Multiscale Model. Simul.* **12**, 1828–1869 (2014).
- [4] L. Chen, *iFEM: an integrated finite element methods package in MATLAB*, University of California at Irvine (2009).
- [5] X.Y. Dai, Z. Liu, L.W. Zhang and A.H. Zhou, *A conjugate gradient method for electronic structure calculations*, *SIAM J. Sci. Comput.* **39**, A2702–A2740 (2017).
- [6] X.Y. Dai, Q. Wang and A.H. Zhou, *Gradient flow based Kohn-Sham density functional theory model*, *Multiscale Model. Simul.* **18**, 1621–1663 (2020).
- [7] X.Y. Dai, L.W. Zhang and A.H. Zhou, *Convergent and orthogonality preserving schemes for approximating the Kohn-Sham orbitals*, arXiv:2111.02779 (2021).
- [8] J.A. Duersch, M.Y. Shao, C. Yang and M. Gu, *A robust and efficient implementation of LOBPCG*, *SIAM J. Sci. Comput.* **40**, C655–C676 (2018).
- [9] A. Edelman, T.A. Arias and S.T. Smith, *The geometry of algorithms with orthogonality constraints*, *SIAM J. Matrix Anal. Appl.* **20**, 303–353 (1998).
- [10] Y. Elbaz, D. Furman and M.C. Toroker, *Modeling diffusion in functional materials: From density functional theory to artificial intelligence*, *Adv. Funct. Mater.* **30**, 1900778 (2020).
- [11] C. Fiolhais, F. Nogueira and M. Marques, *A Primer in Density Functional Theory*, Springer Science & Business Media (2003).
- [12] A.P. Gaiduk, *Theory of Model Kohn-Sham Potentials and Its Applications*, The University of Western Ontario (2013).
- [13] B. Gao, G.H. Hu, Y. Kuang and X. Liu, *An orthogonalization-free parallelizable framework for all-electron calculations in density functional theory*, *SIAM J. Sci. Comput.* **44**, B723–B745 (2020).
- [14] P. Giannozzi et al., *Quantum espresso: A modular and open-source software project for quantum simulations of materials*, *Journal of physics: Condensed matter* **21**, 395502 (2009).
- [15] G.H. Hu, H.H. Xie and F. Xu, *A multilevel correction adaptive finite element method for Kohn-Sham equation*, *J. Comput. Phys.* **355**, 436–449 (2018).
- [16] L. Hung, C. Huang and E.A. Carter, *Preconditioners and electron density optimization in orbital-free density functional theory*, *Commun. Comput. Phys.* **12**, 135–161 (2012).
- [17] R.D. Johnson III, *NIST Computational Chemistry Comparison and Benchmark Database, NIST Standard Reference Database Number 101 Release 22, May 2022*, <https://cccbdb.nist.gov/energy2x.asp>.
- [18] Y. Kuang and G.H. Hu, *An adaptive FEM with ITP approach for steady Schrödinger equation*, *Int. J. Comput. Math.* **95**, 187–201 (2018).

- [19] Y. Kuang and G.H. Hu, *On stabilizing and accelerating SCF using ITP in solving Kohn-Sham equation*, Commun. Comput. Phys. **28**, 999–1018 (2020).
- [20] L. Lin, J.F. Lu and L.X. Ying, *Numerical methods for Kohn-Sham density functional theory*, Acta Numer. **28**, 405–539 (2019).
- [21] S. McArdle, T. Jones, S. Endo, Y. Li, S.C. Benjamin and X. Yuan, *Variational ansatz-based quantum simulation of imaginary time evolution*, npj Quantum Inform. **5**, 1–6 (2019).
- [22] R.G. Parr and W.T. Yang, *Density-Functional Theory of Atoms and Molecules*, OUP USA (1994).
- [23] S.M. Schnepp, G. Erion and T. Weiland, *Development of a self-consistent particle-in-cell (PIC) code using a time-adaptive mesh technique*, in: *Proc. 10th European Particle Accelerator Conference (EPAC)*, 2182–2184, Citeseer (2006).
- [24] E. Tsuchida and M. Tsukada, *Adaptive finite-element method for electronic-structure calculations*, Phys. Rev. B, **54**, 7602 (1996).
- [25] J.C. Xu and Z.M. Zhang, *Analysis of recovery type a posteriori error estimators for mildly structured grids*, Math. Comput. **73**, 1139–1152 (2004).
- [26] C. Yang, J.C. Meza and L.W. Wang, *A constrained optimization algorithm for total energy minimization in electronic structure calculations*, J. Comput. Phys. **217**, 709–721 (2006).
- [27] L. Yang, Y.D. Shen, Z.C. Hu and G.H. Hu, *An implicit solver for the time-dependent Kohn-Sham equation*, Numer. Math. Theor. Meth. Appl. **14**, 261–284 (2021).
- [28] X. Zhang, J.W. Zhu, Z.W. Wen and A.H. Zhou, *Gradient type optimization methods for electronic structure calculations*, SIAM J. Sci. Comput. **36**, C265–C289 (2014).
- [29] Y.K. Zhou, Y. Saad, M.L. Tiago and J.R. Chelikowsky, *Self-consistent-field calculations using Chebyshev-filtered subspace iteration*, J. Comput. Phys. **219**, 172–184 (2006).