Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2019-0024

Vol. 28, No. 3, pp. 999-1018
September 2020

On Stabilizing and Accelerating SCF Using ITP in Solving Kohn–Sham Equation

Yang Kuang[1] and Guanghui Hu[1,2,*]

[1] *Department of Mathematics, University of Macau, Macao SAR, China.*
[2] *Zhuhai UM Science & Technology Research Institute, Guangdong Province, China.*

Received 11 February 2019; Accepted (in revised version) 11 April 2020

Abstract. It is found that imaginary time propagation method can effectively deliver a convergent result in solving Kohn–Sham equation, but a sufficient long simulation is needed to reach an accurate enough result, while the self-consistent field iteration method for Kohn–Sham equation can be more efficient when it works, but it sometimes suffers from divergence. In this work, we take advantage of the convergence of imaginary time propagation method by generating a quality initial guess to improve the behavior of self-consistent field iteration. A number of numerical experiments successfully show that i). for those self-consistent field iterations which are sensitive to the initial guess, the results obtained from imaginary time propagation method make the iterations converge, and ii). generally, the convergence of self-consistent field iteration can be accelerated by imaginary time propagation method. It is shown that all-electron models can be resolved well with the proposed method.

AMS subject classifications: 35Q55, 65N30

Key words: Self-consistent field iteration, imaginary time propagation, finite element method, all-electron calculation.

1 Introduction

With the development of the hardware, as well as the requirements appeared in modern physical and chemical experiments, quality numerical simulations for all-electron Kohn–Sham models in density functional theory have been attracting more and more attention, please refer to [20,26,28,30,34] and references therein.

There are two popular approaches for solving Kohn–Sham model, i.e., the imaginary time propagation (ITP) method [3,6–8,14,27], and the self-consistent field (SCF) iteration method [4,5,20,26,32,33]. The implementation of ITP is quite simple. By introducing an

[*] Corresponding author. *Email addresses:* kuangyoung0107@gmail.com (Y. Kuang), garyhu@umac.mo (G. Hu)

imaginary time by Wick rotation, the complex-valued time-dependent Kohn–Sham equation is transformed to a real-valued time-dependent equation whose solution approaches asymptotically the ground state of the given system. Based on our numerical experience [25] which will also be illustrated in the context, it is noted that a sufficiently long simulation is needed by ITP method to obtain the ground state with an accurate enough result. As another widely-used method, SCF iteration method introduces a sequence of iterations to resolve the nonlinearity of the Kohn-Sham equation. More specifically, in each iteration, a generalized eigenvalue problem is derived with a given electron density, then the new electron density is generated by solving this generalized eigenvalue problem. The ground state of the given system is obtained when the iteration converges. As a brief comparison, the implementation of ITP method is simple, and the solving of eigenvalue problem is avoided. Although it may bring nontrivial challenges on solving eigenvalue problems, the SCF method could be more efficient when it works. Unfortunately, there are lots of evidence [24, 36, 37] to show the possible failures on the convergence of SCF iteration. The situation could become worse when all-electron Kohn–Sham models are considered where the mesh grid around the singularities should be dense enough to resolve the singularities. If a uniform mesh is adopted in the simulations, a great amount of degrees of freedom will be required to obtain an accurate result, which makes the computation less of efficiency. In this work, we follow the theoretical framework by Huang et al. [17] to derive a general mesh density function for any atomic or molecular system, and apply the result in generating the radial mesh in the numerical examples.

In solving the Kohn–Sham equation with a random initial guess on a nonuniform mesh, the following two issues may bring troubles in the convergence of SCF iteration: i) the condition number of the discretized Hamiltonian is too large, and ii) the initial guess is quite far away from the solution. On the one hand, the linear system becomes difficult to solve when the minimum mesh size $h_{\min}$ is small. This is due to the fact that the condition number of discretized matrix will become larger when $h_{\min}$ goes smaller [15]. On the other hand, the random initial guess could lead to the divergence of the SCF iteration. Due to the nonlinearity of the Kohn–Sham system, the quality of the initial guess plays an important role in the convergence of the iterative method. A bad initial guess which is far away from the convergence region will lead to divergent results, which can be observed from the numerical examples in this paper.

To improve the convergence of SCF, one could provide a good initial guess to the SCF iteration. Based on our numerical experience, it is found that ITP can provide an acceptable result rapidly in just a few steps even starting from a random initial guess. This motivates us to take the advantage of ITP to obtain a quality initial condition for the SCF iteration.

In this work, starting from a random initial guess, firstly the ITP method is applied to propagate the random initial guess for a few steps, and then the ITP solution is served as the initial guess of the SCF iteration to solve the Kohn–Sham equation. Due to the flexibility in handling the complex boundary condition and complicated computational domain, the finite element method is adopted in spatial discretization in this paper. It is noted that we do the orthonormalization process after each propagation. We adopt the backward Euler scheme for the temporal discretization for its simplicity and the allowance for larger time step than the explicit methods. A number of numerical experiments successfully show that i). for those self-consistent field iterations which are sensitive to the initial guess, the results obtained from imaginary time propagation method make the iterations converge, and ii). generally, the convergence of self-consistent field iteration can be accelerated by imaginary time propagation method. In addition, based on the numerical experiments, the CPU time occupied by the ITP part only take no more than 10% of the whole CPU time. As a conclusion, our proposed method can solve the Kohn–Sham equation very efficient.

This paper is organized as follows. In Section 2, the Kohn–Sham equation and its finite element discretized form are introduced, and a radial mesh generating strategy is presented. In Section 3 we propose a new numerical method based on the imaginary time propagation technique. Numerical results are reported in Section 4 to verify the convergence and efficiency of the proposed algorithm. Finally, the conclusion is given in Section 5.

2 Kohn–Sham equation

We consider a molecular system in $\mathbb{R}^3$ consisting of M nuclei with charges $\{Z_1,\cdots,Z_M\}$ located at the positions $\{\boldsymbol{R}_1,\cdots,\boldsymbol{R}_M\}$, and N_e electrons in the non-relativistic and spin-unpolarized setting. Denote the number of occupied orbitals by N_o. Since electrons are fermions which implies that at most two electrons occupy the same orbital, then $N_o = N_e/2$ if N_e is even, otherwise $N_o = (N_e+1)/2$. Thus the ground state solution of the system can be obtained by solving the lowest N_o eigenpairs of the following Kohn–Sham equation (in atomic units)

$$\begin{cases} H\psi_k(\boldsymbol{r}) = \varepsilon_k \psi_k(\boldsymbol{r}), & k=1,2,\cdots,N_o, \\ \displaystyle\int \psi_k \psi_l d\boldsymbol{r} = \delta_{kl}, & k,l=1,2,\cdots,N_o, \end{cases} \tag{2.1}$$

where δ_{kl} is the Kronecker operator and $\boldsymbol{r}=(x,y,z)$ stands for the spatial coordinates. Assume $\{\varepsilon_k\}$ is the set of the eigenvalues of the Hamiltonian operator H corresponding to the eigenstates $\{\phi_k\}$ with an increasing order, i.e., $\varepsilon_1 \leq \cdots \leq \varepsilon_k \leq \cdots$. The Hamiltonian H consists of two parts, the kinetic potential part $-\nabla^2/2$ and the effective potential part $V_{eff}([n];\boldsymbol{r})$ which is given as follows

$$V_{eff} = V_{ext}(\boldsymbol{r}) + V_{Har}([n];\boldsymbol{r}) + V_{xc}([n];\boldsymbol{r}), \tag{2.2}$$

where n is the electronic density which can be written as

$$n(\boldsymbol{r}) = \sum_{k=1}^{N_o} f_k |\psi_k(\boldsymbol{r})|^2, \tag{2.3}$$

with f_k the occupation number of the k-th orbital. The first term of the effective potential is the external potential which describes the electrostatic potential due to the nuclei which takes the following form

$$V_{ext}(\boldsymbol{r}) = -\sum_{I=1}^{M} \frac{Z_I}{|\boldsymbol{r}-\boldsymbol{R}_I|}. \tag{2.4}$$

The second term is the Hartree potential describing the Coulomb repulsion among the electrons, which can be written as

$$V_{Har}([n];\boldsymbol{r}) = \int \frac{n(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|} d\boldsymbol{r}'. \tag{2.5}$$

And the last term V_{xc} stands for the exchange-correlation potential, which is caused by the Pauli exclusion principle and other non-classical Coulomb interactions. The analytical expression for the exchange-correlation term is unknown and therefore approximations for exchange-correlation term are required. In this paper, the local density approximation (LDA) from the library Libxc [31] is used in simulations.

Due to the flexibility in handling the complex computational domain and complicated boundary conditions, the finite element method is applied to discretize the Kohn–Sham equation in this work. In practical computations, we always use a bounded polyhedral domain $\Omega \subset \mathbb{R}^3$ to be served as the computational domain owing to the exponential decay behavior of the ground state wavefunction of the Schrödinger equation [1]. Thus the variational form of the Kohn–Sham equation (2.1) on Ω can be formulated as: Find $(\varepsilon_k, \psi_k) \in \mathbb{R} \times H_0^1(\Omega)$, $k = 1,2,\cdots,N_o$, such that

$$\int_{\Omega} \left(\frac{1}{2} \nabla \psi_k \nabla \varphi + V_{eff} \psi_k \varphi \right) d\boldsymbol{r} = \varepsilon_k \int_{\Omega} \psi_k \varphi d\boldsymbol{r}, \quad \forall \varphi \in H_0^1(\Omega), \tag{2.6}$$

where $H^1(\Omega)$ is the standard Sobolev space, $H_0^1(\Omega) = \{\varphi \in H^1(\Omega) : \varphi = 0 \text{ on } \partial\Omega\}$. Then the total energy of the system can be formulated as

$$E_{tot} = \frac{1}{2} \sum_{k=1}^{N_o} f_k \int_{\Omega} |\nabla \psi_k|^2 d\boldsymbol{r} + \int_{\Omega} \epsilon_{xc}(n(\boldsymbol{r})) d\boldsymbol{r} - \int_{\Omega} \sum_{k=1}^{M} \frac{Z_k n(\boldsymbol{r})}{|\boldsymbol{r}-\boldsymbol{R}_k|} d\boldsymbol{r} + \frac{1}{2} \iint \frac{n(\boldsymbol{r}) n(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|} d\boldsymbol{r}' d\boldsymbol{r}, \tag{2.7}$$

where ϵ_{xc} is the exchange-correlation energy per volume. Furthermore, the Hartree potential V_{Har} can be obtained by solving the following Poisson equation

$$-\nabla^2 V_{Har}(\boldsymbol{r}) = 4\pi n(\boldsymbol{r}), \tag{2.8}$$

with the boundary condition obtained from the multipole expansion method [5].

Assume that the finite element space $V_h \subset H_0^1(\Omega)$ is constructed on the bounded domain Ω partitioned by $\mathcal{T} = \{\mathcal{T}_K, K = 1,2,\cdots,N_{ele}\}$, where N_{ele} denotes the total number of elements of $\mathcal{T}$. In this paper, the linear tetrahedron finite element is used (see Fig. 1). The basis functions are linear and can be constructed following [9]. They have a property

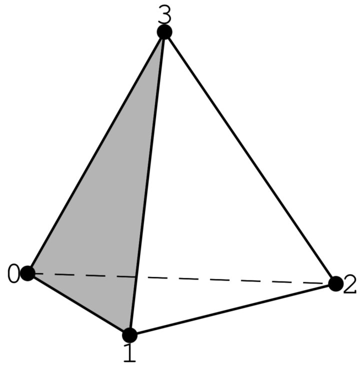

Figure 1: The linear tetrahedron finite element $\mathcal{T}_K$.

that each basis function takes value 1 at its associated node while 0 at all other nodes. For examples, in the linear finite element $\mathcal{T}_K$ in Fig. 1, the basis function φ_0 associated with point 0 takes value 1 at points 0, and value 0 at points 1,2,3. We further assume that $\{\varphi_i\}_{i=1}^{N_{bas}}$, $i=1,\cdots,N_{bas}$ are the set of basis functions, $\{\boldsymbol{r}_i\}_{i=1}^{N_{bas}}$ are the set of nodes in space V_h, and N_{bas} denotes the dimension of the V_h. Consequently, for any function $\psi(\boldsymbol{r})$, it can be approximated as ψ^h in space V_h with the form

$$\psi^h=\sum_{i=1}^{N_{bas}}\psi_i^h\varphi_i, \tag{2.9}$$

where ψ_i^h is equivalent to the value of $\psi(\boldsymbol{r})$ at the i-th node, i.e, $\psi(\boldsymbol{r}_i)$.

In this paper, the scalar product of ψ and ϕ on domain Ω is denoted by $\langle\psi,\phi\rangle_\Omega$ and is evaluated as $\sum_K\langle\psi,\phi\rangle_{\mathcal{T}_K}$. And in each element $\mathcal{T}_K$, the integral is obtained from numerical quadrature with Gaussian type quadrature points as follows

$$\langle\psi,\phi\rangle_{\mathcal{T}_K}=\int_{\mathcal{T}_K}\psi(\boldsymbol{r})\phi(\boldsymbol{r})d\boldsymbol{r}\approx\sum_{l=1}^{N_q}\omega_l\psi(\boldsymbol{r}_l)\phi(\boldsymbol{r}_l), \tag{2.10}$$

where N_q is the number of numerical quadrature points, and $\{\boldsymbol{r}_l\}$ are the quadrature points which are generated following from J. Yu [38]. Precisely, we let $N_q=4$ and all the quadrature points locate inside the element with the same weight $\omega_l=0.25$.

With the above notations, the discretized variation form of (2.6) becomes: Find $(\varepsilon_k^h,\psi_k^h)\in\mathbb{R}\times V_h$, $k=1,2,\cdots,N_o$ such that

$$\frac{1}{2}\langle\nabla\psi_k^h,\,\nabla\varphi\rangle_\Omega+\langle V_{eff}\psi_k^h,\,\varphi\rangle_\Omega=\varepsilon_k^h\langle\psi_k^h,\,\varphi\rangle_\Omega,\quad\forall\varphi\in V_h, \tag{2.11}$$

where the term with the superscript h means that this term is related to the finite dimensional space V_h.

The k-th wavefunction ψ_k in the finite element space V_h can be approximated by

$$\psi_k^h=\sum_{i=1}^{N_{bas}}\psi_{k,i}^h\varphi_i. \tag{2.12}$$

To find out $\boldsymbol{\psi}_k^h=(\psi_{k,1}^h,\cdots,\psi_{k,N_{bas}}^h)^T$, we thus need to solve the following generalized eigenvalue problem

$$A\boldsymbol{\psi}_k^h=\varepsilon_k B\boldsymbol{\psi}_k^h, \tag{2.13}$$

where B is positive-definite and A and B are both Hermitian matrices. The entries of A, B are

$$A_{ij}=\frac{1}{2}\langle\nabla\varphi_i,\,\nabla\varphi_j\rangle_\Omega+\langle V_{eff}\varphi_i,\,\varphi_j\rangle_\Omega, \tag{2.14}$$

$$B_{ij}=\langle\varphi_i,\,\varphi_j\rangle_\Omega, \tag{2.15}$$

respectively. To solve the generalized eigenvalue problem, we use the popular locally optimal block preconditioned conjugate gradient (LOBPCG) method [21,22].

On the same finite element space, the Hartree potential V_{Har} can be represented as

$$V_{Har}^h=\sum_{i=1}^{N_{bas}}V_{Har,i}^h\varphi_i, \tag{2.16}$$

where $\{V_{Har,i}^h\}$ are coefficients. Denoted by $\boldsymbol{V}_{Har}^h=(V_{Har,1}^h,\cdots,V_{Har,N_{bas}}^h)^T$, then the discretized form for the Poisson equation (2.8) can be expressed as

$$S\boldsymbol{V}_{Har}^h=\boldsymbol{f}, \tag{2.17}$$

with a Dirichlet boundary condition following the strategy in the paper [5], in which a multipole expansion approximation is adopted for the boundary values. Here S is the stiffness matrix with entry

$$S_{ij}=\langle\nabla\varphi_i,\,\nabla\varphi_j\rangle_\Omega, \tag{2.18}$$

and $\boldsymbol{f}$ is a vector with entry

$$f_i=\langle 4\pi n,\,\varphi_i\rangle_\Omega. \tag{2.19}$$

In this paper, the linear system (2.17) is solved by the algebraic multigrid (AMG) method [10].

Towards the quality description of external potential in Kohn–Sham equation, we use a radial mesh which following the mesh generating framework by Huang and Russel [17]. We aware that there are a lot of works on designing the radial mesh for the all-electron calculations, please refer to [13, 23, 26, 29, 35] for more details. Applying this framework in [17], we generate a mesh with mesh size function near the nucleus taking the form $h(\boldsymbol{r})=r^{6/5}$ in practical simulations, where $h(\boldsymbol{r})$ is defined to be the diameter of the element where the spatial point $\boldsymbol{r}$ belongs to. As a result, for the system with external potential V_{ext} (see Eq. (2.4)) the mesh size function can be designed as follows

$$h(\boldsymbol{r}) = \min\left\{\beta Z_1^{-\frac{2}{5}} r_1^{\frac{6}{5}}, \cdots, \beta Z_M^{-\frac{2}{5}} r_M^{\frac{6}{5}}, \gamma\right\}, \tag{2.20}$$

with $r_I = |\boldsymbol{R}_I - \boldsymbol{r}|$. In practice, the parameters $\beta = 0.125$ and $\gamma = 8$ are sufficient for systems under study, and they will be used in most simulations unless explicitly stated. The software Gmsh [16] is used to generate the radial meshes. By some certain commands, the singularities can be forced to node points of the 3D mesh and the mesh will be generated following the mesh size function (2.20). Details of the implementation can be found in Appendix A.

With the quality radial mesh, we then can solve the Kohn–Sham equation on the finite element space built on the mesh.

3 Self-consistent field iteration and imaginary time propagation

It is well known that the Kohn–Sham equation (2.1) is a nonlinear equation since the effective potential term depends on the wavefunctions. To solve this nonlinear problem, the most popular way is the self-consistent field iteration which is illustrated in Fig. 2. Briefly, an initial guess for electron density is required, and the electron density is updated after solving the linearized Kohn–Sham equation in each iteration. The iteration will be stopped if the difference between the current density and the previous density is small enough. However, in the practical simulations, SCF sometimes suffers from divergence [24, 36, 37]. In this section, two divergent examples are demonstrated first and then the ITP-SCF method is presented to stabilize and accelerate the convergence of SCF iteration. Note that in this paper, we are using a simple mixing scheme

$$n^{(s+1)} = (1-a)\tilde{n}^{(s+1)} + a n^{(s)}, \tag{3.1}$$

where $n^{(s)}$ is the density at s-th step, while $\tilde{n}^{(s+1)}$ is the density from solving the eigenvalue problem. Here $a = 0.3$ is used in our simulation to generate $n^{(s+1)}$.

3.1 Failure of SCF iteration

Lithium is the lightest metallic element and it has a single s valence electron. Thus it always serves as the starting point for a theoretical understanding of metal clusters [2, 11, 19]. We study the ground state of Li_7 (D_{5h}) [2] with structure as depicted in the left of Fig. 3. For the ground state calculation of Li_7, a random initial guess is provided for the SCF iteration in solving the all-electron Kohn–Sham equation (2.1). The computational mesh is generated from the strategy present in the last section. The difference $\|n_{new} - n_{old}\|_2$ between current electron density n_{new} and previous electron density n_{old} for Li_7 is displayed in the right of Fig. 3. From this figure, we can observe that the difference of

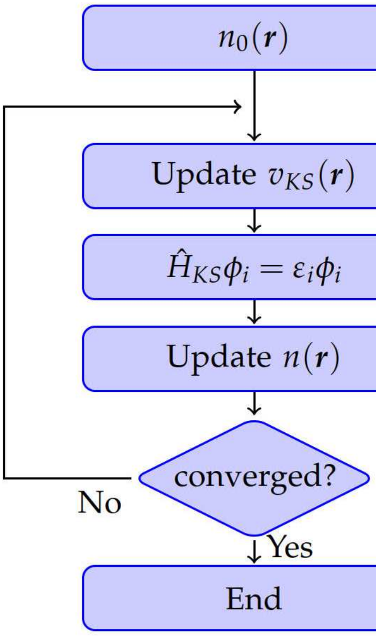
$n_0(\boldsymbol{r})$
Update $v_{KS}(\boldsymbol{r})$
$\hat{H}_{KS}\phi_i = \varepsilon_i\phi_i$
Update $n(\boldsymbol{r})$
converged?
No
Yes
End

Figure 2: Flowchart of SCF iteration.

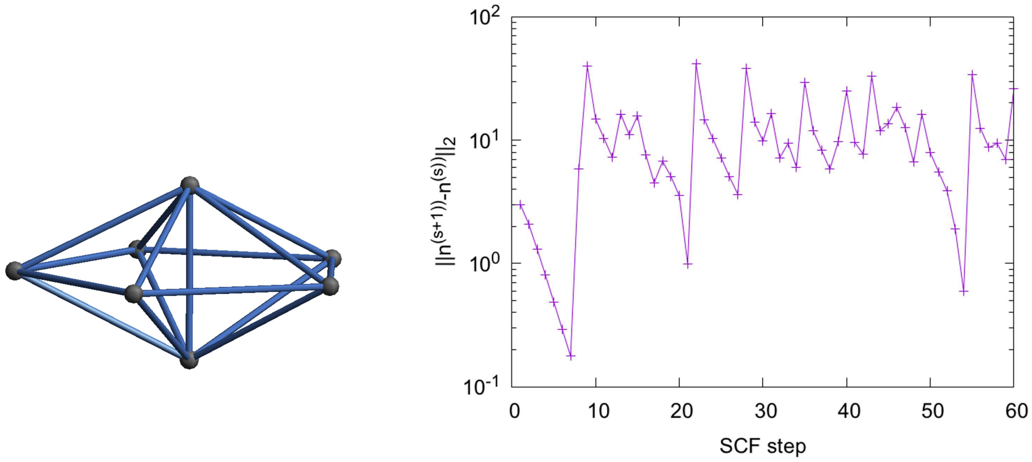

Figure 3: Ground state calculations of Li_7. Left: the chemical structure for Li_7. Right: density difference versus SCF iteration numbers for a random initial guess.

density oscillates a lot from the 8-th SCF step and finally there is no convergent result at least within 60 SCF steps.

This divergent phenomenon is also found in the all-electron calculation of the molecule azobenzene ($C_{12}H_{10}N_2$) which is certainly one of the most important organic molecules [18]. The chemical structure of the molecule azobenzene is displayed in the left of Fig. 4

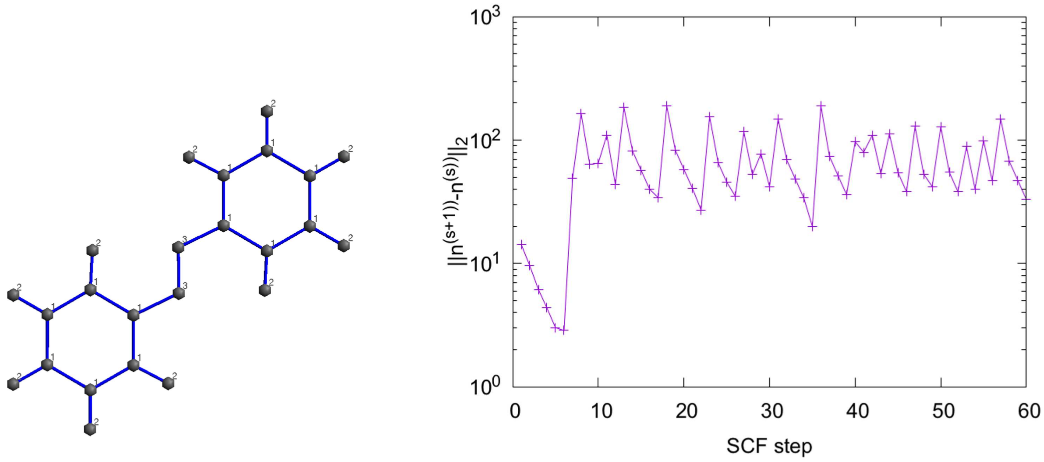

Figure 4: Ground state calculations of $C_{12}H_{10}N_2$. Left: the chemical structure for $C_{12}H_{10}N_2$, label 1,2,3 represent C,H,N atom, respectively. Right: density difference versus SCF iteration numbers for a random initial guess.

and the difference between current electron density and previous electron density is displayed in the right of Fig. 4. The total energy oscillates much more dramatic than that of Li_7 and behaves divergent within 60 SCF steps.

The quality of the initial guess plays an important role in the convergence of the nonlinear solvers. Therefore, to improve the convergence of SCF, one can provide a good initial guess to the SCF iteration. Based on our numerical experience, it is found that ITP can provide an acceptable result rapidly in just a few steps even starting from a random initial guess. This motivates us to take the advantage of ITP to obtain a quality initial condition for the SCF iteration. In this way, the convergence of SCF iteration is stabilized and accelerated.

3.2 Imaginary time propagation

The imaginary time propagation method (ITP) can be used to calculate the ground state of a quantum system. The basic idea of the ITP method origins from the fact that the set of eigenstates $\{\phi_j\}$ of the Hamiltonian H forms a real orthonormal basis on its domain. To illustrate the ITP method, we consider the ground state of a quantum system which can be modeled by a stationary Kohn–Sham equation as shown in (2.1). Actually, this time-independent equation originates from the time-dependent Schrödinger equation

$$i\frac{\partial\psi(\boldsymbol{r},t)}{\partial t}=H\psi(\boldsymbol{r},t),\quad \psi(\boldsymbol{r},0)=\psi_0(\boldsymbol{r}), \tag{3.2}$$

where $\psi(\boldsymbol{r},t)$ means that ψ is a function depending on both $\boldsymbol{r}$ and t, and here $\psi_0(\boldsymbol{r})$ is the initial condition and H is the Hamiltonian. A Wick rotation of the time coordinate,

$t=-i\tau$, transforms (3.2) into the following type equation:

$$-\frac{\partial\psi(\boldsymbol{r},\tau)}{\partial\tau}=H\psi(\boldsymbol{r},\tau),\quad \psi(\boldsymbol{r},0)=\psi_0(\boldsymbol{r}), \tag{3.3}$$

with the formal solution $\psi(\boldsymbol{r},\tau)=e^{-\tau H}\psi_0\boldsymbol{r}$. After expanding the initial condition ψ_0 in the basis of eigenstates $\{\phi_i\}$,

$$\psi_0(\boldsymbol{r})=\sum_i c_i\phi_i(\boldsymbol{r}),\quad c_i=\langle\phi_i(\boldsymbol{r}),\psi_0(\boldsymbol{r})\rangle,$$

the time evolution of (3.3) is given by

$$\psi(\boldsymbol{r},\tau)=e^{-\tau H}\psi_0(\boldsymbol{r})=\sum_i e^{-\tau E_i}c_i\phi_i(\boldsymbol{r})=e^{-\tau E_0}\sum_i e^{-\tau(E_i-E_0)}c_i\phi_i(\boldsymbol{r}). \tag{3.4}$$

Asymptotically, if the initial state ψ_0 is not orthogonal to the ground state ϕ_0, then after a sufficiently long time propagation, we get $\psi(\boldsymbol{r},\tau)\to e^{-\tau E_0}c_0\phi_0$ since the other exponents decay more rapidly. The wavefunction for the ground state then can be obtained by normalizing $\psi(\boldsymbol{r},\tau)$. To achieve the excited states, we can propagate several different initial states simultaneously in time with an orthonormalization process after each step which can be implemented by the modified Gram–Schmidt method.

Similar with the discretization in Section 2, the time evolution problem (3.3) will be solved in a bounded polyhedral domain Ω. Then the variational form of the equation (3.3) on Ω can be illustrated as: Find $\psi_{h,k}(t):[0,\infty)\to H_0^1(\Omega)$, $k=1,2,\cdots,N_o$, such that

$$\Big\langle\frac{d\psi_{h,k}(t)}{dt},\,\varphi\Big\rangle=-\Big\langle\frac{1}{2}\nabla\psi_{h,k}(t),\,\nabla\varphi\Big\rangle-\langle V_{eff}(t)\psi_{h,k}(t),\,\varphi\rangle,\quad \forall\varphi\in H_0^1(\Omega), \tag{3.5}$$

where $\langle\psi,\phi\rangle$ is short for $\langle\psi,\phi\rangle_\Omega$ here and after. With the finite element space $V_h\subset H_0^1(\Omega)$ constructing on Ω partitioned by $\mathcal{T}=\{\mathcal{T}_K,K=1,2,\cdots,N_{ele}\}$, the semi-discretized variational form of (3.3) becomes: Find $\psi_{h,k}(t):[0,\infty)\to V_h$, $k=1,2,\cdots,N_o$, such that

$$\Big\langle\frac{d\psi_{h,k}(t)}{dt},\,\varphi\Big\rangle=-\Big\langle\frac{1}{2}\nabla\psi_{h,k}(t),\,\nabla\varphi\Big\rangle-\langle V_{eff}(t)\psi_{h,k}(t),\,\varphi\rangle,\quad \forall\varphi\in V_h. \tag{3.6}$$

In this work, the backward Euler method is applied as the time scheme of the time-dependent equation (3.2) for its simplicity and the allowance for larger time step than the explicit methods. Then at imaginary time $t^{(m)}=mdt$, where dt is the time step, the full-discretized variational form for equation (3.2) after applying backward Euler time scheme can be formulated as: Find $\{\psi_k^{(m+1)}\}\in V_h$, $k=1,2,\cdots,N_o$ (for simplicity the subscript h is ignored here), such that $\forall\varphi\in V_h$

$$\Big\langle\frac{\psi_k^{(m+1)}-\psi_k^{(m)}}{dt},\varphi\Big\rangle=-\Big\langle\frac{1}{2}\nabla\psi_k^{(m+1)},\nabla\varphi\Big\rangle-\langle V_{eff}^{(m)}\psi_k^{(m+1)},\varphi\rangle, \tag{3.7}$$

here $V_{eff}^{(m)}$ is evaluated based on $\{\psi_k^{(m)}\}$. Rearranging the above formula and we have

$$\langle\psi_k^{(m+1)},\varphi\rangle+\frac{dt}{2}\langle\nabla\psi_k^{(m+1)},\nabla\varphi\rangle+dt\langle V_{eff}^{(m)}\psi_k^{(m+1)},\varphi\rangle=\langle\psi_k^{(m)},\varphi\rangle, \tag{3.8}$$

After expanding the wavefunctions by $\psi_k^{(m+1)}=\sum_{i=1}^{N_{bas}}\psi_{k,i}^{(m+1)}\varphi_i$ and denoting by $\Psi_k^{(m+1)}=(\psi_{k,1}^{(m+1)},\cdots,\psi_{k,N_{bas}}^{(m+1)})$, the following linear systems are obtained

$$C\Psi_k^{(m+1)}=B\Psi_k^{(m)},\quad k=1,\cdots,N_o, \tag{3.9}$$

where B is the mass matrix with form (2.15) and the entry of C can be written as

$$C_{i,j}=\int_\Omega\left\{\varphi_j\varphi_i+\frac{dt}{2}\nabla\varphi_j\cdot\nabla\varphi_i+dtV_{eff}^{(m)}\varphi_j\varphi_i\right\}d\boldsymbol{r}. \tag{3.10}$$

All the wavefunctions should be propagated from $t^{(m)}$ to $t^{(m+1)}$, which means the linear system (3.9) would be solved for N times with different right hand sides. The efficient algebraic multigrid (AMG) solver is adopted to solve the system (3.9).

Remark 3.1. It is noted that AMG solver is used in two processes in our algorithm, i.e., the generation of the Hartree potential, and the solving of the linear system in the backward Euler scheme. Since the radial mesh is generated in advance, and it is fixed during the whole simulation, the AMG solver is actually initialized at the beginning, and reused when needed.

A worth noticing phenomenon is that even given a random initial guess for the time evolution problem (3.2), the total energy will rapidly approach to the exact total ground state energy in the first several steps, which is observed in all the simulations in this work. We take the evaluation of total ground state energy of lithium atom for instance. In this simulation, the computational domain is set as $[-20,20]^3$ and the domain is partitioned following the strategy proposed in the end of Section 2. By letting $\beta=0.075,\gamma=8$, a computational mesh with 86862 degrees of freedom is obtained. On this mesh, we take time step as $dt=0.4$, and the equation is propagated forward until $t=50$. The convergence history for the total energy is displayed in the left of Fig. 5. From this figure, we can see that the total energy decreases to around -7 a.u. within propagation time $t=2$, which means it only take 4 to 5 steps, ITP can obtain a total energy less than 10% relative error. However, the speed of convergence becomes slow as time propagates which also can be found in the right of Fig. 5. To achieve accurate enough result using ITP alone, the equation should be propagated for a very long time.

3.3 ITP-SCF method

It is thus natural to utilize the fast convergence property of the imaginary time propagation method in the first several time propagations to combine with the SCF methods.

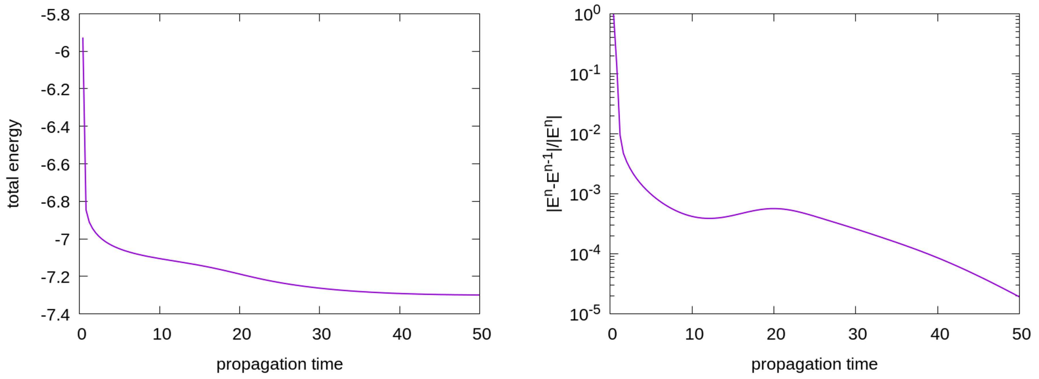

Figure 5: ITP to obtain the ground state of lithium atom. Left: the convergence history for total energy. Right: the relative error for the total energy versus the propagation time.

Algorithm 1 ITP-SCF method for ground state total energy calculation

```
Require: The chemical structure for the system, $m=1$
Generating a radial mesh $\mathcal{T}$ based on the given chemical structure using strategy presented in Section 2.
Building the finite element space $V_h$ on the initial mesh $\mathcal{T}$ and generating a random initial guess for wavefunctions $\{\psi_k^{(0)}, k=1,\cdots,N\}$.
do
    Propagating the time evolution problem (3.2) for one imaginary time steps
    Orthonormalizing the wavefunctions.
while $m++<m_{itp}$
Evaluating the total ground state energy $E_{tot}$.
do
    $E_{tot}^0=E_{tot}$.
    Updating the Hamiltonian using wavefunctions obtained in previous step.
    Solving the Kohn–Sham equation (2.1) to obtain the eigenpairs $\{\psi_i,\varepsilon_i\}$ and the total ground state energy $E_{tot}$.
while $|E_{tot}-E_{tot}^0|>tol$
Output the total energy $E_{tot}$.
```

Precisely, starting with the random initial guess, we can first use ITP method to propagate the equivalent time evolution problem within few imaginary time steps. Then the obtained results is set as the initial guess for the SCF iterations. Obviously the results from ITP method is closer to the exact solutions than the random initial guess. Therefore, with this method, the convergence for the SCF iteration will behave robuster and faster. Denoting this method by ITP-SCF method, the algorithm is described in Algorithm 1.

To improve the quality of the random initial guess, we only need to propagate the time evolution problem (3.2) for m_{itp} steps, where m_{itp} is a small number. For small

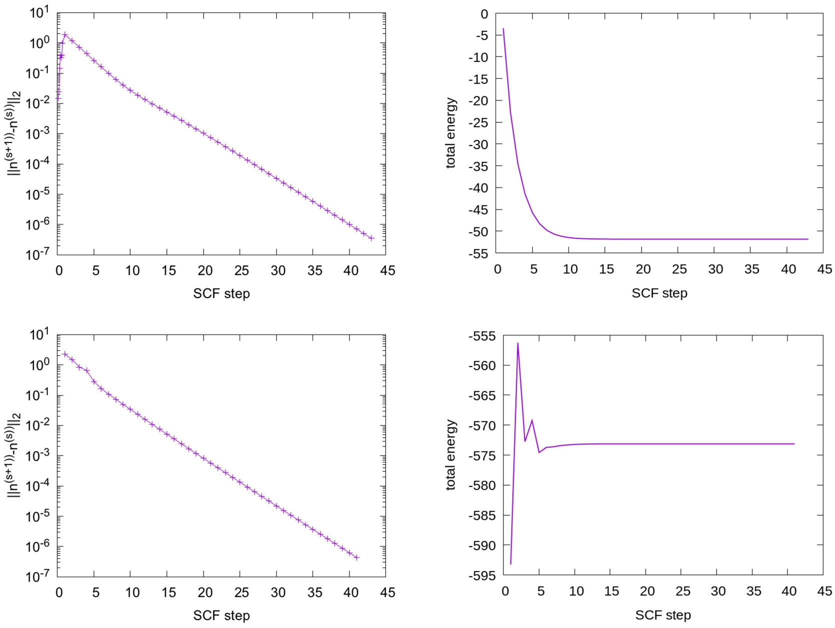

Figure 6: Ground state calculations of Li_7 and $C_{12}H_{10}N_2$ using ITP-SCF. Top left: density difference versus SCF iteration numbers. Top right: total energy of Li_7 versus SCF iteration numbers for ITP generated initial guess. Bottom left: density difference versus SCF iteration numbers. Bottom right: total energy of $C_{12}H_{10}N_2$ versus SCF iteration numbers for ITP generated initial guess.

systems within 10 electrons, m_{itp} can be chosen around 5 with the time step dt as large as possible. As the scale of the system tends to be large, the minimum mesh size of the radial mesh becomes small, which then results in the reduction of the allowed dt. In these cases, m_{itp} should be increased. Nevertheless, since the solving of the linear system (3.9) is efficient, the initial guess generating part using ITP contributes small proportion to the total computational cost, which can be controlled within 10% in practice.

Now we apply the presented ITP-SCF method to the simulations of two molecules of which the SCF method failed to converge as stated in the beginning of this section. The numerical results for Li_7 and $C_{12}H_{10}N_2$ are displayed in Fig. 6. Using the presented method, both SCF iterations for Li_7 and azobenzene are converged. From the two simulations starting with random initial guess, it is verified that the SCF iteration using ITP improved initial condition behaves robuster than the SCF iteration without using ITP. In the following section, the efficiency of the presented method is verified by plenty examples.

4 Numerical examples

In this section, numerous examples ranging from atoms to molecules are simulated to illustrate the efficiency of the presented method. The hardware is a Dell Precision T5610 workstation with Intel(R) Xeon(R) CPU E5-2630 v2 @ 2.60GHz (6 cores, 15 M cache), and 64 Gb of RAM. The software is the C++ library AFEABIC [4,5].

4.1 Lithium atom

Now we test the lithium atom system which has 3 electrons associated with two 1s orbitals and one 2s orbital. Similar with the above example, the SCF iteration with random initial guess and that with ITP-generated initial guess are implemented on a fixed mesh generated using the strategy presented in the end of Section 2. We compare the two methods in CPU time account to show the efficiency of the presented method. Numerical results can be found in Fig. 7. A radial mesh with 50 756 degrees of freedoms which is denser in the center than the radial mesh for simulation of hydrogen atom is generated using parameter $\beta = 0.125$, $\gamma = 8$. From the comparison of the CPU time which can be viewed in the bottom right of Fig. 7, more than 25% reduction of the CPU time for the method with ITP-generated initial guess compared with the method with random initial guess is observed. Note that compared with the simulation of hydrogen atom the proportion of computational cost for ITP decreases significantly. This is because the eigensolver for the linear system is called only once after the initial guess generation using ITP, while for the nonlinear system, the eigensolver will be called after each self-consistent field step until the convergence criterion for SCF is satisfied. As a result, the proportion of computational cost for eigensolver is increased and in contrast that for initial guess generating is decreased.

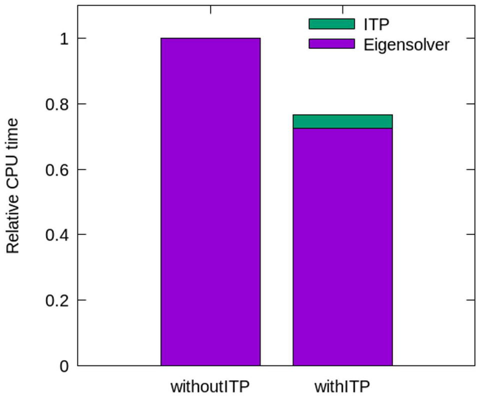

Figure 7: Relative CPU time comparison for example lithium atom.

4.2 Methane molecule

The presented method now is applied to the ground state simulation for the molecule methane (CH_4), which has 10 electrons in total. The numerical results are displayed in Fig. 8. Similar CPU time reduction with the above two examples can be observed from the

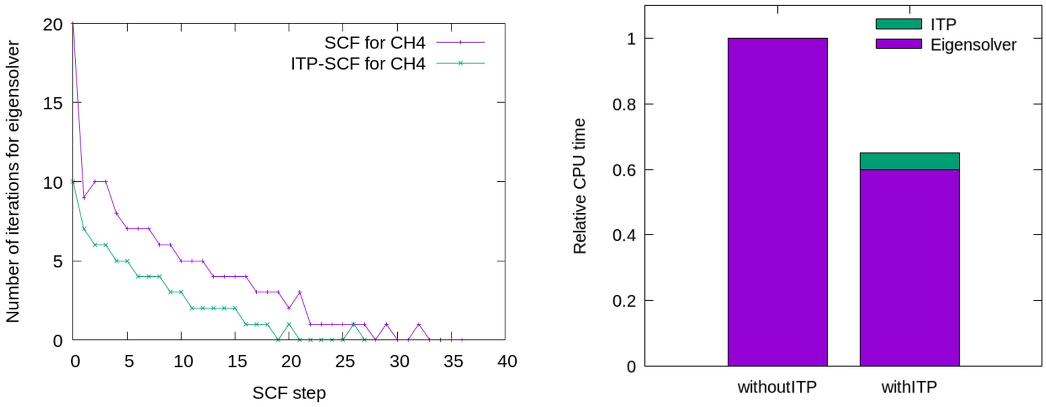

Figure 8: Results of methane. Left: comparison between SCF and ITP-SCF in terms of the number of iterations of the eigensolver LOBPCG. Right: relative CPU time comparisons.

right of Fig. 8. Moreover, we would like to explain the reduction of the computational cost from numerical aspect. Note that at each step of SCF, we solve the eigenvalue problem with the LOBPCG [21] method which is an iterative method. The number of iterations of LOBPCG at each SCF step is recorded as described in the left of Fig. 8. From this figure, we observe the following two points. Firstly, the ITP initial guess can reduce the number of iterations of LOBPCG. As displayed in the left of Fig. 8, at any SCF step, the number of iterations for LOBPCG in ITP-SCF method is less than or equal to that of LOBPCG in SCF method. Secondly, the total steps for SCF is decreased with ITP generated initial guess. In this simulation, 36 steps for SCF with random initial guess is required to achieve convergence, while only 27 SCF steps is required for method with ITP generated initial guess. In summary, the SCF with ITP initial guess require less computational cost for the SCF with random initial guess.

4.3 Summary for the numerical examples

In addition to the above three examples, a lot of simulations for ground state energies for different atoms and molecules are implemented. The computational cost reduction of the presented ITP-SCF method is demonstrated in the Table 1. Specifically, in the mesh size function (2.20) parameter γ is set to be 8 in all simulations and $\beta=0.125$ in most simulations except in lithium example $\beta=0.075$, in benzene (C_6H_6) example $\beta=0.175$ and in azobenzene ($C_{12}H_{10}N_2$) example $\beta=0.3$.

Note that the referenced energy for Li_7 (D_{5h}) is from [2], for Li_{14} which is on a $2\times 1\times 1$ body-centered-cubic (bcc) lattice is from [11], and for other atoms or molecules are from the database CCCBDB [12]. Numerical results of Li_7, Li_{14} and $C_{12}H_{10}N_2$ show that the presented ITP-SCF method behaves robuster than the SCF method which fails to converge in these simulations. Furthermore, ITP-SCF method is efficient than SCF for the other simulations list in Table 1 that 20% to 30% computational time can be saved.

Table 1: Results for atoms and molecules. CPU time are compared for ITP-SCF method and SCF method in unit second (s) of each simulation. "#DOF" means the number of degrees of freedom. "Referenced" represents the referenced total ground state energy. Column "Percentage" stands for the CPU time of ITP-SCF dividing by the CPU time of SCF method. "FTC" means fail to converge. Column "Energy" are values obtained using ITP-SCF.

	#DOF	ITP-SCF(s)	Percentage	SCF(s)	Energy	Referenced
Li	50 756	385	77.94%	494	-7.307	-7.395
Li_2	94 552	1431	75.98%	1884	-14.699	-14.727
Li_3	137 711	5107	81.42%	6272	-22.059	-22.100
Li_7	264 405	33568	-	FTC	-51.811	-52.26
Li_{14}	607 401	297918	-	FTC	-103.73	-105.28
LiH	63 677	552	76.46%	723	-7.91	-7.933
CH_4	141 444	2313	65.79%	3515	-40.054	-40.262
C_6H_6	108 351	14534	82.78%	17556	-230.847	-230.859
$C_{12}H_{10}N_2$	160 291	85255	-	FTC	-573.117	-572.53

5 Conclusion

In this paper, an efficient ITP–SCF numerical framework is presented for solving all-electron Kohn–Sham model. Numerical experiments show that with the ITP-computed results, the presented method can speed up the convergence of SCF and effectively avoid the failure of SCF iteration. However, since the presented radial mesh is fixed during the simulation, to obtain more accurate result one has to regenerate the mesh and implement the algorithm again. Furthermore, for dynamical simulations such as Born–Oppenheimer molecule dynamics, the presented method is less of flexibility since each movement of the nucleus would also result in a reconstruction of the computational mesh. Hopefully, these can be handled using h-adaptive finite element framework [4,25] which is able to provide a dynamic mesh generating way to improve the quality of the mesh, and this would be our future work.

Acknowledgments

The authors would like to thank the support from FDCT 029/2016/A1 of Macao S.A.R., MYRG2019-00154-FST and MYRG2017-00189-FST from University of Macau, and National Natural Science Foundation of China (Grants No. 11922120, 11871489 and 11401608).

A Generation of the radial mesh

The three-dimensional mesh with mesh size function (2.20) can be efficiently and rapidly generated by the finite element mesh generator Gmsh [16]. In addition to the casual way to generate the uniform three-dimensional mesh in Gmsh version 3.0, one just needs to calculate the distance from the nucleus by using the variable `Attractor`. Then with the help of another variable `MathEval` in which the mesh size function (2.20) is implemented, the radial mesh is generated. An example code for this part in `LiH.geo` file is displayed below, from which the mesh for lithium hydride (LiH) with two nuclei can be generated, see Fig. 9. The lithium nucleus locates at point $(1.50069,0,0)$ and the hydrogen nucleus locates at point $(-1.50069,0,0)$ on the computational domain. A denser mesh grid distribution can be found around the lithium nucleus compared with the region around the hydrogen nucleus.

```
beta = 0.125;
gamma = 8;
lmin = gamma/1000; // the allowed smallest length

// Vertices for the computational domain
Point(1) = {-20, -20, -20,gamma};
Point(2) = {20, -20, -20,gamma};
Point(3) = {20, 20, -20,gamma};
Point(4) = {-20, 20, -20,gamma};
Point(5) = {-20, -20, 20,gamma};
Point(6) = {20, -20, 20,gamma};
Point(7) = {20, 20, 20,gamma};
Point(8) = {-20, 20, 20,gamma};
// Nuclear positions
Point(9) = {1.50069, 0, 0,lmin};
Point(10) = {-1.50069, 0, 0,lmin};

Line(1) = {8, 7};Line(2) = {7, 6};Line(3) = {6, 5};
Line(4) = {5, 8};Line(5) = {3, 2};Line(6) = {2, 1};
Line(7) = {1, 4};Line(8) = {4, 3};Line(9) = {3, 7};
Line(10) = {2, 6};Line(11) = {8, 4};Line(12) = {5, 1};

Line Loop(13) = {9, 2, -10, -5}; Plane Surface(14) = {13};
Line Loop(15) = {1, -9, -8, -11};Plane Surface(16) = {15};
Line Loop(17) = {8, 5, 6, 7}; Plane Surface(18) = {17};
Line Loop(19) = {3, 12, -6, 10}; Plane Surface(20) = {19};
Line Loop(21) = {12, 7, -11, -4};Plane Surface(22) = {21};
Line Loop(23) = {2, 3, 4, 1}; Plane Surface(24) = {-23};
Surface Loop(25) = {24, 14, 16, 18, 20, 22}; Volume(26) = {25};
Physical Surface(27) = {16}; Physical Surface(28) = {20};
Physical Volume(29) = {26}; Physical Surface(30) = {24};
Physical Surface(31) = {18};
```

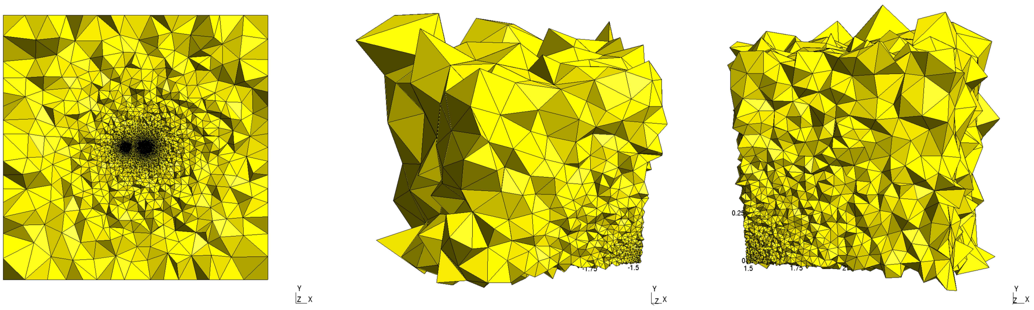

Figure 9: Left: the three dimensional mesh for molecule LiH using the mesh size function (2.20) on the domain $[-20,20]^3$. Middle: the mesh around the hydrogen nucleus $(-1.5,0,0)$ in X-Y plane $[-2.5,-1.5]\times[0,1]$, on which the element shapes are kept. Right: the mesh around the lithium nucleus $(1.5,0,0)$ in X-Y plane $[1.5,2.5]\times[0,1]$. Generated by Gmsh [16].

```
// Nucleus locates on a node
Point {9,10} In Volume {26};

// Attractor field returns the distance to the point
Field[1] = Attractor; Field[1].NodesList = {9};
Field[2] = Attractor; Field[2].NodesList = {10};

// Matheval field returns the mesh size function
Field[3] = MathEval;
Field[3].F = Sprintf("F1^(6/5)*0.644394*%g + %g", beta, lmin);
Field[4] = MathEval;
Field[4].F = Sprintf("F2^(6/5)*1*%g + %g", beta, lmin);
Field[5] = MathEval;
Field[5].F = Sprintf("%g", gamma);

// Use the minimum of size functions
Field[6] = Min;
Field[6].FieldsList = {3,4,5};
Background Field = 6;
```

References

[1] S. Agmon. Lectures on Exponential Decay of Solutions of Second-Order Elliptic Equations: Bounds on Eigenfunctions of N-Body Schröodinger Operations. (MN-29). Princeton University Press, 2014.

[2] A. N. Alexandrova and A. I. Boldyrev. Search for the Li n 0/+1/-1 (n=5-7) lowest-energy structures using the ab initio gradient embedded genetic algorithm (GEGA). Elucidation of

the chemical bonding in the lithium clusters. Journal of Chemical Theory and Computation, 1(4):566–580, 2005.

[3] X. Antoine, C. Besse, R. Duboscq, and V. Rispoli. Acceleration of the imaginary time method for spectrally computing the stationary states of Gross–Pitaevskii equations. Computer Physics Communications, 219:70–78, 2017.

[4] G. Bao, G. 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.

[5] G. Bao, G. Hu, and D. Liu. Numerical solution of the KohnSham equation by finite element methods with an adaptive mesh redistribution technique. Journal of Scientific Computing, 55(2):372–391, 2013.

[6] W. Bao and Y. Cai. Mathematical models and numerical methods for spinor Bose–Einstein condensates. Communications in Computational Physics, 24(4):899–965, 2018.

[7] S. A. Chin, S. Janecek, and E. Krotscheck. Any order imaginary time propagation method for solving the Schrödinger equation. Chemical Physics Letters, 470(4):342–346, 2009.

[8] M. L. Chiofalo, S. Succi, and M. P. Tosi. Ground state of trapped interacting Bose–Einstein condensates by an explicit imaginary-time algorithm. Physical Review E, 62(5):7438, 2000.

[9] P. G. Ciarlet. The Finite Element Method For Elliptic Problems, 2002.

[10] A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, G. N. Miranda, and J. W. Ruge. Robustness and scalability of algebraic multigrid. SIAM Journal on Scientific Computing, 21(5):1886–1908, 2000.

[11] K. Doll, N. M. Harrison, and V. R. Saunders. A density functional study of lithium bulk and surfaces. Journal of Physics: Condensed Matter, 11(26):5007, 1999.

[12] R. D. Johnson III (Ed.). NIST Computational Chemistry Comparison and Benchmark Database, NIST Standard Reference Database Number 101. Release 20, April 2019. http://cccbdb.nist.gov/, DOI:10.18434/T47C7Z.

[13] C. F. Fischer. Hartree–Fock Method for Atoms. A Numerical Approach. John Wiley and Sons, Inc., New York, 1977.

[14] C. Flamant, G. Kolesov, E. Manousakis, and E. Kaxiras. Imaginary-time time-dependent density functional theory and its application for robust convergence of electronic states. Journal of Chemical Theory and Computation, 15(11):6036–6045, 2019.

[15] I. Fried. Bounds on the spectral and maximum norms of the finite element stiffness, flexibility and mass matrices. International Journal of Solids and Structures, 9(9):1013–1034, 1973.

[16] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering, 79(11):1309–1331, 2009.

[17] W. Huang and R. D. Russell. Adaptive Moving Mesh Methods, volume 174. Springer Science & Business Media, 2010.

[18] D. Jacquemin, J. Preat, E. A. Perpète, D. P. Vercauteren, J.-M. André, I. Ciofini, and C. Adamo. Absorption spectra of azobenzenes simulated with time-dependent density functional theory. International Journal of Quantum Chemistry, 111(15):4224–4240, 2011.

[19] R. O. Jones, A. I. Lichtenstein, and J. Hutter. Density functional study of structure and bonding in lithium clusters Li n and their oxides Li n O. The Journal of Chemical Physics, 106(11):4566–4574, 1997.

[20] B. Kanungo and V. Gavini. Large-scale all-electron density functional theory calculations using an enriched finite-element basis. Physical Review B, 95(3):035112, 2017.

[21] A. V. Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. SIAM Journal on Scientific Computing, 23(2):517–

541, 2001.

[22] A. V. Knyazev, M. E. Argentati, I. Lashuk, and E. E. Ovtchinnikov. Block locally optimal preconditioned eigenvalue xolvers (BLOPEX) in HYPRE and PETSc. SIAM Journal on Scientific Computing, 29(5):2224–2239, 2007.

[23] S. Kotochigova, Z. H. Levine, E. L. Shirley, M. D. Stiles, and C. W. Clark. Local-density-functional calculations of the energy of atoms. Physical Review A, 55(1):191, 1997.

[24] J. Koutecký and Vlasta Bonačić. On convergence difficulties in the iterative Hartree–Fock procedure. The Journal of Chemical Physics, 55(5):2408–2413, 1971.

[25] Y. Kuang and G. Hu. An adaptive FEM with ITP approach for steady Schrödinger equation. International Journal of Computer Mathematics, 95(1):187–201, 2018.

[26] L. Lehtovaara, V. Havu, and M. Puska. All-electron density functional theory and time-dependent density functional theory with high-order finite elements. The Journal of Chemical Physics, 131(5):054103, 2009.

[27] L. Lehtovaara, J. Toivanen, and J. Eloranta. Solution of time-independent Schrödinger equation by the imaginary time propagation method. Journal of Computational Physics, 221(1):148–157, 2007.

[28] P. R. Levashov, G. V. Sin'ko, N. A. Smirnov, D. V. Minakov, O. P. Shemyakin, and K. V. Khishchenko. Pseudopotential and full-electron DFT calculations of thermodynamic properties of electrons in metals and semiempirical equations of state. Journal of Physics: Condensed Matter, 22(50):505501, 2010.

[29] Z. H. Levine and J. W. Wilkins. An energy-minimizing mesh for the Schröodinger equation. Journal of Computational Physics, 83(2):361–372, 1989.

[30] Y. Maday. h-P finite element approximation for full-potential electronic structure calculations. In: Partial Differential Equations: Theory, Control and Approximation, pages 349–377. Springer, 2014.

[31] 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.

[32] P. Motamarri and V. Gavini. Configurational forces in electronic structure calculations using Kohn–Sham density functional theory. Physical Review B, 97(16):165132, 2018.

[33] P. Motamarri, M. R Nowak, K. Leiter, J. Knap, and V. Gavini. Higher-order adaptive finite-element methods for Kohn–Sham density functional theory. Journal of Computational Physics, 253:308–343, 2013.

[34] A. R. Oganov and P. I. Dorogokupets. All-electron and pseudopotential study of MgO: Equation of state, anharmonicity, and stability. Physical Review B, 67(22):224110, 2003.

[35] 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.

[36] C. Yang, W. Gao, and J. C. 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.

[37] C. Yang, J. C. Meza, and L.-W. Wang. A trust region direct constrained minimization algorithm for the Kohn–Sham equation. SIAM Journal on Scientific Computing, 29(5):1854–1875, 2007.

[38] J. Yu. Symmetric gaussian quadrature formulae for tetrahedronal regions. Computer Methods in Applied Mechanics and Engineering, 43(3):349–353, 1984.