Accelerating Parallel First-Principles Excited-State Calculation by Low-Rank Approximation with K-Means Clustering

Qingcai Jiang
jqc@mail.ustc.edu.cn
University of Science and Technology of China, Hefei, Anhui, China

Junshi Chen
cjuns@ustc.edu.cn
University of Science and Technology of China, Hefei, Anhui, China

Lingyun Wan
wanly@mail.ustc.edu.cn
University of Science and Technology of China, Hefei, Anhui, China

Xinming Qin
xmqin03@ustc.edu.cn
University of Science and Technology of China, Hefei, Anhui, China

Jielan Li
jielanli@mail.ustc.edu.cn
University of Science and Technology of China, Hefei, Anhui, China

Jie Liu
liujie86@ustc.edu.cn
University of Science and Technology of China, Hefei, Anhui, China

Hong An*
han@ustc.edu.cn
University of Science and Technology of China, Hefei, Anhui, China

Wei Hu*
whuustc@ustc.edu.cn
University of Science and Technology of China, Hefei, Anhui, China and Lawrence Berkeley National Laboratory Berkeley, United States

Jinlong Yang
jlyang@ustc.edu.cn
University of Science and Technology of China, Hefei, Anhui, China

ABSTRACT

First-principles time-dependent density functional theory (TDDFT) is a powerful tool to accurately describe the excited-state properties of molecules and solids in condensed matter physics, computational chemistry and materials science. However, a perceived drawback in TDDFT calculations is its ultrahigh computational cost $O(N^5 \sim N^6)$ and large memory usage $O(N^4)$ especially for plane-wave basis set, confining its applications to large systems containing thousands of atoms. Here, we present a massively parallel implementation of linear-response TDDFT (LR-TDDFT) and reduce the complexity to $O(N^3)$ by combining K-Means clustering based low-rank approximation with iterative eigensolve algorithm. Furthermore, we carefully design the parallel data and task distribution schemes to accommodate with the physical nature in different steps of the computation, also, several optimization methods are employed to effectively handle the matrix operations and data communications of constructing and diagonalizing the LR-TDDFT Hamiltonian. In particular, our method can significantly reduce the cost of computation and memory by nearly 2 orders of magnitude compared to conventional LR-TDDFT calculations. Numerical results demonstrate that our implementation can gain an overall speedup of 10x and efficiently scale up to 12,288 CPU cores for large systems up to 4,096 atoms within dozens of seconds.

*Corresponding authors.

ICPP '22, August 29-September 1, 2022, Bordeaux, France

ACM ISBN 978-1-4503-9733-9/22/08…$15.00
https://doi.org/10.1145/3545008.3545092

CCS CONCEPTS

• **Applied computing** → **Chemistry**; • **Computing methodologies** → *Parallel computing methodologies.*

KEYWORDS

Quantum mechanic calculation, Time-dependent density functional theory, Linear-response, Iterative eigensolver, Low-rank approximation, Parallel implementation

ACM Reference Format:
Qingcai Jiang, Junshi Chen, Lingyun Wan, Xinming Qin, Jielan Li, Jie Liu, Hong An*, Wei Hu*, and Jinlong Yang. 2022. Accelerating Parallel First-Principles Excited-State Calculation by Low-Rank Approximation with K-Means Clustering. In *51st International Conference on Parallel Processing (ICPP '22), August 29-September 1, 2022, Bordeaux, France.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3545008.3545092

1 INTRODUCTION

First-principles calculations based on the Kohn-Sham density functional theory (DFT) [17] has extensive applications in chemistry and material science. To enable computation-based design of new materials and predict their peculiar properties in different types of fields with high accuracy, developing large-scale DFT and time-dependent DFT (TDDFT) [26] methods both in ground-state and excited-state simulations is of significant impact.

Within the framework of TDDFT, there are generally two methods for solving the time-dependent Schrödinger equation [4]. The first approach performs the time evolution by molecular dynamics on real-space grids [31], referred to as real-time TDDFT (RT-TDDFT). Another approach is the most common formula to adopt excited-state properties, which evaluates exactly many-body Schrödinger equation of the time-dependent linear response function formulated in the frequency domain. It exploits Fourier transform to acquire excitation energies and corresponding wavefunctions, referred to as linear-response TDDFT (LR-TDDFT).

Within LR-TDDFT framework, Casida equation [6] is the most commonly used formula to describe the excitation energy and corresponding wavefunctions. To solve the Casida equation, the most time-consuming parts in the LR-TDDFT calculations can be summarized as two parts, one is to explicitly construct the LR-TDDFT Hamiltonian with the complexity of $O(N_e^5)$ with respect to the number of electrons in the system N_e, and the other is to diagonalize the LR-TDDFT Hamiltonian with the complexity of $O(N_e^6)$. As the system expands, the computational and memory cost of LR-TDDFT calculations in a general CPU platform becomes prohibitively expensive, especially on large complete basis sets, such as plane-wave basis set. Therefore, exploring the excited-state properties of systems with thousands of atoms using the LR-TDDFT method is still a very tough task.

The state of the art LR-TDDFT calculations in CP2K with Gaussian-type orbitals (GTOs) allow the study of large-scale systems including aluminosilicate imolite nanotubes, in addition to surface and bulk vacancy defects in MgO and HfO_2 with nearly 1,000 atoms [27]. However, the above software is all implemented under localized basis sets, which are not commensurate with desired accuracy especially when the system is complex. In particular, there has been no breakthrough for a long time with regard to standard plane-wave basis set because of ultra-high computational and memory cost in the LR-TDDFT calculations, which hinders excited-state electronic structure exploration for large-scale periodic systems containing thousands of atoms.

Fortunately, this situation can be immensely improved thanks to appearance of new algorithmic methods and modern high performance computing (HPC) facilitates. For example, low-rank decomposition methods like density fitting approximation, also known as the resolution of identity algorithms [24], can help not only accelerate the construction of LR-TDDFT Hamiltonian but also significantly lower the memory cost. Furthermore, the iterative subspace eigensolver algorithms, such as Davidson [8] and LOBPCG [11], have been successfully applied to simulate excited-state properties by giving an estimation of the lowest k eigenvalues with a favorable computational cost of $O(kN_e^4)$ (k is the number of desired lowest eigenvalues and corresponding eigenvectors). Given all these advances, the large-scale excited-state calculations of LR-TDDFT framework with plane-wave basis set become reachable.

In this work, we present the recently developed low-rank decomposition methods like QR factorization with column pivoting (QRCP) based interpolative separable density fitting (ISDF) [19], which provides us an efficient and accurate way to reduce the ultra-high computational and memory cost during the construction of Hamiltonian in the simulation of LR-TDDFT. To further reduce the time cost and explore parallelism, we propose a K-Means based parallel ISDF algorithm to avoid expensive time costs during QRCP procedure and the terrible parallelism that follows. Also, we reduce the computational and memory cost by implicitly constructing and iteratively diagonalizing the Hamiltonian. Moreover, we perform extensive numerical experiments on Cori supercomputer, the Cray XC40 system in the National Energy Research Scientific Computing Center (NERSC). The results show that our method can gain an overall 10x speedup with negligible error, reduce a large amount of memory footprint, and efficiently scale to massive computation cores, thereby enabling us to study the excited-state properties with larger scale than current state of the art.

The main contributions of this work can be summarized as follows:

(1) A series of parallel algorithms, including K-Means based low-rank decomposition, iterative eigenvalue solver and implicit Hamiltonian method are implemented to reduce the computation and memory cost, expand the system size and accelerate the computation steps in the LR-TDDFT calculations.
(2) We demonstrate that with our algorithms along with parallel implementations and optimizations, we can study the three-dimensional semiconducting silicon systems with 4,096 atoms, this result exceeds current state of the art both in parallel scale and the system scale.
(3) Under extensive experiments, we show that our method can achieve high scalability, with regard to both strong scaling and weak scaling. Also, our method will remain high accuracy even in strong correlation systems especially for two-dimensional magic-angle twisted bilayer graphene (MATBG), thereby providing an insight into the physical nature of complex systems.
(4) We have opensourced our software at https://bitbucket.org/berkeleylab/scales/src/lrtddft/ in the hope that our approach can provide insight for relevant high-performance applications with the same computational characteristics.

The rest of this article is organized as follows. We review related work in section 2. We describe the algorithm for performing LR-TDDFT calculations with plane-wave basis set in section 3. The mathematical methods are introduced in section 4. The parallel implementation is presented in section 5. Then we report the numerical results and following analysis in section 6, and we conclude this work in section 7.

2 RELATED WORKS

Although different types of basis sets can be used in the DFT and TDDFT calculations, plane-wave (PW) basis set [22] in the broadest sense seems current to be the most advantageous for complex periodic solid systems in condensed matter physics and materials science, compared to small localized atomic orbitals (AO) [4] basis set, which is more suitable for molecular systems in quantum chemistry. In particular, PW basis set is complete and allows a faithful analytical evaluation of the total energy, atomic forces, and other physical quantities. But the computational cost of DFT within plane-wave basis set increases rapidly with respect to the number of electrons in the systems because the number of PW basis set is much expensive than the case of small localized AO basis sets ($N_{\mathrm{PW}} \approx 100 \times N_{\mathrm{AO}}$), which hinders its practical applications to large systems containing thousands of atoms. For example, traditional ground-state DFT calculations with plane-wave basis set are exorbitantly expensive due to $O(N_e^3)$ scaling computational and memory complexity with respect to the number of electrons $N_e \approx 1,000$ [30].

Although it's quite difficult for this software with plane-wave basis to expand to large systems, large-scale TDDFT excited-state electronic structure calculations within small localized basis sets (like atomic and Gaussian basis set) for molecular systems have been implemented recently, such as NWChem [28] and QChem [25].

Table 1: Performance comparison of massively parallel excited-state simulation software packages on modern heterogeneous supercomputers, involving different HPC codes (NWChem, CP2K, PWDFT and BerkeleyGW) within different types of basis sets (Plane-wave (PW), Gaussian and mixed Gaussian and plane wave (GPW)).

HPC Software	Year	Theory	Basis set	Method	System	#atoms	Architecture	Reference
NWChem	2016	LR-TDDFT	Gaussian	Explicit	Water molecules	1,890	Intel Xeon	[32]
CP2K	2019	LR-TDDFT	GPW	Explicit	MgO; HfO_2	1,000	Intel Xeon	[27]
PWDFT	2019	RT-TDDFT	PW	Implicit	Silicon	1,536	V100 GPU	[20]
BerkeleyGW	2020	GW	PW	Explicit	Silicon	2,742	V100 GPU	[9]
PWDFT	2021	LR-TDDFT	PW	Implicit	Silicon; Graphene	4,096	Intel Xeon	This work

In detail, NWChem was used to study the excited-state properties of the system containing 120 atoms with 1840 6-311G Gaussian basis set ($Au_{20}Ne_{100}$), and in that work, NWChem efficiently scales to 2,250 CPU cores on the CINECA supercomputer.

For periodic solid systems, the GW approximation derived from the Green's function has also become a powerful formalism for studying single-electron excitations of molecules and the quasi-particle band gaps of solids within many-body effects. Recently, large-scale GW calculations containing 2,742 atoms within the plane-wave basis set in BerkeleyGW [9] have also been implemented in the Summit supercomputer.

3 THEORETICAL ALGORITHMS OF LR-TDDFT

The LR-TDDFT calculations consists of two parts: (1) constructing Hamiltonian and (2) diagonalizing Hamiltonian.

The Hamiltonian we need to construct in LR-TDDFT calculations has the following numerical structure:

$$H = \begin{bmatrix} D + 2V_{\mathrm{Hxc}} & 2W_{\mathrm{Hxc}} \\ -2W_{\mathrm{Hxc}} & -D - 2V_{\mathrm{Hxc}} \end{bmatrix}, \tag{1}$$

where $D\left(i_v i_c, j_v j_c\right) = \left(\varepsilon_{i_c} - \varepsilon_{i_v}\right) \delta_{i_v j_v} \delta_{i_c j_c}$, is an $N_{cv} \times N_{cv}$ ($N_{cv} = N_c \times N_v$, in which N_c is the number of conduction orbitals, N_v is the number of valence orbitals and δ denotes Dirac delta function) matrix. These orbital energies ($\varepsilon_{i_v}(i_v = 1, \ldots, N_v)$ and $\varepsilon_{i_c}(i_c = 1, \ldots, N_c)$) and corresponding orbitals are typically obtained via ground-state Kohn-Sham DFT calculations. The V_{Hxc} and W_{Hxc} matrices represent the Hartree-exchange-correlation integrals.

With the Tamm-Dancoff approximation (TDA) [12], W_{Hxc} matrix is neglectable so the Hamiltonian matrix has the form

$$H = D + 2V_{\mathrm{Hxc}}. \tag{2}$$

In discrete cases, V_{Hxc} is defined as the multiplication of the matrix f_{Hxc} and transposed block face-splitting product (or Block column-wise version of the Khatri-Rao product) matrix [21] $P_{vc} = \{\psi_{i_v}(\mathbf{r})\psi_{i_c}(\mathbf{r})\}$. $\psi_{i_v}(\mathbf{r})$ and $\psi_{i_c}(\mathbf{r})$ stand for the valence and conduction orbitals in real space ($\{\mathbf{r_i}\}_{i=1}^{N_r}$, N_r denotes the number of real space grid points during the calculations).

$$V_{\mathrm{Hxc}} = P_{vc}^{\dagger} f_{\mathrm{Hxc}} P_{vc}, \tag{3}$$

here f_{Hxc} is the kernel of Hartree-exchange-correlation operator

$$\begin{aligned} f_{\mathrm{Hxc}}\left(\mathbf{r}, \mathbf{r}'\right) &= f_{\mathrm{H}}\left(\mathbf{r}, \mathbf{r}'\right) + f_{\mathrm{xc}}[n]\left(\mathbf{r}, \mathbf{r}'\right) \\ &= \frac{1}{|\mathbf{r} - \mathbf{r}'|} + \frac{\delta V_{\mathrm{xc}}[n](\mathbf{r})}{\delta n\left(\mathbf{r}'\right)}, \end{aligned} \tag{4}$$

where $n(\mathbf{r}) = \sum_{i=1}^{N_v} |\psi_i(\mathbf{r})|^2$ encodes the electronic density and f_{xc} is the exchange-correlation potential in LR-TDDFT calculations.

After constructing the Hamiltonian matrix, like other conventional KS-DFT calculations, we need to diagonalize it explicitly to get the excitation wavefunctions X and corresponding excitation energies λ. In our naïve implementation, diagonalization is realized by SYEVD routine in ScaLAPACK [7] with ultra-high complexity of $O(N_v^3 N_c^3) \sim O(N_e^6)$. We explain our implementation in Algorithm 1.

We remark that N_r is generally much larger (1,000×) than N_e and $N_v \approx N_c \approx N_e$ for large normalized plane-wave basis set. And we summarize the computational and memory cost of constructing and diagonalizing Hamiltonian in Table 2.

Algorithm 1 The pseudocode for the LR-TDDFT calculations.

Input: Ground-state energies ϵ_i, wavefunctions $\psi_\mu(\mathbf{r})$ and $\psi_\nu(\mathbf{r})$ distributed according to the row index.

1: **for** each MPI process **do**
2: Initialize $P_{vc}(r) = \{\psi_\mu(r)\psi_\nu(r)\}$ in real space;
3: Carry MPI_Alltoall to wavefunctions Ψ to transfer data distribution scheme from row block partition to column block partition;
4: Transfer $P_{vc}(g)$ into reciprocal space via fast Fourier transform (FFT);
5: Apply the Hartree potential operator in reciprocal space and transfer it back into real space $v_{\mathrm{H}}(g)P_{vc}(g)$;
6: Carry out MPI_Alltoall to wavefunctions Ψ to transfer data distribution scheme from column block partition to row block partition;
7: Compute the Hartree-exchange-correlation integrals V_{Hxc} in real space via general matrix multiply (GEMM);
8: Summarize V_{Hxc} within all MPI tasks by MPI_Allreduce;
9: **end for**
10: Obtain Hamiltonian by computing the difference of Kohn-Sham energy eigenvalues;
11: Diagonalize the Hamiltonian;

Output: Excited-state energies $\{\lambda_i\}$ and wavefunctions $\{x_{ij}\}$

4 ALGORITHM INNOVATIONS

As we can see from Table 2, the Hamiltonian matrix occupies a large fraction of the memory footprint. For example, when $N_c = N_v = 256$ and double-precision is used during the calculation, each process will hold a matrix of 32 GB, which brings about ultra-high

Table 2: Computation and memory complexity for constructing and diagonalizing the LR-TDDFT Hamiltonian matrix with the naïve LR-TDDFT code. Within the plane-wave basis set, $N_r \approx 1,000 \times N_e$ and $N_v \approx N_c \approx N_e$ in the table.

LR-TDDFT		Computation	Memory
Constructing Hamiltonian	Face-splitting product of conduction-valence orbitals	$O(N_v N_c N_r)$	$O(N_v N_c N_r)$
	Fast fourier transform (FFT)	$O(N_v^2 N_c^2 N_r)$	$O(N_v N_c N_r)$
	General matrix multiply (GEMM)	$O(N_v^2 N_c^2 N_r)$	$O(N_v^2 N_c^2)$
	f_{Hxc} kernel	$O(N_v N_c N_r)$	$O(N_v N_c N_r)$
Diagonalizing Hamiltonian	ScaLAPACK::Syevd	$O(N_v^3 N_c^3)$	$O(N_v^2 N_c^2)$

computation cost and communication overhead, hence limiting the studied system size to expand.

4.1 Low-Rank Approximation in LR-TDDFT by ISDF

As shown in Algorithm 1, all computational operations are based on the two-electron integrals $\{\rho_{ij}(\mathbf{r}) := \psi_i(\mathbf{r})\phi_j(\mathbf{r})\}_{1\le i\le m, 1\le j\le n}$ (orbital pair product). But when we look into the matrix $P_{vc}(r)$ constructed from valence and conduction orbitals Ψ and Φ, the information beneath it is commonly markedly redundant. In other words, we can use several much smaller matrices to represent it. So exploiting the numerical rank deficiency of the pair products is the cornerstone to reducing the time cost of this operation and all the related computing-intensive operations. Several low-rank tensor approximations have been proposed, including the Resolution-of-the-identity (RI) [3] approximation and the interpolative separable density fitting (ISDF) [23] decomposition. The key spirit of these low-rank approximations is to carefully choose a set of interpolation points N_μ ($N_\mu = cN_r$, where c is a small preconstant) from all the real space grid points N_r in advance, which can give an accurate representation of all orbital-pair products. So we can represent $\psi_i(\mathbf{r})\phi_j(\mathbf{r})$ with the multiplication of two matrices. One matrix can be viewed as the expansion coefficients matrix $C_\mu^{ij}{}_{1\le\mu\le N_\mu}$ (a third-order tensor), whose each single row is extracted from the two-electron integrals matrix according to interpolation points $\hat{r}_\mu$ for $\mu = 1, \ldots, N_\mu$. The other matrix can be viewed as numerical auxiliary basis functions (ABFs) $\{\zeta_\mu(\mathbf{r})\}_{1\le\mu\le N_\mu}$, for which we will refer to as the interpolating vectors in the rest of the article, so that

$$\psi_i(\mathbf{r})\phi_j(\mathbf{r}) \approx \sum_{\mu=1}^{N_\mu} \zeta_\mu(\mathbf{r}) C_\mu^{ij}. \tag{5}$$

Furthermore, the central idea of the ISDF decomposition different from other low-rank tensor approximations is to decompose the third-order tensor $C_\mu^{ij}{}_{1\le\mu\le N_\mu}$ again, into a transposed block face-splitting product of two matrices $C_\mu^{ij} = \psi_i(\hat{\mathbf{r}}_\mu)\,\phi_j(\hat{\mathbf{r}}_\mu)$. The decomposition scheme of ISDF is shown in Figure 1.

Thus the Hamiltonian matrix can be rewritten as:

$$H = D + 2C^\dagger(\tilde{V}_{Hxc}C), \tag{6}$$

under ISDF with the auxiliary basis set, the Hartree exchange-correlation integrals V_{Hxc} can be projected in this form:

$$\tilde{V}_{Hxc} = \zeta_\mu^\dagger(f_{\mathrm{Hxc}}\zeta_\mu). \tag{7}$$

The parentheses in Equation 6 and 7 represent the order of multiplication.

The time cost of constructing the Hamiltonian (not including ISDF procedure, for which we will later discuss) is now significantly reduced to $O(N_r N_\mu^2 + N_\mu N_v^2 N_c^2)$. Under the ISDF basis set, the first term accounts for the time cost of computing $\tilde{V}_{\mathrm{Hxc}}$ and the second term is the cost of three matrix multiplications and FFTs.

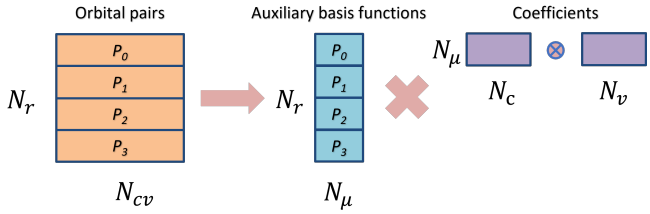

Figure 1: The ISDF decomposition and its parallel strategy for LR-TDDFT. Each process holds a segment of orbital pairs and auxiliary basis functions, and every process holds the same copy of coefficients.

In section 4.1.1 and 4.1.2, we will discuss the main procedures in the ISDF approximation as a background of our improved method.

4.1.1 Selecting the Interpolation Points of ISDF. Consider a discretized matrix Z of size $N_r \times (N_{cv})$ and find N_μ rows of Z so that the remaining rows of Z can be approximated by the linear combination of the selected N_μ rows. This procedure is so-called interpolative decomposition, and one traditional way is using the randomized sampling QR factorization with column pivoting (QRCP) [10] from real space grid points to attain a low-rank approximation of Z

$$Z^T\Pi = \mathrm{QR}, \tag{8}$$

where Z^T is the transpose matrix of Z. QRCP decomposes Z^T into a product of an $N_{cv} \times N_r$ orthogonal matrix Q and an upper triangular matrix R, and Π is a permutation matrix calculated to ensure the value of the diagonal elements of R form a nonincreasing sequence to facilitate the determination of interpolation points. As we finish the QRCP calculation, the value of the diagonal elements of matrix R indicates how significant the corresponding column of the Z^T matrix is, we choose the largest ones as interpolation points. In order to reduce the cost during QRCP procedure, we set a minimum numerical threshold. When the $(N_\mu + 1)$ th diagonal element of matrix R becomes less than this threshold, the factorization is concluded, and the corresponding grid points are picked as the interpolation points. The leading N_μ columns of the permuted Z^T

are considered to be linearly independent. The precision for QRCP to find the suitable interpolation points is promising, however, the matrix Z requires $O(N_r \times N_c \times N_v) \approx O\left(N_e^3\right)$ memory and a standard QRCP procedure also cost the computation time of $O\left(N_e^3\right)$, which are not quite desirable.

4.1.2 Computing the Interpolation Vectors of ISDF. When the interpolation points is determined and the corresponding coefficient matrix is constructed at the same time, the next step is to compute the interpolation vectors, which form auxiliary basis functions (ABFS). We rewrite Equation 5 as

$$Z = \Theta C, \tag{9}$$

Equation 9 is an overdetermined linear system problem with respect to the interpolation vectors $\Theta = \left[\zeta_1, \zeta_2, \ldots, \zeta_{N_\mu}\right]$. In general, we impose the Galerkin condition to solve this overdetermined problem.

$$\Theta = ZC^T \left(CC^T\right)^{-1}. \tag{10}$$

To this end, the solution to Equation 10 is a least-squares approximation problem of Equation 9.

In general, ISDF projects the orbital pairs matrix into a much smaller space, which uses N_μ interpolation points to locally express the whole grid points N_r. In our tests, the traditional QRCP procedure for interpolation points chosen, provided by Linear Algebra PACKage (LAPACK) [2], occupies about 90% of the overall ISDF time. So our focus is placed on finding a cheaper method to accurately find the interpolation points.

4.2 Combining ISDF with K-Means Clustering

To further reduce the expensive QRCP procedure in interpolation points selection, we propose a parallel k-Means clustering based interpolation point sampling algorithm in LR-TDDFT. K-Means clustering is one of the most simple yet effective unsupervised machine learning algorithms, which can reveal the underlying correlation of data (electronic correlation effect in this work) by partitioning the real-space grid points into K non-overlapping clusters according to their range of similarity.

In this work, we use the weighted K-means algorithm to determine N_μ non-overlapping clusters from N_r real-space grid points to further choose corresponding interpolation points.

$$\underset{\mathbf{C}_k, \mathbf{c}_k}{\arg\min} \sum_{k=1}^{N_\mu} \sum_{\mathbf{r}_i \in \mathbf{C}_k} w\left(\mathbf{r}_i\right) |\mathbf{r}_i - \mathbf{c}_k|^2, \tag{11}$$

here, $\mathbf{C}_k$ is the cluster given by

$$\mathbf{C}_k = \left\{ \mathbf{r}_i \;\middle|\; |\mathbf{r}_i - \mathbf{c}_k|^2 \le |\mathbf{r}_i - \mathbf{c}_m|^2 \quad \text{for all } i \right\}. \tag{12}$$

The distance between two points in K-means algorithm is defined as squared Euclidean distances (indicated by $|\mathbf{x} - \mathbf{y}|^2$). Thus, to determine which cluster a grid point belongs to, we need to calculate the mean Euclidean distances dist $(\mathbf{r}_i, \mathbf{c}_k)$ between this grid point $\mathbf{r}_i$ and all centroids $\mathbf{c}_k$. The centroid of a cluster $\mathbf{c}_k$ is defined as the weighted average of it

$$\mathbf{c}_k = \frac{\sum_{\mathbf{r}_j \in C_k} \mathbf{r}_j w\left(\mathbf{r}_j\right)}{\sum_{\mathbf{r}_j \in C_k} w\left(\mathbf{r}_j\right)}, \tag{13}$$

and $w\left(\mathbf{r}_i\right)$ is the weight function for each real-space grid point. In LR-TDDFT calculations with plane-wave basis set, we define weight function as Equation 14 of each row of Z, so it can faithfully represent the norm of orbital pairs

$$w(\mathbf{r}) = \sum_{i,j=1}^{N_\mathrm{r}} |\phi_i(\mathbf{r})|^2 |\phi_j(\mathbf{r})|^2. \tag{14}$$

However, using the original K-Means algorithm with random initialization without concerning any underneath feature of the grid points may yield a terrible convergence problem. Since the grid points of orbital pairs contain specific features, the initialization of centroids should be based on the weight function. At the same time, the weight function vector w(r) is in fact of low rank with plane-wave basis set, which means that we only need to care about the grid points whose weights are non-zero or greater than a threshold during the K-Means procedure. For this reason, we first calculate the weight function at all grid points, and remove the points with weights less than the threshold, then initialize N_μ centroids and apply K-Means algorithm only for the remaining grid points. Specifically, we choose N_μ grid points as the initial centroids whose weight functions are rather large.

Since the K-Means algorithm can be directly parallelized, its parallel performance is quite satisfying. The classification step is the most time-consuming step and can be locally computed for each group of grid points. After this step, the weighted sum and total weight of all clusters can be reduced from centroids and broadcasted to all processors for the next iteration.

To validate the capability of our K-Means approach, according to Table 3, we perform a test of our QRCP and K-Means approach on Si_{64} systems with one processor of Intel Xeon E5-2695 CPU. The results show that we can get the same interpolation points with a much cheaper time cost. It should be noticed that the improved K-Means approach in LR-TDDFT calculation scales as $O(N_\mu N_r'^2)$ and N_r' is much smaller than N_r. And the total computational cost for constructing Hamiltonian is $O(N_\mathrm{r} N_\mu^2 + N_\mu N_\mathrm{v}^2 N_\mathrm{c}^2 + N_\mu N_r'^2)$, which is about 2 orders of magnitude smaller compared to our naïve approach. Also, the memory cost in LR-TDDFT calculations is reduced from $O(N_r N_v N_c + N_v^2 N_c^2)$ to $O(N_\mu N_v N_c + N_v^2 N_c^2)$, although real space points N_r is generally larger than $N_v N_c$, the memory cost $O(N_v^2 N_c^2)$ still consumes an expensive memory footprint, which will be further optimized in Section 4.3.

Table 3: Time (in seconds) spent in selecting interpolation procedure of LR-TDDFT calculations.

N_μ	Selecting interpolation points in ISDF	
	QRCP	K-Means
512	10.12	1.61
1,024	42.16	2.85
2,048	147.27	5.57

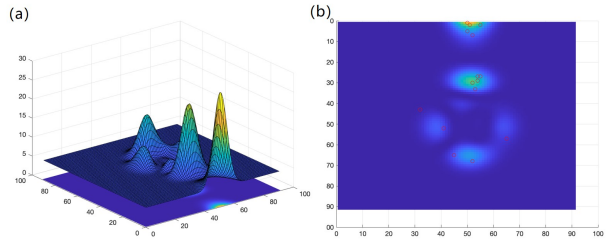

Figure 2: (a) An example of excitation wavefunctions. (b) Projection of excitation wavefunctions and 15 interpolation points chosen by k-Means clustering (indicated by red dots).

4.3 Iterative Eigensolver for Implicitly Constructing Hamiltonian

Constructing and diagonalizing the Hamiltonian occupy almost half of the total wall clock time. According to Amdahl's law [1], to reach a desired overall speedup performance, we also need to accelerate the Hamiltonian diagonalizing step.

To diagonalize H means solving $HX = \Lambda X$ equation, where X represents the coefficient of excitation wavefunctions (eigenvectors) and Λ presents the excitation energies (eigenvalues). In general, the matrix H is large and we always only need a few eigenvalues and eigenvectors. It means that instead of solving the entire diagonalization problem then extracting certain eigenvalues, we only need to find a specific eigen-subspace of H with the smallest eigenvalue. To meet this requirement, we use a parallel locally optimal block preconditioned conjugate gradient (LOBPCG) method, which is a conjugate gradient method, to solve the equation in subspace.

In LOBPCG method, we use that updating formula:

$$X^{(i+1)} = X^{(i)} * C_1^{(i+1)} + W^{(i)} C_2^{(i+1)} + P^{(i)} C_3^{(i+1)}, \tag{15}$$

where W is the preconditioned gradient constructed from:

$$W^{(i)} = K_i^{-1}(HX^{(i)} - X^{(i)}\Theta^{(i)}). \tag{16}$$

K^{-1} is a precondition to accelerate LOBPCG method:

$$K_i = \epsilon_{i_c} - \epsilon_{i_\mu} - \Theta_I^{(i)}. \tag{17}$$

P is an aggregate direction from the previous step:

$$P^{(i)} = W^{(i-1)} C_2^{(i)} + P^{(i-1)} C_3^{(i)}, \tag{18}$$

and when $i = 1$, we choose $P^{(i)} = 0$. If we mark $S_i = [X^{(i)}, W^{(i)}, P^{(i)}]$, then the key step in the LOBPCG method is to project H onto the subspace S_i ($H \in \mathbb{C}^{m\times m}$, $S_i \in \mathbb{C}^{m\times 3k}$, $H_s = S_i^\dagger H S_i \in \mathbb{C}^{3k\times 3k}$ and $C^{(i)} = [C_1^{(i)}, C_2^{(i)}, C_3^{(i)}]^T$) and solve the projected eigenvalue problem $H_s C^{(i+1)} = S_i^\dagger \Lambda_i S_i C^{(i+1)}$. When the subspace S and coefficients C reach a convergence, the corresponding excitation wavefunctions $X = SC$ can be directly computed.

For each iteration in the LOBPCG method, the total computation cost is $3kN_c^2N_v^2 + (3k)^2N_{cv} + (3k)^3 \sim kO(N_e^4)$.

Although LOBPCG is a standard procedure in iteratively diagonalization, but the explicit Hamiltonian cost a $O(N_e^4)$ memory footprint. We notice that $H_s = S_i^\dagger H S_i$ can be expanded as $H_s = S_i^\dagger D S_i + 2S_i^\dagger\{P_{vc}^\dagger[(v_{\mathrm{H}} + f_{\mathrm{xc}})(P_{vc}S_i)]\}$, which means combining with ISDF decomposition, the H can always keep a factored form. After changing the order of calculations, the total computational cost of implicitly constructing and diagonalizing the Hamiltonian H_s is $3kN_\mu N_c N_v + 3kN_\mu N_\mu + (3k)^2N_\mu + (3k)^3 \sim kO(N_e^3)$. The pseudocode of the implicit LOBPCG method is demonstrated in Algorithm 2.

Algorithm 2 Implicit LOBPCG method for solving the LR-TDDFT eigenvalue problem $Hx_i = \lambda_i x_i, i = 1, 2, \ldots, k$.

Input: Hamiltonian H and initial wavefunctions $\{x_i\}_{i=1}^k$.
Initialize the trial subspace $S_1 = [X^{(1)}, W^{(1)}]$ and orthonormalize S_1.
while convergence not reached **do**
Project H onto the subspace S_i: $H_s = S_i^\dagger D S_i + 2S_i^\dagger\{P_{vc}^\dagger[(v_{\mathrm{H}} + f_{\mathrm{xc}})(P_{vc}S_i)]\}$;
Solve the projected eigenvalue problem $H_s C^{(i)} = C^{(i)}\Theta_i$ and obtain the coefficients $C = [C_1^{(i)}, C_2^{(i)}, C_3^{(i)}]^T$ and eigenvalues Θ_i;
Compute $X^{(i)} \leftarrow S_i C^{(i)}$, preconditioned gradient vectors $W^{(i)} = K^{-1}(HX_i - X_i\Theta_i)$ and aggregate direction $P^{(i)} = W^{(i-1)}C_2^{(i)} + P^{(i-1)}C_3^{(i)}$;
Construct the subspace $S_{i+1} \leftarrow [X^{(i)}, W^{(i)}, P^{(i)}]$;
end while
Update $\{x_i\}_{i=1}^k \leftarrow X^{(i)}$.
Output:
Eigenvalues $\{\lambda_i\}_{i=1}^k$ and wavefunctions $\{x_i\}_{i=1}^k$.

The complexity after each step is summarized in Table 4. As we can see, Implicit-Kmeans-ISDF-LOBPCG version ((5) in Table 4) significantly reduces the computation and memory cost by nearly 2 orders of magnitude.

5 PARALLEL IMPLEMENTATION

5.1 Basic Design

To fully take advantage of computing resources provided by the modern HPC systems, we carefully design a two-level MPI-OpenMP hybrid parallelization strategy along with different forms of data distribution fashions in LR-TDDFT implementation depending on their physical nature.

Our method is written within PWDFT (Plane Wave Density Functional Theory) [19], which forms one separate component of the massively parallel quantum chemistry calculations software package DGDFT (Discontinuous Galerkin Density Functional Theory) [18]. For simplicity in implementation and computational scalability, we apply the local-density approximation (LDA) [15] functional in the KS-DFT and LR-TDDFT calculations.

5.2 Parallel Data distribution formula

As we can see from Figure 3, we design three data distribution schemes for the naïve LR-TDDFT implementation. The first one is column block partition, which means each column of wavefunctions Ψ is distributed to each MPI process according to its column index. This data distribution scheme is approvingly efficient to apply Hartree operator since each MPI task is able to perform fast

Table 4: Computational and memory complexity for five different versions for constructing and diagonalizing Hamiltonian in the excited-state electronic structure calculations, including the naïve case and four-level optimized cases. Notice that $N_r \approx 1,000 \times N_e$, $N_\mu \approx 10 \times N_e$, $N_v \approx N_c \approx N_e$ and $1 \leq k \ll N_e$ in the plane-wave basis sets.

LR-TDDFT versions		Constructing Hamiltonian		Diagonalizing Hamiltonian	
		Computation	Memory	Computation	Memory
(1)	Naïve	$O(N_v^2 N_c^2 N_r + N_v N_c N_r)$	$O(N_v^2 N_c^2 + N_r N_v N_c)$	$O(N_r^2 N_v^2 N_c^2)$	$O(N_v^2 N_c^2)$
(2)	QRCP-ISDF	$O(N_r N_\mu^2 + N_\mu N_v^2 N_c^2 + N_\mu N_r^2)$	$O(N_v^2 N_c^2 + N_\mu N_v N_c)$	$O(N_r^2 N_v^2 N_c^2)$	$O(N_v^2 N_c^2)$
(3)	Kmeans-ISDF	$O(N_r N_\mu^2 + N_\mu N_v^2 N_c^2 + N_\mu N_r'^2)$	$O(N_v^2 N_c^2 + N_\mu N_v N_c)$	$O(N_r^2 N_v^2 N_c^2)$	$O(N_v^2 N_c^2)$
(4)	Kmeans-ISDF-LOBPCG	$O(N_r N_\mu^2 + N_\mu N_v^2 N_c^2 + N_\mu N_r'^2)$	$O(N_v^2 N_c^2 + N_\mu N_v N_c)$	$kO(N_v^2 N_c^2)$	$O(N_v^2 N_c^2)$
(5)	Implicit-Kmeans-ISDF-LOBPCG	$O(N_r N_\mu^2 + N_\mu N_v N_c + N_\mu N_r'^2)$	$O(N_v^2 N_c^2 + N_\mu N_v N_c)$	$kO(N_\mu N_v N_c)$	$O(N_\mu^2)$

Fourier transform (FFT) independently in reciprocal space. The second one is row block partition, which means each row of wavefunctions Ψ is distributed to each MPI process based on its column index. This distribution strategy benefits the calculation of the face-splitting product and matrix-matrix multiplication (GEMM). We remark that we use MKL [29] and FFTW [13] to carry out GEMM and FFT operations respectively in our implementation.

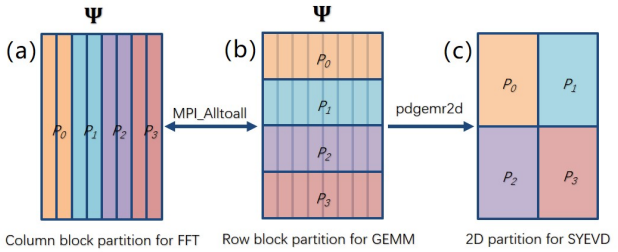

Figure 3: Parallel data and task distribution schemes of LR-TDDFT. (a) column block partition for FFT, (b) row block partition for GEMM and face face-splitting product (c) 2-D parallelization for diagonalization (SYEVD). The parallel scheme is given with 8 wavefunctions and 4 computing processors as examples.

We obtain the ground-state wavefunctions from PWDFT, which is stored with column block partition theme. Then we transform the wavefunctions to row block partition theme to apply the face-splitting product. For the Hamiltonian matrix, we first apply the Hartree operator, which is diagonal in reciprocal space, and then apply the exchange-correlation operator, which is diagonal in real space. To facilitate the calculation steps according to their peculiarities, we use fast Fourier transform (FFT) to convert the P_{vc} from real space to the $\tilde{P}_{vc}$ in reciprocal space ($\{\mathbf{G_i}\}_{i=1}^{N_g}$, N_g indicates grid points in reciprocal space). To apply Fourier transform, the conversion from row block partition to column block partition is carried by MPI_Alltoall routine as demonstrated in Figure 3 (a) and (b). When we finish constructing the Hamiltonian, we need to diagonalize it to obtain excitation energies and corresponding wavefunctions. For the diagonalization step, the two-dimensional block cyclic partition theme as sketched in Figure 3(c) is the most advantageous data distribution type to perform directly diagonalization via the SYEVD routine in the ScaLAPACK library. We perform the data redistribution routine pdgemr2d provided by the ScaLAPACK library to convert the data distribution theme from row block partition to two-dimensional block cyclic partition

5.3 Overlap of Computation and Communication

As shown in line 8 and 9 of Algorithm 1, after we perform General Matrix Multiply (GEMM) to get the Hartree-exchange-correlation integrals in each single process, MPI_Allreduce is used to gather V_{Hxc} in all MPI tasks. There is data dependence because MPI_Allreduce must wait for GEMM to finish computation, in particular when the size of a matrix is large, thus disrupt the overlapping of computation and communication. When the studied system's size increases, although GEMM can be calculated via MKL in a very efficient way, both GEMM and MPI_Allreduce will introduce much time cost. To fully accelerate the process of LR-TDDFT, we make attempts to overlap the step of computation and communication.

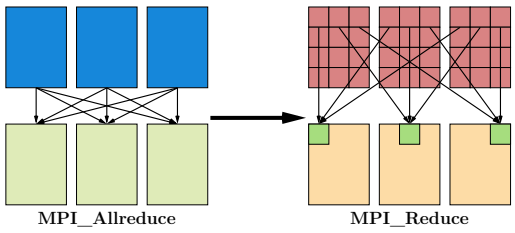

Figure 4: Data reduction optimization, take the first row of Matrix $V_{\mathbf{Hxc}}$ as an example.

First, by analyzing the data partition of LR-TDDFT, we find that in order to calculate the difference between the energy eigenvalues of Kohn-Sham function, not all MPI tasks need to store the entire V_{Hxc} matrix. Therefore, we optimize the data partitioning method shown in Figure 4. Then we get the result of GEMM, each MPI task only needs to store a part of the V_{Hxc} matrix.

The above attempt brings two benefits. First of all, this new data partitioning method can reduce the memory usage of MPI process. Second, we don't need to execute MPI_AllReduce to collect the entire V_{Hxc} matrix, but use MPI_Reduce to transmit a part of the V_{Hxc} matrix to each MPI task according to the index.

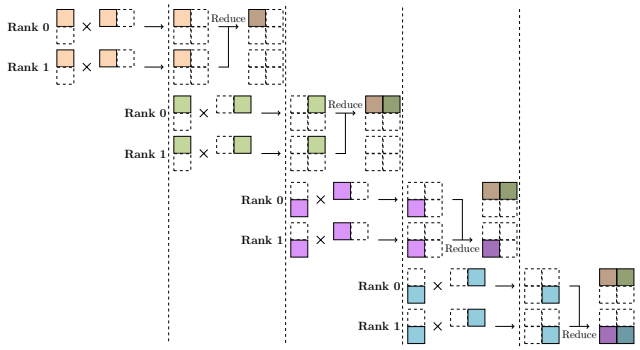

Figure 5: Pipeline approach of GEMM and MPI_Reduce.

As a result of this attempt, we have eliminated part of the data dependence. In more detail, we can divide the matrix into small pieces and manually perform GEMM on these small parts. The basic flow of GEMM and reduction is shown in Figure 5. Once the result of each block is obtained, we can immediately reduce the block matrix to each MPI task through MPI_Reduce. Then, we will allocate the entire V_{Hxc} distributed in each MPI task.

6 NUMERICAL RESULTS AND ANALYSIS

6.1 Setup of the Test Physical Systems and Testing Environment

The testing systems include two parts: (1) cubic silicon systems and (2) Magic angle twisted bilayer graphene with 1180 atoms. For cubic silicon systems, we use various choices of crystal silicon systems with 64, 216, 512, 1,000, 1,728, 2,744, and 4,096 silicon atoms labeled by Si_{64}, Si_{216}, Si_{512}, Si_{1000}, Si_{1728}, Si_{2744}, and Si_{4096}, respectively. We apply the Hartwigsen Goedecker Hutter (HGH) norm-conserving pseudopotential [16] in all of the following tests. The total number of real-space grid points N_r is determined by the kinetic energy cutoff (E_{cut}) defined as $(N_r)_i = \sqrt{2E_{\mathrm{cut}}}L_i/\pi$, where L_i is the length of each supercell along each (x, y and z) coordinate direction. Without additional illustrations, the kinetic energy cutoff in our experiments is 20 Hartree. For example, the number of real-space grid points for a wavefunction matrix in Si_{4096} is $N_r = 166 \times 166 \times 166 = 4{,}574{,}296$.

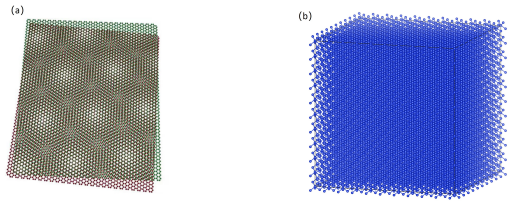

Figure 6: Atomic configurations of (a) MATBG and (b) bulk silicon with 4,096 atoms.

All numerical tests are performed on the National Energy Research Scientific Computing Center (NERSC)'s Cori supercomputer. We run our code on the Haswell partition, whose each node has two sockets, each socket is populated with a 2.3 GHz 16-core Haswell processor (Intel Xeon Processor E5-2698 v3) and 128 GB DDR4 2133 MHz memory. Each core supports 2 hyper-threads and has two 256-bit-wide vector units. Each core has a theoretical peak performance of 36.8 Gflops double-precision operations. In our numerical experiments, if not mentioned particularly, we apply 8 MPI tasks per computing node (4 OpenMP threads per MPI process).

6.2 Numerical Accuracy

We first analyze the numerical accuracy of our code. We compare our naïve version (LR-TDDFT) code and Implicit-Kmeans-ISDF-LOBPCG version (ISDF-LOBPCG) with the current state of the art software Quantum Espresso (QE) [14], which serves as an accuracy benchmark. Due to the bad scalability of QE, we use H_2O and Si_{64} as our benchmark parts. We choose the system of one H_2O molecule with the simulation boxes $11.000 \times 11.000 \times 11.000$ Å^3 and the system of 64 silicon atoms (Si_{64}) with the simulation boxes $20.525 \times 20.525 \times 20.525$ Å^3. The calculations are performed using the Casida calculations and the relative excited energy errors are defined by:

$$\begin{aligned} \Delta E_1 &= (E_{\mathrm{QE}} - E_{\text{LR-TDDFT}})/E_{\mathrm{QE}} \\ \Delta E_2 &= (E_{\mathrm{QE}} - E_{\text{ISDF-LOBPCG}})/E_{\mathrm{QE}} \end{aligned} \tag{19}$$

Table 5 lists the results. We find a good agreement between the results of QE and LR-TDDFT, with a small difference in excitation energies for an identical ordering of states. Our optimizations will only introduce little error, as small as 0.001% in relative error, which is fairly negligible. Results mean that we already dismiss almost all redundant computational costs beneath LR-TDDFT calculations. The accuracy reaches the level we need and results that the accuracy will further improve as kinetic energy cutoff increases.

Table 5: The three lowest excitation energies and corresponding relative errors of H_2O system and Si_{64} system.

QE	LR-TDDFT	ISDF-LOBPCG	ΔE_1	ΔE_2
Single water molecule H_2O (E_{cut} = 100.0 Ha, N_v = 20 and N_c = 4)				
0.398312	0.397830	0.397829	0.121%	0.121%
0.550416	0.546664	0.546664	0.682%	0.682%
0.729568	0.732786	0.732785	-0.441%	-0.441%
Periodic bulk silicon Si_{64} (E_{cut} = 50.0 Ha, N_v = 128 and N_c = 50)				
0.044350	0.043942	0.0439429	0.920%	0.918%
0.044350	0.043942	0.0439429	0.920%	0.918%
0.044350	0.043942	0.0439429	0.920%	0.918%

6.3 Strong Scaling

The most significant measure for the LR-TDDFT software with all the aforementioned optimization is the strong scalability, which reflects how much the growth of hardware resources takes effect for a determined system as adding the numbers of computing resources. As shown in Figure 7, we present the strong scaling performance

of 3 versions of our code: Naïve version, ISDF version and ISDF-LOBPCG version (corresponding to (1), (3) and (5) in Table 4). The testing system contains 1000 silicon atoms and the real space points $N_r = 104 \times 104 \times 104 = 1{,}124{,}864$. We evaluate strong scalability by parallel efficiency defined in Equation 20, and the speedup is compared with wall clock time in 128 CPU cores.

$$\text{Parallel Efficiency} = \frac{\text{Speedup}}{\text{Multiple of CPU cores}} \tag{20}$$

The parallel efficiency of our naïve design maintains above 50% when scaling to 2,048 processing cores. This result is quite acceptable among LR-TDDFT calculations with plane-wave basis set since we need to do a series of collective communication in the global domain.

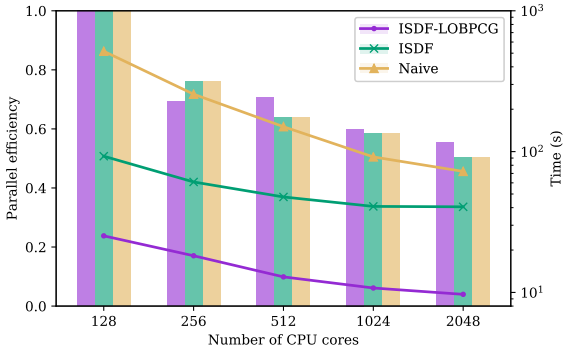

Figure 7: Strong scaling: the wall clock time and parallel efficiency with respect to the number of CPU cores. Lines denote time (in seconds) and bar denotes parallel efficiency.

Also, we give a more detailed analysis by splitting the wall clock time during the procedure of constructing the Hamiltonian into 4 parts: (1) K-Means, (2) FFT, (3) MPI, and (4) GEMM and Allreduce. As Figure 8 shows, due to our parallel design, the K-Means, GEMM, FFT, and even MPI procedure maintain a very convincing strong scaling performance till 2048 CPU cores. But to implement the implicit method, we transform the Hartree-exchange-correlation integrals from a single GEMM operation to a serial of GEMMs and an MPI_Allreduce, which hinders the total time speedup from ideal. Since MPI collective communication routines will bring in extra overhead. To maintain good speedup and scale the system to a larger size, the implicit Hamiltonian method is indispensable, so this method is a trade-off between efficiency and strong scaling. But in our test, GEMM and Allreduce step will only cost 12.87% of the total time of constructing Hamiltonian, the small sacrifice of strong scaling is quite worthy.

In particular, we test Si_{4096} system with 8,192 and 12,288 processing cores and bind 16 OpenMP threads with each MPI process. The corresponding wall-clock time is 14.02 and 10.70 seconds, with a strong scaling performance of 87.34%. Since increasing the number of OpenMP threads can reduce the processes within the calculation, it can straightforwardly reduce the communicational cost, hence improving strong scalability when we apply a large number of CPU cores. This unprecedented speed enables the electronic structure exploration of large-scale systems containing more than 4,000 atoms by performing LR-TDDFT with a very low computation cost.

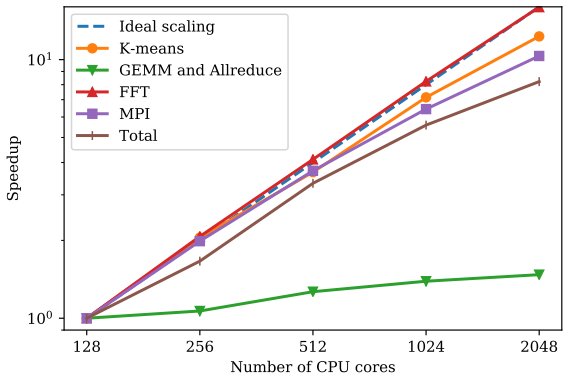

Figure 8: Strong scaling performance of constructing Hamiltonian step in the LR-TDDFT calculations.

6.4 Weak Scaling

One yet critical metric for the LR-TDDFT software is the weak scalability, which reflects the parallel performance for a scaled problem size accompanying a fixed number of CPU cores. In particular, our method can significantly reduce the memory cost during calculation steps of LR-TDDFT simulation, so we can use far fewer computing resources to study a much larger physical system. We use LR-TDDFT-optimized code to test Si_{512}, Si_{1000}, Si_{1728}, Si_{2744} and Si_{4096} systems with 1024 cores and we bind single core to a process, corresponding time is 3.58, 10.23, 26.95, 35.58 and 41.89 seconds. This result suits our computational complexity well.

6.5 Speedup

We also perform tests to validate the speedup ability of our method. We further reduce the computation resources, and bind a single core to an MPI process, thus each process holds only 4 GB of memory. We evaluate tests on different sizes of systems with Naïve and ISDF-LOBPCG version ((1) and (5) in Table 4) code. We observe an average speedup of 9.254x, which is quite convincing. And as sketched in Figure 7, when we apply larger computation resources, we also observe an average of 12.58x speedup. In fact, among all of our numerical results, the average speedup under our optimizations is over 10x, combining the accuracy property, our method can reach quite faster calculations with fewer resources.

Table 6: The wall-clock time (in seconds) and speedups of different sizes of 3D bulk silicon sytems.

Systems	Naïve	ISDF-LOBPCG	Speedup
Si_{64}	3.19	0.24	13.06
Si_{216}	6.95	0.70	9.89
Si_{512}	14.74	1.89	7.79
Si_{1000}	32.15	5.13	6.26

6.6 Application: Ground and Excited States calculations of MATBG

Magic-angle twisted bilayer graphene (MATBG) [5] with Moiré superlattices can trap long-lived interlayer excitons, which provides a good platform to investigate the exciton dynamics effect and many-body effect in condensed matter physics. However, there has been little research on the calculation of moiré excitons in MATBG since such metallic and long-range electron-correlation periodic solid systems are too large to investigate by LR-TDDFT calculations.

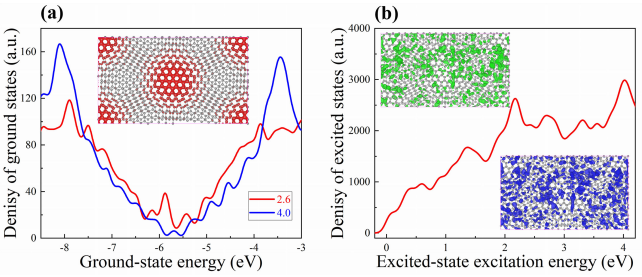

Figure 9: Ground-state and excited-state electronic structures of MATBG. (a) Density of states of ground-state MATBG with different interlayer distances D = 2.6 and 4.0 Å (Inset displays the isosurface of ground-state wavefunctions for D = 2.6 Å), and (b) density of states of excitation energies of MATBG with D= 2.6 Å (Inset displays the isosurfaces of the lowest excited-state electron (blue) and hole (green)).

We use large-scale DFT and LR-TDDFT calculations implemented in PWDFT to investigate the ground-state and excited-state electronic structures of MATBG that contains 1,180 carbon atoms as shown in Figure 9. We calculate the density of states (DOS) of MATBG as shown in Figure 9 (a). In particular, we observe the moiré superlattices trap a number of localized electrons at the Fermi level in MATBG when the interlayer distance is 2.6 Å, due to strong quantum electron-correlation effect, which agrees well with the previous tight-binding models and experimental measurements [5]. However, when the interlayer distance is increased to be 4.0 Å, such localized states disappear as the electron-correlation effect in MATBG. Also we compute the density of states of excitation energies of MATBG (D = 2.6 Å) as shown in Figure 9 (b). A number of excited-states (photoexcited electrons and holes) are produced at the low-lying energy range (0 - 0.5 eV), which may result from the photoexcitation between the strongly localized states of ground-state MATBG.

To conclude, our computational results provide some useful insight into the intrinsic physical mechanism of quantum ground-state and excited-state electronic structures of MATBG, which can help to understand the quantum electron-correlation effect in MATBG and corresponding physical phenomena, such as photoexcitation, superconductivity and topological insulator, in future experiments.

7 CONCLUSION

In this work, we propose several optimizations to accelerate linear-response time-dependent density functional theory (LR-TDDFT) calculations without little accuracy loss. First we combine K-Means clustering with interpolative separable density fitting (ISDF) decomposition to choose the interpolation points at a much cheaper cost, compared to traditional QRCP based ISDF procedure. Then we adopt Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) method to get the lowest k eigenvalue iteratively. At last, we put forward an implicitly LR-TDDFT Hamiltonian constructing and diagonalizing method to further reduce the memory consumption and computation time. We also carefully design the task distribution schemes to ensure the flexibility and efficient use of the computation process. In our experiments, these methods can gain an acceleration of above 10 times while preserving admirable accuracy. By testing Si_{4096} system with 12,288 cores, we also prove that under our parallel design, good strong scalability can be reachable even using a large number of CPU cores. Given all these benefits, we can push the envelope of first-principles excited-state simulations further.

ACKNOWLEDGMENTS

This work is partly supported by the National Natural Science Foundation of China (22173093, 21688102, 22003061, 62102389), the Hefei National Laboratory for Physical Sciences at the Microscale (KF2020003, SK2340002001), the Chinese Academy of Sciences Pioneer Hundred Talents Program (KJ2340007002), the National Key Research and Development Program of China (2016YFA0200604), the Anhui Initiative in Quantum Information Technologies (AHY090400), the Center of Chinese Academy Project for Young Scientists in Basic Research (YSBR-005), the Fundamental Research Funds for the Central Universities (WK2340000091, WK2060000018) from University of Science and Technology of China.

REFERENCES

[1] Gene M Amdahl. 1967. Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*. 483–485.
[2] E Angerson, Z Bai, J Dongarra, A Greenbaum, A McKenney, J Du Croz, S Hammarling, J Demmel, C Bischof, and D Sorensen. 1990. LAPACK: A portable linear algebra library for high-performance computers. In *Supercomputing'90: Proceedings of the 1990 ACM/IEEE Conference on Supercomputing*. IEEE, 2–11.
[3] Francesco Aquilante, Thomas Bondo Pedersen, and Roland Lindh. 2007. Low-cost evaluation of the exchange Fock matrix from Cholesky and density fitting representations of the electron repulsion integrals. *Journal of Chemical Physics* 126, 19 (2007), 194106.
[4] Thomas L Beck. 2000. Real-space mesh techniques in density-functional theory. *Reviews of Modern Physics* 72, 4 (2000), 1041.
[5] Yuan Cao, Valla Fatemi, Shiang Fang, Kenji Watanabe, Takashi Taniguchi, Efthimios Kaxiras, and Pablo Jarillo-Herrero. 2018. Unconventional superconductivity in magic-angle graphene superlattices. *Nature* 556, 7699 (2018), 43–50.
[6] Mark E Casida. 1995. Time-dependent density functional response theory for molecules. In *Recent Advances In Density Functional Methods: (Part I)*. World Scientific, 155–192.
[7] Jaeyoung Choi, Jack J Dongarra, Roldan Pozo, and David W Walker. 1992. ScaLAPACK: A scalable linear algebra library for distributed memory concurrent computers. In *The Fourth Symposium on the Frontiers of Massively Parallel Computation*. IEEE Computer Society, 120–121.
[8] Ernest R Davidson. 1975. The Iterative Calculation of a Few of the Lowest Eigenvalues and Corresponding Eigenvectors of Large Real-Symmetric Matrices. *J. Comput. Phys.* 17, 1 (1975), 87–94.
[9] Mauro Del Ben, Charlene Yang, Zhenglu Li, H Felipe, Steven Louie, and Jack Deslippe. 2020. Accelerating large-scale excited-state GW calculations on leadership HPC systems. In *2020 SC20: International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*. IEEE Computer Society, 36–46.
[10] Jed A Duersch and Ming Gu. 2017. Randomized QR with column pivoting. *SIAM Journal on Scientific Computing* 39, 4 (2017), C263–C291.

[11] Jed A Duersch, Meiyue Shao, Chao Yang, and Ming Gu. 2018. A robust and efficient implementation of LOBPCG. *SIAM Journal on Scientific Computing* 40, 5 (2018), C655–C676.

[12] Alexander L Fetter and John Dirk Walecka. 1971. Quantum theory of many-particle systems. *qtmp* (1971).

[13] Matteo Frigo and Steven G Johnson. 1998. FFTW: An adaptive software architecture for the FFT. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, Vol. 3. IEEE, 1381–1384.

[14] Paolo Giannozzi, Stefano Baroni, Nicola Bonini, Matteo Calandra, Roberto Car, Carlo Cavazzoni, Davide Ceresoli, Guido L Chiarotti, Matteo Cococcioni, Ismaila Dabo, et al. 2009. QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials. *Journal of physics: Condensed matter* 21, 39 (2009), 395502.

[15] Stefan Goedecker, Michael Teter, and Jürg Hutter. 1996. Separable dual-space Gaussian pseudopotentials. *Physical Review B* 54, 3 (1996), 1703.

[16] Christian Hartwigsen, Sephen Gœdecker, and Jürg Hutter. 1998. Relativistic separable dual-space Gaussian pseudopotentials from H to Rn. *Physical Review B* 58, 7 (1998), 3641.

[17] Pierre Hohenberg and Walter Kohn. 1964. Inhomogeneous electron gas. *Phys. Rev.* 136, 3B (1964), B864.

[18] Wei Hu, Lin Lin, and Chao Yang. 2015. DGDFT: A massively parallel method for large scale density functional theory calculations. *Journal of Chemical Physics* 143, 12 (2015), 124110.

[19] Wei Hu, Lin Lin, and Chao Yang. 2017. Interpolative separable density fitting decomposition for accelerating hybrid density functional calculations with applications to defects in silicon. *Journal of Chemical Theory and Computation* 13, 11 (2017), 5420–5431.

[20] Weile Jia, Lin-Wang Wang, and Lin Lin. 2019. Parallel transport time-dependent density functional theory calculations with hybrid functional on summit. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–23.

[21] CG Khatri and C Radhakrishna Rao. 1968. Solutions to some functional equations and their applications to characterization of probability distributions. *Sankhyā: The Indian Journal of Statistics, Series A* (1968), 167–180.

[22] Georg Kresse and Jürgen Furthmüller. 1996. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Physical review B* 54, 16 (1996), 11169.

[23] Jianfeng Lu and Lexing Ying. 2015. Compression of the electron repulsion integral tensor in tensor hypercontraction format with cubic scaling cost. *J. Comput. Phys.* 302 (2015), 329–335.

[24] Schuetz Martin, Usvyat Denis, Lorenz Marco, Cesare Pisani, Lorenzo Maschio, Silvia Maria Casassa, Migen Halo, et al. 2010. Density fitting for correlated calculations in periodic systems. (2010).

[25] Roberto Olivares-Amaya, Weifeng Hu, Naoki Nakatani, Sandeep Sharma, Jun Yang, and Garnet Kin-Lic Chan. 2015. The ab-initio density matrix renormalization group in practice. *Journal of Chemical Physics* 142, 3 (2015), 034102.

[26] Erich Runge and Eberhard KU Gross. 1984. Density-functional theory for time-dependent systems. *Physical Review Letters* 52, 12 (1984), 997.

[27] Jack Strand, Sergey K Chulkov, Matthew B Watkins, and Alexander L Shluger. 2019. First principles calculations of optical properties for oxygen vacancies in binary metal oxides. *Journal of Chemical Physics* 150, 4 (2019), 044702.

[28] Marat Valiev, Eric J Bylaska, Niranjan Govind, Karol Kowalski, Tjerk P Straatsma, Hubertus JJ Van Dam, Dunyou Wang, Jarek Nieplocha, Edoardo Apra, Theresa L Windus, et al. 2010. NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations. *Computer Physics Communications* 181, 9 (2010), 1477–1489.

[29] Endong Wang, Qing Zhang, Bo Shen, Guangyong Zhang, Xiaowei Lu, Qing Wu, and Yajuan Wang. 2014. Intel math kernel library. In *High-Performance Computing on the Intel® Xeon Phi*. Springer, 167–188.

[30] Long Wang, Yue Wu, Weile Jia, Weiguo Gao, Xuebin Chi, and Lin-Wang Wang. 2011. Large scale plane wave pseudopotential density functional theory calculations on GPU clusters. In *SC'11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 1–10.

[31] Kazuhiro Yabana and GF Bertsch. 1996. Time-dependent local-density approximation in real time. *Physical Review B* 54, 7 (1996), 4484.

[32] TJ Zuehlsdorff, Peter D Haynes, Felix Hanke, MC Payne, and Nicholas DM Hine. 2016. Solvent effects on electronic excitations of an organic chromophore. *Journal of Chemical Theory and Computation* 12, 4 (2016), 1853–1861.