Article

High performance computing of DGDFT for tens of thousands of atoms using millions of cores on Sunway TaihuLight

Wei Hu [a], Xinming Qin [a], Qingcai Jiang [b], Junshi Chen [b], Hong An [b],*, Weile Jia [c], Fang Li [d], Xin Liu [d], Dexun Chen [d], Fangfang Liu [e], Yuwen Zhao [e], Jinlong Yang [a],*

[a] Hefei National Laboratory for Physical Sciences at the Microscale, Department of Chemical Physics, and Synergetic Innovation Center of Quantum Information and Quantum Physics, University of Science and Technology of China, Hefei 230026, China
[b] School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China
[c] Department of Mathematics, University of California, Berkeley, CA 94720, USA
[d] National Supercomputing Center, Wuxi 214072, China
[e] Institute of Software Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

ABSTRACT

High performance computing (HPC) is a powerful tool to accelerate the Kohn–Sham density functional theory (KS-DFT) calculations on modern heterogeneous supercomputers. Here, we describe a massively parallel implementation of discontinuous Galerkin density functional theory (DGDFT) method on the Sunway TaihuLight supercomputer. The DGDFT method uses the adaptive local basis (ALB) functions generated on-the-fly during the self-consistent field (SCF) iteration to solve the KS equations with high precision comparable to plane-wave basis set. In particular, the DGDFT method adopts a two-level parallelization strategy that deals with various types of data distribution, task scheduling, and data communication schemes, and combines with the master–slave multi-thread heterogeneous parallelism of SW26010 processor, resulting in large-scale HPC KS-DFT calculations on the Sunway TaihuLight supercomputer. We show that the DGDFT method can scale up to 8,519,680 processing cores (131,072 core groups) on the Sunway TaihuLight supercomputer for studying the electronic structures of two-dimensional (2D) metallic graphene systems that contain tens of thousands of carbon atoms.

© 2020 Science China Press. Published by Elsevier B.V. and Science China Press. All rights reserved.

1. Introduction

The Kohn–Sham density functional theory (KS-DFT) [1,2] is the most powerful methodology to perform first-principles calculations for studying the electronic structures of molecules and solids. However, conventional KS-DFT calculations show cubic computational complexity $\mathcal{O}(N^3)$ with respect to the system size N. The computational cost and memory usage of KS-DFT calculations increase rapidly as the system size and the KS-DFT calculations are only limited to small systems containing hundreds of atoms. Therefore, the KS-DFT calculations become prohibitively expensive for first-principles materials simulations on large-scale systems that contain thousands of atoms.

Several low-scaling methods have been proposed for reducing the computational cost of KS-DFT calculations, such as linear scaling $\mathcal{O}(N)$ methods [3–5], divide-and-conquer (DAC) methods [6] and

fragment molecular orbital (FMO) methods [7]. These low-scaling methods principally rely on the nearsightedness principle in molecules and semiconductors, and have been widely implemented with small localized basis sets in real-space, such as Gaussian [8] and numerical atomic orbitals [4], resulting in the sparse Hamiltonian in real space. Based on these low-scaling methods, several highly efficient KS-DFT materials simulation software packages have been developed, such as SIESTA [9], CP2K [10], CONQUEST [11] and HON-PAS [12], which are beneficial to take full advantage of the massive parallelism available on modern high performance computing (HPC) architectures due to the local data communication of sparse Hamiltonian generated in small localized basis sets.

However, the accuracy of these low-scaling methods strongly depends on the parameters of localized basis sets, and is difficult to be improved systematically, compared to large uniform basis sets, such as plane-waves. Several KS-DFT materials simulation software packages have been developed by using plane wave basis set, such as VASP [13] and QUANTUM ESPRESSO [14]. But such plane wave basis set always requires large number of basis functions for the high accuracy and is difficult to take advantage of the HPC

* Corresponding authors.
  E-mail addresses: han@ustc.edu.cn (H. An), jlyang@ustc.edu.cn (J. Yang).

calculations on modern heterogeneous supercomputers due to the large all-to-all data communications of dense Hamiltonian [15].

The recently developed discontinuous Galerkin density functional theory (DGDFT) [16–20] aims to combine the advantages of both small localized and large uniform basis sets, which can reduce the number of basis functions similar to numerical atomic basis sets, while maintaining the high precision comparable to that of plane-wave basis set. The DGDFT method is discretized on an adaptive local basis (ALB) set [16]. Its unique feature is that each ALB function is strictly localized in a subdomain in real space, which results in the sparse Hamiltonian in unchanged block-tridiagonal structure for both metallic and semiconducting systems. Therefore, the DGDFT method is beneficial to take full advantage of the massive parallelism available on modern heterogeneous supercomputers [17].

It should be noticed that such parallel KS-DFT calculations increasingly require more complicated software development to achieve better parallel performance and scalability across the vastly diverse ecosystem of modern heterogeneous supercomputers, especially, the widely used X86 CPU (Central Processing Unit) architectures. Large-scale KS-DFT calculations have been performed in CP2K [10] and CONQUEST [11] on the Cray supercomputer with X86 architecture. In particular, DGDFT has demonstrated scaling to 128,000 cores on the Edison supercomputer at the USA NERSC platform for performing large-scale KS-DFT calculations on semiconducting phosphorene systems with 14, 000 atoms [17].

In China, the Sunway TaihuLight [21] is a new generation of the fastest supercomputers in the world, which uses the Chinese home-grown SW26010 processors based on a new Sunway master–slave heterogeneous architecture. Different from the widely used X86 CPU architectures, each master processing core can be effectively multi-thread accelerated by 64 slave processing cores on the SW26010 processor, similar to the multi-thread (64 threads) parallelism that bridges the gap between the Open Multi-Processing (OpenMP) (16–32 threads) and Graphics Processing Unit (GPU) (256–512 threads) parallel programming. Such hardware advantage requires the KS-DFT software packages to be reimplemented into the new Sunway TaihuLight supercomputer.

In the present work, we describe a massively parallel implementation of the DGDFT method on the Sunway TaihuLight supercomputer. We demonstrate that the DGDFT method adopts a two-level parallelization strategy that makes use of different types of data distribution, task scheduling, and data communication schemes, and combines with the feature of master–slave multi-thread heterogeneous parallelism of SW26010 processors, resulting in extreme-scale HPC KS-DFT calculations for tens of thousands of atoms using millions of cores on the Sunway TaihuLight supercomputer.

2. Methodology

In this section, we describe the theoretical algorithms and parallel implementation of DGDFT on the Sunway TaihuLight supercomputer in detail. The key spirit of DGDFT is to discretize the global KS equations by using the adaptive local basis (ALB) set in a discontinuous Galerkin (DG) framework [16]. The scalable implementation of DGDFT is based on the two-level parallelization strategy of DGDFT combining with the master–slave multi-thread heterogeneous parallelism of SW26010 processor on the Sunway TaihuLight supercomputer.

2.1. Discontinuous Galerkin density functional theory

The basic idea of DGDFT is the domain decomposition algorithm for generating a new type of basis sets to solve the KS equations

[16]. In the DGDFT method, we partition the global domain Ω into a number of subdomains (called elements), denoted by $\mathcal{T} = \{E_k\}_{k=1}^{M}$ to a collection of all elements (M is the total number of elements). In the current version of DGDFT, we use the periodic boundary condition for both molecules and solids. Therefore, each surface of the element must be shared between two neighboring elements, and $\mathcal{S}$ denotes the collection of all the surfaces.

An example of partitioning the global domain of a graphene system into a number of elements is given in Fig. 1. This is a 2D graphene system G180 containing 180 carbon atoms. According to the grid points in real space, the global domain is partitioned into 16 equal-sized elements in a 2D 4×4 mesh along the x and y directions, respectively. An extended element Q_6 is associated with the central element E_6, and Q_6 includes 9 elements, $E_1, E_2, E_3, E_5, E_6, E_7, E_9, E_{10}$, and E_{11}. There are four surfaces surrounding the central element E_6 with boundary integrals highlighted by green arrows. We solve small local KS equations on this extended element Q_6 only containing a few atoms and obtain a set of eigenfunctions. Then we restrict and truncate these eigenfunctions into the central element E_6 and obtain a new set of ALB functions only localized inside the element E_6, which are adaptive
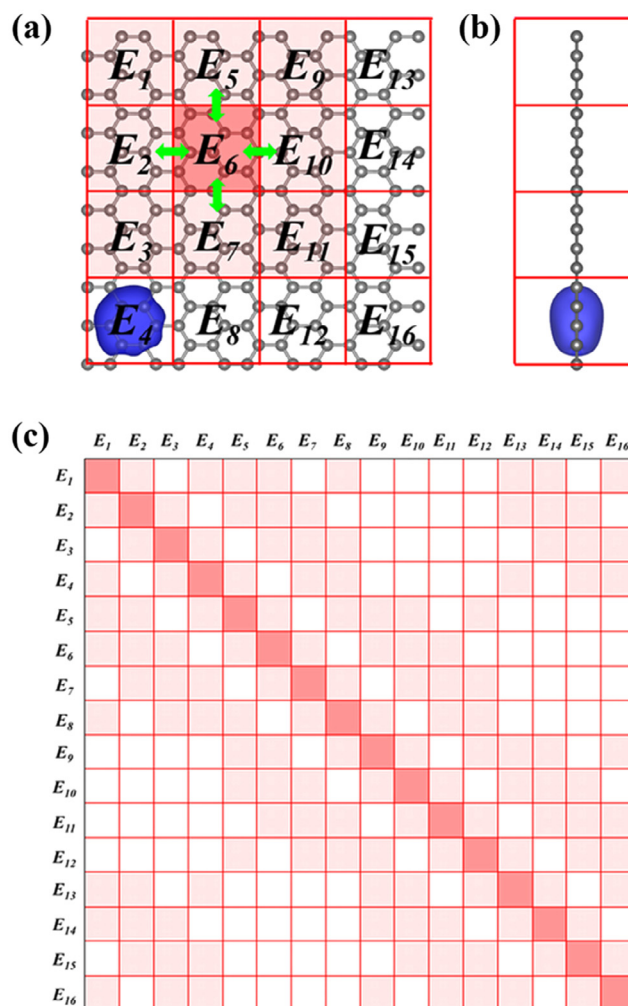

Fig. 1. A 2D graphene system G180 system in 2D partitioned into 16 (4×4) equal-sized elements. (a) An extended element Q_6 associated with the central element E_6, and Q_6 includes 9 elements in a 2D mesh ($E_1, E_2, E_3, E_5, E_6, E_7, E_9, E_{10}$, and E_{11}). There are four surfaces surrounding the central element E_6 with boundary integrals highlighted by green arrows. The first ALB function belonging to the element E_4 is plotted with blue isosurfaces (0.01 Hartree/Bohr^3) in top and (b) side views. (c) The block-tridiagonal sparse structure of DG Hamiltonian matrix H^{DG}. The blocks with nonzero values are highlighted with red areas.

to change according to the atomic and environmental information during the SCF iterations when solving the global KS equations. For example, the first ALB function belonging to the element E_4 is plotted in Fig. 1a and b. This ALB function is strictly localized inside E_4 and is therefore discontinuous across the boundary of elements, resulting in four surfaces surround this element with boundary integrals. Therefore, the ALB functions can be acted as a new type of localized and orthogonal basis set to discretize the global KS equations, resulting in a sparse block-tridiagonal structure of DG Hamiltonian matrix [17] as shown in Fig. 1c.

It should be noticed that the framework of DGDFT to solve the global KS equations is similar to the standard DFT methods discretized on atomic localized basis sets, such as numerical atomic basis orbitals implemented in SIESTA [9]. The key spirit of DGDFT is to use the ALB functions to discretize the Kohn–Sham equations. In particular, the ALB functions are orthogonal, localized, and complete basis set that combines with the advantages of both numerical atomic basis orbitals (localization) and plane waves (orthogonality and completeness). Such features make the DG Hamiltonian keep unchanged in a sparse block-tridiagonal structure during the SCF iterations even for metallic systems. There are several new diagonalization methods implemented in DGDFT, such as CheFIS [19] and PEXSI [22], to take advantage of such block-sparse DG Hamiltonian in the framework of DGDFT. Therefore, there are four time-consuming parts in DGDFT, including generating the ALB functions, constructing and diagonalizing the DG Hamiltonian, as well as computing the electron density, total energy and atomic forces [17]. For the flowchart of DGDFT, except for the first step to generate the ALB functions on-the-fly during the SCF iterations, other three parts of DGDFT are similar to SIESTA. The flowchart of the DGDFT method for solving the global KS equations is given in Fig. 2.

2.1.1. Generating ALB functions

The key spirit of DGDFT is to generate the ALB functions on-the-fly, which are adaptive to change according to the atomic and environmental information during the SCF iterations when solving the global KS equations. In order to generate the ALB functions on each element E_k, we construct an extended element Q_k consisting of a central element E_k and surrounding a set of buffer elements surrounding E_k. An example of partitioning the global domain of a 2D graphene system is shown in the Fig. 1a.

On each extended element Q_k, we solve the local KS equations defined as

$$H^{Q_k}\phi_{k,j}^{Q_k} = \left(-\frac{1}{2}\Delta + V_{\text{eff}}^{Q_k} + V_{\text{nl}}^{Q_k}\right)\phi_{k,j}^{Q_k} = \lambda_{k,j}\phi_{k,j}^{Q_k}, \tag{1}$$

where H^{Q_k} and $\phi_{k,j}^{Q_k}$ are the local Hamiltonian and wavefunctions on the extended element Q_k. These local wavefunctions $\phi_{k,j}^{Q_k}$ satisfy the orthonormality condition. $V_{\text{eff}}^{Q_k} = V_{\text{loc}}^{Q_k} + V_H^{Q_k} + V_{\text{xc}}^{Q_k}$ is the effective potential on the extended element Q_k, including the local pseudopotential $V_{\text{loc}}^{Q_k}$, the Hartree potential $V_H^{Q_k}$ and the exchange–correlation potential $V_{\text{xc}}^{Q_k}$. $V_{\text{nl}}^{Q_k} = \sum_{I,\ell}^{Q_k}\gamma_{I,\ell}b_{I,\ell}b_{I,\ell}$ is the nonlocal pseudopotential ($b_{I,\ell}$ is the ℓ-th projected function of the atom I and $\gamma_{I,\ell}$ is corresponding real scalar).

In the current framework of DGDFT, these local KS equations are discretized on the standard plane-wave basis set with the same accuracy with QUANTUM ESPRESSO [14]. We implement a self-contained module called PWDFT (Plane-wave density functional theory) [23] in DGDFT. When using plane wave basis set, it is viable to use the iterative algorithms to solve the KS-DFT equations $HX = XE$. Because the dimension of Hamiltonian matrix H is above $N_r^{Q_k} = 10^6$ ($N_r^{Q_k}$ is the number of grid points in real space of Q_k), only a percentage of amounts to more than 100 eigenvalues even for small systems containing tens of atoms. There are several iter-
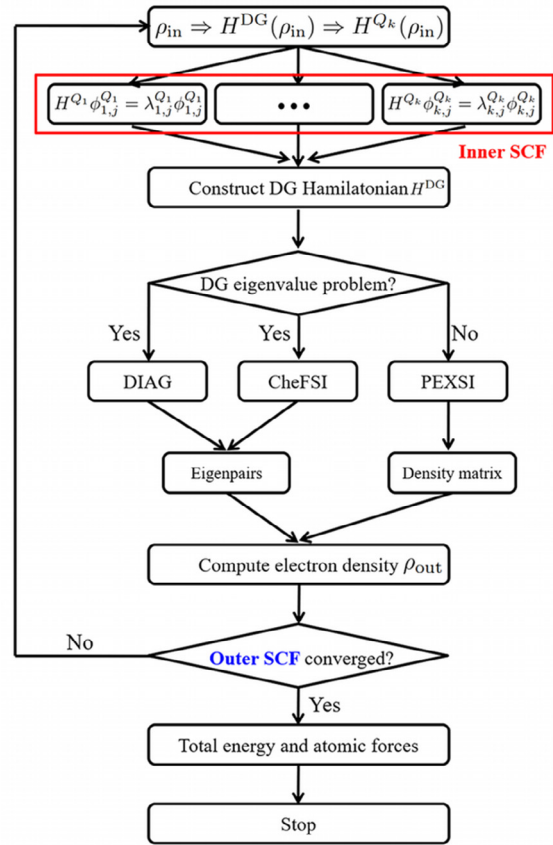

Fig. 2. Flowchart of the DGDFT method. There are four time-consuming parts in DGDFT, including generating the ALB functions in the inner SCF iterations, constructing and diagonalizing the DG Hamiltonian (DIAG, CheFSI and PEXSI), computing the electron density, total energy and atomic forces in the outer SCF iterations. H^{DG} and H^{Q_k} represent the global and local ($H^{Q_k}\phi_{k,j}^{Q_k} = \lambda_{k,j}^{Q_k}\phi_{k,j}^{Q_k}$ on Q_k) KS Hamiltonian, respectively. ρ_{in} and ρ_{out} represent the initial and output density, respectively.

ative algorithms have been developed to solve the KS-DFT equations discretized on the standard plane-wave basis set, such as Davidson [24] and LOBPCG [25] algorithms.

We compute the lowest J_k eigenvalues $\left\{\lambda_{k,j}^{Q_k}\right\}_{j=1}^{J_k}$ and corresponding eigenfunctions $\left\{\phi_{k,j}^{Q_k}\right\}_{j=1}^{J_k}$ on each extended element Q_k in PWDFT. We restrict and truncate $\left\{\phi_{k,j}^{Q_k}\right\}_{j=1}^{J_k}$ from the extended element Q_k to the element E_k, and obtain the truncated and orthogonal vectors $\left\{\phi_{k,j}\right\}_{j=1}^{J_k}$ on each element E_k, which are the so-called ALB functions. In the current framework of DGDFT, we set the same number $J_k = J_b = N_b/M$ of ALB functions on each element, where N_b is the total number of ALB functions and M is the number of elements partitioned in the global domain. Furthermore, the number of ALB functions used in each element is $J_b \approx 4-40N_e/M$, similar to the case of Gaussian and numerical atomic basis sets. It should be noticed that the ALB functions $\left\{\phi_{k,j}\right\}_{j=1}^{J_b}$ are truncated to zero outside of E_k and normalized orthogonally on corresponding local element E_k, which leads to their discontinuity across the boundary of E_k. Therefore, the ALB functions are orthogonal, localized, and complete basis sets that combine the advantages of both atomic localized basis sets (localization) and plane-wave basis sets (orthogonality and completeness).

In PWDFT, our default choice of the KS-DFT eigensolver is the LOBPCG [25] algorithm for small systems always containing less than 100 atoms. The LOBPCG algorithm iteratively solves the

KS-DFT eigenvalue problem of $HX = XE$ ($X = \{x_i(r)\}_{i=1}^{J_b} \in \mathbb{R}^{N_r^{Q_k} \times J_b}$) by searching the minimum of the $\mathrm{Tr}\left[X^T H X\right]$ with the orthogonality constraint $X^T X = I \in \mathbb{R}^{J_b \times J_b}$ in a subspace spanned by $3J_b$ vectors $[X, W, P] \in \mathbb{R}^{N_r^{Q_k} \times 3J_b}$. The eigenvectors X can be updated as

$$X = XC_X + WC_W + PC_P,\qquad(2)$$

where W is a preconditioned residual defined as

$$W = TR = T\left(HX - X\left(X^T H X\right)\right),\qquad(3)$$

where $R = HX - X\left(X^T H X\right)$ is the residual, and T is the Teter preconditioner widely used in the plane-wave basis set. P is the conjugate direction. The coefficients C_X, C_W and C_P can be computed with the lowest J_b eigenpairs of the projected $3J_b \times 3J_b$ generalized eigenvalue problem

$$S^T H S C = S^T S C E,\qquad(4)$$

where $S = [X, W, P]$ is the trial subspace and $C = [C_X, C_W, C_P]^T$ are the coefficients.

2.1.2. Constructing DG Hamiltonian

For the global KS equations, the wavefunctions $\psi_i(r)$ are expanded into a linear combination of ALB functions $\{\phi_{k,j}\}_{j=1}^{J_b}$ defined as

$$\psi_i(r) = \sum_{k=1}^{M} \sum_{j=1}^{J_b} C_{i;k,j}\phi_{k,j}(r).\qquad(5)$$

Under the ALB functions, solving the global Kohn–Sham equations becomes a linear eigenvalue problem

$$\sum_{k,j} H_{k'j';k,j}^{DG} C_{i;k,j} = \lambda_i C_{i;k',j'},\qquad(6)$$

where H^{DG} is the KS Hamiltonian matrix defined as

$$\begin{aligned}
H_{k'j';k,j}^{DG} &= \left(\frac{1}{2}\langle \nabla\phi_{k,j'}, \nabla\phi_{k,j}\rangle_{\mathcal{T}} + \langle \phi_{k,j'}, V_{\mathrm{eff}}\phi_{k,j}\rangle_{\mathcal{T}}\right)\delta_{k,k'} \\
&\quad + \left(\sum_{I,\ell}\gamma_{I,\ell}\langle \phi_{k',j'}, b_{I,\ell}\rangle_{\mathcal{T}}\langle b_{I,\ell}, \phi_{k,j}\rangle_{\mathcal{T}}\right) \\
&\quad + -\frac{1}{2}\langle [[\phi_{k'j'}]], \{\{\nabla\phi_{k,j}\}\}\rangle_{\mathcal{T}} \\
&\quad - \frac{1}{2}\langle \{\{\nabla\phi_{k'j'}\}\}, [[\phi_{k,j}]]\rangle_{\mathcal{T}} + \alpha\langle [[\phi_{k'j'}]], [[\phi_{k,j}]]\rangle_{\mathcal{T}},
\end{aligned}\qquad(7)$$

where V_{eff} denotes the effective one-body potential at the outer SCF iterations, including local pseudopotential V_{loc}, Hartree potential V_H and the exchange–correlation potential $V_{\mathrm{xc}}[\rho]$. $b_{I,\ell}$ and $\gamma_{I,\ell}$ correspond to the nonlocal pseudopotential. For each atom I, there are L_I functions $b_{I,\ell}$, called the projection vector of the nonlocal pseudopotential. The parameter $\gamma_{I,\ell}$ is a real valued scalar. $\langle \cdot, \cdot\rangle_{\mathcal{T}}$ is the sum of the inner product on each element, and $\langle \cdot, \cdot\rangle_{\mathcal{T}}$ is the sum of the inner product on each surface. The symbols $\{\{\cdot\}\}$ and $[[\cdot]]$ represent the average and the jump operators across surfaces respectively used to account for the discontinuity of the basis functions.

In particular, the submatrix $H_{k'j';k,j}^{DG}$ is the (k', k)-th matrix block of $H^{DG} \in \mathbb{R}^{J_b \times J_b}$. These three group terms in Eq. (7) reflect different contributions to the DG Hamiltonian. The first group term represents the kinetic energy and the local pseudopotential, and only contributes to the diagonal blocks $H_{k'j';k,j}^{DG}$. The second group term represents the nonlocal pseudopotentials, and contributes to both the diagonal and off-diagonal blocks of H^{DG}. These two group terms are similar to the case of atomic localized basis sets, such as

numerical atomic basis orbitals implemented in SIESTA. However, the third group term consists of the contributions from boundary integrals different from SIESTA, and contributes to both the diagonal and off-diagonal blocks of H^{DG} as well. Each boundary term involves only two neighboring elements by definition as plotted in Fig. 1a. Therefore, H^{DG} is a block-tridiagonal sparse matrix and the nonzero matrix blocks correspond to interactions between neighboring elements as shown in Fig. 1c.

2.1.3. Diagonalizing DG Hamiltonian

After the DG Hamiltonian is constructed by using the ALB functions, the next step is to solve a standard eigenvalue problem to diagonalize the DG Hamiltonian and obtain other basic physical quantities, such as electron density, total energy and atomic forces. The conventional method is to directly and explicitly diagonalize the DG Hamiltonian by using the standard parallel linear algebra software packages for dense matrices, such as the ScaLAPACK subroutine PDSYEVD (referred as DIAG). The DIAG method is expensive and not scalable on modern heterogeneous supercomputers, because its computational cost scales as $\mathcal{O}\left(N_b^3\right)$ and its parallel scalability is limited to hundreds of cores [17,18]. In the current framework of DGDFT, we utilize the Chebyshev polynomial filtered subspace iteration (CheFSI) [26] algorithm to solve the DG eigenvalue problem $H^{DG} C = \Lambda C$, where $H^{DG} \in \mathbb{R}^{N_b \times N_b}$ and $C \in \mathbb{R}^{N_b \times N_e}$ [19].

The CheFSI algorithm uses a Chebyshev polynomial $p_m(\lambda)$ to construct the map eigenvalues at the low end of occupied states H^{DG} to the dominant eigenvalues of $p_m\left(H^{DG}\right)$. The exponential growth property of the Chebyshev polynomials outside the region $[-1,1]$ can be used to obtain the wanted occupied states, while other unwanted regions are damped in comparison.

During each CheFSI iteration, $p_m\left(H^{DG}\right) \in \mathbb{R}^{N_b \times N_b}$ can be applied to a block of vectors $X = \{x_i\}_{i=1}^{N_e} \in \mathbb{R}^{N_b \times N_e}$ by using the three-term recurrence satisfied by Chebyshev polynomials, written as

$$\begin{aligned}
y_{i;k,j} &= \sum_{k'=1}^{M} \sum_{j'}^{J_b} H_{k,j;k'j'}^{DG} x_{i;k'j'} \\
&= \sum_{k' \in \mathcal{N}(k)} \sum_{j'}^{J_b} H_{k,j;k'j'}^{DG} x_{i;k'j'},
\end{aligned}\qquad(8)$$

where $\mathcal{N}(k)$ denotes the collection of the neighboring elements of the element E_k. This dense matrix–matrix multiplication can be carried out independently over the various columns of X, which takes advantage of the embarrassingly parallel nature of the problem by distributing the columns among separate processing elements.

The key step in the CheFSI algorithm is to project the DG Hamiltonian H^{DG} onto the occupied subspace

$$\hat{H} = \hat{Y}^T H^{DG} \hat{Y},\qquad(9)$$

where $\hat{Y}$ is the orthonormal vectors for the Chebyshev polynomial filtered block of vectors $Y = \{y_i\}_{i=1}^{N_e} \in \mathbb{R}^{N_b \times N_e}$. The eigenvalues Λ and eigenvectors $X \in \mathbb{R}^{N_b \times N_e}$ can be computed by directly diagonalizing the projected DG Hamiltonian $\hat{H} \in \mathbb{R}^{N_e \times N_e}$.

There are three advantages for the CheFSI algorithm to take advantage of the features of ALB functions (Orthogonality and localization) and block-tridiagonal sparse DG Hamiltonian matrix. Firstly, because the ALB functions are orthogonal, we can readily employ the orthogonal CheFSI algorithm, which avoids to compute and orthogonalize the overlap matrix. Secondly, because the ALB functions are localized and completed basis sets, the number (N_b) of ALB functions is comparable to localized basis sets, such as

Gaussian and atomic numerical basis sets. Compared to the cubic scaling $O(N_b^3)$ of the DIAG method, the CheFSI method can reduce the computational cost to $O(N_b N_e^2 + N_e^3)$. It should be noticed that $N_b \approx 4-40 N_e$ for the ALB functions in DGDFT. Therefore, the CheFSI method can speed up more than two orders of magnitude faster than the DIAG method. Finally, orthogonal and localized ALB functions result in the block-sparse structure of the DG Hamiltonian matrix even for metallic systems. This feature can reduce the computational cost of the DG Hamiltonian matrix applied to a block of dense vectors with low data communications.

The third method for diagonalizing the block-tridiagonal sparse DG Hamiltonian H^{DG} is the pole expansion and selected inversion (PEXSI) technique [27]. The PEXSI method is more efficient and scalable than the DIAG and CheFSI methods, because the PEXSI method does not require computing eigenvalues and eigenvectors of H^{DG}. The PEXSI method is designed for sparse matrix operations to take advantage of massively parallel supercomputers with high scalability, which can scale up to 100,000 processors [17,18]. But the PEXSI method is difficult to be implemented and ported on the Sunway TaihuLight supercomputer. Therefore, we choose the CheFSI method to diagonalize the DG Hamiltonian in DGDFT in this work.

2.1.4. Computing electron density

After constructing the H^{DG} matrix and solving the eigenvalue problem, the electron density can be evaluated from

$$
\begin{aligned}
\rho(r) &= \sum_{i=1}^{N_e} |\psi_i(r)|^2 \\
&= \sum_{k=1}^{M} \sum_{j=1}^{J_b} \sum_{k'=1}^{M} \sum_{j'=1}^{J_b} \phi_{k,j}(r) \phi_{k',j'}(r) \left(\sum_{i=1}^{N_e} C_{i;k,j} C_{i;k',j'} \right) \\
&= \sum_{k=1}^{M} \sum_{j=1}^{J_b} \sum_{k'=1}^{M} \sum_{j'=1}^{J_b} \phi_{k,j}(r) \phi_{k',j'}(r) P_{k,j;k',j'},
\end{aligned}
\tag{10}
$$

where P is the density matrix approximated as a matrix function of H_{DG} without knowing $C_{i;k,j}$ explicitly, defined as

$$
P_{k,j;k',j'} = \sum_{i=1}^{N_e} C_{i;k,j} C_{i;k',j'},
\tag{11}
$$

where, each ALB function $\phi_{k,j}(x)$ is strictly localized in the element E_k to eliminate the cross terms involving both k and k'. As a result, the selected blocks, or more specifically, the diagonal blocks of the density matrix $P_{k,j;k,j'}$ are needed to evaluate the electron density.

Other cheap parts, such as total energy [16] and atomic forces [20], can be efficiently computed in the formwork of DGDFT. The total energy in the formwork of DGDFT can be written as

$$
\begin{aligned}
E_{DG}(\{\psi_i\}) &= \frac{1}{2} \sum_{i=1}^{N_e} \langle \nabla \psi_i, \nabla \psi_i \rangle_{\mathscr{T}} + \langle V_{eff}, \rho \rangle_{\mathscr{T}} + \sum_{I=1}^{N_A} \sum_{\ell=1}^{L_I} \gamma_{I,\ell} \\
&\quad \times \sum_{i=1}^{N_e} |\langle b_{I,\ell}, \psi_i \rangle_{\mathscr{T}}|^2 - \sum_{i=1}^{N_e} \langle \{\{\nabla \psi_i\}\}, [[\psi_i]] \rangle_{\mathscr{T}} \\
&\quad + \alpha \sum_{i=1}^{N_e} \langle [[\psi_i]], [[\psi_i]] \rangle_{\mathscr{T}}.
\end{aligned}
\tag{12}
$$

The atomic forces in the DGDFT method are computed with the Hellmann–Feynman theory, written as

$$
F_I^{HF} = \int \rho_{loc,I}(r - R_I) \nabla V_H(r) dr + 2 \sum_{\ell=1}^{L_I} Tr[V_{nl,I,\ell} P] + \sum_{J \neq I} \frac{Z_I Z_J}{|R_I - R_J|^3} (R_I - R_J),
\tag{13}
$$

where $\rho_{loc,I}$ is the local pseudocharge of atom I. It should be noticed that the computational cost of the Hellmann–Feynman force in DGDFT scales linearly with respect to the system size [20].

We have demonstrated that high plane-wave accuracy in the total energy and atomic forces can be achieved with a very small number (4–40) of basis functions per atom in the formwork of DGDFT, compared to fully converged plane-wave calculations [17].

2.2. Parallel implementation of DGDFT

We present the scalable implementation of the DGDFT method based on the two-level parallelization strategy combining with the multi-thread parallelism of Sunway master–slave heterogeneous architecture, thus resulting in extreme-scale HPC DFT calculations on the Sunway TaihuLight supercomputer.

2.2.1. Two level parallelization strategy of DGDFT

In the framework of DGDFT, there is two-level of parallelization that deals with different types of data distribution and communication, and task scheduling schemes as shown in Fig. 3. The DGDFT method use the Message Passing Interface (MPI) for parallel programming to deal with the data communications between different MPI processes.

The first main level is called inter-element parallelization between neighboring elements. The main computation of this level is to construct the DG Hamiltonian matrix by using the ALB functions as shown in Fig. 3a, which uses the column block MPI grid partition for the global 2D MPI process grid. Because the ALB functions are orthogonal and localized basis sets, the DG Hamiltonian matrix can keep in a sparse block-tridiagonal structure unchanged during the outer SCF iterations even for metallic systems. The local
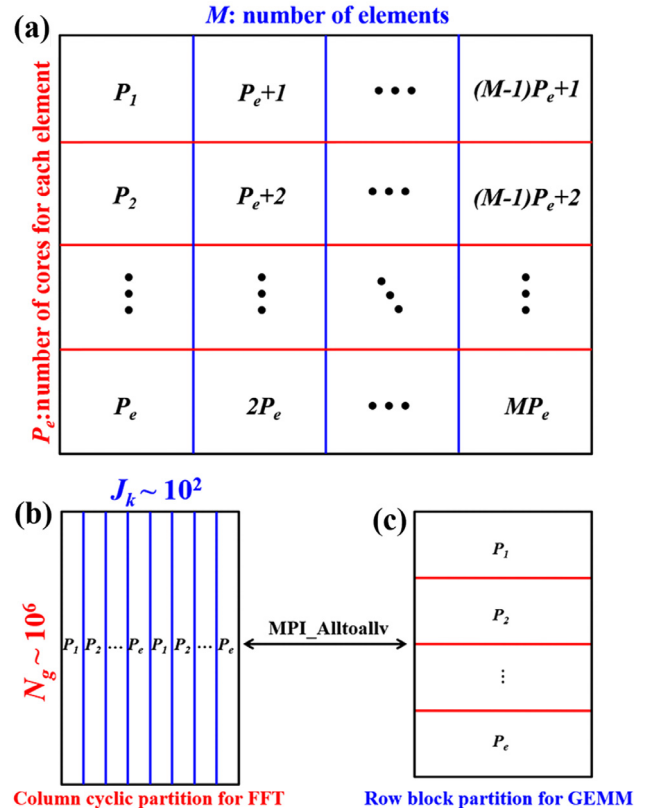

Fig. 3. Parallel data distribution, task scheduling, and data communication of DGDFT. (a) 2D MPI process grid for two level parallelization strategy of DGDFT, especially for constructing and diagonalizing the DG Hamiltonian matrix $H^{DG} \in \mathbb{R}^{N_b \times N_b}$. M is the number of elements partitioned in the system. P_e is the number of MPI processes used in each element. (b) Band parallelization with column cyclic partition (for FFT) and (c) grid parallelization with row block partition (for GEMM), especially for the tall-and-skinny wavefunction matrix $\Phi^{Q_k} \in \mathbb{R}^{N_r^{Q_k} \times J_k}$ ($J_k = J_b \approx 4-40 N_e/M$) on the extended element Q_k in PWDFT.

data communication between neighboring elements is dealt with the MPI programming. In the current framework of DGDFT, the number of MPI processes in this level is fixed and equal to the number M of elements. It should be noticed that this level of inter-element parallelization in DGDFT is highly scalable due to the local data communication between neighboring elements for constructing the block-tridiagonal DG Hamiltonian matrix on tens of thousands of cores on modern heterogeneous HPC supercomputers.

The secondary level is called intra-element parallelization on each element. This level parallelization uses the row block MPI grid partition for the global 2D MPI process grid. The generation of ALB functions for solving the local Kohn–Sham equations on each element in the inner SCF iterations can be efficiently parallelized similar to the case of conventional standard plane-wave DFT software packages, such as VASP [13] and QUANTUM ESPRESSO [14]. The DGDFT software includes a self-contained module called PWDFT [23] for performing conventional standard plane-wave based electronic structure calculations. It should be noticed that plane-wave basis sets always require relatively large number of basis functions for high-accuracy KS-DFT calculations. There are two types of parallelization, called the band and grid parallelization (Fig. 3b and c) [15], in the inner SCF iterations of PWDFT for each row block MPI process P_e in the global 2D MPI process grid. In particular, for the tall-skinny wavefunction matrix $\Phi^{Q_k} \in \mathbb{R}^{N_r^{Q_k} \times J_b}$ on the extended element Q_k in PWDFT, the band parallelization is to use the column cyclic partition especially for FFTs, while the grid parallelization is to use row block partition especially for GEMMs. We use the MPI_Alltoallv function to transfer two types of data partition between the band and grid parallelization. However, large basis sets are not conducive to take full advantage of HPC on modern heterogeneous supercomputers due to the high all-to-all data communication of the dense Hamiltonian matrix generated in large uniform plane-wave basis set. Therefore, PWDFT can only deal with small-scale systems containing hundreds of atoms and scale to thousands of cores [23]. Because each element only contains less than tens of atoms in DGDFT, in the intra-element parallelization, we perform the small-scale KS-DFT calculations for solving the local KS equations by using $P_e < 200$ cores on each element with ultrahigh parallel efficiency of 95% in the inner SCF iterations of PWDFT. In the current framework of DGDFT, the maximum number can be used this level for PWDFT is $J_b = N_b/M$, thus, $1 \leqslant P_e \leqslant N_b/M$. It should be noticed that this level of intra-element parallelization in DGDFT only requires to scale to hundreds of cores for such small-scale KS-DFT calculations.

For the global outer SCF iterations, diagonalizing the DG Hamiltonian matrix is the most expensive part of DGDFT for large-scale materials simulations, three diagonalization methods (DIAG, CheFSI and PEXSI) [17,19] can directly take advantage of such two-level parallelization strategy of 2D MPI process grid in the DGDFT method. From such two-level parallelization strategy, the minimum number N_{min} and maximum number N_{max} of MPI processes used in DGDFT are computed as $N_{min} = M$ and $N_{max} = MJ_b = N_b$. By using this two-level parallelization strategy, DGDFT is highly scalable on hundreds of thousands of cores on modern heterogeneous X86 HPC supercomputers [17].

2.2.2. DGDFT on Sunway TaihuLight supercomputer

The Sunway TaihuLight [21] is the new generation of Chinese home-grown supercomputer, and it ranks the No. 3 on the top500 list in 2019. It consists of 40,960 domestic-designed SW26010 processors, which is based on a master–slave heterogeneous architecture. It should be noted that Sunway processor uses a Reduced Instruction Set Computer (RISC) design, different from the widely used X86 Complex Instruction Set Computer (CISC)

architecture. The architecture of the SW26010 processor is shown in Fig. 4. Each SW26010 processor chip contains 4 core groups (CGs), and each CG acts as a master–slave many-core module. In a single CG module, one management processing element (MPE) works as the master core and 64 computing processing elements (CPEs) arranged in an 8×8 grid serve as the slave cores.

We implement the DGDFT software package in the C/C++ programming language and uses the message passing interface (MPI) for parallel programming. For DGDFT, each MPI process runs on a CG with MPE as a master processing core and can be effectively multi-thread accelerated by 64 PCEs as slave processing cores on the SW26010 processor, similar to the OpenMP and GPU parallel programming.

In particular, the two-level parallelization strategy of DGDFT acts as a process-level parallelism between different CGs and the slave processing acceleration can be considered as a thread-level parallelism by using the 64 slave processing cores on each CG on the Sunway TaihuLight supercomputer. Therefore, the minimum and maximum numbers of processing cores used in DGDFT for the KS-DFT calculations respectively are $65M$ and $65N_b$, where M is the number of elements and N_b is the number of the ALB functions used in the system. Compared to the DGDFT [17] implemented in the Edison supercomputer at the USA NERSC platform, each CG on the Sunway TaihuLight supercomputer can act as a CPU processing core in the Edison supercomputer (includes 5,462 Cray XC30 nodes and each node has 24 cores partitioned among two Intel Ivy Bridge processors). The Sunway TaihuLight supercomputer includes 10,649,600 processing cores, which are much larger than that (131,088 processing cores) in the Edison supercomputer.

The time-consuming cost in DGDFT is spent in the matrix operations, such as the vector-vector, matrix–vector, and matrix–matrix multiplications (DGEMM), matrix diagonalization (DSYEVD), matrix Cholesky factorization (DPOTRF) and fast Fourier transform (FFT). All these matrix operations can be realized through the Basic Linear Algebra Subprograms (BLAS), Linear Algebra PACKage (LAPACK) and Fastest Fourier Transform in the West (FFTW) libraries. Fortunately, most of these subroutines in the BLAS and LAPACK libraries have been reimplemented into the xMath library and accelerated by the slave processing CPEs on each CG of SW26010 processors through the thread-level parallelism on the Sunway TaihuLight supercomputer.
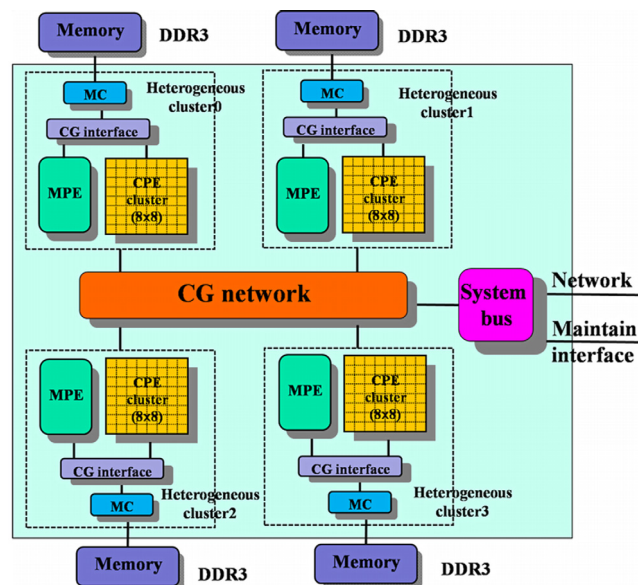

Fig. 4. The SW26010 processor architecture in the Sunway TaihuLight supercomputer [21].

3. Results and discussion

In this section, we demonstrate the computational efficiency and parallel scalability of the DGDFT method to accelerate large-scale KS-DFT calculations on the Sunway TaihuLight supercomputer. We have implemented the DGDFT method as software package also called DGDFT [17], which has been written in the C/C++ programming language with the message passing interface (MPI) for parallel programming. DGDFT supports the Hartwigsen-Goedecker-Hutter (HGH) [28] norm-conserving pseudo-potential. In this work, we use the exchange–correlation functional of local density approximation of Goedecker-Teter-Hutter (LDA-PZ) [29] to describe the electronic structures of metallic graphene systems. It should be noticed that the computational accuracy of the DGDFT method is comparable to standard plane-wave KS-DFT calculations, such as QUANTUM ESPRESSO [14], which has already been validated in our previous works [16–20].

We use the DGDFT method to study the electronic structures of three graphene systems, G180, G2880 and G11520, containing 180, 2880 and 11,520 carbon atoms, respectively. The G2880 and G11520 systems are generated from extending the G180 system in the x and y directions with 4×4 and 8×8 supercells, respectively. We set the kinetic energy cutoff E_{cut} = 55 Ha for three metallic graphene systems, which can reach high accuracy level [20] of 10^{-5} Ha/atom in terms of the error of total energy and 10^{-4} Ha/Bohr in terms of the error of atomic forces, respectively.

Table 1 lists the computational parameters of graphene (G180, G2880 and G11520) systems used in the DGDFT method, including the number N_r of grid points, the numbers N_A of carbon atoms, the numbers N_e of electrons, the number J_b of the ALB functions used in each element, the numbers M of elements, the numbers N_b of the ALB functions, and the minimum $N_{min} = M$ and maximum $N_{max} = MJ_b = N_b$ numbers of CGs used on the Sunway TaihuLight supercomputer. The total number of grid points N_r in real space is determined from the kinetic energy cutoff E_{cut} defined as $(N_r)_i = \sqrt{2E_{cut}}L_i/\pi$, where L_i is the length of supercells along the i-th (x, y and z) coordinate direction.

To illustrate the computational efficiency and parallel scalability of the DGDFT method, we demonstrate the computational time of four time-consuming parts for 2D metallic graphene system (G2880 and G11520) without or with the master–slave multi-thread parallelism on the Sunway TaihuLight supercomputer, including generating the ALB functions, constructing and diagonalizing the DG Hamiltonian and computing the electron density, as shown in Figs. 5 and 6. There are some additional steps such as computing total energy and atomic forces, which are all included in the total wall clock time of outer SCF iterations in the DGDFT calculations.

3.1. Master process parallelism

We first validate the parallel scalability of DGDFT when only using the master process parallelism on the CGs of Sunway archi-

tecture but without slave multi-thread acceleration for the G2880 system. In this case, each CG acts as an MPI process similar to a core in the CPU processor of the widely used X86 architecture. For the G2880 system, we set a large number (45.5) of ALB functions per atom (J_b = 128, M = 1024 and N_b = 131,072) and 5 times inner SCF iterations to generate the ALB functions for achieving high accuracy. In this case, generating the ALB functions becomes the most expensive part in the KS-DFT calculations in DGDFT.

The total time of the DGDFT calculations for the G2880 system is 5024.35 and 257.45 s by using N_{min} = 1024 and N_{max} = 131,072 CGs, respectively. The parallel efficiency is only 15.24% (The speedup is 19.51) when using 131,072 master CGs. In detail, when using 1024 master CGs, the time for four expensive parts are 4296.46 s for generating the ALB functions, 227.19 and 201.33 s for constructing and diagonalizing the DG Hamiltonian, and 201.33 s for computing the electron density, respectively. The time of generating the ALB functions, constructing the DG Hamiltonian, and computing the electron density is reduced to 33.79, 3.05 and 2.44 s, respectively, when using 131,072 master CGs. And corresponding parallel efficiencies can achieve as high as 99.33%, 58.19%, and 64.46%, respectively.

The main part of reducing the total parallel efficiency is to diagonalize the DG Hamiltonian by using the CheFSI method in DGDFT. It should be noticed that the time of diagonalizing the DG Hamiltonian is only reduced from 227.19 to 210.61 s when increasing the number of master CGs from 1024 to 131,072. The major bottleneck of the CheFSI method is to solve the projected subspace eigenvalue problem, which can only use the column block MPI grid partition (1024 MPI processes) in the global 2D MPI process grid. The time for solving this small eigenvalue problem is almost unchanged and dominated in the CheFSI method for the G2880 system. Other parts in the CheFSI method only require the block-matrix multiplications, which can ideally take advantage of the global 2D MPI process grid. Therefore, the computation of these parts is highly scalable parallelized by using the master process parallelism of CGs and the time is negligible in this case for the G2880 system.

3.2. Master–slave multi-thread parallelism

The major advantage of Chinese home-grown SW26010 processors is based on a new Sunway master–slave heterogeneous architecture, which can be efficiently accelerated by the master–slave multi-thread parallelism on the Sunway TaihuLight supercomputer. In this case, when using the master–slave multi-thread parallelism for the G2880 system, the total time can be further reduced to 2863.05 and 69.17 s by using N_{min} = 1024 and N_{max} = 131,072 CGs, respectively, which is much faster than the case (5024.35 and 257.45 s) of the master process parallelism. Furthermore, the total parallel efficiency increases to 32.33% (the speedup is 41.39) when using 131,072 CGs with the master–slave multi-thread parallelism.

In detail, when using 1024 CGs (66,560 processing cores), the time for four expensive parts are 2434.22 s for generating the ALB functions, 170.54 and 36.90 s for constructing and diagonalizing the DG Hamiltonian, and 200.89 s for computing the electron density, respectively. In particular, three parts of generating the ALB functions, constructing and diagonalizing the DG Hamiltonian have been accelerated by factors of 1.76, 1.33 and 5.45 by using the master–slave multi-thread parallelism.

But the part of computing the electron density has not been accelerated by using such master–slave multi-thread parallelism, that is because that most matrix operations of this part are real-to-complex FFTs and MPI data commutations between CGs, which can not benefit from the master–slave multi-thread parallelism on the Sunway TaihuLight supercomputer.
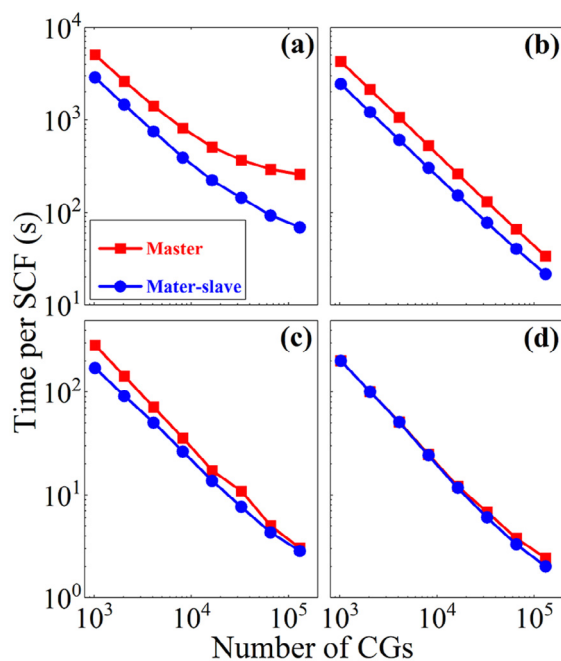
Table 1
Computational parameters of graphene (G180, G2880 and G11520) systems in DGDFT, including the numbers N_e of electrons, the number N_r of grid points in real space, the number J_b of the ALB functions used in each element, the numbers M of elements, the numbers N_b of the ALB functions, and the minimum ($N_{min} = M$) and maximum ($N_{max} = MJ_b = N_b$) numbers of MPI processes (CGs) used on the Sunway TaihuLight supercomputer.

Systems	N_e	N_r	J_b	N_{min}	N_{max}
G180	720	633,600	200	16	3200
G2880	11,520	183,500,800	128	1024	131,072
G11520	46,080	552,075,264	100	2304	230,400

Fig. 5. The wall clock time with respect to the number of CGs (each CG includes 65 processing cores) without or with the master–slave multi-thread parallelism on the Sunway TaihuLight supercomputer used for 2D metallic graphene system G2880 containing 2,880 carbon atoms. Strong scaling of (a) total computational time per outer SCF iteration, including the time for (b) generating the ALB functions, (c) constructing the DG Hamiltonian, and (d) computing the electron density.
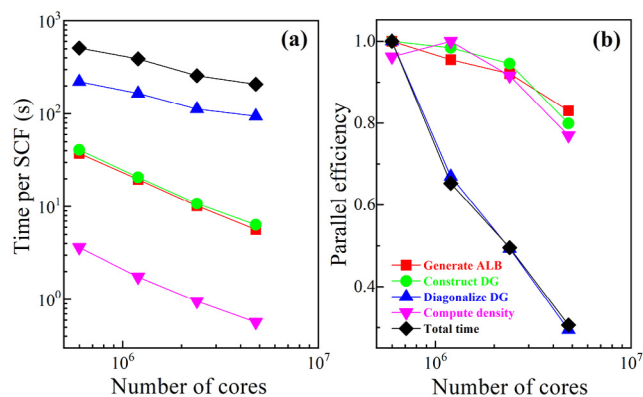

Fig. 6. The wall clock time with respect to the number of cores by using the master–slave multi-thread parallelism on the Sunway TaihuLight supercomputer used for 2D metallic graphene system G11520 containing 11,520 carbon atoms. (a) Strong scaling and (b) parallel efficiency of total computational time per SCF iteration, including the time for generating the ALB functions, constructing and diagonalizing the DG Hamiltonian, and computing the electron density.

3.3. Tens of thousands of atoms materials simulations

As we know that it is a major challenge to perform the KS-DFT calculations for first-principles materials simulations on ultra-large-scale systems containing tens of thousands of atoms, especially for metallic systems. Fig. 6 shows the wall clock time with respect to the number of cores by using the master–slave multi-thread parallelism on the Sunway TaihuLight supercomputer for the G11520 system containing 11,520 carbon atoms. Because these KS-DFT calculations are ultra-large-scale materials simulations, we only set a small number (20.0) of ALB functions per atom ($J_b = 100$, $M = 2304$ and $N_b = 230,400$) and once inner SCF iteration to generate the ALB functions for the G11520 system. In this case, diagonal-

izing the DG Hamiltonian becomes the most expensive part in such ultra-large-scale KS-DFT calculations the DGDFT method. Other three parts become much cheaper and more scalable than that of diagonalizing the DG Hamiltonian.

The total time of the DGDFT calculations for the G11520 system is 509.90 and 208.2 s when respectively using 9216 and 73,728 CGs (599,040 and 4,792,320 cores) with the master–slave multi-thread parallelism. The parallel efficiency is 30.61% (The speedup is 2.45) when using 73,728 CGs (4,792,320 cores). In detail, when using 599,040 cores, the time for four expensive parts is 37.51 s for generating the ALB functions, 40.80 and 221.75 s for constructing and diagonalizing the DG Hamiltonian, and 3.63 s for computing the electron density, respectively. These results are reduced to 5.65, 1.38, 94.16, and 0.56 s, respectively, when using 4,792,320 cores. It should be noticed that three cheap parts, including generating the ALB functions, constructing the DG Hamiltonian and computing the electron density, show high parallel efficiencies up to 82.99%, 79.90%, and 76.95%, respectively, when using 4,792,320 cores. But the parallel efficiency of the expensive part to diagonalize the DG Hamiltonian by using the CheFSI method is only 29.43% when using 4,792,320 cores.

4. Conclusion and outlook

In summary, we demonstrate that DGDFT can be used to push the envelope to investigate the electronic structures of ultra-large-scale metallic systems containing tens of thousands of atoms by combing with the two-level parallelization strategy of DGDFT and the master–slave multi-thread heterogeneous parallelism of the Sunway TaihuLight supercomputer. We show that DGDFT can achieve a high parallel efficiency up to 32.3% (speedup as high as 42,382.9) by using 8,519,680 processing cores (131,072 core groups) on the Sunway TaihuLight supercomputer.

For diagonalizing the block-tridiagonal sparse DG Hamiltonian in the DGDFT method, the PEXSI method is more efficient and scalable than the CheFSI method [17]. But the PEXSI method is difficult to be implemented with the heterogeneous CPU and GPU multi-thread parallelism. In the future work, we try to optimize the parallel implementation of the PEXSI method on the Sunway Taihu-Light supercomputer.

Conflict of interest

The authors declare that they have no conflict of interest.

Acknowledgments

Author contributions

Wei Hu, Hong An and Jinlong Yang designed and supported this project. Wei Hu and Xinming Qin implemented the DGDFT code and performed all the calculations on the Sunway TaihuLight supercomputer. Qingcai Jiang, Junshi Chen, Hong An and Weile Jia optimized the DGDFT code on the Sunway TaihuLight supercomputer. Fang Li, Xin Liu, and Dexun Chen provided the computational resources and compiled the DGDFT code on the Sunway TaihuLight supercomputer. Fangfang Liu and Yuwen Zhao provided the xMath library for DGDFT.

References

[1] Hohenberg P, Kohn W. Inhomogeneous electron gas. Phys Rev 1964;136:B864.
[2] Kohn W, Sham LJ. Self-consistent equations including exchange and correlation effects. Phys Rev 1965;140:A1133.
[3] Goedecker S. Linear scaling electronic structure methods. Rev Mod Phys 1999;71:1085.
[4] Shang H, Xiang H, Li Z, et al. Linear scaling electronic structure calculations with numerical atomic basis set. Int Rev Phys Chem 2010;29:665–91.
[5] Bowler DR, Miyazaki T. O(n) methods in electronic structure calculations. Rep Prog Phys 2012;75:036503.
[6] Yang W. Electron density as the basic variable: a divide-and-conquer approach to the ab initio computation of large molecules. J Mol Struct Theochem 1992;255:461–79.
[7] Zhao Z, Meza J, Wang L. A divide-and-conquer linear scaling three-dimensional fragment method for large scale electronic structure calculations. J Phys Condens Matter 2008;20:294203–10.
[8] Frisch MJ, Pople JA, Binkley JS. Self-consistent molecular orbital methods 25. Supplementary functions for gaussian basis sets. J Chem Phys 1984;80:3265–9.
[9] Soler JM, Artacho E, Gale JD, et al. The siesta method for *ab initio* order-n materials simulation. J Phys Condens Matter 2002;14:2745.
[10] VandeVondele J, Borstnik U, Hutter J. Linear scaling self-consistent field calculations with millions of atoms in the condensed phase. J Chem Theory Comput 2012;8:3565–73.
[11] Gillan MJ, Bowler DR, Torralba AS, et al. Order-n first-principles calculations with the conquest code. Comput Phys Commun 2007;177:14–8.
[12] Qin X, Shang H, Xiang H, et al. HONPAS: A linear scaling open-source solution for large system simulations. Int J Quantum Chem 2015;115:647–55.
[13] Kresse G, Hafner J. *Ab initio* molecular dynamics for liquid metals. Phys Rev B 1993;47:558.
[14] Giannozzi P, Baroni S, Bonini N, et al. QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials. J Phys Condens Matter 2009;21:395502.
[15] Bottin F, Leroux S, Knyazev A, et al. Large-scale *ab initio* calculations based on three levels of parallelization. Comp Mater Sci 2008;42:329–36.
[16] Lin L, Lu J, Ying L. Adaptive local basis set for Kohn-Sham density functional theory in a discontinuous Galerkin framework I: total energy calculation. J Comput Phys 2012;231:2140–54.
[17] Hu W, Lin L, Yang C. DGDFT: A massively parallel method for large scale density functional theory calculations. J Chem Phys 2015;143:124110.
[18] Hu W, Lin L, Yang C. Edge reconstruction in armchair phosphorene nanoribbons revealed by discontinuous galerkin density functional theory. Phys Chem Chem Phys 2015;17:31397–404.
[19] Banerjee AS, Lin L, Hu W, et al. Chebyshev polynomial filtered subspace iteration in the discontinuous galerkin method for large-scale electronic structure calculations. J Chem Phys 2016;145:154101.
[20] Zhang G, Lin L, Hu W, et al. Adaptive local basis set for Kohn-Sham density functional theory in a discontinuous Galerkin framework II: force, vibration, and molecular dynamics calculations. J Comput Phys 2017;335:426–43.
[21] Fu H, Liao J, Yang J, et al. The Sunway TaihuLight supercomputer: system and spplications. Sci China Inf Sci 2016;59:72001.
[22] Lin L, García A, Huhs G, et al. SIESTA-PEXSI: Massively parallel method for efficient and accurate *ab initio* materials simulation without matrix diagonalization. J Phys Condens Matter 2014;26:305503.
[23] Hu W, Lin L, Banerjee AS, et al. Adaptively compressed exchange operator for large-scale hybrid density functional calculations with applications to the adsorption of water on silicene. J Chem Theory Comput 2017;13:1188–98.
[24] Davidson E. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. J Comput Phys 1975;17:87–94.
[25] Knyazev AV. Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method. SIAM J Sci Comput 2001;23:517–41.
[26] Zhou Y, Saad Y, Tiago ML, et al. Self-consistent-field calculations using chebyshev-filtered subspace iteration. J Comput Phys 2006;219:172–84.
[27] Lin L, Chen M, Yang C, et al. Accelerating atomic orbital-based electronic structure calculation via pole expansion and selected inversion. J Phys Condens Matter 2013;25:295501.
[28] Hartwigsen C, Goedecker S, Hutter J. Relativistic separable dual-space gaussian pseudopotentials from H to Rn. Phys Rev B 1998;58:3641.
[29] Goedecker S, Teter M, Hutter J. Separable dual-space gaussian pseudopotentials. Phys Rev B 1996;54:1703.
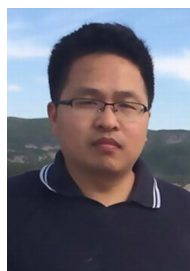
Wei Hu is currently a Professor at the Hefei National Laboratory for Physical Sciences at the Microscale (HFNL) of the University of Science and Technology of China (USTC). He received his Ph.D. degree in Theoretical and Computational Chemistry from the USTC in 2013. Then, he joined the Computational Research Division of Lawrence Berkeley National Laboratory as a postdoctoral fellow for developing high performance computing (HPC) density functional theory (DFT) software. His research interest focuses on the method development and scientific applications of large-scale DFT calculations.
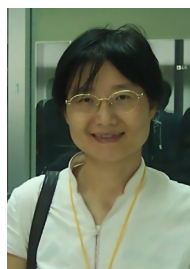
Hong An is currently a Professor at School of Computer Science and Technology, University of Science and Technology of China (USTC). She received her Ph.D. degree in Computer Science from the USTC in 2000. She is the director of the Advanced Computer System Architecture (ACSA) Lab in the USTC. Her main research focuses on parallel computer architecture, parallel programming, operating system design, and high performance computing.

Jinlong Yang is currently a Professor at the University of Science and Technology of China (USTC). He received his Ph.D. degree in Condensed Matter Physics from USTC in 1991. He became the Dean of the School of Chemistry and Materials Science of USTC in 2009 and the Vice President of USTC in 2018. His research interest focuses on the development of first-principles methods and their applications to clusters, nanostructures, solid materials, surfaces, and interfaces.