Subquadratic-scaling real-space random phase approximation correlation energy calculations for periodic systems with numerical atomic orbitals

Rong Shi [1,2] Peize Lin,[2,3] Min-Ye Zhang,[2] Lixin He,[1,*] and Xinguo Ren [2,3,†]

[1]*CAS Key Laboratory of Quantum Information, University of Science and Technology of China, Hefei 230026, Anhui, China*

[2]*Institute of Physics, Chinese Academy of Sciences, Beijing 100190, China*

[3]*Songshan Lake Materials Laboratory, Dongguan 523808, Guangdong, China*

(Received 21 July 2023; revised 24 November 2023; accepted 6 December 2023; published 2 January 2024)

The random phase approximation (RPA) as formulated as an orbital-dependent, fifth-rung functional within the density functional theory framework offers a promising approach for calculating the ground-state energies and the derived properties of real materials. Its widespread use to large-size, complex materials is, however, impeded by the significantly increased computational cost, compared to lower-rung functionals. The standard implementation exhibits an $O(N^4)$-scaling behavior with respect to system size N. In this work, we develop a low-scaling RPA algorithm for periodic systems, based on the numerical atomic orbital (NAO) basis-set framework and a localized variant of the resolution of identity (RI) approximation. The rate-determining step for RPA calculations, the evaluation of noninteracting response function matrix, is reduced from $O(N^4)$ to $O(N^2)$ by just exploiting the sparsity of the RI expansion coefficients, resultant from localized RI (LRI) scheme and the strict locality of NAOs. The computational cost of this step can be further reduced to linear scaling if the decay behavior of the Green's function in real space can be further taken into account. Benchmark calculations against existing **k**-space-based implementation confirms the validity and high numerical precision of the present algorithm and implementation. The new RPA algorithm allows us to readily handle three-dimensional, close-packed solid-state materials with over 1000 atoms. The algorithm and numerical techniques developed in this work also have implications for developing low-scaling algorithms for other correlated methods to be applicable to large-scale extended materials.

DOI: 10.1103/PhysRevB.109.035103

I. INTRODUCTION

Random phase approximation (RPA) [1–3] as formulated within the framework of adiabatic-correction fluctuation-dissipation theorem (ACFDT) [4,5] provides an appealing approach to compute the ground-state energy of interacting many-electron systems [6–9]. It can be viewed as a nonlocal approximation for the exchange-correlation (XC) energy functional within Kohn-Sham (KS) density functional theory (DFT) [10,11]. According to the Jacob's ladder classifying different XC functionals [12], the RPA sits on the top rung of the ladder, and captures seamlessly nonlocal many-electron correlations that are missing in lower-rung functionals. Applications of RPA to real materials show that this approach performs rather well in describing energy differences, in particular the surface adsorption energies [13,14], the reaction barrier heights [15,16], and the delicate energy differences between different polymorphs [17–21]. Despite its promising performance, a widespread use of RPA is hindered by its quickly increasing computational cost with system size. To deal with this issue, a considerable amount of recent works are devoted to developing low-scaling algorithms to speed up the RPA calculations [22–30], paving ways for applying RPA to large-scale, complex materials that are previously out of reach.

The key quantity in RPA calculations is the noninteracting KS density response function χ^0, represented within a suitable basis set. The standard computational scaling for evaluating χ^0 is $O(N^4)$ with N being a measure of system size, for both plane-wave basis sets and the resolution-of-identity (RI) formulation of RPA within atomic-orbital basis sets. The $O(N^4)$ scaling can be reduced to $O(N^3)$ by utilizing the space-time algorithm [31–33], initially developed for the *GW* method [34]. Thanks to the development of the minimax quadrature grid by Kaltak *et al.* [24,35], which enables an efficient discrete Fourier transform from the imaginary time domain to the imaginary frequency domain, the $O(N^3)$ algorithm becomes superior to the standard $O(N^4)$ one at a cross point of system size that can be handled by modern computers. Such a dual real-space and plane-wave formulation of $O(N^3)$ RPA (and analogously *GW* [36]) algorithm was soon extended to Gaussian atomic-orbital framework, combined with the RI technique of different flavors, such as the overlap-metric [26] and attenuated-Coulomb-metric [37] -based RI schemes, pair atomic density fitting (PADF) [38], and the interpolative separable RI scheme [29,30,39]. Benefited further from the spatial locality of atomic orbitals, algorithms and implementations with $O(N)$ to $O(N^3)$ scaling behaviors have been reported [26–30]. Apart from these, radically different approaches based on solving the Riccati

*helx@ustc.edu.cn
†renxg@iphy.ac.cn

equation using the local correlation method [25] as well as on a stochastic formulation of ACFDT-RPA via time-dependent DFT [22,40] have been developed, allowing for linear or even sublinear RPA correlation calculations. Practically, the plane-wave-based implementations are more suitable for describing periodic systems, whereas the atomic-orbital-based implementations are typically applied to finite molecular systems and/or supercell-based Γ-only simulations.

In this work, we present yet another low-scaling algorithm for periodic RPA calculations with finite **k**-point sampling using numerical atomic orbital (NAO) basis sets. In this algorithm, the computational cost for the key step of the RPA calculations, namely, the evaluation of the KS response function matrix, scales quadratically or better with respect to the number of atoms in the unit cell and linearly with the number of **k** points in the Brillouin zone. This is enabled by the localized resolution of identity (LRI) approximation [41], a prescreening of the sparse RI coefficients [42–44], and an efficient imaginary time-to-frequency Fourier transform using the minimax grid [24,35]. The $O(N^2)$ scaling can in fact be made asymptotically linear for insulating systems, if the spatial decay of the Green's function is further taken into account. The algorithm has been implemented in a stand-alone library package called LibRPA, which has been interfaced with two NAO-based first-principles codes FHI-aims [45] and ABACUS [46–48]. The development of LibRPA allows one to do efficient RPA calculations with NAO-based first-principles codes, with necessary inputs provided by the latter.

The LRI approximation, also known as PADF [49,50] or concentric atomic density fitting (CADF) [51,52] in the literature, is crucial for the design of low-scaling algorithms, has been used in periodic hybrid functional [42–44] and G_0W_0 calculations [53], as well as in RPA correlation energy [54] and RPA force calculations for molecules [55] before. Various benchmark calculations showed that this approximation is sufficiently accurate for Hartree-Fock exchange calculations, within the NAO basis set framework, for both molecules [41] and periodic systems [42–44]. The LRI errors are admittedly more pronounced for correlated methods that involve unoccupied states. However, our own experience suggests that LRI also provides adequate accuracy for correlated methods such as PRA and *GW*, for both molecules [41,55] and extended systems [53,56], provided that high-quality auxiliary basis sets can be constructed. Recently, Spadetto *et al.* reported a strategy that makes PADF-RPA work for large molecules by projecting out parts of the basis set giving rise to orbital products that are hard to describe by PADF. While the accuracy of LRI for NAO-based periodic RPA calculations will be benchmarked elsewhere, here we mainly focus on the low-scaling algorithm, the implementation details, and the scaling behavior with respect to both system size and the number of **k** points. Thanks to the existing canonical $O(N^4)$-scaling periodic RPA implementation in FHI-aims , the accuracy and efficacy of the low-scaling RPA implementation can be unambiguously benchmarked. We show that our present implementation can readily treat three-dimensional bulk systems containing over 1000 atoms, with reasonable computational resources.

The paper is organized as follows. The key equations behind the NAO-based low-scaling RPA algorithm are presented in Sec. II, which is followed by Sec. III, which contains a detailed discussion of the actual loop structure adopted in the low-scaling algorithm and the implementation details. Section IV presents the major results, consisting of test calculations that validate the algorithm and implementation by comparing to the existing **k**-space-based algorithm in FHI-aims, and benchmarks of the scaling behavior of the computational cost with respect to system size. In addition, we also discuss the importance of incorporating the sparsity of the Green's function in the algorithm, which brings significant further reduction of the computational cost. Finally, we report a scaling-behavior study for system sizes beyond 1000 atoms by interfacing LibRPA with another NAO-based DFT code, ABACUS. The Appendix presents a detailed derivation of the key equations in Sec. II, and the decay behavior of the Green's function in real space for prototypical systems.

II. THEORETICAL FORMULATION

In this section, we will present the key equations behind the low-scaling algorithm of periodic RPA within the NAO basis-set framework. The formalism should be applicable to Gaussian-type or other types of localized atomic orbitals as well, provided that high-quality auxiliary basis sets (ABSs) are available and the LRI is sufficiently accurate.

Within the ACFDT framework, the RPA correlation energy is formally given by [9]

$$E_c^{\mathrm{RPA}} = \frac{1}{2\pi}\int_0^\infty d\omega\, \mathrm{Tr}[\ln(1-\chi^0(i\omega)v)+\chi^0(i\omega)v], \quad (1)$$

where χ^0 represents the KS independent density-response function on the imaginary frequency, v the bare Coulomb potential, and $\mathrm{Tr}[AB] = \int d\mathbf{r}d\mathbf{r}' A(\mathbf{r},\mathbf{r}')B(\mathbf{r}',\mathbf{r})$. For a periodic system, the spatially nonlocal function $\chi^0(\mathbf{r},\mathbf{r}',i\omega)$ can be represented in terms of a set of Bloch-summed atom-centered auxiliary basis functions (ABFs),

$$\chi^0(\mathbf{r},\mathbf{r}',i\omega) = \frac{1}{N_{\mathbf{k}}}\sum_{\mu,\nu,\mathbf{q}} P_\mu^{\mathbf{q}}(\mathbf{r})\chi^0_{\mu\nu}(\mathbf{q},i\omega)P_\nu^{\mathbf{q}*}(\mathbf{r}'), \quad (2)$$

where the summation over $\mathbf{q}$ goes over the first Brillouin zone (BZ), and $N_{\mathbf{k}}$ is the number of **k** points in the first BZ. In Eq. (2),

$$P_\mu^{\mathbf{q}}(\mathbf{r}) = \sum_{\mathbf{R}} e^{i\mathbf{q}\cdot\mathbf{R}}P(\mathbf{r}-\boldsymbol{\tau}_{\mathcal{U}}-\mathbf{R}), \quad (3)$$

with $\mathcal{U}$ denoting the atom on which the ABF $P_\mu(\mathbf{r})$ is sitting and $\boldsymbol{\tau}_{\mathcal{U}}$ the position of atom $\mathcal{U}$ within the unit cell. Further computing the Coulomb matrix in reciprocal space,

$$\begin{aligned} V_{\mu\nu}(\mathbf{q}) &= \sum_{\mathbf{R}} e^{i\mathbf{q}\cdot\mathbf{R}}V_{\mu\nu}(\mathbf{R}) \\ &= \sum_{\mathbf{q}} e^{i\mathbf{q}\cdot\mathbf{R}}\iint \frac{P_\mu(\mathbf{r}-\boldsymbol{\tau}_{\mathcal{U}})P_\nu(\mathbf{r}'-\boldsymbol{\tau}_{\mathcal{V}}-\mathbf{R})}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}d\mathbf{r}', \end{aligned} \quad (4)$$

the RPA correlation energy can be evaluated using Eq. (1), where $\chi^0(i\omega)$ and v should be interpreted as their respective matrix forms represented in terms of the ABFs, as given by Eqs. (2) and (4). And in this case, $\mathrm{Tr}[AB]$ should be interpreted

as $\mathrm{Tr}[AB] = \frac{1}{N_\mathbf{k}} \sum_{\mu,\nu,\mathbf{q}} [A_{\mu,\nu}(\mathbf{q}) B_{\nu,\mu}(\mathbf{q})]$, with $N_\mathbf{k}$ being the number of $\mathbf{k}$ points in the Brillouin zone. What is described above is a well-defined formalism that yields reliable results, under the condition that the employed RI or LRI approximations are adequately accurate, and the singularity of the Coulomb matrix at $\mathbf{q} = 0$ is properly treated. This above $\mathbf{k}$-space-based formalism has been implemented in FHI-aims, and benchmark calculations have proven its numerical reliability. However, the bottleneck step in RPA calculations, i.e., the evaluation of $\chi^0_{\mu\nu}(\mathbf{q}, i\omega)$, scales quartically with the number of basis functions in the unit cell and quadratically with the number of $\mathbf{k}$ points, preventing its application to large systems.

To address this issue, here we reformulate the approach in real space, particularly taking advantage of the locality offered by NAO basis functions. As usual, we start with the real-space imaginary-time expression of χ^0, given by a simple product of the noninteracting Green's function G^0,

$$\chi^0(\mathbf{r}, \mathbf{r}', i\tau) = -iG^0(\mathbf{r}, \mathbf{r}', i\tau) G^0(\mathbf{r}', \mathbf{r}, -i\tau). \tag{5}$$

Within an AO basis framework, the KS wave functions in $\mathbf{k}$ space are given by

$$\psi_{n\mathbf{k}}(\mathbf{r}) = \sum_i \sum_\mathbf{R} e^{i\mathbf{k}\cdot\mathbf{R}} c_{i,n}(\mathbf{k}) \varphi_i(\mathbf{r} - \mathbf{R} - \boldsymbol{\tau}_I), \tag{6}$$

where $\varphi_i(\mathbf{r})$ is a NAO sitting on atom I (with $\boldsymbol{\tau}_I$ denoting its position within the unit cell), and $c_{i,n}(\mathbf{k})$ are KS eigenvectors. The noninteracting Green's function $G^0(i\tau)$ in the imaginary-time domain can be expanded in terms of the NAOs as

$$G^0(\mathbf{r}, \mathbf{r}', i\tau) = \sum_{i,j} \sum_{\mathbf{R}_1, \mathbf{R}_2} \varphi_i(\mathbf{r} - \mathbf{R}_1 - \boldsymbol{\tau}_I) \times G^0_{i,j}(\mathbf{R}_2 - \mathbf{R}_1, i\tau) \varphi_j(\mathbf{r}' - \mathbf{R}_2 - \boldsymbol{\tau}_J) \tag{7}$$

with

$$G^0_{i,j}(\mathbf{R}, i\tau) = \begin{cases} -i \frac{1}{N_\mathbf{k}} \sum_{n,\mathbf{k}} f_{n\mathbf{k}} c_{i,n}(\mathbf{k}) c^*_{j,n}(\mathbf{k}) e^{-i\mathbf{k}\cdot\mathbf{R}} e^{-(\epsilon_{n,\mathbf{k}} - \mu)\tau} & \tau \leqslant 0, \\ i \frac{1}{N_\mathbf{k}} \sum_{n,\mathbf{k}} (1 - f_{n\mathbf{k}}) c_{i,n}(\mathbf{k}) c^*_{j,n}(\mathbf{k}) e^{-i\mathbf{k}\cdot\mathbf{R}} e^{-(\epsilon_{n,\mathbf{k}} - \mu)\tau} & \tau > 0. \end{cases} \tag{8}$$

Here, $G^0_{i,j}(\mathbf{R}, i\tau)$ is the matrix form of $G^0(i\tau)$ represented in terms of NAOs, μ is the chemical potential, and $\epsilon_{n,\mathbf{k}}$ and $f_{n\mathbf{k}}$ are KS orbital energies and occupation factors. Here, for simplicity, we assume $f_{n\mathbf{k}}$ equals 1 for occupied states and 0 for unoccupied ones. The situation of fractional occupations is more involved and will be discussed separately.

Plugging Eq. (7) into Eq. (5), one has

$$\chi^0(\mathbf{r}, \mathbf{r}', i\tau) = -i \sum_{i,j,k,l} \sum_{\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4} \varphi_i(\mathbf{r} - \mathbf{R}_1 - \boldsymbol{\tau}_I) \varphi_k(\mathbf{r} - \mathbf{R}_3 - \boldsymbol{\tau}_K) G^0_{i,j}(\mathbf{R}_2 - \mathbf{R}_1, i\tau) \times G^0_{l,k}(\mathbf{R}_3 - \mathbf{R}_4, -i\tau) \varphi_j(\mathbf{r}' - \mathbf{R}_2 - \boldsymbol{\tau}_J) \varphi_l(\mathbf{r}' - \mathbf{R}_4 - \boldsymbol{\tau}_L), \tag{9}$$

where $\boldsymbol{\tau}_K$ and $\boldsymbol{\tau}_L$ denote the positions of the atom K and L, on which the basis function $\varphi_k(\mathbf{r})$ and $\varphi_l(\mathbf{r})$ are sitting, respectively. The key idea here is to derive a more compact representation of $\chi^0(\mathbf{r}, \mathbf{r}', i\tau)$ in terms of the ABFs, i.e.,

$$\chi^0(\mathbf{r}, \mathbf{r}', i\tau) = \sum_{\mu \in \mathcal{U}, \nu \in \mathcal{V}} \sum_{\mathbf{R}_1, \mathbf{R}_2} P_\mu(\mathbf{r} - \mathbf{R}_1 - \boldsymbol{\tau}_\mathcal{U}) \chi^0_{\mu,\nu}(\mathbf{R}_2 - \mathbf{R}_1, i\tau) P_\nu(\mathbf{r}' - \mathbf{R}_2 - \boldsymbol{\tau}_\mathcal{V}) \tag{10}$$

with $\mathcal{U}$ and $\mathcal{V}$ denoting the atoms on which the ABFs $P_\mu(\mathbf{r})$ and $P_\nu(\mathbf{r})$ are sitting, and $\boldsymbol{\tau}_\mathcal{U}$ and $\boldsymbol{\tau}_\mathcal{V}$ their respective atomic positions within the unit cell. To this end, we apply the LRI approximation here, which in essence expands the product of two NAOs in terms of the ABFs sitting on the two atoms on which the two NAOs are centering, i.e.,

$$\begin{aligned} \varphi_i(\mathbf{r} - \mathbf{R}_1 - \boldsymbol{\tau}_I) \varphi_k(\mathbf{r} - \mathbf{R}_3 - \boldsymbol{\tau}_K) &\approx \sum_{\mu \in I} C^{\mu(\mathbf{R}_1)}_{i(\mathbf{R}_1),k(\mathbf{R}_3)} P_\mu(\mathbf{r} - \mathbf{R}_1 - \boldsymbol{\tau}_I) + \sum_{\mu \in K} C^{\mu(\mathbf{R}_3)}_{i(\mathbf{R}_1),k(\mathbf{R}_3)} P_\mu(\mathbf{r} - \mathbf{R}_3 - \boldsymbol{\tau}_K) \\ &= \sum_{\mu \in I} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_3 - \mathbf{R}_1)} P_\mu(\mathbf{r} - \mathbf{R}_1 - \boldsymbol{\tau}_I) + \sum_{\mu \in K} C^{\mu(\mathbf{0})}_{i(\mathbf{R}_1 - \mathbf{R}_3),k(\mathbf{0})} P_\mu(\mathbf{r} - \mathbf{R}_3 - \boldsymbol{\tau}_K). \end{aligned} \tag{11}$$

Here we follow the notation adopted in Ref. [53], whereby $\tilde{C}^{\mu(\mathbf{R}_1)}_{i(\mathbf{R}_1),k(\mathbf{R}_3)}$ denote the two-center expansion coefficients with the lattice vector in parentheses indicating the unit cell to which the basis function belongs. Furthermore, $\mu \in I$ ($\mu \in K$) in Eq. (11) signifies that the summation over the ABFs is restricted to those centering at the atom I (K). The second equation of Eq. (11) follows from the translational symmetry of the periodic system, which requires that $C^{\mu(\mathbf{R}_1)}_{i(\mathbf{R}_1),k(\mathbf{R}_3)} = C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_3 - \mathbf{R}_1)}$, with $\mathbf{0}$ here denoting the unit cell at the origin. This implies that the expansion coefficients only depend on one independent lattice vector.

Now, by equalizing Eq. (9) with Eq. (10), and utilizing Eq. (11), it is somewhat lengthy but otherwise straightforward to show that the matrix form of $\chi^0(i\tau)$ in real space is given as follows,

$$\begin{aligned} \chi^0_{\mu,\nu}(\mathbf{R}, i\tau) &= -i \left[\sum_{i \in \mathcal{U}} \sum_{k \in K, \mathbf{R}_1} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)} \left(M^\nu_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau) + M^{\nu*}_{i,k}(\mathbf{R}_1, \mathbf{R}, -i\tau) + Z^\nu_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau) + Z^{\nu*}_{i,k}(\mathbf{R}_1, \mathbf{R}, -i\tau) \right) \right] \\ &= -i \sum_{i \in \mathcal{U}} \sum_{k \in K, \mathbf{R}_1} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)} O^\nu_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau), \end{aligned} \tag{12}$$

where

$$O^{\nu}_{i,k}(\mathbf{R}_1,\mathbf{R},i\tau) = M^{\nu}_{i,k}(\mathbf{R}_1,\mathbf{R},i\tau) + M^{\nu*}_{i,k}(\mathbf{R}_1,\mathbf{R},-i\tau) + Z^{\nu}_{i,k}(\mathbf{R}_1,\mathbf{R},i\tau) + Z^{\nu*}_{i,k}(\mathbf{R}_1,\mathbf{R},-i\tau) \tag{13}$$

and

$$M^{\nu}_{i,k}(\mathbf{R}_1,\mathbf{R},i\tau) = \sum_{j\in\mathcal{V}} G_{i,j}(\mathbf{R},i\tau) N^{\nu}_{j,k}(\mathbf{R}_1,\mathbf{R},i\tau)$$

$$Z^{\nu}_{i,k}(\mathbf{R}_1,\mathbf{R},i\tau) = \sum_{j\in\mathcal{V}} G_{j,k}(\mathbf{R}_1-\mathbf{R},-i\tau) X^{\nu}_{i,j}(\mathbf{R},i\tau) \tag{14}$$

with the intermediate quantities $N^{\nu}_{j,k}(\mathbf{R}_1,\mathbf{R},i\tau)$ and $X^{\nu}_{i,j}(\mathbf{R},i\tau)$ defined as

$$N^{\nu}_{j,k}(\mathbf{R}_1,\mathbf{R},i\tau) = \sum_{l\in L,\mathbf{R}_2} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{l,k}(\mathbf{R}_1-\mathbf{R}_2,-i\tau)$$

$$X^{\nu}_{i,j}(\mathbf{R},i\tau) = \sum_{l\in L,\mathbf{R}_2} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{i,l}(\mathbf{R}_2,i\tau). \tag{15}$$

In deriving the above equations, symmetry properties of the expansion coefficients and index swapping have been used. Details of the derivations are presented in the Appendix A. Equations (12)–(15) are the key underlying equations on which the low-scaling algorithm is based, which will be discussed in the next section.

So far, we have constructed the response function matrix in terms of ABFs in the real-space imaginary-time domain. To compute the RPA correlation energy [Eq. (1)], it is more convenient to work in the $\mathbf{k}$ space and imaginary frequency domain. To this end, Fourier transforms from the real to reciprocal spaces, and from the imaginary time to imaginary frequency domains are sequentially performed for the response function matrix. Considering the symmetry property of χ^0 in time and frequency, i.e., $\chi^0(\mathbf{r},\mathbf{r}',i\omega) = \chi(\mathbf{r}',\mathbf{r},-i\omega)$, $\chi^0(\mathbf{r},\mathbf{r}',i\tau) = \chi^0(\mathbf{r}',\mathbf{r},-i\tau)$, and $\chi^0(\mathbf{R},i\tau) = \chi^0(-\mathbf{R},i\tau)$, the time-to-frequency Fourier transform between the complex axes is simplified to a cosine transformation including an additional factor of $-i$ [24,33,35]

$$\chi^0_{\mu,\nu}(\mathbf{R},i\omega_k) = -i\sum_{j=1}^{N} \gamma_{jk}\chi^0_{\mu,\nu}(\mathbf{R},i\tau_j)\cos(\tau_j\omega_k). \tag{16}$$

Here we adopt the nonuniform imaginary-time $\{i\tau_j\}_{j=1}^{N_\tau}$ and frequency $\{i\omega_k\}_{k=1}^{N_\omega}$ minimax grids from CP2K [57], which have been proven to be accurate [37]. The coefficients γ_{jk} are determined using L^2 minimization [24] during program run. Once the real-space imaginary-frequency χ^0 matrix is obtained from Eq. (16), it is further transformed to the reciprocal space straightforwardly,

$$\chi^0_{\mu,\nu}(\mathbf{q},i\omega) = \sum_{\mathbf{R}} e^{i\mathbf{q}\cdot\mathbf{R}}\chi^0_{\mu,\nu}(\mathbf{R},i\omega). \tag{17}$$

To facilitate the computation of RPA correlation energy, we further introduce an intermediate quantity, i.e., the product of χ^0 and V matrices,

$$\Pi(\mathbf{k},i\omega) = \chi^0(\mathbf{k},i\omega)V(\mathbf{k}). \tag{18}$$

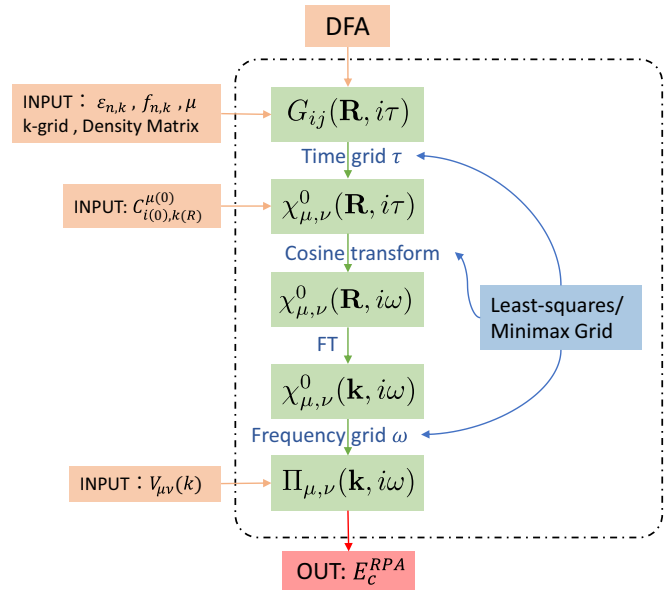


FIG. 1. Major steps in the computation of the RPA correlation energy in the present algorithm: (i) Calculate the noninteracting Green's function within the NAO basis set based on a preceding DFA calculation; (ii) contract two Green's functions to construct $\chi^0_{\mu,\nu}(\mathbf{R},i\tau)$ using LRI; (iii) perform the cosine transformation to obtain $\chi^0_{\mu,\nu}(\mathbf{R},i\omega)$ according to Eq. (16); (iv) Fourier transform (FT) $\chi^0_{\mu,\nu}(\mathbf{R},i\omega)$ to $\chi^0_{\mu,\nu}(\mathbf{k},i\omega)$; v) calculate $\Pi(\mathbf{k},i\omega)$ according to Eq. (18); (vi) calculate E_c^{RPA} via an integration over imaginary frequencies and summation over BZ [Eq. (19)].

The RPA correlation energy for periodic systems per unit cell can be finally obtained as

$$E_c^{\mathrm{RPA}} = \frac{1}{2\pi}\frac{1}{N_{\mathbf{k}}}\sum_{\mathbf{q}}\int_0^{\infty} d\omega\,\mathrm{Tr}\{\ln[1-\Pi(\mathbf{q},i\omega)] + \Pi(\mathbf{q},i\omega)\}$$
$$= \frac{1}{2\pi}\frac{1}{N_{\mathbf{k}}}\sum_{\mathbf{q}}\int_0^{\infty} d\omega\,\ln\{\det[1-\Pi(\mathbf{q},i\omega)]\} + \mathrm{Tr}[\Pi(\mathbf{q},i\omega)], \tag{19}$$

where the property $\mathrm{Tr}[\ln(A)] = \ln[\det(A)]$ is used.

III. IMPLEMENTATION DETAILS

The major steps of computing E_c^{RPA} within the NAO basis sets are illustrated in Fig. 1. After a preceding self-consistent KS calculation with a lower-rung density functional approximation (DFA), one obtains the KS orbitals and orbital energies. The ABFs have been generated beforehand, and so do the LRI coefficients $C^{\mu(\mathbf{0})}_{i(\mathbf{0}),j(\mathbf{R})}$ [Eq. (11)] and the Coulomb matrix $V_{\mu\nu}(\mathbf{k})$ [Eq. (4)]. These quantities have been available previously and used in periodic hybrid functional [43,44] and G_0W_0 calculations [53]. The implementation in the present work begins with the evaluation of real-space imaginary-time independent-particle Green's function $G^0(\mathbf{R},i\tau)$ using eigenvalues and eigenvectors generated using a NAO-based DFT code [cf. Eq. (8)]. After calculating and storing the $G^0_{ij}(\mathbf{R},i\tau)$, the real-space imaginary-time response function

Algorithm 1. Loop structure of evaluating $\chi^0_{\mu\nu}(\mathbf{R}, i\tau)$. $\langle \mathcal{U}(\mathbf{0}), \mathcal{V}(\mathbf{R})\rangle$ denotes an atomic pair with atom $\mathcal{U}$ in the unit cell at origin and atom $\mathcal{V}$ in the unit cell $\mathbf{R}$. The symbol $\mathcal{N}[\mathcal{U}]$ represents the set of neighboring atoms of atom $\mathcal{U}$, and $K(\mathbf{R}_1) \in \mathcal{N}[\mathcal{U}(\mathbf{0})]$ means that the atom K in unit cell $\mathbf{R}_1$ is in the neighborhood of the atom $\mathcal{U}$ in the unit cell at the origin.

for all τ **do**
  for all R do
    for all $\langle \mathcal{U}(\mathbf{0}), \mathcal{V}(\mathbf{R})\rangle$ **do**
      for all $L(\mathbf{R}_2) \in \mathcal{N}[\mathcal{V}(\mathbf{R})]$ **do**
        Calculate $X^{\nu}_{i,j}(\mathbf{R}_1, \mathbf{R}, i\tau)$ [cf. Eq. (15)]
      end for
      for all $K(\mathbf{R}_1) \in \mathcal{N}[\mathcal{U}(\mathbf{0})]$ **do**
        for all $L(\mathbf{R}_2) \in \mathcal{N}[\mathcal{V}(\mathbf{R})]$ **do**
          Calculate $N^{\nu}_{j,k}(\mathbf{R}_1, \mathbf{R}, i\tau)$ [cf. Eq. (15)]
        end for
        Calculate $M^{\nu}_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau), Z^{\nu}_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau)$ [cf. Eq. (14)]
        Calculate $O^{\nu}_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau)$ [cf. Eq. (13)]
        Calculate $\chi^0_{\mu,\nu}(\mathbf{R}, i\tau) \mathrel{+}= C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)} O^{\nu}_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau)$ [cf. Eq. (12)]
      end for
    end for
  end for
end for

matrix $\chi^0_{\mu\nu}(\mathbf{R}, i\tau)$ can be evaluated, which is usually the rate-determining step throughout the whole RPA computation. The inputs needed for this step are the Green's function matrix $G^0_{ij}(\mathbf{R}, i\tau)$ and the LRI coefficients $C^{\mu(\mathbf{0})}_{i(\mathbf{0}),j(\mathbf{R})}$. After $\chi^0_{\mu\nu}(\mathbf{R}, i\tau)$ is obtained, it is relatively straightforward to convert it to $\chi^0_{\mu\nu}(\mathbf{k}, i\omega)$ via the cosine transform and Fourier transform successively. The cosine transform benefits from the recently developed efficient minimax quadrature grids [24,26,36]. With $\chi^0_{\mu\nu}(\mathbf{k}, i\omega)$, one can multiply it with the Coulomb matrix $V_{\mu,\nu}(\mathbf{k})$ to obtain $\Pi_{\mu,\nu}(\mathbf{k}, i\omega)$, and finally compute the RPA correlation energy via Eq. (19).

The essential point of the present work is to reduce the computational scaling of evaluating $\chi^0_{\mu,\nu}(\mathbf{R}, i\tau)$. Algorithm 1 illustrates the loop structure of computing $\chi^0_{\mu,\nu}(\mathbf{R}, i\tau)$ based on Eqs. (12)–(15). The outermost loop goes over all time grid points $\{\tau_j\}_{j=1}^{N_\tau}$, under which one further goes through all lattice vectors $\{\mathbf{R}\}$ within the BvK supercell. For each $(\tau, \mathbf{R})$ pair, the whole $\chi^0_{\mu,\nu}(\mathbf{R}, i\tau)$ matrix is decomposed into blocks associated with individual atomic pairs $\langle \mathcal{U}(\mathbf{0}), \mathcal{V}(\mathbf{R})\rangle$ on which the ABFs P_μ and P_ν are located, respectively. Computing these blocks $\chi^0_{\mu\in\mathcal{U},\nu\in\mathcal{V}}$ separately for each atomic pair $\langle \mathcal{U}(\mathbf{0}), \mathcal{V}(\mathbf{R})\rangle$ and assembling them up, one obtains the entire $\chi^0_{\mu,\nu}$ matrix. Obviously, for a given lattice vector $\mathbf{R}$, the number of such atomic pairs scales as N^2_{at} where N_{at} is the number of atoms in a unit cell.

Now, inside the loop over the atomic pair $\langle \mathcal{U}(\mathbf{0}), \mathcal{V}(\mathbf{R})\rangle$, one still needs to go through atom K in the unit cell of $\mathbf{R}_1$, and atom L in the unit cell of $\mathbf{R}_2$, in order to compute intermediate quantities such as $N^{\nu}_{j,k}(\mathbf{R}_1, \mathbf{R}, i\tau)$ and $X^{\nu}_{i,j}(\mathbf{R}, i\tau)$ as defined in Eq. (15), and finally $\chi^0_{\mu\in\mathcal{U},\nu\in\mathcal{V}}$. The key point here is that the atom $K(\mathbf{R}_1)$ has to be the neighboring atom of the atom $\mathcal{U}(\mathbf{0})$, and $L(\mathbf{R}_2)$ has to be the neighboring atom of $\mathcal{V}(\mathbf{R})$. Outside the neighborhood region, the LRI expansion coefficients will be zero (or insignificantly small) and the K, L atoms there will not contribute. For a finite periodic system, the number of neighboring atoms of a given reference atom is determined by the spatial range (cutoff radii) of NAO basis functions, and does not keep increasing with size and complexity of the unit cell. This means that, in our algorithm, the computational cost required for a block of response function matrix associated with an atomic pair $\langle \mathcal{U}(\mathbf{0}), \mathcal{V}(\mathbf{R})\rangle$ approaches a constant as the size of the system (unit cell) grows. Thus the entire computational cost scales as $N^2_{\text{at}} N_{\mathbf{R}} N_\tau$ or $N^2_{\text{at}} N_{\mathbf{k}} N_\tau$ where $N_{\mathbf{R}}$ is the number of unit cells in the BvK supercell, usually set equal to $N_{\mathbf{k}}$, and N_τ is the number of imaginary time grid points. Note that in practical converged calculations, the size of unit cells (i.e., N_{at}) is not independent of $N_{\mathbf{R}}$ or $N_{\mathbf{k}}$; large unit cells usually mean that one can use fewer $\mathbf{k}$ points or even a single Γ point in the calculations. Thus, $N_{\text{at}} \times N_{\mathbf{k}}$ can be roughly considered as a constant for a given type of system. Furthermore, N_τ depends on the ratio of the smallest to the largest single-particle transitions in the system, primarily determined by the band gap. Thus, the above described algorithm is *de facto* quadratic scaling for evaluating the response function matrix.

In the above analysis of the scaling behavior of Algorithm 1, the sparsity of the Green's function matrix $G^0_{ij}(\mathbf{R}, i\tau)$ itself was not taken into account. In fact, $G^0_{ij}(\mathbf{R}, i\tau)$ at $\tau \to 0^-$ corresponds to the reduced one-electron density matrix, which is known to decay exponentially for insulating systems, and polynomially for metallic systems [58–60], as the distance $|\mathbf{R} + \boldsymbol{\tau}_j - \boldsymbol{\tau}_i|$ between the centers of atomic orbitals i and j gets large. Exploiting this property, one can envision that the number of relevant atomic pairs $\langle \mathcal{U}(\mathbf{0}), \mathcal{V}(\mathbf{R})\rangle$ does not grow quadratically with respect to the unit cell size any longer, but rather linearly. This suggests that a refined algorithm that accounts for the sparsity of the Green's function $G^0_{ij}(\mathbf{R}, i\tau)$ will become asymptotically linear scaling. In practical implementation of this concept, one can introduce a screening threshold η_G, whereby, if the maximal element of the Green's function matrix associated with an atomic pair $\langle \mathcal{U}(\mathbf{0}), \mathcal{V}(\mathbf{R})\rangle$ is smaller than η_G, i.e.,

$$\max\{|G_{i\in\mathcal{U}(\mathbf{0}),j\in\mathcal{V}(\mathbf{R})}(\mathbf{R}, i\tau)|\} < \eta_G, \tag{20}$$

then this atomic pair will be discarded in the evaluation of the χ^0 matrix. Algorithm 2 illustrates the basic idea behind this refined scheme, leading to an asymptotically linear-scaling algorithm for evaluating $\chi^0(\mathbf{R}, i\tau)$. In Sec. IV C we will demonstrate the effect of filtering out the atomic pairs with zero or sufficiently small Green's function matrix elements.

The above-described algorithm for low-scaling RPA correlation energy calculations has been implemented in a stand-alone library called LibRPA, which is currently accessible from GitHub [61]. So far, LibRPA has been interfaced with two NAO-based first-principles code packages, the all-electron FHI-aims code [45] and the pseudopotential-based ABACUS code [47]. Interfacing with other DFT codes that employ NAOs should be straightforward, if the necessary inputs as shown in Fig. 1 can be provided.

Algorithm 2. Refined algorithm for evaluating $\chi^0_{\mu\nu}(\mathbf{R}, i\tau)$ whereby the Green's-function-based screening is incorporated. This will reduce the number of loops in Algorithm 1 for $\langle \mathcal{U}(\mathbf{0}), \mathcal{V}(\mathbf{R}) \rangle$, as well as the matrix multiplications inside the loop, to varying degrees.

for all τ **do**
  for all R do
    for all $\langle \mathcal{U}(\mathbf{0}), \mathcal{V}(\mathbf{R}) \rangle$ **do**
      for all $L(\mathbf{R}_2) \in \mathcal{N}[\mathcal{V}(\mathbf{R})]$ **do**
        if $\max\{|G_{i\in\mathcal{U}(\mathbf{0}),l\in L(\mathbf{R}_2)}(\mathbf{R}_2, i\tau)|\} > \eta_G$ **then**
          Calculate $X^v_{i,j}(\mathbf{R}_1, \mathbf{R}, i\tau)$ [cf. Eq. (15)]
        end if
      end for
      for all $K(\mathbf{R}_1) \in \mathcal{N}[\mathcal{U}(\mathbf{0})]$ **do**
        if $\max\{|G_{i\in\mathcal{U}(\mathbf{0}),j\in\mathcal{V}(\mathbf{R})}(\mathbf{R}, i\tau)|\} > \eta_G$ **then**
          for all $L(\mathbf{R}_2) \in \mathcal{N}[\mathcal{V}(\mathbf{R})]$ **do**
            if $\max\{|G_{l\in L(\mathbf{R}_2),k\in K(\mathbf{R}_1)}(\mathbf{R}_1 - \mathbf{R}_2, -i\tau)|\} > \eta_G$ **then**
              Calculate $N^v_{j,k}(\mathbf{R}_1, \mathbf{R}, i\tau)$ [cf. Eq. (15)]
            end if
          end for
          Calculate $M^v_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau)$ [cf. Eq. (14)]
        end if
        if $\max\{|G_{j\in\mathcal{V}(\mathbf{R}),k\in K(\mathbf{R}_1)}(\mathbf{R}_1 - \mathbf{R}, -i\tau)|\} > \eta_G$ **then**
          Calculate $Z^v_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau)$ [cf. Eq. (14)]
        end if
        Calculate $O^v_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau)$ [cf. Eq. (13)]
        Calculate $\chi^0_{\mu,\nu}(\mathbf{R}, i\tau) += C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)} O^v_{i,k}(\mathbf{R}_1, \mathbf{R}, i\tau)$ [cf. Eq. (12)]
      end for
    end for
  end for
end for

IV. RESULTS

In this section, we set out to benchmark the performance of the low-scaling RPA algorithm and implementation as described in previous sections for selected insulating and semiconducting systems. Both the numerical accuracy and the efficiency of the implementation will be examined here. Regarding the efficiency, we will particularly check the practical scaling behavior of the computational cost with respect to the system size.

A. Accuracy of RPA correlation energy

We first examine the numerical accuracy of our low-scaling algorithm. To this end, we compare the RPA correlation energies as calculated by LibRPA with those produced by the conventional **k**-space implementation in FHI-aims. The conventional implementation is also based on LRI, but the key operations are performed in **k** space, without exploiting the sparsity of the LRI coefficients and the Green's function. This leads to a $O(N^4)$ scaling for calculating the response function matrix $\chi^0(\mathbf{k}, i\omega)$. The algorithm and implementation details follow closely the periodic G_0W_0 implementation as described in Ref. [53]. Production calculations based on such a conventional implementation have been reported in Refs. [21,56].

Table I presents the RPA correlation energies of several semiconductors, as obtained using the real-space, imaginary-time algorithm as implemented in LibRPA, in comparison with those obtained using the conventional **k**-space algorithm as implemented in FHI-aims. The same computational settings (basis sets, **k** grid, RI, and frozen-core approximations) are used in both FHI-aims and LibRPA calculations. Sufficiently many imaginary frequency points (and imaginary time points in case of LibRPA) are used in both types of calculations. Table I indicates that the LibRPA implementation produces nearly identical results as the conventional **k**-space implementation in FHI-aims. The difference in total RPA correlation calculations for all tested systems are below 0.1 meV. This holds for both the FHI-aims-2009 [45] (*tight* setting) and the localized variant of the NAO-VCC-*n*Z [62] (denoted as loc-NAO-VCC-*n*Z) basis sets. Compared to the original NAO-VCC-*n*Z basis sets, within loc-NAO-VCC-*n*Z, the so-called enhanced minimal basis is removed. Furthermore, the basis functions are tightened and reoptimized, making them more suitable for periodic, extended materials. Thus, the real-space, low-scaling algorithm and the reciprocal-space algorithm yield nearly identical results, demonstrating the consistency between the real-space and k-space methods. The

TABLE I. RPA correlation energies for several semiconductors as calculated by the real-space low-scaling algorithm as implemented in LibRPA and by the **k**-space algorithm as implemented in FHI-aims. The FHI-aims *tight* NAO basis sets are used for all semiconductors. For for some of the systems (Si, BN, and MgO), the results obtained using loc-NAO-VCC-3Z basis sets are also presented. A 4×4×4 **k** grid is adopted for all calculations. For FHI-aims calculations, a modified Gauss-Legendre frequency quadrature grid with 80 points is used, and for LibRPA calculations, minimax grids with 18 points for both time and frequency are used. Frozen-core approximation is used for all calculations.

	Basis set	FHI-aims (eV)	LibRPA (eV)	Difference (meV)
Si	*tight*	−15.836307	−15.836364	0.0574
	loc-NAO-VCC-3Z	−18.321370	−18.321399	0.0291
BN	*tight*	−27.816262	−27.816345	0.0831
	loc-NAO-VCC-3Z	−29.428039	−29.428047	0.0085
MgO	*tight*	−11.55797	−11.557899	−0.0704
	loc-NAO-VCC-3Z	−9.991418	−9.991505	0.0877
SiC	*tight*	−64.421342	−64.421246	−0.0955
GaAs	*tight*	−17.562131	−17.562132	0.0007
ZnO	*tight*	−50.186035	−50.18603	−0.0059

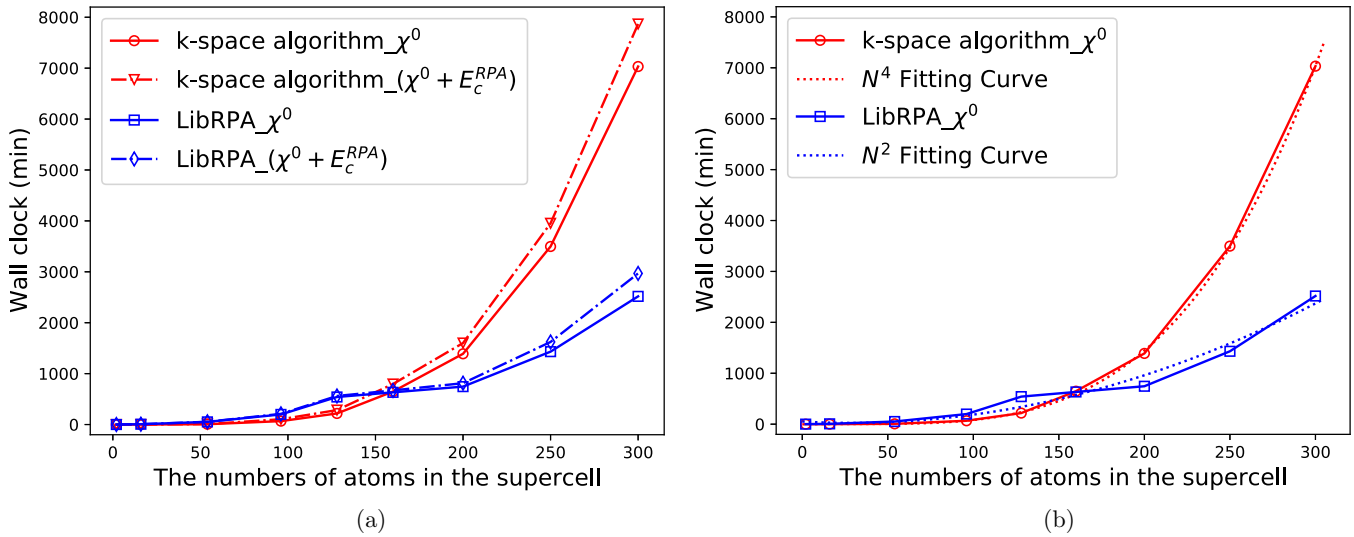


FIG. 2. Scaling behavior of the computation time as a function of system size (number of atoms in the supercell) for both the real-space low-scaling algorithm as implemented in LibRPA and the conventional **k**-space algorithm as implemented in FHI-aims. The test system is C diamond with increasing unit cell sizes. The loc-NAO-VCC-3Z basis set and a single (Γ-only) **k** point is used in the calculations. For FHI-aims calculations, a modified Gauss-Legendre frequency quadrature grid with 40 points are used, whereas for LibRPA calculations, minimax grids with 12 points for both time and frequency are used. The vertical axis represents the time measured or converted to the usage of a compute node with 64 CPU cores (for large systems more than one compute node is needed to run the calculations, and in these cases the reported timing is rescaled as if the calculations were done on one node). Left: the timings for both evaluating χ^0 matrix (solid lines) and the total RPA calculation $\chi^0 + E_c^{\mathrm{RPA}}$ (dash-dotted curves) are presented. Right: $O(N^4)$- and $O(N^2)$-scaling curves (dotted lines) are added by fitting to the data of the conventional and low-scaling algorithms for evaluating the χ^0 matrix, respectively.

remarkably high numerical precision also validates the correctness of the proposed low-scaling RPA algorithm and the actual implementation carried out in LibRPA.

B. Scaling behavior of the real-space algorithm

With the validity of the algorithm and the correctness of the implementation being established, we now check the actual scaling behavior of our implementation with respect to system size. Specifically, we carried out RPA calculations for carbon diamond crystals with increasing supercell size. In Fig. 2, the computational timings of the low-scaling algorithm as implemented in LibRPA (blue curves) and the conventional **k**-space algorithm (red curves) as implemented in FHI-aims are presented as a function of the supercell size (number of atoms). In the left panel of Fig. 2, both the timings for constructing the response function matrix χ^0 (solid lines), and the total computation times including, in addition to the construction of χ^0 matrix, the rest of calculations all the way up to the final evaluation of E_c^{RPA} (dash-dotted lines), are presented. The settings of the computational parameters are chosen such that the two series of calculations yield nearly identical RPA correlation energies for the same system. In the right panel of Fig. 2, we only presented the timings for evaluating the χ^0 matrix, but added the $O(N^4)$ and $O(N^2)$ fitting curves (dotted) for the computational times.

As expected, the computational cost for the **k**-space implementation for evaluating the χ^0 matrix shows a roughly $O(N^4)$ scaling behavior with respect to system size N (here N being the number of atoms in the supercell). In contrast, the real-space implementation in LibRPA shows a significantly reduced scaling behavior, but with a larger prefactor. The crossing point occurs at system size of about 160 C atoms, and after that the low-scaling algorithm starts to gain supremacy. In the benchmark tests presented in Fig. 2, the Green's-function-based screening was not turned on, and thus the computational cost should ideally follow a $O(N^2)$ scaling behavior as described in Algorithm 1. However, due to the fact that we have to increase the compute nodes for the larger systems and the complication arising from parallel efficiency, some of the data points deviate from the ideal $O(N^2)$ behavior. Nevertheless, an overall $O(N^2)$ scaling behavior is observable. For benchmark calculations presented in Fig. 2, systems comprising several dozen atoms can be computed using a single node. However, as the system size increases, the memory consumption becomes a limiting factor, and the demand for additional nodes grows accordingly. For 300-atom systems, we employed 24 nodes. Cross-node parallelization necessitates extra communication time and memory consumption, but our practical tests indicate that the parallel efficiency remains rather satisfactory.

Furthermore, from Fig. 2, one can see that for system size below 300 atoms, the computational cost of evaluating the response function matrix dominates. The rest of the calculations for evaluating the RPA correlation energy, though involving $O(N^3)$ matrix multiplication and Cholesky decomposition (for computing the determinant of $1 - \chi^0 v$), consumes only a small fraction of the total computation time.

In the above test, only a single **k** point (i.e., the Γ point) is used in the BZ sampling. Such a computational setting is suitable for describing systems with large supercells and low symmetries. Next, we check the scaling behavior of the new algorithm with respect to the number of **k** points in the BZ,

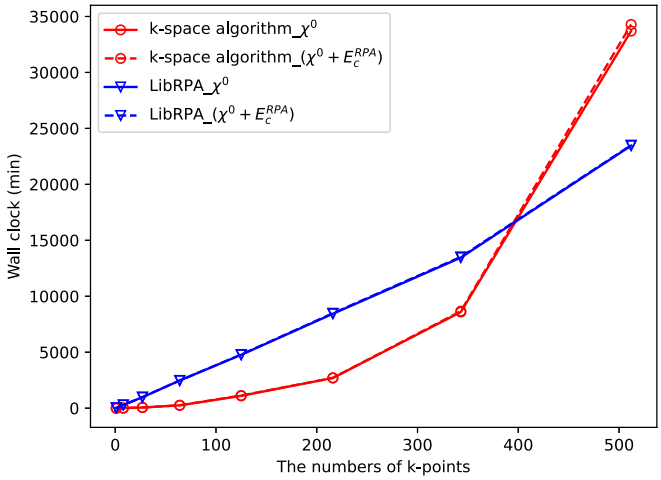


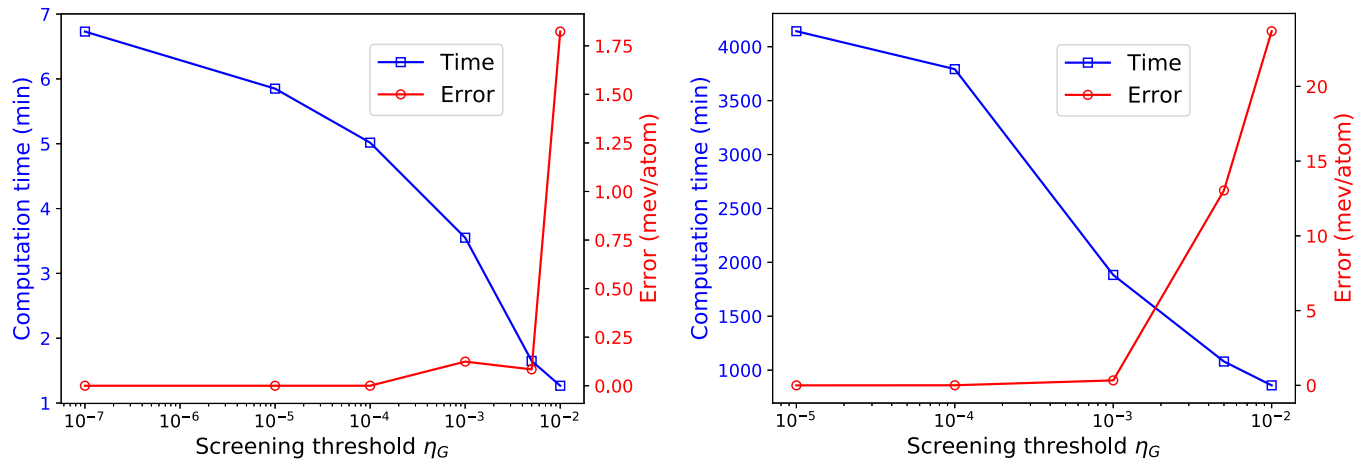
FIG. 3. Scaling behavior of the computation times as a function of number of the **k** points in the BZ sampling for both the real-space low-scaling algorithm as implemented in LibRPA and the conventional **k**-space algorithm as implemented in FHI-aims. The test system is C diamond with a fixed conventional unit cell containing eight atoms. All other computational settings are the same as Fig. 2.

with a fixed unit cell size. In Fig. 3, the computational times are presented as a function of the number of **k** points, for both the low-scaling algorithm and conventional **k**-space algorithm. The chosen system in this test calculation is again the C diamond, albeit with a fixed conventional cell (8 C atoms). Figure 3 shows that the computational cost of the real-space algorithm scales linearly with the number of **k** points, whereas the conventional **k**-space algorithm scales quadratically, as expected. The crossing point occurs in between the $7\times7\times7$ and $8\times8\times8$ **k** meshes. This suggests that the low-scaling algorithm has an advantage only when very dense **k** grid is needed. For practical periodic RPA calculations for simple solids with small unit cells, the conventional **k**-space algorithm is still the preferred method of choice.

C. Green's-function-based screening

In the benchmark tests of the scaling behavior presented in Sec. IV B, the sparsity of the Green's function matrix $G_{ij}(\mathbf{R}, i\tau)$ is not considered. As such, the low-scaling algorithm in theory scales quadratically with respect to system size and linearly with respect to the number of **k** points. If the sparsity of the Green's function is further taken into account, as discussed in Algorithm 2, one should achieve an asymptotically linear-scaling behavior with respect to the system size. In this section, we check how much error in the RPA correlation energy may be incurred if a thresholding parameter of the Green's-function matrix elements is introduced. From this investigation, one may be able to identify a safe parameter value that can be used in practical calculations, and find out what additional speedup one can gain if the Green's-function-based screening is invoked.

Figure 4 shows the computational time (left y axis) and the error in the computed RPA correlation energy (right y axis) as a function of the Green's-function screening parameter η_G (introduced in Algorithm 2). The test systems chosen here are the Ar crystal of $6\times6\times6$ supercell (containing 216 Ar atoms) and C diamond crystal of $8\times8\times8$ supercell (containing 1024 C atoms) with only a single (Γ-only) **k** point. From Fig. 4, one can see that, for such insulating systems, a screening parameter of 10^{-4}–10^{-3} can lead to more than a factor of two reduction of the computational time, yet the incurred error is kept at meV/atom level for the actual RPA calculations. Figure 7 in Appendix B further shows how the computational times and the incurred errors vary with the screening parameter η_G for different supercell size of C diamond system. One can see that a substantial reduction of the computational times is achieved for systems containing several hundreds of atoms. The reduction will be more drastic for even bigger systems. We thus anticipate that the refined low-scaling algorithm that incorporates Green's-function screening will bring significant additional savings, in particular for wide-gap insulators where the Green's function is expected to quickly decay in real space.



(a) Ar crystal (b) C diamond crystal

FIG. 4. Computation times and errors as a function of the threshold of the Green's function screening. Left: the test system is a Γ-only Ar crystal supercell containing 216 Ar atoms. Right: the test system is a Γ-only diamond supercell containing 1024 C atoms.

In Appendix B, the decay behaviors of the Green's-function matrix elements $G_{ij}(\mathbf{R}, i\tau)$ for two selected systems, the Ar crystal and the C diamond crystal, are presented. Figure 6 shows that the Green's function matrix elements decay rather fast as a function of the distance $d = |\mathbf{R} + \boldsymbol{\tau}_J - \boldsymbol{\tau}_I|$ between the atomic centers. The largest amplitude occurs at time $\tau = 0$, corresponding to the density matrix of the system. For Ar, $\max\{G_{ij}(\mathbf{R}, i\tau = 0)\}$ becomes vanishingly small for $d \geqslant 10$ Å; for C diamond, the decaying of $\max\{G_{ij}(\mathbf{R}, i\tau = 0)\}$ is less fast, but its magnitude also becomes rather small for $d > 20$ Å.

Moreover, for finite τ, the amplitude of the imaginary-time Green's function decays rather fast as τ increases, and becomes tiny for the entire distance range when $\tau > 10$ a.u, as can be seen from Fig. 6. Thus, the overall decay behavior of the Green's function in real space is governed by the $\tau = 0$ case, i.e., the density matrix. Consequently, we can expect a linear-scaling behavior of the construction of the χ^0 matrix, within an atomic orbital representation and LRI approximation. The situation is rather similar to the linear-scaling algorithm developed for the construction of the Hartree-Fock exchange matrix in terms of NAO basis sets [42,44]. This above line of reasoning applies perfectly to insulating systems, where the density matrix, and more generally the imaginary-time Green's function, is warranted to decay exponentially in real space [58–60]. For metallic systems, the situation is more complicated since the density matrix (and Green's function) decays much slower in real space. We expect that a linear-scaling behavior can eventually be achieved, but may occur only at very large systems, not in the regime of 10^3 atoms that are tested in the present work.

D. Interface with ABACUS

As a stand-alone library, LibRPA can also be interfaced with other NAO-based DFT codes besides FHI-aims, provided that the LRI infrastructure is available. ABACUS [46,47] is a DFT software that employs NAOs as its primary basis set choice and norm-conserving pseudopotentials for describing core-valence interactions. In particular, the LRI has been implemented in ABACUS, which enabled efficient hybrid functional calculations [43,44,63]. As indicated in Fig. 1, once the LRI expansion coefficients $C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R})}$ and the Coulomb matrix $V_{\mu\nu}(\mathbf{k})$ are available, interfacing an NAO-based DFT code with LibRPA is straightforward. Figure 5 demonstrates the scaling behavior of the computation time of LibRPA interfaced with ABACUS with respect to system size. The test systems consist of Si diamond structures of increasing supercell sizes and only a single $\mathbf{k}$ point is used. The double-ζ plus polarization (DZP) NAO basis set ($2s2p1d$ for Si) is used in the calculations, whereby the compact basis size allows us to go to system size of over 1000 atoms in the supercell. We demonstrate both the computation time for evaluating χ^0 matrix and the total time for the RPA correlation energy calculation, with and without turning on the Green's-function-based screening. Two observations are noteworthy: First, the Green's-function-based screening starts to have an effect for system sizes larger than 200 atoms, and can significantly reduce the computational cost for evaluating χ^0; in fact, for system size between 800 and 1400 atoms, the computational cost indeed shows a linear, or even sublinear scaling with system size, when the Green's-function-based screening is invoked. Second, for system size larger than 800 atoms, the computation of the RPA correlation energy after obtaining the χ^0 matrix, which involves $O(N^3)$ steps, starts to play a significant role and will eventually dominate the calculations for even larger systems. Thus, for very large systems, one will also need to develop more efficient lower-scaling algorithms for executing $\chi^0 V$, and for computing the determinant of $1 - \chi^0 V$. However, this goes beyond the scope of the present paper and will be pursued in future work.

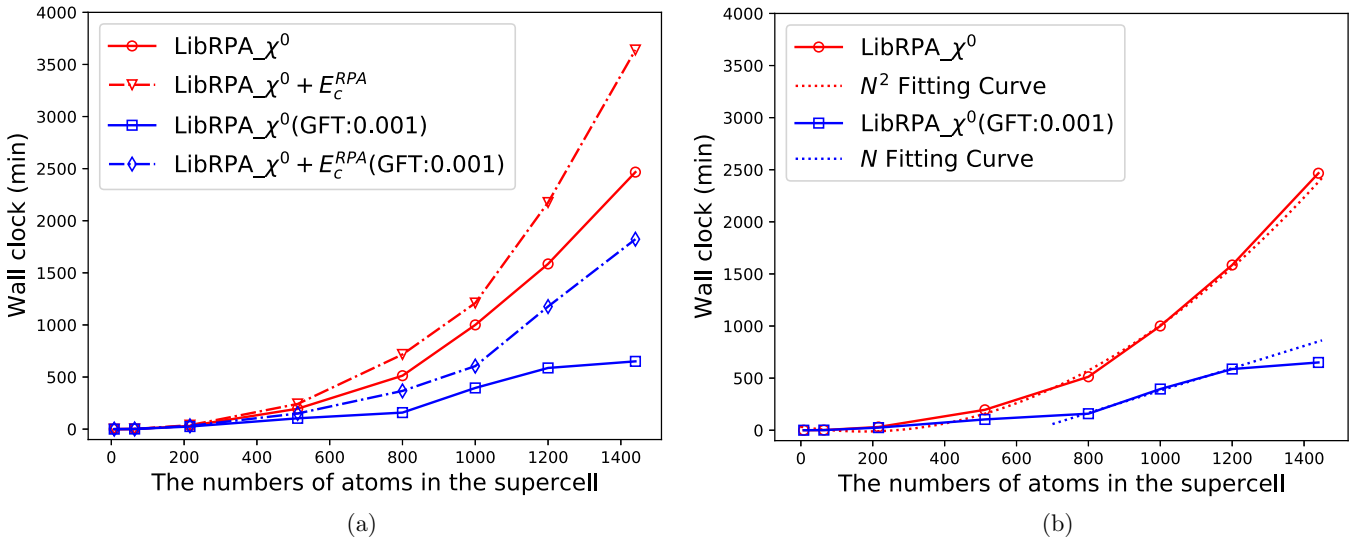


FIG. 5. Scaling behavior the computation times with respect to system size for LibRPA interfaced with ABACUS. The test systems are Si diamond with increasing supercell size. A single $\mathbf{k}$ point and the NAO DZP basis set are used is the calculations. Lef: the timings for both evaluating χ^0 matrix (solid lines) and the total RPA calculation $\chi^0 + E_c^{\mathrm{RPA}}$ (dash-dotted curves) with (blue curves) and without (red curves) switching on the Green's-function-based screening. Right: $O(N)$- and $O(N^2)$-scaling curves (dotted lines) are added by fitting the data of the low-scaling algorithms for evaluating the χ^0 matrix with and without Green's-function-based screening, respectively.

V. SUMMARY

The application of the RPA method to complex materials has been hampered by its quickly increasing computational cost. The rate-determining step for RPA correlation energy calculations in conventional algorithm is the evaluation of the response function matrix χ^0. In this work, we present a low-scaling algorithm for evaluating the χ^0 matrix, by combining the real-space, imaginary-time representation of χ^0, the strict locality of NAO basis functions, as well as the localized resolution of identity. The algorithm has a formal $O(N^2)$ scaling by only taking into account of the sparsity of the LRI expansion coefficients, and becomes linear if the decay behavior of the Green's function in real space is further utilized. Benchmark calculations for systems of increasing sizes confirmed the scaling behavior of the proposed algorithm, and benchmark against the conventional **k**-space algorithm confirms the validity and high numerical precision of the present algorithm. We particularly show that the Green's-function-based screening, which has been so far largely overlooked, can bring significant additional savings for system sizes of over a few hundred atoms. We also observe that the $O(N^3)$-scaling steps in RPA calculations after the χ^0 matrix is obtained, whose computational cost is negligible in conventional algorithm, starts to dominate for system sizes over 1000 atoms. Further work is needed to develop low-scaling algorithms for the $O(N^3)$ steps. Our work sets a new standard for large-scale periodic RPA calculations using atomic orbitals. The low-scaling algorithm we developed and the insights we gained in the present work not only push the limit for RPA calculations, but are also helpful for extending the reach of other correlated methods to unprecedented size of periodic systems.

ACKNOWLEDGMENTS

We acknowledge the funding support by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDB0500201 and by the National Key Research and Development Program of China (Grant No. 2022YFA1403800). This work was also supported by the Chinese National Science Foundation Grants No. 12134012, No. 12374067, and No. 12188101. The numerical calculations in this study were carried out on the ORISE Supercomputer.

APPENDIX A: SPACE-TIME RPA WITHIN NAO

Equations (12)–(15) in the main text are the key equations behind our low-scaling RPA algorithm designed for local atomic basis set framework. These equations are presented in Sec. II without derivation. Due to their importance for the entire algorithm, we derive these equations here for completeness. The starting points are Eqs. (9) and (11) in the main text. Plugging Eq. (11) into Eq. (9), one obtains

$$\begin{aligned}\chi^0(\mathbf{r},\mathbf{r}',i\tau) = &-i\sum_{i,j,k,l}\sum_{\mathbf{R}_1,\mathbf{R}_2,\mathbf{R}_3,\mathbf{R}_4}\left(\sum_{\mu\in I} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_3-\mathbf{R}_1)}P_\mu(\mathbf{r}-\mathbf{R}_1-\boldsymbol{\tau}_I)+\sum_{\mu\in K} C^{\mu(\mathbf{0})}_{i(\mathbf{R}_1-\mathbf{R}_3),k(\mathbf{0})}P_\mu(\mathbf{r}-\mathbf{R}_3-\boldsymbol{\tau}_K)\right)\\ &\times G_{i,j}(\mathbf{R}_2-\mathbf{R}_1,i\tau)G_{l,k}(\mathbf{R}_3-\mathbf{R}_4,-i\tau)\\ &\times\left(\sum_{\nu\in J} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_4-\mathbf{R}_2)}P_\nu(\mathbf{r}'-\mathbf{R}_2-\boldsymbol{\tau}_J)+\sum_{\nu\in L} C^{\nu(\mathbf{0})}_{j(\mathbf{R}_2-\mathbf{R}_4),l(\mathbf{0})}P_\nu(\mathbf{r}'-\mathbf{R}_4-\boldsymbol{\tau}_L)\right)\\ =&\,\chi^{0(A)}(\mathbf{r},\mathbf{r}',i\tau)+\chi^{0(B)}(\mathbf{r},\mathbf{r}',i\tau)+\chi^{0(C)}(\mathbf{r},\mathbf{r}',i\tau)+\chi^{0(D)}(\mathbf{r},\mathbf{r}',i\tau),\end{aligned}\tag{A1}$$

where the full response function naturally splits into four terms, arising from the special structure due to LRI. These four terms correspond to four different ways of placing the ABFs on the four atoms I, J, K, and L, within the LRI approximation. Below we discuss these four terms separately. In Eq. (A1), the first term deals with the situation in which the ABF μ, ν sit on the atom I, J (denoted as $\mu\in I$ and $\nu\in J$), respectively. This term is given by

$$\begin{aligned}\chi^{0(A)}(\mathbf{r},\mathbf{r}',i\tau) =&-i\sum_{i,j,k,l}\sum_{\mathbf{R}_1,\mathbf{R}_2,\mathbf{R}_3,\mathbf{R}_4}\sum_{\mu\in I} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_3-\mathbf{R}_1)}P_\mu(\mathbf{r}-\mathbf{R}_1-\boldsymbol{\tau}_I)G_{i,j}(\mathbf{R}_2-\mathbf{R}_1,i\tau)\\ &\times G_{l,k}(\mathbf{R}_3-\mathbf{R}_4,-i\tau)\sum_{\nu\in J} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_4-\mathbf{R}_2)}P_\nu(\mathbf{r}'-\mathbf{R}_2-\boldsymbol{\tau}_J)\\ =&-i\sum_{\mu,\nu,\mathbf{R}_1,\mathbf{R}_2}P_\mu(\mathbf{r}-\mathbf{R}_1-\boldsymbol{\tau}_I)\sum_{i\in\mathcal{U},j\in\mathcal{V}}\sum_{k,\mathbf{R}_3} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_3-\mathbf{R}_1)}G_{i,j}(\mathbf{R}_2-\mathbf{R}_1,i\tau)\\ &\times\sum_{l,\mathbf{R}_4}\left[C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_4-\mathbf{R}_2)}G_{l,k}(\mathbf{R}_3-\mathbf{R}_4,-i\tau)\right]P_\nu(\mathbf{r}'-\mathbf{R}_2-\boldsymbol{\tau}_J)\\ =&\sum_{\mu,\nu,\mathbf{R}_1,\mathbf{R}_2}P_\mu(\mathbf{r}-\mathbf{R}_1-\boldsymbol{\tau}_\mathcal{U})\chi^{0(A)}_{\mu,\nu}(\mathbf{R}_2-\mathbf{R}_1,i\tau)P_\nu(\mathbf{r}'-\mathbf{R}_2-\boldsymbol{\tau}_\mathcal{V}),\end{aligned}\tag{A2}$$

where

$$\chi^{0(A)}_{\mu,\nu}(\mathbf{R}_2-\mathbf{R}_1,i\tau) = -i\sum_{i\in\mathcal{U},j\in\mathcal{V}}\sum_{k,\mathbf{R}_3}\sum_{l,\mathbf{R}_4} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_3-\mathbf{R}_1)}G_{i,j}(\mathbf{R}_2-\mathbf{R}_1,i\tau)C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_4-\mathbf{R}_2)}G_{l,k}(\mathbf{R}_3-\mathbf{R}_4,-i\tau).\tag{A3}$$

Recall that $\mathcal{U}$ and $\mathcal{V}$ denote the atoms where the ABFs μ, ν are centering, and $\boldsymbol{\tau}_\mathcal{U}$ and $\boldsymbol{\tau}_\mathcal{V}$ are their respective atomic positions in the unit cell. In the above derivation, we have used the fact that, in the present situation, the atom $\mathcal{U} = I$, and $\mathcal{V} = J$ (and hence $\boldsymbol{\tau}_\mathcal{U} = \boldsymbol{\tau}_I$, and $\boldsymbol{\tau}_\mathcal{U} = \boldsymbol{\tau}_J$). We have also used the property that, in the computation of $\chi^{0(A)}(\mathbf{r}, \mathbf{r}', i\tau)$, first looping over the AOs i, j and requiring $\mu \in I$ and $\nu \in J$ is equivalent to first looping over the ABFs μ, ν, and requiring the AOs $i \in \mathcal{U}$ and $j \in \mathcal{V}$. Making use of the translational symmetry, we can, without losing generality, set $\mathbf{R} = \mathbf{R}_2 - \mathbf{R}_1$ and $\mathbf{R}_1 = \mathbf{0}$. Finally we obtain

$$\chi^{0(A)}_{\mu,\nu}(\mathbf{R}, i\tau) = -i \sum_{i\in\mathcal{U}, j\in\mathcal{V}} \sum_{k,\mathbf{R}_3} \sum_{l,\mathbf{R}_4} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_3)} G_{l,k}(\mathbf{R}_3 - \mathbf{R}_4, -i\tau) C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_4-\mathbf{R})} G_{i,j}(\mathbf{R}, i\tau). \tag{A4}$$

Next, we deal with the second term that corresponds to the situation in which $\mu \in I$ and $\nu \in L$ (i.e., $I = \mathcal{U}$ and $L = \mathcal{V}$). Specifically,

$$\begin{aligned}
\chi^{0(B)}(\mathbf{r}, \mathbf{r}', i\tau) &= -i \sum_{i,j,k,l} \sum_{\mathbf{R}_1,\mathbf{R}_2,\mathbf{R}_3,\mathbf{R}_4} \sum_{\mu\in I} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_3-\mathbf{R}_1)} P_\mu(\mathbf{r} - \mathbf{R}_1 - \boldsymbol{\tau}_I) G_{i,j}(\mathbf{R}_2 - \mathbf{R}_1, i\tau) \\
&\quad \times G_{l,k}(\mathbf{R}_3 - \mathbf{R}_4, -i\tau) \sum_{\nu\in L} C^{\nu(\mathbf{0})}_{j(\mathbf{R}_2-\mathbf{R}_4),l(\mathbf{0})} P_\nu(\mathbf{r}' - \mathbf{R}_4 - \boldsymbol{\tau}_L) \\
&\overset{\mathbf{R}_2\leftrightarrow\mathbf{R}_4}{=} -i \sum_{i,j,k,l} \sum_{\mathbf{R}_1,\mathbf{R}_2,\mathbf{R}_3,\mathbf{R}_4} \sum_{\mu\in I} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_3-\mathbf{R})_1} P_\mu(\mathbf{r} - \mathbf{R}_1 - \boldsymbol{\tau}_I) G_{i,j}(\mathbf{R}_4 - \mathbf{R}_1, i\tau) \\
&\quad \times G_{l,k}(\mathbf{R}_3 - \mathbf{R}_2, -i\tau) \sum_{\nu\in L} C^{\nu(\mathbf{0})}_{j(\mathbf{R}_4-\mathbf{R}_2),l(\mathbf{0})} P_\nu(\mathbf{r}' - \mathbf{R}_2 - \boldsymbol{\tau}_L) \\
&= -i \sum_{\mu,\nu,\mathbf{R}_1,\mathbf{R}_2} P_\mu(\mathbf{r} - \mathbf{R}_1 - \boldsymbol{\tau}_\mathcal{U}) \sum_{i\in\mathcal{U}, l\in\mathcal{V}} \sum_{k,\mathbf{R}_3} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_3-\mathbf{R}_1)} G_{i,j}(\mathbf{R}_4 - \mathbf{R}_1, i\tau) \\
&\quad \times \sum_{j,\mathbf{R}_4} C^{\nu(\mathbf{0})}_{j(\mathbf{R}_4-\mathbf{R}_2),l(\mathbf{0})} G_{l,k}(\mathbf{R}_3 - \mathbf{R}_2, -i\tau) P_\nu(\mathbf{r}' - \mathbf{R}_2 - \boldsymbol{\tau}_\mathcal{V}) \\
&= \sum_{\mu,\nu,\mathbf{R}_1,\mathbf{R}_2} P_\mu(\mathbf{r} - \mathbf{R}_1 - \boldsymbol{\tau}_\mathcal{U}) \chi^{0(B)}_{\mu,\nu}(\mathbf{R}_2 - \mathbf{R}_1, i\tau) P_\nu(\mathbf{r}' - \mathbf{R}_2 - \boldsymbol{\tau}_\mathcal{V}),
\end{aligned} \tag{A5}$$

where

$$\chi^{0(B)}_{\mu,\nu}(\mathbf{R}_2 - \mathbf{R}_1, i\tau) = -i \sum_{i\in\mathcal{U}, l\in\mathcal{V}} \sum_{k,\mathbf{R}_3} \sum_{j,\mathbf{R}_4} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_3-\mathbf{R}_1)} G_{i,j}(\mathbf{R}_4 - \mathbf{R}_1, i\tau) C^{\nu(\mathbf{0})}_{j(\mathbf{R}_4-\mathbf{R}_2),l(\mathbf{0})} G_{l,k}(\mathbf{R}_3 - \mathbf{R}_2, -i\tau) \tag{A6}$$

or

$$\chi^{0(B)}_{\mu,\nu}(\mathbf{R}, i\tau) = -i \sum_{i\in\mathcal{U}, l\in\mathcal{V}} \sum_{k,\mathbf{R}_3} \sum_{j,\mathbf{R}_4} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_3)} G_{l,k}(\mathbf{R}_3 - \mathbf{R}, -i\tau) C^{\nu(\mathbf{0})}_{j(\mathbf{R}_4-\mathbf{R}),l(\mathbf{0})} G_{i,j}(\mathbf{R}_4, i\tau). \tag{A7}$$

In the derivation of Eq. (A5), we again used the property that, in the present case, looping over the AOs i, l and requiring $\mu \in I$ and $\nu \in L$ is equivalent to first looping over the ABFs μ, ν and requiring $i \in \mathcal{U}$ and $l \in \mathcal{V}$.

Next comes the third term, corresponding to the situation where $\mu \in K$ and $\nu \in J$ (i.e., $K = \mathcal{U}$ and $J = \mathcal{V}$),

$$\begin{aligned}
\chi^{0(C)}(\mathbf{r}, \mathbf{r}', i\tau) &= -i \sum_{i,j,k,l} \sum_{\mathbf{R}_1,\mathbf{R}_2,\mathbf{R}_3,\mathbf{R}_4} \sum_{\mu\in K} C^{\mu(\mathbf{0})}_{i(\mathbf{R}_1-\mathbf{R}_3),k(\mathbf{0})} P_\mu(\mathbf{r} - \mathbf{R}_3 - \boldsymbol{\tau}_K) G_{i,j}(\mathbf{R}_2 - \mathbf{R}_1, i\tau) \\
&\quad \times G_{l,k}(\mathbf{R}_3 - \mathbf{R}_4, -i\tau) \sum_{\nu\in J} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_4-\mathbf{R}_2)} P_\nu(\mathbf{r}' - \mathbf{R}_2 - \boldsymbol{\tau}_J) \\
&\overset{\mathbf{R}_1\leftrightarrow\mathbf{R}_3}{=} -i \sum_{i,j,k,l} \sum_{\mathbf{R}_1,\mathbf{R}_2,\mathbf{R}_3,\mathbf{R}_4} \sum_{\mu\in K} C^{\mu(\mathbf{0})}_{i(\mathbf{R}_3-\mathbf{R}_1),k(\mathbf{0})} P_\mu(\mathbf{r} - \mathbf{R}_1 - \boldsymbol{\tau}_K) G_{i,j}(\mathbf{R}_2 - \mathbf{R}_3, i\tau) \\
&\quad \times G_{l,k}(\mathbf{R}_1 - \mathbf{R}_4, -i\tau) \sum_{\nu\in J} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_4-\mathbf{R}_2)} P_\nu(\mathbf{r}' - \mathbf{R}_2 - \boldsymbol{\tau}_J) \\
&= -i \sum_{\mu,\nu,\mathbf{R}_1,\mathbf{R}_2} P_\mu(\mathbf{r} - \mathbf{R}_1 - \boldsymbol{\tau}_\mathcal{U}) \sum_{k\in\mathcal{U}, j\in\mathcal{V}} \sum_{i,\mathbf{R}_3} \sum_{l,\mathbf{R}_2} C^{\mu(\mathbf{0})}_{i(\mathbf{R}_3-\mathbf{R}_1),k(\mathbf{0})} G_{i,j}(\mathbf{R}_2 - \mathbf{R}_3, i\tau) \\
&\quad \times C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_4-\mathbf{R}_2)} G_{l,k}(\mathbf{R}_1 - \mathbf{R}_4, -i\tau) P_\nu(\mathbf{r}' - \mathbf{R}_2 - \boldsymbol{\tau}_\mathcal{V}) \\
&= \sum_{\mu,\nu,\mathbf{R}_1,\mathbf{R}_2} P_\mu(\mathbf{r} - \mathbf{R}_1 - \boldsymbol{\tau}_\mathcal{U}) \chi^{0(C)}_{\mu,\nu}(\mathbf{R}_2 - \mathbf{R}_1, i\tau) P_\nu(\mathbf{r}' - \mathbf{R}_2 - \boldsymbol{\tau}_\mathcal{V}),
\end{aligned} \tag{A8}$$

where

$$\chi^{0(C)}_{\mu,\nu}(\mathbf{R}_2-\mathbf{R}_1,i\tau)=-i\sum_{k\in\mathcal{U},j\in\mathcal{V}}\sum_{i,\mathbf{R}_3}\sum_{l,\mathbf{R}_4}C^{\mu(\mathbf{0})}_{i(\mathbf{R}_3-\mathbf{R}_1),k(\mathbf{0})}G_{i,j}(\mathbf{R}_2-\mathbf{R}_3,i\tau)C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_4-\mathbf{R}_2)}G_{l,k}(\mathbf{R}_1-\mathbf{R}_4,-i\tau), \tag{A9}$$

or

$$\chi^{0(C)}_{\mu,\nu}(\mathbf{R},i\tau)=-i\sum_{k\in\mathcal{U},j\in\mathcal{V}}\sum_{i,\mathbf{R}_3}\sum_{l,\mathbf{R}_4}C^{\mu(\mathbf{0})}_{i(\mathbf{R}_3),k(\mathbf{0})}G_{i,j}(\mathbf{R}-\mathbf{R}_3,i\tau)C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_4-\mathbf{R})}G_{l,k}(-\mathbf{R}_4,-i\tau). \tag{A10}$$

Finally we deal with the fourth term, which corresponds to the situation where $\mu\in K$ and $\nu\in L$ (i.e., $K=\mathcal{U}$ and $L=\mathcal{V}$),

$$\begin{aligned}
\chi^{0(D)}(\mathbf{r},\mathbf{r}',i\tau)&=-i\sum_{i,j,k,l}\sum_{\mathbf{R}_1,\mathbf{R}_2,\mathbf{R}_3,\mathbf{R}_4}\sum_{\mu\in K}C^{\mu(\mathbf{0})}_{i(\mathbf{R}_1-\mathbf{R}_3),k(\mathbf{0})}P_\mu(\mathbf{r}-\mathbf{R}_3-\boldsymbol{\tau}_K)G_{i,j}(\mathbf{R}_2-\mathbf{R}_1,i\tau)\\
&\quad\times G_{l,k}(\mathbf{R}_3-\mathbf{R}_4,-i\tau)\sum_{\nu\in L}C^{\nu(\mathbf{0})}_{j(\mathbf{R}_2-\mathbf{R}_4),l(\mathbf{0})}P_\nu(\mathbf{r}'-\mathbf{R}_4-\boldsymbol{\tau}_K)\\
&\overset{\mathbf{R}_1\leftrightarrow\mathbf{R}_3,\mathbf{R}_2\leftrightarrow\mathbf{R}_4}{=}-i\sum_{i,j,k,l}\sum_{\mathbf{R}_1,\mathbf{R}_2,\mathbf{R}_3,\mathbf{R}_4}\sum_{\mu\in K}C^{\mu(\mathbf{0})}_{i(\mathbf{R}_3-\mathbf{R}_1),k(\mathbf{0})}P_\mu(\mathbf{r}-\mathbf{R}_1-\boldsymbol{\tau}_K)G_{i,j}(\mathbf{R}_4-\mathbf{R}_3,i\tau)\\
&\quad\times G_{l,k}(\mathbf{R}_1-\mathbf{R}_2,-i\tau)\sum_{\nu\in L}C^{\nu(\mathbf{0})}_{j(\mathbf{R}_4-\mathbf{R}_2),l(\mathbf{0})}P_\nu(\mathbf{r}'-\mathbf{R}_2-\boldsymbol{\tau}_L)\\
&=-i\sum_{\mu,\nu,\mathbf{R}_1,\mathbf{R}_2}P_\mu(\mathbf{r}-\mathbf{R}_1-\boldsymbol{\tau}_\mathcal{U})\sum_{k\in\mathcal{U},l\in\mathcal{V}}\sum_{i,\mathbf{R}_3}\sum_{j,\mathbf{R}_4}C^{\mu(\mathbf{0})}_{i(\mathbf{R}_3-\mathbf{R}_1),k(\mathbf{0})}G_{i,j}(\mathbf{R}_4-\mathbf{R}_3,i\tau)\\
&\quad\times C^{\nu(\mathbf{0})}_{j(\mathbf{R}_4-\mathbf{R}_2),l(\mathbf{0})}G_{l,k}(\mathbf{R}_1-\mathbf{R}_2,-i\tau)P_\nu(\mathbf{r}'-\mathbf{R}_2-\boldsymbol{\tau}_\mathcal{U})\\
&=\sum_{\mu,\nu,\mathbf{R}_1,\mathbf{R}_2}P_\mu(\mathbf{r}-\mathbf{R}_1-\boldsymbol{\tau}_\mathcal{U})\chi^{0(D)}_{\mu,\nu}(\mathbf{R}_2-\mathbf{R}_1,i\tau)P_\nu(\mathbf{r}'-\mathbf{R}_2-\boldsymbol{\tau}_\mathcal{V}),
\end{aligned} \tag{A11}$$

where

$$\chi^{0(D)}_{\mu,\nu}(\mathbf{R}_2-\mathbf{R}_1,i\tau)=-i\sum_{k\in\mathcal{U},l\in\mathcal{V}}\sum_{i,\mathbf{R}_3}\sum_{j,\mathbf{R}_4}C^{\mu(\mathbf{0})}_{i(\mathbf{R}_3-\mathbf{R}_1),k(\mathbf{0})}G_{i,j}(\mathbf{R}_4-\mathbf{R}_3,i\tau)C^{\nu(\mathbf{0})}_{j(\mathbf{R}_4-\mathbf{R}_2),l(\mathbf{0})}G_{l,k}(\mathbf{R}_1-\mathbf{R}_2,-i\tau), \tag{A12}$$

or

$$\chi^{0(D)}_{\mu,\nu}(\mathbf{R},i\tau)=-i\sum_{k\in\mathcal{U},l\in\mathcal{V}}\sum_{i,\mathbf{R}_3}\sum_{j,\mathbf{R}_4}C^{\mu(\mathbf{0})}_{i(\mathbf{R}_3),k(\mathbf{0})}G_{i,j}(\mathbf{R}_4-\mathbf{R}_3,i\tau)C^{\nu(\mathbf{0})}_{j(\mathbf{R}_4-\mathbf{R}),l(\mathbf{0})}G_{l,k}(-\mathbf{R},-i\tau). \tag{A13}$$

Summing up Eqs. (A4), (A7), (A10), and (A13), we obtain,

$$\begin{aligned}
\chi^{0}_{\mu,\nu}(\mathbf{R},i\tau)&=\chi^{0(A)}_{\mu,\nu}(\mathbf{R},i\tau)+\chi^{0(B)}_{\mu,\nu}(\mathbf{R},i\tau)+\chi^{0(C)}_{\mu,\nu}(\mathbf{R},i\tau)+\chi^{0(D)}_{\mu,\nu}(\mathbf{R},i\tau)\\
&=-i\Bigg[\sum_{i\in\mathcal{U},j\in\mathcal{V}}\sum_{k,\mathbf{R}_3}\sum_{l,\mathbf{R}_4}C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_3)}G_{l,k}(\mathbf{R}_3-\mathbf{R}_4,-i\tau)C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_4-\mathbf{R})}G_{i,j}(\mathbf{R},i\tau)\\
&\quad+\sum_{i\in\mathcal{U},l\in\mathcal{V}}\sum_{k,\mathbf{R}_3}\sum_{j,\mathbf{R}_4}C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_3)}G_{l,k}(\mathbf{R}_3-\mathbf{R},-i\tau)C^{\nu(\mathbf{0})}_{j(\mathbf{R}_4-\mathbf{R}),l(\mathbf{0})}G_{i,j}(\mathbf{R}_4,i\tau)\\
&\quad+\sum_{k\in\mathcal{U},j\in\mathcal{V}}\sum_{i,\mathbf{R}_3}\sum_{l,\mathbf{R}_4}C^{\mu(\mathbf{0})}_{i(\mathbf{R}_3),k(\mathbf{0})}G_{i,j}(\mathbf{R}-\mathbf{R}_3,i\tau)C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})}G_{l,k}(-\mathbf{R}_4,-i\tau)\\
&\quad+\sum_{k\in\mathcal{U},l\in\mathcal{V}}\sum_{i,\mathbf{R}_3}\sum_{j,\mathbf{R}_4}C^{\mu(\mathbf{0})}_{i(\mathbf{R}_3),k(\mathbf{0})}G_{i,j}(\mathbf{R}_4-\mathbf{R}_3,i\tau)C^{\nu(\mathbf{0})}_{j(\mathbf{R}_4-\mathbf{R}),l(\mathbf{0})}G_{l,k}(-\mathbf{R},-i\tau)\Bigg].
\end{aligned} \tag{A14}$$

To facilitate its computation and in particular the design of the loop structure in the low-scaling algorithm, we swap the dummy indices in the summation. Specifically, we perform the following exchanges for orbital indices: $j\leftrightarrow l$ for the second term, $i\leftrightarrow k$ for the third term, $i\leftrightarrow k$, $j\leftrightarrow l$ for the fourth term. And for all terms, further making the following replacement for the lattice vectors: $\mathbf{R}_3\to\mathbf{R}_1$, and $\mathbf{R}_4\to\mathbf{R}_2$, we have

$$\begin{aligned}
\chi^{0}_{\mu,\nu}(\mathbf{R},i\tau)&=-i\Bigg[\sum_{i\in\mathcal{U},j\in\mathcal{V}}\sum_{k,\mathbf{R}_1}\sum_{l,\mathbf{R}_2}C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)}G_{l,k}(\mathbf{R}_1-\mathbf{R}_2,-i\tau)C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})}G_{i,j}(\mathbf{R},i\tau)\\
&\quad+\sum_{i\in\mathcal{U},j\in\mathcal{V}}\sum_{k,\mathbf{R}_1}\sum_{l,\mathbf{R}_2}C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)}G_{j,k}(\mathbf{R}_1-\mathbf{R},-i\tau)C^{\nu(\mathbf{0})}_{l(\mathbf{R}_2-\mathbf{R}),j(\mathbf{0})}G_{i,l}(\mathbf{R}_2,i\tau)
\end{aligned}$$

$$
\begin{aligned}
&+\sum_{i\in\mathcal{U},j\in\mathcal{V}}\sum_{k,\mathbf{R}_1}\sum_{l,\mathbf{R}_2} C^{\mu(\mathbf{0})}_{k(\mathbf{R}_1),i(\mathbf{0})} G_{k,j}(\mathbf{R}-\mathbf{R}_1,i\tau) C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{l,i}(-\mathbf{R}_2,-i\tau)\\
&+\sum_{i\in\mathcal{U},j\in\mathcal{V}}\sum_{k,\mathbf{R}_1}\sum_{l,\mathbf{R}_2} C^{\mu(\mathbf{0})}_{k(\mathbf{R}_1),i(\mathbf{0})} G_{k,l}(\mathbf{R}_2-\mathbf{R}_1,i\tau) C^{\nu(\mathbf{0})}_{l(\mathbf{R}_2-\mathbf{R}),j(\mathbf{0})} G_{j,i}(-\mathbf{R},-i\tau)\Bigg]\\
=-i\Bigg[&\sum_{i\in\mathcal{U},j\in\mathcal{V}}\sum_{k,\mathbf{R}_1}\sum_{l,\mathbf{R}_2} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)}\big(G_{l,k}(\mathbf{R}_1-\mathbf{R}_2,-i\tau) C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{i,j}(\mathbf{R},i\tau)\\
&+G_{j,k}(\mathbf{R}_1-\mathbf{R},-i\tau) C^{\nu(\mathbf{0})}_{l(\mathbf{R}_2-\mathbf{R}),j(\mathbf{0})} G_{i,l}(\mathbf{R}_2,i\tau)+G^*_{j,k}(\mathbf{R}_1-\mathbf{R},i\tau) C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{i,l}(\mathbf{R}_2,-i\tau)\\
&+G^*_{l,k}(\mathbf{R}_1-\mathbf{R}_2,i\tau) C^{\nu(\mathbf{0})}_{l(\mathbf{R}_2-\mathbf{R}),j(\mathbf{0})} G^*_{i,j}(\mathbf{R},-i\tau)\big)\Bigg],
\end{aligned}
\tag{A15}
$$

where we have utilized the symmetry properties for the Green's function, i.e., $G_{i,j}(\mathbf{R},i\tau)=G^*_{j,i}(-\mathbf{R},i\tau)$, and for the expansion coefficients, i.e., $C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)}$. Close inspection of the four terms in Eq. (A15) suggests that the first and fourth terms can be grouped together, and so do the second and third terms. Namely,

$$
\begin{aligned}
\chi^0_{\mu,\nu}(\mathbf{R},i\tau)=-i\Bigg[&\sum_{i\in\mathcal{U}}\sum_{k,\mathbf{R}_1} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)}\Bigg(\sum_{j\in\mathcal{V}} G_{i,j}(\mathbf{R},i\tau)\sum_{l,\mathbf{R}_2} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{l,k}(\mathbf{R}_1-\mathbf{R}_2,-i\tau)\\
&+\sum_{j\in\mathcal{V}} G^*_{i,j}(\mathbf{R},-i\tau)\sum_{l,\mathbf{R}_2} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G^*_{l,k}(\mathbf{R}_1-\mathbf{R}_2,i\tau)+\sum_{j\in\mathcal{V}} G_{j,k}(\mathbf{R}_1-\mathbf{R},-i\tau)\sum_{l,\mathbf{R}_2} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{i,l}(\mathbf{R}_2,i\tau)\\
&+\sum_{j\in\mathcal{V}} G^*_{j,k}(\mathbf{R}_1-\mathbf{R},i\tau)\sum_{l,\mathbf{R}_2} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G^*_{i,l}(\mathbf{R}_2,-i\tau)\Bigg)\Bigg]\\
=-i\Bigg[&\sum_{i\in\mathcal{U}}\sum_{k,\mathbf{R}_1} C^{\mu(\mathbf{0})}_{i(\mathbf{0}),k(\mathbf{R}_1)}\big(M^{\nu}_{i,k}(\mathbf{R}_1,\mathbf{R},i\tau)+M^{\nu*}_{i,k}(\mathbf{R}_1,\mathbf{R},-i\tau)+Z^{\nu}_{i,k}(\mathbf{R}_1,\mathbf{R},i\tau)+Z^{\nu*}_{i,k}(\mathbf{R}_1,\mathbf{R},-i\tau)\big)\Bigg],
\end{aligned}
\tag{A16}
$$

where

$$
\begin{aligned}
M^{\nu}_{i,k}(\mathbf{R}_1,\mathbf{R},i\tau)&=\sum_{j\in\mathcal{V}} G_{i,j}(\mathbf{R},i\tau)N^{\nu}_{j,k}(\mathbf{R}_1,\mathbf{R},i\tau)\\
N^{\nu}_{j,k}(\mathbf{R}_1,\mathbf{R},i\tau)&=\sum_{l,\mathbf{R}_2} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{l,k}(\mathbf{R}_1-\mathbf{R}_2,-i\tau)
\end{aligned}
$$

and

$$
\begin{aligned}
Z^{\nu}_{i,k}(\mathbf{R}_1,\mathbf{R},i\tau)&=\sum_{j\in\mathcal{V}} G_{j,k}(\mathbf{R}_1-\mathbf{R},-i\tau)X^{\nu}_{i,j}(\mathbf{R},i\tau)\\
X^{\nu}_{i,j}(\mathbf{R},i\tau)&=\sum_{l,\mathbf{R}_2} C^{\nu(\mathbf{0})}_{j(\mathbf{0}),l(\mathbf{R}_2-\mathbf{R})} G_{i,l}(\mathbf{R}_2,i\tau).
\end{aligned}
$$

Hence, Eqs. (12)–(15) in the main text are derived.

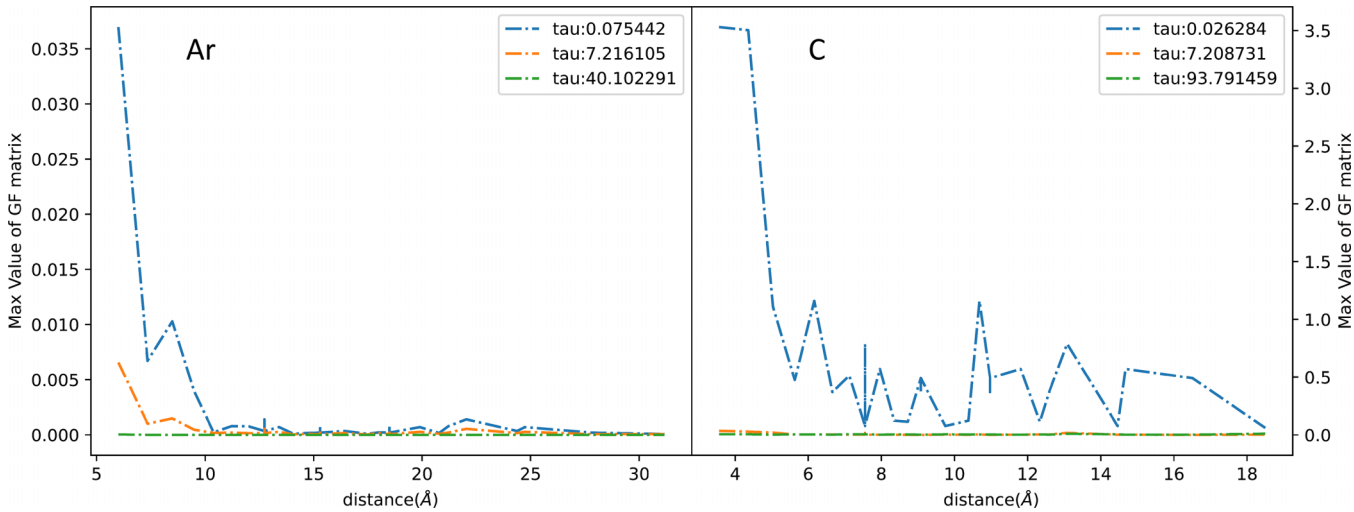
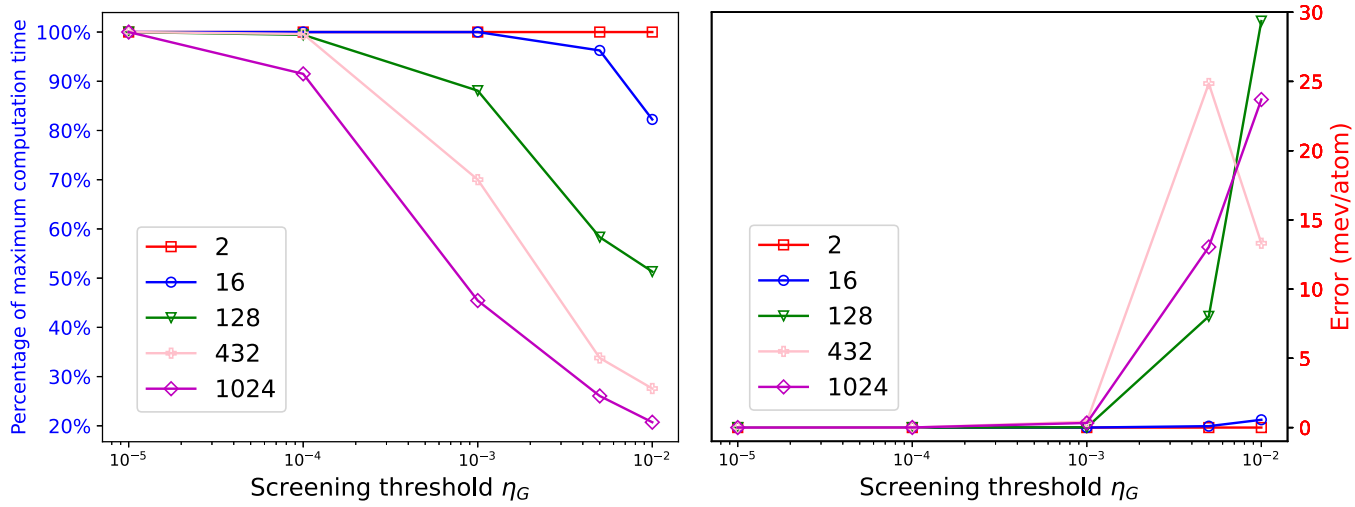


FIG. 6. The maximal matrix elements of Green's function as a function of the distance between the atomic centers for Ar crystal (left) and diamond (right). The calculations were done with ABACUS using NAO DZP basis set and 6×6×6 **k**-point mesh.



(a) Time (b) Error

FIG. 7. Reduction of the computational times and induced errors as a function of the Green's-function-based thresholding parameter η_G. The test system consists of Γ-only diamond supercells with increasing numbers of C atoms. The numbers in the legend indicate the number of carbon atoms in the supercell.

APPENDIX B: DECAY BEHAVIOR OF THE GREEN'S FUNCTION IN REAL SPACE

In Sec. IV C, we showed that significant further computational savings can be achieved if the Green's-function-based screening is incorporated. The scaling behavior of the refined real-space algorithm depends on the decay behavior of the Green's function in real space. In Fig. 6, we present the absolute values of the maximal matrix elements of the imaginary-time Green's function $G_{ij}(\mathbf{R}, i\tau)$ as a function of the distance between the atomic centers for the Ar crystal (left panel) and C diamond crystal (right panel), respectively. Namely, what is plotted are $\max_{i\in I, j\in J} |G_{ij}(\mathbf{R}, i\tau)|$ as a function of $d = |\mathbf{R} + \boldsymbol{\tau}_J - \boldsymbol{\tau}_I|$ at three different time points, i.e., $\tau = 0.075442, 7.216105, 40.102291$ a.u. for Ar and $\tau = 0.026284, 7.208731, 93.791459$ a.u. for C diamond. Figure 6 indicates that one needs much larger supercell size for C than for Ar to achieve a similar effect of Green-function-based screening.

In Fig. 7, the reduction of the computation times [Fig. 7(a)] and the incurred error [Fig. 7(b)] as a function of the Green's-function-based screening parameter η_G are plotted. The calculations were done for Γ-only C supercells of increasing size, containing $n = 2, 16, 128, 432, 1024$ atoms, respectively. The results for different supercell size are indicated by different lines in Fig. 7.

[1] D. Bohm and D. Pines, Phys. Rev. **92**, 609 (1953).
[2] M. Gell-Mann and K. A. Brueckner, Phys. Rev. **106**, 364 (1957).
[3] J. Hubbard, Proc. Roy. Soc. Lond. A **243**, 336 (1958).
[4] D. C. Langreth and J. P. Perdew, Phys. Rev. B **15**, 2884 (1977).
[5] O. Gunnarsson and B. I. Lundqvist, Phys. Rev. B **13**, 4274 (1976).
[6] F. Furche, Phys. Rev. B **64**, 195120 (2001).
[7] M. Fuchs and X. Gonze, Phys. Rev. B **65**, 235109 (2002).
[8] H. Eshuis, J. E. Bates, and F. Furche, Theor. Chem. Acc. **131**, 1084 (2012).
[9] X. Ren, P. Rinke, C. Joas, and M. Scheffler, J. Mater. Sci. **47**, 7447 (2012).
[10] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).
[11] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).
[12] J. P. Perdew and K. Schmidt, in *Density Functional Theory and its Application to Materials*, edited by V. Van Doren, C. Van Alsenoy, and P. Geerlings (AIP, Melville, 2001).
[13] X. Ren, P. Rinke, and M. Scheffler, Phys. Rev. B **80**, 045402 (2009).
[14] L. Schimka, J. Harl, A. Stroppa, A. Grüneis, M. Marsman, F. Mittendorfer, and G. Kresse, Nat. Mater. **9**, 741 (2010).
[15] J. Paier, X. Ren, P. Rinke, G. E. Scuseria, A. Grüneis, G. Kresse, and M. Scheffler, New J. Phys. **14**, 043002 (2012).
[16] X. Ren, P. Rinke, G. E. Scuseria, and M. Scheffler, Phys. Rev. B **88**, 035120 (2013).
[17] S. Lebègue, J. Harl, T. Gould, J. G. Ángyán, G. Kresse, and J. F. Dobson, Phys. Rev. Lett. **105**, 196401 (2010).
[18] Y. Zhang, J. W. Furness, B. Xiao, and J. Sun, J. Chem. Phys. **150**, 014105 (2019).
[19] N. Sengupta, J. E. Bates, and A. Ruzsinszky, Phys. Rev. B **97**, 235136 (2018).
[20] C. Cazorla and T. Gould, Sci. Adv. **5**, eaau5832 (2019).
[21] S. Yang and X. Ren, New J. Phys. **24**, 033049 (2022).
[22] D. Neuhauser, E. Rabani, and R. Baer, J. Phys. Chem. Lett. **4**, 1172 (2013).
[23] J. E. Moussa, J. Chem. Phys. **140**, 014107 (2014).

[24] M. Kaltak, J. Klimeš, and G. Kresse, J. Chem. Theory Comput. **10**, 2498 (2014).
[25] M. Kállay, J. Chem. Phys. **142**, 204105 (2015).
[26] J. Wilhelm, P. Seewald, M. Del Ben, and J. Hutter, J. Chem. Theory Comput. **12**, 5851 (2016).
[27] D. Graf, M. Beuerle, H. F. Schurkus, A. Luenser, G. Savasci, and C. Ochsenfeld, J. Chem. Theory Comput. **14**, 2505 (2018).
[28] A. Luenser, H. F. Schurkus, and C. Ochsenfeld, J. Chem. Theory Comput. **13**, 1647 (2017).
[29] J. Lu and K. Thicke, J. Comput. Phys. **351**, 187 (2017).
[30] I. Duchemin and X. Blase, J. Chem. Phys. **150**, 174120 (2019).
[31] H. N. Rojas, R. W. Godby, and R. J. Needs, Phys. Rev. Lett. **74**, 1827 (1995).
[32] I. D. White, R. W. Godby, M. M. Rieger, and R. J. Needs, Phys. Rev. Lett. **80**, 4265 (1998).
[33] M. M. Rieger, L. Steinbeck, I. White, H. Rojas, and R. Godby, Comput. Phys. Commun. **117**, 211 (1999).
[34] L. Hedin, Phys. Rev. **139**, A796 (1965).
[35] M. Kaltak, J. Klimeš, and G. Kresse, Phys. Rev. B **90**, 054115 (2014).
[36] P. Liu, M. Kaltak, J. Klimeš, and G. Kresse, Phys. Rev. B **94**, 165109 (2016).
[37] J. Wilhelm, P. Seewald, and D. Golze, J. Chem. Theory Comput. **17**, 1662 (2021).
[38] A. Förster and L. Visscher, J. Chem. Theory Comput. **16**, 7381 (2020).
[39] J. Lu and L. Yin, J. Comput. Phys. **302**, 329 (2015).
[40] Y. Gao, D. Neuhauser, R. Baer, and E. Rabani, J. Chem. Phys. **142**, 034106 (2015).
[41] A. C. Ihrig, J. Wieferink, I. Y. Zhang, M. Ropo, X. Ren, P. Rinke, M. Scheffler, and V. Blum, New J. Phys. **17**, 093020 (2015).
[42] S. V. Levchenko, X. Ren, J. Wieferink, R. Johanni, P. Rinke, V. Blum, and M. Scheffler, Comput. Phys. Commun. **192**, 60 (2015).
[43] P. Lin, X. Ren, and L. He, J. Phys. Chem. Lett. **11**, 3082 (2020).
[44] P. Lin, X. Ren, and L. He, J. Chem. Theory Comput. **17**, 222 (2021).
[45] V. Blum, F. Hanke, R. Gehrke, P. Havu, V. Havu, X. Ren, K. Reuter, and M. Scheffler, Comput. Phys. Commun. **180**, 2175 (2009).
[46] M. Chen, G.-C. Guo, and L. He, J. Phys.: Condens. Matter **22**, 445501 (2010).
[47] P. Li, X. Liu, M. Chen, P. Lin, X. Ren, L. Lin, C. Yang, and L. He, Comput. Mater. Sci. **112**, 503 (2016).
[48] The ABACUS software webpage: http://abacus.ustc.edu.cn.
[49] P. Merlot, T. Kjrgaard, T. Helgaker, R. Lindh, F. Aquilante, S. Reine, and T. B. Pedersen, J. Comput. Chem. **34**, 1486 (2013).
[50] L. N. Wirz, S. S. Reine, and T. B. Pedersen, J. Chem. Theory Comput. **13**, 4897 (2017).
[51] D. S. Hollman, H. F. Schaefer, and E. F. Valeev, J. Chem. Phys. **140**, 064109 (2014).
[52] X. Wang, C. A. Lewis, and E. F. Valeev, J. Chem. Phys. **153**, 124116 (2020).
[53] X. Ren, F. Merz, H. Jiang, Y. Yao, M. Rampp, H. Lederer, V. Blum, and M. Scheffler, Phys. Rev. Mater. **5**, 013807 (2021).
[54] E. Spadetto, P. H. T. Philipsen, A. Förster, and L. Visscher, J. Chem. Theory Comput. **19**, 1499 (2023).
[55] M. N. Tahir, T. Zhu, H. Shang, J. Li, V. Blum, and X. Ren, J. Chem. Theory Comput. **18**, 5297 (2022).
[56] I. Y. Zhang, A. J. Logsdail, X. Ren, S. V. Levchenko, L. Ghiringhelli, and M. Scheffler, New J. Phys. **21**, 013025 (2019).
[57] CP2K github repository: https://github.com/cp2k/cp2k.
[58] W. Kohn, Int. J. Quantum Chem. **56**, 229 (1995).
[59] S. Goedecker, Rev. Mod. Phys. **71**, 1085 (1999).
[60] E. Prodan and W. Kohn, Proc. Natl. Acad. Sci. USA **102**, 11635 (2005).
[61] LibRPA github repository: https://github.com/Srlive1201/LIBRPA.
[62] I. Y. Zhang, X. Ren, P. Rinke, V. Blum, and M. Scheffler, New J. Phys. **15**, 123033 (2013).
[63] Y. Ji, P. Lin, X. Ren, and L. He, J. Phys. Chem. A **126**, 5924 (2022).