JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

Enabling Large-Scale Condensed-Phase Hybrid Density Functional Theory Based *Ab Initio* Molecular Dynamics. 1. Theory, Algorithm, and Performance

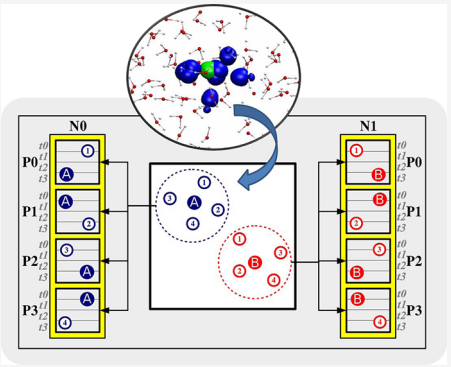
Hsin-Yu Ko, Junteng Jia, Biswajit Santra, Xifan Wu, Roberto Car, and Robert A. DiStasio Jr.*

Cite This: *J. Chem. Theory Comput.* 2020, 16, 3757−3785

Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: By including a fraction of exact exchange (EXX), hybrid functionals reduce the self-interaction error in semilocal density functional theory (DFT) and thereby furnish a more accurate and reliable description of the underlying electronic structure in systems throughout biology, chemistry, physics, and materials science. However, the high computational cost associated with the evaluation of all required EXX quantities has limited the applicability of hybrid DFT in the treatment of large molecules and complex condensed-phase materials. To overcome this limitation, we describe a linear-scaling approach that utilizes a local representation of the occupied orbitals (e.g., maximally localized Wannier functions (MLWFs)) to exploit the sparsity in the real-space evaluation of the quantum mechanical exchange interaction in finite-gap systems. In this work, we present a detailed description of the theoretical and algorithmic advances required to perform MLWF-based *ab initio* molecular dynamics (AIMD) simulations of large-scale condensed-phase systems of interest at the hybrid DFT level. We focus our theoretical discussion on the integration of this approach into the framework of Car−Parrinello AIMD, and highlight the central role played by the MLWF-product potential (i.e., the solution of Poisson's equation for each corresponding MLWF-product density) in the evaluation of the EXX energy and wave function forces. We then provide a comprehensive description of the `exx` algorithm implemented in the open-source `Quantum ESPRESSO` program, which employs a hybrid `MPI/OpenMP` parallelization scheme to efficiently utilize the high-performance computing (HPC) resources available on current- and next-generation supercomputer architectures. This is followed by a critical assessment of the accuracy and parallel performance (e.g., strong and weak scaling) of this approach when AIMD simulations of liquid water are performed in the canonical (*NVT*) ensemble. With access to HPC resources, we demonstrate that `exx` enables hybrid DFT-based AIMD simulations of condensed-phase systems containing 500−1000 atoms (e.g., $(H_2O)_{256}$) with a wall time cost that is comparable to that of semilocal DFT. In doing so, `exx` takes us one step closer to routinely performing AIMD simulations of complex and large-scale condensed-phase systems for sufficiently long time scales at the hybrid DFT level of theory.



I. INTRODUCTION

In view of its quite favorable balance of accuracy and computational cost, Kohn−Sham (KS) density functional theory[1−4] (DFT) has become the most widely used electronic structure method for *ab initio* molecular dynamics (AIMD) simulations of large molecules and complex condensed-phase materials.[5−7] Within the framework of KS-DFT, the total ground-state energy (E) is given as the sum of the following contributions:

$$E = E_{kin} + E_{ext} + E_H + E_{xc} \quad (1)$$

in which E_{kin} is the KS kinetic energy, E_{ext} is the external potential, which accounts for the nuclear−electronic and nuclear−nuclear potential energies (as well as any external fields), E_H is the Hartree energy, i.e., the average (classical) Coulomb interaction energy of the electrons, and E_{xc} is the electronic exchange−correlation (xc) energy. Explicit forms for all of the energy contributions in eq 1 are known except E_{xc}, the approximation of which is still the subject of active research to date.

Functional approximations to E_{xc} are often described as the rungs of "Jacob's Ladder," which connect the Hartree world to the exact solution of the time-independent Schrödinger equation.[8] In this hierarchical classification of DFT, the first rung is given by the local (spin) density approximation (LDA),[4,9] in which the form of E_{xc}^{LDA} is obtained from the solution to the homogeneous electron gas. As such, LDA

Received: November 25, 2019
Published: February 11, 2020

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

works particularly well for systems with a (nearly) uniform electron density ($\rho(\mathbf{r})$), e.g., the valence electrons in metallic solids. The next rung includes xc functionals based on the semilocal generalized gradient approximation (GGA),[10−12] which utilize the gradient of the electron density ($\nabla\rho(\mathbf{r})$) to correct the LDA description of systems with spatially varying $\rho(\mathbf{r})$, e.g., molecules and heterogeneous materials. At the current time, GGAs such as the nonempirical Perdew−Burke−Ernzerhof (PBE) xc functional[12] are the computational workhorses for AIMD simulations of condensed-phase systems containing hundreds to thousands of atoms. In this size regime, GGA-based approaches provide a favorable compromise between accuracy and computational cost and have been quite successful in qualitatively (and sometimes even quantitatively) describing a number of systems and processes of interest throughout chemistry, physics, and materials science.

Despite such widespread success, GGA functionals are unable to account for nonlocal electron correlation effects, which are responsible for the ubiquitous class of dispersion (or van der Waals) interactions. As such, several approaches have been devised to incorporate these long-range forces into the framework of DFT[13−16] and include effective pairwise models,[17−21] methods that account for many-body dispersion interactions,[22−26] as well as nonlocal xc functionals.[27−29] We note in passing that third-rung meta-GGA functionals, which incorporate second-derivative information via the Laplacian ($\nabla^2\rho(\mathbf{r})$) or the kinetic-energy density ($\tau(\mathbf{r})$), are able to account for intermediate-range correlation effects.[30−36] As such, these approaches have experienced a resurgence with the recent introduction of the SCAN functional,[35] which has shown promising results for bulk water systems[37−39,41,42] and interfacial water.[40,43]

Another significant shortcoming associated with GGA (as well as meta-GGA) functionals is their propensity to suffer from self-interaction error (SIE), an artifact in approximate xc functionals that manifests as a spurious interaction between an electron and itself.[44,45] In the presence of SIE, $\rho(\mathbf{r})$ is too delocalized, which in turn often leads to deleterious effects such as inadequate descriptions of transition states and charge transfer complexes,[46−48] underestimation of band gaps,[49] overestimation of lattice parameters in a wide variety of solids,[50] as well as excessive proton delocalization in liquid water,[51−53] to name a few. While SIE can be largely eliminated by self-interaction correction (SIC)-based methods,[44,54−56] the most commonly adopted approach for ameliorating the SIE present in semilocal KS-DFT is through the admixture of a fraction of exact exchange (EXX) in the underlying GGA (or meta-GGA) xc functional.[57] These so-called hybrid (or hyper-GGA) xc functionals constitute the fourth rung in the DFT hierarchy and can be written as (shown here as a correction to a GGA xc functional)

$$E_{\mathrm{xc}}^{\mathrm{hybrid}} = a_{\mathrm{x}}E_{\mathrm{xx}} + (1 - a_{\mathrm{x}})E_{\mathrm{x}}^{\mathrm{GGA}} + E_{\mathrm{c}}^{\mathrm{GGA}} \tag{2}$$

in which E_{xx} is the EXX energy, $E_{\mathrm{x}}^{\mathrm{GGA}}$ is the GGA exchange energy, and $E_{\mathrm{c}}^{\mathrm{GGA}}$ is the GGA correlation energy. The mixing parameter (a_{x}) in this expression depends on the hybrid xc functional approximation,[57−59] the optimal value of which (for a given system) can be determined from a self-consistent GW calculation.[60] By reducing the SIE, hybrid xc functionals are typically more accurate than GGA (or meta-GGA) approaches, in particular for the prediction of lattice parameters,[50] reaction energy barriers,[46−48] and band gaps.[61] In this work, we limit our focus to the nonempirical PBE0[58] hybrid xc functional, in which $a_{\mathrm{x}} = 0.25$ and the PBE GGA functional[12] is used for $E_{\mathrm{x}}^{\mathrm{GGA}}$ and $E_{\mathrm{c}}^{\mathrm{GGA}}$. Application of our approach (which is described below) to other popular hybrid xc functionals such as B3LYP[11,57] is straightforward.

For a closed-shell system with N_{o} doubly occupied orbitals (bands), E_{xx} can be written as

$$E_{\mathrm{xx}} = -\sum_{ij}\int \mathrm{d}\mathbf{r}\int \mathrm{d}\mathbf{r}'\,\frac{\phi_i^*(\mathbf{r})\,\phi_j^*(\mathbf{r}')\,\phi_j(\mathbf{r})\,\phi_i(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} \tag{3}$$

in which ϕ_i and ϕ_j represent the occupied KS orbitals and the sum extends over all N_{o} states. Defining the orbital-product density as

$$\rho_{ij}(\mathbf{r}) \equiv \phi_i^*(\mathbf{r})\,\phi_j(\mathbf{r}) \tag{4}$$

and the corresponding orbital-product potential (i.e., the Coulomb potential felt by a test charge located at $\mathbf{r}$ originating from the $\rho_{ij}(\mathbf{r}')$ charge distribution) as

$$v_{ij}(\mathbf{r}) \equiv \int \mathrm{d}\mathbf{r}'\,\frac{\rho_{ij}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} \tag{5}$$

allows one to express eq 3 in the following compact form:

$$E_{\mathrm{xx}} = -\sum_{ij}\int \mathrm{d}\mathbf{r}\,\rho_{ij}(\mathbf{r})\,v_{ji}(\mathbf{r}) \tag{6}$$

Evaluation of $v_{ji}(\mathbf{r})$ is therefore of central importance in EXX calculations. For periodic systems, this quantity is usually computed through the convolution theorem (shown here at the Γ-point only),

$$\rho_{ji}(\mathbf{r}) \xrightarrow{\mathrm{fwdFFT}} \rho_{ji}(\mathbf{G})$$

$$v_{ji}(\mathbf{G}) = 4\pi\frac{\rho_{ji}(\mathbf{G})}{|\mathbf{G}|^2} \xrightarrow{\mathrm{invFFT}} v_{ji}(\mathbf{r}) \tag{7}$$

in which $\rho_{ji}(\mathbf{G})$ and $v_{ji}(\mathbf{G})$ are the Fourier coefficients of $\rho_{ji}(\mathbf{r})$ and $v_{ji}(\mathbf{r})$, respectively. We note in passing that the divergence of $v_{ji}(\mathbf{G})$ when $\mathbf{G} = \mathbf{0}$ needs to be treated with care when E_{xx} is evaluated using reciprocal-space methods. In the real-space algorithm described herein, we sidestep this divergence as $v_{ji}(\mathbf{G}=\mathbf{0})$ is implicitly determined by the boundary conditions imposed during the solution of Poisson's equation (i.e., $v_{ji}(r\to\infty) = 0$). The computational scaling associated with both the forward (`fwdFFT`) and inverse (`invFFT`) fast Fourier transforms is $O(N_{\mathrm{FFT}}\log N_{\mathrm{FFT}})$, where N_{FFT} is the size of the reciprocal space (planewave) grid, which grows linearly with system size. Since the evaluation of E_{xx} in eq 6 requires a sum over the contributions from all $N_{\mathrm{o}}(N_{\mathrm{o}}+1)/2$ unique pairs of occupied orbitals, the overall computational scaling becomes $O(N_{\mathrm{o}}^2 N_{\mathrm{FFT}}\log N_{\mathrm{FFT}})$. Neglecting the logarithmic dependence, the resulting cubic-scaling cost makes this reciprocal-space EXX algorithm quite computationally demanding and limits routine performance of hybrid DFT-based AIMD simulations on large-scale condensed-phase systems. Hence, most condensed-phase calculations with hybrid DFT still remain limited to predicting energetic and structural properties in the absence of thermal effects.

Significant progress has been made to accelerate condensed-phase EXX calculations by employing the following theoretical and numerical techniques: range separation[62] or truncation[63]

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

of the underlying Coulomb operator, implementation of massively parallel algorithms,[64−67] employment of auxiliary atom-centered (localized) basis sets,[68−70] adaptive compression (low-rank decomposition) of the EXX operator (ACE),[71−73] use of the projected commutator direct inversion of the iterative subspace (PC-DIIS) method to reduce the number of self-consistent field (SCF) iterations,[74] utilization of sparsity through localization methods (e.g., maximally localized Wannier functions (MLWFs),[75−78] recursive subspace bisection (RSB),[79,80] selected columns of the density matrix (SCDM),[81−83] and other localized representations[84]), as well as combinations thereof.[63,68,85−87]

To enable large-scale hybrid DFT-based AIMD simulations in the condensed phase, the most promising methods for reducing the intrinsic computational cost and scaling associated with EXX exploit sparsity via localized representations of the occupied space or density matrix. For example, the RSB method of Gygi and co-workers[79,80] uses a noniterative algebraic decomposition of the wave function coefficients, which provides a transformation from the occupied KS eigenstates to a set of localized orbitals that are contained within prescribed domains in real space. This method has already enabled a number of AIMD simulations using hybrid xc functionals (e.g., computational investigations into the density of ice at finite temperature,[88] ion solvation,[89,90] and the structural and vibrational properties of liquid water[51−53]) and is particularly convenient for simulating heterogeneous systems such as solid−liquid interfaces[91] due to the ease of selecting the prescribed localization domains. The SCDM method by Damle, Lin, and Ying exploits the sparsity of the off-diagonal elements of the density matrix[81−83] and does not rely on an initial guess to iteratively localize the occupied space. As such, this approach sidesteps issues related to gauge invariance and can furnish more robust (i.e., noniterative) localized orbitals than other optimization-based schemes.[92] The MLWF formalism introduced by Marzari and Vanderbilt[75] uses an iterative scheme to obtain a localized representation of the occupied KS orbitals by minimizing the total spread functional (e.g., the sum of the spreads of the individual localized orbitals) and therefore extends the well-known Boys orbital localization scheme[93] used in quantum chemistry into the condensed phase. MLWFs have shown great promise as both qualitative and quantitative analysis tools due to their similarity to the orbitals encountered in molecular orbital (MO) theory (i.e., bonding and lone pairs) and the fact that they allow one to obtain molecular multipole moments,[94−96] partition the charge density[97] and/or electrostatic potential,[98] and even compute nonbonded dispersion interactions[99] in complex condensed-phase environments. Numerous algorithms (such as `wannier90`[100]) for obtaining MLWFs have been incorporated into a number of existing community codes such as `Quantum ESPRESSO (QE)`,[101,102] `SIESTA`,[103] `ABINIT`,[104] `NWChem`,[105] `GPAW`,[106] `CP2K`,[107] and `VASP`,[108] which makes this localization scheme readily available and quite practical for *a posteriori* analyses of DFT-based calculations and AIMD simulations. Furthermore, the MLWF localization scheme is particularly suitable for large-scale hybrid DFT-based AIMD simulations since a Car−Parrinello-like propagation of MLWFs has already been demonstrated,[109−111] making the computational cost associated with orbital localization negligible between AIMD steps. In light of this computationally efficient orbital localization scheme, the wide availability of MLWFs, and the promise of a robust tool for on-the-fly analytics, we will now focus our discussion on the development and implementation of a linear-scaling (order(N)) MLWF-based EXX algorithm that can be used to perform large-scale condensed-phase AIMD simulations at the hybrid DFT level of theory.

In this work, we will focus on Car−Parrinello molecular dynamics (CPMD)[112] simulations of sufficiently large and finite-gap condensed-phase systems such that the first Brillouin zone can be accurately sampled at the Γ-point. Extensions to Born−Oppenheimer molecular dynamics (BOMD) and metallic systems[113,114] are possible and will be discussed in future work. Working at the Γ-point allows us to consider real-valued orbitals only, i.e., $\phi_i(\mathbf{r}) = \phi_i^*(\mathbf{r})$, from which it follows that $\rho_{ij}(\mathbf{r}) = \rho_{ji}(\mathbf{r})$ and $v_{ij}(\mathbf{r}) = v_{ji}(\mathbf{r})$ in eqs 4−6. Without loss of generality, we will also assume that the total wave function is closed shell (spin-unpolarized). Under these conditions, one can show that the set of MLWFs, which are obtained via an orthogonal (unitary) transformation of the occupied KS eigenstates, i.e.,

$$\tilde{\phi}_i(\mathbf{r}) = \sum_j U_{ij}\phi_j(\mathbf{r}) \tag{8}$$

have a significantly smaller support (or compact domain) than the entire simulation cell and are in fact exponentially localized in real space.[75,115−119] These features of the MLWF representation of the occupied space provide a theoretical and computational framework for exploiting the natural sparsity in the real-space evaluation of the EXX energy (and wave function forces) that we will explore in this work.

To demonstrate that the use of MLWFs leads to a linear-scaling EXX approach, consider the expression for E_{xx} in eq 6. Since this quantity is invariant to orthogonal transformations of the occupied orbitals (see section II.C), evaluation of E_{xx} can be performed exactly within the MLWF representation. The first level of computational savings originates from the fact that a given MLWF only appreciably overlaps with a subset of neighboring MLWFs. This makes the number of nonvanishing EXX pair interactions *per orbital* independent of the system size and thereby reduces the total number of orbital pairs required in the summation over i and j in eq 6. In addition, one can further exploit the fact that a numerically exact evaluation of E_{xx} only requires that the spatial integral in eq 6 be performed on the support of the orbital-product density. Since this quantity is sparse in the MLWF representation, this integration can be restricted to a real-space domain that is also independent of the system size. Taken together, these observations can be leveraged to construct a computationally efficient and linear-scaling MLWF-based algorithm for computing E_{xx}. For a more detailed description of the theoretical underpinnings of this approach and the associated algorithmic implementation, see sections II.C and III, respectively.

The initial concept and several pilot algorithms for this MLWF-based EXX approach[76,78] have already been successfully used to enable a number of large-scale hybrid DFT-based applications, e.g., computational investigations into the electronic structure of semiconducting solids,[120,121] the structural properties of ambient liquid water,[78,122,123] the structure and dynamics of aqueous ionic solutions,[124,125] and the thermal properties of the pyridine-I molecular crystal.[126] In this manuscript, we build upon our earlier work by presenting a detailed description of the theoretical and algorithmic advances that are required to perform accurate and efficient MLWF-

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

based AIMD simulations of large-scale condensed-phase systems at the hybrid DFT level. We focus our theoretical discussion on the integration of this approach into the CPMD framework by providing a detailed derivation of the EXX contributions to the equations of motion underlying fixed-cell CPMD simulations in the microcanonical (*NVE*) and canonical (*NVT*) ensembles. In particular, we include an in-depth discussion of a dual-level strategy, which describes how the use of localized orbitals (like MLWFs) can lead to a linear-scaling EXX algorithm by exploiting the underlying sparsity in the real-space evaluation of the exchange interaction. Influenced by the work of Gygi and co-workers,[79,80,91] we also introduce the concepts of MLWF-orbital and MLWF-product domains, which can be used to design an algorithmic framework that has the potential to enable accurate and efficient hybrid DFT simulations of condensed-phase systems with widely varying MLWF spreads.

In addition to this theoretical discussion, we also provide a comprehensive description of a massively parallel algorithm that extends well beyond our earlier pilot algorithms and uses this MLWF-based approach to compute all of the EXX contributions needed during hybrid DFT simulations of large-scale condensed-phase systems in the *NVE* and *NVT* ensembles. Recently implemented in the pseudopotential- and planewave-based open-source `QE` package,[102] the so-called `exx` module exploits this dual-level linear-scaling strategy and employs a hybrid message-passing interface (`MPI`) and open multiprocessing (`OpenMP`) parallelization scheme to efficiently utilize high-performance computing (HPC) resources. Compared with earlier pilot versions, `exx` significantly improves the applicability of our MLWF-based approach to large-scale AIMD (i.e., the strong-scaling limit) by introducing a hybrid parallelization scheme (which allows users to exploit both internode and intranode computational resources), a completely revised algorithm (which balances computation and communication and reduces the overall memory footprint), and a more flexible and general-purpose implementation (which accommodates a wide range of users, including those with limited computational resources as well as those working at the massively parallel HPC limit).

This is followed by a critical assessment of the accuracy and parallel performance (e.g., strong and weak scaling) of our implementation when AIMD simulations of liquid water are performed in the *NVT* ensemble on multiple different HPC architectures. In doing so, we demonstrate that `exx` enables hybrid DFT-based AIMD simulations of $(H_2O)_{256}$—a condensed-phase system containing >750 atoms—with a wall time cost that is comparable to semilocal DFT and minimal errors in the EXX contribution to the total energy, wave function forces, ionic forces, and binding energetics. As such, the work described herein will further enable us to utilize the fourth rung of DFT in the study of the structure, properties, and dynamics of a number of important condensed-phase systems, as well as perform hybrid DFT-based AIMD simulations across extended length and time scales that have been prohibitively difficult to access to date.

Although the current version of `exx` is restricted to condensed-phase systems in fixed orthorhombic simulation cells, an extension of this approach that treats general Bravais lattices and allows for hybrid DFT-based AIMD simulations in the isobaric−isoenthalpic (*NpH*) and isobaric−isothermal (*NpT*) ensembles will be discussed in the next paper in this series. Since `exx` is quite modular, this algorithm can also be incorporated into any planewave-based DFT code; when combined with linear-scaling GGA codes such as `PARSEC`,[127,128] `BigDFT`,[129] `ONETEP`,[130] or `CONQUEST`,[131] `exx` could also be leveraged to achieve a fully (overall) linear-scaling hybrid DFT approach. We note in passing that the MLWF-based EXX approach described herein also sets the stage for performing large-scale condensed-phase AIMD simulations based on quantum chemical (i.e., wave function theory) methodologies. Since a majority of the theoretical and algorithmic developments presented in this work are directly applicable to the iterative solution of the Hartree−Fock (HF) equations, this approach can be extended to enable a hierarchy of post-HF local electron correlation methods. Additional directions also include range-separated hybrids (RSH)[132−136] as well as fifth-rung xc functionals (e.g., MLWF-based GW approaches[137,138]).

The remainder of the paper is organized as follows. In section II, we describe the theoretical framework for performing CPMD simulations at the hybrid DFT level of theory within the MLWF representation. Section III contains details of our massively parallel algorithmic implementation in the open-source `QE` package. This is followed by a detailed systematic analysis of the accuracy and computational performance of the current implementation in section IV. The paper is then completed in section V, which provides some brief conclusions as well as the future outlook of AIMD simulations using hybrid DFT.

II. THEORY

In this section, we describe the theory behind our real-space MLWF-based framework for performing large-scale AIMD simulations of finite-gap condensed-phase systems at the hybrid DFT level of theory. We will focus the discussion below on the equations of motion underlying fixed-cell CPMD simulations in the *NVE* and *NVT* ensembles. Extension to constant-pressure CPMD simulations will be discussed in a forthcoming paper in this series. Although we limit our scope here to CPMD, which provides a computationally efficient localized orbital propagation scheme,[109−111] a cost-effective and competitive extension to BOMD has been achieved by our group and will also be addressed in another paper in this series.

II.A. Index Conventions.

We will utilize the following conventions for the various indices encountered in this work:

- *i*, *j*, *k*: indices for the N_o occupied orbitals (or MLWFs)
- *a*, *b*, *c*: indices corresponding to the Cartesian directions **x**, **y**, and **z**
- *I*, *J*, *K*: indices for the N_A ions
- *q*: index for points on the real-space grid
- *l*, *m*: indices for spherical harmonics

II.B. EXX-Based CPMD in the *NVE* Ensemble.

II.B.1. Equations of Motion. In CPMD simulations, fictitious dynamics are introduced on the N_o occupied KS orbitals $\{\phi_i(\mathbf{r})\}$ via artificial (fictitious) masses μ. Hence, CPMD simulations in the *NVE* ensemble are governed by the following equations of motion for the electronic and ionic degrees of freedom:[6]

$$\mu\ddot{\phi}_i(\mathbf{r}) = -\left(\frac{\delta E}{\delta \phi_i^*(\mathbf{r})}\right) + \sum_j \Lambda_{ij}\phi_j(\mathbf{r}) \tag{9}$$

$$M_I\ddot{\mathbf{R}}_I = -(\nabla_{\mathbf{R}_I}E) \tag{10}$$

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

in which Newton's dot notation is used to indicate time derivatives, E is the total ground-state DFT energy in eq 1, $-(\delta E/\delta\phi_i^*(\mathbf{r}))$ is the force acting on the ith occupied KS wave function, Λ_{ij} is a Lagrange multiplier enforcing orthonormality in $\{\phi_i(\mathbf{r})\}$, and $-\nabla_{\mathbf{R}_I}E$ is the force acting on the Ith ion (which is located at $\mathbf{R}_I$ with mass M_I). In an originless (e.g., planewave) basis, the equations of motion in eqs 9 and 10 will only depend on E_{xx} via the wave function forces, $-(\delta E/\delta\phi_i^*(\mathbf{r}))$, which are discussed in detail below.

II.B.2. EXX Contribution to the Wave Function Forces. In KS-DFT, E_{xc} is a functional of the electron density, which is given by $\rho(\mathbf{r}) = 2\sum_i \phi_i^*(\mathbf{r})\,\phi_i(\mathbf{r})$. As such, one can write the E_{xc} contribution to the (negative of the) wave function force for the ith KS orbital as the action of the so-called xc potential, $v_{xc}(\mathbf{r}) \equiv (\delta E_{xc}/\delta\rho(\mathbf{r}))$, on the orbital itself, i.e.,

$$\left(\frac{\delta E_{xc}}{\delta\phi_i^*(\mathbf{r})}\right) = \left(\frac{\delta E_{xc}}{\delta\rho(\mathbf{r})}\right)\left(\frac{\delta\rho(\mathbf{r})}{\delta\phi_i^*(\mathbf{r})}\right) = 2v_{xc}(\mathbf{r})\,\phi_i(\mathbf{r}) \tag{11}$$

Since the explicit functional dependence of E_{xx} (in E_{xc}^{hybrid}) on $\rho(\mathbf{r})$ is unknown, one needs special procedures such as the optimized effective potential (OEP) method[139] to derive the EXX contribution to the wave function forces within a strict KS-DFT scheme. In this work, we adopt a generalized KS-DFT scheme (i.e., by allowing for an orbital-dependent $v_{xc}(\mathbf{r})$), which requires significantly less computational effort and yields the same ground-state energies as the OEP formalism. In this approach (which is currently the standard practice in the field), we compute the corresponding orbital-dependent EXX wave function forces, $D_{xx}^i(\mathbf{r}) = -(\delta E_{xx}/\delta\phi_i^*(\mathbf{r}))$, by taking the functional derivative of E_{xx} in eq 6 with respect to $\phi_i^*(\mathbf{r})$, yielding

$$D_{xx}^i(\mathbf{r}) = \sum_j v_{ij}(\mathbf{r})\,\phi_j(\mathbf{r}) \equiv \sum_j D_{xx}^{ij}(\mathbf{r}) \tag{12}$$

To derive this expression, we have used eqs 4 and 5 for the orbital-product density and potential, $\rho_{ij}(\mathbf{r})$ and $v_{ij}(\mathbf{r})$, and defined $D_{xx}^{ij}(\mathbf{r})$ as the action of $v_{ij}(\mathbf{r})$ on $\phi_j(\mathbf{r})$. From eq 12, it is again clear that the evaluation of the orbital-product potential, $v_{ij}(\mathbf{r})$, is of central importance to the calculation of $D_{xx}^i(\mathbf{r})$.

II.C. Real-Space EXX Calculations: Linear Scaling via Orbital Localization. The efficient evaluation of $v_{ij}(\mathbf{r})$—which is a required ingredient for computing E_{xx} and $D_{xx}^i(\mathbf{r})$—is key to enabling large-scale condensed-phase AIMD simulations at the hybrid DFT level of theory. In this section, we will describe a linear-scaling EXX method that exploits the natural sparsity of the quantum mechanical exchange interaction in real space via the use of a localized (MLWF) representation of the occupied orbitals. Within this framework, $\tilde{v}_{ij}(\mathbf{r})$ (which is the MLWF analogue of $v_{ij}(\mathbf{r})$ in eq 5) only needs to be computed for *overlapping pairs* of MLWFs on a real-space domain that is independent of the system size, thereby paving the way to a linear-scaling EXX method in the condensed phase (see section III for algorithmic details). As such, the cornerstone of our method is the efficient real-space evaluation of $\tilde{v}_{ij}(\mathbf{r})$, which is accomplished herein via the solution of Poisson's equation on a system-size-independent real-space domain for each overlapping MLWF pair.

For a finite-gap condensed-phase system, the occupied KS orbitals (or bands) can be mapped via an orthogonal transformation onto a unique set of MLWFs (see eq 8) that are exponentially localized in real space[75,115−119] and have a significantly smaller support than the entire simulation cell. As such, the MLWF representation of the occupied space allows one to exploit the underlying sparsity in the quantum mechanical exchange interaction and provides a theoretical and computational framework for substantially reducing the computational scaling and cost associated with EXX-based approaches.

To see how MLWFs can be leveraged to attain a linear-scaling EXX algorithm, we first transform the canonical E_{xx} expression in eq 3 into the MLWF representation. Since $\phi_j(\mathbf{r}) = \sum_i (U^{-1})_{ji}\tilde{\phi}_i(\mathbf{r})$ (cf. eq 8), E_{xx} can be written as follows:

$$E_{xx} = -\sum_{ij}\sum_{\substack{kk'\\k''k'''}}\int d\mathbf{r}\int d\mathbf{r}'\,\frac{\tilde{\phi}_k(\mathbf{r})\,\tilde{\phi}_{k'}(\mathbf{r}')\,\tilde{\phi}_{k''}(\mathbf{r})\,\tilde{\phi}_{k'''}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} \times (U^{-1})_{ik}(U^{-1})_{jk'}(U^{-1})_{jk''}(U^{-1})_{ik'''} \tag{13}$$

Utilizing the fact that $\mathbf{U}\mathbf{U}^{\mathrm{T}} = \mathbf{U}\mathbf{U}^{-1} = \mathbf{I}$ for an orthogonal matrix, summation over i and j in this expression leads to $\sum_{ij}(U^{-1})_{ik}(U^{-1})_{jk'}(U^{-1})_{jk''}(U^{-1})_{ik'''} = \sum_i U_{ki}(U^{-1})_{ik'''}\sum_j U_{k'j}(U^{-1})_{jk''} = \delta_{kk'''}\delta_{k'k''}$, from which we see that

$$E_{xx} = -\sum_{ij}\int d\mathbf{r}\int d\mathbf{r}'\,\frac{\tilde{\phi}_i(\mathbf{r})\,\tilde{\phi}_j(\mathbf{r}')\,\tilde{\phi}_j(\mathbf{r})\,\tilde{\phi}_i(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} \tag{14}$$

upon dummy variable substitutions of $k \to i$ and $k' \to j$. This proof demonstrates that the expression for evaluating E_{xx} is invariant to the orthogonal transformation between the KS and MLWF representations. In fact, this invariance property of E_{xx} also holds for any arbitrary orbital representation $\{\psi_i(\mathbf{r})\}$ that is derived from an orthogonal rotation $\mathbf{U}'$ within the occupied KS subspace (i.e., $\psi_i(\mathbf{r}) = \sum_j U'_{ij}\phi_j(\mathbf{r})$). In analogy to eq 6, the MLWF expression for E_{xx} in eq 14 can also be written in the following compact form:

$$E_{xx} = -\sum_{ij}\int d\mathbf{r}\,\tilde{\rho}_{ij}(\mathbf{r})\,\tilde{v}_{ij}(\mathbf{r}) \tag{15}$$

in terms of the MLWF-product density,

$$\tilde{\rho}_{ij}(\mathbf{r}) \equiv \tilde{\phi}_i(\mathbf{r})\,\tilde{\phi}_j(\mathbf{r}) = \tilde{\rho}_{ji}(\mathbf{r}) \tag{16}$$

and the corresponding MLWF-product potential,

$$\tilde{v}_{ij}(\mathbf{r}) \equiv \int d\mathbf{r}'\,\frac{\tilde{\rho}_{ij}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} = \tilde{v}_{ji}(\mathbf{r}) \tag{17}$$

We note in passing that while E_{xx} is invariant to any orthogonal transformation, the values of $\tilde{\rho}_{ij}(\mathbf{r})$ and $\tilde{v}_{ij}(\mathbf{r})$—despite the fact that they have the same expressions as those given in eqs 4 and 5—do in fact depend on the employed representation. It is this freedom in the choice of the orthogonal transformation that allows one to select an appropriate localized orbital representation (e.g., MLWF) for exploiting the underlying sparsity in the EXX interaction. Throughout this work, we will therefore dress each of the MLWF-specific quantities with a tilde to distinguish them from their analogous expressions in the canonical KS representation.

Given the expression for E_{xx} in the MLWF representation (cf. eqs 14 and 15), the corresponding EXX contributions to the wave function forces that are required to propagate the CPMD equations of motion (eqs 9 and 10) can be derived

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

following the same procedure given above in section II.B.2. In this regard, the wave function force on the ith MLWF, $\tilde{D}^{i}_{xx}(\mathbf{r}) = -(\delta E_{xx}/\delta \tilde{\phi}_i^*(\mathbf{r}))$, can be obtained from eqs 14−17, yielding

$$\tilde{D}^{i}_{xx}(\mathbf{r}) = \sum_j \tilde{v}_{ij}(\mathbf{r})\,\tilde{\phi}_j(\mathbf{r}) \equiv \sum_j \tilde{D}^{ij}_{xx}(\mathbf{r}) \tag{18}$$

where $\tilde{D}^{ij}_{xx}(\mathbf{r})$ has been defined as the action of $\tilde{v}_{ij}(\mathbf{r})$ on $\tilde{\phi}_j(\mathbf{r})$. Here, $\tilde{D}^{i}_{xx}(\mathbf{r})$ and $\tilde{D}^{ij}_{xx}(\mathbf{r})$ also depend on the MLWF representation and therefore take on different values when compared to their KS analogues in eq 12. From eqs 15 and 18, it is again clear that the evaluation of the MLWF-product potential, $\tilde{v}_{ij}(\mathbf{r})$, is the cornerstone of our MLWF-based EXX approach.

With the expressions required for the evaluation of E_{xx} and $\tilde{D}^{i}_{xx}(\mathbf{r})$ in hand, we will now discuss in detail how MLWFs lead to a linear-scaling EXX algorithm by exploiting the underlying sparsity in the exchange interaction. Since the set of MLWFs are exponentially localized in real space and therefore have a significantly smaller support than the entire simulation cell, this allows us to exploit two levels of sparsity during the computational evaluation of all required EXX-related quantities. The first level of computational savings originates from the fact that a given MLWF, $\tilde{\phi}_i(\mathbf{r})$, will only appreciably overlap with a number, $\tilde{n}_i$, of neighboring MLWFs. For all other MLWFs, the product density, $\tilde{\rho}_{ij}(\mathbf{r}) = \tilde{\phi}_i(\mathbf{r})\,\tilde{\phi}_j(\mathbf{r})$ (and hence the corresponding product potential, $\tilde{v}_{ij}(\mathbf{r})$), will be vanishingly small. In these cases, the contributions to E_{xx} and $\tilde{D}^{i}_{xx}(\mathbf{r})$ are numerically zero, and this directly reduces the number of terms that are required in the summation over j in eqs 15 and 18. As such, the number of EXX pair interactions *per orbital* becomes independent of system size (assuming a fixed system density), which reduces the total number of orbital pairs, N_{pair}, from $O(N_o^2)$ to $O(N_o)$, i.e., $N_{pair} = N_o(N_o + 1)/2 \rightarrow \tilde{n}N_o$. In this last expression, $\tilde{n} = \max_i\{\tilde{n}_i\} < N_o$ is independent of the system size; hence, $\tilde{n}N_o$ represents an upper bound to the number of EXX pair interactions in our approach. Since the contributions from the omitted MLWF pairs to E_{xx} and $\tilde{D}^{i}_{xx}(\mathbf{r})$ are vanishingly small, this reduction in N_{pair} still allows for a numerically exact evaluation of all EXX-related quantities. Although this leads to significant computational savings, the overall scaling associated with evaluating these quantities is still formally quadratic as the real-space domain associated with the simulation cell, Ω, grows linearly with the size of the system.

To achieve linear scaling with system size, one can further exploit the fact that the set of exponentially localized MLWFs have a substantially smaller support than Ω. This allows us to employ real-space domains that are independent of system size and still maintain a numerically exact evaluation of E_{xx} and $\tilde{D}^{i}_{xx}(\mathbf{r})$. To harness this second level of computational savings, we follow the work of Gygi and co-workers[79,80,91] by defining an MLWF-orbital domain as $\Omega_i = \{\mathbf{r} \in \Omega | \, |\tilde{\phi}_i(\mathbf{r})| > 0\}$, in which ϵ is a small (but finite) positive threshold. For small enough values of ϵ, Ω_i will encompass the support of $\tilde{\phi}_i(\mathbf{r})$ in real space (see Figure 1). This domain is focused around the so-called MLWF center, $\tilde{\mathbf{C}}_i$, which is given by the expectation value (or first moment) of $\mathbf{r}$, i.e., $\tilde{\mathbf{C}}_i = \langle \tilde{\phi}_i | \mathbf{r} | \tilde{\phi}_i \rangle = \int d\mathbf{r}\, \mathbf{r}\, \tilde{\rho}_{ii}(\mathbf{r})$. In analogy, we also define an MLWF-product domain as $\Omega_{ij} \equiv \Omega_i \cap \Omega_j$, which (for sufficiently small ϵ values) encompasses the support of $\tilde{\rho}_{ij}(\mathbf{r})$ (see Figure 1). Since Ω_{ij} corresponds to the points in real space where $\tilde{\phi}_i(\mathbf{r})$ and $\tilde{\phi}_j(\mathbf{r})$ *are both non-negligible*, this domain is even more sparse than Ω_i or Ω_j. When $i = j$, one can straightforwardly compute the corresponding center of charge for $\tilde{\rho}_{ii}(\mathbf{r})$ as $\tilde{\mathbf{C}}_{ii} = \int d\mathbf{r}\, \mathbf{r}\, \tilde{\rho}_{ii}(\mathbf{r}) / \int d\mathbf{r}\, \tilde{\rho}_{ii}(r)$. Since $\tilde{\rho}_{ii}(\mathbf{r})$ integrates to unity, $\tilde{\mathbf{C}}_{ii} = \tilde{\mathbf{C}}_i$, which is simply the center of the ith MLWF given above. When $i \neq j$, $\tilde{\rho}_{ij}(\mathbf{r})$ now corresponds to a localized charge distribution with a *vanishing monopole* due to the orthogonality of the MLWFs; hence, the center of this charge distribution cannot be analogously defined as $\int d\mathbf{r}\, \mathbf{r}\, \tilde{\rho}_{ij}(\mathbf{r}) / \int d\mathbf{r}\, \tilde{\rho}_{ij}(r)$. As such, we utilize an analogue of the standard gauge in molecular quantum mechanics for an electrically neutral system, wherein the "center of charge" is taken as the position at which the nuclear (ionic) dipole moment vanishes. This allows us to define $\tilde{\mathbf{C}}_{ij} = \int d\mathbf{r}\, \mathbf{r}\, |\tilde{\rho}_{ij}(\mathbf{r})| \,/\, \int d\mathbf{r}\, |\tilde{\rho}_{ij}(\mathbf{r})|$ as the corresponding center for $\tilde{\rho}_{ij}(\mathbf{r})$. With all sectors of this charge distribution positive, $|\tilde{\rho}_{ij}(\mathbf{r})|$ now has a sizable monopole and a well-defined center of charge given by $\tilde{\mathbf{C}}_{ij}$. By construction, this choice of gauge recovers the correct center of charge when $i = j$, i.e., $\tilde{\mathbf{C}}_{ii} = \tilde{\mathbf{C}}_i$, and is therefore consistent with the expression used above for $\tilde{\rho}_{ii}(\mathbf{r})$.

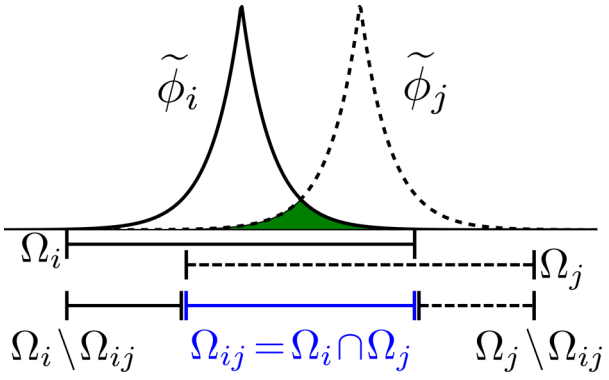


Figure 1. Graphical depiction of an overlapping pair of exponentially localized MLWFs, $\tilde{\phi}_i(\mathbf{r})$ (solid) and $\tilde{\phi}_j(\mathbf{r})$ (dashed). As such, the supports for $\tilde{\phi}_i(\mathbf{r})$ and $\tilde{\phi}_j(\mathbf{r})$ are compact and labeled by Ω_i and Ω_j, respectively. The support of the corresponding MLWF-product density, $\tilde{\rho}_{ij}(\mathbf{r}) = \tilde{\phi}_i(\mathbf{r})\,\tilde{\phi}_j(\mathbf{r})$ (green), is also compact and labeled by $\Omega_{ij} = \Omega_i \cap \Omega_j$. As described in the text, a numerically exact evaluation of the EXX contribution to the energy (E_{xx}) requires the solution to Poisson's equation for the near-field potential ($\tilde{v}_{ij}(\mathbf{r})$) on the Ω_{ij} domain (see eqs 19 and 21), while a numerically exact evaluation of the EXX contribution to the wave function forces ($\tilde{D}^{ij}_{xx}(\mathbf{r})$ and $\tilde{D}^{ji}_{xx}(\mathbf{r})$) requires a converged multipole expansion for the far-field potential ($\tilde{v}_{ij}(\mathbf{r})$) on the $\Omega_j \setminus \Omega_{ij}$ and $\Omega_i \setminus \Omega_{ij}$ domains (see eqs 20 and 22).

Within this framework, both Ω_i and Ω_{ij} are system-size-independent and substantially smaller than Ω. Furthermore, since Ω_{ij} is defined as the overlapping region between two exponentially decaying MLWFs, $\tilde{\phi}_i(\mathbf{r})$ and $\tilde{\phi}_j(\mathbf{r})$, the extent of this domain is smaller than both Ω_i and Ω_j, which holds true even when $i = j$. From eqs 15 and 16, one sees that a numerically exact evaluation of E_{xx} (neglecting self-consistency effects, vide infra) only requires summation over overlapping ij pairs (denoted by $\langle ij \rangle$) and spatial integration over Ω_{ij}, i.e.,

$$E_{xx} = -\sum_{\langle ij \rangle} \int_{\Omega_{ij}} d\mathbf{r}\, \tilde{\rho}_{ij}(\mathbf{r}) \tilde{v}_{ij}(\mathbf{r}) \tag{19}$$

In the same breath, eq 18 shows that a numerically exact evaluation of $\tilde{D}^{ij}_{xx}(\mathbf{r})$ only requires the action of $\tilde{v}_{ij}(\mathbf{r})$ over Ω_j, i.e.,

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

$$\tilde{D}_{xx}^{ij}(\mathbf{r}) = \tilde{v}_{ij}(\mathbf{r})\,\tilde{\phi}_j(\mathbf{r}) \qquad \mathbf{r} \in \Omega_j \tag{20}$$

This implies that one only needs $\tilde{v}_{ij}(\mathbf{r})$ on Ω_{ij} for a numerically exact evaluation of E_{xx}, and $\tilde{v}_{ij}(\mathbf{r})$ on Ω_j for a numerically exact evaluation of $\tilde{D}_{xx}^{ij}(\mathbf{r})$. As such, the evaluation of $\tilde{v}_{ij}(\mathbf{r})$ can also be restricted to system-size-independent real-space domains, despite the fact that this quantity is formally nonzero across Ω and asymptotically goes as $1/r$ for $i = j$ (due to the nonvanishing monopole associated with $\tilde{\rho}_{ii}(\mathbf{r})$) and $1/r^2$ (or higher order) for $i \neq j$ (due to the vanishing monopole associated with $\tilde{\rho}_{ij}(\mathbf{r})$). This leads to even further computational savings as $\tilde{v}_{ij}(\mathbf{r})$ can be obtained exactly by solving Poisson's equation (PE) over Ω_{ij} in the near field,

$$\nabla^2 \tilde{v}_{ij}(\mathbf{r}) = -4\pi\tilde{\rho}_{ij}(\mathbf{r}) \qquad \mathbf{r} \in \Omega_{ij} \tag{21}$$

subject to Dirichlet boundary conditions given by an appropriately converged multipole expansion (ME) of $\tilde{\rho}_{ij}(\mathbf{r})$ in the far field, i.e.,

$$\tilde{v}_{ij}(\mathbf{r}) = 4\pi \sum_{lm} \frac{Q_{lm}}{(2l+1)} \frac{Y_{lm}(\theta,\varphi)}{r^{l+1}} \qquad \mathbf{r} \notin \Omega_{ij} \tag{22}$$

In this expression, $\tilde{\mathbf{C}}_{ij}$ is taken as the origin, $\mathbf{r} = (r, \theta, \varphi)$ is given in spherical polar coordinates, $Y_{lm}(\theta,\varphi)$ are the spherical harmonics, and

$$Q_{lm} = \int_{\Omega_{ij}} \mathrm{d}\mathbf{r}\, Y_{lm}^{*}(\theta,\varphi)\, r^{l} \tilde{\rho}_{ij}(\mathbf{r}) \tag{23}$$

are the multipole moments corresponding to $\tilde{\rho}_{ij}(\mathbf{r})$. For typical systems, a ME with a maximum value of $l = 6$ is sufficiently converged.[76,78] We note in passing that the ME in eq 22 serves a dual purpose and will also be employed during the evaluation of $\tilde{D}_{xx}^{ij}(\mathbf{r})$, which requires $\tilde{v}_{ij}(\mathbf{r})$ on Ω_j. In other words, $\tilde{D}_{xx}^{ij}(\mathbf{r})$ is computed with $\tilde{v}_{ij}(\mathbf{r})$ on Ω_{ij} via the solution to the PE in eq 21, and $\tilde{v}_{ij}(\mathbf{r})$ on the $\Omega_j \backslash \Omega_{ij}$ domain, i.e., for all points in Ω_j that are not contained in Ω_{ij}, via the ME in eq 22.

This discussion again clearly highlights that an efficient real-space evaluation of $\tilde{v}_{ij}(\mathbf{r})$—on compact and system-size-independent domains—is the cornerstone of our linear-scaling MLWF-based EXX approach. In the next section, we will focus our discussion on the algorithmic implementation of this approach, which can be used to perform large-scale condensed-phase AIMD simulations with hybrid DFT.

III. IMPLEMENTATION AND ALGORITHMIC DETAILS

In this section, we describe the implementation of our linear-scaling MLWF-based EXX algorithm in the CP module of QE.[101,102] This algorithm has been implemented as a standalone module named exx, which has been integrated with the MLWF-enabled semilocal DFT routines in QE via a portable input/output interface (see flowchart in Figure 2). During each CPMD step, the main input required for exx includes the current set of MLWFs, $\{\tilde{\phi}_i(\mathbf{r})\}$, while the output produced by this module includes E_{xx} and $\{\tilde{D}_{xx}^{i}(\mathbf{r})\}$. As such, adaptation of the exx module to other periodic DFT codes should be straightforward, as long as the capability to produce MLWFs "on-the-fly" during CPMD simulations is available (vide infra). In fact, the current exx module only requires that the input orbitals are sufficiently local and form an orthonormal set, and can therefore accommodate (with appropriate modifications) other orbital localization schemes such as RSB[79,80] and SCDM.[81−83] To enable large-scale EXX-based AIMD using this approach, we employ a hybrid MPI and OpenMP parallelization scheme that allows us to differentially exploit both internode and intranode computational resources provided by massively parallel supercomputer architectures.

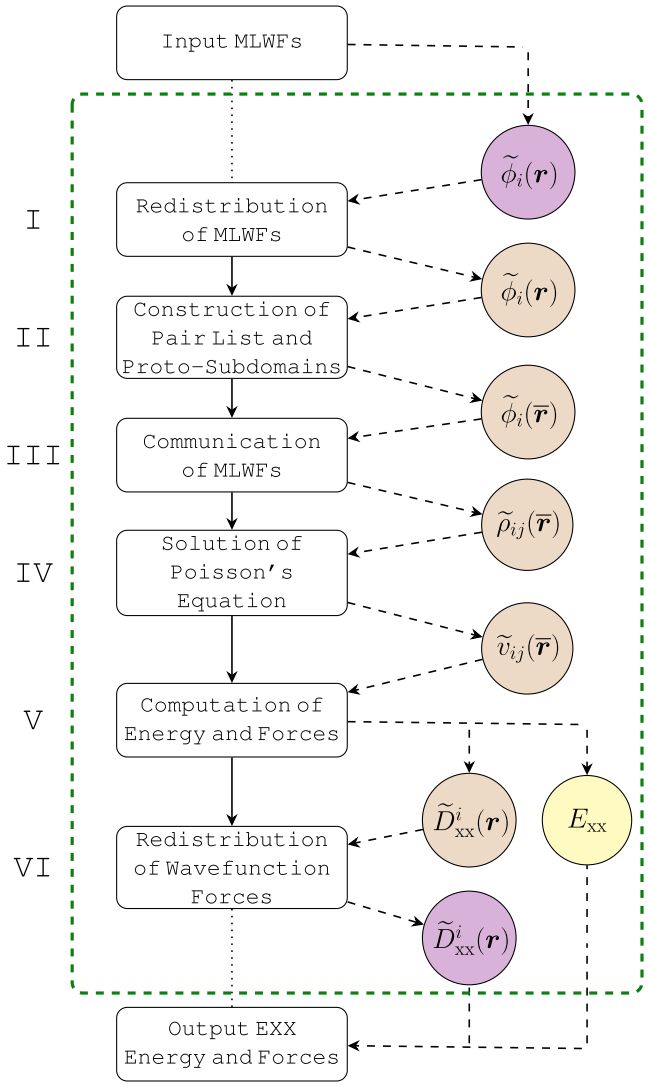


Figure 2. Flowchart of the exx module (dashed green box) in CP. As described in the main text, the input required by this module includes the current set of MLWFs, $\{\tilde{\phi}_i(\mathbf{r})\}$, at each CPMD step. The output produced by exx includes the EXX energy (E_{xx}) and the EXX contribution to the wave function forces ($\{\tilde{D}_{xx}^{i}(\mathbf{r})\}$). Purple (brown) circles indicate that a given quantity is represented according to the GRID (ORBITAL) data distribution scheme (see Figure 3 and section III.B), while the pale yellow circles represent data that are globally broadcasted. The $\overline{\mathbf{r}}$ notation indicates local (relative) Cartesian coordinates in a given subdomain. For a detailed description of each step in the exx module, see sections III.C.1−III.C.6.

III.A. MLWF-Based EXX-CPMD: Prerequisites. To start a CPMD simulation, one needs to reach the electronic ground state for a given initial configuration of the system via a SCF calculation. In the CP module of QE, the iterative solution of the nonlinear KS equations is accomplished by using either conjugate gradient (CG) or second-order damped dynamics (SODD) to minimize the fictitious kinetic energy associated with the electronic degrees of freedom (while keeping the ions

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

fixed).[140] During the SODD minimization, the proto-KS orbitals are evolved according to the following equations of motion (which are equivalent to eq 9 with an additional damping term):

$$\mu\ddot{\varphi}_i(\mathbf{r}) = D_i(\mathbf{r}) - 2\mu\gamma\dot{\varphi}_i(\mathbf{r}) \tag{24}$$

in which $\{\varphi_i(\mathbf{r})\}$ are the proto-KS orbitals during the SCF calculation, $D_i(\mathbf{r}) \equiv -(\delta E/\delta\varphi_i^*(\mathbf{r})) + \sum_j \Lambda_{ij}\varphi_j(\mathbf{r})$ is the force acting on the ith orbital, and γ is a damping parameter. To evolve the proto-KS orbitals, eq 24 can be integrated to yield:[140]

$$\varphi_i(\mathbf{r},\tau+\Delta\tau) = \frac{2}{1+\Gamma}\varphi_i(\mathbf{r},\tau) - \frac{1-\Gamma}{1+\Gamma}\varphi_i(\mathbf{r},\tau-\Delta\tau) + \frac{\Delta\tau^2}{1+\Gamma}\frac{D_i(\mathbf{r},\tau)}{\mu} \tag{25}$$

in which $\Delta\tau$ is the time step for the fictitious proto-KS dynamics and $\Gamma \equiv \gamma\Delta\tau$.[141] Upon convergence of the SODD procedure, $\{\varphi_i(\mathbf{r},\tau)\}$ becomes a set of ground-state KS orbitals, which is chosen as the initial condition for the AIMD simulation (i.e., $\{\phi_i(\mathbf{r},t=0)\}$). In doing so, cubic-scaling matrix operations such as diagonalization of the Fock (or effective Hamiltonian) matrix are completely sidestepped during the SCF procedure; as such, this approach does not require (nor produce) unoccupied/virtual states and provides a solid foundation upon which one can build a fully linear-scaling DFT (or HF) code base.

In fact, this CP-like approach to the SCF solution of the KS equations can be combined with the MLWF localization procedure by performing a nested SODD optimization of the Marzari–Vanderbilt[75,77] functional to incrementally localize the proto-KS orbitals between each SCF step.[109] In CP, this is accomplished by splitting eq 25 into an extrapolation step,

$$\chi_j(\mathbf{r},\tau+\Delta\tau) = \frac{2}{1+\Gamma}\tilde{\varphi}_j(\mathbf{r},\tau) - \frac{1-\Gamma}{1+\Gamma}\tilde{\varphi}_j(\mathbf{r},\tau-\Delta\tau) + \frac{\Delta\tau^2}{1+\Gamma}\frac{\tilde{D}_j(\mathbf{r},\tau)}{\mu} \tag{26}$$

followed by a localization step,

$$\tilde{\varphi}_i(\mathbf{r},\tau+\Delta\tau) = \sum_j U_{ij}(\tau+\Delta\tau)\,\chi_j(\mathbf{r},\tau+\Delta\tau) \tag{27}$$

In the extrapolation step, an intermediary set of orbitals, $\{\chi_j(\mathbf{r},\tau+\Delta\tau)\}$, is formed via SODD evolution of the proto-MLWF orbitals, $\{\tilde{\varphi}_j(\mathbf{r})\}$, according to $\tilde{D}_j(\mathbf{r},\tau) \equiv -(\delta E/\delta\tilde{\varphi}_j^*(\mathbf{r},\tau)) + \sum_k \tilde{\Lambda}_{jk}(\tau)\tilde{\varphi}_k(\mathbf{r},\tau)$, the force acting on the jth proto-MLWF orbital (which includes $\{\tilde{\Lambda}_{jk}(\tau)\}$, the set of Lagrange multipliers needed to preserve orthonormality). In the localization step, the proto-MLWFs, $\{\tilde{\varphi}_j(\mathbf{r},\tau+\Delta\tau)\}$, are incrementally localized via a unitary (orthogonal) transformation over $\{\chi_i(\mathbf{r},\tau+\Delta\tau)\}$, the intermediary set of orbitals obtained during the extrapolation step in eq 26. This unitary transformation is accomplished via $\mathbf{U}(\tau+\Delta\tau)$, a matrix that is generated from a nested SODD optimization[109] of the Marzari–Vanderbilt functional.[75,77] During the SCF procedure, it is computationally unfavorable (and *unnecessary*) to converge the nested SODD minimization for $\mathbf{U}$ at each step, as the current (unconverged) orbitals do not yet represent the ground electronic state. In this regard, only a few nested SODD steps (e.g., a maximum of 20 during the SCF procedure for liquid water) were used to incrementally localize the proto-MLWF orbitals. Upon convergence of this combined SCF and localization procedure, the set of proto-MLWF orbitals, $\{\tilde{\varphi}_i(\mathbf{r},\tau)\}$, becomes the set of ground-state MLWF orbitals, which can now be chosen as the initial condition for an MLWF-based AIMD simulation (i.e., $\{\tilde{\phi}_i(\mathbf{r},t=0)\}$). As such, this approach provides a cost-effective alternative to the standard *a posteriori* procedure of localizing the canonical (Bloch) occupied orbitals from a fully converged SCF calculation.

At the hybrid DFT level, we adopt this CP-like approach to incrementally localize the occupied orbitals *during* the EXX-based SCF procedure, thereby avoiding a preliminary EXX calculation in reciprocal space. Since the incrementally localized proto-MLWF orbitals are not equivalent to the final set of MLWFs at a given SCF step, the orbital-dependent EXX contributions to $v_{xc}(\mathbf{r})$ are approximately evaluated via eq 20; however, the resulting errors are inconsequential as the incremental refinement of the localized orbitals (and therefore $\tilde{D}_{xx}^i(\mathbf{r})$) at each step leads to the desired set of MLWFs upon SCF convergence. Since our approach is based on an incremental "on-the-fly" refinement of the proto-MLWF orbitals during the SCF procedure, it is therefore unsuitable for standard Fock matrix diagonalization routines, in which global rotations between the occupied and virtual orbitals during each diagonalization step would lead to marked delocalization of the occupied states. Such delocalization would require substantial effort (essentially from scratch) to relocalize the orbitals after each diagonalization step and would thereby nullify the computational savings obtained from a sparse real-space evaluation of all EXX-related quantities.

In practice, EXX-based SCF calculations in CP take advantage of the incremental nature of the aforementioned MLWF refinement process by starting with a relatively inexpensive semilocal xc functional (e.g., PBE[12] for PBE0[58,142]), which stabilizes $\rho(\mathbf{r})$ and initiates the orbital localization procedure. Once the semilocal DFT iterations reach ≈10 times the target SCF convergence threshold, the orbitals are typically quite localized and closely resemble the final set of MLWFs corresponding to the chosen semilocal xc functional. At this point, the exx module is activated to perform the remaining steps required to reach SCF convergence at the hybrid DFT level, upon which one obtains the final set of MLWFs corresponding to the chosen hybrid xc functional. For all systems tested, this approach has a significantly reduced computational cost when compared to the alternative procedure of (i) performing a standard (canonical) PBE calculation, (ii) localizing the converged PBE orbitals from scratch, and (iii) using these localized PBE orbitals as input into the incremental localization procedure described above to perform a PBE0 SCF calculation to convergence. For insulating systems with small band gaps (e.g., InN), one should exercise caution when using GGA orbitals for the initial guess in this procedure, as the GGA xc functional might incorrectly predict a metallic system with the wrong band-index ordering.[120] With that caveat in mind, we also note that the computational cost associated with this initial SCF procedure is completely negligible when compared to the overall CPMD simulation, which is the focus of this work. In future work, we hope to further optimize this incremental SCF procedure to enable high-throughput hybrid DFT-based single-point energy evaluations on large-scale condensed-phase systems (as well as linear-scaling EXX-based BOMD).

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757–3785

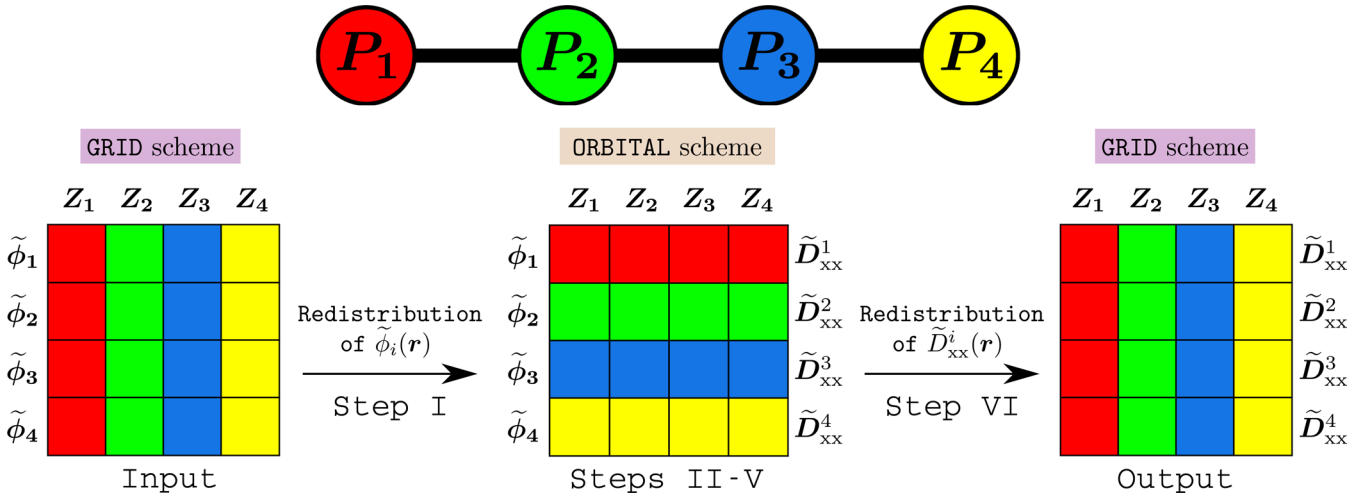


Figure 3. Schematic illustration of the GRID and ORBITAL data distribution schemes in QE. For simplicity, we consider a system consisting of a single water molecule with $N_o = 4$ MLWFs ($\tilde{\phi}_i(\mathbf{r})$), a simulation cell consisting of a real-space simple-cubic grid that has been partitioned into $N_{slab} = 4$ slabs along the z-direction (Z_i), and a pool of $N_{proc} = 4$ MPI processes (P_i). As depicted at the top of the figure, each of these MPI processes (and the corresponding data it holds in local memory) is assigned a color: P_1 (red), P_2 (green), P_3 (blue), and P_4 (yellow). As input into the exx module, the $\tilde{\phi}_i(\mathbf{r})$ are provided in the GRID scheme, in which a given MPI process, P_i, holds the data corresponding to *all* N_o MLWFs on *one* slab, Z_i, of the real-space grid. During Step I of the exx module (section III.C.1), the $\tilde{\phi}_i(\mathbf{r})$ are redistributed according to the ORBITAL scheme, in which a given MPI process, P_i, holds the data corresponding to only *one* MLWF, $\tilde{\phi}_i(\mathbf{r})$, across *all* N_{slab} slabs of the real-space grid. As described in sections III.C.2−III.C.5, Steps II−V involve selective communication of the $\tilde{\phi}_i(\mathbf{r})$ between MPI processes and computation of all EXX-related quantities (E_{xx} and $\{\tilde{D}^i_{xx}(\mathbf{r})\}$). At the end of Step V, the $\{\tilde{D}^i_{xx}(\mathbf{r})\}$ are stored according to the ORBITAL scheme and are redistributed back to the GRID scheme during Step VI (section III.C.6), the final step of the exx module.

In analogy to the incremental localization procedure described above for EXX-based SCF calculations (cf. eqs 26 and 27), we have also introduced this nested SODD determination of $\mathbf{U}$ into the propagation of the CPMD equations of motion. This is accomplished by a CPMD propagation step,

$$\chi_j(\mathbf{r},t+\Delta t) = 2\tilde{\phi}_j(\mathbf{r},t) - \tilde{\phi}_j(\mathbf{r},t-\Delta t) + \frac{\Delta t^2}{\mu}\tilde{D}_j(\mathbf{r},t) \tag{28}$$

in which an intermediary set of orbitals, $\{\chi_j(\mathbf{r},t+\Delta t)\}$, is formed via CPMD evolution (with time step Δt) of the MLWF orbitals, $\{\tilde{\phi}_j(\mathbf{r})\}$. During the CPMD propagation step, these intermediary orbitals become slightly more delocalized than the set of MLWFs (yet remain on the ground-state potential energy surface) and are therefore refined by a subsequent localization step,

$$\tilde{\phi}_i(\mathbf{r},t+\Delta t) = \sum_j U_{ij}(t+\Delta t)\,\chi_j(\mathbf{r},t+\Delta t) \tag{29}$$

in which the unitary transformation $\mathbf{U}$ is generated by *tightly converging* the nested SODD optimization of the Marzari−Vanderbilt functional. By doing so after every CPMD propagation step, we ensure that the resulting $\{\tilde{\phi}_i(\mathbf{r},t)\}$ and $\{\tilde{D}_i(\mathbf{r},t)\}$ are indeed the MLWFs and the forces acting on them. We note in passing that the need to perform the additional localization step in eq 29 reflects the lack of gauge invariance in the electronic CPMD equations of motion within the MLWF representation.[110,111] Nevertheless, the intermediary orbitals generated by eq 28 are typically good approximations to the MLWFs and thereby provide a rather good initial guess to the SODD localization procedure.[109] As a result, the localization procedure typically converges with a small number of nested SODD iterations (e.g., three to four iterations for the liquid water systems in section IV.B.1), which results in minimal computational overhead when compared to the cost of the EXX calculation. Moving forward, this incremental localization scheme could be avoided using the field-theoretic approach proposed by Tuckerman and co-workers,[111] which introduces additional fictitious dynamics on a set of gauge fields to enable "on-the-fly" propagation of the MLWF transformation ($\mathbf{U}$) matrix.

III.B. MLWF-Based EXX-CPMD: Data Distribution Schemes. As mentioned above, we employ a hybrid MPI/OpenMP parallelization scheme to enable large-scale EXX-based AIMD on massively parallel supercomputer architectures containing thousands of nodes. Our algorithm, which is described in section III.C below, is primarily based upon the MPI distributed-memory paradigm, which requires specific data distribution schemes to minimize communication overhead and maximize computational efficiency. During a GGA-based CPMD simulation in QE, the orbitals, charge density, and potential are constantly transformed between real and reciprocal space via the fwdFFT and invFFT operations. With all real-space quantities numerically represented on a grid (mesh) that is discretized along the corresponding lattice vectors, QE employs the GRID data distribution scheme to scatter these quantities across N_{proc} MPI processes (ranks). In the GRID data distribution scheme (see Figure 3), the real-space grid is partitioned into N_{slab} slabs along the z axis. Assuming $N_{proc} = N_{slab}$ for simplicity, each MPI process will hold the data corresponding to *all* distributed real-space quantities on a *single* slab of the real-space grid. In doing so, this data distribution scheme facilitates efficient parallel FFT by dividing the 3D FFT into a set of 2D FFTs (each of which can be executed by a given MPI process within a given slab) followed by a 1D FFT along the direction of the slab partition.

As depicted in Figure 2, the input to the exx module in QE includes the current set of MLWFs, $\{\tilde{\phi}_i(\mathbf{r})\}$, at each CPMD step. These MLWFs are distributed across MPI processes

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

according to the GRID data distribution scheme, in which a given process holds the data corresponding to all MLWFs on a given slab of the real-space grid. Although the GRID scheme is convenient for efficient parallel FFT, this data distribution model is far from ideal for an efficient massively parallel implementation of our MLWF-based EXX approach. As such, we have introduced an alternative ORBITAL data distribution scheme in QE (see Figure 3), in which a given MPI process now holds quantities like $\tilde{\phi}_i(\mathbf{r})$ and $\tilde{D}^i_{xx}(\mathbf{r})$ for a *single* MLWF across the *entire* real-space grid (for the case in which $N_{proc} = N_o$; for other cases, see the discussion below in section III.C.1). The details behind the transformation between the GRID and ORBITAL data distribution schemes are provided below in sections III.C.1 and III.C.6.

The ORBITAL data distribution scheme is particularly suited for our real-space MLWF-based EXX algorithm, since this approach is centered around orbital sparsity and the efficient evaluation of $\tilde{v}_{ij}(\mathbf{r})$. For one, the ORBITAL scheme allows us to utilize a significantly larger number of MPI processes ($N_{proc} \gg N_{slab}$), as the number of MLWFs or overlapping MLWF pairs (both of which grow linearly with system size) quickly exceeds N_{slab} (which grows with the cubic root of the system size). The ORBITAL scheme also allows us to exploit intranode parallelization with N_{thread} OpenMP threads during the most computationally intensive steps in our algorithm, e.g., solving the PE to obtain $\tilde{v}_{ij}(\mathbf{r})$ (see section III.C.4). As a result, this hybrid MPI/OpenMP parallelization scheme not only provides us with access to even more computational resources during EXX-based simulations but also allows us to sidestep the prohibitively large data communication overhead associated with an MPI-based solution to the PE.

III.C. MLWF-Based EXX-CPMD: Algorithm. In this section, we provide a detailed description for each of the steps inside the exx module in QE. Our discussion will follow the flowchart depicted in Figure 2, in which the current set of MLWFs in real space ($\{\tilde{\phi}_i(\mathbf{r})\}$, distributed according to the GRID scheme) are provided as input into the exx module. Subsequent output of exx includes the EXX energy (E_{xx}) as well as the EXX contribution to the wave function forces ($\{\tilde{D}^i_{xx}(\mathbf{r})\}$, which are again distributed according to the GRID scheme). This preserves compatibility with the rest of CP, and allows for a modular exx codebase.

III.C.1. Step I: Redistribution of MLWFs. In the exx algorithm, the assignment of MLWFs to a given MPI process is based on $\zeta \equiv N_{proc}/N_o$, i.e., the ratio of available MPI processes to the number of MLWFs. When $\zeta = 1$, there is one MPI process per MLWF, and each process, P_i, is assigned a unique MLWF, $\tilde{\phi}_i$. With limited computational resources ($N_{proc} < N_o$), $\zeta < 1$ and multiple MLWFs are assigned to each process; as such, a balanced distribution of MLWFs across MPI processes is only possible when N_{proc} is a divisor of N_o. In the strong-scaling limit, our exx algorithm allows ζ to take on integer values greater than 1, in which a given MLWF is assigned to multiple MPI processes. Unless otherwise specified, we will assume that $\zeta = 1$ throughout the remainder of section III.C.

Given the current set of MLWFs in real space, $\{\tilde{\phi}_i(\mathbf{r})\}$, which are distributed among the available N_{proc} MPI processes according to the GRID scheme, the first step in the exx module is the forward redistribution of these quantities into the ORBITAL data distribution scheme. For this purpose, each MPI process collects an assigned $\tilde{\phi}_i(\mathbf{r})$ across the entire real-space grid via an ALL-TO-ALL internode communication step, as shown in Figure 3. This ALL-TO-ALL communication is performed twice per CPMD step: once here in the forward redistribution of $\{\tilde{\phi}_i(\mathbf{r})\}$ from the GRID to the ORBITAL scheme, and once in Step VI in the inverse redistribution of $\{\tilde{D}^i_{xx}(\mathbf{r})\}$ from the ORBITAL to the GRID scheme (see section III.C.6). As discussed in section IV, this communication overhead represents a substantial fraction of the total cost associated with the exx module and can be significantly reduced by a more sophisticated communication scheme over select subsets of the MPI process pool, i.e., those containing the regions of real space containing the relevant MLWF-orbital domain, Ω_i (see section II.C). This algorithmic improvement is currently underway and will be described in future work.

III.C.2. Step II: Construction of Pair List and Proto-Subdomains. With the MLWFs distributed among MPI processes according to the ORBITAL scheme, we now explain how the exx module exploits the sparsity of the MLWFs and utilizes system-size-independent subdomains of Ω during the computation of all EXX-related quantities. To accomplish this goal, we will first describe the construction of the so-called unique MLWF-pair list, $\mathcal{L}$, which not only contains the relevant set of overlapping MLWF pairs but also determines how the computational workload associated with these pairs is distributed among the pool of available MPI processes. This is followed by a detailed description of the set of "proto-subdomains" employed in the exx module, which represent computationally efficient alternatives to the formal Ω_i and Ω_{ij} subdomains introduced above in section II.C.

Construction of the MLWF-Pair List. To exploit the first level of computational savings, which originates from the fact that MLWFs are exponentially localized and only overlap with a limited number of neighbors, two MLWFs, $\tilde{\phi}_i(\mathbf{r})$ and $\tilde{\phi}_j(\mathbf{r})$, are considered an overlapping pair if $|\tilde{\mathbf{C}}_i - \tilde{\mathbf{C}}_j| < R_{pair}$. A judicious choice for R_{pair} is required for accurately calculating all EXX-related quantities, and an analysis of the convergence of E_{xx} with respect to R_{pair} will be provided in section IV.A.1.

At the current point in the algorithm, each $\tilde{\phi}_i(\mathbf{r})$ is stored according to the ORBITAL data distribution scheme on one (or more) MPI processes (depending on the value of ζ employed during runtime; see section III.C.1). For simplicity, we will discuss the $\zeta = 1$ case first, in which there is only one MPI process per MLWF, and each process, P_i, is assigned a unique MLWF, $\tilde{\phi}_i(\mathbf{r})$. As such, P_i lacks direct access to $\tilde{\phi}_j(\mathbf{r})$ for $j \neq i$, which is required for the construction of $\tilde{\rho}_{ij}(\mathbf{r})$ and the subsequent computation of $\tilde{v}_{ij}(\mathbf{r})$ for evaluating the $\langle ij \rangle$-pair contribution to E_{xx}, $\tilde{D}^{ij}_{xx}(\mathbf{r})$, and $\tilde{D}^{ji}_{xx}(\mathbf{r})$. Although $\tilde{\rho}_{ii}(\mathbf{r})$ can be constructed *locally* on P_i to evaluate the $\langle ii \rangle$-pair (self-pair) contribution to each of these quantities, all of the other $\langle ij \rangle$-pair contributions will require communication between MPI processes.

To design an MLWF-based EXX algorithm that achieves a minimal time to solution while efficiently utilizing all parallel computational resources, one needs to (i) minimize the total computational workload, (ii) minimize the number of interprocess communication events, and (iii) maintain a balanced workload among the pool of available MPI processes. To accomplish this goal, we now describe the procedure employed in the exx module to construct the so-called unique MLWF-pair list, $\mathcal{L}$, which defines the computation and

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

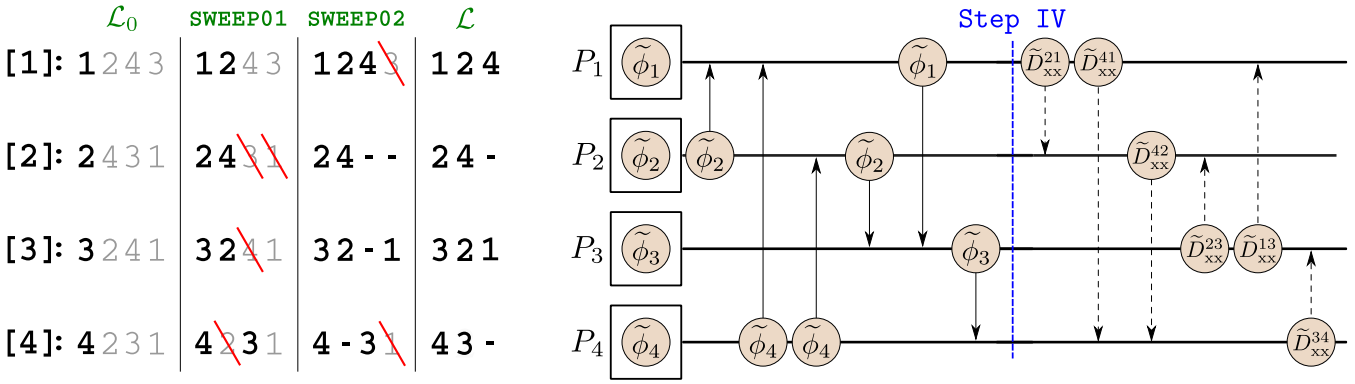


Figure 4. Graphical depiction of the unique MLWF-pair list construction process and corresponding MLWF communication scheme in the `exx` module. For simplicity, we will again consider a single water molecule with $N_o = 4$ MLWFs ($\tilde{\phi}_i$) and a pool of $N_{proc} = 4$ `MPI` processes (P_i), i.e., $\zeta = 1$. Starting with the nonunique MLWF-pair list, $\mathcal{L}_0$, which contains all possible permutations of overlapping MLWF pairs, the step-by-step procedure employed to transform $\mathcal{L}_0$ into the unique MLWF-pair list, $\mathcal{L}$, is depicted in the left panel. Since all of the MLWFs are mutually overlapping in a single water molecule, $\mathcal{L}_0[i]$ contains $\{j\}$, the indices corresponding to all MLWFs (including $j = i$), which have been sorted according to $|\tilde{\mathbf{C}}_i - \tilde{\mathbf{C}}_j|$. During the process of reducing $\mathcal{L}_0$ to $\mathcal{L}$, i is always selected (bold black font) to remain in $\mathcal{L}_0[i]$, while the remaining nonunique indices are shown in gray. In the first sweep (SWEEP01) of Algorithm 1, the next element $j \in \mathcal{L}_0[i]$ is selected to remain in $\mathcal{L}_0[i]$, while the now redundant index i is removed (red slash) from $\mathcal{L}_0[j]$. During the first sweep in this example, the indices 2, 4, 2, 3 were selected to remain in $\mathcal{L}_0[1]$, $\mathcal{L}_0[2]$, $\mathcal{L}_0[3]$, $\mathcal{L}_0[4]$, while the corresponding redundant indices, 1, 2, 3, 4, were removed from $\mathcal{L}_0[2]$, $\mathcal{L}_0[4]$, $\mathcal{L}_0[2]$, $\mathcal{L}_0[3]$. This process is repeated until all of the MLWF-pair redundancy is removed from $\mathcal{L}_0$, upon which one is left with the final $\mathcal{L}$. For a single water molecule, only two sweeps are required to reach this stage; at that point, all of the unique MLWF pairs have been assigned to a given $\mathcal{L}[i]$, and no redundant indices remain. With each MLWF stored according to the `ORBITAL` data distribution scheme (in which $\tilde{\phi}_i$ is assigned to P_i), the final $\mathcal{L}$ determines how the computational workload will be distributed among the pool of available `MPI` processes. Even in this simple example, there exists a mismatch in the number of pairs assigned to each process, with three MLWF pairs assigned to P_1 and P_3, and only two MLWF pairs assigned to P_2 and P_4. Such discrepancies are expected (even for the most homogeneous systems) and lead to an imbalance in the computational workload. By virtue of the `ORBITAL` data distribution scheme, $\mathcal{L}$ also determines the corresponding MLWF communication protocol, which is depicted in the right panel for the single water molecule. After a given $\tilde{\phi}_j$ is directly communicated to P_i, this `MPI` process forms $\tilde{\rho}_{ij}$, solves the corresponding PE for $\tilde{v}_{ij}$ (depicted by the blue dashed line), and computes the $\langle ij\rangle$-pair contribution to E_{xx}, $\tilde{D}^{ij}_{xx}$, and $\tilde{D}^{ji}_{xx}$ (which is sent back to P_j). As an example, consider $\mathcal{L}[2]$, which contains two indices (2, 4). Since P_2 already holds $\tilde{\phi}_2$, $\tilde{\rho}_{22}$ (required for computing $\tilde{v}_{22}$) can be constructed locally without the need for any interprocess communication. To construct $\tilde{\rho}_{24}$ (and hence $\tilde{v}_{24}$), $\tilde{\phi}_4$ is sent from P_4 to P_2 (solid arrow). After the corresponding $\tilde{D}^{42}_{xx}$ is formed, it is shipped back to P_4 (dashed arrow). Besides this straightforward dependency between receiving an MLWF and computing the corresponding EXX contributions to the wave function forces, the communication in the `exx` module has no discernible time axis in the figure above.

communication protocol in our algorithm. To construct $\mathcal{L}$, the indices corresponding to *all* overlapping MLWF pairs (as determined by the aforementioned criteria based on R_{pair}) are first assembled into the nonunique MLWF-pair list, $\mathcal{L}_0$, which contains all possible permutations ij and ji of these overlapping MLWF pairs. Since the $\langle ij\rangle$- and $\langle ji\rangle$-pair contributions to E_{xx} are equivalent (cf. eqs 15−17), it is clear that $\mathcal{L}_0$ is redundant and contains twice as many pairs as needed.

Before discussing the procedure used to determine $\mathcal{L}$, we first demonstrate that exploiting such redundancy within the more parallelizable `ORBITAL` data distribution scheme leads to the requirement for two interprocess communication events per unique MLWF pair. To see this more clearly, one only needs to consider a minimalistic system that contains a single $\langle ij\rangle$ pair of overlapping MLWFs. Throughout this example, spatial arguments will be suppressed (e.g., $\tilde{\phi}_i$ will be used instead of $\tilde{\phi}_i(\mathbf{r})$), since all computation and communication events will be performed using system-size-independent subdomains (vide infra). With $\tilde{\phi}_i$ located on P_i and $\tilde{\phi}_j$ located on P_j, we will first consider what happens if the inherent pair redundancy is *not* exploited. In this case, $\tilde{\phi}_i$ ($\tilde{\phi}_j$) is first communicated to P_j (P_i) for a total of two interprocess communication events. At this point, each `MPI` process constructs the corresponding MLWF-product density, $\tilde{\rho}_{ij} = \tilde{\rho}_{ji}$, and proceeds to compute $\tilde{v}_{ij} = \tilde{v}_{ji}$ by solving two equivalent PEs. Since the solution of the PE is the dominant computational step in our EXX algorithm, this will count for a total of two computation events. With $\tilde{v}_{ij} = \tilde{v}_{ji}$ available on both P_i and P_j, each process is now in a position to compute the $\langle ij\rangle$- and $\langle ji\rangle$-pair contributions to E_{xx} via eq 15, as well as $\tilde{D}^{ij}_{xx} = \tilde{v}_{ij}\tilde{\phi}_j$ and $\tilde{D}^{ji}_{xx} = \tilde{v}_{ji}\tilde{\phi}_i$ via eq 18. As depicted in Figure 2, the $\{\tilde{D}^{i}_{xx}\}$ are needed in the `ORBITAL` data distribution scheme before these quantities are finally redistributed back to the `GRID` scheme to ensure compatibility with the other modules in `QE` (see Figure 3). As such, the local evaluation of $\tilde{D}^{ij}_{xx}$ on P_i and $\tilde{D}^{ji}_{xx}$ on P_j directly provides these quantities in the requisite `ORBITAL` data distribution scheme without the need for any additional communication. Hence, the total cost per unique MLWF pair amounts to two units of communication followed by two units of computation, when pair redundancy is not exploited.

Since the removal of all MLWF-pair redundancy is crucial for minimizing the total number of computational events (and hence the overall time to solution), we now consider the case where this inherent pair redundancy is exploited. In this case, only $\tilde{\phi}_j$ would be sent to P_i with an associated cost of one unit of communication, and P_i will therefore be solely responsible for computing all EXX-related quantities. Since the $\langle ij\rangle$- and $\langle ji\rangle$-pair contributions to E_{xx} are equivalent, these quantities can be computed on P_i via a single computation event (i.e., the solution to the corresponding PE), and then sent to any other process with minimal communication (i.e., one double-precision number for each pair contribution to E_{xx}). Although

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

$\tilde{D}_{xx}^{ij} \neq \tilde{D}_{xx}^{ji}$, this also poses no problem as P_i has direct access to $\tilde{\phi}_i$ and $\tilde{\phi}_j$, and hence both $\tilde{D}_{xx}^{ij}$ and $\tilde{D}_{xx}^{ji}$ can be computed *locally*. With the requirement that the $\{\tilde{D}_{xx}^{i}\}$ are stored in the ORBITAL data distribution scheme, this will incur an additional communication event as $\tilde{D}_{xx}^{ji}$ is shipped back to P_j. Hence, exploiting the inherent MLWF-pair redundancy reduces the computational workload by half (as expected), but it does not change the requirement for two communication events per unique MLWF pair.

During this nonredundant evaluation of the $\langle ij \rangle$- and $\langle ji \rangle$-pair contributions, the fact that P_j was idle while P_i performed all of the required computations creates an imbalance in the computational workload assigned to each MPI process. With the freedom to assign the computational workload associated with the $\langle ij \rangle$ pair to either P_i or P_j, the exx module is now tasked with determining how the total computational workload will be distributed among the pool of available MPI processes. Armed with knowledge of the total number of nonunique MLWF pairs in the system (via $\mathcal{L}_0$) as well as the use of system-size-independent subdomains to regularize the computational cost associated with the solution to each PE (vide infra), the process for doing so involves a static load-balancing algorithm that seeks to minimize the overall time to solution by reducing the imbalances present in the computational workload, and hence the number of idle processes. Although it is certainly possible in the current version of the algorithm, we chose not to involve a third process, P_k, in the evaluation of the $\langle ij \rangle$-pair contribution, as this would introduce two additional communication events, i.e., $\tilde{\phi}_i$ to P_k and $\tilde{D}_{xx}^{ij}$ back to P_i (in addition to $\tilde{\phi}_j$ to P_k and $\tilde{D}_{xx}^{ji}$ back to P_j). In this regard, the local computation of $\tilde{D}_{xx}^{ij}$ on P_i not only avoids additional unnecessary communication events but also allows for reduced storage requirements as this quantity can be cumulatively incremented (over multiple j) within a single array.

This static load-balancing algorithm can be represented by the so-called unique MLWF-pair list, $\mathcal{L}$, the construction of which is described in the left panel of Figure 4 (for the illustrative case of a single water molecule) as well as in Algorithm 1 (for the general case). We start with the $\mathcal{L}_0$ array, which contains all possible permutations of overlapping MLWF pairs, i.e., $\mathcal{L}_0[i]$ (the ith row of $\mathcal{L}_0$) is populated with a list of indices, $\{j\}$, corresponding to all $\tilde{\phi}_j$ that overlap with $\tilde{\phi}_i$. For each i, the indices $j \in \mathcal{L}_0[i]$ are sorted into ascending order on the basis of their vicinity to $\tilde{\phi}_i$ via $|\tilde{\mathbf{C}}_i - \tilde{\mathbf{C}}_j|$. By construction, each $\mathcal{L}_0[i]$ also contains i (self-pair) and will retain this nonredundant index throughout the refinement of $\mathcal{L}_0$ to $\mathcal{L}$ in Algorithm 1. While there are still redundant pairs in $\mathcal{L}_0$, this algorithm will consecutively sweep over MLWFs to locate redundant pairs such as $\langle ij \rangle$ and $\langle ji \rangle$; in our approach, this is tantamount to finding both $j \in \mathcal{L}_0[i]$ and $i \in \mathcal{L}_0[j]$. Once located, the algorithm eliminates this redundancy from $\mathcal{L}_0$ by removing the index i from $\mathcal{L}_0[j]$. At the end of these sweeps, all of the redundancies in $\mathcal{L}_0$ are removed, and we are left with $\mathcal{L}$, the unique MLWF-pair list. This list contains the minimum number of computational tasks required to evaluate all EXX-related quantities and dictates how this computational workload will be distributed among the pool of available MPI processes. By virtue of the ORBITAL data distribution scheme, $\mathcal{L}$ also encodes the communication protocol that will be followed throughout the remainder of the exx module (see section III.C.3). With $\zeta = 1$, this amounts to sending $\tilde{\phi}_j \rightarrow P_i$ and $\tilde{D}_{xx}^{ji} \rightarrow P_j$ for each unique $\langle ij \rangle$ pair, as depicted in the right panel of Figure 4.

Algorithm 1 Refinement of $\mathcal{L}_0$ to $\mathcal{L}$

```
any_removal ← TRUE
while any_removal do
    any_removal ← FALSE
    for i = 1, N_o do
        for j ≠ i ∈ L_0[i] do
            if i ∈ L_0[j] then
                L_0[j] ← L_0[j] \ {i}
                any_removal ← TRUE
                break
            end if
        end for
    end for
end while
L ← L_0
```

By construction, static load-balancing algorithms (such as Algorithm 1) yield fairly well-balanced workload distributions by mitigating potential imbalances during the refinement of $\mathcal{L}_0$ to $\mathcal{L}$. Here, we note that the distance-based sorting of the indices in each row of $\mathcal{L}_0$ is crucial for avoiding severe workload imbalances due to sequential index ordering. In this regard, an equally effective load-balancing algorithm would be possible by performing random sweeps over row indices (and completely avoiding the initial distance-based sorting procedure). Since the number of overlapping pairs per MLWF will often change throughout an AIMD simulation, this static load-balancing algorithm is performed during each MD step in an attempt to determine an optimal workload balance. For a detailed discussion regarding the performance of this static load-balancing algorithm during CPMD simulations of liquid water, as well as future possible improvements of this approach, see section IV.B.1.

When computational resources are limited, the exx module can utilize fewer MPI processes during runtime (i.e., $\zeta < 1$). In this case, multiple MLWFs are contiguously assigned to each process, and a balanced distribution of the workload (within the framework defined by Algorithm 1) is more likely when N_{proc} is a divisor of N_o; as such, this is the current recommended setting whenever applicable (see section III.C.1). In the isolated water molecule example in Figure 4, the use of $\zeta = 1/2$ would start with P_1 holding $\tilde{\phi}_1$ and $\tilde{\phi}_2$, and P_2 holding $\tilde{\phi}_3$ and $\tilde{\phi}_4$. After running Algorithm 1 to generate $\mathcal{L}$, the workload associated with a given MLWF is mapped onto the process holding this orbital. This results in five units of computation assigned to each MPI process: two self-pairs, one local pair (in which both MLWFs are held on the same process), and two nonlocal pairs (in which one of the MLWFs is held on a different process); e.g., P_1 would be responsible for $\langle ii \rangle = \langle 11 \rangle$ and $\langle ii \rangle = \langle 22 \rangle$ (two self-pairs), $\langle ij \rangle = \langle 12 \rangle$ (one local pair), and $\langle ij \rangle = \langle 14 \rangle$ and $\langle ij \rangle = \langle 24 \rangle$ (two nonlocal pairs). In this case, the workload is optimally balanced and the maximum number of computation events per process is only 5/3 times (instead of 2 times) larger than $\zeta = 1$. This allows for a more computationally efficient means to performing an EXX calculation, albeit with a longer time to solution.

With access to massively parallel resources ($\zeta > 1 \in \mathbb{N}$), each $\tilde{\phi}_i$ is now replicated and stored in memory on the P_i, P_{i+N_o}, ..., $P_{i+(\zeta-1)N_o}$ processes. After running Algorithm 1

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

to generate $\mathcal{L}$, the workload associated with a given MLWF is split into ζ parts, each of which is assigned to one of the processes holding this orbital. For the isolated water molecule, the use of $\zeta = 2$ ($\zeta = 1$) results in processes assigned with 1−2 (2−3) computational tasks. This reduces the maximum number of computation events per process from 3 to 2 and hence lowers the overall time to solution. However, this gain comes at the expense of increasing the workload imbalance from 1/3 (i.e., processes with the lightest workload idling for ≈1/3 of the time) to 1/2 and is therefore a less efficient use of the available computational resources.

Construction of Proto-Subdomains. As discussed throughout this work, the efficient evaluation of $\tilde{v}_{ij}(\mathbf{r})$ is the cornerstone of our MLWF-based EXX approach. To exploit the sparsity of the MLWFs and still retain a numerically exact evaluation of $\tilde{v}_{ij}(\mathbf{r})$, this quantity is computed via the solution to the corresponding PE for all points in Ω that are contained in Ω_{ij} (see eq 21). Since the PE is a boundary-value problem, the required boundary conditions are provided by the ME of $\tilde{\rho}_{ij}(\mathbf{r})$ about $\tilde{\mathbf{C}}_{ij}$ on the thin shell of the real-space grid surrounding Ω_{ij} (see eqs 22 and 23). By computing $\tilde{v}_{ij}(\mathbf{r})$ for all $\mathbf{r} \in \Omega_{ij}$, E_{xx} can be computed in a numerically exact fashion (cf. eq 19). However, the numerically exact evaluation of the EXX contribution to the wave function forces, $\tilde{D}^{ij}_{xx}(\mathbf{r})$ and $\tilde{D}^{ji}_{xx}(\mathbf{r})$, requires $\tilde{v}_{ij}(\mathbf{r})$ for all $\mathbf{r} \in \Omega_j$ and $\mathbf{r} \in \Omega_i$, respectively (see eq 20). Since $\Omega_{ij} \subset \Omega_j$ and $\Omega_{ij} \subset \Omega_i$, $\tilde{D}^{ij}_{xx}(\mathbf{r})$ and $\tilde{D}^{ji}_{xx}(\mathbf{r})$ are evaluated with $\tilde{v}_{ij}(\mathbf{r})$ from the solution to the PE for all $\mathbf{r} \in \Omega_{ij}$. For $\mathbf{r} \in \Omega_j \backslash \Omega_{ij}$ and $\mathbf{r} \in \Omega_i \backslash \Omega_{ij}$, $\tilde{v}_{ij}(\mathbf{r})$ can be conveniently and accurately supplied by a sufficiently converged ME of $\tilde{\rho}_{ij}(\mathbf{r})$. As such, the ME serves the dual purpose of providing the necessary boundary conditions for the PE as well as the far-field $\tilde{v}_{ij}(\mathbf{r})$ required for a numerically exact computation of both $\tilde{D}^{ij}_{xx}(\mathbf{r})$ and $\tilde{D}^{ji}_{xx}(\mathbf{r})$.

To exploit this second level of computational savings, which originates from the fact that a numerically exact evaluation of all EXX-related quantities can be restricted to real-space domains that are system-size-independent and significantly smaller than Ω, the `exx` module introduces an alternative formulation of the Ω_i and Ω_{ij} subdomains described above in section II.C. To begin, we first note that subdomains like Ω_i and $\Omega_{ij} = \Omega_i \cap \Omega_j$ are formally defined as the points in Ω for which $\tilde{\phi}_i(\mathbf{r})$ and $\tilde{\rho}_{ij}(\mathbf{r})$ are non-negligible (i.e., larger than some predetermined numerical cutoff). As such, both of these subdomains can have irregular and even disjointed shapes. However, this is a cumbersome and computationally demanding definition that would require screening substantial sectors of Ω for each pair of MLWFs during every CPMD step. To combat this issue and still maintain a numerically exact evaluation of all required quantities, one could simply utilize two concentric *spherical* subdomains per $\langle ij \rangle$ pair, i.e., $\Theta(\tilde{\mathbf{C}}_{ij},R^{ij}_{PE})$ and $\Theta(\tilde{\mathbf{C}}_{ij},R^{ij}_{ME})$, which are spheres centered at $\tilde{\mathbf{C}}_{ij}$ with radii R^{ij}_{PE} and R^{ij}_{ME} chosen to be large enough to encompass Ω_{ij} and $\Omega_i \cup \Omega_j$, respectively. In doing so, the corresponding PE, $\nabla^2\tilde{v}_{ij}(\mathbf{r}) = -4\pi\tilde{\rho}_{ij}(\mathbf{r})$, could then be solved without any domain truncation error on $\Theta(\tilde{\mathbf{C}}_{ij},R^{ij}_{PE})$, which is significantly smaller than Ω. Computing the ME of $\tilde{\rho}_{ij}(\mathbf{r})$ on the $\Theta(\tilde{\mathbf{C}}_{ij},R^{ij}_{ME})\backslash\Theta(\tilde{\mathbf{C}}_{ij},R^{ij}_{PE})$ shell would again provide the necessary boundary conditions for the PE as well as the far-field $\tilde{v}_{ij}(\mathbf{r})$ needed for evaluating both $\tilde{D}^{ij}_{xx}(\mathbf{r})$ on Ω_j and $\tilde{D}^{ji}_{xx}(\mathbf{r})$ on Ω_i. Since both of these subdomains are contained in $\Theta(\tilde{\mathbf{C}}_{ij},R^{ij}_{ME})$, $\tilde{D}^{ij}_{xx}(\mathbf{r})$ and $\tilde{D}^{ji}_{xx}(\mathbf{r})$ can also be computed in a numerically exact fashion on a subset of points contained in Ω.

To efficiently utilize this concept of concentric spherical subdomains in the `exx` module, we assemble two fixed-size proto-subdomains, $\Theta(\mathbf{C}_0,R_{PE})$ and $\Theta(\mathbf{C}_0,R_{ME})$, centered around a predetermined origin, $\mathbf{C}_0$, which is chosen to be one of the grid points in Ω. When dealing with all computations involving a given $\langle ij \rangle$ pair, these proto-subdomains are simply translated to $\tilde{\mathbf{C}}_{ij}$, which will be approximated (with no discernible error) by $\mathbf{C}_{ij}$, the closest grid point in Ω (see Figure 5 and section III.C.1). Since these fixed-size proto-subdomains will be used for all $\langle ij \rangle$ pairs, their radii should be chosen such that $R_{PE} = \max_{ij}\{R^{ij}_{PE}\}$ and $R_{ME} = \max_{ij}\{R^{ij}_{ME}\}$. With judicious choices for R_{PE} and R_{ME} (see sections IV.A.1 to IV.A.2), these proto-subdomains allow for an accurate evaluation of all EXX-related quantities and have several algorithmic advantages that will be described below.

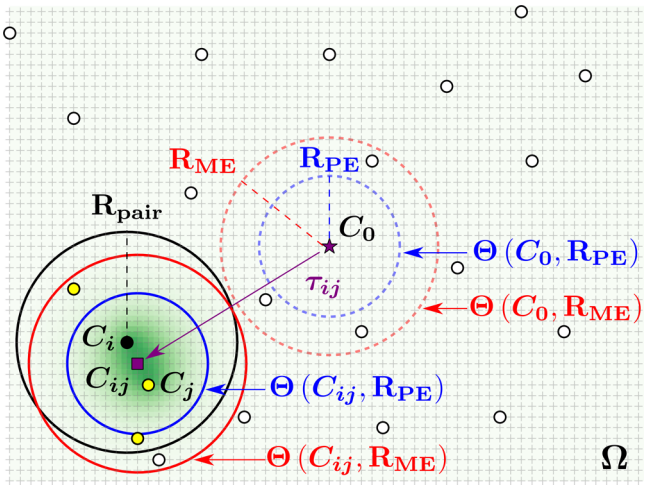


Figure 5. Graphical depiction of the proto-subdomains used in the `exx` module. Dots are used to denote the MLWF centers, $\tilde{\mathbf{C}}_i$, which are approximated by the closest points, $\mathbf{C}_i$, on the real-space grid, Ω. The dashed blue and red circles bound the two concentric spherical proto-subdomains, $\Theta(\mathbf{C}_0,R_{PE})$ and $\Theta(\mathbf{C}_0,R_{ME})$, which are assembled around $\mathbf{C}_0$ (purple star) with radii R_{PE} and R_{ME}, respectively. Pair-exchange interactions involving $\tilde{\phi}_i(\mathbf{r})$ include all overlapping MLWFs (yellow dots), whose centers are located within a distance, R_{pair}, of $\tilde{\mathbf{C}}_i \approx \mathbf{C}_i$ (black dot) that is large enough to account for all $\tilde{\phi}_k(\mathbf{r})$ with $\Omega_{ik} \neq \emptyset$. For the overlapping $\langle ij \rangle$ pair, the $\Theta(\mathbf{C}_0,R_{PE})$ and $\Theta(\mathbf{C}_0,R_{ME})$ proto-subdomains are translated across Ω via a rigid grid offset, τ_{ij}, to form $\Theta(\mathbf{C}_{ij},R_{PE})$ and $\Theta(\mathbf{C}_{ij},R_{ME})$, which are centered at $\mathbf{C}_{ij} \approx \tilde{\mathbf{C}}_{ij}$ (purple square). To evaluate the $\langle ij \rangle$ contribution to all EXX-related quantities, the corresponding PE, $\nabla^2\tilde{v}_{ij}(\mathbf{r}) = -4\pi\tilde{\rho}_{ij}(\mathbf{r})$, is solved for $\tilde{v}_{ij}(\mathbf{r})$ on the $\Theta(\mathbf{C}_{ij},R_{PE})$ subdomain, which encompasses Ω_{ij}, the support of $\tilde{\rho}_{ij}(\mathbf{r})$ (shaded in dark green). Boundary conditions for the PE (as well as the far-field $\tilde{v}_{ij}(\mathbf{r})$) are computed via a ME of $\tilde{\rho}_{ij}(\mathbf{r})$ on the $\Theta(\mathbf{C}_{ij},R_{ME})\backslash\Theta(\mathbf{C}_{ij},R_{PE})$ shell surrounding $\Theta(\mathbf{C}_{ij},R_{PE})$.

With R_{PE} and R_{ME} in hand, we now describe the construction of these proto-subdomains around $\mathbf{C}_0$, an arbitrary center that is coincident with a grid point in Ω. In this work, the grid point closest to the center of Ω was chosen as the reference $\mathbf{C}_0$, since this allows us to avoid the use of both the minimum image convention and wrap-around (periodic) boundary conditions during grid point screening. Assembly of the proto-subdomains begins by looping over grid points, $\mathbf{r} \in \Omega$, and determining whether or not a given grid point is contained within $\Theta(\mathbf{C}_0,R_{PE})$ or $\Theta(\mathbf{C}_0,R_{ME})\backslash\Theta(\mathbf{C}_0,R_{PE})$. For each grid point contained in either proto-subdomain, we increment the corresponding counter (q' or q'') and store its

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

relative (*local*) Cartesian coordinates (in $\overline{\mathbf{r}}_{\mathrm{PE}}$ or $\overline{\mathbf{r}}_{\mathrm{ME}}$) and (*global*) grid point indices (in $\mathbf{g}^0_{\mathrm{PE}}$ or $\mathbf{g}^0_{\mathrm{ME}}$), as depicted in Algorithm 2.

Algorithm 2 Proto-Subdomain Construction

$q' \leftarrow 0$; $q'' \leftarrow 0$
foreach $\boldsymbol{r} \in \Omega$ **do**
  if $|\boldsymbol{r} - \boldsymbol{C}_0| \leq R_{\mathrm{PE}}$ **then**
    $q' \leftarrow q' + 1$
    $\overline{\boldsymbol{r}}_{\mathrm{PE}}[q'] \leftarrow \boldsymbol{r} - \boldsymbol{C}_0$
    $\boldsymbol{g}^0_{\mathrm{PE}}[q'] \leftarrow \mathtt{NINT}\left[N_{\mathrm{grid},a} r_a / |\boldsymbol{L}_a|\right], \quad a = 1,2,3$
  else if $R_{\mathrm{PE}} < |\boldsymbol{r} - \boldsymbol{C}_0| \leq R_{\mathrm{ME}}$ **then**
    $q'' \leftarrow q'' + 1$
    $\overline{\boldsymbol{r}}_{\mathrm{ME}}[q''] \leftarrow \boldsymbol{r} - \boldsymbol{C}_0$
    $\boldsymbol{g}^0_{\mathrm{ME}}[q''] \leftarrow \mathtt{NINT}\left[N_{\mathrm{grid},a} r_a / |\boldsymbol{L}_a|\right], \quad a = 1,2,3$
  end if
end for
$N_{\mathrm{PE}} \leftarrow q'$
$N_{\mathrm{ME}} \leftarrow q' + q''$

Incrementing these counters throughout the loop over $\mathbf{r} \in \Omega$ yields N_{PE} and N_{ME}, the (fixed) number of points in $\Theta(\mathbf{C}_0,R_{\mathrm{PE}})$ and $\Theta(\mathbf{C}_0,R_{\mathrm{ME}})$. By storing the relative Cartesian coordinates, $\mathbf{r} - \mathbf{C}_0$, we now have a set of *local* coordinates that are invariant to rigid translations of $\Theta(\mathbf{C}_0,R_{\mathrm{PE}})$ and $\Theta(\mathbf{C}_0,R_{\mathrm{ME}})$, thereby avoiding the need to recompute these coordinates for every $\langle ij \rangle$ pair. This also provides a convenient platform for precomputing a number of quantities (e.g., $\mathbf{r}$ in spherical polar coordinates, the set of spherical harmonics, etc.) that are required during the ME of $\tilde{\rho}_{ij}(\mathbf{r})$ (cf. eqs 22 and 23). For each point in the proto-subdomains, we also store its *global* grid point indices, which are given by three integer values, (g^0_1, g^0_2, g^0_3), representing the position of a given grid point along the cell (lattice) vectors, $\mathbf{L}_1$, $\mathbf{L}_2$, and $\mathbf{L}_3$ (which are aligned with the Cartesian directions for the orthorhombic cells considered in this work). For an orthogonal grid, which has $N_{\mathrm{grid},a}$ equispaced grid points along each of the $\mathbf{L}_a$ lattice vectors (with grid spacing $\delta\xi_a = |\mathbf{L}_a|/N_{\mathrm{grid},a}$), the global grid index along $\mathbf{L}_a$ is given by $g^0_a = r_a/\delta\xi_a = N_{\mathrm{grid},a} r_a/|\mathbf{L}_a|$. Since $\mathbf{r}$ is always coincident with a grid point in Ω, $\{g^0_a\}$ is formally an array of integers; this is enforced in a floating-point environment using the nearest integer function, NINT.

This accumulated data is then concatenated to form two $3 \times N_{\mathrm{ME}}$ arrays as follows: the local coordinates are stored in a double-precision array,

$$\overline{\mathbf{r}}[q] = \begin{Bmatrix} \overline{\mathbf{r}}_{\mathrm{PE}}[q] & q = 1, \ldots, N_{\mathrm{PE}} \\ \overline{\mathbf{r}}_{\mathrm{ME}}[q - N_{\mathrm{PE}}] & q = N_{\mathrm{PE}} + 1, \ldots, N_{\mathrm{ME}} \end{Bmatrix} \tag{30}$$

while the global grid indices are stored in an integer array,

$$\mathbf{g}^0[q] = \begin{Bmatrix} \mathbf{g}^0_{\mathrm{PE}}[q] & q = 1, \ldots, N_{\mathrm{PE}} \\ \mathbf{g}^0_{\mathrm{ME}}[q - N_{\mathrm{PE}}] & q = N_{\mathrm{PE}} + 1, \ldots, N_{\mathrm{ME}} \end{Bmatrix} \tag{31}$$

By storing all of the subdomain data in this scheme, only a single local index, q, is required for labeling the elements in these arrays. This still maintains access to the $\Theta(\mathbf{C}_0,R_{\mathrm{PE}})$ and $\Theta(\mathbf{C}_0,R_{\mathrm{ME}})$ proto-subdomains (as well as the $\Theta(\mathbf{C}_0,R_{\mathrm{ME}})\backslash\Theta(\mathbf{C}_0,R_{\mathrm{PE}})$ shell) through knowledge of N_{PE} and N_{ME}, the number of elements in each proto-subdomain. As such, this scheme provides us with a compact representation for the sparse quantities required in our EXX algorithm, as well as a convenient mapping between data stored in the proto-subdomain representation and the real-space grid (Ω) representation. This is crucial for loading and off-loading data to and from Ω, as it only requires communication of the relevant sectors of Ω for sparse quantities like $\tilde{\rho}_{ij}(\mathbf{r})$ and $\tilde{v}_{ij}(\mathbf{r})$.

III.C.3. Step III: Communication of MLWFs. By virtue of the ORBITAL data distribution scheme, the unique MLWF-pair list, $\mathcal{L}$, not only determines the computational workload associated with each MPI process but also encodes the communication protocol that will be followed throughout the remainder of the exx module (see Figure 4). With a support that is significantly smaller than Ω, the communication of any given MLWF on the entire real-space grid is clearly neither efficient nor necessary in our EXX algorithm. To reduce the communication overhead associated with each overlapping $\langle ij \rangle$ pair, the exx module employs the proto-subdomains ($\Theta(\mathbf{C}_0,R_{\mathrm{PE}})$ and $\Theta(\mathbf{C}_0,R_{\mathrm{ME}})$) introduced in section III.C.2. As discussed above, these system-size-independent proto-subdomains provide a compact data representation for the storage and communication of sparse quantities such as $\tilde{\phi}_i$, $\tilde{\rho}_{ij}$, $\tilde{v}_{ij}$, and $\tilde{D}^{ij}_{\mathrm{xx}}$ (or $\tilde{D}^{ji}_{\mathrm{xx}}$). To utilize $\Theta(\mathbf{C}_0,R_{\mathrm{PE}})$ and $\Theta(\mathbf{C}_0,R_{\mathrm{ME}})$ in practice, these proto-subdomins must be translated across Ω to form the subdomains, $\Theta(\mathbf{C}_{ij},R_{\mathrm{PE}})$ and $\Theta(\mathbf{C}_{ij},R_{\mathrm{ME}})$, required for evaluating all quantities associated with a given $\langle ij \rangle$ pair, as shown in Figure 5.

Before describing the translation of these proto-subdomains, we now discuss the employed convention used for $\tilde{\mathbf{C}}_{ij}$ and remind the reader of the flexibility one has in defining this quantity for neutral charge distributions like $\tilde{\rho}_{ij}(\mathbf{r})$ (see section II.C). In the exx module, $\tilde{\mathbf{C}}_{ij}$ is defined as the midpoint between the ith and jth MLWF centers, i.e., $\tilde{\mathbf{C}}_{ij} = (\tilde{\mathbf{C}}_i + \tilde{\mathbf{C}}_j)/2$, which represents an excellent approximation to the aforementioned gauge used in molecular quantum mechanics and an algorithmically convenient choice. By utilizing the MLWF centers, this definition for $\tilde{\mathbf{C}}_{ij}$ accounts for the spatial distribution of each MLWF through its first moment and becomes equivalent to the conventional definition, $\tilde{\mathbf{C}}_{ij} = \int d\mathbf{r}\, \mathbf{r}\, |\tilde{\rho}_{ij}(\mathbf{r})| \,/ \int d\mathbf{r}\, |\tilde{\rho}_{ij}(\mathbf{r})|$, for a number of different symmetric cases (e.g., when both $\tilde{\phi}_i(\mathbf{r})$ and $\tilde{\phi}_j(\mathbf{r})$ have the same spread and are spherically symmetric with respect to $\tilde{\mathbf{C}}_i$ and $\tilde{\mathbf{C}}_j$, when $\tilde{\rho}_{ij}(\mathbf{r})$ is centrosymmetric with respect to the midpoint, etc.). Furthermore, this choice for $\tilde{\mathbf{C}}_{ij}$ recovers the correct center of charge for $\tilde{\rho}_{ii}(\mathbf{r})$, i.e., $\tilde{\mathbf{C}}_{ii} \rightarrow \tilde{\mathbf{C}}_{ii} = \tilde{\mathbf{C}}_i$. Algorithmically speaking, this convention for $\tilde{\mathbf{C}}_{ij}$ is also quite useful, as it only requires knowledge of the MLWF centers, which are available throughout a CPMD simulation.

As mentioned above, the exx module employs one additional simplification when dealing with MLWF and MLWF-pair centers: these quantities are approximated by the closest grid points in Ω and denoted throughout by either $\mathbf{C}_i$ or $\mathbf{C}_{ij}$. This algorithmic simplification leads to no appreciable error during evaluation of E_{xx} and $\{\tilde{D}^i_{\mathrm{xx}}\}$ and allows us to rigidly translate the proto-subdomains to the appropriate center, $\mathbf{C}_{ij}$, as needed. For an orthogonal grid, the component of the required grid translation vector, $\boldsymbol{\tau}^{ij}$, along a given lattice vector, $\mathbf{L}_a$, is given by

$$\tau^{ij}_a = \mathrm{NINT}[N_{\mathrm{grid},a}(\mathbf{C}_{ij} - \mathbf{C}_0)_a/|\mathbf{L}_a|] = \mathrm{NINT}\left[\frac{(\mathbf{C}_{ij} - \mathbf{C}_0)_a}{\delta\xi_a}\right] \tag{32}$$

Application of $\boldsymbol{\tau}^{ij}$ to a given proto-subdomain leaves the radius and local Cartesian coordinates untouched and simply offsets the global grid indices as follows:

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

$$g_a^{ij}[q] = \mathrm{MOD}[g_a^0[q]+\tau_a^{ij},N_{\mathrm{grid},a}] \tag{33}$$

thereby resulting in a subdomain centered at $\mathbf{C}_{ij}$. The use of the remainder function, MOD, in eq 33 enforces the appropriate wrap-around boundary conditions; as such, this equation is specific to the grid convention used in QE, in which the grid points (along $\mathbf{L}_a$) are numbered from 0, 1, ..., $N_{\mathrm{grid},a} - 1$. For codes that number these grid points from 1, 2, ..., $N_{\mathrm{grid},a}$, eq 33 should be modified as follows: $g_a^{ij}[q] = \mathrm{MOD}[g_a^0[q] + \tau_a^{ij} - 1,N_{\mathrm{grid},a}] + 1$.

With each MLWF stored according to the ORBITAL data distribution scheme, the MPI process ($\zeta = 1$) or processes ($\zeta > 1$) that are currently storing $\tilde{\phi}_i(\mathbf{r})$ on Ω are responsible for sending this MLWF to another MPI process (or processes) according to the computation and communication protocol outlined by $\mathcal{L}$. In order to do so, $\tilde{\phi}_i(\mathbf{r})$ is off-loaded onto the appropriately translated subdomains, $\Theta(\mathbf{C}_{ij},R_{\mathrm{ME}})$, corresponding to the overlapping $\langle ij\rangle$ pairs that will be handled remotely (i.e., on other processes); all of the information required to do so is provided by local access to $\mathcal{L}$ and $\{\mathbf{C}_{ij}\}$, as both of these arrays have been broadcast to all processes. For each of these overlapping $\langle ij\rangle$ pairs, a sparse quantity like $\tilde{\phi}_i(\mathbf{r})$ is now stored in the more compact $\Theta(\mathbf{C}_{ij},R_{\mathrm{ME}})$ representation via the use of three relatively small arrays: $\bar{\mathbf{r}}$, $\mathbf{g}^{ij}$, and $\tilde{\phi}_i(\bar{\mathbf{r}}) \equiv \tilde{\phi}_i(\bar{\mathbf{r}}+\mathbf{C}_{ij})$, with associated sizes (types) of $3 \times N_{\mathrm{ME}}$ (double-precision), $3 \times N_{\mathrm{ME}}$ (integer), and $1 \times N_{\mathrm{ME}}$ (double-precision), respectively. Here, we remind the reader that all of the data on the $\Theta(\mathbf{C}_{ij},R_{\mathrm{PE}})$ subdomain and $\Theta(\mathbf{C}_{ij},R_{\mathrm{ME}})\backslash\Theta(\mathbf{C}_{ij},R_{\mathrm{PE}})$ shell are contained within $\Theta(\mathbf{C}_{ij},R_{\mathrm{ME}})$ and can easily be accessed using the local grid indexing scheme outlined in eqs 30 and 31. As mentioned above, the local Cartesian coordinates stored in the $\bar{\mathbf{r}}$ array are invariant to translations of the proto-subdomains; as such, this information does not need to be recomputed for each translated subdomain and can be broadcast across all processes. Communication of the MLWFs on these compact subdomains then proceeds according to $\mathcal{L}$ among the pool of available MPI processes. Once $\tilde{\phi}_i(\bar{\mathbf{r}})$ is received by a given process, $\tilde{\rho}_{ij}(\bar{\mathbf{r}})$ is assembled on the $\Theta(\mathbf{C}_{ij},R_{\mathrm{PE}})$ subdomain and the exx module begins the process of solving the corresponding PE.

III.C.4. Step IV: Solution of Poisson's Equation. On the basis of $\mathcal{L}$, each MPI process, P_i, now holds an assigned MLWF-product density $\tilde{\rho}_{ij}(\bar{\mathbf{r}})$ as well as the relevant quantities that map the $\Theta(\mathbf{C}_{ij},R_{\mathrm{PE}})$ and $\Theta(\mathbf{C}_{ij},R_{\mathrm{ME}})$ subdomains onto Ω (i.e., N_{PE}, N_{ME}, $\{\bar{\mathbf{r}}\}$, and $\{\mathbf{g}^{ij}\}$). As such, P_i has all of the required information to compute $\tilde{v}_{ij}(\bar{\mathbf{r}})$ on the $\Theta(\mathbf{C}_{ij},R_{\mathrm{PE}})$ and $\Theta(\mathbf{C}_{ij},R_{\mathrm{ME}})\backslash\Theta(\mathbf{C}_{ij},R_{\mathrm{PE}})$ subdomains. On $\Theta(\mathbf{C}_{ij},R_{\mathrm{PE}})$, $\tilde{v}_{ij}(\bar{\mathbf{r}})$ is obtained via the solution of the PE in eq 21. On $\Theta(\mathbf{C}_{ij},R_{\mathrm{ME}})\backslash\Theta(\mathbf{C}_{ij},R_{\mathrm{PE}})$, $\tilde{v}_{ij}(\bar{\mathbf{r}})$ is obtained via the ME in eqs 22 and 23, which provides the appropriate boundary conditions for the PE as well as the far-field potential.

While the ME of $\tilde{\rho}_{ij}(\bar{\mathbf{r}})$ (about $\mathbf{C}_{ij}$) can be straightforwardly computed using the local coordinates, $\{\bar{\mathbf{r}}\}$, the PE requires a discrete representation of the Laplacian operator for computing numerical second derivatives on these subdomains. Since the subdomains employed in the exx module are coincident with the underlying real-space grid (taken here to be orthogonal), the Laplacian operator can be expressed as a sum of second partial derivatives along each of the lattice directions, $\nabla^2 = \sum_a \frac{\partial^2}{\partial \xi_a^2}$, in which ξ_a is a coordinate of $\hat{\mathbf{L}}_a \equiv \mathbf{L}_a/|\mathbf{L}_a|$. At a given grid point, $\boldsymbol{\xi}_0$, the second partial derivative of a function, $f(\boldsymbol{\xi})$, along $\mathbf{L}_a$ was discretized via the standard central-difference approach:[143]

$$\left.\frac{\partial^2 f(\boldsymbol{\xi})}{\partial \xi_a^2}\right|_{\boldsymbol{\xi}=\boldsymbol{\xi}_0} = \sum_{q=-n}^{n} w_q \frac{f(\boldsymbol{\xi}_0 + q\delta\xi_a\hat{\mathbf{L}}_a)}{\delta\xi_a^2} \tag{34}$$

In this expression, the associated discretization error depends on the number, n, of (equispaced) neighboring grid points on each side of $\boldsymbol{\xi}_0$ and w_q is the central-difference coefficient for the qth grid point. We note in passing that only $w_{|q|}$ is required due to the central symmetry ($w_q = w_{-q}$) of the equispaced finite-difference stencil. Discretization of this derivative results in a $(2n + 1)$-point stencil along each grid direction, $\mathbf{L}_a$; as such, the discrete 3D Laplacian operator corresponds to a finite-difference stencil covering $3 \times 2n + 1 = 6n + 1$ grid points. We note in passing that the choice of $n = 3$ (with corresponding central-difference coefficients[143] given by $w_0 = -49/18$, $w_1 = +3/2 = w_{-1}$, $w_2 = -3/20 = w_{-2}$, and $w_3 = +1/90 = w_{-3}$) yields a second derivative with an associated discretization error of $O(\delta\xi^6)$; this choice is the default option in the exx module as it yields a well-converged value for E_{xx}.[76,78]

With this discrete representation of the Laplacian, we can express the PE in eq 21, $\nabla^2\tilde{v}_{ij}(\bar{\mathbf{r}}) = -4\pi\tilde{\rho}_{ij}(\bar{\mathbf{r}})$, as the following set of sparse linear equations on the $\Theta(\mathbf{C}_{ij},R_{\mathrm{PE}})$ subdomain:

$$\nabla_{\mathrm{PE}}^2\tilde{\mathbf{v}}_{ij} = -4\pi(\tilde{\boldsymbol{\rho}}_{ij} - \tilde{\boldsymbol{\rho}}_{ij}^{b}) \tag{35}$$

In this expression, ∇_{PE}^2 is a sparse $N_{\mathrm{PE}} \times N_{\mathrm{PE}}$ matrix containing the discretized Laplacian (whose stencil coverage has been restricted to $\Theta(\mathbf{C}_{ij},R_{\mathrm{PE}})$), $\tilde{\mathbf{v}}_{ij}$ is a $N_{\mathrm{PE}} \times 1$ vector representing the (currently unknown) MLWF-product potential, and $\tilde{\boldsymbol{\rho}}_{ij}$ is a $N_{\mathrm{PE}} \times 1$ vector containing the MLWF-product density. The final term on the right-hand side, $\tilde{\boldsymbol{\rho}}_{ij}^{b} \equiv -(1/4\pi)(\nabla^2 - \nabla_{\mathrm{PE}}^2)\tilde{\mathbf{v}}_{ij}$, is the so-called boundary charge, which accounts for the part(s) of the Laplacian stencil that extend outside of $\Theta(\mathbf{C}_{ij},R_{\mathrm{PE}})$ (and into the $\Theta(\mathbf{C}_{ij},R_{\mathrm{ME}})\backslash\Theta(\mathbf{C}_{ij},R_{\mathrm{PE}})$ shell) and have been truncated in the ∇_{PE}^2 representation of this operator. In doing so, $\tilde{\boldsymbol{\rho}}_{ij}^{b}$ accounts for the Dirichlet boundary conditions provided by the ME form of the potential on the $\Theta(\mathbf{C}_{ij},R_{\mathrm{ME}})\backslash\Theta(\mathbf{C}_{ij},R_{\mathrm{PE}})$ shell surrounding $\Theta(\mathbf{C}_{ij},R_{\mathrm{PE}})$ (see eq 22) and therefore allows for a numerically exact solution of the PE using ∇_{PE}^2, a Laplacian whose stencil coverage has been restricted to the $\Theta(\mathbf{C}_{ij},R_{\mathrm{PE}})$ subdomain. This restricts the PE to the support of $\tilde{\rho}_{ij}$ and substantially reduces the dimensionality and associated computational cost of solving the PE for each overlapping MLWF pair.

The system of sparse linear equations in eq 35 is then solved (for $\tilde{\mathbf{v}}_{ij}$) using an iterative conjugate-gradient (CG) approach. Since the solution of the PE is notoriously difficult to parallelize efficiently over MPI tasks, the CG-based PE solver in the exx module is largely parallelized over OpenMP threads to allow for an efficient real-space evaluation of $\tilde{\mathbf{v}}_{ij}$. The efficient solution of the PE for each overlapping MLWF pair is the cornerstone of our MLWF-based EXX algorithm, and the performance of the CG-based PE solver will be discussed in section IV.B.2. During CPMD simulations, the number of CG iterations required to solve a given PE can be substantially reduced by using a polynomial extrapolation[144] of the potential from the previous CPMD steps as the initial guess. More

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

detailed considerations of this extrapolation scheme as well as extensions to BOMD will be discussed in future work.

III.C.5. Step V: Computation of Energy and Forces. After a process, P_i, arrives at the solution to the PE for one of its assigned pairs (i.e., for a given $j \in \mathcal{L}[i]$), this process now holds the corresponding MLWF-product potential $\tilde{v}_{ij}(\bar{\mathbf{r}})$ on the entire $\Theta(\mathbf{C}_{ij},R_{ME})$ subdomain. With $\tilde{v}_{ij}(\bar{\mathbf{r}})$ on the $\Theta(\mathbf{C}_{ij},R_{PE})$ subdomain (via the solution of the PE) and $\tilde{v}_{ij}(\bar{\mathbf{r}})$ on the $\Theta(\mathbf{C}_{ij},R_{ME})\backslash\Theta(\mathbf{C}_{ij},R_{PE})$ shell (via the ME of $\tilde{\rho}_{ij}(\bar{\mathbf{r}})$), P_i is now ready to evaluate the $\langle ij\rangle$ contribution to the EXX energy (E_{xx}) and wave function forces ($\tilde{D}^{ij}_{xx}(\bar{\mathbf{r}})$ and $\tilde{D}^{ji}_{xx}(\bar{\mathbf{r}})$).

The evaluation of E_{xx} is quite straightforward via eq 19. Here, we remind the reader that a numerically exact evaluation of this quantity only requires integration on the $\Theta(\mathbf{C}_{ij},R_{PE})$ subdomain; this integration over the support of $\tilde{\rho}_{ij}(\mathbf{r})$ is also parallelized over OpenMP threads and is therefore quite computationally efficient. Partial summations over the assigned $\langle ij\rangle$ pairs on each process are accumulated to form E_{xx} with minimal associated communication (i.e., one double-precision number for E_{xx} per MPI process).

With $\tilde{v}_{ij}(\bar{\mathbf{r}})$ in hand, P_i is also in position to compute both $\tilde{D}^{ij}_{xx}(\bar{\mathbf{r}}) = \tilde{v}_{ij}(\bar{\mathbf{r}})\,\tilde{\phi}_j(\bar{\mathbf{r}})$ and $\tilde{D}^{ji}_{xx}(\bar{\mathbf{r}}) = \tilde{v}_{ij}(\bar{\mathbf{r}})\,\tilde{\phi}_i(\bar{\mathbf{r}})$ on the entire $\Theta(\mathbf{C}_{ij},R_{ME})$ subdomain. For each $\tilde{D}^{ji}_{xx}(\bar{\mathbf{r}})$ computed on P_i, this quantity is shipped back to P_j (assuming $\zeta = 1$ here for simplicity), which requires communication of N_{ME} double-precision numbers for each $\tilde{D}^{ji}_{xx}(\bar{\mathbf{r}})$; this array is equivalent in size to $\tilde{\phi}_j(\bar{\mathbf{r}})$ and represents the necessary second communication event described in section III.C.2. Since P_i has access to $\tilde{v}_{ij}(\bar{\mathbf{r}})\ \forall\ j \in \mathcal{L}[i]$, this process accumulates $\tilde{D}^{i}_{xx}(\bar{\mathbf{r}}) = \sum_j \tilde{D}^{ij}_{xx}(\bar{\mathbf{r}})$ into a local temporary array that is the size of the global real-space grid; as P_i receives a given $\tilde{D}^{ij}_{xx}(\bar{\mathbf{r}})$ array from P_j, this quantity is also accumulated into this local temporary array. When all $\tilde{D}^{ij}_{xx}(\bar{\mathbf{r}})$ contributions are accounted for, this temporary array on P_i holds the final $\tilde{D}^{i}_{xx}(\mathbf{r})$ according to the ORBITAL data distribution scheme.

III.C.6. Step VI: Redistribution of Wave Function Forces. At this stage, all of the EXX-related quantities have been evaluated; E_{xx} has been accumulated and broadcast to all processes, while $\{\tilde{D}^{i}_{xx}(\mathbf{r})\}$ is stored in the ORBITAL data distribution scheme. As such, the remaining task for the exx module is the transformation of $\{\tilde{D}^{i}_{xx}(\mathbf{r})\}$ to the GRID data distribution scheme for compliance with the core functions in QE (see section III.B). This redistribution is essentially the reverse operation of the GRID → ORBITAL redistribution of the MLWFs described in Figure 3 and section III.C.1.

At this stage, the QE executable exits the exx module with E_{xx} and $\{\tilde{D}^{i}_{xx}(\mathbf{r})\}$ (in the GRID data distribution scheme) as output. These EXX-related quantities are then added to their semilocal exchange analogues with the appropriate EXX fraction, a_x, given in eq 2. With the EXX contribution to the wave function (MLWF) forces, the CPMD equations of motion are now propagated forward via eqs 9 and 10.

IV. ACCURACY AND PERFORMANCE

During the implementation of the exx module, we have introduced three parameters: R_{pair}, R_{PE}, and R_{ME} (see section III.C.2). R_{pair} is used to determine whether or not two MLWFs, $\tilde{\phi}_i(\mathbf{r})$ and $\tilde{\phi}_j(\mathbf{r})$, are considered to be an overlapping $\langle ij\rangle$ pair via $|\tilde{\mathbf{C}}_i - \tilde{\mathbf{C}}_j| < R_{pair}$. For all overlapping $\langle ij\rangle$ pairs, R_{PE} and R_{ME} are the radii of the spherical $\Theta(\mathbf{C}_{ij},R_{PE})$ and $\Theta(\mathbf{C}_{ij},R_{ME})$ subdomains, which are centered at $\mathbf{C}_{ij}$ and chosen to cover the product density, $\tilde{\rho}_{ij}(\mathbf{r})$, and individual orbitals, $\tilde{\phi}_i(\mathbf{r})$ and $\tilde{\phi}_j(\mathbf{r})$, respectively. In order to efficiently perform large-scale hybrid DFT-based AIMD simulations, judicious choices for R_{pair}, R_{PE}, and R_{ME} are required to balance the performance and accuracy and are therefore the focus of this section. In section IV.A, we introduce a systematic selection of these parameters based on user-defined error thresholds for E_{xx} and $\{\tilde{D}^{i}_{xx}(\mathbf{r})\}$. In section IV.B, we discuss the intranode (OpenMP) and internode (MPI) parallel scaling performance when simulating liquid water using our MLWF-based EXX approach on several different supercomputer architectures.

IV.A. Parameters and Convergence Criteria. In this section, a systematic determination of all required parameters will be demonstrated using a snapshot from a liquid water simulation containing $(H_2O)_{64}$ in a cubic cell with $L = 23.52$ bohr. We begin by performing a reference single-point energy evaluation in QE at the PBE0 level with a planewave kinetic energy cutoff of 85 Ry. In this reference calculation (which yields E^{ref}_{xx}), we use the largest possible values for all three parameters such that (i) all *radially* overlapping MLWF pairs are included with $|\tilde{\mathbf{C}}_i - \tilde{\mathbf{C}}_j| < R_{pair} = L/2 = 11.76$ bohr and (ii) both proto-subdomains ($\Theta(\mathbf{C}_0,R_{PE})$ and $\Theta(\mathbf{C}_0,R_{ME})$) are contained within the simulation cell (i.e., $R_{ME} = L/2 = 11.76$ bohr and $R_{PE} = L/2 - n\max_a\{\delta\xi_a\} = 11.11$ bohr). Subtraction of $n\max_a\{\delta\xi_a\}$ (with $n = 3$) from R_{PE} allows us to retain a thin shell of the real-space grid surrounding $\Theta(\mathbf{C}_0,R_{PE})$; this shell provides the boundary conditions for the PE and accommodates the part(s) of the discretized Laplacian stencil that extend beyond $\Theta(\mathbf{C}_0,R_{PE})$. We note in passing that the energetic contributions to E^{ref}_{xx} from MLWF pairs with $|\tilde{\mathbf{C}}_i - \tilde{\mathbf{C}}_j| > R_{pair} = L/2 = 11.76$ bohr (within the minimum image convention) are completely negligible (i.e., 2.0×10^{-5} kcal/mol or $\approx 4.8 \times 10^{-6}$%).

As such, this reference calculation provides an approximately exact evaluation of E_{xx} using the exx module, albeit at an excessive computational cost. More specifically, the computational cost associated with the CG solution of the PE in exx scales as $O(N^{4/3}_{PE})$, with N_{PE} asymptotically growing as $O(R^3_{PE})$. In the same breath, the associated communication and memory footprint in exx scale as $O(N_{ME})$, with N_{ME} asymptotically growing as $O(R^3_{ME})$. In addition, both computation and communication in exx scale as $O(N_{pair})$, with N_{pair} asymptotically growing as $O(R^3_{pair})$ (for homogeneous systems with constant densities). As such, judicious choices for R_{pair}, R_{PE}, and R_{ME} are required to balance the performance and accuracy of our algorithm (see sections IV.A.1 and IV.A.2).

Since the number of self-pairs (with $i = j$) is N_o and the number of non-self-pairs (with $i \neq j$) is $(\tilde{n} - 1)N_o$ (see section II.C), the computational cost associated with evaluating the contribution to E_{xx} from non-self-pairs will dominate that from self-pairs when MLWF-based EXX calculations are performed. However, the energetic contribution to E_{xx} from the self-pairs tends to dominate the contribution from non-self-pairs, which is primarily due to the fact that the overlap between an MLWF and itself is substantially larger than the overlap between an MLWF and any one of its neighbors. In the reference calculation described above, for example, the energetic contribution to E^{ref}_{xx} is dominated ($\approx$85%) by the self-pairs

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

while the computational cost to evaluate E_{xx}^{ref} (≈87%) mainly originates from the non-self-pairs. Taken together, these observations suggest that we can further balance the performance and accuracy of our algorithm by employing a larger value of R_{PE} for the self-pairs (R_{PE}^{s}, which defines the $\Theta(\mathbf{C}_0,R_{PE}^{s})$ proto-subdomain) and a smaller value of R_{PE} for the non-self-pairs (R_{PE}^{ns}, which defines the $\Theta(\mathbf{C}_0,R_{PE}^{ns})$ proto-subdomain). Since R_{PE}^{s} will in general be larger than R_{PE}^{ns}, we will employ the same strategy by using a larger value of R_{ME} for the self-pairs (R_{ME}^{s}) and a smaller value of R_{ME} for the non-self-pairs (R_{ME}^{ns}), which define the $\Theta(\mathbf{C}_0,R_{ME}^{s})$ and $\Theta(\mathbf{C}_0,R_{ME}^{ns})$ proto-subdomains, respectively. Physically speaking, the choice to use larger (smaller) R_{PE} and R_{ME} values for the self (non-self) pairs is also justified by the fact that (i) self MLWF-product densities ($\tilde{\rho}_{ii}(\mathbf{r})$) have a larger support than non-self-MLWF-product densities ($\tilde{\rho}_{ij}(\mathbf{r})$) due to the increased overlap between an MLWF and itself, and (ii) self MLWF-product potentials ($\tilde{v}_{ii}(\mathbf{r})$) are generally longer-ranged than non-self-MLWF-product potentials ($\tilde{v}_{ij}(\mathbf{r})$) due to the absence of a monopolar contribution in the non-self cases (see section II.C).

IV.A.1. Convergence of the EXX Contribution to the Energy. As an initial test, E_{xx} will be computed using the converged MLWFs obtained during the calculation of E_{xx}^{ref}. Since the effects of self-consistency are neglected in this test (and will be investigated below), we can quickly assess the convergence of E_{xx} with respect to E_{xx}^{ref} as a function of R_{pair}, R_{PE}^{s}, and R_{PE}^{ns} (without the need for modifying R_{ME}^{s} and R_{ME}^{ns}). We do so by independently varying each of the R_{pair}, R_{PE}^{s}, and R_{PE}^{ns} parameters, while keeping all other parameters fixed at their largest possible values (see Figure 6). From this figure,

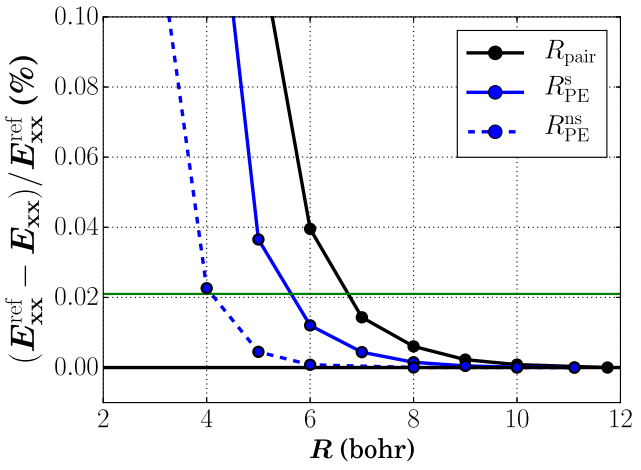


Figure 6. Convergence of E_{xx} as a function of R_{pair}, R_{PE}^{s}, and R_{PE}^{ns} on a snapshot of liquid water containing $(H_2O)_{64}$ in a cubic cell with L = 23.52 bohr. Relative errors (in %) with respect to E_{xx}^{ref} are evaluated individually by varying R_{pair} (solid black line), R_{PE}^{s} (solid blue line), and R_{PE}^{ns} (dashed blue line), while keeping all other parameters set to their maximum allowed values (see text for more details). An overall relative error of ≈0.02% corresponds to the default parameter values in QE (i.e., R_{pair} = 8.0 bohr, R_{PE}^{s} = 6.0 bohr, R_{PE}^{ns} = 5.0 bohr) and is depicted by the solid green line.

one can immediately see that the relative (percent) error in E_{xx} rapidly decays as each of these parameter values is increased. This observation can be justified by considering the fact that each MLWF (in finite-gap systems) is exponentially localized, and hence products of MLWFs (i.e., $\tilde{\rho}_{ii}(\mathbf{r})$ or $\tilde{\rho}_{ij}(\mathbf{r})$) are also exponentially localized. As such, increasing R_{PE}^{s} (or R_{PE}^{ns}) leads to spherical PE domains that increasingly cover the exponentially decaying tails of $\tilde{\rho}_{ii}(\mathbf{r})$ (or $\tilde{\rho}_{ij}(\mathbf{r})$); as seen in eq 19 (and the surrounding discussion), this results in rapid convergence to the reference value for E_{xx}. Since E_{xx} converges more quickly with R_{PE}^{ns} (than R_{PE}^{s}), this finding confirms our physical intuition that $R_{PE}^{s} > R_{PE}^{ns}$ and further justifies our use of separate self- and non-self-proto-subdomains as a means to improving the balance between performance and accuracy in this algorithm. By increasing R_{pair}, the incremental contribution to E_{xx} from (more) distant MLWF pairs becomes negligible as $\tilde{\rho}_{ij}(\mathbf{r}) \to 0\ \forall\ \mathbf{r}$, which also results in rapid convergence to the reference value for E_{xx}.

On the basis of this initial convergence test, we have chosen R_{pair} = 8.0 bohr, R_{PE}^{s} = 6.0 bohr, and R_{PE}^{ns} = 5.0 bohr as the default parameter set in QE; for EXX-based simulations of liquid water, the overall relative error is ≈0.02%, which is essentially additive (in each of these parameters) and typically rather stringent when ground-state energies, binding/cohesive energetics, and ionic forces are obtained in the condensed phase (vide infra). To increase the convergence of E_{xx}, a general rule of thumb is to first increase R_{PE}^{s}, since the self-pair contribution is the dominant contribution to E_{xx} yet requires evaluation of fewer terms (and is therefore significantly cheaper to compute) than the contribution from non-self-pairs. In this example, one can further reduce the relative error in E_{xx} by an additional factor of 2 (i.e., to ≈0.01%) by simply increasing R_{PE}^{s} from 6.0 to 7.0 bohr, with negligible (<1%) additional computational cost.

In this convergence test, we have neglected the effects of orbital self-consistency when determining E_{xx}. To quantify this effect, we first performed a fully self-consistent EXX calculation using the default parameter values for R_{pair}, R_{PE}^{s}, and R_{PE}^{ns} determined above (while keeping R_{ME}^{s} and R_{ME}^{ns} set to the maximum reference value of $L/2$ = 11.76 bohr). In doing so, we found that the inclusion of orbital self-consistency leads to a negligible (<0.01%) variation in E_{xx}, which indicates that (i) there is excellent agreement between the PE and ME evaluations of the far-field (beyond R_{PE}^{s} and R_{PE}^{ns}) contribution to the wave function forces ($\tilde{D}_{xx}^{i}(\mathbf{r})$) and (ii) the default value of R_{pair} is sufficient to capture all relevant overlapping MLWF pairs in this system. In a non-self-consistent calculation, the ME only provides the boundary conditions for the PE and therefore has no direct effect on E_{xx}, provided that R_{ME}^{s} (R_{ME}^{ns}) is larger than R_{PE}^{s} (R_{PE}^{ns}) by the extent of the Laplacian stencil (i.e., $n \max_a\{\delta\xi_a\}$; see section IV.A). In a self-consistent calculation, however, R_{ME}^{s} and R_{ME}^{ns} govern the accuracy and cost of obtaining E_{xx} via the sparse evaluation of $\{\tilde{D}_{xx}^{i}(\mathbf{r})\}$ (see eqs 18 and 20 and the surrounding discussion); in other words, larger values for R_{ME}^{s} and R_{ME}^{ns} lead to more accurate E_{xx} values (via the convergence of the orbitals, which is self-consistently driven by $\{\tilde{D}_{xx}^{i}(\mathbf{r})\}$), although this is accompanied by a higher computational cost (as well as communication overhead and memory footprint) during the EXX calculation. To quantify this effect (and determine the appropriate default values for R_{ME}^{s} and R_{ME}^{ns}), we now study the convergence of $\{\tilde{D}_{xx}^{i}(\mathbf{r})\}$ with respect to these parameters.

IV.A.2. Convergence of the EXX Contribution to the Wave Function Forces. To study the convergence of $\{\tilde{D}_{xx}^{i}(\mathbf{r})\}$ and determine the default values for R_{ME}^{s} and R_{ME}^{ns}, use of the reference calculation performed above (in which all parameters were set to the largest possible values) is inconvenient as it lacks the flexibility to vary R_{ME}^{s} and R_{ME}^{ns} (as these parameters

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

must be larger than R_{PE}^{s} and R_{PE}^{ns} to provide the boundary conditions for the PE). Since the use of the default parameter values for R_{pair}, R_{PE}^{s}, and R_{PE}^{ns} (with R_{ME}^{s} and R_{ME}^{ns} each set to the maximum reference value of $L/2 = 11.76$ bohr) reproduces E_{xx}^{ref} with negligible error (i.e., to within 0.02%), we will base our convergence tests on this as our new reference calculation. As an initial test, $\{\tilde{D}_{xx}^{i}(\mathbf{r})\}$ (as a function of R_{ME}^{s} and R_{ME}^{ns}) will be assessed using the converged MLWFs and $\{\tilde{D}_{xx}^{i,ref}(\mathbf{r})\}$ obtained during this *new* reference calculation. Since the effects of self-consistency are neglected in this test (and will be investigated below), we can quickly assess the convergence of $\{\tilde{D}_{xx}^{i}(\mathbf{r})\}$ with respect to $\{\tilde{D}_{xx}^{i,ref}(\mathbf{r})\}$ as a function of R_{ME}^{s} and R_{ME}^{ns}. We do so by independently varying the R_{ME}^{s} and R_{ME}^{ns} parameters, while keeping all other parameters (R_{pair}, R_{PE}^{s}, and R_{PE}^{ns}) fixed at their default values (see Figure 7). By employing this new reference

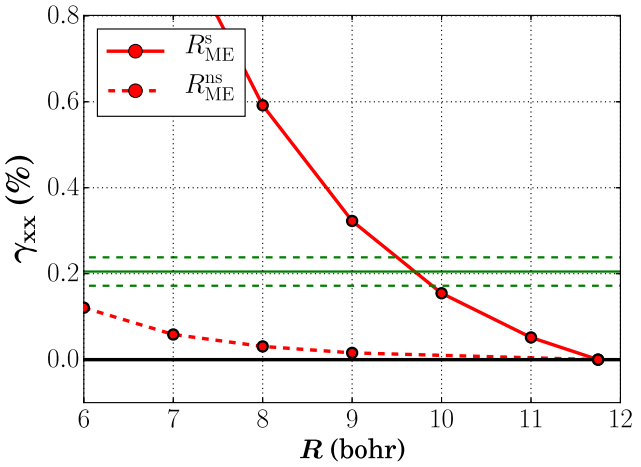


Figure 7. Convergence of $\{\tilde{D}_{xx}^{i}(\mathbf{r})\}$ (via the γ_{xx} metric defined in eqs 36 and 37) as a function of R_{ME}^{s} and R_{ME}^{ns} on a snapshot of liquid water containing $(H_2O)_{64}$ in a cubic cell with $L = 23.52$ bohr. Relative errors (in %) with respect to $\{\tilde{D}_{xx}^{i,ref}(\mathbf{r})\}$ are evaluated by varying R_{ME}^{s} (solid red line) and R_{ME}^{ns} (dashed red line), while keeping all other parameters (R_{pair}, R_{PE}^{s}, R_{PE}^{ns}) set to their default values in QE (see text for more details). An overall relative error of ≈0.2% (with a standard deviation of ±0.03%) corresponds to the default parameter values in QE (R_{ME}^{s} = 10.0 bohr, R_{ME}^{ns} = 7.0 bohr) and is depicted by the solid (dashed) green line.

calculation, we can now vary R_{ME}^{s} from 7.65 bohr (i.e., $R_{PE}^{s} + n \max_a \delta\xi_a$) to a maximum value of $L/2 = 11.76$ bohr and can vary R_{ME}^{ns} from 5.65 bohr (i.e., $R_{PE}^{ns} + n \max_a \delta\xi_a$) to a maximum value of $L/2 = 11.76$ bohr. To quantify the error in $\{\tilde{D}_{xx}^{i}(\mathbf{r})\}$, we will utilize the relative L^1-norm for each $\tilde{D}_{xx}^{i}(\mathbf{r})$:

$$\gamma_{xx}^{i} \equiv \frac{\int d\mathbf{r}\, |\tilde{D}_{xx}^{i,ref}(\mathbf{r}) - \tilde{D}_{xx}^{i}(\mathbf{r})|}{\int d\mathbf{r}\, |\tilde{D}_{xx}^{i,ref}(\mathbf{r})|} \tag{36}$$

which can then be averaged over MLWFs to furnish the following convergence metric:

$$\gamma_{xx} = \frac{1}{N_o} \sum_i \gamma_{xx}^{i} \tag{37}$$

From Figure 7, one can again see that the relative (percent) error in $\{\tilde{D}_{xx}^{i}(\mathbf{r})\}$ rapidly decays as each of these parameter values is increased. Since γ_{xx} converges more quickly with R_{ME}^{ns} (than R_{ME}^{s}), this finding also confirms our physical intuition that $R_{ME}^{s} > R_{ME}^{ns}$ and further justifies our use of separate self- and non-self-proto-subdomains as a means to improving the balance between performance and accuracy in this algorithm. As discussed above in sections II.C and IV.A, this results from the fact that self MLWF-product potentials ($\tilde{v}_{ii}(\mathbf{r})$) are generally longer-ranged than non-self-MLWF-product potentials ($\tilde{v}_{ij}(\mathbf{r})$) due to the absence of a monopolar contribution in the non-self cases. Since the cost of our algorithm (i.e., computation, communication, and memory) is dominated by the non-self contributions and scales cubically with R_{ME}^{ns}, it is preferable to choose the smallest possible value for this parameter. The plots depicted in Figure 7 clearly suggest that R_{ME}^{ns} is converged for $R \gtrsim 7.0$ bohr and (the noticeably slower) R_{ME}^{s} only begins to plateau for $R \gtrsim 10.0$ bohr. When used in conjunction with the default parameter values for R_{pair}, R_{PE}^{s}, and R_{PE}^{ns} determined above, parameter values of R_{ME}^{s} = 10.0 bohr and R_{ME}^{ns} = 7.0 bohr lead to an overall relative error of γ_{xx} ≈ 0.2% for this snapshot of liquid water. To see how this $\gamma_{xx} \approx$ 0.2% error translates into the final value of E_{xx}, we again performed a fully self-consistent EXX calculation on this $(H_2O)_{64}$ snapshot; in doing so, we found that the use of these parameter values leads to a completely negligible error (≈10^{-7}%) in E_{xx}. We further note that a number of EXX-based CPMD simulations of solid and liquid aqueous systems have been performed by our group using these parameter values; in all cases, we have found that the appropriate constant of motion was reasonably maintained. As such, we have set R_{ME}^{s} = 10.0 bohr and R_{ME}^{ns} = 7.0 bohr (in addition to R_{pair} = 8.0 bohr, R_{PE}^{s} = 6.0 bohr, and R_{PE}^{ns} = 5.0 bohr) as the default parameters used in QE.

As seen above for R_{PE}^{s}, one can further reduce γ_{xx} by an additional factor of 2 (i.e., to ≈0.1%) by increasing R_{ME}^{s} from 10.0 to 11.0 bohr, with negligible (<1%) additional computational cost; for sufficiently large simulation cells, increasing R_{ME}^{s} is therefore another efficient way to improve the accuracy of the EXX calculation. Even with R_{ME}^{s} = 11.0 bohr, however, one can still observe a finite slope in the tail of the R_{ME}^{s} curve in Figure 7. This is an artifact of performing this convergence test on $(H_2O)_{64}$, and a stricter convergence of γ_{xx} with R_{ME}^{s} would be observed with a larger simulation cell. To estimate the effect of this artifact on γ_{xx}, we performed an exponential fit (with R^2 > 0.9999) to the R_{ME}^{s} curve and found that the residual error in γ_{xx} (i.e., that was caused by truncating R_{ME}^{s} to $L/2$ = 11.76 bohr) is ≈0.1%. On the basis of the self-consistency test described above, we expect that this residual error in γ_{xx} would only lead to a negligible (≈10^{-7}%) error in E_{xx} for our $(H_2O)_{64}$ test system; when treating similarly sized (or slightly smaller) systems, we recommend that users quantify this truncation error and potentially set R_{ME}^{s} to the largest possible value (i.e., $R_{ME}^{s} = L/2$).

IV.A.3. Transferability of the Default EXX Parameters. To provide an alternative gauge for the 0.02% and 0.2% error thresholds in E_{xx} and $\{\tilde{D}_{xx}^{i}(\mathbf{r})\}$, we also considered the errors in the binding energy and ionic forces in this $(H_2O)_{64}$ snapshot by comparing the results obtained with the default and reference (i.e., $R_{pair} = R_{ME}^{s} = R_{ME}^{ns} = L/2 = 11.76$ bohr and $R_{PE}^{s} = R_{PE}^{ns} = 11.11$ bohr; see section IV.A) parameter sets. For the binding energy (per H_2O molecule), we found an error of 0.04 kcal/mol using the default parameters in `exx`, which is comparable to the typical pseudopotential error. For the $3N$ ionic force components (with N being the number of atoms), the default parameter set leads to a mean absolute error of 6.2 × 10^{-5} Ha/bohr and a maximum absolute error of 2.5 × 10^{-4}

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

Ha/bohr, which is approximately half of the default convergence criteria used during structural/geometry optimizations in QE. Taken together, all of these tests strongly indicate that the default parameter values in exx are more than adequate for EXX calculations on systems like liquid water.

We note in passing that our choice to treat self ($\langle ii \rangle$) and overlapping non-self ($\langle ij \rangle$) MLWF pairs differently using the $\Theta(\mathbf{C}_0,R^{s}_{PE})$ and $\Theta(\mathbf{C}_0,R^{ns}_{PE})$ proto-subdomains is only the first step toward exploiting the concept of variable-size subdomains during MLWF-based EXX calculations. By use of a single set of R^{s}_{PE} and R^{ns}_{PE} values, this choice is particularly well-suited for condensed-phase systems characterized by a narrow distribution of MLWF spreads (e.g., liquid water, wherein each H_2O molecule has a set of four similarly localized MLWFs). As such, we expect that the chosen default parameter values determined above for bulk liquid water will yield similar errors in E_{xx} and $\{\tilde{D}^{i}_{xx}(\mathbf{r})\}$ for systems with similarly large band gaps. For systems with smaller gaps (and hence more diffuse MLWFs), one would need to use more stringent parameter values to obtain a similar level of accuracy; as such, a series of test calculations (in analogy to those above) should be run to determine the optimal R^{s}_{PE} and R^{ns}_{PE} values prior to performing large-scale production AIMD simulations. For systems with a wider distribution of MLWF spreads (due to a smaller band gap and/or a more heterogeneous environment), our current algorithm would be forced to sacrifice computational efficiency for accuracy (vide infra), since R^{s}_{PE} and R^{ns}_{PE} (as well as R^{s}_{ME} and R^{ns}_{ME}) would need to be large enough to provide sufficient cover for the most diffuse MLWF in the system (and would therefore be overkill for MLWFs with substantially smaller spreads). As pointed out by Dawson and Gygi,[91] this issue is particularly important in small-gap heterogeneous condensed-phase systems, such as solvated semiconducting nanoparticles and water−semiconductor interfaces.

Although the current implementation of exx would sacrifice efficiency when applied to such challenging cases, we would still argue that the algorithmic framework of exx is general enough to perform accurate hybrid DFT-based CPMD simulations of finite-gap systems. While certain hacks can be used to ameliorate the efficiency degradation in these cases, we have chosen to focus on a comprehensive revision of our algorithm that will explicitly account for the MLWF-orbital (Ω_i) and MLWF-product (Ω_{ij}) domains introduced in section II.C. Inspired by the work of Gygi and co-workers,[79,80,91] we are currently working on a significantly more efficient (vide infra) β-version of exx in which each MLWF will have a spread-dependent Ω_i and each overlapping $\langle ij \rangle$ pair will have an overlap-dependent Ω_{ij}. As such, MLWFs with larger spreads will automatically be treated more accurately without sacrificing computational efficiency for MLWFs with smaller spreads. In addition, more distant MLWF pairs will have smaller MLWF-product domains by construction due to the overlap dependence in $\Omega_{ij} = \Omega_i \cap \Omega_j$. In doing so, we expect that the β-version of exx will be a single EXX algorithm that is general enough to accurately and efficiently handle condensed-phase systems ranging from large-gap homogeneous systems like liquid water (with a narrow distribution of MLWF spreads) to small-gap heterogeneous systems like solvated semiconducting nanoparticles (with a wide distribution of MLWF spreads). When exx is paired with an orbital localization scheme that can appropriately treat small-to-vanishing band gap systems,[113,114] we expect that the β-version of this algorithm will also be an important step toward treating large-scale metallic systems with screened and/or range-separated exchange.[49,62,132−136]

IV.A.4. Tight Convergence to the Electronic Ground State. While exx is designed to perform efficient AIMD simulations, this module can also be adapted to evaluate precise ground-state energetics (e.g., to within an uncertainty of $\Delta E < 10^{-8}$ Ha), which are needed for property evaluations, numerical phonon calculations, etc. In order to achieve tight convergence to the electronic ground state using exx, the default convergence criteria used during the CG solution to the PE (exx_poisson_eps = 10^{-6} au) and the nested SODD optimization of the Marzari−Vanderbilt functional (tolw = 10^{-8} au) must be tightened accordingly. Doing so minimizes the noise in the force acting on $\{\tilde{\phi}_i(\mathbf{r})\}$ ($\{\tilde{D}_i(\mathbf{r})\}$ in eq 26) and reduces oscillatory behavior in the energy profile during the SODD-based SCF procedure.

IV.B. Parallel Scaling and Performance. As demonstrated above in section IV.A, judicious choices for the five parameter values in exx allow one to evaluate all EXX-related quantities with a high level of accuracy. To enable large-scale EXX-based AIMD simulations using this approach, we employ a hybrid MPI/OpenMP parallelization scheme that allows us to minimize the wall time cost (i.e., time to solution) by exploiting both internode and intranode computational resources provided by massively parallel supercomputer architectures (see section III). Here, we remind the reader that our massively parallel implementation of exx seamlessly distributes the major computational workload across thousands of MPI processes. Within each MPI process, the CG-based PE solver is further parallelized over OpenMP threads. The scaling and performance of the internode (MPI, first level) and intranode (OpenMP, second level) levels in our hybrid parallelization scheme were evaluated by performing large-scale simulations of liquid water with exx on the IBM Blue Gene/Q platform (*Mira*), and will be discussed below in sections IV.B.1 and IV.B.2, respectively. In section IV.B.3, the computational performance of exx will also be considered on the *Cori* Haswell and KNL architectures.

IV.B.1. Internode Parallelization via MPI. For the first level of parallelization, the exx module employs internode MPI communication to distribute the computational workload associated with a given EXX calculation across (many) thousands of compute nodes. To critically assess the computational performance of this parallelization level, which is at the very heart of our massively parallel algorithm, we performed a strong-scaling analysis (i.e., by varying the number of processing elements for a fixed problem size) and a weak-scaling analysis (i.e., by varying the problem size for a fixed ratio of problem size to number of processing elements). To investigate the strong and weak scaling of exx, we performed a series of 12 different EXX-based CPMD simulations of liquid water, in which (i) the problem (system) size was varied to include N_{water} = 64, 128, 256 water molecules (each with $N_o = 4 \times N_{water}$ MLWFs) and (ii) the number of processing elements (N_{proc} MPI processes) was varied via $\zeta = N_{proc}/N_o = 1/2, 1, 2, 4$. In these calculations, we used one MPI process per node on the *Mira* IBM Blue Gene/Q platform and used all of the 64 hyperthreads available per node for the intranode OpenMP parallelization.

Initial structures for $(H_2O)_{64}$, $(H_2O)_{128}$, and $(H_2O)_{256}$ were systematically generated using the following procedure: (i) randomly packing N_{water} = 64, 128, 256 water molecules into

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

Table 1. Computational Timings Profile for CPMD Simulations of Liquid Water at the Hybrid PBE0 Level on the *Mira* IBM Blue Gene/Q Platform Using the `exx` Module in `QE`[a]

parameters					QE module timings					breakdown of $\langle t_{exx}\rangle$					
N_{water}	N_o	N_{proc}	ζ	N_{tg}	$\langle t_{GGA}\rangle$	$\langle t_{MLWF}\rangle$	$\langle t_{exx}\rangle$	$\langle t_{total}\rangle$	$\langle t_{exx}\rangle/\langle t_{GGA}\rangle$	$\langle t_{exx}^{comp}\rangle$	(f_{exx}^{comp})	$\langle t_{exx}^{comm}\rangle$	(f_{exx}^{comm})	$\langle t_{exx}^{idle}\rangle$	(f_{exx}^{idle})
64	256	128	1/2	1	2.81	0.16	7.34	10.31	2.6	4.07	(55.4)	0.96	(13.1)	2.31	(31.5)
64	256	256	1	1	1.97	0.17	3.83	5.97	1.9	2.05	(53.4)	0.52	(13.5)	1.27	(33.1)
64	256	512	2	2	1.02	0.16	2.74	3.92	2.7	1.06	(38.9)	0.39	(14.3)	1.28	(46.8)
64	256	1024	4	4	0.63	0.16	1.70	2.49	2.7	0.54	(32.0)	0.37	(21.6)	0.79	(46.5)
128	512	256	1/2	1	5.19	1.43	8.27	14.89	1.6	4.60	(55.6)	1.21	(14.7)	2.46	(29.8)
128	512	512	1	2	2.64	0.43	4.35	7.42	1.7	2.35	(54.1)	0.64	(14.8)	1.36	(31.2)
128	512	1024	2	4	1.57	0.41	3.04	5.02	1.9	1.25	(41.0)	0.51	(16.9)	1.28	(42.1)
128	512	2048	4	8	0.96	0.41	1.96	3.33	2.0	0.67	(34.4)	0.48	(24.8)	0.80	(40.9)
256	1024	512	1/2	2	6.39	2.77	8.34	17.50	1.3	4.19	(50.2)	1.58	(18.9)	2.57	(30.8)
256	1024	1024	1	4	3.59	1.20	4.80	9.59	1.3	2.23	(46.5)	1.11	(23.2)	1.46	(30.4)
256	1024	2048	2	8	2.23	1.13	3.33	6.69	1.5	1.26	(38.0)	0.76	(22.9)	1.30	(39.1)
256	1024	4096	4	16	1.59	1.08	2.41	5.08	1.5	0.77	(31.9)	0.82	(34.2)	0.82	(33.9)

[a]Parameters include: (i) the system size, which was varied to include N_{water} = 64, 128, 256 water molecules (each with $N_o = 4 \times N_{water}$ MLWFs); (ii) the number of `MPI` processes (N_{proc}), which was varied to cover $\zeta = N_{proc}/N_o = 1/2, 1, 2, 4$; and (iii) the level of task-group parallelization in `QE`, which was varied to include N_{tg} = 1, 2, 4, 8, 16 task groups (see below and text for more details). All timings have been averaged over 50 CPMD steps and are reported (in s/step) for the following `QE` modules: $\langle t_{GGA}\rangle$, the wall time associated with the underlying GGA calculation; $\langle t_{MLWF}\rangle$, the wall time associated with optimizing the Marzari−Vanderbilt functional (i.e., to localize the MLWFs between CPMD steps as shown in eq 29); $\langle t_{exx}\rangle$, the wall time spent in the `exx` module; and $\langle t_{total}\rangle$, the total wall time associated with a given CPMD step.[145] To gauge the reproducibility of these timings, each CPMD simulation was run in triplicate; the observed fluctuations were always smaller than the precision reported (i.e., $< 10^{-2}$ s) and were not included in the table. Also shown is the ratio, $\langle t_{exx}\rangle/\langle t_{GGA}\rangle$, demonstrating that the wall times required to perform EXX-based CPMD simulations with the `exx` module are approximately 1−3× that of the underlying GGA. All $\langle t_{exx}\rangle$ timings were further broken down into the wall time dedicated to computation events ($\langle t_{exx}^{comp}\rangle$), communication overhead ($\langle t_{exx}^{comm}\rangle$), and processor idling ($\langle t_{exx}^{idle}\rangle$); the corresponding fractions of the $\langle t_{exx}\rangle$ wall time (f_{exx}^{comp}, f_{exx}^{comm}, and f_{exx}^{idle}) were reported as percentages (%). All timings reflect the fact that one `MPI` process was executed per node on *Mira*, and all 16 physical cores (up to 64 hyperthreads) per node were used for the intranode `OpenMP` parallelization. To utilize all available `MPI` processes during the underlying GGA calculation, $N_{tg} = 2^n \geq 1$ was set to the maximum possible value such that $N_{tg} \times N_{slab} \leq N_{proc}$ (see text for more details).

simple cubic unit cells with lattice parameters chosen to match the targeted density of 0.993 g/cm^3; (ii) equilibrating each of these randomly packed structures via MD simulations in the *NVT* ensemble using the `TIP4P2005` force field[146] at 300 K for 1.0 ns in `GROMACS`;[147] (iii) further equilibrating each of the `TIP4P2005` structures via CPMD simulations in the *NVT* ensemble using the PBE0 hybrid xc functional[58,142] at 330 K for ≈25 fs (i.e., 500 CPMD steps). The computational timings reported herein are meant to reflect the wall time spent during EXX-based CPMD simulations in the *NVT* ensemble and were therefore averaged over 50 additional CPMD steps starting from the equilibrated structures obtained using this three-step procedure. Here we note that the `TIP4P2005` MD simulations performed in step ii were used to equilibrate the *intermolecular* degrees of freedom in these systems, and the additional CPMD simulations performed in step iii were used to ensure that the *intramolecular* degrees of freedom (i.e., the OH bonds and HOH angles) were equilibrated at the PBE0 level. Since the temperature of these systems will rapidly increase once the rigid-molecule `TIP4P2005` constraint is lifted, this additional CPMD equilibration step is important when a representative average is determined for the wall time cost during EXX-based CPMD simulations in the *NVT* ensemble. During the CPMD simulations (i.e., the 500 equilibration steps and subsequent 50 production steps), the temperature (of the ions) was controlled using massive Nosé−Hoover thermostats, each with a chain length of 4.[148,149] The nuclear and electronic degrees of freedom were integrated using the standard Verlet algorithm and a time step of 2.0 au (≈0.05 fs); to ensure a clear adiabatic separation between the electronic and nuclear degrees of freedom during the `CP` dynamics, we used a fictitious electronic mass of 100 au and the nuclear mass of deuterium for each hydrogen atom. Interactions between the valence electrons and the ions (consisting of the nuclei and their corresponding frozen-core electrons) were treated using the Hamann-Schlüter−Chiang−Vanderbilt (HSCV) type norm-conserving pseudopotentials[150,151] distributed with the `Qbox` package.[152] The valence electronic pseudowave functions were expanded in a planewave basis set that includes planewaves with a kinetic energy up to 85 Ry. Mass preconditioning was applied to all Fourier components of the electronic pseudowave functions with a kinetic energy above 25 Ry.[140] To enable distributed storage of all real-space quantities according to the `GRID` data distribution scheme in `QE` (see Figure 3), the real-space (and simple-cubic) grids utilized in these calculations were partitioned into N_{slab} = 140, 176, 220 slabs along the *z*-direction for $(H_2O)_{64}$, $(H_2O)_{128}$, and $(H_2O)_{256}$, respectively. All computational timings were generated using an in-house development version of `QE` (based on v5.0.2).[153]

Computational timings for each of these 12 CPMD simulations of liquid water at the hybrid PBE0 level on *Mira* are presented in Table 1. In this table, all timings have been averaged over 50 CPMD steps and are reported (in s/step) for the following four `QE` modules: (i) the wall time associated with the underlying GGA calculation ($\langle t_{GGA}\rangle$); (ii) the wall time associated with MLWF localization between each CPMD step via nested SODD optimization of the Marzari−Vanderbilt functional ($\langle t_{MLWF}\rangle$; see eq 29); (iii) the wall time spent in the `exx` module ($\langle t_{exx}\rangle$; see Figure 2); and (iv) the total wall time associated with a given CPMD step ($\langle t_{total}\rangle$).[145] To ensure a fair comparison between $\langle t_{GGA}\rangle$ and $\langle t_{exx}\rangle$, the underlying GGA calculation utilized all `MPI` processes available to the `exx` module. This was accomplished using an existing two-tier

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

parallelization scheme in QE, which allows for a computationally efficient execution of the fwdFFT/invFFT operation (i.e., a typical bottleneck during GGA-based CPMD simulations). The first parallelization tier takes advantage of the fact that the real-space grid has been partitioned into N_{slab} slabs along the z-direction (with each slab distributed to a particular MPI process); this allows one to split the 3D fwdFFT and invFFT operations into 2D intraslab FFT operations (which are executed in parallel without the need for additional communication) and 1D interslab FFT operations (which are also executed in parallel but require communication among the pool of MPI processes). At the first tier, the underlying GGA calculation can utilize up to $N_{proc} = N_{slab}$ MPI processes. To enable the use of $N_{proc} > N_{slab}$ MPI processes (when available), the second parallelization tier (i.e., task-group parallelization) is employed to further distribute the independent 3D FFT operations associated with the N_o orbitals. At the second tier, $N_{tg} = 2^n \geq 1$ can be set to the maximum possible value such that $N_{tg} \times N_{slab} \leq N_{proc}$, thereby enabling the underlying GGA calculation to utilize up to $N_{proc} = N_{tg} \times N_{slab}$ MPI processes. We note in passing that the scalability of task-group parallelization depends on the communication bandwidth and will often deteriorate when $N_{tg} \gg 4$. With this approach, the underlying GGA calculation is able to utilize the pool of available MPI processes, thereby ensuring a reasonably fair comparison between the $\langle t_{GGA} \rangle$ and $\langle t_{exx} \rangle$ timings.

In the QE module timings in Table 1, we observed that the exx module is still the overall bottleneck during hybrid DFT-based CPMD simulations. When both the natural sparsity of the exchange interaction and our massively parallel implementation are exploited, the wall time cost to evaluate all EXX-related quantities is comparable to that of the underlying GGA (i.e., $\langle t_{exx} \rangle / \langle t_{GGA} \rangle$ is now within the range ≈1−3). We further stress that this ratio steadily decreases with increasing system size due to the more favorable scaling of the exx module (see below). Since the exx module requires MLWFs (and the underlying GGA does not), we now discuss the additional cost needed to perform the nested SODD optimization of the Marzari−Vanderbilt functional ($\langle t_{MLWF} \rangle$) between each CPMD step. In all CPMD simulations performed herein, this MLWF refinement procedure (see section III.A) only required three to four SODD steps (on average) per CPMD step. As such, $\langle t_{MLWF} \rangle$ only represents a minor contribution to $\langle t_{total} \rangle$ for systems containing <500 atoms (e.g., $(H_2O)_{64}$ and $(H_2O)_{128}$). For larger systems (e.g., $(H_2O)_{256}$), $\langle t_{MLWF} \rangle$ can become quite substantial (≈10−20% of $\langle t_{total} \rangle$) as the MLWF procedure requires cubic-scaling matrix operations. As such, a more efficient MLWF localization procedure (which takes advantage of the sparsity of the MLWFs) will be required to efficiently utilize the exx module for system sizes that are significantly larger than $(H_2O)_{256}$.

On the basis of the timings in Table 1, we assessed the strong-scaling behavior of the exx module by analyzing how $\langle t_{exx} \rangle$ changes as the number of processing elements was varied for a fixed problem size. This was accomplished by changing $\zeta = N_{proc}/N_o = 1/2, 1, 2, 4$ when $(H_2O)_{64}$, $(H_2O)_{128}$, and $(H_2O)_{256}$ were simulated (see Figure 8). For each system size, we computed the strong-scaling efficiency via

$$\eta_{MPI}^{strong}(\zeta) \equiv \frac{\zeta_{ref} \cdot \langle t_{exx} \rangle_{\zeta_{ref}}}{\zeta \cdot \langle t_{exx} \rangle_{\zeta}} = \frac{\frac{1}{2} \cdot \langle t_{exx} \rangle_{\zeta=1/2}}{\zeta \cdot \langle t_{exx} \rangle_{\zeta}} \quad (38)$$

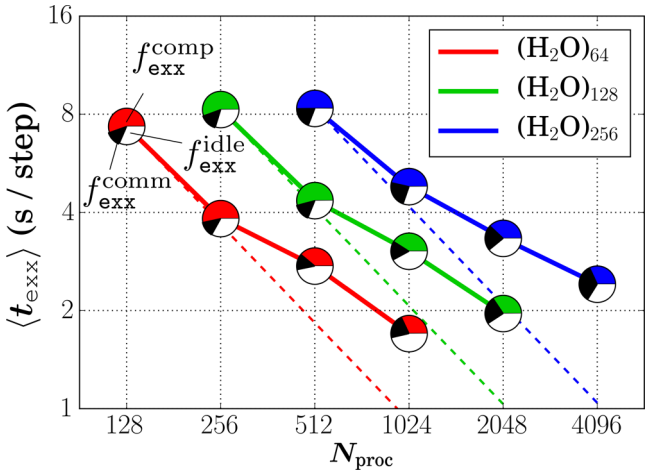


Figure 8. Strong-scaling analysis of the MPI internode parallelization level in exx during CPMD simulations of liquid water at the hybrid PBE0 level on the *Mira* IBM Blue Gene/Q platform. For a fixed system size (i.e., $(H_2O)_{64}$ (red line), $(H_2O)_{128}$ (green line), and $(H_2O)_{256}$ (blue line)), the wall time spent in the exx module ($\langle t_{exx} \rangle$ in s/step, averaged over 50 CPMD steps) is plotted versus the number of MPI processes (N_{proc}), which were varied to include $\zeta = N_{proc}/N_o = 1/2, 1, 2, 4$. For reference, ideal strong-scaling timings were plotted as dashed lines for each system size and were computed with respect to the corresponding $\zeta_{ref} = 1/2$ timings. Pie plots were used to illustrate the fraction of $\langle t_{exx} \rangle$ dedicated to computation events (f_{exx}^{comp}, colored), communication overhead (f_{exx}^{comm}, black), and processor idling (f_{exx}^{idle}, white).

in which $\zeta_{ref} = 1/2$ was chosen as the reference (or baseline) ζ value (as this represents a realistic computational setup) and $\langle t_{exx} \rangle_{\zeta}$ is the wall time spent in the exx module for a given ζ. When the efficiency is averaged over all three systems, we find that η_{MPI}^{strong} decreases to ≈93% ($\zeta = 1$), ≈66% ($\zeta = 2$), and ≈50% ($\zeta = 4$) as the number of processing elements is increased (see below for a more detailed discussion). For even higher ζ, the number of MPI processes becomes comparable to the number of overlapping $\langle ij \rangle$ pairs; as such, η_{MPI}^{strong} is expected to deteriorate even further for $\zeta \gg 4$.

On the basis of the timings in Table 1, we also assessed the weak-scaling behavior of the exx module by analyzing how $\langle t_{exx} \rangle$ changes as the problem (system) size was varied for a fixed ratio of problem size to number of processing elements. This was accomplished by considering $(H_2O)_{64}$, $(H_2O)_{128}$, and $(H_2O)_{256}$ for fixed values of $\zeta = N_{proc}/N_o \in \{1/2, 1, 2, 4\}$ (see Figure 9). For each ζ value, we computed the weak-scaling efficiency via

$$\eta_{MPI}^{weak}(N_{water}) \equiv \frac{\langle t_{exx} \rangle_{N_{water}^{ref}}}{\langle t_{exx} \rangle_{N_{water}}} = \frac{\langle t_{exx} \rangle_{N_{water}=64}}{\langle t_{exx} \rangle_{N_{water}}} \quad (39)$$

in which $N_{water}^{ref} = 64$ was chosen as the reference (or baseline) system size (as this represents a realistic computational setup) and $\langle t_{exx} \rangle_{N_{water}}$ is the wall time spent in the exx module for a given N_{water}. When the efficiency is averaged over all four ζ values, we find that η_{MPI}^{weak} decreases to ≈89% ($N_{water} = 128$) and ≈81% ($N_{water} = 256$) as the system size is increased. As shown in Figure 9, the exx module is quite scalable as the system size is increased, and the time-to-solution can be kept (relatively) constant for systems as large as $(H_2O)_{256}$ provided that a consistent (i.e., fixed ζ) amount of computational resources are available (see below for a more detailed discussion).

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

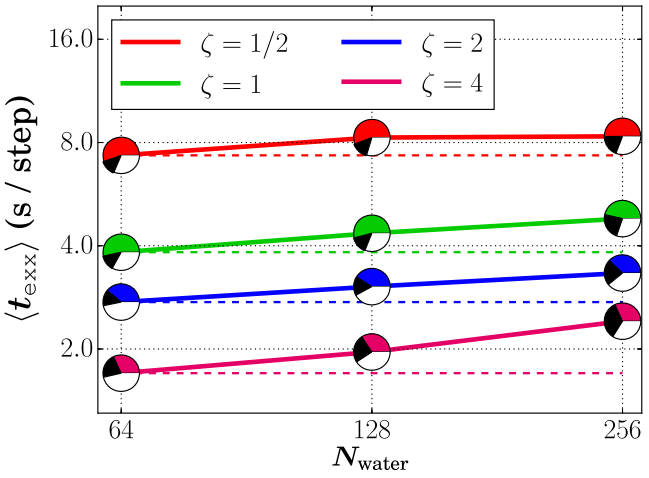


Figure 9. Weak-scaling analysis of the MPI internode parallelization level in exx during CPMD simulations of liquid water at the hybrid PBE0 level on the *Mira* IBM Blue Gene/Q platform. For a fixed ratio of system size to number of processing elements (i.e., $\zeta = 1/2$ (red line), $\zeta = 1$ (green line), $\zeta = 2$ (blue line), and $\zeta = 4$ (magenta line)), the wall time spent in the exx module ($\langle t_{exx} \rangle$ in s/step, averaged over 50 CPMD steps) is plotted versus the system size, which was varied to include $N_{water} = 64$, 128, 256 water molecules. For reference, ideal weak-scaling timings were plotted as dashed lines for each ζ and were computed with respect to the corresponding $(H_2O)_{64}$ timings. Pie plots were again used to illustrate the fraction of $\langle t_{exx} \rangle$ dedicated to computation events (f_{exx}^{comp}, colored), communication overhead (f_{exx}^{comm}, black), and processor idling (f_{exx}^{idle}, white).

Despite the fact that the strong- and weak-scaling efficiencies of the exx module are not perfect, our algorithm is still able to furnish all EXX-related quantities in ≈2.4 s for the largest system considered herein, i.e., $(H_2O)_{256}$ with $\zeta = 4$. As such, the exx module in QE enables relatively long (e.g., 10−100 ps) CPMD simulations for large-scale condensed-phase systems consisting of 500−1000 atoms at the hybrid DFT level of theory. Quite interestingly, the overall cost of the exx module in this case (see Table 1 and Figures 8 and 9) can be decomposed into roughly equal contributions from computation ($f_{exx}^{comp} \equiv \langle t_{exx}^{comp} \rangle / \langle t_{exx} \rangle \approx 1/3$), communication ($f_{exx}^{comm} \equiv \langle t_{exx}^{comm} \rangle / \langle t_{exx} \rangle \approx 1/3$), and idling ($f_{exx}^{idle} \equiv \langle t_{exx}^{idle} \rangle / \langle t_{exx} \rangle \approx 1/3$). This breakdown of $\langle t_{exx} \rangle$ demonstrates that the exx algorithm is *not* computation bound (as one might expect for the relatively large number of computation events required for hybrid DFT). As discussed below, there still remains significant room for algorithmic improvements that would combat the relatively high cost associated with the communication overhead and processor idling, both of which are currently under development by our group and will be the topic of future work. When exx is combined with state-of-the-art preconditioners during the CG solution of the PE (which are also under intense development by our group), the computational cost of the current exx algorithm can be significantly sped up, which would further enable hybrid DFT-based AIMD simulations of large-scale condensed-phase systems across sufficiently longer time scales.

Computation Events. When further breaking down the computation events in the exx module (see Figure 2) that contribute to $\langle t_{exx}^{comp} \rangle$, we find that the computational costs associated with Step IV (Solution of Poisson's Equation) and Step V (Computation of Energy and Forces) scale nearly ideally with N_{proc} (e.g., $\eta_{MPI}^{strong} > 90\%$) and N_{water} (e.g., $\eta_{MPI}^{weak} > 99\%$). However, the computational effort in Step II (Construction of Pair List) required for determining the unique pair list (see Figure 4 and section III.C.2) was implemented in serial (in the current version of the exx module) and does not scale with N_{proc}. In addition, the computational cost associated with Step II grows quadratically with system size ($O(N_o^2)$); although this step is quite cheap for smaller system sizes (e.g., $(H_2O)_{64}$ and $(H_2O)_{128}$), this cost can become more substantial for larger systems (e.g., $(H_2O)_{256}$). As a result, Step II (in its current form) leads to some of the deterioration (particularly for $(H_2O)_{256}$) seen in η_{MPI}^{strong} and η_{MPI}^{weak} (see Figures 8 and 9). In future versions of exx, we plan to mitigate this unnecessary computational cost by parallelizing Step II over MPI processes and using a Verlet list (which will be updated periodically throughout a given CPMD simulation) to avoid unnecessary consideration of distant MLWF pairs; as such, these improvements will increase both the strong- and weak-scaling efficiencies of exx. Although Step IV does scale nearly ideally with N_{proc} and N_{water}, the computational cost associated with solving the PE for each overlapping $\langle ij \rangle$ pair still remains the dominant contribution to $\langle t_{exx}^{comp} \rangle$. In the near future, we plan to significantly reduce this primary source of computational effort by using more sophisticated guesses (for $\tilde{v}_{ij}(\mathbf{r})$) in conjunction with novel preconditioners during the CG solution of the PE. An additional future direction to overcome this computational hurdle might also involve offloading all computation events (not necessarily limited to the solution of the PE) to graphical processing units (GPUs), which provide significantly higher computational throughput than CPUs.

Communication Overhead. In addition to the computation events described above, the communication overhead in Figure 2 also contributes to the degradation of η_{MPI}^{strong} and η_{MPI}^{weak} observed in Figures 8 and 9. Here, this nonideal scaling behavior mainly originates from (i) the sending/receiving of MLWFs ($\{\tilde{\phi}_i(\mathbf{r})\}$) in Step III (Communication of MLWFs) and the sending/receiving of wave function forces ($\{\tilde{D}_{xx}^i(\mathbf{r})\}$) at the conclusion of Step V (Computation of Energy and Forces) as well as (ii) the redistribution of $\{\tilde{\phi}_i(\mathbf{r})\}$ in Step I (Redistribution of MLWFs) and the redistribution of $\{\tilde{D}_{xx}^i(\mathbf{r})\}$ in Step VI (Redistribution of Wave Function Forces). In this regard, the former is more important for relatively smaller systems (e.g., $(H_2O)_{64}$ and $(H_2O)_{128}$) employing fewer processing elements (e.g., $\zeta = 1/2$ and $\zeta = 1$) due to the fact that each MPI process needs to send/receive significantly more MLWFs and wave function forces (see sections III.C.3 and III.C.5). In the same breath, the latter dominates for larger systems (e.g., $(H_2O)_{256}$) employing more processing elements (e.g., $\zeta = 2$ and $\zeta = 4$) due to the ALL-TO-ALL communication events in Steps I and VI (see Figure 3 and sections III.C.1 and III.C.6). As a result, these communication events lead to a noticeable upward tilt in the strong- and weak-scaling curves in Figures 8 and 9, which is particularly evident in the large system and ζ limit. To attack the communication overhead associated with the sending/receiving of $\{\tilde{\phi}_i(\mathbf{r})\}$ and $\{\tilde{D}_{xx}^i(\mathbf{r})\}$, we plan to implement an asynchronous (nonblocking) communication protocol that will overlap with the computation events in Steps IV and V. By doing so, the serial communication−computation−communication process in Steps III−V (i.e., communication of $\{\tilde{\phi}_i(\mathbf{r})\}$, followed by the solution of the PE for all overlapping pairs and computation of the $\langle ij \rangle$ contribution to E_{xx} and $\{\tilde{D}_{xx}^i(\mathbf{r})\}$, followed by

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

communication of $\{\tilde{D}_{xx}^{ij}(\mathbf{r})\}$) can be overlapped to effectively mask the communication overhead (see the right panel of Figure 4). To attack the communication overhead associated with the redistribution of $\{\tilde{\phi}_i(\mathbf{r})\}$ and $\{\tilde{D}_{xx}^{i}(\mathbf{r})\}$, we plan to exploit the locality of the MLWFs by only performing the redistribution on the basis of the compact supports of each MLWF. By doing so, this algorithmic improvement has the potential to completely eliminate the unnecessary `ALL-TO-ALL` communication events in Steps I and VI, which would significantly reduce the communication overhead in the `exx` module. As an added bonus, this approach would also allow for a more accurate evaluation of $\{\tilde{D}_{xx}^{ij}(\mathbf{r})\}$ on Ω_j via eq 20, thereby eliminating any residual error in the wave function forces (see Figure 7). In addition to the above strategies, we also plan to port the `exx` module to parallel GPU architectures (e.g., with NVLink technology), which will allow us to exploit faster peer-to-peer connections and further reduce the communication overhead.

Processor Idling. The last and most critical issue that limits the strong- and weak-scaling efficiency in the `exx` algorithm is processor idling due to workload imbalance. This imbalance mainly originates from (i) the imperfect distribution of overlapping $\langle ij \rangle$ pairs across the pool of `MPI` processes (see Figure 4 and section III.C.2) and (ii) the variability in the number of CG steps required during the solution of the PE for each overlapping $\langle ij \rangle$ pair. In the current `exx` module, these issues are primarily due to the static load-balancing algorithm described in section III.C.2, which assumes that the computational workload (i.e., the number of CG steps) associated with the solution of the PE is equivalent for each $\langle ij \rangle$ pair and limits this computation to the P_i or P_j `MPI` processes only (and not P_k for example). To attack the processor idling, we plan to remove these limitations by employing a task-based load-balancing algorithm that will account for the workload imbalance using a dynamic scheduler and has the flexibility to assign (and even reassign) a given $\langle ij \rangle$ task to any available `MPI` process. In addition to the load-balancing algorithm, we also plan to implement more intelligent initial guesses for $\tilde{v}_{ij}(\mathbf{r})$ (to aid in the convergence of the CG solution to the PE) as well as the aforementioned asynchronous communication protocol (which will overlap with the computational events in Step IV) and expect that both of these algorithmic improvements will also mitigate the processor idling in the `exx` module. The current `exx` module also faces challenges associated with processor idling when there is an inherent workload imbalance (in the number of overlapping pairs per MLWF) due to the physical nature of the problem, i.e., the (inherent and transient) heterogeneity present in systems containing interfaces as well as disordered systems (like liquids). To balance the workload in such heterogeneous systems, one could employ a different parallelization level for each MLWF (i.e., $\zeta = \zeta(i)$), which would allow the `exx` module to dynamically adopt the number of processing elements dedicated to a given MLWF on the basis of its number of overlapping $\langle ij \rangle$ pairs.

IV.B.2. Intranode Parallelization via `OpenMP`. Within each `MPI` process, the `exx` module uses `OpenMP` threading to further parallelize the following operations for each overlapping $\langle ij \rangle$ pair: (i) the CG solution of the PE for the near-field $\tilde{v}_{ij}(\mathbf{r})$ (Step IV in Figure 2), (ii) the multipole expansion for the far-field $\tilde{v}_{ij}(\mathbf{r})$ (Step IV), and (iii) other (less computationally intensive) operations (e.g., proto-subdomain construction in Step II, $\{\tilde{\phi}_i(\mathbf{r})\}$ loading/off-loading in Step III, and E_{xx} integration in Step V). To critically assess the strong-scaling performance of the intranode `OpenMP` parallelization (in analogy to the internode `MPI` parallelization in section IV.B.1), we analyzed how $\langle t_{exx}^{\text{Step IV}} \rangle$ (the typical computational bottleneck in the `exx` module) changes as the number of `OpenMP` threads (N_{thread}) was varied during a CPMD simulation of $(H_2O)_{64}$ with $\zeta = 1$. To maintain a consistent internode communication pattern, we used one `MPI` process per node on the *Mira* IBM Blue Gene/Q architecture; since each node contains 16 physical cores, five different levels of `OpenMP` parallelization were assessed by varying $N_{\text{thread}} \in \{1, 2, 4, 8, 16\}$ threads across $N_{\text{core}} = N_{\text{thread}}$ physical cores (see Figure 10). For each N_{thread} value, we computed the strong-scaling efficiency via

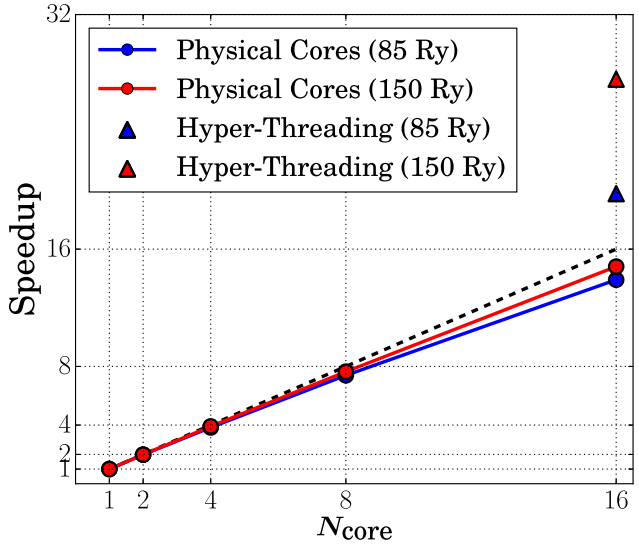


Figure 10. Strong-scaling analysis of the `OpenMP` intranode parallelization level in `exx` during CPMD simulations of $(H_2O)_{64}$ at the hybrid PBE0 level on the *Mira* IBM Blue Gene/Q platform. For a fixed system and basis set size (i.e., $(H_2O)_{64}$ with a planewave kinetic energy cutoff of 85 Ry (blue line) and 150 Ry (red line)), the speedup in the wall time spent in Step IV, Solution of Poisson's Equation, of Figure 2 ($\langle t_{exx}^{\text{Step IV}} \rangle$, averaged over 50 CPMD steps) is plotted versus the number of physical cores, which were varied to include $N_{\text{thread}} = N_{\text{core}} = 1, 2, 4, 8, 16$. Beyond the maximum number of physical cores ($N_{\text{core}} = 16$) per node on *Mira*, the `OpenMP` intranode parallelization level can further utilize hyperthreading technology to access up to $N_{\text{thread}} = 64$ (hyper)threads (depicted by the blue and red triangles). For reference, the ideal strong-scaling performance (using up to $N_{\text{core}} = 16$) was plotted as a dashed line and normalized to unity for $N_{\text{core}} = 1$.

$$\eta_{\text{OpenMP}}^{\text{strong}}(N_{\text{thread}}) \equiv \frac{\langle t_{exx}^{\text{Step IV}} \rangle_{N_{\text{thread}}=1}}{N_{\text{thread}} \cdot \langle t_{exx}^{\text{Step IV}} \rangle_{N_{\text{thread}}}} \tag{40}$$

in which $N_{\text{thread}} = N_{\text{core}} = 1$ was chosen as the reference (or baseline) `OpenMP` setting and $\langle t_{exx}^{\text{Step IV}} \rangle_{N\text{thread}}$ is the wall time spent in Step IV of the `exx` module for a given N_{thread}. Here, we find that the computational costs associated with Step IV scale very well with N_{thread} (e.g., $\eta_{\text{OpenMP}}^{\text{strong}} = 84\%$ when using all 16 physical cores) when using a typical constant-volume (*NVT*) planewave basis setting (i.e., 85 Ry kinetic energy cutoff). With a heavier workload (i.e., a typical constant-

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

Table 2. Computational Timings Profile for CPMD Simulations of Liquid Water at the Hybrid PBE0 Level on the *Mira* IBM Blue Gene/Q, *Cori* Haswell, and *Cori* KNL Platforms Using the `exx` Module in QE[a]

architecture		QE module timings					breakdown of $\langle t_{exx} \rangle$					
machine	CPU	$\langle t_{GGA} \rangle$	$\langle t_{MLWF} \rangle$	$\langle t_{exx} \rangle$	$\langle t_{total} \rangle$	$\langle t_{exx} \rangle / \langle t_{GGA} \rangle$	$\langle t_{exx}^{comp} \rangle$	(f_{exx}^{comp})	$\langle t_{exx}^{comm} \rangle$	(f_{exx}^{comm})	$\langle t_{exx}^{idle} \rangle$	(f_{exx}^{idle})
Mira	IBM Blue Gene/Q	2.64	0.43	4.35	7.42	1.7	2.35	(54.1)	0.64	(14.8)	1.36	(31.2)
Cori	Haswell	1.07	0.72	1.66	3.45	1.6	0.84	(50.8)	0.37	(22.2)	0.45	(27.0)
Cori	KNL	4.45	1.76	6.53	12.74	1.5	3.51	(53.6)	1.50	(23.0)	1.53	(23.4)
Cori	KNL (no hyperthreading)	5.38	1.15	3.57	10.10	0.7	1.84	(51.7)	1.03	(28.8)	0.70	(19.5)

[a]All timings (in s/step) correspond to an average over 50 CPMD steps for $(H_2O)_{128}$ with $\zeta = 1$ and $N_{tg} = 2$ (see Table 1 and the text for more details). All timings reflect the fact that one `MPI` process was executed per node on the given architecture, and all available physical cores per node were used for the intranode `OpenMP` parallelization; unless otherwise specified, hyperthreading was fully activated on each physical core.

pressure (NpT) planewave basis setting with a 150 Ry kinetic energy cutoff), we find that the strong-scaling efficiency of the `exx` module is significantly better and maintains a nearly ideal efficiency of $\eta^{strong}_{OpenMP} = 92\%$ when using all 16 physical cores. We note in passing that hyperthreading each physical core on *Mira* into four logical cores yields an additional boost (i.e., 30−40% speedup) in the computational performance of `exx`.

IV.B.3. Performance on Other Supercomputer Architectures. To provide an additional assessment of the performance of the `exx` module, we performed an analogous computational analysis on the *Cori* Haswell and KNL supercomputer architectures housed at the National Energy Research Scientific Computing Center (NERSC). For simplicity, we considered the $(H_2O)_{128}$ test case described above and limited our analysis to the most common $\zeta = 1$ case (in which $N_{proc} = N_o = 512$). For the `MPI` parallelization level, we employed one `MPI` process per node on the *Cori* Haswell and KNL architectures. For the `OpenMP` parallelization level, we used all physical cores on each node (i.e., 24 and 68 for the Haswell and KNL architectures, respectively). When specified, hyperthreading was fully activated on each physical core, which corresponds to a maximum total of 48 and 272 `OpenMP` threads for each Haswell and KNL node, respectively. As shown in Table 2, the `exx` module behaves quite consistently across all three architectures considered (i.e., *Mira* IBM Blue Gene/Q, *Cori* Haswell, and *Cori* KNL). Here, we first note that $\langle t_{exx} \rangle / \langle t_{GGA} \rangle$ is fairly constant and fluctuates between 1.5 and 1.7; as such, the `exx` module enables hybrid DFT calculations with a wall time cost that is comparable to semilocal DFT on all three architectures. We further note that the fractional breakdown of $\langle t_{exx} \rangle$ into computation, communication, and processor idling is also similar; as such, our comprehensive three-pronged strategy (vide supra) to attack each of these contributions is likely to lead to a robust `exx` module with significantly improved performance across a wide array of HPC architectures. When comparing $\langle t_{exx} \rangle$ across these architectures, we find that *Cori* Haswell (with 512 × 24 = 12, 288 physical cores) has a faster turnaround (by ≈2.6 times) than *Mira* IBM Blue Gene/Q (with 512 × 16 = 8, 192 physical cores), while *Cori* KNL (with 512 × 68 = 34, 816 physical cores) is noticeably slower (by ≈0.67 times). We note in passing that hyperthreading introduces noticeable performance improvements on the IBM Blue Gene/Q and Haswell architectures but leads to a significant decrease in the performance of `exx` on the KNL architecture. For instance, deactivating the hyperthreading option on KNL leads to an ≈80% speedup in $\langle t_{exx} \rangle$ and an ≈20% slowdown in $\langle t_{GGA} \rangle$; in doing so, $\langle t_{exx} \rangle / \langle t_{GGA} \rangle = 0.7$ and the EXX calculation is now faster than the corresponding GGA calculation.

V. CONCLUSIONS AND FUTURE OUTLOOK

In this work, we presented a detailed discussion of the theoretical framework, algorithmic implementation, and computational performance of a linear scaling approach that exploits sparsity in the real-space evaluation of the EXX interaction in finite-gap condensed-phase systems by utilizing a localized (MLWF) representation of the occupied orbitals. Our theoretical discussion focused on the integration of this approach into CPMD and highlighted the central role played by $\tilde{v}_{ij}(\mathbf{r})$—the MLWF-product potential obtained via the CG solution of Poisson's equation for the corresponding MLWF-product density $\tilde{\rho}_{ij}(\mathbf{r})$—in the evaluation of the EXX energy and wave function forces. We then provided a comprehensive description of the `exx` algorithm, which has been implemented in the `CP` module of the open-source `QE` package and employs a hybrid `MPI/OpenMP` parallelization scheme to efficiently utilize the HPC resources available on current- and next-generation supercomputer architectures. This was followed by a critical assessment of the accuracy and parallel performance (e.g., strong and weak scaling) of this approach when large-scale AIMD simulations of liquid water were performed in the canonical (NVT) ensemble. With access to HPC resources, we demonstrated that `exx` enables us to compute the EXX contribution to the energy and wave function forces for $(H_2O)_{256}$, a condensed-phase system containing ≈750 atoms, in just under 2.4 s. With a wall time cost that is comparable to semilocal DFT results, the `exx` module takes us one step closer to routinely performing AIMD simulations of complex and large-scale condensed-phase systems for sufficiently long time scales at the hybrid DFT level of theory.

In its current form, the `exx` module can also be used for high-throughput applications such as the generation of high-quality AIMD data for training, developing, and testing next-generation machine-learning and neural-network-based force fields.[123,154−156] Despite the favorable scalability and computational performance of `exx`, however, we found that this algorithm is not computation-bound for larger systems (such as $(H_2O)_{256}$) with an overall wall time that can be roughly split into three equivalent contributions: computation events, communication overhead, and processor idling (due to workload imbalance). As such, there still exits significant room for improving the performance of the `exx` module, and we are currently in the process of implementing a comprehensive three-pronged strategy that will attack each of these contributions and significantly reduce the overall wall time cost. Inspired by the work of Gygi and co-workers,[79,80,91] we are also implementing an MLWF-specific domain strategy (as outlined in sections II.C and IV.A.3) that will allow the β-

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

version of exx to perform accurate and efficient hybrid DFT simulations of condensed-phase systems ranging from large-gap homogeneous systems like liquid water (with a narrow distribution of MLWF spreads) to small-gap heterogeneous systems like solvated semiconducting nanoparticles (with a wide distribution of MLWF spreads).

In addition to improving the performance and generality of exx, our group is also in the process of extending this approach to perform BOMD simulations, as well as generalizing this MLWF-based framework to enable linear-scaling and highly accurate evaluations of screened and range-separated exchange.[49,62,132−136] Other improvements can also be straightforwardly incorporated into exx such as alternative localization schemes that are better suited to treat heterogeneous systems[80,91] and metals,[113,114] or furnish localized orbitals in a noniterative fashion to avoid convergence issues.[81−83,92] From a wave function theory point of view, exx is also a viable approach for the mean-field HF approximation and is a logical starting point for enabling condensed-phase AIMD with local electron correlation methods.

Interested users can find exx implemented in the most recent version of QE,[102] with additional information (including a detailed description of the exx keywords) available in the QE user's manual online.[157] In the next paper in this series, we will generalize our MLWF-based EXX approach to treat arbitrary Bravais-lattice-based simulation cells, as well as derive (and implement) the EXX contributions to the cell forces (i.e., the stress tensor). These extensions to exx are needed for performing constant-pressure AIMD simulations in the *NpH* and *NpT* ensembles and will enable us to model large condensed-phase systems under realistic thermodynamic conditions (i.e., at finite T and p) at the hybrid DFT level of theory.

AUTHOR INFORMATION

Corresponding Author

Robert A. DiStasio Jr. − *Department of Chemistry and Chemical Biology, Cornell University, Ithaca, New York 14853, United States;* orcid.org/0000-0003-2732-194X; Email: distasio@cornell.edu

Authors

Hsin-Yu Ko − *Department of Chemistry and Chemical Biology, Cornell University, Ithaca, New York 14853, United States; Department of Chemistry, Princeton University, Princeton, New Jersey 08544, United States;* orcid.org/0000-0003-1619-6514

Junteng Jia − *Department of Chemistry and Chemical Biology, Cornell University, Ithaca, New York 14853, United States*

Biswajit Santra − *Department of Chemistry, Princeton University, Princeton, New Jersey 08544, United States; Department of Physics, Temple University, Philadelphia, Pennsylvania 19122, United States;* orcid.org/0000-0003-3609-2106

Xifan Wu − *Department of Physics, Temple University, Philadelphia, Pennsylvania 19122, United States*

Roberto Car − *Department of Chemistry and Department of Physics, Princeton University, Princeton, New Jersey 08544, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.9b01167

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

H.-Y.K., J.J., and R.A.D. acknowledge partial support from Cornell University through start-up funding and the Center for Alkaline Based Energy Solutions (CABES), an Energy Frontier Research Center funded by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Award No. DE-SC0019445. H.-Y.K. and R.C. gratefully acknowledge support from the Chemistry in Solution and at Interfaces (CSI), a Computational Chemical Science Center funded by the U.S. Department of Energy under Grant No. DE-SC0019394. X.W. acknowledges support from National Science Foundation through Award No. DMR-1552287. This research used resources of the National Energy Research Scientific Computing (NERSC) Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-06CH11357. Additional resources were provided by the Terascale Infrastructure for Groundbreaking Research in Science and Engineering (TIGRESS) High Performance Computing Center and Visualization Laboratory at Princeton University.

REFERENCES

(1) Hohenberg, P.; Kohn, W. Inhomogeneous Electron Gas. *Phys. Rev.* **1964**, *136*, B864−B871.

(2) Kohn, W.; Sham, L. J. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* **1965**, *140*, A1133−A1138.

(3) Jones, R. O.; Gunnarsson, O. The Density Functional Formalism, Its Applications and Prospects. *Rev. Mod. Phys.* **1989**, *61*, 689−746.

(4) Parr, R. G.; Yang, W. *Density-Functional Theory of Atoms and Molecules*; Oxford University Press: New York, 1989.

(5) Burke, K. Perspective on Density Functional Theory. *J. Chem. Phys.* **2012**, *136*, 150901.

(6) Marx, D.; Hutter, J. *Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods*; Cambridge University Press: Cambridge, 2009.

(7) Iftimie, R.; Minary, P.; Tuckerman, M. E. Ab Initio Molecular Dynamics: Concepts, Recent Developments, and Future Trends. *Proc. Natl. Acad. Sci. U. S. A.* **2005**, *102*, 6654−6659.

(8) Perdew, J. P.; Schmidt, K. Jacob's Ladder of Density Functional Approximations for the Exchange-Correlation Energy. In *Density Functional Theory and Its Application to Materials: Antwerp, Belgium, Jun. 8−10, 2000*; Van Doren, V. E., Van Alsenoy, C., Geerlings, P., Eds.; AIP Publishing: Melville, NY, 2001; p 1.

(9) Ceperley, D. M.; Alder, B. J. Ground State of the Electron Gas by a Stochastic Method. *Phys. Rev. Lett.* **1980**, *45*, 566−569.

(10) Becke, A. D. Density-Functional Exchange-Energy Approximation with Correct Asymptotic Behavior. *Phys. Rev. A: At., Mol., Opt. Phys.* **1988**, *38*, 3098−3100.

(11) Lee, C.; Yang, W.; Parr, R. G. Development of the Colle-Salvetti Correlation-Energy Formula into a Functional of the Electron Density. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1988**, *37*, 785−789.

(12) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1996**, *77*, 3865−3868.

(13) Klimeš, J.; Michaelides, A. Perspective: Advances and Challenges in Treating van der Waals Dispersion Forces in Density Functional Theory. *J. Chem. Phys.* **2012**, *137*, 120901.

(14) Grimme, S.; Hansen, A.; Brandenburg, J. G.; Bannwarth, C. Dispersion-Corrected Mean-Field Electronic Structure Methods. *Chem. Rev.* **2016**, *116*, 5105−5154.
(15) Hermann, J.; DiStasio, R. A., Jr.; Tkatchenko, A. First-Principles Models for van der Waals Interactions in Molecules and Materials: Concepts, Theory, and Applications. *Chem. Rev.* **2017**, *117*, 4714−4758.
(16) Berland, K.; Cooper, V. R.; Lee, K.; Schröder, E.; Thonhauser, T.; Hyldgaard, P.; Lundqvist, B. I. Van der Waals Forces in Density Functional Theory: A Review of the vdW-DF Method. *Rep. Prog. Phys.* **2015**, *78*, 066501.
(17) Becke, A. D.; Johnson, E. R. Exchange-Hole Dipole Moment and the Dispersion Interaction Revisited. *J. Chem. Phys.* **2007**, *127*, 154108.
(18) Tkatchenko, A.; Scheffler, M. Accurate Molecular van der Waals Interactions from Ground-State Electron Density and Free-Atom Reference Data. *Phys. Rev. Lett.* **2009**, *102*, 073005.
(19) Grimme, S.; Antony, J.; Ehrlich, S.; Krieg, H. A Consistent and Accurate Ab Initio Parametrization of Density Functional Dispersion Correction (DFT-D) for the 94 Elements H-Pu. *J. Chem. Phys.* **2010**, *132*, 154104.
(20) Ferri, N.; DiStasio, R. A., Jr.; Ambrosetti, A.; Car, R.; Tkatchenko, A. Electronic Properties of Molecules and Surfaces with a Self-Consistent Interatomic van der Waals Density Functional. *Phys. Rev. Lett.* **2015**, *114*, 176802.
(21) Caldeweyher, E.; Bannwarth, C.; Grimme, S. Extension of the D3 Dispersion Coefficient Model. *J. Chem. Phys.* **2017**, *147*, 034112.
(22) Tkatchenko, A.; DiStasio, R. A., Jr.; Car, R.; Scheffler, M. Accurate and Efficient Method for Many-Body van der Waals Interactions. *Phys. Rev. Lett.* **2012**, *108*, 236402.
(23) DiStasio, R. A., Jr.; von Lilienfeld, O. A.; Tkatchenko, A. Collective Many-Body van der Waals Interactions in Molecular Systems. *Proc. Natl. Acad. Sci. U. S. A.* **2012**, *109*, 14791−14795.
(24) DiStasio, R. A., Jr.; Gobre, V. V.; Tkatchenko, A. Many-Body van der Waals Interactions in Molecules and Condensed Matter. *J. Phys.: Condens. Matter* **2014**, *26*, 213202.
(25) Ambrosetti, A.; Reilly, A. M.; DiStasio, R. A., Jr.; Tkatchenko, A. Long-Range Correlation Energy Calculated from Coupled Atomic Response Functions. *J. Chem. Phys.* **2014**, *140*, 18A508.
(26) Blood-Forsythe, M. A.; Markovich, T.; DiStasio, R. A., Jr.; Car, R.; Aspuru-Guzik, A. Analytical Nuclear Gradients for the Range-Separated Many-Body Dispersion Model of Noncovalent Interactions. *Chem. Sci.* **2016**, *7*, 1712−1728.
(27) Dion, M.; Rydberg, H.; Schröder, E.; Langreth, D. C.; Lundqvist, B. I. Van der Waals Density Functional for General Geometries. *Phys. Rev. Lett.* **2004**, *92*, 246401.
(28) Vydrov, O. A.; Van Voorhis, T. Nonlocal van der Waals Density Functional Made Simple. *Phys. Rev. Lett.* **2009**, *103*, 063004.
(29) Lee, K.; Murray, É. D.; Kong, L.; Lundqvist, B. I.; Langreth, D. C. Higher-Accuracy van der Waals Density Functional. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2010**, *82*, 081101.
(30) Ghosh, S. K.; Parr, R. G. Phase-Space Approach to the Exchange-Energy Functional of Density-Functional Theory. *Phys. Rev. A: At., Mol., Opt. Phys.* **1986**, *34*, 785−791.
(31) Becke, A. D.; Roussel, M. R. Exchange Holes in Inhomogeneous Systems: A Coordinate-Space Model. *Phys. Rev. A: At., Mol., Opt. Phys.* **1989**, *39*, 3761−3767.
(32) Tao, J.; Perdew, J. P.; Staroverov, V. N.; Scuseria, G. E. Climbing the Density Functional Ladder: Nonempirical Meta−Generalized Gradient Approximation Designed for Molecules and Solids. *Phys. Rev. Lett.* **2003**, *91*, 146401.
(33) Zhao, Y.; Truhlar, D. G. A New Local Density Functional for Main-Group Thermochemistry, Transition Metal Bonding, Thermochemical Kinetics, and Noncovalent Interactions. *J. Chem. Phys.* **2006**, *125*, 194101.
(34) Sun, J.; Haunschild, R.; Xiao, B.; Bulik, I. W.; Scuseria, G. E.; Perdew, J. P. Semilocal and Hybrid Meta-Generalized Gradient Approximations Based on the Understanding of the Kinetic-Energy-Density Dependence. *J. Chem. Phys.* **2013**, *138*, 044113.
(35) Sun, J.; Ruzsinszky, A.; Perdew, J. P. Strongly Constrained and Appropriately Normed Semilocal Density Functional. *Phys. Rev. Lett.* **2015**, *115*, 036402.
(36) Yu, H. S.; Li, S. L.; Truhlar, D. G. Perspective: Kohn-Sham Density Functional Theory Descending a Staircase. *J. Chem. Phys.* **2016**, *145*, 130901.
(37) Sun, J.; Remsing, R. C.; Zhang, Y.; Sun, Z.; Ruzsinszky, A.; Peng, H.; Yang, Z.; Paul, A.; Waghmare, U.; Wu, X.; Klein, M. L.; Perdew, J. P. Accurate First-Principles Structures and Energies of Diversely Bonded Systems from an Efficient Density Functional. *Nat. Chem.* **2016**, *8*, 831−836.
(38) Chen, M.; Ko, H.-Y.; Remsing, R. C.; Andrade, M. F. C.; Santra, B.; Sun, Z.; Selloni, A.; Car, R.; Klein, M. L.; Perdew, J. P.; Wu, X. Ab Initio Theory and Modeling of Water. *Proc. Natl. Acad. Sci. U. S. A.* **2017**, *114*, 10846−10851.
(39) Zheng, L.; Chen, M.; Sun, Z.; Ko, H.-Y.; Santra, B.; Dhuvad, P.; Wu, X. Structural, Electronic, and Dynamical Properties of Liquid Water by Ab Initio Molecular Dynamics Based on SCAN Functional within the Canonical Ensemble. *J. Chem. Phys.* **2018**, *148*, 164505.
(40) Calegari Andrade, M. F.; Ko, H.-Y.; Car, R.; Selloni, A. Structure, Polarization, and Sum Frequency Generation Spectrum of Interfacial Water on Anatase TiO_2. *J. Phys. Chem. Lett.* **2018**, *9*, 6716−6721.
(41) LaCount, M.; Gygi, F. Ensemble First-Principles Molecular Dynamics Simulations of Water Using the SCAN Meta-GGA Density Functional. *J. Chem. Phys.* **2019**, *151*, 164101.
(42) Xu, J.; Chen, M.; Zhang, C.; Wu, X. First-Principles Study of the Infrared Spectrum in Liquid Water from a Systematically Improved Description of H-Bond Network. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2019**, *99*, 205123.
(43) Calegari Andrade, M. F.; Ko, H.-Y.; Zhang, L.; Car, R.; Selloni, A. Free Energy of Proton Transfer at the Water−TiO_2 Interface from Ab Initio Deep Potential Molecular Dynamics. *Chem. Sci.* **2020**, *11*, 2335−2341.
(44) Perdew, J. P.; Zunger, A. Self-Interaction Correction to Density-Functional Approximations for Many-Electron Systems. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1981**, *23*, 5048−5079.
(45) Cohen, A. J.; Mori-Sánchez, P.; Yang, W. Insights into Current Limitations of Density Functional Theory. *Science* **2008**, *321*, 792−794.
(46) Gräfenstein, J.; Kraka, E.; Cremer, D. The Impact of the Self-Interaction Error on the Density Functional Theory Description of Dissociating Radical Cations: Ionic and Covalent Dissociation Limits. *J. Chem. Phys.* **2004**, *120*, 524−539.
(47) Lundberg, M.; Siegbahn, P. E. M. Quantifying the Effects of the Self-Interaction Error in DFT: When Do the Delocalized States Appear? *J. Chem. Phys.* **2005**, *122*, 224103.
(48) LeBlanc, L. M.; Dale, S. G.; Taylor, C. R.; Becke, A. .D.; Day, G. M.; Johnson, E. R. Pervasive Delocalisation Error Causes Spurious Proton Transfer in Organic Acid-Base Co-Crystals. *Angew. Chem.* **2018**, *130*, 15122−15126.
(49) Janesko, B. G.; Henderson, T. M.; Scuseria, G. E. Screened Hybrid Density Functionals for Solid-State Chemistry and Physics. *Phys. Chem. Chem. Phys.* **2009**, *11*, 443−454.
(50) Marsman, M.; Paier, J.; Stroppa, A.; Kresse, G. Hybrid Functionals Applied to Extended Systems. *J. Phys.: Condens. Matter* **2008**, *20*, 064201.
(51) Zhang, C.; Donadio, D.; Gygi, F.; Galli, G. First Principles Simulations of the Infrared Spectrum of Liquid Water Using Hybrid Density Functionals. *J. Chem. Theory Comput.* **2011**, *7*, 1443−1449.
(52) Zhang, C.; Wu, J.; Galli, G.; Gygi, F. Structural and Vibrational Properties of Liquid Water from van der Waals Density Functionals. *J. Chem. Theory Comput.* **2011**, *7*, 3054−3061.
(53) Gaiduk, A. P.; Gustafson, J.; Gygi, F.; Galli, G. First-Principles Simulations of Liquid Water Using a Dielectric-Dependent Hybrid Functional. *J. Phys. Chem. Lett.* **2018**, *9*, 3068−3073.
(54) Perdew, J. P. Size-Consistency, Self-Interaction Correction, and Derivative Discontinuity. In *Density Functional Theory of Many-*

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

Electron Systems, Advances in Quantum Chemistry Vol. *21*; Trickey, S. B., Ed.; Academic Press, 1990; p 113.
(55) Pederson, M. R.; Perdew, J. P. Self-Interaction Correction in Density Functional Theory: The Road Less Traveled. Ψ_k *Newsletter Scientific Highlight of the Month*, February (2012), see http://www.psi-k.org/newsletters/News_109/Highlight_109.pdf.
(56) Pederson, M. R.; Ruzsinszky, A.; Perdew, J. P. Communication: Self-Interaction Correction with Unitary Invariance in Density Functional Theory. *J. Chem. Phys.* **2014**, *140*, 121103.
(57) Becke, A. D. Density-Functional Thermochemistry. III. The Role of Exact Exchange. *J. Chem. Phys.* **1993**, *98*, 5648−5652.
(58) Perdew, J. P.; Ernzerhof, M.; Burke, K. Rationale for Mixing Exact Exchange with Density Functional Approximations. *J. Chem. Phys.* **1996**, *105*, 9982−9985.
(59) Becke, A. D. A New Mixing of Hartree-Fock and Local Density-Functional Theories. *J. Chem. Phys.* **1993**, *98*, 1372−1377.
(60) Skone, J. H.; Govoni, M.; Galli, G. Self-Consistent Hybrid Functional for Condensed Systems. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2014**, *89*, 195112.
(61) Garza, A. J.; Scuseria, G. E. Predicting Band Gaps with Hybrid Density Functionals. *J. Phys. Chem. Lett.* **2016**, *7*, 4165−4170.
(62) Heyd, J.; Scuseria, G. E.; Ernzerhof, M. Hybrid Functionals Based on a Screened Coulomb Potential. *J. Chem. Phys.* **2003**, *118*, 8207−8215.
(63) Guidon, M.; Hutter, J.; VandeVondele, J. Robust Periodic Hartree-Fock Exchange for Large-Scale Simulations Using Gaussian Basis Sets. *J. Chem. Theory Comput.* **2009**, *5*, 3010−3021.
(64) Duchemin, I.; Gygi, F. A Scalable and Accurate Algorithm for the Computation of Hartree-Fock Exchange. *Comput. Phys. Commun.* **2010**, *181*, 855−860.
(65) Bylaska, E. J.; Tsemekhman, K.; Baden, S. B.; Weare, J. H.; Jonsson, H. Parallel Implementation of Γ-Point Pseudopotential Plane-Wave DFT with Exact Exchange. *J. Comput. Chem.* **2011**, *32*, 54−69.
(66) Barnes, T. A.; Kurth, T.; Carrier, P.; Wichmann, N.; Prendergast, D.; Kent, P. R. C.; Deslippe, J. Improved Treatment of Exact Exchange in Quantum ESPRESSO. *Comput. Phys. Commun.* **2017**, *214*, 52−58.
(67) Varini, N.; Ceresoli, D.; Martin-Samos, L.; Girotto, I.; Cavazzoni, C. Enhancement of DFT-Calculations at Petascale: Nuclear Magnetic Resonance, Hybrid Density Functional Theory and Car-Parrinello Calculations. *Comput. Phys. Commun.* **2013**, *184*, 1827−1833.
(68) Guidon, M.; Hutter, J.; VandeVondele, J. Auxiliary Density Matrix Methods for Hartree-Fock Exchange Calculations. *J. Chem. Theory Comput.* **2010**, *6*, 2348−2364.
(69) Hu, W.; Lin, L.; Yang, C. Interpolative Separable Density Fitting Decomposition for Accelerating Hybrid Density Functional Calculations with Applications to Defects in Silicon. *J. Chem. Theory Comput.* **2017**, *13*, 5420−5431.
(70) Dong, K.; Hu, W.; Lin, L. Interpolative Separable Density Fitting through Centroidal Voronoi Tessellation with Applications to Hybrid Functional Electronic Structure Calculations. *J. Chem. Theory Comput.* **2018**, *14*, 1311−1320.
(71) Lin, L. Adaptively Compressed Exchange Operator. *J. Chem. Theory Comput.* **2016**, *12*, 2242−2249.
(72) Lin, L.; Lindsey, M. Convergence of Adaptive Compression Methods for Hartree-Fock-Like Equations. *Commun. Pure Appl. Math* **2019**, *72*, 451−499.
(73) Jia, W.; Lin, L. Fast Real-Time Time-Dependent Hybrid Functional Calculations with the Parallel Transport Gauge and the Adaptively Compressed Exchange Formulation. *Comput. Phys. Commun.* **2019**, *240*, 21−29.
(74) Hu, W.; Lin, L.; Yang, C. Projected Commutator DIIS Method for Accelerating Hybrid Functional Electronic Structure Calculations. *J. Chem. Theory Comput.* **2017**, *13*, 5458−5467.
(75) Marzari, N.; Vanderbilt, D. Maximally Localized Generalized Wannier Functions for Composite Energy Bands. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1997**, *56*, 12847−12865.
(76) Wu, X.; Selloni, A.; Car, R. Order-N Implementation of Exact Exchange in Extended Insulating Systems. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2009**, *79*, 085102.
(77) Marzari, N.; Mostofi, A. A.; Yates, J. R.; Souza, I.; Vanderbilt, D. Maximally Localized Wannier Functions: Theory and Applications. *Rev. Mod. Phys.* **2012**, *84*, 1419−1475.
(78) DiStasio, R. A., Jr.; Santra, B.; Li, Z.; Wu, X.; Car, R. The Individual and Collective Effects of Exact Exchange and Dispersion Interactions on the Ab Initio Structure of Liquid Water. *J. Chem. Phys.* **2014**, *141*, 084502.
(79) Gygi, F. Compact Representations of Kohn-Sham Invariant Subspaces. *Phys. Rev. Lett.* **2009**, *102*, 166406.
(80) Gygi, F.; Duchemin, I. Efficient Computation of Hartree−Fock Exchange Using Recursive Subspace Bisection. *J. Chem. Theory Comput.* **2013**, *9*, 582−587.
(81) Damle, A.; Lin, L.; Ying, L. Compressed Representation of Kohn−Sham Orbitals via Selected Columns of the Density Matrix. *J. Chem. Theory Comput.* **2015**, *11*, 1463−1469.
(82) Damle, A.; Lin, L.; Ying, L. Computing Localized Representations of the Kohn−Sham Subspace via Randomization and Refinement. *SIAM J. Sci. Comput.* **2017**, *39*, B1178−B1198.
(83) Damle, A.; Lin, L.; Ying, L. SCDM-k: Localized Orbitals for Solids via Selected Columns of the Density Matrix. *J. Comput. Phys.* **2017**, *334*, 1−15.
(84) Mountjoy, J.; Todd, M.; Mosey, N. J. Exact Exchange with Non-Orthogonal Generalized Wannier Functions. *J. Chem. Phys.* **2017**, *146*, 104108.
(85) Izmaylov, A. F.; Scuseria, G. E.; Frisch, M. J. Efficient Evaluation of Short-Range Hartree-Fock Exchange in Large Molecules and Periodic Systems. *J. Chem. Phys.* **2006**, *125*, 104103.
(86) Guidon, M.; Schiffmann, F.; Hutter, J.; VandeVondele, J. Ab Initio Molecular Dynamics Using Hybrid Density Functionals. *J. Chem. Phys.* **2008**, *128*, 214104.
(87) Carnimeo, I.; Baroni, S.; Giannozzi, P. Fast Hybrid Density-Functional Computations Using Plane-Wave Basis Sets. *Electronic Structure* **2019**, *1*, 015009.
(88) Gaiduk, A. P.; Gygi, F.; Galli, G. Density and Compressibility of Liquid Water and Ice from First-Principles Simulations with Hybrid Functionals. *J. Phys. Chem. Lett.* **2015**, *6*, 2902−2908.
(89) Zhang, C.; Pham, T. A.; Gygi, F.; Galli, G. Communication: Electronic Structure of the Solvated Chloride Anion from First Principles Molecular Dynamics. *J. Chem. Phys.* **2013**, *138*, 181102.
(90) Gaiduk, A. P.; Zhang, C.; Gygi, F.; Galli, G. Structural and Electronic Properties of Aqueous NaCl Solutions from Ab Initio Molecular Dynamics Simulations with Hybrid Density Functionals. *Chem. Phys. Lett.* **2014**, *604*, 89−96.
(91) Dawson, W.; Gygi, F. Performance and Accuracy of Recursive Subspace Bisection for Hybrid DFT Calculations in Inhomogeneous Systems. *J. Chem. Theory Comput.* **2015**, *11*, 4655−4663.
(92) Damle, A.; Lin, L. Disentanglement via Entanglement: A Unified Method for Wannier Localization. *Multiscale Model. Simul.* **2018**, *16*, 1392−1410.
(93) Boys, S. F. Construction of Some Molecular Orbitals to Be Approximately Invariant for Changes from One Molecule to Another. *Rev. Mod. Phys.* **1960**, *32*, 296−299.
(94) Resta, R. Quantum-Mechanical Position Operator in Extended Systems. *Phys. Rev. Lett.* **1998**, *80*, 1800−1803.
(95) Silvestrelli, P. L.; Parrinello, M. Water Molecule Dipole in the Gas and in the Liquid Phase. *Phys. Rev. Lett.* **1999**, *82*, 3308−3311.
(96) Souza, I.; Wilkens, T.; Martin, R. M. Polarization and Localization in Insulators: Generating Function Approach. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2000**, *62*, 1666−1683.
(97) Kirchner, B.; Hutter, J. Solvent Effects on Electronic Properties from Wannier Functions in a Dimethyl Sulfoxide/Water Mixture. *J. Chem. Phys.* **2004**, *121*, 5133−5142.
(98) Sagui, C.; Pomorski, P.; Darden, T. A.; Roland, C. Ab Initio Calculation of Electrostatic Multipoles with Wannier Functions for Large-Scale Biomolecular Simulations. *J. Chem. Phys.* **2004**, *120*, 4530−4544.

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

(99) Silvestrelli, P. L. Van der Waals Interactions in DFT Made Easy by Wannier Functions. *Phys. Rev. Lett.* **2008**, *100*, 053002.
(100) Mostofi, A. A.; Yates, J. R.; Lee, Y.-S.; Souza, I.; Vanderbilt, D.; Marzari, N. wannier90: A Tool for Obtaining Maximally-Localised Wannier Functions. *Comput. Phys. Commun.* **2008**, *178*, 685−699.
(101) Giannozzi, P.; Baroni, S.; Bonini, N.; Calandra, M.; Car, R.; Cavazzoni, C.; Ceresoli, D.; Chiarotti, G. L; Cococcioni, M.; Dabo, I.; Dal Corso, A.; de Gironcoli, S.; Fabris, S.; Fratesi, G.; Gebauer, R.; Gerstmann, U.; Gougoussis, C.; Kokalj, A.; Lazzeri, M.; Martin-Samos, L.; Marzari, N.; Mauri, F.; Mazzarello, R.; Paolini, S.; Pasquarello, A.; Paulatto, L.; Sbraccia, C.; Scandolo, S.; Sclauzero, G.; Seitsonen, A. P; Smogunov, A.; Umari, P.; Wentzcovitch, R. M QUANTUM ESPRESSO: A Modular and Open-Source Software Project for Quantum Simulations of Materials. *J. Phys.: Condens. Matter* **2009**, *21*, 395502.
(102) Giannozzi, P.; Andreussi, O.; Brumme, T.; Bunau, O.; Nardelli, M. B.; Calandra, M.; Car, R.; Cavazzoni, C.; Ceresoli, D.; Cococcioni, M.; Colonna, N.; Carnimeo, I.; Corso, A. D.; de Gironcoli, S.; Delugas, P.; DiStasio, R. A., Jr.; Ferretti, A.; Floris, A.; Fratesi, G.; Fugallo, G.; Gebauer, R.; Gerstmann, U.; Giustino, F.; Gorni, T.; Jia, J.; Kawamura, M.; Ko, H.-Y.; Kokalj, A.; Küçükbenli, E.; Lazzeri, M.; Marsili, M.; Marzari, N.; Mauri, F.; Nguyen, N. L.; Nguyen, H.-V.; Otero-de-la Roza, A.; Paulatto, L.; Poncé, S.; Rocca, D.; Sabatini, R.; Santra, B.; Schlipf, M.; Seitsonen, A. P.; Smogunov, A.; Timrov, I.; Thonhauser, T.; Umari, P.; Vast, N.; Wu, X.; Baroni, S. Advanced Capabilities for Materials Modelling with Quantum ESPRESSO. *J. Phys.: Condens. Matter* **2017**, *29*, 465901.
(103) Soler, J. M.; Artacho, E.; Gale, J. D.; García, A.; Junquera, J.; Ordejón, P.; Sánchez-Portal, D. The SIESTA Method for Ab Initio Order-N Materials Simulation. *J. Phys.: Condens. Matter* **2002**, *14*, 2745−2779.
(104) Gonze, X.; Amadon, B.; Anglade, P. M.; Beuken, J. M.; Bottin, F.; Boulanger, P.; Bruneval, F.; Caliste, D.; Caracas, R.; Côté, M.; Deutsch, T.; Genovese, L.; Ghosez, P.; Giantomassi, M.; Goedecker, S.; Hamann, D. R.; Hermet, P.; Jollet, F.; Jomard, G.; Leroux, S.; Mancini, M.; Mazevet, S.; Oliveira, M. J. T.; Onida, G.; Pouillon, Y.; Rangel, T.; Rignanese, G. M.; Sangalli, D.; Shaltaf, R.; Torrent, M.; Verstraete, M. J.; Zerah, G.; Zwanziger, J. W. ABINIT: First-Principles Approach to Material and Nanosystem Properties. *Comput. Phys. Commun.* **2009**, *180*, 2582−2615.
(105) Valiev, M.; Bylaska, E. J.; Govind, N.; Kowalski, K.; Straatsma, T. P.; Van Dam, H. J. J.; Wang, D.; Nieplocha, J.; Apra, E.; Windus, T. L.; de Jong, W. A. NWChem: A Comprehensive and Scalable Open-Source Solution for Large Scale Molecular Simulations. *Comput. Phys. Commun.* **2010**, *181*, 1477−1489.
(106) Enkovaara, J.; Rostgaard, C.; Mortensen, J. J.; Chen, J.; Dułak, M.; Ferrighi, L.; Gavnholt, J.; Glinsvad, C.; Haikola, V.; Hansen, H. A.; Kristoffersen, H. H.; Kuisma, M.; Larsen, A. H.; Lehtovaara, L.; Ljungberg, M.; Lopez-Acevedo, O.; Moses, P. G.; Ojanen, J.; Olsen, T.; Petzold, V.; Romero, N. A.; Stausholm-Møller, J.; Strange, M.; Tritsaris, G. A.; Vanin, M.; Walter, M.; Hammer, B.; Häkkinen, H.; Madsen, G. K. H.; Nieminen, R. M.; Nørskov, J. K.; Puska, M.; Rantala, T. T.; Schiøtz, J.; Thygesen, K. S.; Jacobsen, K. W. Electronic Structure Calculations with GPAW: A Real-Space Implementation of the Projector Augmented-Wave Method. *J. Phys.: Condens. Matter* **2010**, *22*, 253202.
(107) Hutter, J.; Iannuzzi, M.; Schiffmann, F.; VandeVondele, J. cp2k: Atomistic Simulations of Condensed Matter Systems. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2014**, *4*, 15−25.
(108) Kresse, G.; Joubert, D. From Ultrasoft Pseudopotentials to the Projector Augmented-Wave Method. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1999**, *59*, 1758−1775.
(109) Sharma, M.; Wu, Y.; Car, R. Ab Initio Molecular Dynamics with Maximally Localized Wannier Functions. *Int. J. Quantum Chem.* **2003**, *95*, 821−829.
(110) Iftimie, R.; Thomas, J. W.; Tuckerman, M. E. On-the-Fly Localization of Electronic Orbitals in Car-Parrinello Molecular Dynamics. *J. Chem. Phys.* **2004**, *120*, 2169−2181.
(111) Thomas, J. W.; Iftimie, R.; Tuckerman, M. E. Field Theoretic Approach to Dynamical Orbital Localization in Ab Initio Molecular Dynamics. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2004**, *69*, 125105.
(112) Car, R.; Parrinello, M. Unified Approach for Molecular Dynamics and Density-Functional Theory. *Phys. Rev. Lett.* **1985**, *55*, 2471.
(113) Cornean, H. D.; Gontier, D.; Levitt, A.; Monaco, D. Localised Wannier Functions in Metallic Systems. *Ann. Henri Poincare* **2019**, *20*, 1367−1391.
(114) Damle, A.; Levitt, A.; Lin, L. Variational Formulation for Wannier Functions with Entangled Band Structure. *Multiscale Model. Simul.* **2019**, *17*, 167−191.
(115) Kohn, W. Analytic Properties of Bloch Waves and Wannier Functions. *Phys. Rev.* **1959**, *115*, 809−821.
(116) Des Cloizeaux, J. Analytical Properties of *n*-Dimensional Energy Bands and Wannier Functions. *Phys. Rev.* **1964**, *135*, A698−A707.
(117) Nenciu, G. Existence of the Exponentially Localised Wannier Functions. *Commun. Math. Phys.* **1983**, *91*, 81−85.
(118) Niu, Q. Theory of the Quantized Adiabatic Particle Transport. *Mod. Phys. Lett. B* **1991**, *05*, 923−931.
(119) Panati, G.; Pisante, A. Bloch Bundles, Marzari-Vanderbilt Functional and Maximally Localized Wannier Functions. *Commun. Math. Phys.* **2013**, *322*, 835−875.
(120) Wu, X.; Walter, E. J.; Rappe, A. M.; Car, R.; Selloni, A. Hybrid Density Functional Calculations of the Band Gap of $Ga_xIn_{1-x}N$. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2009**, *80*, 115201.
(121) Chen, J.; Wu, X.; Selloni, A. Electronic Structure and Bonding Properties of Cobalt Oxide in the Spinel Structure. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2011**, *83*, 245204.
(122) Santra, B.; DiStasio, R. A., Jr.; Martelli, F.; Car, R. Local Structure Analysis in Ab Initio Liquid Water. *Mol. Phys.* **2015**, *113*, 2829−2841.
(123) Ko, H.-Y.; Zhang, L.; Santra, B.; Wang, H.; E, W.; DiStasio, R. A., Jr.; Car, R. Isotope Effects in Liquid Water via Deep Potential Molecular Dynamics. *Mol. Phys.* **2019**, *117*, 3269−3281.
(124) Bankura, A.; Santra, B.; DiStasio, R. A., Jr.; Swartz, C. W.; Klein, M. L.; Wu, X. A Systematic Study of Chloride Ion Solvation in Water Using van der Waals Inclusive Hybrid Density Functional Theory. *Mol. Phys.* **2015**, *113*, 2842−2854.
(125) Chen, M.; Zheng, L.; Santra, B.; Ko, H.-Y.; DiStasio, R. A., Jr.; Klein, M. L.; Car, R.; Wu, X. Hydroxide Diffuses Slower than Hydronium in Water Because Its Solvated Structure Inhibits Correlated Proton Transfer. *Nat. Chem.* **2018**, *10*, 413−419.
(126) Ko, H.-Y.; DiStasio, R. A., Jr.; Santra, B.; Car, R. Thermal Expansion in Dispersion-Bound Molecular Crystals. *Phys. Rev. Materials* **2018**, *2*, 055603.
(127) Kronik, L.; Makmal, A.; Tiago, M. L.; Alemany, M. M. G.; Jain, M.; Huang, X.; Saad, Y.; Chelikowsky, J. R. PARSEC − the Pseudopotential Algorithm for Real-Space Electronic Structure Calculations: Recent Advances and Novel Applications to Nano-Structures. *Phys. Status Solidi B* **2006**, *243*, 1063−1079.
(128) Saad, Y.; Chelikowsky, J. R.; Shontz, S. M. Numerical Methods for Electronic Structure Calculations of Materials. *SIAM Rev.* **2010**, *52*, 3−54.
(129) Mohr, S.; Ratcliff, L. E.; Genovese, L.; Caliste, D.; Boulanger, P.; Goedecker, S.; Deutsch, T. Accurate and Efficient Linear Scaling DFT Calculations with Universal Applicability. *Phys. Chem. Chem. Phys.* **2015**, *17*, 31360−31370.
(130) Skylaris, C.-K.; Haynes, P. D.; Mostofi, A. A.; Payne, M. C. Introducing ONETEP: Linear-Scaling Density Functional Simulations on Parallel Computers. *J. Chem. Phys.* **2005**, *122*, 084119.
(131) Bowler, D. R.; Choudhury, R.; Gillan, M. J.; Miyazaki, T. Recent Progress with Large-Scale Ab Initio Calculations: The CONQUEST Code. *Phys. Status Solidi B* **2006**, *243*, 989−1000.
(132) Gerber, I. C.; Ángyán, J. G. Hybrid Functional with Separated Range. *Chem. Phys. Lett.* **2005**, *415*, 100−105.

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785

(133) Vydrov, O. A.; Scuseria, G. E. Assessment of a Long-Range Corrected Hybrid Functional. *J. Chem. Phys.* **2006**, *125*, 234109.
(134) Baer, R.; Livshits, E.; Salzner, U. Tuned Range-Separated Hybrids in Density Functional Theory. *Annu. Rev. Phys. Chem.* **2010**, *61*, 85−109.
(135) Kronik, L.; Stein, T.; Refaely-Abramson, S.; Baer, R. Excitation Gaps of Finite-Sized Systems from Optimally Tuned Range-Separated Hybrid Functionals. *J. Chem. Theory Comput.* **2012**, *8*, 1515−1531.
(136) Karolewski, A.; Kronik, L.; Kümmel, S. Using Optimally Tuned Range Separated Hybrid Functionals in Ground-State Calculations: Consequences and Caveats. *J. Chem. Phys.* **2013**, *138*, 204115.
(137) Chen, W.; Wu, X.; Car, R. X-Ray Absorption Signatures of the Molecular Environment in Water and Ice. *Phys. Rev. Lett.* **2010**, *105*, 017802.
(138) Swartz, C. W.; Wu, X. Ab Initio Studies of Ionization Potentials of Hydrated Hydroxide and Hydronium. *Phys. Rev. Lett.* **2013**, *111*, 087801.
(139) Kümmel, S.; Kronik, L. Orbital-Dependent Density Functionals: Theory and Applications. *Rev. Mod. Phys.* **2008**, *80*, 3−60.
(140) Tassone, F.; Mauri, F.; Car, R. Acceleration Schemes for Ab Initio Molecular-Dynamics Simulations and Electronic-Structure Calculations. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1994**, *50*, 10561−10573.
(141) The typical choice for Γ is ≈0.1; e.g., see the "`electron_damping`" parameter in http://www.quantum-espresso.org/Doc/INPUT_CP.html.
(142) Adamo, C.; Barone, V. Toward Reliable Density Functional Methods without Adjustable Parameters: The PBE0 Model. *J. Chem. Phys.* **1999**, *110*, 6158−6170.
(143) Fornberg, B. Generation of Finite Difference Formulas on Arbitrarily Spaced Grids. *Math. Comp.* **1988**, *51*, 699−706.
(144) Press, W. H.; Flannery, B. P.; Teukolsky, S. A.; Vetterling, W. T. *Numerical Recipes in Fortran 77*; Cambridge University Press: Cambridge, 1992; p 102.
(145) Despite the fact that the real-space MLWFs had already been generated (with no additional cost) during the formation of the real-space electron density (needed to evaluate the semilocal xc energy), we found an additional (and completely unnecessary) `invFFT` call that specifically regenerates the real-space MLWFs in the current version of `QE`. As such, we have not included this additional cost in $\langle t_{\text{total}} \rangle$ and plan to remove this unnecessary routine from future versions of `QE`.
(146) Abascal, J. L. F.; Vega, C. A General Purpose Model for the Condensed Phases of Water: TIP4P/2005. *J. Chem. Phys.* **2005**, *123*, 234505.
(147) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High Performance Molecular Simulations through Multi-Level Parallelism from Laptops to Supercomputers. *SoftwareX* **2015**, *1−2*, 19−25.
(148) Martyna, G. J.; Klein, M. L.; Tuckerman, M. Nosé-Hoover Chains: The Canonical Ensemble via Continuous Dynamics. *J. Chem. Phys.* **1992**, *97*, 2635−2643.
(149) Tobias, D. J.; Martyna, G. J.; Klein, M. L. Molecular Dynamics Simulations of a Protein in the Canonical Ensemble. *J. Phys. Chem.* **1993**, *97*, 12959−12966.
(150) Hamann, D. R.; Schlüter, M.; Chiang, C. Norm-Conserving Pseudopotentials. *Phys. Rev. Lett.* **1979**, *43*, 1494−1497.
(151) Vanderbilt, D. Optimally Smooth Norm-Conserving Pseudopotentials. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1985**, *32*, 8412−8415.
(152) Gygi, F. Architecture of Qbox: A Scalable First-principles Molecular Dynamics Code. *IBM J. Res. Dev.* **2008**, *52*, 137−144.
(153) This development version is accessible at https://gitlab.com/kosinyj/exx_module_version_one_demo.
(154) Han, J.; Zhang, L.; Car, R.; E, W. Deep Potential: A General Representation of a Many-Body Potential Energy Surface. *Commun. Comput. Phys.* **2018**, *23*, 629−639.
(155) Zhang, L.; Han, J.; Wang, H.; Car, R.; E, W. Deep Potential Molecular Dynamics: A Scalable Model with the Accuracy of Quantum Mechanics. *Phys. Rev. Lett.* **2018**, *120*, 143001.
(156) Zhang, L.; Han, J.; Wang, H.; Saidi, W.; Car, R.; E, W. In *Advances in Neural Information Processing Systems 31*; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates: Red Hook, 2018; pp 4436−4446.
(157) *Quantum ESPRESSO: cp.x input description*. http://www.quantum-espresso.org/Doc/INPUT_CP.html (accessed February 6, 2020).

https://dx.doi.org/10.1021/acs.jctc.9b01167
J. Chem. Theory Comput. 2020, 16, 3757−3785