

Enabling Large-Scale Condensed-Phase Hybrid Density Functional Theory-Based *Ab Initio* Molecular Dynamics II: Extensions to the Isobaric–Isoenthalpic and Isobaric–Isothermal Ensembles

Hsin-Yu Ko, Biswajit Santra, and Robert A. DiStasio, Jr.*

Cite This: *J. Chem. Theory Comput.* 2021, 17, 7789–7813

Read Online

ACCESS |

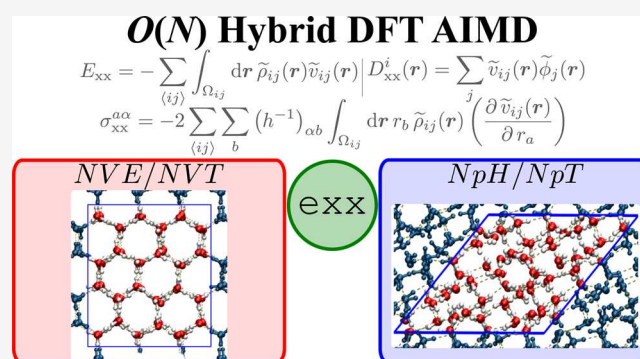


Metrics & More



Article Recommendations

ABSTRACT: In the previous paper of this series [Ko, H.-Y. *et al.* *J. Chem. Theory Comput.* 2020, 16, 3757–3785], we presented a theoretical and algorithmic framework based on a localized representation of the occupied space that exploits the inherent sparsity in the real-space evaluation of the exact exchange (EXX) interaction in finite-gap systems. This was accompanied by a detailed description of `exx`, a massively parallel hybrid message-passing interface MPI/OpenMP implementation of this approach in Quantum ESPRESSO (QE) that enables linear scaling hybrid density functional theory (DFT)-based *ab initio* molecular dynamics (AIMD) in the microcanonical/canonical (NVE/NVT) ensembles of condensed-phase systems containing 500–1000 atoms (in fixed orthorhombic cells) with a wall time cost comparable to semi-local DFT. In this work, we extend the current capabilities of `exx` to enable hybrid DFT-based AIMD simulations of large-scale condensed-phase systems with general and fluctuating cells in the isobaric–isoenthalpic/isobaric–isothermal (NpH/NpT) ensembles. The theoretical extensions to this approach include an analytical derivation of the EXX contribution to the stress tensor for systems in general simulation cells with a computational complexity that scales linearly with system size. The corresponding algorithmic extensions to `exx` include optimized routines that (i) handle both static and fluctuating simulation cells with non-orthogonal lattice symmetries, (ii) solve Poisson's equation in general/non-orthogonal cells *via* an automated selection of the auxiliary grid directions in the Natan–Kronik representation of the discrete Laplacian operator, and (iii) evaluate the EXX contribution to the stress tensor. Using this approach, we perform a case study on a variety of condensed-phase systems (including liquid water, a benzene molecular crystal polymorph, and semi-conducting crystalline silicon) and demonstrate that the EXX contributions to the energy and stress tensor simultaneously converge with an appropriate choice of `exx` parameters. This is followed by a critical assessment of the computational performance of the extended `exx` module across several different high-performance computing architectures *via* case studies on (i) the computational complexity due to lattice symmetry during NpT simulations of three different ice polymorphs (i.e., ice Ih, II, and III) and (ii) the strong/weak parallel scaling during large-scale NpT simulations of liquid water. We demonstrate that the robust and highly scalable implementation of this approach in the extended `exx` module is capable of evaluating the EXX contribution to the stress tensor with negligible cost (<1%) as well as all other EXX-related quantities needed during NpT simulations of liquid water (with a very tight 150 Ry planewave cutoff) in ≈ 5.2 s ($(\text{H}_2\text{O})_{128}$) and ≈ 6.8 s ($(\text{H}_2\text{O})_{256}$) per AIMD step. As such, the extended `exx` module presented in this work brings us another step closer to routinely performing hybrid DFT-based AIMD simulations of sufficient duration for large-scale condensed-phase systems across a wide range of thermodynamic conditions.



1. INTRODUCTION

Molecular dynamics (MD) is a deterministic numerical simulation method for efficiently sampling high-dimensional potential energy surfaces (PES) in systems of importance throughout biology, chemistry, physics, and materials science.^{1,2} Following the fundamental postulates of statistical mechanics, the trajectory of an MD simulation can be used to determine the thermodynamic properties of a system, as well as connect such macroscopic quantities to microscopic behavior.

As such, MD simulations are commonly used to furnish detailed microscopic-level insights into a wide range of

Received: November 13, 2020

Published: November 14, 2021



ACS Publications

© 2021 American Chemical Society

7789

<https://doi.org/10.1021/acs.jctc.0c01194>
J. Chem. Theory Comput. 2021, 17, 7789–7813

phenomena, including (but not limited to) the assembly and structure of large-scale nanostructures and materials,^{3–7} chemical reactions and kinetics,^{8–12} as well as complex biological processes.^{13–16} In practice, MD simulations of finite-sized systems are performed in the statistical mechanical ensemble corresponding to the thermodynamic conditions used to prepare and characterize the system of interest. In the microcanonical (*NVE*) ensemble, for example, the particle number (*N*), volume (*V*), and total internal energy (*E*) of the system are kept constant, which corresponds to an isolated system under adiabatic conditions. In canonical (*NVT*) MD simulations, the energy associated with endothermic and exothermic processes is exchanged with a thermostat at a fixed temperature (*T*), which allows one to account for thermal effects at constant *V* (or constant system density, *N/V*). To account for an externally applied pressure (*p*), a barostat can be introduced to facilitate sampling in the isobaric–isoenthalpic (*NpH*, when decoupled from a thermostat) and isobaric–isothermal (*NpT*, when coupled to a thermostat) ensembles, thereby enabling direct comparison to a larger swath of experiments (as most are performed at constant *p* instead of constant *V*). Another ensemble worth mentioning includes the grand canonical (*μVT*) ensemble, which fixes the chemical potential (*μ*) and enables MD simulations of open systems in contact with thermal and particle reservoirs. Since MD is an importance sampling technique, it can also be used to efficiently generate high-quality data (e.g., positions, ionic forces, cell forces) that can be used to learn complex high-dimensional PES *via* machine learning-based approaches.^{17–20}

Assuming that the system is ergodic, the accuracy of a given MD simulation in predicting equilibrium properties is primarily governed by the quality of the ionic forces and the stress tensor (or cell forces) used when propagating the corresponding equations of motion. As such, a physically sound approach for obtaining these forces is given by first-principles-based electronic structure theory, which is the foundation for *ab initio* MD (AIMD) simulations.^{21,22} With the AIMD technique, the nuclear PES is generated on-the-fly from the electronic ground state and does not require any empirical input, thereby allowing for a quantum mechanical treatment of structural, electronic/dielectric, and dynamical properties, as well as any potential chemical reactions that may occur.²¹ Due to its favorable balance between accuracy and computational cost, Kohn–Sham (KS) density functional theory (DFT)^{23,24} is the predominant electronic structure theory in AIMD, especially when performing large-scale simulations of complex condensed-phase materials. Within the KS-DFT framework, the total ground-state energy (*E*, which is not to be confused with the total internal energy of the system mentioned above) is comprised of the following terms: the KS (or mean-field) electronic kinetic energy (*E*_{kin}), the external potential energy (*E*_{ext}, which includes contributions from nucleus–electron and nucleus–nucleus interactions, as well as any other interactions with external fields), the Hartree potential energy (*E*_H, the classical description of the electron–electron interactions), and the so-called exchange–correlation (xc) energy (*E*_{xc}, which accounts for all remaining many-body electron exchange and correlation effects). While DFT provides an exact solution for the ground-state density (and properties) in principle, the exact functional form for *E*_{xc} remains unknown to date and must be approximated in practice.^{25–33}

When treating condensed-phase systems (such as solids and liquids), the most commonly used approaches for computing *E*_{xc} are the generalized gradient approximation (GGA) functionals such as those put forth by Perdew, Burke, and Ernzerhof (PBE)³⁴ as well as Becke, Lee, Yang, and Parr (BLYP),^{35,36} which express *E*_{xc} as a functional of the electron density, *ρ*(*r*), and its gradient, ∇*ρ*(*r*). Although such approaches are computationally efficient, the accuracy of a GGA functional is primarily limited by (i) its inability to fully describe non-local/long-range correlation effects such as dispersion (or van der Waals (vdW)) interactions^{37–40} and (ii) its propensity to suffer from self-interaction error (SIE), in which each electron spuriously interacts with itself.^{41,42} Without a complete and physically sound description of dispersion/vdW interactions, GGA–DFT faces difficulties when determining the structure of liquid water,⁴³ investigating drug–DNA binding,⁴⁴ predicting the structures and relative stabilities of molecular crystal polymorphs,⁴⁵ as well as quantifying the cohesion in asteroids.^{46,47} In addition, the presence of SIE at the GGA–DFT level leads to *ρ*(*r*) that are typically more delocalized, which results in a number of shortcomings including (but not limited to) excessive proton delocalization in liquid water,^{48–50} inadequate descriptions of transition states and charge-transfer complexes,^{51–53} as well as overestimation of lattice parameters.⁵⁴ To account for dispersion/vdW forces in GGA–DFT, a number of different approaches have been suggested in the literature,^{37–40} which range from effective pairwise models^{55–59} to more sophisticated many-body approaches^{44,60–63} and fully non-local xc functionals.^{64–66} To mitigate the SIE, hybrid-GGA functionals⁶⁷ admix a fraction of exact exchange (EXX) into *E*_{xc} as follows

$$E_{xc}^{\text{hybrid}} = a_x E_{xx} + (1 - a_x) E_x^{\text{GGA}} + E_c^{\text{GGA}} \quad (1)$$

where $0 < a_x < 1$ is a constant, *E*_{xx} is the EXX energy, and *E*_x^{GGA} and *E*_c^{GGA} are the GGA exchange and correlation contributions to *E*_{xc}, respectively. When compared to evaluating *E*_{xc} at the GGA level, the computational complexity introduced by the EXX contribution in eq 1 is significantly higher. As such, the efficient evaluation of *E*_{xx} is the key limitation to performing hybrid DFT-based AIMD simulations of large-scale condensed-phase systems and has triggered much attention in the community.^{43,68–97} For a more detailed summary of these approaches, we recommend the reader to the first paper in this series,⁹⁸ which will be referred to as PAPER-I throughout this work.

As discussed in PAPER-I,⁹⁸ a linear scaling yet numerically accurate evaluation of *E*_{xx} can be accomplished for large-scale finite-gap condensed-phase systems by employing a localized representation of the occupied orbitals^{43,80} (e.g., maximally localized Wannier functions (MLWFs)).^{79,81} In that work, we provided an in-depth discussion of the theoretical background, accuracy, and performance of a massively parallel implementation (the *exx* module) of this MLWF-based EXX approach in the pseudopotential- and planewave-based open-source Quantum ESPRESSO (QE) package,^{99,100} and again refer the reader back to this work for additional details. As briefly summarized below, this algorithm achieves *O*(*N*) scaling by using localized orbitals to exploit the natural sparsity in the EXX interaction in real space i.e., this quantum mechanical interaction is short-ranged and only occurs in regions of orbital overlap. Letting {*φ*_{*i*}(*r*)} be the set of MLWFs obtained *via* an

orthogonal (unitary) transformation of the occupied KS eigenstates, $\{\phi_i(\mathbf{r})\}$, i.e., $\tilde{\phi}_i(\mathbf{r}) = \sum_j U_{ij} \phi_j(\mathbf{r})$, we first note that E_{xx} is invariant to such transformations and can be written as follows in the MLWF representation

$$E_{xx} = - \sum_{ij}^{N_o} \int d\mathbf{r} \int d\mathbf{r}' \frac{\tilde{\rho}_{ij}(\mathbf{r}) \tilde{\rho}_{ij}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \quad (2)$$

or equivalently

$$E_{xx} = - \sum_{ij}^{N_o} \int d\mathbf{r} \tilde{\rho}_{ij}(\mathbf{r}) \tilde{v}_{ij}(\mathbf{r}) \quad (3)$$

In these expressions (shown here without loss of generality for a closed-shell system with N_o occupied orbitals), $\tilde{\rho}_{ij}(\mathbf{r})$ is the so-called MLWF-product density

$$\tilde{\rho}_{ij}(\mathbf{r}) \equiv \tilde{\phi}_i(\mathbf{r}) \tilde{\phi}_j(\mathbf{r}) \quad (4)$$

and $\tilde{v}_{ij}(\mathbf{r})$ is the corresponding MLWF-product potential

$$\tilde{v}_{ij}(\mathbf{r}) \equiv \int d\mathbf{r}' \frac{\tilde{\rho}_{ij}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \quad (5)$$

i.e., the electrostatic potential felt by a test charge at $\mathbf{r}$ originating from the charge distribution $\tilde{\rho}_{ij}(\mathbf{r}')$. Since the focus of this work is large-scale condensed-phase systems with finite gaps, the first Brillouin zone can be sampled at the Γ point only; as such, we have the flexibility to work with real-valued KS orbitals (and MLWFs), and so $\tilde{\rho}_{ij}(\mathbf{r}) = \tilde{\rho}_{ji}(\mathbf{r})$ and $\tilde{v}_{ij}(\mathbf{r}) = \tilde{v}_{ji}(\mathbf{r})$. In this work, we again follow PAPER-I⁹⁸ by dressing all MLWF-specific quantities with tildes and leaving quantities that are invariant to the MLWF representation unmodified (e.g., E_{xx} in eqs 2 and 3). Since the MLWFs in finite-gap systems are exponentially localized in real space^{79,101–105} and have a significantly smaller support than the entire simulation cell, Ω , the use of MLWFs (or any other localized representation which spans the occupied space) allows us to exploit two levels of sparsity when computing E_{xx} (as well as all other EXX-related quantities, *vide infra*). Considering the expression for E_{xx} in the MLWF representation, one can immediately see that a numerically accurate evaluation of eq 3 only requires contributions from overlapping pairs of MLWFs (i.e., when $\tilde{\rho}_{ij}(\mathbf{r}) \neq 0$). Hence, the first level of computational savings in our approach originates from the fact that a given MLWF is exponentially localized and will only appreciably overlap with a finite number of neighboring MLWFs. As such, the number of EXX pair interactions per orbital becomes independent of system size (assuming a fixed system density), and the quadratic sum over MLWFs in eq 3 can be replaced with a linear sum over overlapping pairs of MLWFs without loss of accuracy. To harness the second level of computational savings, we define the MLWF-orbital domain corresponding to $\tilde{\phi}_i(\mathbf{r})$ as $\Omega_i \equiv \{\mathbf{r} \in \Omega \mid |\tilde{\phi}_i(\mathbf{r})| > \varepsilon\}$. Hence, Ω_i encompasses the support of $\tilde{\phi}_i(\mathbf{r})$ and thereby delineates the region of space where this MLWF is non-negligible. In the above expression, we follow the approach outlined by Gygi and co-workers,^{82,83,106} and neglect the regions of space where $|\tilde{\phi}_i(\mathbf{r})|$ is less than a small threshold ε . In analogy, we also define the MLWF-product domain corresponding to a pair of overlapping MLWFs, $\tilde{\phi}_i(\mathbf{r})$ and $\tilde{\phi}_j(\mathbf{r})$, as $\Omega_{ij} \equiv \Omega_i \cap \Omega_j$, which encompasses the support of $\tilde{\rho}_{ij}(\mathbf{r})$ (see Figure 1 of PAPER-I⁹⁸ for a schematic illustration of these domains). Considering again the energy expression in eq 3, one can also see that a numerically

accurate evaluation of the contribution to E_{xx} from each overlapping MLWF pair only requires spatial integration over Ω_{ij} (given that ε is sufficiently small). As such, the costly integration over Ω (i.e., the entire simulation cell) in eq 3 can be replaced with spatial integrals over system-size-independent Ω_{ij} domains. By accounting for both of these sparsity levels, eq 3 for E_{xx} can now be rewritten as the following working expression

$$E_{xx} = - \sum_{\langle ij \rangle} \int_{\Omega_{ij}} d\mathbf{r} \tilde{\rho}_{ij}(\mathbf{r}) \tilde{v}_{ij}(\mathbf{r}) \quad (6)$$

in which $\langle ij \rangle$ indicates that the summation over i and j only includes overlapping MLWF pairs and each integral is performed on the corresponding Ω_{ij} domain. With a judicious choice of cutoff parameters (see PAPER-I⁹⁸), the `exx` module in QE is able to compute E_{xx} in a numerically accurate fashion at a computational cost that scales linearly with system size.

From eq 6, it is clear that an accurate and efficient real-space evaluation of $\tilde{v}_{ij}(\mathbf{r})$ is of central importance to developing a numerically accurate and linear scaling algorithm for computing E_{xx} (as well as all other EXX-related quantities, *vide infra*) in large-scale condensed-phase systems. In the `exx` algorithm,⁹⁸ this is accomplished by an efficient conjugate-gradient (CG) solution to Poisson's equation (PE) for $\tilde{v}_{ij}(\mathbf{r})$ in the near field

$$\nabla^2 \tilde{v}_{ij}(\mathbf{r}) = -4\pi \tilde{\rho}_{ij}(\mathbf{r}) \quad \mathbf{r} \in \Omega_{ij} \quad (7)$$

subject to boundary conditions provided by a sufficiently converged multipole expansion (ME) of $\tilde{v}_{ij}(\mathbf{r})$ in the far field

$$\tilde{v}_{ij}(\mathbf{r}) = 4\pi \sum_{lm} \frac{Q_{lm}}{(2l+1)} \frac{Y_{lm}(\theta, \varphi)}{r^{l+1}} \quad \mathbf{r} \notin \Omega_{ij} \quad (8)$$

In this expression, $\mathbf{r} = (r, \theta, \varphi)$ is given in spherical polar coordinates, $Y_{lm}(\theta, \varphi)$ are the spherical harmonics, and

$$Q_{lm} = \int_{\Omega_{ij}} d\mathbf{r} Y_{lm}^*(\theta, \varphi) r^l \tilde{\rho}_{ij}(\mathbf{r}) \quad (9)$$

are the multipole moments of $\tilde{\rho}_{ij}(\mathbf{r})$. In addition to providing the boundary conditions required during the CG solution of the PE, the ME in eq 8 is also used when computing the EXX contribution to the wavefunction forces, which formally requires $\tilde{v}_{ij}(\mathbf{r})$ on Ω_i and Ω_j (see Section 2.2.2).

In PAPER-I,⁹⁸ we presented a linear scaling and numerically accurate algorithm for computing the EXX contribution to the energies and wavefunction forces in fixed orthorhombic cells, thereby enabling large-scale hybrid DFT-based AIMD simulations in the NVE and NVT ensembles for a wide array of condensed-phase systems. With access to high-performance computing (HPC) resources, the hybrid message-passing interface (MPI) and open multi-processing (OpenMP)-based implementation of `exx` in QE enables us to compute the EXX contributions to the energy and wavefunction forces for (H₂O)₂₅₆, a condensed-phase system containing >750 atoms, in approximately 2.4 s on the IBM Blue Gene/Q architecture. As such, the current `exx` module (and earlier pilot versions) has already enabled computational investigations into a number of important condensed-phase systems, including the electronic structure of semi-conducting solids,^{107,108} the structure and local order of ambient liquid water,^{43,109} the structural and dynamical properties of aqueous ionic solutions,^{110,111} the thermal properties of the pyridine-I

molecular crystal,¹¹² as well as the subtle isotope effects on the structure of liquid water.²⁰

In this work, we extend the capabilities of the `exx` module by deriving and implementing (i) the EXX contribution to the stress tensor (or cell forces), which is required for performing constant-pressure simulations, and (ii) a stable and efficient representation for the Laplacian during the solution of the PE in general/non-orthogonal simulation cells. In doing so, we enable accurate and efficient hybrid DFT-based AIMD simulations of large-scale condensed-phase systems (with arbitrary symmetries) in the *NpH* and *NpT* (as well as *NVE* and *NVT*) ensembles using the `exx` module. Since most experiments are performed at constant p and T (instead of constant V and T), this development will enable more sophisticated computational investigations into large-scale condensed-phase systems across a wider range of thermodynamic conditions. The rest of this study is organized as follows. In Section 2, we derive the EXX contribution to the stress tensor within the framework of our MLWF-based EXX approach, which is required for propagating the Car–Parrinello (CP) equations of motion under constant-pressure conditions.^{21,22,113} In Sections 3 and 4, we provide a detailed discussion of the algorithmic extensions implemented in the `exx` module as well as its accuracy and performance when simulating condensed-phase systems in the *NpT* ensemble. The paper is then ended with some brief conclusions in Section 5.

2. THEORY

In this section, we expand the theoretical framework underlying our linear-scaling hybrid DFT approach⁹⁸ to enable constant-pressure simulations of condensed-phase systems with general/non-orthogonal cells. We focus the discussion around the CP equations of motion (in conjunction with the Parrinello–Rahman barostat¹¹³) used to propagate the electronic, ionic, and cell degrees of freedom during constant-pressure simulations in the *NpH* ensemble; with the introduction of an appropriately chosen thermostat (for the ionic degrees of freedom), this approach can easily be extended to sample the *NpT* ensemble. Although the scope of this discussion is limited to the CPMD variant of AIMD, which provides a computationally efficient scheme for propagating localized orbitals,^{114–116} a novel and cost-effective extension to enable Born–Oppenheimer MD using this approach will be addressed in a forthcoming paper. When used in conjunction with second-order damped dynamics¹¹⁷ (or other global optimization techniques such as CG) on the ionic and cell degrees of freedom, variable-cell optimizations (in the absence of thermal and nuclear quantum fluctuations) are also possible with the approach presented herein.

2.1. Index Conventions. Following PAPER-I, we will utilize the following conventions for the 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
- α, β, γ : indices corresponding to the cell (lattice) vectors L_1, L_2 , and L_3
- I, J, K : indices for the N_A ions
- p, q : indices for the points on the real-space grid (with p not to be confused with the pressure)
- l, m : indices for spherical harmonics

2.2. EXX-Based CPMD in the *NpH* Ensemble.

2.2.1. Equations of Motion. In constant-pressure CPMD simulations, fictitious dynamics are introduced on the N_o occupied KS orbitals $\{\phi_i(\mathbf{r})\}$ and the simulation cell tensor $\mathbf{h}$ via artificial/fictitious masses μ (not to be confused with the chemical potential) and W , respectively. In this work, $\mathbf{h}$ is a 3×3 matrix defined as $\mathbf{h} \equiv (L_1 L_2 L_3)$ or $h_{\alpha\alpha} \equiv (L_\alpha)_\alpha$, where L_1, L_2 , and L_3 are the corresponding cell (lattice) vectors. The simulation cell volume will be denoted by $V = \det(\mathbf{h})$. Constant-pressure (*NpH*) CPMD simulations with the Parrinello–Rahman barostat¹¹³ are governed by the following equations of motion for the electronic, ionic, and cell degrees of freedom²²

$$\mu \ddot{\phi}_i(\mathbf{r}) = -\left(\frac{\delta E}{\delta \phi_i^*(\mathbf{r})}\right) + \sum_j \Lambda_{ij} \phi_j(\mathbf{r}) \quad (10)$$

$$M_I \ddot{\mathbf{S}}_I = -\mathbf{h}^{-1}(\nabla_{\mathbf{R}_I} E) - M_I \mathcal{G}^{-1} \dot{\mathcal{G}} \dot{\mathbf{S}}_I \quad (11)$$

$$W \ddot{\mathbf{h}} = (\mathbf{\Pi} - p\mathbf{1})(\mathbf{h}^T)^{-1} V \quad (12)$$

in which Newton's dot notation was used to indicate time derivatives, E is the total ground-state DFT energy (including the nuclear–nuclear repulsion), $-(\delta E / \delta \phi_i^*(\mathbf{r}))$ is the force acting on the i -th occupied KS wavefunction, Λ_{ij} is a Lagrange multiplier enforcing orthonormality in $\{\phi_i(\mathbf{r})\}$, $-\nabla_{\mathbf{R}_I} E$ is the force acting on the I -th ion (located at $\mathbf{R}_I$ with mass M_I), $\mathcal{G} = \mathbf{h}^T \mathbf{h}$ is the so-called metric tensor, $\mathbf{\Pi}$ is the total internal stress tensor, p is the applied (external) pressure, and $\mathbf{1}$ is the identity matrix. For the fluctuating simulation cells encountered in constant-pressure CPMD, it is more convenient to work in crystal (fractional) coordinates $\mathbf{S}_I$ for the ions, which are independent of the dynamical variables associated with the cell degrees of freedom and are related to the Cartesian coordinates via $\mathbf{R}_I = \mathbf{h} \mathbf{S}_I$ or $\mathbf{S}_I = \mathbf{h}^{-1} \mathbf{R}_I$.

The components of the 3×3 total internal stress tensor, $\mathbf{\Pi}$, can be further decomposed into kinetic (kin) and potential (pot) contributions as follows²²

$$\Pi_{ab} = \Pi_{ab}^{\text{kin}} + \Pi_{ab}^{\text{pot}} \quad (13)$$

In this expression, the kinetic contribution (Π_{ab}^{kin}) originates from the ionic kinetic energy via

$$\Pi_{ab}^{\text{kin}} = \frac{1}{V} \sum_I M_I \sum_{\alpha\beta} h_{\alpha\alpha} \dot{\mathbf{S}}_{I\alpha} \dot{\mathbf{S}}_{I\beta} h_{\beta\beta} \quad (14)$$

while the potential contribution (Π_{ab}^{pot}) arises from the cell derivatives, $\sigma^{a\alpha}$, of the ionic potential energy (i.e., the DFT energy)

$$\Pi_{ab}^{\text{pot}} = -\frac{1}{V} \sum_{\alpha} \left(\frac{\partial E}{\partial h_{\alpha\alpha}} \right) h_{b\alpha} \equiv -\frac{1}{V} \sum_{\alpha} \sigma^{a\alpha} h_{b\alpha} \quad (15)$$

2.2.2. EXX Contribution to the Wavefunction Forces. Since the explicit functional dependence of E_{xx} on the total one-electron density, $\rho(\mathbf{r}) \equiv 2 \sum_i |\phi_i(\mathbf{r})|^2 = 2 \sum_i |\tilde{\phi}_i(\mathbf{r})|^2$, is unknown, one would need to use special methods such as the optimized effective potential (OEP) technique¹¹⁸ to derive the EXX contribution to the wavefunction forces within a strict KS-DFT scheme. In this work, we instead adopt a generalized KS-DFT scheme by allowing for an orbital-dependent xc potential, as this approach (which is currently standard practice in the field) yields the same ground-state energies as the OEP formalism at

a fraction of the computational cost. Given the working expression for E_{xx} in eq 6, the EXX contribution to the wavefunction forces (which is needed to propagate the electronic degrees of freedom in eq 10) can be derived in a straightforward manner (see PAPER-I⁹⁸ for more details). In the MLWF representation, the wavefunction force acting on the i -th MLWF, $\tilde{D}_{xx}^i(\mathbf{r}) = -(\delta E_{xx}/\delta \tilde{\phi}_i^*(\mathbf{r}))$, takes on the following form

$$\tilde{D}_{xx}^i(\mathbf{r}) = \sum_j \tilde{v}_{ij}(\mathbf{r}) \tilde{\phi}_j(\mathbf{r}) \equiv \sum_j \tilde{D}_{xx}^{ij}(\mathbf{r}) \quad (16)$$

in which the sum only includes the $\tilde{\phi}_j(\mathbf{r})$ that overlap with $\tilde{\phi}_i(\mathbf{r})$. Here, we again follow PAPER-I by dressing all MLWF-specific quantities with tildes and leaving quantities that are invariant to the MLWF representation unmodified. Since $\tilde{\phi}_j(\mathbf{r})$ is exponentially localized, a numerically accurate evaluation of $\tilde{D}_{xx}^{ij}(\mathbf{r})$ in eq 16 only requires the action of $\tilde{v}_{ij}(\mathbf{r})$ on $\tilde{\phi}_j(\mathbf{r})$ over the system-size-independent Ω_j domain, i.e., where $\tilde{\phi}_j(\mathbf{r})$ is non-negligible. Taken together with the fact that the number of overlapping MLWF pairs is also system-size-independent (for a given i), the entire set of $\{\tilde{D}_{xx}^i(\mathbf{r})\}$ can therefore be evaluated in linear time. From eq 16, it is again clear that an accurate and 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.

2.2.3. EXX Contribution to the Stress Tensor. The remaining quantity needed to propagate the equations of motion during constant-pressure CPMD simulations at the hybrid DFT level is the EXX contribution to the stress tensor in eq 12. As can be seen in eqs 13–15, the EXX contribution is only present in the potential part of Π and arises from $\sigma^{aa} = (\partial E/\partial h_{aa})$, the derivative of the DFT energy with respect to the cell tensor (i.e., the so-called cell derivatives). As such, the EXX contribution to Π^{pot} in eq 15 requires evaluation of $\sigma_{xx}^{aa} = (\partial E_{xx}/\partial h_{aa})$, which takes on the following form (cf. eq 2)

$$\sigma_{xx}^{aa} = - \sum_{ij} \frac{\partial}{\partial h_{aa}} \int d\mathbf{r} \int d\mathbf{r}' \frac{\tilde{\rho}_{ij}(\mathbf{r}) \tilde{\rho}_{ij}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \quad (17)$$

To compute these cell derivatives, it is again more convenient to work in crystal coordinates, as was done above in the NpH equations of motion for the ionic degrees of freedom (see eq 11). For the electrons, the transformation between crystal coordinates $\mathbf{s}$ and Cartesian coordinates $\mathbf{r}$ is completely analogous and is given by $\mathbf{r} = \mathbf{h}\mathbf{s}$ or $\mathbf{s} = \mathbf{h}^{-1}\mathbf{r}$. Since the Jacobian for this transformation is given by $\det(d\mathbf{r}/d\mathbf{s}) = \det(\mathbf{h}) = V$, the relationship between an MLWF in Cartesian and crystal coordinates is $\tilde{\phi}_i(\mathbf{r}) = \tilde{\phi}_i(\mathbf{s})/\sqrt{V}$, from which it follows that

$$\tilde{\rho}_{ij}(\mathbf{r}) = \frac{1}{V} \tilde{\rho}_{ij}(\mathbf{s}) \quad (18)$$

Using this expression and the fact that $d\mathbf{r} = V d\mathbf{s}$, we can transform eq 17 into crystal coordinates, namely,

$$\sigma_{xx}^{aa} = - \sum_{ij} \frac{\partial}{\partial h_{aa}} \int d\mathbf{s} \int d\mathbf{s}' \frac{\tilde{\rho}_{ij}(\mathbf{s}) \tilde{\rho}_{ij}(\mathbf{s}')}{|\mathbf{h}(\mathbf{s} - \mathbf{s}')|} \quad (19)$$

in which all factors of V (arising from the transformations of the MLWF-product densities and differentials) have canceled. Since crystal coordinates are independent of the dynamical variables associated with the cell, the only remaining

dependence on $\mathbf{h}$ is in the denominator of the integrand in eq 19. Letting $\Delta\mathbf{s} = \mathbf{s} - \mathbf{s}'$, we can now perform the relevant derivative as follows

$$\frac{\partial |\mathbf{h}\Delta\mathbf{s}|^{-1}}{\partial h_{aa}} = \frac{\partial (\Delta\mathbf{s}^T \mathbf{h}^T \mathbf{h} \Delta\mathbf{s})^{-1/2}}{\partial h_{aa}} = - \frac{\sum_{\beta} (\Delta s_{\alpha} h_{\alpha\beta} \Delta s_{\beta})}{|\mathbf{h}\Delta\mathbf{s}|^3} \quad (20)$$

Plugging this expression into eq 19 and transforming back to Cartesian coordinates yields

$$\sigma_{xx}^{aa} = \sum_{ij} \sum_b (h^{-1})_{ab} \int d\mathbf{r} \int d\mathbf{r}' \tilde{\rho}_{ij}(\mathbf{r}) \tilde{\rho}_{ij}(\mathbf{r}') \frac{\Delta r_b \Delta r_a}{|\Delta\mathbf{r}|^3} \quad (21)$$

in which $\Delta\mathbf{r} = \mathbf{r} - \mathbf{r}'$. Further reduction of this expression is possible by splitting the integrand into two terms via $\Delta r_b = r_b - r'_b$, using the fact that $\Delta\mathbf{r}' = \mathbf{r}' - \mathbf{r} = -\Delta\mathbf{r}$, and then noticing that these terms are equivalent after swapping the $\mathbf{r}$ and $\mathbf{r}'$ dummy variables. After doing so, we can now write eq 21 in the following intermediate form

$$\sigma_{xx}^{aa} = 2 \sum_{ij} \sum_b (h^{-1})_{ab} \int d\mathbf{r} \int d\mathbf{r}' \tilde{\rho}_{ij}(\mathbf{r}) \tilde{\rho}_{ij}(\mathbf{r}') \frac{r_b \Delta r_a}{|\Delta\mathbf{r}|^3} \quad (22)$$

This expression can be further simplified by separating the integrand as follows

$$\sigma_{xx}^{aa} = 2 \sum_{ij} \sum_b (h^{-1})_{ab} \int d\mathbf{r} r_b \tilde{\rho}_{ij}(\mathbf{r}) \left[\int d\mathbf{r}' \tilde{\rho}_{ij}(\mathbf{r}') \frac{\Delta r_a}{|\Delta\mathbf{r}|^3} \right] \quad (23)$$

and then realizing that the term inside the square brackets (to within a sign) is the derivative of $\tilde{v}_{ij}(\mathbf{r})$ with respect to the a -th Cartesian component, that is,

$$\frac{\partial \tilde{v}_{ij}(\mathbf{r})}{\partial r_a} = \frac{\partial}{\partial r_a} \int d\mathbf{r}' \frac{\tilde{\rho}_{ij}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} = - \int d\mathbf{r}' \tilde{\rho}_{ij}(\mathbf{r}') \frac{\Delta r_a}{|\Delta\mathbf{r}|^3} \quad (24)$$

where we have used eq 5. As such, we now arrive at the final expression for the EXX cell derivatives needed during hybrid DFT-based CPMD simulations in the NpH (or NpT) ensemble

$$\sigma_{xx}^{aa} = -2 \sum_{ij} \sum_b (h^{-1})_{ab} \int d\mathbf{r} r_b \tilde{\rho}_{ij}(\mathbf{r}) \left(\frac{\partial \tilde{v}_{ij}(\mathbf{r})}{\partial r_a} \right) \quad (25)$$

In analogy to the working expression for E_{xx} in eq 6, a linear-scaling and numerically accurate evaluation of eq 25 is also possible by (i) replacing the quadratic sum over MLWFs with a linear sum over overlapping MLWF pairs ($\sum_{ij} \rightarrow \sum_{\langle ij \rangle}$) and (ii) performing the spatial integrals over system-size-independent Ω_{ij} domains instead of Ω (i.e., the entire simulation cell). Doing so leads us to the following working expression for σ_{xx}^{aa} in our MLWF-based EXX approach (cf. eq 6)

$$\sigma_{xx}^{aa} = -2 \sum_{\langle ij \rangle} \sum_b (h^{-1})_{ab} \int_{\Omega_{ij}} d\mathbf{r} r_b \tilde{\rho}_{ij}(\mathbf{r}) \left(\frac{\partial \tilde{v}_{ij}(\mathbf{r})}{\partial r_a} \right) \quad (26)$$

From eq 26, it is clear that once $\tilde{v}_{ij}(\mathbf{r})$ is evaluated (which is also required for computing E_{xx} and $\{\tilde{D}_{xx}^i(\mathbf{r})\}$), its gradient provides the remaining ingredients needed to compute the EXX contribution to the stress tensor via eq 15, that is,

$$(\Pi_{xx}^{\text{pot}})_{ab} = -\frac{1}{V} \sum_{\alpha} \sigma_{xx}^{\alpha\alpha} h_{b\alpha} \quad (27)$$

2.2.3.1. Isotropic Constraints on Π_{xx}^{pot} . In general, all components of Π (i.e., the full stress tensor) are utilized when propagating the equations of motion for the cell degrees of freedom during constant-pressure CPMD simulations (see eq 12). In certain cases, however, constraints can be applied to Π , which allow one to maintain desired lattice symmetries, avoid shear stress in fluids, and/or suppress phase transitions during constant-pressure CPMD simulations. For instance, it is common practice to enforce isotropic constraints on Π during NpH (or NpT) simulations of solids or liquids in simple cubic cells. In order to do so, the off-diagonal components of Π are set to zero and the diagonal components are replaced by the internal pressure of the system, i.e.,

$$\Pi_{ab} = p^{\text{int}} \delta_{ab} \text{ (isotropic)} \quad (28)$$

in which p^{int} is equivalent to the isotropic average of Π

$$p^{\text{int}} \equiv \frac{1}{3} \sum_a \Pi_{aa} = \frac{1}{3} \text{Tr } \Pi \quad (29)$$

From the EXX point of view, this is tantamount to replacing Π_{xx}^{pot} in eq 27 with

$$(\Pi_{xx}^{\text{pot}})_{ab} = p_{xx}^{\text{int}} \delta_{ab} \text{ (isotropic)} \quad (30)$$

where

$$p_{xx}^{\text{int}} = \frac{1}{3} \text{Tr } \Pi_{xx}^{\text{pot}} \quad (31)$$

is the EXX contribution to the internal pressure. By applying an equal (isotropic) cell force along each lattice vector, the simulation cell is not subjected to shear stress and maintains its shape throughout the MD trajectory.

When performing such isotropic NpH (or NpT) simulations of solids or liquids in simple cubic cells (with side length L and $h_{aa} = L\delta_{aa}$), p_{xx}^{int} does not even require evaluating all of the diagonal components of Π_{xx}^{pot} and can be simplified as follows (cf. eqs 27 and 31)

$$\begin{aligned} p_{xx}^{\text{int}} &= -\frac{1}{3V} \sum_a \sum_{\alpha} \sigma_{xx}^{\alpha\alpha} h_{a\alpha} = -\frac{1}{3V} \sum_a \sum_{\alpha} \sigma_{xx}^{\alpha\alpha} L\delta_{a\alpha} \\ &= -\frac{L}{3V} \sum_a \sigma_{xx}^{\alpha\alpha} = -\frac{L}{3V} \text{Tr } \sigma_{xx} \text{ (simple cubic)} \end{aligned} \quad (32)$$

Since $\partial h_{aa}/\partial L = \partial(L\delta_{aa})/\partial L = \delta_{aa}$ for a simple cubic cell, the trace over cell derivatives in eq 32 is equivalent to the derivative of E_{xx} with respect to L , that is,

$$\begin{aligned} \left(\frac{\partial E_{xx}}{\partial L} \right) &= \sum_a \sum_{\alpha} \left(\frac{\partial E_{xx}}{\partial h_{a\alpha}} \right) \left(\frac{\partial h_{a\alpha}}{\partial L} \right) = \sum_a \sum_{\alpha} \left(\frac{\partial E_{xx}}{\partial h_{a\alpha}} \right) \delta_{a\alpha} \\ &= \sum_a \left(\frac{\partial E_{xx}}{\partial h_{aa}} \right) = \text{Tr } \sigma_{xx} \text{ (simple cubic)} \end{aligned} \quad (33)$$

which allows us to write p_{xx}^{int} in the following alternative form

$$p_{xx}^{\text{int}} = -\frac{L}{3V} \left(\frac{\partial E_{xx}}{\partial L} \right) \text{ (simple cubic)} \quad (34)$$

Since $\mathbf{r} = L\mathbf{s}$ in a simple cubic cell, the evaluation of $(\partial E_{xx}/\partial L)$ can be further simplified as follows (cf. eqs 17–19)

$$\begin{aligned} \left(\frac{\partial E_{xx}}{\partial L} \right) &= -\sum_{ij} \frac{\partial}{\partial L} \int d\mathbf{r} \int d\mathbf{r}' \frac{\tilde{\rho}_{ij}(\mathbf{r}) \tilde{\rho}_{ij}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \\ &= -\sum_{ij} \frac{\partial}{\partial L} \int d\mathbf{s} \int d\mathbf{s}' \frac{\tilde{\rho}_{ij}(\mathbf{s}) \tilde{\rho}_{ij}(\mathbf{s}')}{L|\mathbf{s} - \mathbf{s}'|} \\ &= \frac{1}{L} \sum_{ij} \int d\mathbf{s} \int d\mathbf{s}' \frac{\tilde{\rho}_{ij}(\mathbf{s}) \tilde{\rho}_{ij}(\mathbf{s}')}{L|\mathbf{s} - \mathbf{s}'|} \\ &= \frac{1}{L} \sum_{ij} \int d\mathbf{r} \int d\mathbf{r}' \frac{\tilde{\rho}_{ij}(\mathbf{r}) \tilde{\rho}_{ij}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} = -\frac{E_{xx}}{L} \text{ (simple cubic)} \end{aligned} \quad (35)$$

By combining eqs 34 and 35, we arrive at the following expression for p_{xx}^{int} in a simple cubic cell

$$p_{xx}^{\text{int}} = -\frac{L}{3V} \left(\frac{\partial E_{xx}}{\partial L} \right) = \frac{E_{xx}}{3V} \text{ (simple cubic)} \quad (36)$$

As such, the EXX contribution to p^{int} (as well as Π) is trivial and only requires evaluation of E_{xx} when performing isotropic NpH (or NpT) simulations of solids or liquids in simple cubic cells at the hybrid DFT level of theory.

3. IMPLEMENTATION AND ALGORITHMIC DETAILS

In PAPER-I,⁹⁸ we presented a massively parallel implementation of our linear-scaling MLWF-based EXX algorithm (i.e., the `exx` module), which enabled hybrid DFT-based AIMD simulations of large-scale condensed-phase systems with fixed orthorhombic unit cells in the NVE and NVT ensembles. In this section, we describe an algorithmic extension to the `exx` module that enables such hybrid DFT simulations in the NpH and NpT (as well as the NVE and NVT) ensembles for systems with general/non-orthogonal cells. To do so, we first briefly review the `exx` module (Section 3.1) and then describe our extensions to `exx`, which includes algorithms that (i) handle fluctuating simulation cells with non-orthorhombic lattice symmetries (Sections 3.2 and 3.3) and (ii) compute the previously derived (Section 2.2.3) analytical evaluation of the EXX contribution to the stress tensor (Section 3.4).

3.1. Review of the `exx` Module. In this section, we briefly review the implementation of our linear-scaling MLWF-based EXX algorithm in `exx`, a standalone module which has been integrated (via a portable input/output interface) with the MLWF-enabled semi-local DFT routines in the `CP` module of QE.¹⁰⁰ To enable hybrid DFT simulations of large-scale condensed-phase systems using this approach, the `exx` module employs a dual-level MPI/OpenMP parallelization scheme, which is able to exploit both internode and intranode HPC resources. As depicted in the flowchart in Figure 1, the main input required for the `exx` module includes the current set of MLWFs at each CPMD step, $\{\tilde{\phi}_i(\mathbf{r})\}$, while the output produced by `exx` includes the corresponding EXX contributions to the energy (E_{xx}), wavefunction forces ($\{\tilde{D}_{xx}^i(\mathbf{r})\}$), and cell derivatives/stress tensor (σ_{xx} ; see Section 3.4). Given the capability to generate “on-the-fly” MLWFs during CPMD simulations, it should be reasonably straightforward to integrate the `exx` module into other (periodic) DFT codebases. Since the only input requirement of `exx` is an orthonormal set of sufficiently localized orbitals, the use of alternative localization schemes (e.g., recursive subspace bisection,^{82,83} selected columns of the density matrix,^{84–86}

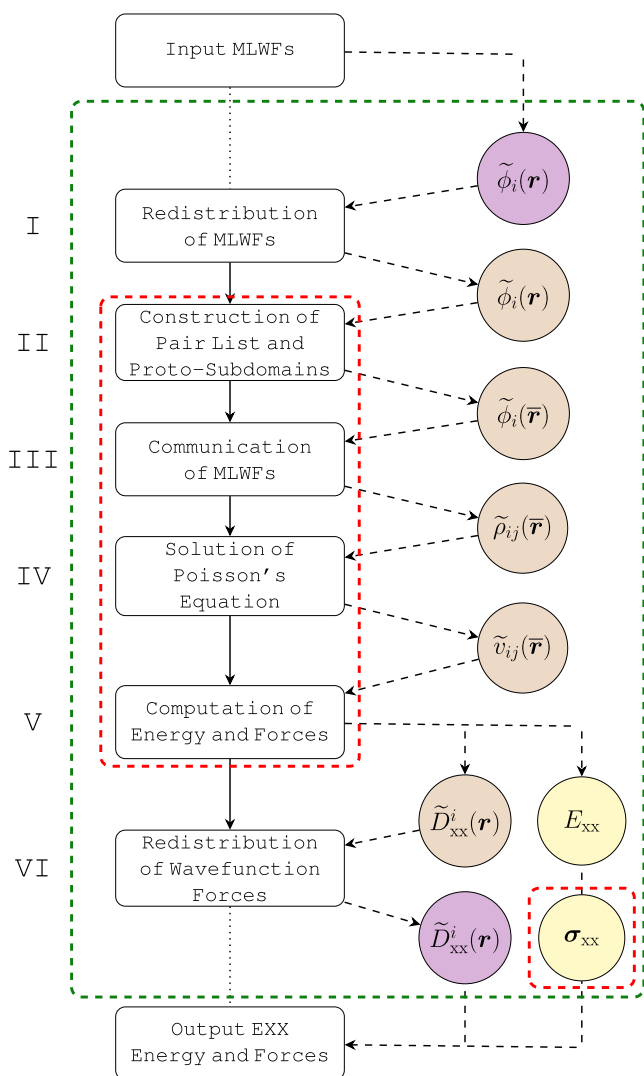


Figure 1. Flowchart of the `exx` module (dashed green box) in QE with extensions (dashed red boxes) for performing constant-volume (NVE/NVT) and constant-pressure (NpH/NpT) MLWF-based hybrid DFT simulations in general/non-orthogonal cells. As summarized in the main text, the input required by `exx` includes the set of MLWFs at the current CPMD step ($\{\tilde{\phi}_i(\mathbf{r})\}$), while the output produced by the extended `exx` module includes the EXX contributions to the energy (E_{xx}), wavefunction forces ($\{\tilde{D}_{xx}^i(\mathbf{r})\}$), and cell derivatives (σ_{xx}). Purple (brown) circles denote that a given quantity is represented according to the default GRID (customized⁹⁸ ORBITAL) data distribution scheme in QE. Pale yellow circles denote the data that are globally broadcast *via* MPI during the execution of the `exx` module. For a detailed description of each step, see Sections III.C.1–III.C.6 in PAPER-I⁹⁸ as well as Sections 3.1–3.4 in the current study.

and condensed-phase Pipek–Mezey¹¹⁹) are also possible with slight modifications to the code.

3.1.1. Step I (Redistribution of MLWFs). As mentioned above, the input to the `exx` module is $\{\tilde{\phi}_i(\mathbf{r})\}$, the current set of MLWFs at a given CPMD step. In QE, the so-called GRID scheme is employed when distributing the data corresponding to real-space quantities such as $\{\tilde{\phi}_i(\mathbf{r})\}$; in this scheme, each of the N_{proc} MPI processes holds the data corresponding to all N_o MLWFs on a subset of the real-space grid (see Sections III.A, III.B, and Figure 3 in PAPER-I⁹⁸). To efficiently utilize massively parallel HPC resources, Step I of the `exx` module

redistributes the $\{\tilde{\phi}_i(\mathbf{r})\}$ data from the GRID scheme to an alternative ORBITAL data distribution scheme, in which each MPI process now holds the data corresponding to a subset of MLWFs across the entire real-space grid (see Sections III.B, III.C.1, and Figure 3 in PAPER-I⁹⁸). In the `exx` module, the assignment of MLWFs across the pool of available MPI processes is governed by $\zeta \equiv N_{\text{proc}}/N_o$, that is, the ratio of MPI processes to MLWFs; when $\zeta = 1$ (which is a common mode for running `exx`), each MPI process P_i is assigned a single MLWF $\tilde{\phi}_i$. When $N_{\text{proc}} < N_o$ (i.e., $\zeta < 1$, less computational resources), multiple MLWFs are assigned to each P_i ; although the `exx` module allows for any (rational) value of $\zeta < 1$, a balanced distribution of MLWFs across MPI processes is only (currently) possible when N_{proc} is an exact divisor of N_o . When $N_{\text{proc}} > N_o$ (i.e., $\zeta > 1$, HPC resources), each MLWF is assigned to multiple MPI processes; in this case, the current `exx` module only allows for integer values for $\zeta > 1$. Unless otherwise specified, we will assume that $\zeta = 1$ throughout the rest of this work.

3.1.2. Step II (Construction of Pair List and Proto-Subdomains). With the $\{\tilde{\phi}_i(\mathbf{r})\}$ now distributed according to the ORBITAL scheme, the `exx` module enters Step II and generates a unique list of overlapping $\langle ij \rangle$ MLWF pairs to avoid redundant computation (see Section III.B.2, Algorithm 1, and Figures 4 and 5 in PAPER-I⁹⁸). Each overlapping $\langle ij \rangle$ pair is determined based on the criteria that $|\tilde{\mathbf{C}}_i - \tilde{\mathbf{C}}_j| \leq R_{\text{pair}}$; that is, the distance between two MLWF centers ($\tilde{\mathbf{C}}_i = \langle \tilde{\phi}_i | \mathbf{r} | \tilde{\phi}_i \rangle$ and $\tilde{\mathbf{C}}_j = \langle \tilde{\phi}_j | \mathbf{r} | \tilde{\phi}_j \rangle$) must be less than or equal to a user-defined radial distance cutoff (R_{pair}). The `exx` module then constructs the so-called unique MLWF pair list, $\mathcal{L}$, which determines how the computational workload will be distributed across the pool of available MPI processes, and therefore defines the computation and communication protocol in our algorithm. In constructing $\mathcal{L}$, the `exx` module removes $\langle ij \rangle$ and $\langle ji \rangle$ pair redundancy (which minimizes the overall computational workload) and then attempts to balance the workload among MPI processes while keeping the number of interprocess communication events minimal. During Step II, the `exx` module also generates two concentric spherical proto-subdomains, $\Theta(\mathbf{C}_0, R_{\text{PE}})$ and $\Theta(\mathbf{C}_0, R_{\text{ME}})$, which will be used later when computing each $\tilde{v}_{ij}(\mathbf{r})$ *via* the solution to Poisson's equation in the near field (PE) and a multipolar expansion in the far field (ME). Centered around $\mathbf{C}_0$ (the grid-resolved center of Ω), the sizes of these spherical proto-subdomains are determined by user-defined radii, i.e., $R_{\text{PE}} \in \{R_{\text{PE}}^s, R_{\text{PE}}^{\text{ns}}\}$ and $R_{\text{ME}} \in \{R_{\text{ME}}^s, R_{\text{ME}}^{\text{ns}}\}$ for $\langle ii \rangle$ (self-, s) and $\langle ij \rangle$ (non-self, ns) pairs; judicious choices for these parameters determine the accuracy and performance of the `exx` module (see Section III.C.2 and Figure 5, as well as Section IV and Figures 6 and 7 in PAPER-I⁹⁸). For each point in these proto-subdomains, we store the local (relative) Cartesian coordinates ($\tilde{\mathbf{r}} = \mathbf{r} - \mathbf{C}_0$) as well as the global grid point indices ($\mathbf{g}^0$) along the three lattice directions (L_1 , L_2 , and L_3). Based on these stored quantities, the $\Theta(\mathbf{C}_0, R_{\text{PE}})$ and $\Theta(\mathbf{C}_0, R_{\text{ME}})$ proto-subdomains will be used (during future steps) to generate the pair-specific $\Theta(\mathbf{C}_{ij}, R_{\text{PE}})$ and $\Theta(\mathbf{C}_{ij}, R_{\text{ME}})$ subdomains *via* a rigid translation from $\mathbf{C}_0$ to $\mathbf{C}_{ij}$, the grid-resolved midpoint of $\tilde{\mathbf{C}}_i$ and $\tilde{\mathbf{C}}_j$ (see Section III.C.2, Algorithm 2, and Figure 5 in PAPER-I⁹⁸).

3.1.3. Step III (Communication of MLWFs). For each overlapping $\langle ij \rangle$ pair in $\mathcal{L}$ (computed in Step II), the MPI process P_j (which holds $\tilde{\phi}_j(\mathbf{r})$ according to the ORBITAL scheme) first off-loads $\tilde{\phi}_j(\mathbf{r})$ onto the $\Theta(\mathbf{C}_{ij}, R_{\text{ME}})$ subdomain and then sends this orbital to P_i . With $\tilde{\phi}_j(\tilde{\mathbf{r}})$ on $\Theta(\mathbf{C}_{ij}, R_{\text{ME}})$ and

$\tilde{\phi}_i(\mathbf{r})$ stored locally according to the ORBITAL scheme, P_i now computes $\tilde{\rho}_{ij}(\mathbf{r})$ on the smaller $\Theta(C_{ij}R_{PE})$ subdomain (a formal subset of $\Theta(C_{ij}R_{ME})$) by multiplying these two MLWFs (see Section III.C.3 and Figures 4 and 5 in PAPER-I⁹⁸).

3.1.4. Step IV (Solution of PE). In Step IV, each MPI process P_i will first compute the far-field MLWF-product potential ($\tilde{v}_{ij}(\mathbf{r})$) on $\Theta(C_{ij}R_{ME}) \setminus \Theta(C_{ij}R_{PE})$ via a ME (see eqs 8 and 9). Each P_i then computes the near-field $\tilde{v}_{ij}(\mathbf{r})$ by solving the PE on $\Theta(C_{ij}R_{PE})$ (eq 7, with boundary conditions provided by the far-field $\tilde{v}_{ij}(\mathbf{r})$) using a finite-difference representation of the Laplacian operator¹²⁰ in conjunction with an iterative CG solver that has been efficiently parallelized over N_{thread} OpenMP threads (see Section III.C.4 and Figure 10 in PAPER-I⁹⁸).

3.1.5. Step V (Computation of Energy and Forces). With $\tilde{v}_{ij}(\mathbf{r})$ on $\Theta(C_{ij}R_{ME})$ for each $\langle ij \rangle$ pair (constructed using the combined near- and far-field solutions computed in Step IV), P_i then computes the $\langle ij \rangle$ contribution to the EXX energy (E_{xx}) and wavefunction forces ($\tilde{D}_{xx}^{ij}(\mathbf{r})$ and $\tilde{D}_{xx}^i(\mathbf{r})$). Following eq 6, E_{xx} is evaluated on $\Theta(C_{ij}R_{PE})$ (a fixed-size spherical representation for Ω_{ij}) and is accumulated via a straightforward MPI_SUM over the partial $\langle ij \rangle$ contributions computed on each MPI process. With $\tilde{v}_{ij}(\mathbf{r})$ in hand, P_i is also well-positioned to compute both $\tilde{D}_{xx}^{ij}(\mathbf{r}) = \tilde{v}_{ij}(\mathbf{r})\tilde{\phi}_i(\mathbf{r})$ and $\tilde{D}_{xx}^i(\mathbf{r}) = \tilde{v}_{ij}(\mathbf{r})\tilde{\phi}_j(\mathbf{r})$, which are required for $\tilde{D}_{xx}^i(\mathbf{r})$ and $\tilde{D}_{xx}^j(\mathbf{r})$, the total wavefunction forces acting on $\tilde{\phi}_i(\mathbf{r})$ and $\tilde{\phi}_j(\mathbf{r})$ (see eq 16). Both of these contributions are evaluated on $\Theta(C_{ij}R_{ME})$, a fixed-size spherical domain that should be chosen to be large enough (via the user-defined R_{ME} parameter) to cover the relevant sectors of both $\tilde{D}_{xx}^{ij}(\mathbf{r})$ and $\tilde{D}_{xx}^i(\mathbf{r})$. Since the far-field $\tilde{v}_{ij}(\mathbf{r})$ is dipolar at lowest order (due to the vanishing monopole associated with $\tilde{\rho}_{ij}(\mathbf{r})$), this quantity decays as $1/r^2$ for $i \neq j$; as such, a judicious choice for R_{ME} ensures rapid convergence in the $\langle ij \rangle$ (and $\langle ji \rangle$) contributions to the wavefunction forces (see Section II.C and Figure 7 in PAPER-I⁹⁸). After computing both $\tilde{D}_{xx}^{ij}(\mathbf{r})$ and $\tilde{D}_{xx}^i(\mathbf{r})$, $\tilde{D}_{xx}^{ij}(\mathbf{r})$ is locally accumulated on P_i to form $\tilde{D}_{xx}^i(\mathbf{r})$, while $\tilde{D}_{xx}^i(\mathbf{r})$ is shipped back (via MPI) to P_j where it is accumulated to form $\tilde{D}_{xx}^j(\mathbf{r})$ (see Section III.C.5 and Figure 4 in PAPER-I⁹⁸).

3.1.6. Step VI (Redistribution of Wavefunction Forces). At this stage, all EXX-related quantities have been evaluated; E_{xx} has been accumulated and broadcast to all MPI processes, while $\{\tilde{D}_{xx}^i(\mathbf{r})\}$ is now stored in the ORBITAL data distribution scheme. For compliance with the CP module in QE, $\{\tilde{D}_{xx}^i(\mathbf{r})\}$ is redistributed from the ORBITAL to the GRID scheme in this last step (see Sections III.A, III.B, and III.C.6 and Figure 3 in PAPER-I⁹⁸).

In order to extend our MLWF-based approach to enable constant-volume (NVE/NVT) and constant-pressure (NpH/NpT) hybrid DFT simulations of condensed-phase systems described by general/non-orthogonal cells, we have made a series of modifications to the `exx` module. Each of these modifications are described in detail below and are delineated by the red dashed boxes in the `exx` flowchart provided in Figure 1. In Section 3.2, we describe our modifications to Step II and Step III, which deal with proto-subdomain construction for (potentially fluctuating) simulation cells with general lattice symmetries. Our extensions to Step IV, which enable an efficient CG solution of the PE on non-orthogonal real-space domains, are detailed in Section 3.3. In Section 3.4, we present the needed extensions to Step V during constant-pressure CPMD simulations, i.e., analytical evaluation of the EXX

contribution to the stress tensor via the cell derivatives (σ_{xx}), as derived in Section 2.2.3.

3.2. Extension of the `exx` Module: Subdomains in Constant-Pressure CPMD. In this section, we describe our modifications to Step II and Step III of the `exx` module regarding the construction and selection of proto-subdomains during constant-volume and constant-pressure CPMD simulations of condensed-phases systems with general/non-orthogonal cells.

3.2.1. Proto-subdomain Construction for General/Non-orthogonal Simulation Cells. Treatment of general/non-orthogonal cells is a fairly straightforward extension to the orthorhombic case discussed previously (see Section III.C.2 and Algorithm 2 of PAPER-I⁹⁸) and requires the following two distinctions. For one, the lattice vectors (L_1, L_2, L_3) no longer coincide with the Cartesian directions (which are labeled using Roman indices a, b, c) and therefore require a distinct index convention (i.e., Greek indices α, β, γ) as defined in Section 2.1. In addition, the transformation between Cartesian and crystal coordinates requires the full cell tensor, i.e., $\mathbf{r} = \mathbf{h}\mathbf{s}$ (as opposed to the simpler $\mathbf{r}_a = |L_a|s_a$ in the orthorhombic case). For a general/non-orthogonal cell with $N_{\text{grid},\alpha}$ equispaced grid points along each of the L_α lattice vectors (with grid spacing $\delta s_\alpha = |L_\alpha|/N_{\text{grid},\alpha}$), the global grid index along L_α is given by $g_\alpha = N_{\text{grid},\alpha}s_\alpha$.

Given user-defined values for R_{PE} and R_{ME} (for both self (s) and non-self (ns) cases as discussed in Section IV of PAPER-I⁹⁸), the `exx` module now employs a general/non-orthogonal variant of Algorithm 2 in PAPER-I⁹⁸ during the construction of the $\Theta(C_0R_{PE})$ and $\Theta(C_0R_{ME})$ proto-subdomains, each of which contains $N_{PE} \in \{N_{PE}^s, N_{PE}^{ns}\}$ and $N_{ME} \in \{N_{ME}^s, N_{ME}^{ns}\}$ grid points, respectively. When compared to the original algorithm for orthorhombic simulation cells, the only difference lies in the use of the full cell tensor (instead of the lattice dimensions) during the computation of the $\mathbf{g}_{PE}^0$ and $\mathbf{g}_{ME}^0$ global grid indices. In practice, this leads to a revised assignment of $\mathbf{g}_{PE}^0[q']$ and $\mathbf{g}_{ME}^0[q'']$ with $\text{NINT}[N_{\text{grid},\alpha}(\mathbf{h}^{-1}\mathbf{r})_\alpha]$ (for $\alpha = 1, 2, 3$) instead of the original form, i.e., $\text{NINT}[N_{\text{grid},a}r_a/|L_a|]$ (for $a = 1, 2, 3$). As such, the resultant proto-subdomains reflect the symmetry of the underlying (general/non-orthogonal) simulation cell.

Following the same conventions defined in PAPER-I⁹⁸ the $\Theta(C_0R_{PE})$ and $\Theta(C_0R_{ME})$ proto-subdomains are again stored by the modified `exx` module as a set of local (relative) Cartesian coordinates in a $3 \times N_{ME}$ double-precision array

$$\bar{\mathbf{r}}[q] = \begin{cases} \bar{\mathbf{r}}_{PE}[q] & q = 1, \dots, N_{PE} \\ \bar{\mathbf{r}}_{ME}[q - N_{PE}] & q = N_{PE} + 1, \dots, N_{ME} \end{cases} \quad (37)$$

and a set of global grid indices in a $3 \times N_{ME}$ integer array

$$\mathbf{g}^0[q] = \begin{cases} \mathbf{g}_{PE}^0[q] & q = 1, \dots, N_{PE} \\ \mathbf{g}_{ME}^0[q - N_{PE}] & q = N_{PE} + 1, \dots, N_{ME} \end{cases} \quad (38)$$

With such compact representations of the $\Theta(C_0R_{PE})$ and $\Theta(C_0R_{ME})$ proto-subdomains, the `exx` module is now positioned to construct the $\Theta(C_{ij}R_{PE})$ subdomain (for computing the $\langle ij \rangle$ contributions to E_{xx} and σ_{xx}) as well as the $\Theta(C_{ij}R_{ME})$ subdomain (for computing $\tilde{D}_{xx}^{ij}(\mathbf{r})$ and $\tilde{D}_{xx}^i(\mathbf{r})$). As discussed in Section III.C.3 of PAPER-I⁹⁸ these subdomains can be conveniently obtained via a rigid translation of the $\Theta(C_0R_{PE})$ and $\Theta(C_0R_{ME})$ proto-subdomains from C_0 to C_{ij} (which is a crucial operation when

communicating the MLWFs among MPI processes in Step III of the `exx` module). For general/non-orthogonal cells, the component of the required grid translation vector (τ^j) along a lattice vector L_α is evaluated *via* an intermediate mapping to crystal coordinates ($s = h^{-1}(C_{ij} - C_0)$) and given by

$$\begin{aligned}\tau_\alpha^{ij} &= \text{NINT}[N_{\text{grid},\alpha} [h^{-1}(C_{ij} - C_0)]_\alpha] \\ &= \text{NINT}\left[\frac{|L_\alpha| [h^{-1}(C_{ij} - C_0)]_\alpha}{\delta\xi_\alpha}\right]\end{aligned}\quad (39)$$

where we have used the fact that $\delta\xi_\alpha = |L_\alpha|/N_{\text{grid},\alpha}$. Application of τ^j to a given proto-subdomain leaves the radius (R_{PE} or R_{ME}) and local Cartesian coordinates ($\bar{r}$) unchanged and simply offsets the global grid indices as follows

$$g_\alpha^{ij}[q] = \text{MOD}[g_\alpha^0[q] + \tau_\alpha^{ij}, N_{\text{grid},\alpha}] \quad (40)$$

thereby resulting in a subdomain ($\Theta(C_{ij}R_{\text{PE}})$ or $\Theta(C_{ij}R_{\text{ME}})$) that is centered at C_{ij} and has the symmetry of the underlying simulation cell.

3.2.2. Proto-subdomain Selection during Constant-Pressure CPMD Simulations. During an MLWF-based CPMD simulation of an insulating system, the band gap is not expected to have substantial variations; as such, individual MLWF spreads will fluctuate, but the size/extent of the support associated with these exponentially decaying functions will remain essentially constant throughout the trajectory. Here, we note that this assumption may break down (to varying extents) for small-gap and/or substantially inhomogeneous systems¹⁰⁶ (e.g., solvated semiconducting nanoparticles, water–semiconductor interfaces, surface adsorption of gas-phase molecules) as well as systems undergoing bond breaking and formation; in such cases, the use of MLWF-specific subdomains will be necessary to ensure a sufficiently converged evaluation of all EXX-related quantities and will therefore be addressed in future versions of `exx`. For fixed-cell simulations (e.g., `NVE/NVT`), the size of the $\Theta(C_{ij}R_{\text{PE}})$ and $\Theta(C_{ij}R_{\text{ME}})$ subdomains (i.e., the translated $\Theta(C_0R_{\text{PE}})$ and $\Theta(C_0R_{\text{ME}})$ proto-subdomains) is kept fixed throughout CPMD simulations by the `exx` module; for most systems (not including the pathological examples listed above), this choice results in a high-fidelity evaluation of E_{xx} and $\{\tilde{D}_{\text{xx}}^i(r)\}$ (as well as σ_{xx} *vide infra*). As such, all proto-subdomain-related quantities in `exx`, which include the radii ($R_{\text{PE}} \in \{R_{\text{PE}}^s, R_{\text{PE}}^{\text{ns}}\}$ and $R_{\text{ME}} \in \{R_{\text{ME}}^s, R_{\text{ME}}^{\text{ns}}\}$), the number of local grid points ($N_{\text{PE}} \in \{N_{\text{PE}}^s, N_{\text{PE}}^{\text{ns}}\}$ and $N_{\text{ME}} \in \{N_{\text{ME}}^s, N_{\text{ME}}^{\text{ns}}\}$), the local (relative) Cartesian coordinates ($\{\bar{r}\}$), and the global grid indices ($\{g^0\}$), are pre-computed prior to the first MD step and fixed throughout the simulation.

When a fluctuating cell is employed (e.g., during `NpH/NpT` simulations), the size and shape of Ω can vary significantly, while $\{\Omega_i\}$ (again for non-pathological systems) is expected to retain a similar size/extent (but potentially a different shape) throughout the MD trajectory. As such, we are now faced with the question of how to define the $\Theta(C_{ij}R_{\text{PE}})$ and $\Theta(C_{ij}R_{\text{ME}})$ subdomains during constant-pressure simulations with `exx`. In this work, we consider two common subdomain choices for CPMD simulations with fluctuating cells. As a first option, the subdomains could be chosen such that the radii (i.e., R_{PE} and R_{ME}) are fixed throughout the simulation; this leads to fixed quasi-spherical subdomain shapes with varying numbers of points (i.e., N_{PE} and N_{ME}) as the cell fluctuates. Algorithmically speaking, the use of fixed R_{PE} and R_{ME} has the disadvantages of (i) requiring the computation of $\{\bar{r}\}$ and $\{g^0\}$ during each

CPMD step, (ii) introducing an imbalance in the computational workload and associated memory requirements between CPMD steps, and (iii) complicating the extrapolation schemes used for the $\tilde{v}_{ij}(\bar{r})$ initial guess during the iterative solution of the PE.

To combat these algorithmic issues, we have opted to employ an alternative option in `exx`—choosing subdomains with a fixed number of grid points throughout constant-pressure CPMD simulations, with N_{PE} and N_{ME} values determined by the initially chosen proto-subdomains. More specifically, we retain the following (initial) proto-subdomain-related quantities throughout a given `NpH/NpT` simulation: the number of grid points (N_{PE} and N_{ME}), the global grid indices ($\{g^0\}$), and the relative scaled (not Cartesian) coordinates ($\{\bar{s}\} = \{h_0^{-1}\bar{r}\}$, where h_0 is the initial cell tensor). In other words, the subdomains employed in our approach do not have fixed radii and are therefore no longer (necessarily) quasi-spherical in shape; instead, these subdomains deform with the underlying fluctuating cell. In doing so, this scheme directly addresses all of the algorithmic disadvantages that accompany the use of subdomains with fixed radii. For one, there is no need for the additional computational overhead associated with computing $\{\bar{r}\}$ and $\{g^0\}$ by screening Ω at each CPMD step; in this case, $\{\bar{r}\}$ is straightforwardly obtained *via* $\{\bar{r}\} = \{h\bar{s}\}$ (where h is the current cell tensor) and $\{g^0\}$ is simply stored in memory. In addition, the complications associated with workload/memory imbalances as well as extrapolation schemes (for the PE guess) are largely eliminated with the use of a fixed number of grid points in each subdomain.

In the presence of severely anisotropic cell fluctuations (e.g., as one might encounter during a phase transition with large uniaxial strain), this approach should be further modified to ensure that the substantially deformed subdomains still provide adequate support for evaluating E_{xx} , $\{\tilde{D}_{\text{xx}}^i(r)\}$, and σ_{xx} . This can be accomplished with the re-assembly (from scratch) of appropriately sized quasi-spherical proto-subdomains based on a pre-defined strain criteria or a given stride (e.g., every 1000 CPMD steps) throughout the simulation. Doing so would ensure a sufficiently converged evaluation of all EXX-related quantities and still retain all of the algorithmic advantages mentioned above.

It is also worth noting that both of these subdomain choices (i.e., fixed radii or fixed number of points) are subject to Pulay-like errors²² during constant-pressure CPMD simulations. Such errors originate from the use of discrete Laplacian representations—whose accuracy is governed by the grid point spacing ($\{\delta\xi_\alpha\}$) in Ω —during the solution of the PE. In the `exx` module, the accumulation of such errors is largely mitigated by the default use of a sufficiently accurate finite-difference representation of the Laplacian operator (i.e., with an associated error of $O(\delta\xi_\alpha^6)$, *vide infra*), which can be reduced even further (at a linear computational cost) by simply employing a higher-order stencil (see Section 3.3).

3.3. Extension of the `exx` Module: Solving PE in an Arbitrary Simulation Cell. In this section, we describe the extensions introduced in `exx` to enable the solution of the PE for each overlapping $\langle ij \rangle$ MLWF pair (i.e., $\nabla^2 \tilde{v}_{ij}(\bar{r}) = -4\pi \tilde{\rho}_{ij}(\bar{r})$, see eq 7) in condensed-phase systems described by general/non-orthogonal simulation cells. Throughout this discussion, we will therefore consider the most general case in which the lattice vectors ($\{L_1, L_2, L_3\}$) are non-orthogonal and therefore

not necessarily aligned with the standard unit Cartesian directions ($\{\hat{\mathbf{e}}_x, \hat{\mathbf{e}}_y, \hat{\mathbf{e}}_z\}$), as one would encounter with orthorhombic (e.g., simple cubic) cells.

While a ME about C_{ij} (which is used to obtain the boundary conditions for the PE as well as the far-field solution for $\tilde{v}_{ij}(\mathbf{r})$) can be straightforwardly computed using eqs 8 and 9, the near-field solution for $\tilde{v}_{ij}(\mathbf{r})$ requires a discrete representation for the Laplacian operator when computing numerical second derivatives during the solution of the PE. Since the subdomains employed in the `exx` module are coincident with the underlying real-space grid, it is most computationally efficient to employ a discrete representation for the Laplacian that is aligned with L_1 , L_2 , and L_3 . To proceed, we employ the unit lattice vectors as the basis for this tilted (non-Cartesian) space, that is, $\hat{L}_\alpha \equiv L_\alpha/|L_\alpha|$ for $\alpha \in \{1, 2, 3\}$, such that a given position vector $\mathbf{v} \in \mathbb{R}^3$ can be written using either Cartesian ($\mathbf{r} = \{r_x, r_y, r_z\}$) or tilted/non-Cartesian ($\boldsymbol{\xi} = \{\xi_1, \xi_2, \xi_3\}$) coordinates. Direct solution of the PE on these subdomains (i.e., without the need for interpolation of $\tilde{\rho}_{ij}(\mathbf{r})$ and $\tilde{v}_{ij}(\mathbf{r})$ to and from an auxiliary Cartesian grid) will therefore require a coordinate transformation that connects the Laplacian operator in these two representations *via* the corresponding Jacobian matrix ($J = \partial \boldsymbol{\xi} / \partial \mathbf{r}$).

Since the tilted/non-Cartesian coordinates (which use the unit lattice vectors as a basis) are related to crystal coordinates (which use the lattice vectors as a basis) for any arbitrary position vector $\mathbf{v}$, namely

$$\mathbf{v} = \sum_\alpha s_\alpha \mathbf{L}_\alpha = \sum_\alpha s_\alpha |L_\alpha| \hat{L}_\alpha = \sum_\alpha \xi_\alpha \hat{L}_\alpha \quad (41)$$

one sees that $\xi_\alpha = |L_\alpha| s_\alpha = |L_\alpha| \sum_a (h^{-1})_{aa} r_a$. Using this relationship, one can derive an explicit expression for J as follows

$$\begin{aligned} J_{a\alpha} &= \left(\frac{\partial \xi_\alpha}{\partial r_a} \right) = \frac{\partial}{\partial r_a} \left[|L_\alpha| \sum_b (h^{-1})_{ab} r_b \right] = |L_\alpha| \sum_b (h^{-1})_{ab} \delta_{ba} \\ &= |L_\alpha| (h^{-1})_{a\alpha} \end{aligned} \quad (42)$$

With the Jacobian in eq 42, the Cartesian gradient operator, $\nabla_r \equiv (\partial/\partial r_x, \partial/\partial r_y, \partial/\partial r_z)$, can be written in terms of the directional derivatives along the (unit) lattice vectors, $\nabla_\xi \equiv (\partial/\partial \xi_1, \partial/\partial \xi_2, \partial/\partial \xi_3)$, *via* $\nabla_r = J \nabla_\xi$. These expressions can in turn be used to derive the desired form for the Laplacian operator, i.e.,

$$\begin{aligned} \nabla_r^2 &= \nabla_r \cdot \nabla_r = \sum_a (\nabla_r)_a (\nabla_r)_a = \sum_a \left[\sum_\alpha J_{a\alpha} (\nabla_\xi)_\alpha \right] \\ &\times \left[\sum_\beta J_{a\beta} (\nabla_\xi)_\beta \right] = \sum_{\alpha\beta} F_{\alpha\beta} (\nabla_\xi)_\alpha (\nabla_\xi)_\beta \end{aligned} \quad (43)$$

in which $F_{\alpha\beta} \equiv \sum_a J_{a\alpha} J_{a\beta}$ is an element of the symmetric $F = J^T J$ matrix.

Using the Clairaut–Schwarz theorem, the Laplacian in eq 43 can be further split into a sum over pure ($\partial^2/\partial \xi_\alpha^2$) and mixed ($\partial^2/\partial \xi_\alpha \partial \xi_\beta$) second partial derivatives as follows

$$\nabla_r^2 = \sum_\alpha \left[F_{\alpha\alpha} \frac{\partial^2}{\partial \xi_\alpha^2} + 2 \sum_{\beta > \alpha} F_{\alpha\beta} \frac{\partial^2}{\partial \xi_\alpha \partial \xi_\beta} \right] \quad (44)$$

The pure derivatives in eq 44 can be straightforwardly represented by standard central-difference formulae along each of the lattice vectors; at a given grid point, ξ_0 , these pure derivatives are evaluated using the following working expression (shown here for a generic function, $f(\xi)$, along L_α)

$$\left. \frac{\partial^2 f(\xi)}{\partial \xi_\alpha^2} \right|_{\xi=\xi_0} = \sum_{q=-n}^n w_q \frac{f(\xi_0 + q \delta \xi_\alpha \hat{L}_\alpha)}{\delta \xi_\alpha^2} \quad (45)$$

In this expression, the sum is over the n neighboring grid points (along L_α) located on each side of ξ_0 , and $w_q = w_{-q}$ is the central-difference coefficient¹²⁰ for the q -th neighboring grid point. As such, the finite-difference representation of a pure second derivative results in a $(2n+1)$ -point stencil along the given grid direction with an associated discretization error of $O(\delta \xi_\alpha^{2n})$. The default option in `exx` is $n=3$ with a discretization error of $O(\delta \xi_\alpha^6)$, as this choice furnishes well-converged values for all EXX-related quantities.^{43,80,98} In this case, the corresponding central-difference coefficients¹²⁰ are given by $w_0 = -49/18$, $w_1 = +3/2 = w_{-1}$, $w_2 = -3/20 = w_{-2}$, and $w_3 = +1/90 = w_{-3}$.

While the pure derivatives in eq 44 can be accurately and efficiently evaluated using standard central-difference techniques, there is considerable flexibility when evaluating the mixed derivatives in this expression. Here, we remind the reader that direct calculation of each mixed derivative $\partial^2/\partial \xi_\alpha \partial \xi_\beta$ in eq 44 would require consecutive finite-difference evaluations of the $\partial/\partial \xi_\alpha$ and $\partial/\partial \xi_\beta$ first derivatives. However, the number of stencil points in such an approach would scale quadratically with n ^{121,122} and would therefore result in a substantially more expensive EXX algorithm for non-orthogonal simulation cells.

3.3.1. Natan–Kronik (NK) Representation of ∇^2 : Elimination of Mixed Derivatives via Auxiliary Grid Directions. To alleviate this quadratic complexity, we follow the approach proposed by Natan, Kronik, and coworkers,¹²² which has roots in earlier work by Brandt and Diskin¹²¹ (in the two-dimensional (2D) theory of sonic flow), and will be referred to as NK throughout the rest of the study. Before describing the NK approach for treating three-dimensional (3D) general/non-orthogonal cells (as well as our algorithmic implementation for dealing with fluctuating cells during constant-pressure simulations in `exx`), we first review the core idea behind the NK approach, that is, the use of grid-resolved auxiliary direction(s) to eliminate the computationally expensive mixed derivative(s) in eq 44.

To do so, we will first consider the simplest non-orthogonal case, a 2D simulation cell with lattice vectors L_α, L_β . In this case, $F_{\alpha\beta}$ is the only non-zero off-diagonal term in F (see eq 43), and hence $\partial^2/\partial \xi_\alpha \partial \xi_\beta$ is the only mixed partial derivative in eq 44. In what follows, we will assume that $\delta \xi_\alpha \approx \delta \xi_\beta$ (i.e., the grid spacings in the L_α and L_β directions are approximately equivalent), which is typically enforced by the planewave (kinetic energy) cutoff and the fast Fourier transform (FFT) algorithm. Under this assumption, the NK approach (for a 2D non-orthogonal cell) involves choosing a single-unit auxiliary direction ($\hat{L}'$) that meets the following criteria: (i) $\hat{L}'$ is non-axial, i.e., distinct from $\hat{L}_\alpha$ and $\hat{L}_\beta$, (ii) $\hat{L}'$ lies in the plane defined by $\hat{L}_\alpha$ and $\hat{L}_\beta$ (or equivalently L_α and L_β), (iii) $\hat{L}'$ is coincident with the underlying real-space grid (i.e., $\hat{L}'$ is grid-resolved), and (iv) $\hat{L}'$ corresponds to the nearest-neighbor grid direction (i.e., $\hat{L}'$ has the smallest possible grid spacing, $\delta \xi'$).

To ensure that all the four of these criteria are satisfied, $\hat{\mathbf{L}}'$ can be written in the following compact form

$$\hat{\mathbf{L}}' = \frac{\hat{\mathbf{L}}_\alpha + \kappa \hat{\mathbf{L}}_\beta}{|\hat{\mathbf{L}}_\alpha + \kappa \hat{\mathbf{L}}_\beta|} \equiv \frac{\hat{\mathbf{L}}_\alpha + \kappa \hat{\mathbf{L}}_\beta}{d} \quad (46)$$

in which κ is defined as

$$\kappa \equiv -\left(\frac{\delta \xi_\beta}{\delta \xi_\alpha}\right) \text{sgn}[\hat{\mathbf{L}}_\alpha \cdot \hat{\mathbf{L}}_\beta] = -\left(\frac{\delta \xi_\beta}{\delta \xi_\alpha}\right) \text{sgn}[\cos \Phi] \quad (47)$$

When the angle Φ between $\hat{\mathbf{L}}_\alpha$ and $\hat{\mathbf{L}}_\beta$ is obtuse (acute), this convention for κ makes $\hat{\mathbf{L}}'$ the grid-resolved bisector of Φ (the supplementary angle to Φ), as depicted in Figure 2. This

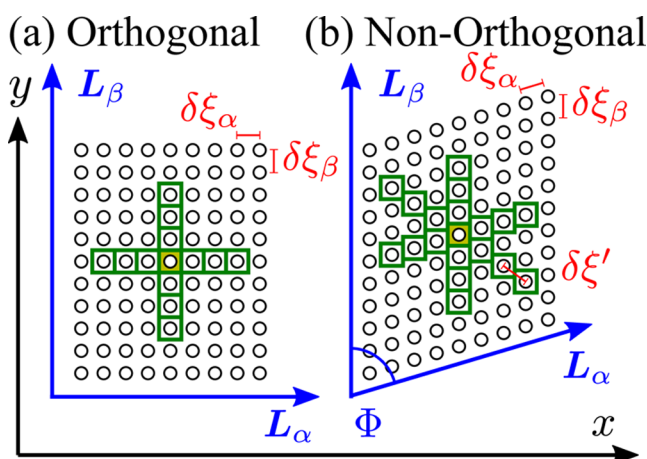


Figure 2. Graphical depiction of the grid-resolved directions used in the NK representation of the 2D Laplacian (∇_r^2) in (a) an orthogonal ($\mathbf{L}_\alpha \perp \mathbf{L}_\beta$) and (b) a non-orthogonal ($\mathbf{L}_\alpha \nparallel \mathbf{L}_\beta$) cell. Each of these discretized Laplacians is centered at a given grid point (ξ_0 , highlighted in yellow) and represented by a finite-difference stencil which covers the neighboring $\pm n$ grid points (shown here for $n = 3$) in each required derivative direction (see eq 45). In the 2D orthogonal case, the NK Laplacian takes on the standard form for ∇_r^2 and includes pure derivatives $\partial^2/\partial \xi_\alpha^2$ and $\partial^2/\partial \xi_\beta^2$ along lattice vectors $\mathbf{L}_\alpha$ and $\mathbf{L}_\beta$ (with corresponding grid spacings $\delta \xi_\alpha$ and $\delta \xi_\beta$). In the 2D non-orthogonal case, the NK Laplacian (in addition to $\partial^2/\partial \xi_\alpha^2$ and $\partial^2/\partial \xi_\beta^2$) replaces the mixed derivative $\partial^2/\partial \xi_\alpha \partial \xi_\beta$ with a pure derivative $\partial^2/\partial \xi'^2$ along a grid-resolved auxiliary direction $\hat{\mathbf{L}}'$ (with grid spacing $\delta \xi'$). To maximize the accuracy of this finite-difference Laplacian, the nearest-neighbor grid-resolved direction (i.e., with the smallest possible $\delta \xi'$) is chosen as $\hat{\mathbf{L}}'$: when the angle Φ between $\mathbf{L}_\alpha$ and $\mathbf{L}_\beta$ is obtuse (acute), $\hat{\mathbf{L}}'$ is chosen to be the grid-resolved bisector of Φ (the supplementary angle to Φ , as shown above).

choice for $\hat{\mathbf{L}}'$ also has the smallest possible $\delta \xi'$, which allows us to retain the highest degree of accuracy (at a given discretization order) in the finite-difference representation of ∇_r^2 (see eq 45).

With these expressions in hand, the first and second partial derivatives with respect to ξ' (the coordinate associated with $\hat{\mathbf{L}}'$) take on the following form

$$\frac{\partial}{\partial \xi'} = \frac{1}{d} \left(\frac{\partial}{\partial \xi_\alpha} + \kappa \frac{\partial}{\partial \xi_\beta} \right) \quad (48)$$

and

$$\frac{\partial^2}{\partial \xi'^2} = \frac{1}{d^2} \left(\frac{\partial^2}{\partial \xi_\alpha^2} + \kappa^2 \frac{\partial^2}{\partial \xi_\beta^2} + 2\kappa \frac{\partial^2}{\partial \xi_\alpha \partial \xi_\beta} \right) \quad (49)$$

Equation 49 can then be rearranged to express the mixed partial derivative

$$\frac{\partial^2}{\partial \xi_\alpha \partial \xi_\beta} = \frac{1}{2\kappa} \left(-\frac{\partial^2}{\partial \xi_\alpha^2} - \kappa^2 \frac{\partial^2}{\partial \xi_\beta^2} + d^2 \frac{\partial^2}{\partial \xi'^2} \right) \quad (50)$$

as a linear combination of pure derivatives along the $\hat{\mathbf{L}}_\alpha$ and $\hat{\mathbf{L}}_\beta$ unit lattice vectors, as well as the $\hat{\mathbf{L}}'$ unit auxiliary vector.¹²² After plugging eq 50 into 44, one arrives at the NK Laplacian with a total of $N_{\text{pure}} = 3$ pure derivatives, i.e.,

$$\nabla_r^2 = \left(F_{\alpha\alpha} - \frac{F_{\alpha\beta}}{\kappa} \right) \frac{\partial^2}{\partial \xi_\alpha^2} + (F_{\beta\beta} - \kappa F_{\alpha\beta}) \frac{\partial^2}{\partial \xi_\beta^2} + \frac{F_{\alpha\beta} d^2}{\kappa} \frac{\partial^2}{\partial \xi'^2} \quad (51)$$

each of which can now be accurately and efficiently evaluated using standard central-difference techniques (see eq 45), and the computationally expensive ($O(n^2)$) direct evaluation of the mixed derivative is completely avoided.

For the general 3D case, up to three lattice vectors can be mutually non-orthogonal. For each pair of non-orthogonal lattice vectors, the corresponding off-diagonal element in $\mathbf{F}$ will be non-zero, thereby necessitating the corresponding mixed derivative in eq 44. In this work, we follow the original NK prescription^{122,123} in which the pure derivatives in eq 44 are always evaluated along the unit lattice directions, $\{\hat{\mathbf{L}}_1, \hat{\mathbf{L}}_2, \hat{\mathbf{L}}_3\}$. Evaluation of the mixed derivative(s) in eq 44 will then require one ($N_{\text{aux}} = 1$) to three ($N_{\text{aux}} = 3$) additional grid-resolved unit auxiliary directions, $\{\hat{\mathbf{L}}'_p\}$, each of which can be written as a linear combination of the unit lattice vectors

$$\hat{\mathbf{L}}'_p = \sum_{\alpha} a_{p\alpha} \hat{\mathbf{L}}_\alpha \quad p = 1, 2, \dots, N_{\text{aux}} \quad (52)$$

In this expression, the expansion coefficients, $\{a_{p\alpha}\}$, are then chosen to satisfy all of the requirements of the 3D NK approach (see Section 3.3.2).

Following the procedure described above, we take the first and second partial derivatives of eq 52 with respect to ξ'_p (the coordinate associated with $\hat{\mathbf{L}}'_p$), which yields

$$\frac{\partial}{\partial \xi'_p} = \sum_{\alpha} a_{p\alpha} \frac{\partial}{\partial \xi_\alpha} \quad (53)$$

and

$$\frac{\partial^2}{\partial \xi_p'^2} = \sum_{\alpha} \left(a_{p\alpha}^2 \frac{\partial^2}{\partial \xi_\alpha^2} + 2 \sum_{\beta > \alpha} a_{p\alpha} a_{p\beta} \frac{\partial^2}{\partial \xi_\alpha \partial \xi_\beta} \right) \quad (54)$$

Unlike eq 49, the pure derivative along a given auxiliary direction in eq 54 generally contains contributions from more than one mixed derivative (since $a_{p\alpha} a_{p\beta}$ is generally non-vanishing). To address this issue, the NK approach seeks to find a linear combination, $\sum_p b_p (\partial^2/\partial \xi_p'^2)$, that has the same mixed derivative contribution as that in eq 44, i.e.,

$$\sum_p b_p \sum_{\alpha, \beta > \alpha} a_{p\alpha} a_{p\beta} \frac{\partial^2}{\partial \xi_\alpha \partial \xi_\beta} = \sum_{\alpha, \beta > \alpha} F_{\alpha\beta} \frac{\partial^2}{\partial \xi_\alpha \partial \xi_\beta} \quad (55)$$

This expression can be written in matrix form

$$\begin{bmatrix} a_{11}a_{12} & a_{21}a_{22} & a_{31}a_{32} \\ a_{11}a_{13} & a_{21}a_{23} & a_{31}a_{33} \\ a_{12}a_{13} & a_{22}a_{23} & a_{32}a_{33} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \equiv \mathbf{M}\mathbf{b} = \begin{bmatrix} F_{12} \\ F_{13} \\ F_{23} \end{bmatrix} \quad (56)$$

whose solution ($\mathbf{b}$) can be used to eliminate the mixed derivatives in eq 44 and derive the following working expression for the 3D NK Laplacian

$$\nabla_r^2 = \sum_{\alpha} \left(F_{\alpha\alpha} - \sum_p b_p a_{p\alpha}^2 \right) \frac{\partial^2}{\partial \xi_{\alpha}^2} + \sum_p b_p \frac{\partial^2}{\partial \xi_p'^2} \quad (57)$$

following the analogous procedure used above to derive eqs 50 and 51 for the 2D non-orthogonal case. In this expression, each pure derivative can again be accurately and efficiently evaluated using standard central-difference techniques (see eq 45), thereby avoiding the computationally expensive ($O(n^2)$) direct evaluation of the mixed derivatives. Using the approach outlined here,¹²³ the number of auxiliary directions (N_{aux}) is typically equal to the number (N_{off}) of non-zero off-diagonal elements in F , thereby leading to a total of $N_{\text{pure}} = N_{\text{off}} + 3$ pure derivatives in eq 57 (and a corresponding central-difference stencil which contains $N_{\text{stcl}} = 2nN_{\text{pure}} + 1$ points via eq 45). In the orthorhombic case, $N_{\text{off}} = 0$ and the NK Laplacian in eq 57 reduces to the standard Laplacian with $N_{\text{pure}} = 3$ pure derivatives along the lattice directions.

3.3.2. Algorithmic Implementation of the NK Scheme for Fluctuating Simulation Cells. During constant-pressure (NpH/NpT) AIMD simulations, the size and shape of the cell will constantly change due to instantaneous fluctuations and/or ongoing phase transitions throughout the trajectory. During such fluctuations, the number of auxiliary directions required to evaluate eq 57 could range from zero (e.g., orthorhombic) to three (e.g., triclinic). As such, we have implemented an automated algorithm in `exx` (executed at the beginning of each MD step) that chooses a set of auxiliary directions which meets all of the requirements of the NK approach (see Algorithm 1) and holds for the 2D and 3D non-orthogonal cases described above (i.e., eqs 51 and 57). In particular, this algorithm identifies a set of grid-resolved unit auxiliary directions, $\{\hat{\mathbf{L}}_p'\}$, that satisfy the following criteria: (i) each $\hat{\mathbf{L}}_p'$ is non-axial, i.e., distinct from the $\hat{\mathbf{L}}_1, \hat{\mathbf{L}}_2$, and $\hat{\mathbf{L}}_3$ unit lattice directions; (ii) each $\hat{\mathbf{L}}_p'$ has the minimum possible grid spacing $\delta\xi_p'$; and (iii) the $\mathbf{M}$ matrix constructed using $\{\hat{\mathbf{L}}_p'\}$ is non-singular (see eq 56).

Input into Algorithm 1 is $\hat{\mathbf{v}}$, a list containing $N_q \gg 3$ candidate non-axial auxiliary directions, $\hat{\mathbf{L}}_q' = \sum_{\alpha} \tilde{a}_{q\alpha} \hat{\mathbf{L}}_{\alpha}$, each of which has been sorted (in ascending order) by grid spacing ($\delta\xi_q'$); by providing this list as input, criterion (i) is automatically satisfied. To generate $\hat{\mathbf{v}}$, we start from a reference grid point (ξ_0) and sweep through surrounding shells of grid points ($G_S = \{\xi \in \text{NINT}[\max_{\alpha} |\xi_{\alpha} - (\xi_0)_{\alpha}| / \delta\xi_{\alpha}] = S\}$ for $S = 1, 2, \dots$) to locate $\{\hat{\mathbf{L}}_q'\}$.¹²⁴ Defining $\delta\xi_S$ as the largest spacing seen in the first shell (i.e., $\delta\xi_S \equiv \max_{\xi \in G_1} |\xi - \xi_0|$), the search stops at the S -th shell if $|\xi - \xi_0| > \delta\xi_S \forall \xi \in G_S$; doing so efficiently ensures that we do not miss any of the first N_q candidate auxiliary directions. In practice, `exx` uses a default value of $N_q = 15$, which is larger than the 10 non-axial grid points in G_1 ¹²⁴ and should suffice for almost all cases; if necessary, N_q can be increased for

simulation cells with very small ($\approx 0^\circ$) or very large ($\approx 180^\circ$) angles between lattice directions.

Algorithm 1 Choice of NK Auxiliary Directions

```

Input:  $\hat{\mathbf{v}}[q] = \sum_{\alpha} \tilde{a}_{q\alpha} \hat{\mathbf{L}}_{\alpha}$  ( $q = 1, 2, \dots, N_q$ )
 $p \leftarrow 1$ ;  $\mathbf{a} \leftarrow [\mathbf{0}]_{3 \times 3}$ ;  $\mathbf{M} \leftarrow [\mathbf{0}]_{3 \times 3}$ 
for ( $q = 1, N_q$ ) do
   $\{a_{p1}, a_{p2}, a_{p3}\} \leftarrow \{\tilde{a}_{q1}, \tilde{a}_{q2}, \tilde{a}_{q3}\}$   $\triangleright$  propose candidate
   $\{M_{1p}, M_{2p}, M_{3p}\} \leftarrow \{a_{p1}a_{p2}, a_{p1}a_{p3}, a_{p2}a_{p3}\}$ 
  if ( $p == 1$ ) then
    is_accepted  $\leftarrow$  TRUE
  else if ( $p == 2$ ) then
    is_accepted  $\leftarrow$  ( $M_{:,1} \nparallel M_{:,2}$ )
  else if ( $p == 3$ ) then
    is_accepted  $\leftarrow$  ( $\det \mathbf{M} \neq 0$ )
  end if
  if (is_accepted) then
     $p \leftarrow p + 1$ 
    if ( $p > 3$ ) then
      break
    end if
  end if
end for

```

In a loop over $\hat{\mathbf{L}}_q'$ in $\hat{\mathbf{v}}$, Algorithm 1 now seeks to find the set of auxiliary directions, $\{\hat{\mathbf{L}}_p'\}$, which satisfy the remaining criteria (ii) and (iii). Since $\hat{\mathbf{L}}_1'$ (the first element in $\hat{\mathbf{v}}$) has the smallest grid spacing, this candidate auxiliary direction is automatically assigned to be $\hat{\mathbf{L}}_1'$; algorithmically speaking, this corresponds to setting $a_{1\alpha} = \tilde{a}_{1\alpha}$ for $\alpha = 1, 2, 3$ and populating $\mathbf{M}_{:,1}$ (i.e., the first column of $\mathbf{M}$). After successfully identifying $\hat{\mathbf{L}}_1'$, the loop then continues to the next element of $\hat{\mathbf{v}}$ in the search for $\hat{\mathbf{L}}_2'$. In a loop over q (which runs from 2, 3, ...), $\hat{\mathbf{L}}_q'$ becomes the proposed candidate for $\hat{\mathbf{L}}_2'$; that is, $a_{2\alpha}$ is temporarily assigned to $\tilde{a}_{q\alpha}$ and $\mathbf{M}_{:,2}$ is populated accordingly. If $\mathbf{M}_{:,2}$ is non-parallel to $\mathbf{M}_{:,1}$ (determined via the Cauchy–Schwarz inequality), then $\hat{\mathbf{L}}_q'$ is assigned to be $\hat{\mathbf{L}}_2'$; if not, the loop continues to the next element in $\hat{\mathbf{v}}$. After successfully identifying $\hat{\mathbf{L}}_2'$, the loop then continues to the next element of $\hat{\mathbf{v}}$ in the search for $\hat{\mathbf{L}}_3'$; that is, $a_{3\alpha}$ is temporarily assigned to $\tilde{a}_{q\alpha}$ and $\mathbf{M}_{:,3}$ is again populated accordingly. If $\mathbf{M}$ is non-singular (i.e., $\det \mathbf{M} \neq 0$), then $\hat{\mathbf{L}}_q'$ is assigned to be $\hat{\mathbf{L}}_3'$ and Algorithm 1 terminates; if not, the loop continues to the next element in $\hat{\mathbf{v}}$.

Upon successful execution, the output of Algorithm 1 is $\{\hat{\mathbf{L}}_p'\}$, the final set of auxiliary directions (which satisfies all of the criteria given above), and $\mathbf{M}$, which can be trivially inverted to obtain $\mathbf{b}$ via eq 56. With $\{\hat{\mathbf{L}}_p'\}$ and $\mathbf{b}$ in hand, the NK Laplacian in eq 57 can now be evaluated, allowing for a computationally efficient treatment of non-orthogonal cells during constant-pressure simulations in `exx` (see Section 4.2 for a detailed computational timings profile of CPMD simulations of ice Ih, II, and III at the hybrid DFT level using this approach).

3.4. Extension of the `exx` Module: Computation of the EXX Contribution to the Stress Tensor. Using Algorithm 1 in conjunction with the NK representation for the Laplacian (see Section 3.3), the `exx` module is now equipped to solve the PE for systems with fluctuating and non-orthogonal simulation cells. For each overlapping $\langle ij \rangle$ pair, the `exx` module leverages this new capability to compute the corresponding MLWF-product potential ($\tilde{v}_{ij}(\vec{r})$) during Step IV (see Figure 1). This quantity is the cornerstone of our

MLWF-based EXX approach and is required for evaluating all of the EXX-related contributions (E_{xx} , $\{\tilde{D}_{\text{xx}}^i(\mathbf{r})\}$, σ_{xx}) to the CPMD equations of motion in eqs 10–12. Since the evaluation of E_{xx} and $\{\tilde{D}_{\text{xx}}^i(\mathbf{r})\}$ have been discussed extensively in PAPER-I,⁹⁸ we focus the following discussion on the extensions to `exx` needed for computing σ_{xx} via eq 26 during Step V (see Figure 1). In this working expression, one can immediately see that a numerically accurate evaluation of the $\langle ij \rangle$ contribution to σ_{xx} only requires integration over Ω_{ij} (in analogy to the evaluation of E_{xx} via eq 6). In fact, once the gradient of $\tilde{v}_{ij}(\mathbf{r})$ is evaluated (*vide infra*), the computation of σ_{xx} follows a similar procedure to that used for E_{xx} (see Section III.C.5 of PAPER-I⁹⁸): (i) for each overlapping $\langle ij \rangle$ pair, integration over the $\Theta(C_{ij}R_{\text{PE}})$ subdomain on a given MPI process is efficiently parallelized over N_{thread} OpenMP threads and (ii) partial summations over the $\langle ij \rangle$ pairs assigned to each MPI process are then accumulated via MPI_REDUCE (using the MPI_SUM operation) to form σ_{xx} with minimal associated communication (i.e., 3×3 double-precision numbers per MPI process).

Since the integral needed to evaluate each $\langle ij \rangle$ contribution to σ_{xx} is restricted to the $\Theta(C_{ij}R_{\text{PE}})$ subdomain, each component of the Cartesian gradient of $\tilde{v}_{ij}(\mathbf{r})$ in eq 26 (i.e., $\partial \tilde{v}_{ij}(\mathbf{r}) / \partial r_a$) only needs to be evaluated on $\Theta(C_{ij}R_{\text{PE}})$ as well. With the Jacobian derived in eq 42, the Cartesian gradient operator, ∇_r , can be written in terms of the (pure) directional derivatives along the unit lattice vectors, $\nabla_{\hat{\mathbf{e}}_i}$, via $\nabla_r = \mathbf{J} \nabla_{\hat{\mathbf{e}}}$; as such, there is no need to introduce auxiliary lattice directions as done above when using the NK representation of ∇^2 . In analogy to eq 45, the derivatives in eq 26 can be accurately and efficiently evaluated using standard central-difference formulae along each of the lattice vectors (shown here for a generic function, $f(\xi)$, along L_a)

$$\left. \frac{\partial f(\xi)}{\partial \xi_a} \right|_{\xi=\xi_0} = \sum_{q=-n}^n w_q \frac{f(\xi_0 + q \delta \xi_a \hat{L}_a)}{\delta \xi_a} \quad (58)$$

In this expression, the sum is over the n neighboring grid points located on each side of ξ_0 (along L_a), and the corresponding anti-symmetric $(2n + 1)$ -point stencil uses the following central-difference coefficients (with $w_q = -w_{-q}$):¹²⁰ $w_0 = 0$, $w_1 = +3/4$, $w_2 = -3/20$, and $w_3 = +1/60$. The default option in `exx` is $n = 3$ with a discretization error of $O(\delta \xi_a^{2n}) = O(\xi_a^6)$, as this choice furnishes well-converged values for E_{xx} and $\{\tilde{D}_{\text{xx}}^i(\mathbf{r})\}$ ^{43,80,98} as well as σ_{xx} .

Here, we stress to the reader that eq 26 provides an analytical expression for $\sigma_{\text{xx}} = \partial E_{\text{xx}} / \partial \mathbf{h}$ (i.e., the cell derivatives of E_{xx}), and the finite-difference evaluation of $\partial \tilde{v}_{ij}(\mathbf{r}) / \partial r_a$ (via eq 58) is needed since $\tilde{v}_{ij}(\mathbf{r})$ is not analytical and only known on the real-space grid. As such, the approach for computing σ_{xx} in `exx` is simultaneously more accurate and more computationally efficient than numerical differentiation of E_{xx} with respect to $\mathbf{h}$ (which would require perturbing each element of $\mathbf{h}$ by $\pm \delta$ and then re-computing E_{xx} for each of these cell displacements). Unlike the numerical differentiation of E_{xx} with respect to $\mathbf{h}$, which requires $2 - 12 \times$ the cost of evaluating E_{xx} (depending on the number of non-zero elements in $\mathbf{h}$), the computational complexity of evaluating eq 26 in `exx` is comparable to a single application of the Laplacian during the CG solution of the PE. As such, computation of the EXX contribution to the stress tensor (via σ_{xx}) only requires a small fraction of the cost associated with computing E_{xx} for all of the

simulations performed in this work, the cost associated with σ_{xx} was <1% of the wall time spent in the `exx` module.

4. ACCURACY AND PERFORMANCE

In this section, we critically assess the accuracy and computational performance of `exx`, which uses a dual-level MPI/OpenMP parallelization scheme to exploit both inter-node and intranode HPC resources during hybrid DFT simulations of large-scale condensed-phase systems. We will focus on the extensions to `exx` introduced in this work (see Section 3) that enable constant-pressure (NpH/NpT) simulations at the hybrid DFT level for general/non-orthogonal cells using the CP module of QE.¹⁰⁰ We begin by exploring the accuracy of the extended `exx` module when computing E_{xx} and σ_{xx} for a variety of condensed-phase systems, including ambient liquid water, a benzene molecular crystal polymorph, and semi-conducting crystalline silicon, in Section 4.1. We then study the effects of lattice symmetry on computational complexity in the `exx` module via a detailed case study of three different ice polymorphs (Ih, II, and III) in Section 4.2. In particular, we perform and analyze a series of short NpT CPMD simulations on these ice phases (in conjunction with specific angular constraints on each cell tensor) to investigate how the number of non-orthogonal cell directions affects the performance of `exx`. In Section 4.3, we investigate the computational performance and parallel scaling of `exx` during constant-pressure simulations of large-scale condensed-phase systems via a strong and weak scaling analysis of liquid water (i.e., ranging from $(\text{H}_2\text{O})_{64}$ to $(\text{H}_2\text{O})_{256}$) in the NpT ensemble (in analogy to that performed in PAPER-I⁹⁸ in the NVT ensemble). In all cases, the performance of `exx` will be examined across a wide array of HPC architectures, including Mira IBM Blue Gene/Q, Cori Haswell, and Cori KNL.

4.1. Accuracy of the EXX Contributions to the Energy and Cell Forces. In PAPER-I,⁹⁸ we used a snapshot of ambient liquid water (i.e., $(\text{H}_2\text{O})_{64}$ at the equilibrium density, 85 Ry planewave cutoff) to determine the default `exx` parameters used in QE. Here, we remind the reader that there are five key parameters used when performing a hybrid DFT calculation with `exx` (see Section 3.1): (i) R_{pair} , a radial cutoff used to determine whether or not two MLWFs, $\tilde{\phi}_i$ and $\tilde{\phi}_j$, are an overlapping $\langle ij \rangle$ pair based on their center-to-center distance (i.e., $|\mathbf{C}_i - \mathbf{C}_j| \leq R_{\text{pair}}$); (ii)–(iii) R_{PE}^s and $R_{\text{PE}}^{\text{ns}}$, the radii of the fixed-size spherical domains over which PE (eq 7) is solved for the near-field potential ($\tilde{v}(\mathbf{r})$) for self (s , $\langle ii \rangle$) and non-self (ns , $\langle ij \rangle$) overlapping pairs; and (iv)–(v) R_{ME}^s and $R_{\text{ME}}^{\text{ns}}$, the outer radii of the concentric spherical shells (with inner radii R_{PE}^s and $R_{\text{PE}}^{\text{ns}}$) over which the ME (eqs 8 and 9) is performed for the far-field potential ($\tilde{v}(\mathbf{r})$) for the self and non-self-overlapping pairs. In doing so, we demonstrated that these parameters govern both the accuracy and performance of `exx`, and judicious choices for each ensured rapid convergence of E_{xx} and $\{\tilde{D}_{\text{xx}}^i(\mathbf{r})\}$ for $(\text{H}_2\text{O})_{64}$ (see Figures 6 and 7 as well as Sections IV.A.1 and IV.A.2 in PAPER-I⁹⁸).

To test the new capabilities of the extended `exx` module (i.e., E_{xx} and σ_{xx} for general/non-orthogonal cells) as well as the transferability of the default `exx` parameters, we now investigate the convergence of E_{xx} and σ_{xx} on three different condensed-phase systems: ambient liquid water, a benzene molecular crystal polymorph, and semi-conducting crystalline silicon. For consistency with PAPER-I,⁹⁸ we included a snapshot of ambient liquid water at the equilibrium density;

however, we have doubled the system size to $(\text{H}_2\text{O})_{128}$ ($L = 15.645$ Å) to investigate any finite-size effects on the exx parameters from using $(\text{H}_2\text{O})_{64}$ in PAPER-I⁹⁸ and increased the plane-wave cutoff from 85 to 150 Ry (a typical setting employed during constant-pressure NpH/NpT simulations of aqueous systems). To go beyond liquid water, we also carried out a case study on the monoclinic benzene-II molecular crystal polymorph¹²⁵—a non-orthogonal and anisotropic system with a similar band gap. To do so, we considered a $(\text{C}_6\text{H}_6)_{16}$ system (with $a = 10.834$ Å, $b = 10.752$ Å, $c = 15.064$ Å, $\alpha = \gamma = 90^\circ$, and $\beta = 110^\circ$) constructed from a $2 \times 2 \times 2$ supercell of the experimentally assigned unit cell (which contains two benzene molecules).¹²⁵ During all calculations on $(\text{C}_6\text{H}_6)_{16}$, we used a 110 Ry plane-wave cutoff. As an even more stringent test on the exx module, we also considered semi-conducting crystalline silicon—a system with a significantly smaller band gap and therefore substantially more diffuse (less-localized) MLWFs. In this case, we constructed a cubic Si_{216} snapshot (with $L = 16.29$ Å) as a $3 \times 3 \times 3$ supercell of the classic eight-atom diamond structure in a cubic unit cell. During all calculations on Si_{216} , we used a 35 Ry plane-wave cutoff. A graphical depiction of each of these three systems can be found in the inset of Figure 3.

Using the procedure outlined in Section IV.A of PAPER-I⁹⁸ to determine the default exx parameter values, we first performed a series of reference single-point energy calculations on each of these systems at the PBE0^{128,129} level. This was accomplished by self-consistently solving for the electronic ground state with all EXX parameters in the exx module set to their largest possible values: R_{pair} , R_{ME}^s , and $R_{\text{ME}}^{\text{ns}}$ are set to the radius of the largest sphere that can be contained within each simulation cell; $R_{\text{PE}}^s = R_{\text{PE}}^{\text{ns}} = R_{\text{ME}}^s - n \max_{\alpha} \{\delta \xi_{\alpha}^s\}$ (with $n = 3$), which provides us with a thin shell (halo region) on the real-space grid needed for the PE boundary conditions. The E_{xx} and σ_{xx} values obtained from these calculations are then used as reference values (i.e., $E_{\text{xx}}^{\text{ref}}$ and $\sigma_{\text{xx}}^{\text{ref}}$) to gauge the accuracy of the exx module when computing these quantities using different parameter values. In analogy to the previously used error metric for $\tilde{D}_{\text{xx}}(r)$ (see eq 36, Figure 7, and Section IV.A.2 in PAPER-I⁹⁸), we define the relative errors in E_{xx} and σ_{xx} as follows

$$\mathcal{E}(E_{\text{xx}}) = \frac{\|E_{\text{xx}} - E_{\text{xx}}^{\text{ref}}\|_1}{\|E_{\text{xx}}^{\text{ref}}\|_1} \quad (59)$$

$$\mathcal{E}(\sigma_{\text{xx}}) = \frac{\|\sigma_{\text{xx}} - \sigma_{\text{xx}}^{\text{ref}}\|_1}{\|\sigma_{\text{xx}}^{\text{ref}}\|_1} \quad (60)$$

in which $\|\cdot\|_1$ denotes the one-norm of the inserted quantity. Based on these relative error definitions, we first investigated the accuracy of the default EXX parameters (which were determined using $(\text{H}_2\text{O})_{64}$ at the equilibrium density, 85 Ry plane-wave cutoff) when computing E_{xx} and σ_{xx} for the three systems described above (see Table 1). When assessing the accuracy of the default exx parameters for these systems, we again follow PAPER-I⁹⁸ by computing E_{xx} and σ_{xx} using the converged MLWFs obtained during the corresponding reference calculations; a more detailed (and fully self-consistent) investigation of these parameters in anisotropic/heterogeneous systems will be addressed in a forthcoming paper in this series.

Table 1. Relative One-Norm Errors in E_{xx} and σ_{xx} Using the Default Exx Parameters in QE (i.e., $R_{\text{pair}} = 8.0$ Bohr, $R_{\text{PE}}^s = 6.0$ Bohr, $R_{\text{PE}}^{\text{ns}} = 5.0$ Bohr, $R_{\text{ME}}^s = 10.0$ Bohr, and $R_{\text{ME}}^{\text{ns}} = 7.0$ Bohr) for Three Different Condensed-Phase Systems^a

	$(\text{H}_2\text{O})_{64}$ (cubic)	$(\text{H}_2\text{O})_{128}$ (cubic)	$(\text{C}_6\text{H}_6)_{16}$ (monoclinic)	Si_{216} (cubic)
$\mathcal{E}(E_{\text{xx}})$ (in %)	0.02 ^b	0.02	0.13	1.72
$\mathcal{E}(\sigma_{\text{xx}})$ (in %)		0.03	0.19	2.29

^aAll errors were computed using eqs 59 and 60 with respect to the reference values for these quantities obtained with all exx parameters set to their largest possible values. See text for more details. ^bSystem used to determine default exx parameters in PAPER-I.⁹⁸

As depicted in Table 1, the default exx parameters reproduce E_{xx} with very high fidelity for $(\text{H}_2\text{O})_{128}$ —in this case, the accuracy of exx is equivalent to that found in PAPER-I⁹⁸ for $(\text{H}_2\text{O})_{64}$, that is, $\mathcal{E}(E_{\text{xx}}) \approx 0.02\%$. As expected, the default parameters determined in PAPER-I⁹⁸ seem to be well-converged for ambient liquid water with respect to both system size and basis set size. When applied to the $(\text{C}_6\text{H}_6)_{16}$ molecular crystal, we find that the default exx parameters are also quite transferable, yielding $\mathcal{E}(E_{\text{xx}}) \approx 0.10\%$. In this case, we attribute the slight decrease in accuracy to the increased variability in the MLWF spreads in the benzene molecular crystal, which contains C–H σ -bonds (which have a similar spread to the MLWFs in liquid water) as well as a set of more diffuse C–C τ -bonds; as such, converging E_{xx} in this system will require (on average) a slightly larger support for $\tilde{\rho}_{ij}(r)$ during the solution of PE (*vide infra*). In Si_{216} , the MLWFs are significantly more delocalized than those in both liquid water and the benzene molecular crystal due to the smaller band gap in this semi-conductor.¹⁰¹ As such, the default exx parameters now yield a more sizable error of $\mathcal{E}(E_{\text{xx}}) \approx 1.72\%$, as tight convergence of E_{xx} in this more challenging system will require an increase in R_{PE}^s and $R_{\text{PE}}^{\text{ns}}$ (to provide a larger support for the more diffuse $\tilde{\rho}_{ij}(r)$) as well as R_{pair} (to account for the increased number of overlapping MLWF pairs). Here, we also note that $\mathcal{E}(\sigma_{\text{xx}})$ is slightly (but consistently) larger than $\mathcal{E}(E_{\text{xx}})$ in all the three of these cases; this systematic trend will be discussed below.

Since the accuracy required during an E_{xx} (or σ_{xx}) calculation will depend on the system and/or application, we now perform a systematic study of how these quantities converge in $(\text{H}_2\text{O})_{128}$, $(\text{C}_6\text{H}_6)_{16}$, and Si_{216} as a function of the exx parameters. Following the procedure outlined in PAPER-I,⁹⁸ we again start with the converged MLWFs obtained during the reference calculations described above (in which all exx parameters were set to their largest possible values). We then track how E_{xx} and σ_{xx} converge with respect to (i) changes in R_{pair} while keeping all other exx variables at their reference values (Figure 3, top panel) and (ii) simultaneous changes in R_{PE}^s and $R_{\text{PE}}^{\text{ns}}$ while again keeping all other exx variables at their reference values (Figure 3, bottom panel). As mentioned above, a more detailed (and fully self-consistent) investigation of these parameters (for a number of different anisotropic/heterogeneous systems) will be addressed in a forthcoming paper in this series. As depicted in Figure 3, the convergence behavior of $\mathcal{E}(E_{\text{xx}})$ in $(\text{H}_2\text{O})_{128}$ is essentially identical to that in $(\text{H}_2\text{O})_{64}$ (cf. Figure 6 in PAPER-I⁹⁸). For $(\text{C}_6\text{H}_6)_{16}$, we find that E_{xx} rapidly converges with both R_{pair} and $\{R_{\text{PE}}^s, R_{\text{PE}}^{\text{ns}}\}$; in this

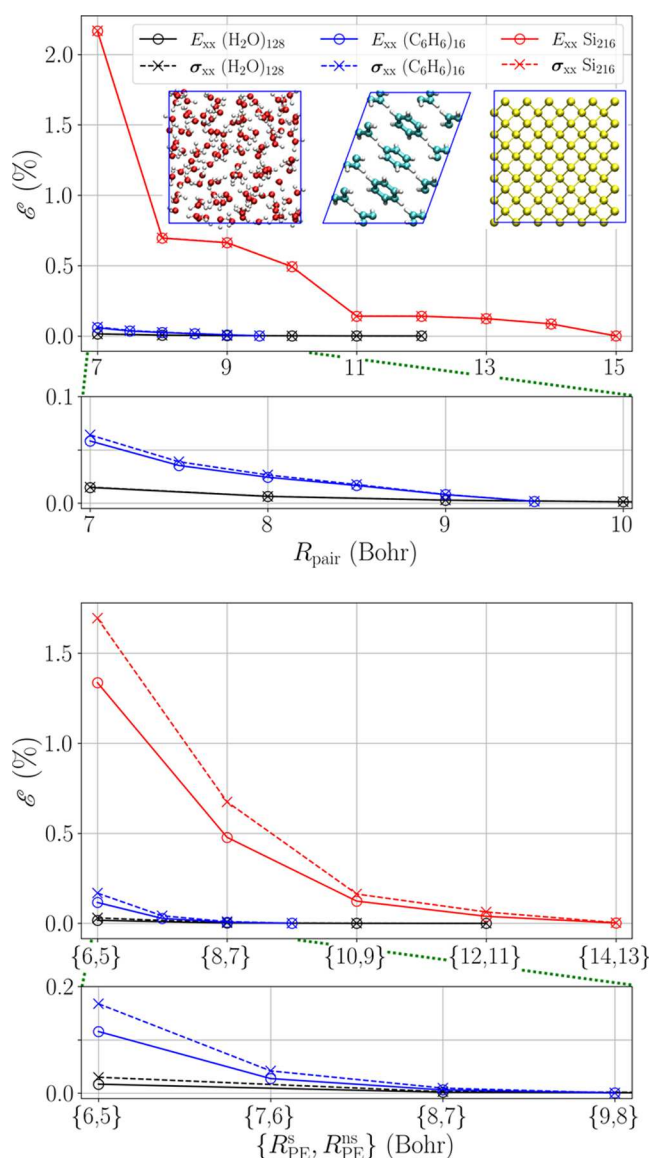


Figure 3. Convergence of E_{xx} (open circles, $\circ$) and σ_{xx} (crosses, $\times$) in $(\text{H}_2\text{O})_{128}$ (black), $(\text{C}_6\text{H}_6)_{16}$ (blue), and Si_{216} (red) as a function of R_{pair} (top panel) and $\{R_{\text{PE}}^s, R_{\text{PE}}^{\text{ns}}\}$ (bottom panel). As described in the main text, all other $\text{e}xx$ parameters were set to the maximum allowed values during each convergence study.

case, increasing the radii used during the solution of PE (to account for the more diffuse τ bonds on each benzene ring) is more important than increasing R_{pair} when tight convergence (i.e., $\mathcal{E}(E_{xx}) < 0.10\%$) is desired. For Si_{216} , the convergence of E_{xx} with respect to both R_{pair} and $\{R_{\text{PE}}^s, R_{\text{PE}}^{\text{ns}}\}$ is slower due to the substantially more delocalized MLWFs in this semi-conducting system. In this case, $\mathcal{E}(E_{xx})$ originates from the need for increased real-space domains during the solution of PE (primary contribution) as well as the inclusion of more distant overlapping MLWF pairs (secondary but still sizable contribution). Here, we find that systematically (and simultaneously) increasing both R_{PE}^s and $R_{\text{PE}}^{\text{ns}}$ leads to a smooth and exponential decay in $\mathcal{E}(E_{xx})$, reflecting the exponential decay rate of the MLWFs in this finite-gap system.¹⁰¹ As R_{pair} was increased, the observed decreases in $\mathcal{E}(E_{xx})$ are tiered (as opposed to the smoother decay seen in $(\text{H}_2\text{O})_{128}$), reflecting the crystalline structure in this atomic solid. Even in this more

challenging system, the MLWFs are still exponentially localized and therefore have a finite support in real space; as such, the $\text{e}xx$ module can still furnish E_{xx} to a pre-defined accuracy level—albeit with additional computational cost—by simply increasing the $\text{e}xx$ parameters beyond their default values. For example, “chemical accuracy” (i.e., 1 kcal/mol) in the PBE0 binding energy of Si_{216} can be achieved by increasing $\{R_{\text{PE}}^s, R_{\text{PE}}^{\text{ns}}\}$ from $\{6.0, 5.0\}$ Bohr (default values) to $\{8.0, 7.0\}$ Bohr while leaving R_{pair} at 8.0 Bohr (default value). When compared against the default setting in $\text{e}xx$, the use of these more accurate parameters leads to an $\approx 50\%$ increase in computational cost; however, this is still a significant speed-up and $\approx 60\times$ less than the cost of the reference calculation.

Since the calculation of E_{xx} and σ_{xx} uses the same overlapping MLWF pairs (cf. eqs 6 and 26), the accuracy of these quantities will primarily be governed by $\{R_{\text{PE}}^s, R_{\text{PE}}^{\text{ns}}\}$, i.e., the coverage of $\tilde{\rho}_{ij}(\mathbf{r})$ during the solution of PE. As expected, we found that $\mathcal{E}(E_{xx})$ and $\mathcal{E}(\sigma_{xx})$ exhibited a similar convergence rate with respect to these $\text{e}xx$ parameters for all three systems considered herein, although $\mathcal{E}(\sigma_{xx})$ was consistently slightly larger than $\mathcal{E}(E_{xx})$ in all cases (see Table 1 and Figure 3, top panel). While a small portion of this difference is attributed to inherent limitations when comparing relative errors in scalar and matrix quantities (cf. eqs 59 and 60), the difference between $\mathcal{E}(\sigma_{xx})$ and $\mathcal{E}(E_{xx})$ is more pronounced for smaller $\{R_{\text{PE}}^s, R_{\text{PE}}^{\text{ns}}\}$ values. Hence, we attribute this difference to the larger intrinsic error when computing the integrand of σ_{xx} in eq 26, which involves a displacement-weighted derivative of the MLWF-product potential, that is i.e., $r_b(\partial \tilde{v}_{ij}(\mathbf{r}) / \partial r_a)$, as opposed to the integrand of E_{xx} in eq 6, which only involves $\tilde{v}_{ij}(\mathbf{r})$ itself.

Here, we note in passing that the need to scan for the set of optimal parameters in $\text{e}xx$ will be largely eliminated in a forthcoming paper in this series, where the entire $\text{e}xx$ module will be restructured based on variable-size supports for each MLWF. By intrinsically accounting for the size/shape/extent of each MLWF and treating each overlapping MLWF pair according to a user-defined level of accuracy, this restructured $\text{e}xx$ module will be able to treat challenging systems such as crystalline Si as well as complex multi-component/multi-phase systems without the need to sacrifice accuracy for computational performance (or vice versa).

4.2. Computational Complexity due to Lattice Symmetry. For a more consistent comparison with the analysis of $\text{e}xx$ in PAPER-I,⁹⁸ we now refocus our discussion on aqueous systems (i.e., ice and liquid water) while assessing the computational performance of the extended $\text{e}xx$ module when treating general/non-orthogonal systems and using larger basis sets (e.g., as needed during constant-pressure simulations with fluctuating cells). To explore the effects of lattice symmetry on computational complexity, we first carried out a detailed case study on the Ih, II, and III polymorphs of ice. More specifically, we performed and analyzed short (i.e., 50 steps) NpT CPMD simulations on these ice phases (in conjunction with specific angular constraints on each lattice) to investigate how the number of non-orthogonal cell directions affects the performance of $\text{e}xx$. As a first case, we considered the orthorhombic/tetragonal/cubic lattice systems, in which evaluation of the NK Laplacian in eq 57 is the simplest and requires $N_{\text{aux}} = 0$ auxiliary grid directions (i.e., $N_{\text{pure}} = 3$ pure derivatives along the lattice directions). In this

case, we chose ice III as the example system (which is tetragonal in the absence of thermal fluctuations) and applied a series of angular constraints ($L_1 \perp L_2$, $L_1 \perp L_3$, and $L_2 \perp L_3$) to maintain orthogonality among all lattice vectors during the short NpT simulation. As a second case, we considered the monoclinic/hexagonal/rhombohedral lattice systems, in which evaluation of the NK Laplacian requires $N_{\text{aux}} = 1$ auxiliary grid direction (for a total of $N_{\text{pure}} = 4$ pure derivatives). In this case, ice Ih was chosen as the example system (which is hexagonal in the absence of thermal fluctuations), and $L_1 \perp L_2$ and $L_2 \perp L_3$ angular constraints were applied during the NpT simulation to maintain $N_{\text{pure}} = 4$. As a third case, we considered the triclinic lattice system, in which evaluation of the NK Laplacian is the most complex and requires $N_{\text{aux}} = 3$ auxiliary grid directions (for a total of $N_{\text{pure}} = 6$ pure derivatives). Here, we employed ice II as the example system; although this polymorph is rhombohedral in the absence of thermal fluctuations, we started the NpT simulation with ice II in a triclinic cell. We then allowed the NpT simulation to proceed without any angular constraints to mimic the cell fluctuations of a triclinic system with $N_{\text{pure}} = 6$ pure derivatives (rather than $N_{\text{pure}} = 4$ for a perfect rhombohedral lattice). By including these three cases (with $N_{\text{pure}} = 3, 4, 6$), this study essentially covers all seven 3D lattice systems^{123,126} and will now be used to evaluate the performance of `exx`.

Computational timings for each of these ice phases were generated using an in-house development version of QE (that is based on v5.0.2)¹²⁷ at the PBE0^{128,129} hybrid DFT level. Each ice polymorph was modeled using a simulation cell containing $(\text{H}_2\text{O})_{96}$ (each with $N_0 = 4 \times N_{\text{water}} = 384$ MLWFs) with initial snapshots taken from NpT simulations of ice Ih, II, and III at the corresponding experimental triple point (i.e., $p = 2.1$ kBar and $T = 238$ K). In ice Ih and III, proton disorder was introduced using an algorithm that enforces the Bernal–Fowler ice rules¹³⁰ as well as the additional constraint of vanishing polarization.^{131,132} For the proton-ordered ice II phase, the supercell was made by directly replicating the unit cell containing $(\text{H}_2\text{O})_{12}$ provided in ref 133. With the angular constraints described directly above, we performed a series of short CPMD simulations in the NpT ensemble (at the same p and T) for a duration of 50 steps. The pressure was controlled using a Parrinello–Rahman barostat,¹¹³ and the temperature was maintained by attaching massive Nosé–Hoover chain thermostats^{134,135} (each with a chain length of 4) to the ionic degrees of freedom. All NpT simulations were performed at the Γ -point only and employed a planewave kinetic energy cutoff of 150 Ry; the corresponding CPMD equations of motion (eqs 10–12) were integrated using the standard Verlet algorithm and a time step of 2.0 au (≈ 0.05 fs). Planewave kinetic energies were modified following Bernasconi *et al.*¹³⁶ to maintain a constant planewave kinetic energy cutoff of 130 Ry throughout each NpT simulation.¹³⁷ To ensure an adiabatic separation between the electronic and nuclear degrees of freedom, the fictitious electronic mass was set to $\mu = 100$ au; in addition, the nuclear mass of deuterium was used for each hydrogen atom. To improve the stability of the fictitious electron dynamics, mass preconditioning¹¹⁷ was applied to all Fourier components of the electronic (pseudo-) wavefunctions with a kinetic energy >25 Ry. Hamann–Schlüter–Chiang–Vanderbilt-type norm-conserving pseudopotentials^{138,139} provided by the `Qbox` package¹⁴⁰ were used to treat the interactions between the valence electrons and the ions. All `exx`-related parameters were set to the default values

determined in PAPER-I,⁹⁸ that is, $R_{\text{pair}} = 8.0$ Bohr, $R_{\text{PE}}^s = 6.0$ Bohr, $R_{\text{PE}}^{\text{ns}} = 5.0$ Bohr, $R_{\text{ME}}^s = 10.0$ Bohr, and $R_{\text{ME}}^{\text{ns}} = 7.0$ Bohr.

All timings were obtained using 1536 nodes (i.e., $\zeta \equiv N_{\text{proc}}/N_0 = 4$) on the following HPC architectures: *Mira* IBM Blue Gene/Q, *Cori* Haswell, and *Cori* KNL (see Table 2). In all cases, the reported timings were obtained using one process per node for the internode MPI parallelization (first parallelization level) and all available physical cores per node (i.e., 16 for *Mira* IBM Blue Gene/Q, 32 for *Cori* Haswell, and 68 for *Cori* KNL) for the intranode OpenMP parallelization (second parallelization level). Task-group parallelization (with $N_{\text{tg}} = 4$) was also employed to improve the computational efficiency associated with the 3D FFT operations in the non-`exx` portions of QE. Hyperthreading was fully activated on each physical core except for *Cori* KNL, where hyperthreading was deactivated due to performance degradation in both the `exx` and non-`exx` modules in QE.

For each ice phase (and on each HPC architecture), we found that the wall time associated with computing the EXX contribution to the stress tensor ($\langle t_{\text{exx}}^{\text{stress}} \rangle$) was $<0.5\%$ of the average wall time spent in the `exx` module ($\langle t_{\text{exx}} \rangle$). This is not surprising as the evaluation of eq 26 is comparable to a single CG step during the solution of the PE. As such, we will focus our discussion below on the more significant computational cost associated with solving the PE ($\langle t_{\text{exx}}^{\text{PE}} \rangle$). Since the real-space grids employed during these NpT simulations were based on a planewave cutoff of 150 Ry (which is needed for fluctuating cell simulations), both $\langle t_{\text{exx}}^{\text{PE}} \rangle$ and $\langle t_{\text{exx}} \rangle$ will be larger than that found during fixed-cell NVT simulations in `exx` with a more conventional cutoff of ≈ 85 Ry. In all cases, $\langle t_{\text{exx}}^{\text{PE}} \rangle$ comprises $\approx 30\%$ of $\langle t_{\text{exx}} \rangle$, and this finding is quite consistent with the detailed performance analysis of `exx` in PAPER-I,⁹⁸ in which $\langle t_{\text{exx}} \rangle$ was (approximately) split evenly between computation, communication, and processor idling during large-scale NVT simulations of liquid water ($(\text{H}_2\text{O})_{64} - (\text{H}_2\text{O})_{256}$) with $\zeta = 4$. On each HPC architecture, we find that $\langle t_{\text{exx}}^{\text{PE}} \rangle$ and $\langle t_{\text{exx}} \rangle$ follow the same trend, in which ice Ih has the least computational cost, followed by ice III and then ice II.

As discussed in Section 3.3, the first factor that will affect the performance of `exx` during NpT simulations is the number of grid points in the finite-difference (stencil) representation of ∇^2 ($N_{\text{stcl}} = 2nN_{\text{pure}} + 1$), which directly depends on the total number of pure derivatives ($N_{\text{pure}} = N_{\text{aux}} + 3$) in the NK Laplacian (see eq 57). For typical condensed-phase systems such as liquid water, $n = 3$ (with a discretization error of $O(\delta\zeta^6)$) is sufficiently converged when computing all EXX-related quantities;^{43,80,98} with this choice for n , $N_{\text{stcl}} = 19, 25, 37$ for the (angularly constrained) NpT simulations of ice III, Ih, and II reported in Table 2. The second factor that will affect the performance is the number of grid points in the Poisson subdomain for each overlapping MLWF pair. Since there are significantly more non-self than self-pairs, the computational cost associated with solving the PE is dominated by the non-self-pairs;⁹⁸ as such, we only report the number of grid points in each $\Theta(C_{ij}, R_{\text{PE}}^{\text{ns}})$ subdomain. While the number of points ($N_{\text{PE}}^{\text{ns}}$) in the PE subdomain is similar for ice III ($\sim 280,000$) and ice Ih ($\sim 292,000$), the noticeably larger $N_{\text{PE}}^{\text{ns}}$ in ice II ($\sim 415,000$) originates from the underlying real-space grid assignment by the FFT algorithm in QE. Although the grid spacings along the lattice vectors are comparable among these three ice phases (due to the identical planewave cutoff), the lattice vectors in ice II (unlike III and Ih) do not correspond to the grid directions with minimal spacings; as such, the

Table 2. Computational Timing Profiles for NpT CPMD Simulations of Ice Ih , II , and III (Each Modeled by $(H_2O)_{96}$ at the Hybrid PBE0 Level on *Mira* IBM Blue Gene/Q, *Cori* Haswell, and *Cori* KNL Using the Extended **exx Module in QE^a**

lattice system(s)	orthorhombic	monoclinic	
	tetragonal cubic	hexagonal rhombohedral	triclinic
Example	ice III	ice Ih	ice II
Angular Constraints	$\begin{bmatrix} L_1 \perp L_2 \\ L_1 \perp L_3 \\ L_2 \perp L_3 \end{bmatrix}$	$\begin{bmatrix} L_1 \perp L_2 \\ L_2 \perp L_3 \end{bmatrix}$	[None]
N_{aux}	0	1	3
N_{pure}	3	4	6
N_{stcl}	19	25	37
N_{PE}^{ns} ^b	~280,000	~292,000	~415,000
$\langle N_{CG} \rangle^b$	121	89	126
$\mathcal{K}^b$	~3,100	~2,400	~3,200
$\langle N_{pair} \rangle$	7.0	4.3	6.5
<i>Mira</i> IBM Blue Gene/Q (s/step)			
$\langle t_{exx}^{stress} \rangle$	0.02	0.01	0.02
$\langle t_{exx}^{PE} \rangle$	2.00	1.30	3.40
$\langle t_{exx} \rangle$	5.92	3.83	10.25
$\langle t_{exx}^{PE} \rangle / \langle t_{exx} \rangle$	0.34	0.34	0.33
<i>Cori</i> Haswell (s/step)			
$\langle t_{exx}^{stress} \rangle$	0.01	0.01	0.02
$\langle t_{exx}^{PE} \rangle$	1.09	0.58	1.48
$\langle t_{exx} \rangle$	3.24	2.17	4.96
$\langle t_{exx}^{PE} \rangle / \langle t_{exx} \rangle$	0.34	0.27	0.30
<i>Cori</i> KNL ^c (s/step)			
$\langle t_{exx}^{stress} \rangle$	0.02	0.01	0.03
$\langle t_{exx}^{PE} \rangle$	1.59	1.01	2.96
$\langle t_{exx} \rangle$	4.85	3.89	7.88
$\langle t_{exx}^{PE} \rangle / \langle t_{exx} \rangle$	0.33	0.26	0.38

^aThese NpT simulations cover all the seven 3D lattice systems,^{123,126} which have been grouped into three different categories according to the number of auxiliary grid directions (N_{aux}) used in the NK Laplacian in eq 57; the listed angular constraints were applied throughout each simulation to maintain the targeted N_{aux} value. All timings (in s/step) were averaged over 50 CPMD steps and correspond to the mean wall times associated with computing the EXX contribution to the stress tensor ($\langle t_{exx}^{stress} \rangle$), solving the PE for all overlapping MLWF pairs ($\langle t_{exx}^{PE} \rangle$), and running through the entire **exx** module ($\langle t_{exx} \rangle$); also shown are the $\langle t_{exx}^{PE} \rangle / \langle t_{exx} \rangle$ ratios. Other relevant properties include the total number of pure derivatives ($N_{pure} = N_{aux} + 3$), the number of stencil points in the finite-difference representation of the NK Laplacian ($N_{stcl} = 2nN_{pure} + 1$, shown here for $n = 3$), the number of grid points in each $\Theta(C_{ij}R_{PE}^{ns})$ (Poisson) subdomain (N_{PE}^{ns} , shown here for non-self $\langle ij \rangle$ pairs only), the average number of CG iterations ($\langle N_{CG} \rangle$) required to solve each PE, the condition number ($\mathcal{K}$) of the sparse PE operator (i.e., $-\nabla^2$), and the average number of overlapping $\langle ij \rangle$ pairs assigned to each MPI process ($\langle N_{pair} \rangle$). All timings were obtained with $\zeta = 4$, $N_g = 4$, 1536 nodes (using one MPI process and all the available physical cores per node). ^bThe architecture dependence of the FFT algorithm leads to slight variations in N_{PE}^{ns} , $\langle N_{CG} \rangle$, and $\mathcal{K}$; *Cori* values are provided. ^cUsing `OMP_PROC_BIND = true` and `OMP_PLACES = cores`.

presence of the non-axial grid direction with minimal spacing (i.e., the grid-resolved trigonal axis, which is one of the auxiliary grid directions in the NK Laplacian identified using Algorithm 1) leads to a denser grid and hence the larger apparent N_{PE}^{ns} in ice II. Since $N_{stcl} \times N_{PE}^{ns}$ is the total number of floating-point operations required for computing the action of the Laplacian over the $\Theta(C_{ij}R_{PE}^{ns})$ subdomain (i.e., the left-hand side of eq 7), this quantity can be taken as a proxy for the computational cost per CG iteration when solving the PE. However, this quantity is not necessarily a robust sole predictor of the computational timings in **exx**; in fact, this measure would predict that NpT simulations of ice III would be similar (or slightly more efficient) to ice Ih and substantially more efficient than ice II, which is in contrast to the timings reported in Table 2.

To account for this discrepancy, two additional factors need to be taken into consideration, that is, the average number of CG iterations required to solve each PE ($\langle N_{CG} \rangle$) and the average number of overlapping MLWF pairs assigned to each MPI process ($\langle N_{pair} \rangle$). Since $\langle N_{CG} \rangle$ is largely governed by the condition number ($\mathcal{K}$), which is the ratio between the largest and smallest eigenvalues of the sparse NK Laplacian ($-\nabla^2$), we also provide $\mathcal{K}$ values in Table 2 corresponding to the first snapshot in each NpT simulation. Here, we find that the NK Laplacian is more well-conditioned for ice Ih ($\mathcal{K} \sim 2400$) than ice III (~ 3100) and ice II (~ 3200); as a result, the CG solution of the PE in ice Ih needed the least number of iterations ($\langle N_{CG} \rangle = 89$), while ice III and ice II had larger but similar $\langle N_{CG} \rangle$ values of 121 and 126, respectively. Quite interestingly, the NK Laplacian in the non-orthogonal ice Ih and ice II cases seems to be relatively well conditioned when compared to the orthogonal ice III case, despite the fact that N_{stcl} and N_{PE}^{ns} are significantly larger for both ice Ih and ice II. This finding highlights the strength of the NK approach (as well as our automated fluctuating-cell extension in Algorithm 1) when treating systems with non-orthogonal simulation cells, as the selection of auxiliary directions is a non-trivial procedure that can lead to severe numerical instabilities if done incorrectly. Taking ice II as an example, choosing the grid-resolved obtuse-angle bisector for each pair of lattice vectors as the three auxiliary directions (i.e., a naive 3D generalization of the non-orthogonal 2D NK procedure outlined in eqs 46 and 47 and depicted in Figure 2) leads to a Laplacian that is no longer negative semi-definite; as a result, the CG solution to the PE requires an excessively large number of iterations if and when it converges.

Since $\langle N_{pair} \rangle$ is roughly proportional to the total number of overlapping MLWF pairs in the system (which is determined by the $|\tilde{C}_i - \tilde{C}_j| < R_{pair}$ criterion), $\langle N_{pair} \rangle$ for the lower-density ice Ih phase ($\langle N_{pair} \rangle = 4.3$) is significantly less than that found in the higher-density ice III (7.0) and ice II (6.5) phases. With this information in hand, it is now clear why **exx**-based NpT simulations of ice Ih have the lowest $\langle t_{exx}^{PE} \rangle$ among the ice phases. Although ice Ih has intermediate values for N_{stcl} and N_{PE}^{ns} (and hence an intermediary computational cost per CG iteration), this ice phase has the lowest $\langle N_{pair} \rangle$ (due to its relatively lower density) and the lowest $\langle N_{CG} \rangle$ (due to its relatively lower $\mathcal{K}$ value); as such, each CPMD step will require CG solutions to the least number of PEs and the solution to each PE requires the least number of CG iterations. To explain why $\langle t_{exx}^{PE} \rangle$ for ice II is larger than ice III (in which both $\langle N_{CG} \rangle$ and N_{pair} are similar), we again reiterate that ice II has the largest N_{stcl} and N_{PE}^{ns} values and therefore requires the

largest number of floating-point operations per CG step. Although a more detailed analysis of the communication and processor idling would be required to fully explain the total exx timings during these NpT simulations, we can still justify the $\langle t_{\text{exx}} \rangle$ ordering among these ice phases by noting that (i) the $\langle t_{\text{exx}}^{\text{PE}} \rangle / \langle t_{\text{exx}} \rangle$ ratio is $\approx 30\%$ for all three ice phases and (ii) the communication overhead is roughly proportional to $N_{\text{ME}}^{\text{ns}}$ (which is proportional to $N_{\text{PE}}^{\text{ns}}$). Since ice II has the largest values for $\langle t_{\text{exx}}^{\text{PE}} \rangle$ and $N_{\text{PE}}^{\text{ns}}$, both computation and communication costs will be largest for this ice phase; with computation and communication comprising a majority of $\langle t_{\text{exx}} \rangle$, the increased wall times observed across all the three HPC architectures are not only reasonable but expected for NpT simulations of this higher-density (and non-orthogonal) ice polymorph.

From this discussion, it is clear that N_{aux} (or $N_{\text{pure}} = N_{\text{aux}} + 3$) governs N_{stcl} and hence modulates (in conjunction with $N_{\text{PE}}^{\text{ns}}$) the number of floating-point operations during each step in the iterative CG solution to the PE. In the ice II case presented above, we intentionally performed the NpT simulation without angular constraints to showcase a triclinic lattice with $N_{\text{aux}} = 3$ (or $N_{\text{pure}} = 6$), thereby allowing for non-constrained microscopic cell fluctuations in ice II. In doing so, the computational cost of this simulation was $\approx 50\%$ higher than the one in which ice II would be constrained to maintain rhombohedral symmetry with $N_{\text{aux}} = 1$ (or $N_{\text{pure}} = 4$), i.e., the naturally occurring and macroscopically observed lattice symmetry for this ice phase. In addition to the application of angular constraints to change N_{stcl} (via $N_{\text{stcl}} = 2nN_{\text{pure}} + 1 = 2n(N_{\text{aux}} + 3) + 1$), alternative cell choices may also be used to control the size/extent (and hence computational complexity) of the NK Laplacian. For instance, a hexagonal (or rhombohedral) lattice with $N_{\text{pure}} = 4$ can be transformed into an orthorhombic lattice with $N_{\text{pure}} = 3$; for the well-known hexagonal case (with $L_1 \perp L_2$), one can construct an orthorhombic (super-)cell with lattice vectors $\{L'_1, L'_2, L'_3\}$ such that $L'_1 = L_1$, $L'_2 = 2L_2 + L_1$, and $L'_3 = L_3$. However, this reduction in N_{pure} (and hence N_{stcl}) is accompanied by the increased complexity of dealing with a simulation cell containing twice as many atoms; while such an increase in system size may be cumbersome for AIMD simulations, the additional degrees of freedom can also prove useful when describing the proton disorder in a system like ice Ih.

4.3. Parallel Scaling and Performance. Having discussed the computational complexity associated with different lattice symmetries, we now move on to assess the performance and parallel scaling of the extended exx module when applied to large-scale NpT simulations of liquid water. In close analogy to the critical assessment of exx during NVT simulations of liquid water in Section IV.B of PAPER-I,⁹⁸ this section will focus on the internode (MPI) parallelization level via a strong scaling analysis (in which the number of processing elements is varied for a fixed problem size) and a weak scaling analysis (in which the problem size is varied for a fixed ratio of problem size to the number of processing elements). We will also briefly discuss the intranode (OpenMP) parallelization level (which is particularly relevant for NpT simulations using relatively large planewave basis sets) as well as the general performance of exx across several different HPC architectures (e.g., *Mira* IBM Blue Gene/Q, *Cori* Haswell, and *Cori* KNL).

Unless otherwise specified, the computational timings for each of the following liquid water simulations were obtained using the same planewave/pseudopotential/CPMD settings

and exx parameters as those employed above for the ice Ih, II, and III phases in Section 4.2. In contrast with the ice simulations (in which the system size was fixed at $(\text{H}_2\text{O})_{96}$, $T = 238$ K, $p = 2.1$ kBar, and $\zeta = 4$), we follow the same profiling procedure given in Section IV.B of PAPER-I⁹⁸ by performing a series of 12 different EXX -based CPMD simulations of liquid water at $T = 300$ K and $p = 1.0$ bar, in which (i) the system size was varied to include $N_{\text{water}} = 64, 128, 256$ water molecules (each of which has $N_o = 4 \times N_{\text{water}}$ MLWFs) and (ii) the number of processing elements (N_{proc} MPI processes) was varied by changing $\zeta = N_{\text{proc}}/N_o$. Initial snapshots for each liquid water system were prepared following the equilibration procedure detailed in PAPER-I;⁹⁸ in the NpT simulations performed in this work, all instantaneous cell fluctuations were constrained to maintain simple cubic symmetry (i.e., $L_1 \perp L_2$, $L_1 \perp L_3$, $L_2 \perp L_3$, and $|L_1| = |L_2| = |L_3|$) with $N_{\text{aux}} = 0$ and $N_{\text{pure}} = 3$. Strong and weak scaling tests were performed on *Mira* IBM Blue Gene/Q (using $\zeta \in \{1/2, 1, 2, 4\}$ and $N_{\text{water}} \in \{64, 128, 256\}$) with an additional assessment of the extended exx module on *Cori* Haswell and *Cori* KNL (using $\zeta = 1$ and $N_{\text{water}} = 128$). In each case, we again use one process per node for the internode MPI parallelization and all the available physical cores per node for the intranode OpenMP parallelization (with the hyperthreading settings described in Section 4.2); following the discussion in Section IV.B of PAPER-I,⁹⁸ the highest possible task group parallelization level was employed (via $N_{\text{tg}} \in \{1, 2, 4, 8\}$) for the 3D FFT operations in the non- exx portions of QE.

When compared to the previous strong scaling tests of exx on liquid water in the NVT ensemble (with a fixed simulation cell and real-space grid compatible with an 85 Ry planewave cutoff, see Figure 8 in PAPER-I⁹⁸), we again observe similar MPI performance for the extended exx module in the NpT ensemble (with a fluctuating simulation cell and real-space grid compatible with the significantly larger 150 Ry planewave cutoff, see Figure 4). For a given (and fixed) system size, we follow PAPER-I⁹⁸ and define the strong scaling efficiency of exx with respect to a reference ζ value (i.e., $\zeta_{\text{ref}} = 1/2$, a commonly used setting for AIMD simulations of liquid water) as

$$\eta_{\text{MPI}}^{\text{strong}}(\zeta) \equiv \frac{\zeta_{\text{ref}} \cdot \langle t_{\text{exx}} \rangle_{\zeta_{\text{ref}}}}{\zeta \cdot \langle t_{\text{exx}} \rangle_{\zeta}} = \frac{\frac{1}{2} \cdot \langle t_{\text{exx}} \rangle_{\zeta=1/2}}{\zeta \cdot \langle t_{\text{exx}} \rangle_{\zeta}} \quad (61)$$

in which $\langle t_{\text{exx}} \rangle_{\zeta}$ is the wall time spent in exx when using a specific ζ value. For $\zeta > 1/2$, we find that $\eta_{\text{MPI}}^{\text{strong}}$ (when averaged over $(\text{H}_2\text{O})_{64}$, $(\text{H}_2\text{O})_{128}$, and $(\text{H}_2\text{O})_{256}$) decreases to $\approx 93\%$ ($\zeta = 1$), $\approx 67\%$ ($\zeta = 2$), and $\approx 52\%$ ($\zeta = 4$). Quite interestingly, the strong scaling performance of exx in the more demanding NpT ensemble is nearly identical to that observed for the same systems in the NVT ensemble (see Figure 8 and the surrounding discussion in PAPER-I⁹⁸), where we reported $\eta_{\text{MPI}}^{\text{strong}}$ values of $\approx 93\%$ ($\zeta = 1$), $\approx 66\%$ ($\zeta = 2$), and $\approx 50\%$ ($\zeta = 4$). In general, the exx module is more efficient for smaller ζ values (i.e., $\zeta \leq 1$) since the use of massively parallel HPC resources ($\zeta \gg 1$) is intrinsically more susceptible to processor idling (due to the larger computational workload imbalance associated with more MPI processes) and also requires additional/duplicate MLWF communication across the larger pool of MPI processes. See below for a more detailed breakdown of $\langle t_{\text{exx}} \rangle$ into computation, communication, and processor idling, as well as a discussion on how these components influence the strong scaling efficiency of exx .

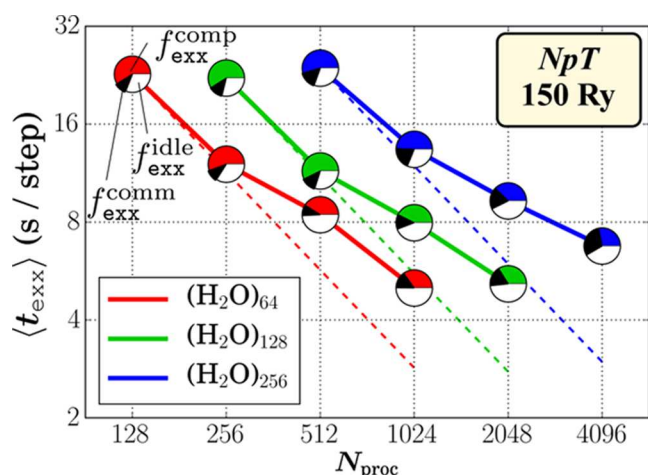


Figure 4. Strong scaling analysis of the extended exx module in QE during NpT CPMD simulations of liquid water at the hybrid PBE0 level on Mira IBM Blue Gene/Q. For a fixed system size [$N_{\text{water}} = 64$ (red line), 128 (green line), 256 (blue line)], the mean wall times (averaged over 50 CPMD steps) spent in the exx module ($\langle t_{\text{exx}} \rangle$ in s/step) are plotted against the number of MPI processes (N_{proc} varied via $\zeta \equiv N_{\text{proc}}/N_0 \in \{1, 2, 1, 2, 4\}$). For comparison, ideal strong scaling wall times (dashed lines) were computed with respect to the $\zeta_{\text{ref}} = 1/2$ case (see eq 61). The inset pie charts also depict the fraction/percent of $\langle t_{\text{exx}} \rangle$ dedicated to computation ($f_{\text{exx}}^{\text{comp}}$, colored), communication ($f_{\text{exx}}^{\text{comm}}$, black), and processor idling ($f_{\text{exx}}^{\text{idle}}$, white).

When compared to the previous weak scaling tests of exx (on liquid water in the NVT ensemble, see Figure 9 in PAPER-I⁹⁸), however, we observe a substantial improvement in the MPI performance of the extended exx module during large-scale NpT simulations (see Figure 5). For a given (and fixed) ζ

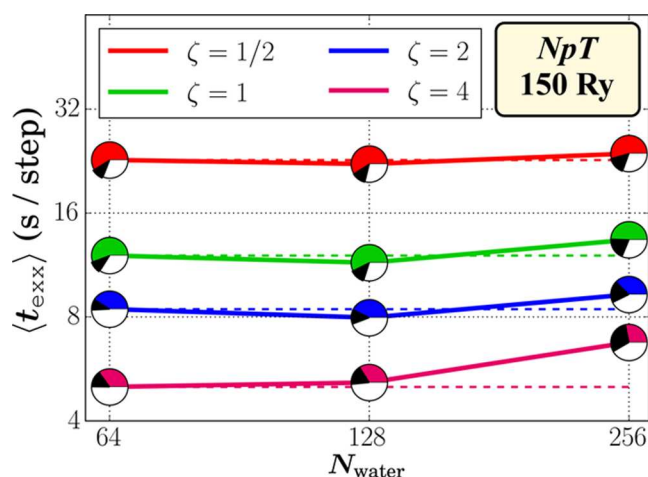


Figure 5. Weak scaling analysis of the extended exx module in QE during NpT CPMD simulations of liquid water at the hybrid PBE0 level on Mira IBM Blue Gene/Q. For a fixed ratio of system size to the number of processing elements [$\zeta = 1/2$ (red line), 1 (green line), 2 (blue line), and 4 (magenta line)], the mean wall times (averaged over 50 CPMD steps) spent in the exx module ($\langle t_{\text{exx}} \rangle$ in s/step) are plotted against the system size (N_{water} varied to include $(H_2O)_{64}$, $(H_2O)_{128}$, and $(H_2O)_{256}$). For comparison, the ideal weak scaling wall times (dashed lines) correspond to linear (or $O(N)$) scaling and were computed with respect to the $N_{\text{water}} = 64$ case (see eq 62). The inset pie charts again depict the fraction/percent of $\langle t_{\text{exx}} \rangle$ dedicated to computation ($f_{\text{exx}}^{\text{comp}}$, colored), communication ($f_{\text{exx}}^{\text{comm}}$, black), and processor idling ($f_{\text{exx}}^{\text{idle}}$, white).

value, we again follow PAPER-I⁹⁸ and define the weak scaling efficiency of exx with respect to a reference system size (i.e., $N_{\text{water}}^{\text{ref}} = 64$, a commonly used system size for AIMD simulations of liquid water) as

$$\eta_{\text{MPI}}^{\text{weak}}(N_{\text{water}}) \equiv \frac{\langle t_{\text{exx}} \rangle_{N_{\text{water}}^{\text{ref}}}}{\langle t_{\text{exx}} \rangle_{N_{\text{water}}}} = \frac{\langle t_{\text{exx}} \rangle_{N_{\text{water}}=64}}{\langle t_{\text{exx}} \rangle_{N_{\text{water}}}} \quad (62)$$

in which $\langle t_{\text{exx}} \rangle_{N_{\text{water}}}$ is the wall time spent in exx for a specific N_{water} . For $N_{\text{water}} > 64$, we find that $\eta_{\text{MPI}}^{\text{weak}}$ (when averaged over $\zeta \in \{1/2, 1, 2, 4\}$) first slightly increases to $\approx 103\%$ ($N_{\text{water}} = 128$) and then decreases to $\approx 89\%$ ($N_{\text{water}} = 256$). These weak scaling efficiencies are marked improvements over the NVT values of $\approx 89\%$ ($N_{\text{water}} = 128$) and $\approx 81\%$ ($N_{\text{water}} = 256$) reported in Figure 9 (and the surrounding discussion) in PAPER-I⁹⁸ and demonstrate that the extended exx module is exhibiting close to linear (or $O(N)$) scaling behavior in the $(H_2O)_{64}$ – $(H_2O)_{256}$ system size regime in the more demanding NpT ensemble. Here, we note in passing that the observed $\eta_{\text{MPI}}^{\text{weak}}$ value exceeding 100% for $(H_2O)_{128}$ is merely an artifact of choosing $(H_2O)_{64}$ as the reference system size as well as averaging over all the four ζ values; as such, we interpret this result as a simple indication that exx is scaling nearly ideally when the system is doubled from $(H_2O)_{64}$ to $(H_2O)_{128}$. When using a relatively low amount of computational resources (e.g., $\zeta = 1/2, 1$), we find that the weak scaling behavior of exx is quite close to linear scaling in the $(H_2O)_{64}$ – $(H_2O)_{256}$ system size regime (Figure 5). However, the scalability starts to degrade for $(H_2O)_{256}$ at the $\zeta \geq 2$ level, which we largely attribute to (i) increased communication due to the underlying ALL-TO-ALL MPI operations in the data redistribution steps (Steps I and VI in Figure 1) and (ii) increased processor idling due to the inherent difficulty with balancing the workload across a larger number of MPI processes (see below and Section IV.B in PAPER-I⁹⁸). Furthermore, we also note that the weak scaling efficiency of exx (in both the NVT and NpT ensembles) is significantly better than its strong scaling efficiency; however, this result is not surprising as it is (in general) more efficient to distribute the additional workload associated with an increased system size over a larger number of processing units rather than use the increased processing resources to reduce the overall time to solution for a fixed system size.

Here, we remind the reader that the exx module only represents one portion of an overall hybrid DFT calculation: input into exx is the current set of MLWFs at a given CPMD step; output from exx is $E_{\text{xx}}\{\tilde{D}_{\text{xx}}^i(\mathbf{r})\}$, and σ_{xx} . As such, several other modules in QE (some of which are not necessarily linear scaling) are required to perform the remaining non- exx tasks (i.e., all other GGA–DFT operations as well as MLWF localization) and will ultimately dominate the overall scalability of a hybrid DFT calculation. For instance, the cost associated with MLWF localization, which contains some cubic scaling matrix operations, can become more substantial for larger system sizes (e.g., ≈ 10 – 20% of the total wall time for $(H_2O)_{256}$); see Table 1 in PAPER-I⁹⁸ and the surrounding text for a more detailed discussion. As such, incorporating the exx module into an overall linear scaling GGA code—in conjunction with a more efficient on-the-fly orbital localization procedure—could be a viable strategy for achieving a fully (overall) linear scaling hybrid DFT approach.

For the largest systems considered in this work (i.e., $(H_2O)_{128}$ and $(H_2O)_{256}$), the extended exx module can

Table 3. Computational Timing Profiles for NpT CPMD Simulations of Liquid Water at the Hybrid PBE0 Level on *Mira* IBM Blue Gene/Q, *Cori* Haswell, and *Cori* KNL Using the Extended Exx Module in QE^a

architecture	QE module timings					breakdown of $\langle t_{\text{exx}} \rangle$					
	$\langle t_{\text{GGA}} \rangle$	$\langle t_{\text{MLWF}} \rangle$	$\langle t_{\text{exx}} \rangle$	$\langle t_{\text{total}} \rangle$	$\langle t_{\text{exx}} \rangle / \langle t_{\text{GGA}} \rangle$	$\langle t_{\text{exx}}^{\text{comp}} \rangle$	$\langle f_{\text{exx}}^{\text{comp}} \rangle$	$\langle t_{\text{exx}}^{\text{comm}} \rangle$	$\langle f_{\text{exx}}^{\text{comm}} \rangle$	$\langle t_{\text{exx}}^{\text{idle}} \rangle$	$\langle f_{\text{exx}}^{\text{idle}} \rangle$
<i>Mira</i> IBM Blue Gene/Q	2.79	0.59	11.49	14.87	4.1	6.58	(57.3)	1.47	(12.8)	3.43	(29.9)
<i>Cori</i> Haswell	1.16	1.26	5.37	7.79	4.6	2.96	(55.1)	0.77	(14.3)	1.65	(30.6)
<i>Cori</i> KNL (no hyperthreading ^b)	12.25	3.71	9.16	25.12	0.7	4.85	(52.9)	2.18	(23.8)	2.13	(23.3)
<i>Cori</i> KNL (no hyperthreading ^c)	5.10	1.98	8.20	15.28	1.6	4.61	(56.2)	1.60	(19.5)	1.99	(24.3)

^aAll timings (in s/step) were averaged over 50 CPMD steps and correspond to the mean wall times associated with completing all GGA (non-exx) contributions to the simulation ($\langle t_{\text{GGA}} \rangle$), optimizing the Marzari–Vanderbilt functional ($\langle t_{\text{MLWF}} \rangle$), needed to re-localize the MLWFs between each CPMD step), running through the entire exx module ($\langle t_{\text{exx}} \rangle$, i.e., Steps I–VI in Figure 1), as well as performing a given CPMD step ($\langle t_{\text{total}} \rangle$).¹⁴¹ Also included are the $\langle t_{\text{exx}} \rangle / \langle t_{\text{GGA}} \rangle$ ratios, as well as a breakdown of $\langle t_{\text{exx}} \rangle$ into the following components: computation ($\langle t_{\text{exx}}^{\text{comp}} \rangle$), communication ($\langle t_{\text{exx}}^{\text{comm}} \rangle$), and processor idling ($\langle t_{\text{exx}}^{\text{idle}} \rangle$); for convenience, the fraction/percent of $\langle t_{\text{exx}} \rangle$ dedicated to each of these components (i.e., $f_{\text{exx}}^{\text{comp}}$, $f_{\text{exx}}^{\text{comm}}$, and $f_{\text{exx}}^{\text{idle}}$) are reported as percentages of $\langle t_{\text{exx}} \rangle$. All timings were obtained during NpT simulations of $(\text{H}_2\text{O})_{128}$ with $\zeta = 1$, $N_{\text{tg}} = 2$, and 512 nodes on each architecture (using one MPI process and all the available physical cores per node). Hyperthreading was fully activated on each physical core, except for *Cori* KNL, where hyperthreading was disabled to prevent performance degradation (cf. Table 2 in PAPER-I⁹⁸). See text for more details. ^bUsing default OpenMP settings (i.e., the same settings used in PAPER-I⁹⁸). ^cUsing OMP_PROC_BIND = true and OMP_PLACES = cores.

evaluate all EXX-related quantities required to propagate the constant-pressure CPMD equations of motion in eqs 10–12 in ≈ 5.2 s/step for $(\text{H}_2\text{O})_{128}$ and ≈ 6.8 s/step for $(\text{H}_2\text{O})_{256}$ using massively parallel HPC resources (i.e., $\zeta = 4$) on the *Mira* IBM Blue Gene/Q platform. When compared to NVT simulations of liquid water using exx and the same computational resources (cf. ≈ 2.0 s/step for $(\text{H}_2\text{O})_{128}$ and 2.4 s/step for $(\text{H}_2\text{O})_{256}$, see Table 1 of PAPER-I⁹⁸), the increased wall times observed here mainly originate from the larger planewave cutoff (cf. 150 Ry for NpT vs 85 Ry for NVT) and hence the larger number of points in the real-space grid (*vide infra*). In practice, 50 ps NpT simulations of large systems such as $(\text{H}_2\text{O})_{128}$ and $(\text{H}_2\text{O})_{256}$ would therefore require ≈ 1.0 – 1.3 months using similar HPC resources and a more conventional CPMD time step of 0.10 fs. As such, the extended exx module enables very challenging large-scale NpT simulations for extended length scales at the hybrid DFT level of theory.

Similar to PAPER-I,⁹⁸ we further investigate the exx wall times by breaking $\langle t_{\text{exx}} \rangle$ into the following contributions: computation events ($\langle t_{\text{exx}}^{\text{comp}} \rangle$), communication overhead ($\langle t_{\text{exx}}^{\text{comm}} \rangle$), and processor idling due to workload imbalance ($\langle t_{\text{exx}}^{\text{idle}} \rangle$). For convenience, the fraction/percent of $\langle t_{\text{exx}} \rangle$ dedicated to each of these components (i.e., $f_{\text{exx}}^{\text{comp}}$, $f_{\text{exx}}^{\text{comm}}$, and $f_{\text{exx}}^{\text{idle}}$) are depicted as pie charts in Figures 4 and 5. For the 12 NpT simulations performed in this work, we find that all three of these components are larger in magnitude than in the NVT case and still represent sizable contributions to $\langle t_{\text{exx}} \rangle$. As mentioned above, the increased wall times reported herein are a direct consequence of the larger planewave cutoffs employed during constant-pressure NpT simulations; by increasing the cutoff from 85 Ry (NVT) to 150 Ry (NpT), the density of real-space grid points in Ω (as well as $\Theta(C_{ij}R_{\text{PE}})$ and $\Theta(C_{ij}R_{\text{ME}})$) increases by a factor of 2.5–2.7 $\times$. For computational cost, the larger N_{PE} increases the number of steps (as well as the computational complexity per step) during the iterative CG solution to the PE (see Section 4.2), while the larger N_{ME} increases the cost of the ME. For the communication overhead, the larger grid density requires sending/receiving larger chunks of data during the forward/backward redistribution (e.g., Steps I and VI in Figure 1, to maintain compatibility with QE) as well as the internal communication needed to compute each $\langle ij \rangle$ contribution to the energy, wavefunction forces, and stress tensor (e.g., Steps III–V). With

an increased computational cost per overlapping MLWF pair, the larger N_{PE} and N_{ME} also lead to more extended processor idling times due to the intrinsic imperfect distribution of $\langle ij \rangle$ pairs across MPI processes (see Sections III.C.2 and IV.B.1 in PAPER-I⁹⁸). Cell fluctuations during NpT simulations further impact the processor idling in exx by introducing larger variability in the time to solution for each PE (due primarily to variable-quality guesses based on previous CPMD steps) as well as additional imbalance in the computational workload (due to the more diverse local environments sampled by each MLWF).

For small ζ values ($\zeta \leq 1$), we find that exx is technically computation bound, with $f_{\text{exx}}^{\text{comp}} \approx 56\%$, $f_{\text{exx}}^{\text{comm}} \approx 13\%$, and $f_{\text{exx}}^{\text{idle}} \approx 31\%$ (when averaged over $\zeta = 1/2$ and $\zeta = 1$ for $(\text{H}_2\text{O})_{64}$, $(\text{H}_2\text{O})_{128}$, and $(\text{H}_2\text{O})_{256}$), although the wall time associated with communication overhead and processor idling ($\approx 44\%$) still remains substantial. With HPC resources ($\zeta \gg 1$), the balance among computation and processor idling is now switched, with $f_{\text{exx}}^{\text{comp}} \approx 36\%$, $f_{\text{exx}}^{\text{comm}} \approx 18\%$, and $f_{\text{exx}}^{\text{idle}} \approx 46\%$ (when averaged over $\zeta = 2$ and $\zeta = 4$ for $(\text{H}_2\text{O})_{64}$, $(\text{H}_2\text{O})_{128}$, and $(\text{H}_2\text{O})_{256}$), but the combined computation and communication cost ($\approx 54\%$) is technically dominant. In this limit, we have previously observed a roughly equal distribution of $f_{\text{exx}}^{\text{comm}} \approx f_{\text{exx}}^{\text{comp}} \approx f_{\text{exx}}^{\text{idle}} \approx 33\%$ during large-scale NVT simulations of liquid water (i.e., $(\text{H}_2\text{O})_{256}$ with $\zeta = 4$, see Table 1 and Figures 8 and 9 in Section IV.B.1 of PAPER-I⁹⁸); in the more challenging NpT case investigated here, the role of processor idling has become even more prominent in determining the overall time to solution, while the (albeit reduced) relative contributions from computation and communication are still considerable. As such, we are in the process of developing a comprehensive three-pronged theoretical and algorithmic approach (i.e., the β version of exx) that specifically addresses each of these sizable contributions to $\langle t_{\text{exx}} \rangle$ and will enable hybrid DFT-based NpT simulations of even larger systems and longer durations.

We complete this section with a brief discussion on intranode OpenMP parallelization efficiency as well as the overall performance of exx when performing large-scale NpT simulations on different HPC architectures. Regarding the OpenMP strong scaling efficiency, we point the reader to Figure 10 (as well as the surrounding text in Section IV.B.2) in PAPER-I,⁹⁸ where we specifically investigated the performance

of `exx` during Step IV (the computational bottleneck of `exx`) using two different planewave cutoffs: 85 and 150 Ry (to mimic the typical settings employed during *NVT* and *NpT* simulations).¹⁴² When performing these simulations, we found that `exx` maintains high strong scaling efficiencies with $\eta_{\text{openMP}}^{\text{strong}}$ values (see eq 40 in PAPER-I⁹⁸) of $\approx 84\%$ (85 Ry) and $\approx 92\%$ (150 Ry) as the number of OpenMP threads was increased from 1 (single-thread limit) to 16 (complete activation of all physical cores) per *Mira* IBM Blue Gene/Q node (with a further 30–40% boost when all the 64 hyperthreads were activated). Since the computational workload assigned to each thread increases with the planewave cutoff, the OpenMP efficiency of `exx` generally increases during large cutoff (*NVT* or *NpT*) simulations; as such, we expect that `exx` will also benefit from the use of advanced vectorization techniques as well as offloading to graphics processing units.

As a final assessment of the extended `exx` module, we repeated the $\zeta = 1$ *NpT* CPMD simulations of $(\text{H}_2\text{O})_{128}$ on the *Cori* Haswell and *Cori* KNL supercomputer architectures located at the National Energy Research Scientific Computing Center (NERSC). In analogy to the *NVT* timing profiles provided in Table 2 of PAPER-I,⁹⁸ Table 3 shows that there exists some variability in the individual QE module timings across all the three architectures, with $\langle t_{\text{GGA}} \rangle$ ranging from ≈ 1.2 s/step (Haswell) to ≈ 12.3 s/step (KNL) and $\langle t_{\text{exx}} \rangle$ ranging from ≈ 5.4 s/step (Haswell) to ≈ 11.5 s/step (IBM Blue Gene/Q). With $\langle t_{\text{exx}} \rangle / \langle t_{\text{GGA}} \rangle = 0.7\text{--}4.6$, we again observe that the extended `exx` module requires a wall time cost that is comparable to semi-local DFT and therefore enables large-scale constant-pressure AIMD simulations at the hybrid DFT level. Here, we note in passing that the performance of `exx` (as well as the non-`exx` portions of QE) on *Cori* KNL is quite sensitive to the OpenMP settings as well as the use of hyperthreading. For instance, refining the default OpenMP settings on *Cori* KNL (by specifying `OMP_PROC_BIND = true` and `OMP_PLACES = cores`) leads to an $\approx 60\%$ reduction in $\langle t_{\text{GGA}} \rangle$ from 12.3 s/step to 5.1 s/step, accompanied by a more modest (but still noticeable) $\approx 10\%$ reduction in $\langle t_{\text{exx}} \rangle$ from 9.2 to 8.2 s/step. In fact, a 150 Ry *NpT* simulation of $(\text{H}_2\text{O})_{128}$ on *Cori* KNL using these refined settings ($\langle t_{\text{GGA}} \rangle = 5.1$ s/step) can actually be performed faster than an 85 Ry *NVT* simulation of the same system using the default OpenMP settings ($\langle t_{\text{GGA}} \rangle = 5.4$ s/step, see Table 2 of PAPER-I⁹⁸). Within the `exx` module, we find that the breakdown of $\langle t_{\text{exx}} \rangle$ into computation, communication, and processor idling is very similar across these three HPC architectures and quite consistent with that reported in Table 2 of PAPER-I⁹⁸ for the analogous *NVT* case. In this system size and ζ regime, `exx` is technically computation bound ($f_{\text{exx}}^{\text{comp}} = 55.4 \pm 1.9\%$), with communication ($f_{\text{exx}}^{\text{comm}} = 17.6 \pm 5.0\%$) and processor idling ($f_{\text{exx}}^{\text{idle}} = 27.0 \pm 3.8\%$) accounting for the rest of the time spent in the `exx` module. With sizable contributions from all three components, this observation once again reiterates the need for a comprehensive three-pronged strategy in the next-generation `exx` codebase. Hence, the combination of the current (and next-generation) `exx` codebase—along with an overall linear scaling GGA implementation and a more efficient on-the-fly orbital localization scheme—could be a viable route toward a fully linear scaling hybrid DFT approach.

5. CONCLUSIONS AND FUTURE OUTLOOK

In this work, we present several theoretical and algorithmic developments to our linear scaling and real-space MLWF-based EXX approach⁹⁸ that enable constant-pressure CPMD simulations (in the *NpH* and/or *NpT* ensembles) of large-scale finite-gap condensed-phase systems in general/non-orthogonal cells at the hybrid DFT level. For the theoretical extension to this approach, we derived an analytical expression for the EXX contribution to the stress tensor for systems with general and fluctuating simulation cells with a computational complexity that scales linearly with system size. When used in conjunction with the previously developed theoretical approaches for obtaining the EXX contribution to the energy and wavefunction forces,⁹⁸ this work provides the remaining ingredient needed for propagating the CPMD equations of motion under constant-pressure conditions and hence an overall order-*N* method for performing large-scale hybrid DFT-based CPMD simulations in the *NVE/NVT* as well as *NpH/NpT* ensembles. For the algorithmic extension to this approach, we have incorporated a number of new routines into the `exx` module in Quantum ESPRESSO (QE) that have been optimized to (i) provide generalized subdomains that handle both static and fluctuating simulation cells with non-orthogonal lattice symmetries, (ii) solve PE in general/non-orthogonal cells via an automated selection of the auxiliary grid directions in the Natan–Kronik (NK) representation of the discrete Laplacian operator, and (iii) evaluate the EXX contribution to the stress tensor using the analytical expression derived in this work.

This was followed by a case study demonstrating that one can use `exx`—with an appropriate choice of parameters—to tightly and simultaneously converge the EXX contributions to the energy and stress tensor for a wide variety of condensed-phase systems (including liquid $(\text{H}_2\text{O})_{128}$, the monoclinic benzene-II polymorph, and the semi-conducting Si_{216} crystal). We also provided a critical assessment of the computational performance of the extended massively parallel hybrid MPI/OpenMP-based `exx` module across several different HPC architectures (e.g., *Mira* IBM Blue Gene/Q, *Cori* Haswell, and *Cori* KNL) via detailed case studies on (i) the computational complexity due to lattice symmetry during short *NpT* simulations of the ice Ih, II, and III polymorphs at their corresponding triple point and (ii) the strong and weak scaling of `exx` during large-scale *NpT* simulations of ambient liquid water ranging from $(\text{H}_2\text{O})_{64}$ to $(\text{H}_2\text{O})_{256}$. In doing so, we found that evaluation of the EXX contribution to the stress tensor required negligible ($<1\%$) computational overhead for all systems tested, thereby providing a simultaneously more accurate and more computationally efficient approach than direct numerical differentiation of E_{xx} with respect to *h*. We also demonstrate that the extended `exx` module remains quite robust and highly scalable when performing challenging *NpT* simulations of liquid water (with a very tight 150 Ry planewave cutoff); here, we found that the MPI strong scaling behavior remains essentially the same as that observed during 85 Ry *NVT* simulations in PAPER-I,⁹⁸ while the MPI weak scaling efficiency of `exx` becomes noticeably improved. With these theoretical and algorithmic advances, the extended `exx` module brings us another step closer to routinely performing high-fidelity hybrid DFT-based AIMD simulations of sufficient duration for complex and large-scale condensed-phase systems across a wide range of thermodynamic conditions.

Moving forward, our group is in the process of further improving the strong and weak scaling efficiencies of `exx` by implementing a comprehensive three-pronged strategy that simultaneously attacks the remaining contributions from computation, communication, and processor idling to the wall time cost. Our group is also actively working on a variable subdomain generalization of the `exx` module for an accurate and computationally efficient treatment of EXX in heterogeneous systems with multiple phases and/or components, which is needed for the study of physical processes and chemical reactions in diverse environments and complex interfaces. Other future research directions include optimizing `exx` for performing high-throughput calculations needed for machine-learning intra-/inter-molecular potentials of condensed-phase systems, as well as extending `exx` to sample other statistical ensembles (i.e., μVT) needed for simulating even larger swaths of experimental conditions at the hybrid DFT level.

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; orcid.org/0000-0003-1619-6514

Biswajit Santra – Department of Physics, Temple University, Philadelphia, Pennsylvania 19122, United States; orcid.org/0000-0003-3609-2106

Complete contact information is available at:
<https://pubs.acs.org/10.1021/acs.jctc.0c01194>

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

All authors thank Roberto Car, Amir Natan, Tatsuhiro Onodera, and Leeor Kronik for helpful scientific discussions. This material is based on the work supported by the National Science Foundation under grant no. CHE-1945676. R.A.D. also gratefully acknowledges the financial support from an Alfred P. Sloan Research Fellowship. This research used resources of the National Energy Research Scientific Computing 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.

REFERENCES

- (1) Frenkel, D.; Smit, B. *Understanding Molecular Simulation: from Algorithms to Applications*; Academic Press: New York, 2001.
- (2) Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids*; Clarendon Press: Oxford, 1989.
- (3) Finocchi, F.; Galli, G.; Parrinello, M.; Bertoni, C. M. Microscopic Structure of Amorphous Covalent Alloys Probed by Ab Initio Molecular Dynamics: SiC. *Phys. Rev. Lett.* **1992**, *68*, 3044–3047.
- (4) Johnson, R. R.; Johnson, A. T. C.; Klein, M. L. Probing the Structure of DNA-Carbon Nanotube Hybrids with Molecular Dynamics. *Nano Lett.* **2008**, *8*, 69–75.
- (5) Trabuco, L. G.; Villa, E.; Mitra, K.; Frank, J.; Schulten, K. Flexible Fitting of Atomic Structures into Electron Microscopy Maps Using Molecular Dynamics. *Structure* **2008**, *16*, 673–683.
- (6) Zhao, G.; Perilla, J. R.; Yufenyuy, E. L.; Meng, X.; Chen, B.; Ning, J.; Ahn, J.; Gronenborn, A. M.; Schulten, K.; Aiken, C.; Zhang, P. Mature HIV-1 Capsid Structure by Cryo-Electron Microscopy and All-Atom Molecular Dynamics. *Nature* **2013**, *497*, 643–646.
- (7) Martelli, F.; Ko, H.-Y.; Oğuz, E. C.; Car, R. Local-Order Metric for Condensed-Phase Environments. *Phys. Rev. B* **2018**, *97*, 064105.
- (8) Bergsma, J. P.; Gertner, B. J.; Wilson, K. R.; Hynes, J. T. Molecular Dynamics of a Model S_N2 Reaction in Water. *J. Chem. Phys.* **1987**, *86*, 1356–1376.
- (9) van Gunsteren, W. F.; Berendsen, H. J. C. Computer Simulation of Molecular Dynamics: Methodology, Applications, and Perspectives in Chemistry. *Angew. Chem., Int. Ed.* **1990**, *29*, 992–1023.
- (10) Craig, I. R.; Manolopoulos, D. E. Chemical Reaction Rates from Ring Polymer Molecular Dynamics. *J. Chem. Phys.* **2005**, *122*, 084106.
- (11) Van Voorhis, T.; Kowalczyk, T.; Kaduk, B.; Wang, L.-P.; Cheng, C.-L.; Wu, Q. The Diabatic Picture of Electron Transfer, Reaction Barriers, and Molecular Dynamics. *Annu. Rev. Phys. Chem.* **2010**, *61*, 149–170.
- (12) Santra, B.; Ko, H.-Y.; Yeh, Y.-W.; Martelli, F.; Kaganovich, I.; Raitses, Y.; Car, R. Root-Growth of Boron Nitride Nanotubes: Experiments and Ab Initio Simulations. *Nanoscale* **2018**, *10*, 22223–22230.
- (13) Cheatham, T. E., III; Kollma, P. A. Observation of the A-DNA to B-DNA Transition During Unrestrained Molecular Dynamics in Aqueous Solution. *J. Mol. Biol.* **1996**, *259*, 434–444.
- (14) Sugita, Y.; Okamoto, Y. Replica-Exchange Molecular Dynamics Method for Protein Folding. *Chem. Phys. Lett.* **1999**, *314*, 141–151.
- (15) Karplus, M.; McCammon, J. A. Molecular Dynamics Simulations of Biomolecules. *Nat. Struct. Mol. Biol.* **2002**, *9*, 646–652.
- (16) Martelli, F.; Ko, H.-Y.; Borrallo, C. C.; Franzese, G. Structural Properties of Water Confined by Phospholipid Membranes. *Front. Phys.* **2018**, *13*, 136801.
- (17) 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.
- (18) 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.
- (19) Zhang, L.; Han, J.; Wang, H.; Saidi, W.; Car, R. 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.
- (20) 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.
- (21) Car, R.; Parrinello, M. Unified Approach for Molecular Dynamics and Density-Functional Theory. *Phys. Rev. Lett.* **1985**, *55*, 2471–2474.
- (22) Marx, D.; Hutter, J. *Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods*; Cambridge University Press: Cambridge, 2009.
- (23) Hohenberg, P.; Kohn, W. Inhomogeneous Electron Gas. *Phys. Rev.* **1964**, *136*, B864.
- (24) Kohn, W.; Sham, L. J. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* **1965**, *140*, A1133.
- (25) Parr, R. G.; Yang, W. *Density-Functional Theory of Atoms and Molecules*; Oxford University Press: New York, 1989.
- (26) *Lecture Notes in Physics*; Fiolhais, C., Nogueira, F., Marques, M., Eds.; Springer: New York, 2003; Vol. 620.
- (27) Becke, A. D. Perspective: Fifty Years of Density-Functional Theory in Chemical Physics. *J. Chem. Phys.* **2014**, *140*, 18A301.

- (28) Mardirossian, N.; Head-Gordon, M. Thirty Years of Density Functional Theory in Computational Chemistry: An Overview and Extensive Assessment of 200 Density Functionals. *Mol. Phys.* **2017**, *115*, 2315–2372.
- (29) Medvedev, M. G.; Bushmarinov, I. S.; Sun, J.; Perdew, J. P.; Lyssenko, K. A. Density Functional Theory is Straying from the Path Toward the Exact Functional. *Science* **2017**, *355*, 49–52.
- (30) Kepp, K. P. Comment on: Density Functional Theory is Straying from the Path Toward the Exact Functional. *Science* **2017**, *356*, 496.
- (31) Hammes-Schiffer, S. A Conundrum for Density Functional Theory. *Science* **2017**, *355*, 28–29.
- (32) Medvedev, M. G.; Bushmarinov, I. S.; Sun, J.; Perdew, J. P.; Lyssenko, K. A. Response to Comment on: Density Functional Theory is Straying from the Path Toward the Exact Functional. *Science* **2017**, *356*, 496.
- (33) Lehtola, S.; Steigemann, C.; Oliveira, M. J. T.; Marques, M. A. L. Recent Developments in libxc—A Comprehensive Library of Functionals for Density Functional Theory. *SoftwareX* **2018**, *7*, 1–5.
- (34) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1996**, *77*, 3865–3868.
- (35) Becke, A. D. Density-Functional Exchange-Energy Approximation with Correct Asymptotic Behavior. *Phys. Rev. A: At, Mol, Opt. Phys.* **1988**, *38*, 3098–3100.
- (36) 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.
- (37) 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.
- (38) Grimme, S.; Hansen, A.; Brandenburg, J. G.; Bannwarth, C. Dispersion-Corrected Mean-Field Electronic Structure Methods. *Chem. Rev.* **2016**, *116*, 5105–5154.
- (39) 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.
- (40) 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.
- (41) 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.
- (42) Cohen, A. J.; Mori-Sánchez, P.; Yang, W. Insights into Current Limitations of Density Functional Theory. *Science* **2008**, *321*, 792–794.
- (43) 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.
- (44) 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.
- (45) Hoja, J.; Ko, H.-Y.; Neumann, M. A.; Car, R.; DiStasio, R. A., Jr.; Tkatchenko, A. Reliable and Practical Computational Description of Molecular Crystal Polymorphs. *Sci. Adv.* **2019**, *5*, No. eaau3338.
- (46) Scheeres, D. J.; Hartzell, C. M.; Sánchez, P.; Swift, M. Scaling Forces to Asteroid Surfaces: The Role of Cohesion. *Icarus* **2010**, *210*, 968–984.
- (47) Rozitis, B.; MacLennan, E.; Emery, J. P. Cohesive Forces Prevent the Rotational Breakup of Rubble-Pile Asteroid (29075) 1950 DA. *Nature* **2014**, *512*, 174–176.
- (48) 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.
- (49) 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.
- (50) 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.
- (51) 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.
- (52) 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.
- (53) 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., Int. Ed.* **2018**, *130*, 15122–15126.
- (54) Marsman, M.; Paier, J.; Stroppa, A.; Kresse, G. Hybrid Functionals Applied to Extended Systems. *J. Phys.: Condens. Matter* **2008**, *20*, 064201.
- (55) Becke, A. D.; Johnson, E. R. Exchange-Hole Dipole Moment and the Dispersion Interaction Revisited. *J. Chem. Phys.* **2007**, *127*, 154108.
- (56) 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.
- (57) 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.
- (58) 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.
- (59) Caldeweyher, E.; Bannwarth, C.; Grimme, S. Extension of the D3 Dispersion Coefficient Model. *J. Chem. Phys.* **2017**, *147*, 034112.
- (60) 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.
- (61) 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.
- (62) 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.
- (63) 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.
- (64) 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.
- (65) Vydrov, O. A.; Van Voorhis, T. Nonlocal van der Waals Density Functional Made Simple. *Phys. Rev. Lett.* **2009**, *103*, 063004.
- (66) Lee, K.; Murray, E. 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.
- (67) Becke, A. D. Density-Functional Thermochemistry. III. The Role of Exact Exchange. *J. Chem. Phys.* **1993**, *98*, 5648–5652.
- (68) Heyd, J.; Scuseria, G. E.; Ernzerhof, M. Hybrid Functionals Based on a Screened Coulomb Potential. *J. Chem. Phys.* **2003**, *118*, 8207–8215.
- (69) 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.
- (70) Duchemin, I.; Gygi, F. A Scalable and Accurate Algorithm for the Computation of Hartree-Fock Exchange. *Comput. Phys. Commun.* **2010**, *181*, 855–860.

- (71) 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.
- (72) 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.
- (73) 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.
- (74) Guidon, M.; Hutter, J.; VandeVondele, J. Auxiliary Density Matrix Methods for Hartree-Fock Exchange Calculations. *J. Chem. Theory Comput.* **2010**, *6*, 2348–2364.
- (75) 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.
- (76) 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.
- (77) Lin, L. Adaptively Compressed Exchange Operator. *J. Chem. Theory Comput.* **2016**, *12*, 2242–2249.
- (78) 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.
- (79) Marzari, N.; Vanderbilt, D. Maximally Localized Generalized Wannier Functions for Composite Energy Bands. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1997**, *56*, 12847–12865.
- (80) 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.
- (81) 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.
- (82) Gygi, F. Compact Representations of Kohn-Sham Invariant Subspaces. *Phys. Rev. Lett.* **2009**, *102*, 166406.
- (83) Gygi, F.; Duchemin, I. Efficient Computation of Hartree-Fock Exchange Using Recursive Subspace Bisection. *J. Chem. Theory Comput.* **2013**, *9*, 582–587.
- (84) 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.
- (85) 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.
- (86) Damle, A.; Lin, L.; Ying, L. Localized Orbitals for Solids via Selected Columns of the Density Matrix. *J. Comput. Phys.* **2017**, *334*, 1–15.
- (87) Mountjoy, J.; Todd, M.; Mosey, N. J. Exact Exchange with Non-Orthogonal Generalized Wannier Functions. *J. Chem. Phys.* **2017**, *146*, 104108.
- (88) 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.
- (89) Guidon, M.; Schiffmann, F.; Hutter, J.; VandeVondele, J. Ab Initio Molecular Dynamics using Hybrid Density Functionals. *J. Chem. Phys.* **2008**, *128*, 214104.
- (90) Carnimeo, I.; Baroni, S.; Giannozzi, P. Fast Hybrid Density-Functional Computations Using Plane-Wave Basis Sets. *Electron. Struct.* **2019**, *1*, 015009.
- (91) Chawla, S.; Voth, G. A. Exact Exchange in Ab Initio Molecular Dynamics: An Efficient Plane-Wave Based Algorithm. *J. Chem. Phys.* **1998**, *108*, 4697–4700.
- (92) Sorouri, A.; Foulkes, W. M. C.; Hine, N. D. M. Accurate and Efficient Method for the Treatment of Exchange in a Plane-Wave Basis. *J. Chem. Phys.* **2006**, *124*, 064105.
- (93) Boffi, N. M.; Jain, M.; Natan, A. Efficient Computation of the Hartree-Fock Exchange in Real-Space with Projection Operators. *J. Chem. Theory Comput.* **2016**, *12*, 3614–3622.
- (94) Mandal, S.; Debnath, J.; Meyer, B.; Nair, N. N. Enhanced Sampling and Free Energy Calculations with Hybrid Functionals and Plane Waves for Chemical Reactions. *J. Chem. Phys.* **2018**, *149*, 144113.
- (95) Mandal, S.; Nair, N. N. Speeding-Up Ab Initio Molecular Dynamics with Hybrid Functionals Using Adaptively Compressed Exchange Operator Based Multiple Timestepping. *J. Chem. Phys.* **2019**, *151*, 151102.
- (96) Mandal, S.; Nair, N. N. Efficient Computation of Free Energy Surfaces of Chemical Reactions Using Ab Initio Molecular Dynamics with Hybrid Functionals and Plane Waves. *J. Comput. Chem.* **2020**, *41*, 1790–1797.
- (97) Mandal, S.; Thakkur, V.; Nair, N. N. Achieving an Order of Magnitude Speedup in Hybrid-Functional- and Plane-Wave-Based Ab Initio Molecular Dynamics: Applications to Proton-Transfer Reactions in Enzymes and in Solution. *J. Chem. Theory Comput.* **2021**, *17*, 2244–2255.
- (98) Ko, H.-Y.; Jia, J.; Santra, B.; Wu, X.; Car, R.; DiStasio, R. A., Jr. Enabling Large-Scale Condensed-Phase Hybrid Density Functional Theory Based Ab Initio Molecular Dynamics. 1. Theory, Algorithm, and Performance. *J. Chem. Theory Comput.* **2020**, *16*, 3757–3785.
- (99) 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.; Gougousis, 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.
- (100) Giannozzi, P.; Andreussi, O.; Brumme, T.; Bunau, O.; Buongiorno Nardelli, M.; Calandra, M.; Car, R.; Cavazzoni, C.; Ceresoli, D.; Cococcioni, M.; Colonna, N.; Carnimeo, I.; Dal Corso, A.; 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.
- (101) Kohn, W. Analytic Properties of Bloch Waves and Wannier Functions. *Phys. Rev.* **1959**, *115*, 809–821.
- (102) des Cloizeaux, J. Analytical Properties of n-Dimensional Energy Bands and Wannier Functions. *Phys. Rev.* **1964**, *135*, A698–A707.
- (103) Nenciu, G. Existence of the Exponentially Localised Wannier Functions. *Commun. Math. Phys.* **1983**, *91*, 81–85.
- (104) Niu, Q. Theory of the Quantized Adiabatic Particle Transport. *Mod. Phys. Lett. B* **1991**, *05*, 923–931.
- (105) Panati, G.; Pisante, A. Bloch Bundles, Marzari-Vanderbilt Functional and Maximally Localized Wannier Functions. *Commun. Math. Phys.* **2013**, *322*, 835–875.
- (106) 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.
- (107) Wu, X.; Walter, E. J.; Rappe, A. M.; Car, R.; Selloni, A. Hybrid Density Functional Calculations of the Band Gap of $\text{Ga}_{1-x}\text{In}_x\text{N}$. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2009**, *80*, 115201.

- (108) 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.
- (109) Santra, B.; DiStasio, R. A., Jr.; Martelli, F.; Car, R. Local Structure Analysis in *Ab Initio* Liquid Water. *Mol. Phys.* **2015**, *113*, 2829–2841.
- (110) 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.
- (111) 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.
- (112) Ko, H.-Y.; DiStasio, R. A., Jr.; Santra, B.; Car, R. Thermal Expansion in Dispersion-Bound Molecular Crystals. *Phys. Rev. Mater.* **2018**, *2*, 055603.
- (113) Parrinello, M.; Rahman, A. Crystal Structure and Pair Potentials: A Molecular-Dynamics Study. *Phys. Rev. Lett.* **1980**, *45*, 1196–1199.
- (114) Sharma, M.; Wu, Y.; Car, R. *Ab Initio* Molecular Dynamics with Maximally Localized Wannier Functions. *Int. J. Quantum Chem.* **2003**, *95*, 821–829.
- (115) 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.
- (116) 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.
- (117) 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.
- (118) Kümmel, S.; Kronik, L. Orbital-Dependent Density Functionals: Theory and Applications. *Rev. Mod. Phys.* **2008**, *80*, 3–60.
- (119) Jónsson, E. Ö.; Lehtola, S.; Puska, M.; Jónsson, H. Theory and Applications of Generalized Pipek–Mezey Wannier Functions. *J. Chem. Theory Comput.* **2017**, *13*, 460–474.
- (120) Fornberg, B. Generation of Finite Difference Formulas on Arbitrarily Spaced Grids. *Math. Comput.* **1988**, *51*, 699–706.
- (121) Brandt, A.; Diskin, B. Multigrid Solvers for Nonaligned Sonic Flows. *SIAM J. Sci. Comput.* **1999**, *21*, 473–501.
- (122) Natan, A.; Benjamini, A.; Naveh, D.; Kronik, L.; Tiago, M. L.; Beckman, S. P.; Chelikowsky, J. R. Real-Space Pseudopotential Method for First Principles Calculations of General Periodic and Partially Periodic Systems. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2008**, *78*, 075109.
- (123) The use of all three unit lattice directions in addition to (one to three) grid-resolved unit auxiliary direction(s) may not always be the most efficient choice when satisfying the NK criteria outlined in Section 3.3.2. For example, the nearest-neighbor directions in certain simulation cells (e.g., rhombohedral cells with an angle $<30^\circ$) may not include $\{\hat{L}_1, \hat{L}_2, \hat{L}_3\}$, as pointed out by Natan, Kronik, and coworkers.¹²² Due to the complexities associated with choosing the auxiliary directions and the potential risk of breaking the semi-positive-definiteness of the Laplacian, we follow the original NK approach by always evaluating the pure derivatives in eq 44 along the three unit lattice directions.
- (124) For increased computational efficiency, sweeps over grid-point shells account for inversion symmetry in the real-space grid, and only include non-axial grid points with $\xi_3 - (\xi_0)_3 > 0$ and $\xi_2 - (\xi_0)_2 > 0$ when $\xi_3 - (\xi_0)_3 = 0$. As such, there are only 10 grid points in $\mathbb{G}_1$.
- (125) Fourme, R.; André, D.; Renaud, M. A Redetermination and Group-Refinement of the Molecular Packing of Benzene II at 25 Kilobars. *Acta Crystallogr., Sect. A: Cryst. Phys., Diff., Theor. Gen. Crystallogr.* **1971**, *27*, 1275–1276.
- (126) The specific case of $N_{\text{pure}} = 5$ is also possible for certain triclinic systems, but was not specifically investigated in this case study. For instance, consider a triclinic cell with $\hat{L}_1 = \hat{x}$, $\hat{L}_2$ and $\hat{L}_3$ and located in the xy and xz planes, and NK auxiliary directions (i.e., the output of Algorithm 1) located within the xy , xz , and yz planes. In this case, the final number of NK auxiliary directions would be $N_{\text{aux}} = 2$, and would correspond to the grid-resolved angle bisectors between: (i) $\hat{L}_1$ and $\hat{L}_2$ and (ii) $\hat{L}_1$ and $\hat{L}_3$.
- (127) A development version of the code used in this work is available for download upon request.
- (128) Perdew, J. P.; Ernzerhof, M.; Burke, K. Rationale for Mixing Exact Exchange with Density Functional Approximations. *J. Chem. Phys.* **1996**, *105*, 9982–9985.
- (129) Adamo, C.; Barone, V. Toward Reliable Density Functional Methods without Adjustable Parameters: The PBE0 Model. *J. Chem. Phys.* **1999**, *110*, 6158–6170.
- (130) Bernal, J. D.; Fowler, R. H. A Theory of Water and Ionic Solution, with Particular Reference to Hydrogen and Hydroxyl Ions. *J. Chem. Phys.* **1933**, *1*, 515–548.
- (131) Hayward, J. A.; Reimers, J. R. Unit Cells for the Simulation of Hexagonal Ice. *J. Chem. Phys.* **1997**, *106*, 1518–1529.
- (132) Cota, E.; Hoover, W. G. Computer Simulation of Hexagonal Ice. *J. Chem. Phys.* **1977**, *67*, 3839–3840.
- (133) Santra, B.; Klimeš, J.; Tkatchenko, A.; Alfè, D.; Slater, B.; Michaelides, A.; Car, R.; Scheffler, M. On the Accuracy of van der Waals Inclusive Density-Functional Theory Exchange-Correlation Functionals for Ice at Ambient and High Pressures. *J. Chem. Phys.* **2013**, *139*, 154702.
- (134) Martyna, G. J.; Klein, M. L.; Tuckerman, M. Nosé-Hoover Chains: The Canonical Ensemble via Continuous Dynamics. *J. Chem. Phys.* **1992**, *97*, 2635–2643.
- (135) 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.
- (136) Bernasconi, M.; Chiarotti, G. L.; Focher, P.; Scandolo, S.; Tosatti, E.; Parrinello, M. First-Principle-Constant Pressure Molecular Dynamics. *J. Phys. Chem. Solids* **1995**, *56*, 501–505.
- (137) To maintain a constant kinetic energy cutoff of E_0 , a smooth step function of height A and width σ was added to the kinetic factor G^2 as follows:¹³⁶ $G^2 \rightarrow G^2 + A[1 + \text{erf}(1/2G^2 - E_0/\sigma)]$. In this work, we used $A = 200$ Ry, $\sigma = 15$ Ry, and $E_0 = 130$ Ry, which correspond to the following input parameters in QE: $\text{qcutoff} = 200$, $\text{q2sigma} = 15$, $\text{ecfixed} = 130$, and $\text{ecutwfc} = 150$.
- (138) Hamann, D. R.; Schlüter, M.; Chiang, C. Norm-Conserving Pseudopotentials. *Phys. Rev. Lett.* **1979**, *43*, 1494–1497.
- (139) Vanderbilt, D. Optimally Smooth Norm-Conserving Pseudopotentials. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1985**, *32*, 8412–8415.
- (140) Gygi, F. Architecture of Qbox: A Scalable First-Principles Molecular Dynamics Code. *IBM J. Res. Dev.* **2008**, *52*, 137–144.
- (141) As discussed in PAPER-1,⁹⁸ we found a deprecated `invFFT` call in the current version of QE that unnecessarily recomputes the MLWFs in real space. As such, the $\langle t_{\text{total}} \rangle$ values reported in Table 3 do not include the computational cost associated with this routine.
- (142) Although these tests were based on short (50-step) NVT simulations, the average number of floating point operations executed during each CPMD step (for the simulation with the 150 Ry cutoff) is essentially the same as that during a (similarly short) NpT simulation using the same cutoff. Taken together with the fact that the evaluation of σ_{xx} introduces minor ($<1\%$) computational overhead (see Section 4.2), we believe that this test is an accurate and reliable assessment of the OpenMP strong-scaling efficiency of the extended `exx` module introduced in this work.