JCTC Journal of Chemical Theory and Computation

Article

pubs.acs.org/JCTC

Electron Correlation in the Condensed Phase from a Resolution of Identity Approach Based on the Gaussian and Plane Waves Scheme

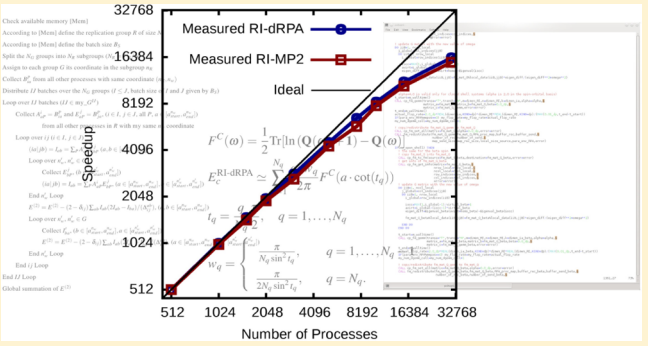
Mauro Del Ben,*[,†] Jürg Hutter,*[,†] and Joost VandeVondele*[,‡]

[†]Institute of Physical Chemistry, University of Zürich, Winterthurerstrasse 190, CH-8057 Zürich, Switzerland

[‡]Department of Materials, ETH Zürich, Wolfgang-Pauli-Strasse 27, CH-8093 Zürich, Switzerland

Supporting Information

ABSTRACT: The second-order Møller−Plesset perturbation energy (MP2) and the Random Phase Approximation (RPA) correlation energy are increasingly popular post-Kohn−Sham correlation methods. Here, a novel algorithm based on a hybrid Gaussian and Plane Waves (GPW) approach with the resolution-of-identity (RI) approximation is developed for MP2, scaled opposite-spin MP2 (SOS-MP2), and direct-RPA (dRPA) correlation energies of finite and extended system. The key feature of the method is that the three center electron repulsion integrals ($\mu\nu|P$) necessary for the RI approximation are computed by direct integration between the products of Gaussian basis functions $\mu\nu$ and the electrostatic potential arising from the RI fitting densities P. The electrostatic potential is obtained in a plane waves basis set after solving the Poisson equation in Fourier space. This scheme is highly efficient for condensed phase systems and offers a particularly easy way for parallel implementation. The RI approximation allows to speed up the MP2 energy calculations by a factor 10 to 15 compared to the canonical implementation but still requires $O(N^5)$ operations. On the other hand, the combination of RI with a Laplace approach in SOS-MP2 and an imaginary frequency integration in dRPA reduces the computational effort to $O(N^4)$ in both cases. In addition to that, our implementations have low memory requirements and display excellent parallel scalability up to tens of thousands of processes. Furthermore, exploiting graphics processing units (GPU), a further speedup by a factor ∼2 is observed compared to the standard only CPU implementations. In this way, RI-MP2, RI-SOS-MP2, and RI-dRPA calculations for condensed phase systems containing hundreds of atoms and thousands of basis functions can be performed within minutes employing a few hundred hybrid nodes. In order to validate the presented methods, various molecular crystals have been employed as benchmark systems to assess the performance, while solid LiH has been used to study the convergence with respect to the basis set and system size in the case of RI-MP2 and RI-dRPA.



1. INTRODUCTION

Density Functional Theory (DFT) has become the most widely used quantum mechanical tool in chemistry and physics for predicting properties of materials ranging from molecules to condensed phase systems. The strength of DFT lies in a good compromise between accuracy and computational effort. In fact, existing implementations of DFT in its local and semilocal formulation display a computational complexity growing only $O(N) - O(N^3)$, where N represents the system size. However, going beyond these simple formulations is required to make significant progress on the way to high accuracy.

In this respect, Perdew et al.[1] have proposed a systematic classification of the existing approximate density functionals based on the information they employ. This classification is known as "Jacob's ladder". The fifth rung of "Jacob's ladder" includes methods that make use of information from the occupied and virtual orbitals and allows for taking into account the nonlocal dynamical electron correlation contributions, including the long-range van der Waals (vdW) dispersion interactions. Most of the 5^{th} rung approaches are based on either the random phase approximation (RPA), introduced in the framework of DFT via the so-called adiabatic-connection fluctuation−dissipation theorem (ACFD),[2−10] or many body perturbation theory (MBPT) in the form of second-order Møller−Plesset (MP2)[11−17] perturbation theory.

The drawback connected with these improved functionals is the computational cost, that grows significantly compared to standard DFT approaches of lower rung. In fact, even if low-order scaling methods exist,[18−32] the formal computational effort scales between $O(N^4)$ to $O(N^5)$ with respect to systems size. Furthermore, these methods exhibit the same slow energy convergence as correlated wave function methods due to the electron coalescence cusp,[33−35] implying that these calculations need larger basis sets than standard DFT to reach a similar convergence, unless explicitly correlated[36] or range separa-

Received: March 19, 2013
Published: May 6, 2013

tion[37,38] methods are employed. These are the reasons that have impeded the widespread use of fifth rung density functional approaches, limiting their application to small size systems.

In order to perform these calculations with acceptable time to solution and to extend the applicability of these methods to larger systems, the development of efficient algorithms as well as computer implementations, suitable for massively parallel machines, are of prime interest. In this respect, the resolution of identity approximation (RI) has shown to be a powerful technique to reduce the prefactor and the scaling for both RPA and MP2 calculations;[39−50] furthermore several parallel MP2 and RI-MP2 implementations have been proposed[51−62] displaying good scalability up to few hundred cores, but none of them have been yet reported in the case of RPA.

Here, we present a novel algorithm for the resolution of identity approximation based on Gaussian and Plane Wave (GPW) approach.[63] Of central importance in the RI approximation is the computation of the three center electron repulsion integrals of the type $(\mu\nu|P) = \Sigma_R(\mu\nu|R)L_{PR}^{-1}$, where **L** is obtained by the Cholesky decomposition of the two center electron repulsion integrals $(Q|R)$ over auxiliary Gaussian basis functions. The key aspect of our method is that the three center electron repulsion integrals of the type $(\mu\nu|P)$ are computed by direct integration between the product of the Gaussian basis function $(\mu\nu|$ and the electrostatic potential arising from the RI fitting densities $(P| = \Sigma_R(R|L_{PR}^{-1}$. Within the GPW approach, the electrostatic potential is obtained by solving the Poisson equation in Fourier space, after expanding the RI fitting densities in a plane wave (PW) auxiliary basis set. Note that the PW auxiliary basis is a natural choice for periodic systems, but it can equally be used for gas phase or surface calculations. Indeed, once the density is specified on a regular grid efficient methods are available for solving the Poisson equation with free (for example cluster or slab) boundary conditions.[64−66] The GPW scheme is highly efficient, especially for condensed phase systems, and particularly suitable for parallel implementation, since the $(\mu\nu|P)$ integrals can be computed in a communication free way simply by distributing independent $(P|$ vectors over processes.

Alternative RI methods suitable for the condensed phase that do not rely on a PW auxiliary basis have been described in the literature. A first approach, density fitting (DF) has been reported for the calculation of the local-MP2 energy for periodic systems.[25,45,46,67] In this approach, the large majority of the fitting functions (FFs) are chosen to be Poisson-type functions (PTFs), simplifying the calculation of the Coulomb two-electron integrals into overlap integrals and allowing for rapidly converging lattice sums.[45] Nevertheless, PTF need to be augmented with a small set of Gaussian type functions to be able to describe multipole moments. A second approach, which employs an RI expansion for calculating the Coulomb term in periodic systems, has been proposed by Burow and co-workers.[68] Here, the lattice sums are partitioned into a near-field and far-field part, the first treated analytically, the latter using the periodic continuous fast multipole method.[69,70] Compared to the GPW scheme, these methods are more involved. However, the simplicity of the GPW method has as a drawback that all-electron calculations are not possible and that pseudopotentials have to be employed in order to have densities that are smooth. The Gaussian and Augmented Plane Wave (GAPW) scheme[71,72] overcomes this limitation and is suitable for all-electron calculations, but our RI implementation is currently limited to the GPW method only.

We apply the RI approximation to the computation of the MP2, scaled opposite-spin MP2 (SOS-MP2),[48,73] and direct-RPA (dRPA)[49,74,75] correlation energies, for condensed and finite systems. In the case of the computation of the MP2 energy the RI approximation reduces drastically the prefactor for the overall calculation, giving speedups, compared to standard implementations, up to a factor of fifteen, but still retaining the formal $O(N^5)$ scaling. On the other hand, in the SOS-MP2 and dRPA cases, the RI approximation not only speeds up the calculation of the integrals but also allows, in both cases, for reducing the scaling to $O(N^4)$. This is achieved in the SOS-MP2 case by a Laplace approach[48,76,77] and in dRPA by reformulating the correlation energy expression in terms of an imaginary frequency integral.[49]

The parallel algorithms for the calculation of the RI integrals within the GPW approach and for the computation of the MP2, SOS-MP2, and dRPA correlation energies within the RI approximation have been implemented in CP2K.[78] The presented parallel implementations display excellent parallel scalability and efficiency up to several thousands of processes and allow for correlation energy calculation on systems comprised of hundreds of atoms and thousands of basis functions within minutes. We also report the performance of hybrid implementations making use of graphics processing units (GPUs) showing further speedup compared to a standard CPU-only implementation.

The remainder of the paper is organized as follows: first the basic theoretical framework of each method together with the employed notation are introduced, and then the parallel implementations and performance are discussed in detail. Finally, several benchmark calculations for different condensed phase systems are reported in order to validate the proposed approaches.

2. THEORY

In this section the theoretical framework of the methods is briefly presented referring to the original works for more details. First the resolution of the identity approximation for two electron repulsion integrals (ERIs) is introduced, and then its application to the different correlation methods is formulated giving the working expressions employed in the implementations. Finally, the Gaussian and Plane-Waves scheme is reviewed, and its application together with the RI approximation is discussed. The following index notation has been adopted: $i,j,k,\ldots$ refer to canonical occupied molecular orbitals (MOs), $a,b,c,\ldots$ to canonical virtual MOs, $\mu,\nu,\lambda,\ldots$ to atomic orbital basis set functions (AO), $P,Q,R,\ldots$ to auxiliary basis set functions (AUX). The one electron MO and AO functions are symbolized respectively with ψ and ϕ. The number of occupied and virtual orbitals is denoted by o and v, while the total number of primary and auxiliary basis functions are denoted as n and N_a. In order to express, in general, the system size, the symbol N is used.

2.1. The Resolution of the Identity Approximation.

The two electron repulsion integrals, in Mulliken notation, of the type $(ia|jb)$ are of central importance for all the methods presented in this paper. Within the RI approximation,[79,80] based on the Coulomb metric,[81] these integrals are factorized according to

$$(ia|jb)_{RI} = \sum_{PQ} (ia|P)(P|Q)^{-1}(Q|jb) \tag{1}$$

here the $(P|Q)^{-1}$ are the matrix elements of the inverse of the matrix $(P|Q)$, where each element is given by

$$(P|Q) = \int\int \phi_P(\vec{r}_1)\frac{1}{|\vec{r}_1 - \vec{r}_2|}\phi_Q(\vec{r}_2)d\vec{r}_1 d\vec{r}_2 \tag{2}$$

The auxiliary basis set size N_a grows only linearly with the system size.[41]

The main advantage of the RI approximation is that four center integrals of the type $(ia|jb)$ are computed from three and two center ERIs. This allows to strongly reduce the effort for the integral computation without significant loss of accuracy.[41,82]

Since the $(P|Q)$ matrix is positive definite the calculation of $(P|Q)^{-1}$ can be efficiently performed by a Cholesky decomposition of $(P|Q)$

$$(P|Q) = \sum_{R} L_{PR}L^{T}_{RQ} \tag{3}$$

followed by the efficient inversion of the triangular matrix **L**, such that

$$(P|Q)^{-1} = \sum_{R} L^{-T}_{PR}L^{-1}_{RQ} \tag{4}$$

In this way the factorization of the $(ia|jb)$ integrals can be expressed in a compact form as

$$(ia|jb)_{RI} = \sum_{P} B^{ia}_{P}B^{jb}_{P} \tag{5}$$

Here **B** is a matrix with ov rows and N_a columns, given by

$$B^{ia}_{P} = \sum_{R} (ia|R)L^{-1}_{PR} \tag{6}$$

Since the three center integrals $(ia|R)$ are computed starting from integrals over AOs $(\mu\nu|R)$, the final expression for the B^{ia}_{P} elements reads

$$(ia|P) = \sum_{\nu} C_{\nu a}\sum_{\mu} C_{\mu i}\sum_{R} (\mu\nu|R)L^{-1}_{PR} \tag{7}$$

where **C** is the MO coefficient matrix.

The computation of the **B** matrix can thus be summarized as follows. First the two center integrals $(P|Q)$ are computed and from that, via Cholesky decomposition and triangular inversion, $\mathbf{L}^{-1}$. These two steps formally scale $O(N^2)$ and $O(N^3)$, respectively. Then the three center integrals $(\mu\nu|R)$ are computed and subsequently transformed using the **C** and $\mathbf{L}^{-1}$ matrices (eq 7). In this case the integral computation requires formally $O(N^3)$ operations, while the integral transformations scale $O(N^4)$.

This means that, within RI approximation, the asymptotically dominating time determining step in computing **B** is the index transformation scaling formally $O(N^4)$.

2.2. RI-MP2 Method. In Second Order Møller−Plesset Perturbation Theory, the correlation energy $E^{(2)}$ for a closed shell system is given by

$$E^{(2)} = -\sum_{i\le j}^{o}(2 - \delta_{ij})\sum_{ab}^{v}\frac{(ia|jb)[2(ia|jb) - (ib|ja)]}{\varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j} \tag{8}$$

where ε_a and ε_i are orbital energies.

In a canonical MP2 energy algorithm the time limiting step is the computation of the $(ia|jb)$ integrals obtained from the ERIs over AO $(\mu\nu|\lambda\sigma)$ via four consecutive integral transformations:

$$(ia|jb) = \sum_{\mu} C_{\mu i}\sum_{\nu} C_{\nu a}\sum_{\lambda} C_{\lambda j}\sum_{\sigma} C_{\sigma b}(\mu\nu|\lambda\sigma) \tag{9}$$

The computational effort for each of the four quarter transformations, if the occupied orbitals are transformed before the virtual, and sparsity is not considered, is $O(on^4)$, $O(o^2n^3)$, $O(o^2vn^2)$, and $O(o^2v^2n)$, making the MP2 energy calculation a method scaling as $O(N^5)$.[83]

The application of the RI approximation to the MP2 energy calculation is straightforward.[40] It consists simply in the replacement of the $(ia|jb)$ integrals with the approximated $(ia|jb)_{RI}$ given in eq 5. The computation of the $(ia|jb)_{RI}$ requires $O(o^2v^2N_a)$ operations implying that the RI-MP2 method is also scaling $O(N^5)$.

The main reason for the speedup observed in RI-MP2 lies in the fact that ERIs over four indices are replaced by ERIs over two and three indices, which strongly reduces the effort in the integral computation part. However, this means that the speedup becomes less and less pronounced increasing the system size. Indeed, for large systems, when the $O(N^5)$ step dominates, RI-MP2 and standard MP2 have a similar computational cost.

2.3. Laplace Transform Scaled Opposite-Spin RI-MP2 Method. The scaled opposite-spin second order correlation energy $E^{(2)}_{SOS}$ is defined as[48,73]

$$E^{(2)}_{SOS} = c_{SOS}E^{(2)}_{OS} \tag{10}$$

where c_{SOS} is a scaling factor (usually 1.3), and $E^{(2)}_{OS}$ is the opposite spin component of the MP2 energy:

$$E^{(2)}_{OS} = -\sum_{ia}\sum_{jb}\frac{(ia|jb)^2}{\varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j} = -\sum_{ia}\sum_{jb}\frac{(ia|jb)^2}{\Delta^{ab}_{ij}} \tag{11}$$

The energy denominator Δ^{ab}_{ij} in eq 11 can be rewritten using the Laplace transformation $1/x = \int_0^\infty e^{-xt}dt$ so that[76,77]

$$E^{(2)}_{OS} = -\int_0^\infty dt\sum_{ia}\sum_{jb}(ia|jb)^2 e^{-t\Delta^{ab}_{ij}} \tag{12}$$

The integral over t in eq 12 can be approximated by a numerical quadrature. Considering N_q quadrature points $E^{(2)}_{OS}$ can be written as

$$\begin{aligned} E^{(2)}_{OS} &= -\sum_{q}^{N_q}\sum_{ia}\sum_{jb} w_q(ia|jb)^2 e^{-t_q\Delta^{ab}_{ij}} \\ &= -\sum_{q}^{N_q}\sum_{ia}\sum_{jb}(\overline{ia|jb})^2 \end{aligned} \tag{13}$$

where $(\overline{ia|jb})$ represents an ERI calculated over scaled molecular orbitals defined as

$$\overline{\psi}^q_a = w_q^{\frac{1}{8}}\psi_a e^{-\frac{1}{2}t_q\varepsilon_a} \tag{14}$$

$$\overline{\psi}^q_i = w_q^{\frac{1}{8}}\psi_i e^{\frac{1}{2}t_q\varepsilon_i} \tag{15}$$

The Laplace SOS-MP2 energy can thus be evaluated employing a slightly modified version of a canonical MP2

algorithm at the price of $O(N^5)$ operations. However, the introduction of the RI approximation, in this case, allows for a reduction of the formal computational effort to $O(N^4)$.[57] In fact the integrals $(\overline{ia}|\overline{jb})$ within RI can be approximated as

$$(\overline{ia}|\overline{jb})_{RI} = \sum_P B_P^{\overline{ia}} B_P^{\overline{jb}} \tag{16}$$

$$B_P^{\overline{ia}} = w_q^{\frac{1}{4}} B_P^{ia} e^{\frac{1}{2} t_q (\varepsilon_i - \varepsilon_a)} \tag{17}$$

substituting eq 16 into 13 and carrying out the two independent summations over ia and jb gives

$$\begin{aligned} E_{OS}^{(2)} &= -\sum_q^{N_q} \sum_{ia} \sum_{jb} \sum_{PR} B_P^{\overline{ia}} B_P^{\overline{jb}} B_R^{\overline{ia}} B_R^{\overline{jb}} \\ &= -\sum_q^{N_q} \sum_{PR} \bar{Q}_{PR} \bar{Q}_{PR} \\ &= -\sum_q^{N_q} \mathrm{Tr}(\bar{Q} \bar{Q}^T) \end{aligned} \tag{18}$$

The $\bar{\mathbf{Q}}$ matrix is a $N_a \times N_a$ matrix calculated as

$$\bar{Q}_{PR} = \sum_{ia} B_P^{\overline{ia}} B_R^{\overline{ia}} \tag{19}$$

This means that, for each quadrature point q, the calculation of the associated $\bar{\mathbf{Q}}$ requires only $O(ovN_a^2)$ operations. The overall computational effort is thus $O(N_q ovN_a^2)$. Since the number of quadrature points is independent of system size, the RI-Laplace-SOS-MP2 method scales $O(N^4)$. The quadrature points can be found with a minimax approximation,[84,85] and as few as 6−8 points can yield micro-Hartree accuracy.

2.4. RI Direct Random Phase Approximation Correlation Energy Method. The RPA correlation energy[69] is given as the difference between the zero point energy of two harmonic oscillator excitation problems for which the first includes correlated ground state (RPA) and the second not (configuration interaction singles CIS)[74,75,86]

$$E_c^{\mathrm{RPA}} = \frac{1}{2} \sum_i (\omega_i - \nu_i) = \frac{1}{2} \mathrm{Tr}(\omega - \mathbf{A}) \tag{20}$$

where ω is the diagonal matrix of the positive RPA excitation energies that can be obtained from the non-Hermitian eigenvalue problem

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ -\mathbf{B} & -\mathbf{A} \end{pmatrix} \begin{pmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{Y} & \mathbf{X} \end{pmatrix} = \begin{pmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{Y} & \mathbf{X} \end{pmatrix} \begin{pmatrix} \omega & \mathbf{0} \\ \mathbf{0} & -\omega \end{pmatrix} \tag{21}$$

Within the direct-RPA (dRPA) approach, that is RPA without including exchange contributions,[49,75] the orbital rotation Hessian matrices $\mathbf{A}$ and $\mathbf{B}$ are defined as

$$(A - B)_{ia,jb} = (\varepsilon_a - \varepsilon_i)\delta_{ij}\delta_{ab} \tag{22}$$

$$(A + B)_{ia,jb} = (\varepsilon_a - \varepsilon_i)\delta_{ij}\delta_{ab} + 2(ia|jb) \tag{23}$$

here all matrices have dimension $ov \times ov$. It is known from time dependent density functional theory that eq 21 can be transformed to a Hermitian problem[87]

$$\mathbf{MZ} = \mathbf{Z}\omega^2, \quad \mathbf{Z}^{\mathbf{T}}\mathbf{Z} = \mathbf{1} \tag{24}$$

where

$$\mathbf{M} = (\mathbf{A} - \mathbf{B})^{1/2}(\mathbf{A} + \mathbf{B})(\mathbf{A} - \mathbf{B})^{1/2} \tag{25}$$

It can be finally shown that eq 20 can be rewritten in term of the square root of $\mathbf{M}$ as[86]

$$E_c^{\mathrm{dRPA}} = \frac{1}{2} \mathrm{Tr}(\mathbf{M}^{1/2} - \mathbf{A}) \tag{26}$$

According to eq 26 a straightforward approach for computing E_c^{dRPA} implies the storage and calculation of $\mathbf{M}^{1/2}$ and so requires $O(N^4)$ memory and $O(N^6)$ computational effort.

However, the dRPA correlation energy within the RI approximation $E_c^{\mathrm{RI\text{-}dRPA}}$ can be expressed in terms of a frequency integral[49]

$$E_c^{\mathrm{RI\text{-}dRPA}} = \frac{1}{2} \int_{-\infty}^{+\infty} \frac{d\omega}{2\pi} \mathrm{Tr}(\ln(\mathbf{1} + \mathbf{Q}(\omega)) - \mathbf{Q}(\omega)) \tag{27}$$

Here the frequency dependent matrix $\mathbf{Q}(\omega)$ has size $N_a \times N_a$ and is given by $\mathbf{Q}(\omega) = 2\mathbf{B}^T\mathbf{G}(\omega)\mathbf{B}$ where $\mathbf{G}(\omega)$ is a diagonal $ov \times ov$ matrix with elements

$$G_{ia,ia}(\omega) = (\varepsilon_a - \varepsilon_i)((\varepsilon_a - \varepsilon_i)^2 + \omega^2)^{-1}$$

For a given ω, the computation of the integrand function in eq 27 requires $O(N^4)$ operations. The integral of eq 27 can be efficiently calculated by Clenshaw-Curtis numerical quadrature,[88] and usually 30−40 quadrature points are enough for micro-Hartree accuracy.

Thus the introduction of the resolution of the identity (RI) approximation to the MO-ERIs and the frequency integration techniques for computing $E_c^{\mathrm{RI-dRPA}}$ lead to a reduction of the computational cost to $O(N^4 N_q)$ and $O(N^3)$ storage only, where N_q is the number of points used in the numerical quadrature of the integral in eq 27.

2.5. RI Gaussian and Plane-Waves Method. The Gaussian and Plane-Waves (GPW) method has been shown to be an efficient approach for computing ERIs especially when periodic boundary conditions are considered.[63] The basis of the GPW approach for computation of the ERIs is the direct formulation of the half transformed integrals of the type $(ia|\lambda\sigma)$ in terms of the electrostatic potential v^{ia} of the pair density ρ^{ia}

$$\begin{aligned} (ia|\lambda\sigma) &= \int\int \psi_i(\vec{r}_1)\psi_a(\vec{r}_1)\frac{1}{\vec{r}_{12}}\phi_\lambda(\vec{r}_2)\phi_\sigma(\vec{r}_2)d\vec{r}_1 d\vec{r}_2 \\ &= \int \left[\int \frac{\rho^{ia}(\vec{r}_1)}{\vec{r}_{12}} d\vec{r}_1\right] \phi_\lambda(\vec{r}_2)\phi_\sigma(\vec{r}_2)d\vec{r}_2 \\ &= \int v^{ia}(\vec{r}_2)\phi_\lambda(\vec{r}_2)\phi_\sigma(\vec{r}_2)d\vec{r}_2 \end{aligned} \tag{28}$$

The form of the last equation is essentially identical to the one used in the GPW method[89] to compute matrix elements of the Hartree potential. Thus, the highly efficient implementation of that operation in CP2K[78] can be directly used, and we refer to ref 90 for a detailed discussion.

Within the RI approximation, two types of ERIs have to be computed, the two center $(P|Q)$ and three center $(ia|P)$. Focusing on the three center case, they are computed, eq 7, starting from the integrals over AOs that are subsequently transformed with the two matrices $\mathbf{C}$ and $\mathbf{L}^{-1}$. Employing the GPW method, eq 28, the index transformation over the auxiliary basis can be avoided, since it is possible to directly compute half transformed integrals for an associated density ρ as

Assign each process its coordinate (n_P, n_w)

Create ranges $[P_{start}^{n_P}, P_{end}^{n_P}]$, $[a_{start}^{n_w}, a_{end}^{n_w}]$

Loop over P auxiliary basis functions $(P_{start}^{n_P} \le P \le P_{end}^{n_P})$

Calculate density $\rho^P(\vec{R}) = \phi_P(\vec{R})$ on the real space grid

Transfer $\rho^P(\vec{R}) \rightarrow \rho^P(\vec{G})$: $\rho^P(\vec{G}) = \mathbf{FFT}[\rho^P(\vec{R})]$

Solve Poisson's Equation: $\rho^P(\vec{G}) \rightarrow v^P(\vec{G})$

Transfer $v^P(\vec{G}) \rightarrow v^P(\vec{R})$: $v^P(\vec{R}) = \mathbf{FFT}^{-1}[v^P(\vec{G})]$

Integrate potential in real space: $I_Q^P = \int v^P(\vec{R})\phi_Q(\vec{R})d\vec{R}$ (all AUX Q)

Store I_Q^P

End P Loop

Redistribute I_Q^P integrals into $(Q|P)$ parallel distributed matrix

Cholesky decomposition of $(Q|P) = \mathbf{L}\mathbf{L}^{\mathbf{T}}$ (SCALAPACK)

Triangular inversion of $\mathbf{L} \rightarrow \mathbf{L}^{-1}$

Collect and store $\mathbf{L}^{-1}$ rows $M_R^P = L_{PR}^{-1}$ (all AUX R, $P_{start}^{n_P} \le P \le P_{end}^{n_P}$)

Loop over P auxiliary basis functions $(P_{start}^{n_P} \le P \le P_{end}^{n_P})$

Calculate density $\rho^P(\vec{R}) = \sum_R \phi_R(\vec{R})M_R^P$ on the real space grid

Transfer $\rho^P(\vec{R}) \rightarrow \rho^P(\vec{G})$: $\rho^P(\vec{G}) = \mathbf{FFT}[\rho^P(\vec{R})]$

Solve Poisson's Equation: $\rho^P(\vec{G}) \rightarrow v^P(\vec{G})$

Transfer $v^P(\vec{G}) \rightarrow v^P(\vec{R})$: $v^P(\vec{R}) = \mathbf{FFT}^{-1}[v^P(\vec{G})]$

Integrate potential in real space: $(\mu\nu|P) = S_{\mu\nu}^P = \int \phi_\mu(\vec{R})\phi_\nu(\vec{R})v^P(\vec{R})d\vec{R}$ (all μ,ν)

Index transformation $\mathbf{V} = (\mathbf{C}_o^\dagger \mathbf{S}^P)\mathbf{C}_v$

Redistribute and store $B_{ia}^P = V_{ia}$ (all i, $a_{start}^{n_w} \le a \le a_{end}^{n_w}$)

End P Loop

Figure 1. Pseudocode of the parallel algorithm for computing the **B** matrix of ERIs $(ia|P)$ with the GPW approach.

$$
\begin{aligned}
(\mu\nu|P) &= \sum_R (\mu\nu|R)L_{PR}^{-1} \\
&= \int\int \phi_\mu(\vec{r}_1)\phi_\nu(\vec{r}_1)\frac{1}{\vec{r}_{12}}\left[\sum_R \phi_R(\vec{r}_2)L_{PR}^{-1}\right]d\vec{r}_1 d\vec{r}_2 \\
&= \int \phi_\mu(\vec{r}_1)\phi_\nu(\vec{r}_1)\left[\int \frac{\rho^P(\vec{r}_2)}{\vec{r}_{12}}d\vec{r}_2\right]d\vec{r}_1 \\
&= \int \phi_\mu(\vec{r}_1)\phi_\nu(\vec{r}_1)v^P(\vec{r}_1)d\vec{r}_1
\end{aligned} \tag{29}
$$

The same approach holds for the $(P|Q)$ integrals with the only difference that the potential is calculated from the density associated with a single Gaussian auxiliary basis function.

Of central importance in GPW is then the representation of the density on a regular grid, which is equivalent to an expansion of the density in an auxiliary basis of plane waves (PW). The expansion is given by

$$
\rho^P(\vec{R}) \approx \frac{1}{\Omega}\sum_{|\vec{G}|\le G_c} \rho^P(\vec{G})e^{i\vec{G}\cdot\vec{r}} \tag{30}
$$

where the sum over the reciprocal lattice vectors $\vec{G}$ is determined by the resolution of the grid. $\rho^P(\vec{G})$ are the Fourier coefficients of the density, and Ω is the volume of the simulation cell. Conventionally, the resolution of the grid is specified as the energy cutoff $1/2G_c^2$ that limits the kinetic energy of the PWs. Fast Fourier transforms (FFTs) efficiently change representation between real space $(\rho^P(\vec{R}))$ and reciprocal space $(\rho^P(\vec{G}))$. In particular, for a grid with S grid points, the transformation can be performed in linear scaling time $(O(S \log S))$. In reciprocal space, it becomes straightforward to solve the Poisson equation for the potential v^P

$$
v^P(\vec{G}) = \frac{4\pi}{G^2}\rho^P(\vec{G}) \tag{31}
$$

and an additional back FFT (FFT^{-1}) will yield the potential in real space. For $\vec{G} = 0$ the value of the potential is set to zero; this corresponds to a constant shift to the potential (or redefinition of the vacuum level).[91] Thanks to the orthogonality of the occupied-virtual orbitals this shift has no influence on the final value of the $(ia|P)$ integrals.[92]

Once the potential v^P is available on a regular real space grid, the numerical integration over the basis functions is performed by summing the product of the value of the potential and the primitive Gaussian functions (PGFs) over the grid points. For a given $|P)$, all matrix elements that are nonzero within a given threshold (ε_{grid}) can be obtained in linear scaling time. A further gain in efficiency is obtained by employing a multigrid technique that represents the potential v^P on grids with increasingly coarser grid spacing. The accuracy of the multigrid scheme is fixed by specifying a relative cutoff (E_{cut}^{rel}) that specifies the E_{cut} of the grid that will be employed for a primitive Gaussian function (PGF) with exponent 1.0.

Finally, $(\mu\nu|P)$ integrals are transformed into MO ERIs using (sparse) matrix multiplication. Introducing for a given $|P)$ vector the matrix of ERIs $\mathbf{S}^P$ $(S_{\mu\nu}^P = (\mu\nu|P))$, the ia elements of the matrix B_P^{ia} are obtained by two matrix multiplication as $\mathbf{C}_o^\dagger \mathbf{S}^P \mathbf{C}_v$, where $\mathbf{C}_o$ and $\mathbf{C}_v$ represent the coefficient matrices of

the occupied and virtual orbitals. The multiplication by $\mathbf{C}_o$ can exploit the sparsity of $\mathbf{S}^P$, implying an $O(no)$ scaling per $|P)$ vector, while the final multiplication cannot exploit sparsity and is asymptotically dominant, scaling as $O(onv)$. The thresholding in the sparse matrix multiplication is enforced using a threshold $\varepsilon_{filter} \approx \varepsilon_{grid}$.

As we have shown for the MP2-GPW method,[63] the overall accuracy of the MP2 energy can be well controlled and is on the order of $10^{-7}-10^{-8}$ au per heavy atom for E_{cut} = 300 Ry, E_{cut}^{rel} = 50 Ry, and $\varepsilon_{filter} = \varepsilon_{grid} = 10^{-8}$.

3. PARALLEL IMPLEMENTATION OF THE RI-GPW METHODS

In this section the algorithms and the parallelization strategies for the methods introduced previously are presented and discussed in detail. The algorithms are split in two steps: the first deals with the computation of the ERIs $(ia|P)$, Section 3.1, and is in common for all methods, and the second is specific to the type of correlation energy calculated (Sections 3.2, 3.3, 3.4).

3.1. GPW Calculation of the $(ia|P)$ ERIs. The pseudocode for the parallel algorithm for computing the $(ia|P)$ ERIs with the GPW approach is presented in Figure 1; while its main features are summarized in Table 1, the rest of the section discusses this figure in detail.

Table 1. Features of the Parallel Algorithm for Computing the B Matrix of ERIs $(ia|P)$ with the GPW Approach Expressed As "Order Of" the Calculation Parameters[a]

	memory	execution time
Generation of $\mathbf{M}^P$ Vectors		
ρ^P calculation	S/N_w	N_aS/N_p
FFT and Poisson solver	S/N_w	$N_a\mathrm{Slog}(S)/N_p$
I_Q^P calculation (v^P integration)	N_a^2/N_G	N_a^2/N_p
I_Q^P redistribution into $(P\|Q)$	N_a^2/N_G	N_a^2
Cholesky decomposition of $(P\|Q)$	N_a^2/N_p	N_a^3/N_p
generation of $\mathbf{L}^{-1}$	N_a^2/N_p	N_a^3/N_p
collect $\mathbf{M}^P$ vectors	N_a^2/N_G	N_a^2
Generation of $(ia\|P)$ Integrals ($\mathbf{B}$ Matrix)		
ρ^P calculation	S/N_w	N_aS/N_p
FFT and Poisson solver	S/N_w	$N_aS\log(S)/N_p$
$S_{\mu\nu}^P$ calculation (v^P integration)	$(S+n)/N_w$	N_an/N_p
1st index transformation	on/N_w	N_aon/N_p
2nd index transformation	ov/N_w	N_aovn/N_p
redistribution and storage $\mathbf{B}$	ovN_a/N_p	N_aov/N_G

[a] n and N_a are the number of primary and auxiliary basis functions, o and v are the number of occupied and virtual orbitals, S is the grid size, N_G and N_w are the number of groups and group size, N_p is the number of processes. N_G, N_w, and N_p are related by $N_p = N_GN_w$. The notation for the individual step is taken from the algorithm in Figure 1.

The presented algorithm follows closely the integral computation part of our MP2-GPW method[63] with the main difference that all computations that were based on ia occupied-virtual pairs now are performed for auxiliary basis related quantities. The parallelization is achieved with a multilevel hybrid OpenMP/MPI scheme and a careful process layout. The first level of parallelization corresponds to distributing the work performed for a single given auxiliary basis function ϕ_P or vector $|P) = \Sigma_R\phi_RL_{PR}^{-1}$. The second level of parallelization corresponds to a distribution of these nearly independent calculations. The N_p processes available in total are therefore split in N_G groups, each group working on a given ϕ_P or $|P)$ and each consisting of N_w processes ($N_p = N_GN_w$). The first level of parallelization is complicated, involving parallel FFTs, halo-exchanges, and sparse matrix multiplications over N_w processes. However, this level is readily available, as it corresponds to the standard parallelization scheme for DFT calculations in CP2K.[90] The second level of parallelization is more straightforward, since it only requires a few steps of intergroup redistribution of two center ERIs $(Q|P)$ in order to calculate $\mathbf{L}^{-1}$.

The total work load for the integral computation is distributed by splitting the total number N_a of auxiliary basis function into N_G ranges $[P_{start}^{n_p}, P_{end}^{n_p}]$, each of them labeled with a given n_P coordinate and assigned to the corresponding group. Additionally, each of the N_w processes within a group is given an index n_w, so that a process is uniquely identified by its coordinate (n_P, n_w). Finally, the a virtual index is split in N_w ranges $[a_{start}^{n_w}, a_{end}^{n_w}]$, while a splitting of the occupied i is not necessary.

The algorithm is split into two parts, the first related to the calculation of $\mathbf{L}^{-1}$ the second to the computation of the $\mathbf{B}$ matrix of $(ia|P)$. In the two steps there are common features; in fact, in both cases, the basic structure of the GPW machinery can be recognized, consisting of a density $\rho^P(\vec{R})$ calculation on the real space grid, the computation of the associated potential $v(\vec{R})$, and the final integration of $v(\vec{R})$ over Gaussian elements of the basis. The difference in the two cases is that for the computation of $\mathbf{L}^{-1}$, $\rho(\vec{R})$ is related to a single auxiliary basis function and the associated potential is integrand over the auxiliary basis set functions, while for the computation of the $(ia|P)$ ERIs $\rho(\vec{R}) = \Sigma_R\phi_RL_{PR}^{-1}$ and the integration is performed over pair elements of the primary basis $(\mu\nu)$.

At the end of the first loop over auxiliary basis elements, each group stores a slice of the $(Q|P)$ matrix (labeled as I_Q^P in the pseudocode in Figure 1) comprised all Q and $P \in [P_{start}^{n_p}, P_{end}^{n_p}]$. In order to perform efficiently the Cholesky decomposition $(Q|P) = \mathbf{L}\mathbf{L}^T$ and subsequently the $\mathbf{L}$ inversion, the I_Q^P are redistributed into a 2D layout of a parallel distributed matrix as used in scalapack. Once the $\mathbf{L}^{-1}$ matrix is computed a similar redistribution procedure is again performed in order to collect the rows of $\mathbf{L}^{-1}$ ($M_R^P = L_{PR}^{-1}$, all R, $P \in [P_{start}^{n_p}, P_{end}^{n_p}]$) necessary for the next step. The time spent for these two redistribution steps is not scaling well with the number of processes; nevertheless, the communication effort grows only $O(N^2)$, and the associated time, for the benchmark calculations performed so far, is negligibly small.

The second loop over auxiliary basis vector $|P)$ allows for the final computation of the $(ia|P)$ integrals. The time for the calculation of the $\mathbf{S}^P$ matrix is linear scaling with the system size for a given P, since only pairs of overlapping Gaussians $(\mu\nu|$ need to be considered, and only a finite number of grid points within a spherical region around the center of the PGF is required. This implies that the total time for the integration of the potential in the RI-GPW method is $O(N^2)$, which is a great reduction compared to the canonical GPW-MP2 algorithm, where the corresponding task is $O(N^3)$.

Moreover, another advantage of the RI-GPW method, compared to standard RI integral implementation, is that only two integral transformations from AO to MO basis are required. The third transformation, which in conventional RI-MP2 implementation is performed as $(ia|P) = \Sigma_R(ia|R)L_{PR}^{-1}$ and requires $O(ovN_a^2)$ operations, is no longer needed.

dx.doi.org/10.1021/ct4002202 | J. Chem. Theory Comput. 2013, 9, 2654−2671

Check available memory [Mem]
According to [Mem] define the replication group R of size N_r
According to [Mem] define the batch size B_S
Split the N_G groups into N_R subgroups ($N_R = N_G/N_r$)
Assign to each group G its coordinate in the subgroup n_R
Collect B^P_{ia} from all other processes with same coordinate (n_R, n_w)
Distribute IJ batches over the N_G groups ($I \leq J$, batch size of I and J given by B_S)
Loop over IJ batches ($IJ \in$ my_G^{IJ})
  Collect $A^i_{aP} = B^P_{ia}$ and $E^j_{aP} = B^P_{ja}$, ($i \in I$, $j \in J$, all P, $a \in [a^{n_w}_{start}, a^{n_w}_{end}]$)
    from all other processes in R with my same n_w coordinate
  Loop over ij ($i \in I$, $j \in J$)
    $(ia|jb) = I_{ab} = \sum_P A^i_{aP} E^j_{bP}$ ($a,b \in [a^{n_w}_{start}, a^{n_w}_{end}]$)
    Loop over n'_w, $n'_w \in G$
      Collect E^j_{bP}, ($b \in [a^{n'_w}_{start}, a^{n'_w}_{end}]$)
      $(ia|jb) = I_{ab} = \sum_P A^i_{aP} E^j_{bP}$, ($a \in [a^{n_w}_{start}, a^{n_w}_{end}], b \in [a^{n'_w}_{start}, a^{n'_w}_{end}]$)
    End n'_w Loop
    $E^{(2)} = E^{(2)} - (2 - \delta_{ij}) \sum_{ab} I_{ab}(2I_{ab} - I_{ba})/(\Delta^{ab}_{ij})$, ($a,b \in [a^{n_w}_{start}, a^{n_w}_{end}]$)
    Loop over n'_w, $n'_w \in G$
      Collect I'_{ba}, ($b \in [a^{n_w}_{start}, a^{n_w}_{end}], a \in [a^{n'_w}_{start}, a^{n'_w}_{end}]$)
      $E^{(2)} = E^{(2)} - (2 - \delta_{ij}) \sum_{ab} I_{ab}(2I_{ab} - I'_{ba})/(\Delta^{ab}_{ij})$, ($a \in [a^{n_w}_{start}, a^{n_w}_{end}], b \in [a^{n'_w}_{start}, a^{n'_w}_{end}]$)
    End n'_w Loop
  End ij Loop
End IJ Loop
Global summation of $E^{(2)}$

Figure 2. Pseudocode of the parallel algorithm for computing the RI-MP2 energy from the ERIs $(ia|P)$.

Table 2. Features of the Parallel Algorithm for Computing the Correlation Energy with the Different Methods Expressed As "Order Of" the Calculation Parameters[a]

	memory	execution time
	RI-MP2 Algorithm (Section 3.2)	
replication of $(ia\|K)$ into R groups	ovN_a/N_wN_r	$(\log_2(N_R)/N_G)(ovN_a/N_w)$
$(ia\|jb)$ integral generation	$2B_SvN_a/N_w$	$o^2v^2N_a/N_p$
MP2 energy accumulation	v^2/N_w	o^2v^2/N_p
communication	$2B_SvN_a/N_w$	o^2vN_a/N_pB_S
	RI-dRPA Algorithm (Section 3.3)	
creation of $\mathbf{B}$ matrix	ovN_a/N_wN_r	$[(\log_2(N_R) + (N_wN_r - 1)/N_p)][(ovN_a)/(N_rN_w)]$
calculation of $\mathbf{B}'(\omega) = \mathbf{G}(\omega)\mathbf{B}$	ovN_a/N_wN_r	N_qovN_a/N_p
calculation of $\mathbf{Q}(\omega) = 2\mathbf{B}^T\mathbf{B}'(\omega)$	ovN_a/N_wN_r	$N_qovN^2_a/N_p$
calculation of $\mathrm{Tr}[\ln(\mathbf{Q}(\mathbf{w}) + \mathbf{1}) - \mathbf{Q}(\mathbf{w})]$	N^2_a/N_wN_r	$N_qN^3_a/N_p$
	Laplace-RI-SOS-MP2 Algorithm (Section 3.4)	
calculation of $\mathbf{B}'$	ovN_a/N_wN_r	N_qovN_a/N_p
calculation of $\bar{\mathbf{Q}} = \mathbf{B}^T\mathbf{B}'$	ovN_a/N_wN_r	$N_qovN^2_a/N_p$
calculation of $\mathrm{Tr}[\bar{\mathbf{Q}}\,\bar{\mathbf{Q}}^T]$	N^2_a/N_wN_r	$N_qN^2_a/N_p$

[a]The meaning of the different symbols is referred to in Table 1 with the only addition of N_R and N_r that represent the number of replication/integration group and the size of the replication/integration group ($N_G = N_RN_r$), B_S that is the batch size for ij in RI-MP2, and N_q that is the number of quadrature points used for the numerical quadrature in Laplace-RI-SOS-MP2 and RI-dRPA. The notation of the individual step is referred to the different algorithms.

At the end of the RI-GPW integral algorithm each process stores the matrix elements B^P_{ia} for all i, $P \in [P^{n_P}_{start}, P^{n_P}_{end}]$, and $a \in [a^{n_w}_{start}, a^{n_w}_{end}]$.

3.2. RI-MP2 Method. Once the integrals $(ia|P)$ are available the calculation of the RI-MP2 energy, in a serial algorithm, is straightforward, since the only tasks are related to the $(ia|jb)$ integral generation, eq 5, and the energy accumulation, eq 8.

In a parallel algorithm the main complication is introduced by the distributed storage of the $(ia|P)$ integrals. In particular, due to the features of the of the RI-GPW algorithm, previously described, each process stores elements B^P_{ia} for all occupied i, $P \in [P^{n_P}_{start}, P^{n_P}_{end}]$, and $a \in [a^{n_w}_{start}, a^{n_w}_{end}]$. Since N_w is usually small compared to the total number of processes, the virtual index a is distributed over a small number of MPI tasks within the

group G, while the auxiliary index P is distributed over the large amount of N_G groups.

In our RI-MP2 parallel algorithm, the $(ia|jb)$ integral generation proceeds as follow: first the independent ij pairs ($i \leq j$) are statically distributed over the N_G groups, for each ij pair, the full range of the auxiliary index P is collected on a local buffer from all other groups, while keeping the virtual index distribution within the group, finally the $(ia|jb)$ integrals are generated for the actual ij pair in a matrix-multiplication fashion (eq 5) requiring only a small amount of communication within the group. Once the $(ia|jb)$ are available, they are accumulated into the MP2 energy according to eq 8, requiring an additional negligible amount of communication within the group.

With this choice, the main source of intergroup communication in the parallel algorithm is related to the redistribution of the B_{ia}^P integrals, required for each ij pair. In order to perform this task efficiently, so that the required communication is reduced increasing the number of processes, a scheme involving replication of the B_{ia}^P into subgroup and batch communication for the ij pairs has been designed. Both replication and batching allow to improve the efficiency in communication since, in the first case, the number of processes that have to communicate with each other is reduced, in the latter because the number of messages is reduced. The available memory is thus used to reduce communication.

The pseudocode of the RI-MP2 energy calculation algorithm is sketched in Figure 2, and the main features of the algorithm are summarized in Table 2. In a first stage, according to the available memory the replication group size N_r is defined and integrals are replicated among processes. As shown in Table 2 the time involved in this step increases linearly with the logarithm of the number of replication group N_R, while it decreases when the number of processes ($N_G N_w$) is increased. We have observed that a ratio N_R/N_G of ∼0.1 is usually a good compromise between the time necessary for the replication and the gain in communication in the subsequent phase.

Once the R subgroups have been created and the B_{ia}^P integrals replicated, the maximum possible batch size B_S is defined based on the available memory per process. The total number of IJ batches ($I \leq J$) are then distributed statically over the groups. For load balancing reasons, the number of IJ batches is restricted to be a multiple of the number of groups N_G, and the remaining ij single pairs are again statically distributed over groups.

At this point, each group loops over its assigned IJ batches and collects the $A_{aP}^i = B_{ia}^P$ and $E_{aP}^j = B_{ja}^P$ integrals from the other members of the replication group R. The index ranges are $i \in I$, $j \in J$, all P, $a \in [a_{start}^{n_w}, a_{end}^{n_w}]$. This means that if $N_w = 1$, then the index a spans the full range of virtual orbitals and all the subsequent operations are performed locally. If $N_w > 1$, then the E_{aP}^j integrals have to be exchanged in a parallel matrix-multiplication-like fashion for the generation of the $(ia|jb)$ integrals that are then contracted into the MP2 energy, requiring an additional amount of in-group communication. As stated before, the in-group communication, which involves few processes, is usually negligible compared to the total communication time, and it is thus not taken into account in Table 2.

The total number of times the communication routine is invoked is

$$O\left(\frac{o^2}{B_S^2 N_G}\right)$$

while the time required for each event of communication (considered to be proportional to the message sizes) is

$$O\left(\frac{v N_a B_S}{N_w}\right)$$

This makes the total time spent in communication

$$O\left(\frac{o^2 v N_a}{N_p B_S}\right)$$

that is, communication is reduced when increasing the number of processes $N_p = N_w N_G$ and increasing the batch size B_S.

In the presented algorithm, the time determining step is the $(ia|jb)$ integral generation that is essentially a local matrix−matrix multiplication. This allows to fully exploit the performance of highly optimized routines, such as DGEMM. Moreover, this step can be further accelerated by employing a hybrid implementation that utilizes graphics processing units (GPUs).

3.3. RI-dRPA Method. The presented dRPA correlation energy implementation is based on the method developed by Eshuis et al.[49] In this approach, the calculation of the integral in eq 27 is accomplished by Clenshaw-Curtis numerical quadrature.[88] Here, only the main features of such a method are reported referring to the original paper for more details.

Given N_q quadrature points, the working expressions for the numerical quadrature of eq 27 are

$$F^C(\omega) = \frac{1}{2}\mathrm{Tr}[\ln(\mathbf{Q}(\omega) + \mathbf{1}) - \mathbf{Q}(\omega)] \tag{32}$$

$$E_c^{\text{RI-dRPA}} \simeq \sum_{q=1}^{N_q} \frac{a \cdot w_q}{2\pi} F^C(a \cdot \cot(t_q)) \tag{33}$$

$$t_q = \frac{q}{N_q}\frac{\pi}{2}, \quad q = 1, \ldots, N_q \tag{34}$$

$$w_q = \begin{cases} \dfrac{\pi}{N_q \sin^2 t_q}, & q = 1, \ldots, N_q - 1 \\ \dfrac{\pi}{2N_q \sin^2 t_q}, & q = N_q \end{cases} \tag{35}$$

where a is a scaling parameter that ensures that the grid points are adaptively distributed over the spectrum of eigenvalues of matrix $\mathbf{M}$ (eqs 24 and 25). Following ref 49, the calculation of a can be performed with $O(N^2)$ computational cost; in our algorithm the bisection method is used instead of Newton−Raphson.

The pseudocode of the parallel algorithm for the calculation of $E_c^{\text{RI-dRPA}}$ is reported in Figure 3. The parallel algorithm is based on a two level work load distribution. The first level corresponds to the distribution of the work necessary for a given quadrature point q, the second to the distribution of the independent quadrature points over subgroups of processes (the integration groups R).

The second level of parallelization is straightforward; first, according to the available memory, the minimum size $N_r^{\min}$ for the integration group is defined. Then, the actual size N_r is obtained such that $N_r \geq N_r^{\min}$, and the number of integration

Check available memory [Mem]

According to [Mem] define the minimum size $N_r^{\min}$ of the integration group R

Define the size N_r and number $N_R = N_G/N_r$ of integration group, such that:

$N_r \geq N_r^{\min}$ and N_R is a divisor of the number of quadrature points N_q

Split the N_G groups into N_R subgroups

Assign to each integration group R its subset of quadrature points $\{q^{\text{my}}\}$

Create the matrix $\mathbf{B}$ of size $ov \times N_a$ distributed within the integration group R

Collect B_{ia}^P integrals from all other processes and fill $\mathbf{B}$ matrix

According to N_q calculate Clenshaw-Curtis weight $\{w_q\}$ and abscissa $\{t_q\}$

Calculate the scaling parameter a

Loop over q quadrature points ($q \in \{q^{\text{my}}\}$)

Calculate $\omega = a \cdot \cot(t_q)$

Calculate $\mathbf{B}'(\omega) = \mathbf{G}(\omega)\mathbf{B}$

Calculate $\mathbf{Q}(\omega) = 2\mathbf{B}^{\mathbf{T}}\mathbf{B}'(\omega)$

Calculate $F^C(\omega) = \frac{1}{2}\text{Tr}[\ln(\mathbf{Q}(\omega)+\mathbf{1}) - \mathbf{Q}(\omega)]$

$E_c^{\text{RI-dRPA}} = E_c^{\text{RI-dRPA}} + F^C(\omega) \cdot a \cdot w_q/(2\pi)$

End q Loop

Global summation of $E_c^{\text{RI-dRPA}}$ across the integration groups

Figure 3. Pseudocode of the parallel algorithm for computing the RI-dRPA energy from the ERIs $(ia|P)$ and Clenshaw-Curtis numerical quadrature of eq 27.

group $N_R = N_G/N_r$ is an exact divisor of the total number of quadrature points N_q. In this way, each integration group R has an identically sized N_q/N_R subset of quadrature points $\{q^{\text{my}}\}$.

Once N_r is defined, the $\mathbf{B}$ matrix, defined in scalapack format, has to be made available within the integration group R. This is accomplished in two steps. In the first step, the parallel distributed matrix $\mathbf{B}$, of size $ov \times N_a$, is allocated over the members of R, each process in R is identified with its coordinate n_r. Subsequently, the locally held data (the $(ia|P) = B_{ia}^P$ distributed as described in Section 3.1) are redistributed within R according to new ranges of $\mathbf{B}$. This step consists of in-group communication and requires

$$O\left(\frac{(N_wN_r - 1)}{N_wN_r}\frac{ovN_a}{N_p}\right)$$

effort. The second step consists of intergroup communication, requiring replication of data across the integration groups. Since the integration groups have all the same size (N_wN_r) and the matrix $\mathbf{B}$ is created in the same manner, it means that $\mathbf{B}$ retains the same structure for all groups, that is, all processes with the same coordinate n_r (belonging to different groups) have the same ranges for the rows and columns. Consequently, in the second step, only processes that have the same coordinate n_r communicate, replicating their local data across the integrations groups. This requires a communication effort of

$$O\left(\frac{\log_2(N_R)}{N_wN_r}ovN_a\right)$$

giving the total cost reported in Table 2.

After the $\mathbf{B}$ matrix is created, the algorithm proceeds independently for each integration group. As a first task the matrix $\mathbf{B}'$ is calculated as $\mathbf{G}(\omega)\mathbf{B}$. Since $\mathbf{G}(\omega)$ is a $ov \times ov$ diagonal matrix with elements $G_{ia,ia}(\omega) = (\varepsilon_a - \varepsilon_i)((\varepsilon_a - \varepsilon_i)^2 + \omega^2)^{-1}$ the calculation of $\mathbf{B}'$ requires only

$$O\left(\frac{ovN_a}{N_wN_r}\right)$$

operations for each quadrature point without interprocess communication.

The time determining step of the algorithm is the calculation of the $N_a \times N_a$ matrix $\mathbf{Q}(\omega)$ computed as $2\mathbf{B}^{\mathbf{T}}\mathbf{B}'(\omega)$. This task is performed as a standard parallel matrix−matrix multiplication using scalapack. Consequently the computational effort per quadrature point is $O((ovN_a^2)/(N_wN_r))$, while the communication is expected to scale as $O(1/(N_wN_r)^{1/2})$.

The last computationally demanding task is the calculation of $\text{Tr}[\ln(\mathbf{Q}(\omega) + \mathbf{1})]$. This step can be efficiently carried out by considering the identity $\text{Tr}[\ln \mathbf{A}] = \ln(\text{Det}[\mathbf{A}])$, that is

$$\text{Tr}[\ln(\mathbf{Q}(\omega) + 1)] = 2\sum_{i=1}^{N_a}\ln(U_{ii}) \tag{36}$$

where the $\mathbf{U}$ matrix is the Cholesky decomposition of $\mathbf{Q}(\omega)+\mathbf{1}$.

3.4. Laplace-RI-SOS-MP2 Method. The Laplace-RI-SOS-MP2 algorithm is closely related to the RI-dRPA algorithm; in fact in both cases a numerical quadrature is required together with an $O(N^4)$ matrix−matrix multiplication step.

Contrary to the RI-dRPA case, optimal integration grids for the numerical quadrature in the Laplace-transform MP2 method can be obtained relatively easily. In fact, the energy denominator Δ_{ij}^{ab} depends only on the occupied and virtual orbital energies such that $\Delta_{ij}^{ab} \in [E_{\min}, E_{\max}]$, where $E_{\min} = 2(\varepsilon_{\text{LUMO}} - \varepsilon_{\text{HOMO}})$ and $E_{\max} = 2(\varepsilon_{\max} - \varepsilon_{\min})$ being that $\varepsilon_{\max}$ and $\varepsilon_{\min}$ are the maximum and minimum values of the orbital energies, respectively. This implies that, in order to reach reasonable accuracy (10^{-5}−10^{-6} Hartree) in the computation of the $E_{OS}^{(2)}$, in general, fewer integration points are required compared to the RI-dRPA case. Different quadrature techniques such as Gauss-Laguerre, Gauss-Legendre,[93] and least-squares (LS) quadrature[77] have been employed. In our implementation, the minimax approximation[84,85] has been adopted. The minimax approach has some appealing features such as uniformity of the error along the whole range and error of alternating sign, leading to possible error cancellation. It has been shown that the minimax method in the Laplace-transform MP2 framework remarkably outperforms classical quadrature techniques, while giving comparable accuracy compared to the LS approach.[94]

As shown in the pseudocode given in Figure 4, the algorithm for computing $E_{OS}^{(2)}$ proceeds in the same way as in the RI-dRPA case till the generation of the $\mathbf{B}$ matrix. When the $\mathbf{B}$ matrix has been replicated and redistributed within each integration group, the minimax parameters $\{w_q^*\}$ and $\{t_q^*\}$ are determined for the range $\Delta_R = E_{\max}/E_{\min}$, that is equivalent to find the minimax approximation of the $1/x$ function for $x \in [1,\Delta_R]$. These parameters are then scaled by $E_{\min}$, that is, the range of the approximation is shifted to $[E_{\min},E_{\max}]$.

Each integration group R will perform all the tasks for its preassigned quadrature points $\{q^{\text{my}}\}$ in parallel within the members of R. The required tasks for a given quadrature point consist of updating the $\mathbf{B}'$ matrix with the actual weight w_q and abscissa t_q, perform the matrix−matrix multiplication $\bar{\mathbf{Q}} = \mathbf{B}^{\mathbf{T}}\mathbf{B}'$, and increment the $E_{OS}^{(2)}$ with the q^{th} contribution given by $\text{Tr}[\bar{\mathbf{Q}}\,\bar{\mathbf{Q}}^{\mathbf{T}}]$. The computational effort required for each of these individual steps is reported in Table 2. While the update of $\mathbf{B}'$ and the $O(N^4)$ matrix multiplication are analogous to the corresponding steps in the RI-dRPA algorithm, the calculation

Check available memory [Mem]

According to [Mem] define the minimum size $N_r^{\min}$ of the integration group R

Define the size N_r and number $N_R = N_G/N_r$ of integration group, such that:

$N_r \geq N_r^{\min}$ and N_R is a divisor of the number of quadrature points N_q

Split the N_G groups into N_R subgroups

Assign to each integration group R its subset of quadrature points $\{q^{my}\}$

Create the matrix $\mathbf{B}$ of size $ov \times N_a$ distributed within the integration group R

Collect B_{ia}^P integrals from all other processes and fill $\mathbf{B}$ matrix

Calculate $E_{\min} = 2(\varepsilon_{\mathrm{LUMO}} - \varepsilon_{\mathrm{HOMO}})$ and $E_{\max} = 2(\varepsilon_{\max} - \varepsilon_{\min})$

Calculate $\Delta_R = E_{\max}/E_{\min}$

According to N_q and Δ_R calculate minimax weight $\{w_q^*\}$ and abscissa $\{t_q^*\}$

Scale minimax parameters $w_q = w_q^*/E_{\min}$, $t_q = t_q^*/E_{\min}$

Loop over q quadrature points ($q \in \{q^{my}\}$)

  Loop over local columns P

    Loop over local rows ia

      $B_{ia}^{P\prime} = B_{ia}^P \cdot \sqrt{w_q} \cdot \exp(t_q(\varepsilon_i - \varepsilon_a))$

    End ia Loop

  End P Loop

  Calculate $\overline{\mathbf{Q}} = \mathbf{B}^{\mathbf{T}}\mathbf{B}'$

  $E_{OS}^{(2)} = E_{OS}^{(2)} - \mathrm{Tr}[\overline{\mathbf{Q}}\,\overline{\mathbf{Q}}^{\mathbf{T}}]$

End q Loop

Global summation of $E_{OS}^{(2)}$ across the integration groups

Figure 4. Pseudocode of the parallel algorithm for computing the Laplace-RI-SOS-MP2 energy from the ERIs ($ia|P$) and the minimax approximation for the numerical quadrature in eq 18.

of $\mathrm{Tr}[\overline{\mathbf{Q}}\ \overline{\mathbf{Q}}^{\mathrm{T}}]$ is only $O(N^2)$ and is performed without interprocesses communication.

3.5. Hybrid CPU/GPU Implementations. For the methods presented in this paper, the time determining step, from a computational complexity point of view, is a matrix−matrix multiplication. In the case of the parallel RI-MP2 algorithm it is a process-local matrix multiplication (DGEMM), while in the other two cases it is a parallel matrix multiplication (PDGEMM).

New accelerator hardware, such as graphics processing units (GPUs), can perform these operations efficiently, with a performance exceeding that of several traditional CPU codes both in time to solution as well as energy efficiency.

In the current GPU implementation, only the matrix multiplication step is performed on the GPU. This implies that the impact is limited to sufficiently large systems, where this part is dominating.

4. BENCHMARK CALCULATIONS

4.1. Computational Details. *The EXX/RPA Formalism.* The exact exchange (EXX) and random phase approximation correlation energies formalism (EXX/RPA) has been extensively applied to a large variety of systems including isolated molecules,[5−8,10,37,50,95−97] solids,[98−102] surfaces,[100,103−106] and van der Waals bonded crystals.[107−109]

Within the framework of EXX/RPA formalism the total energy is given as

$$E_{\mathrm{tot}}^{\mathrm{EXX/RPA}} = E_{\mathrm{tot}}^{\mathrm{HF}} + E_{\mathrm{c}}^{\mathrm{RPA}} = (E_{\mathrm{tot}}^{\mathrm{DFT}} - E_{\mathrm{xc}}^{\mathrm{DFT}}) + E_{\mathrm{x}}^{\mathrm{EXX}} + E_{\mathrm{c}}^{\mathrm{RPA}} \quad (37)$$

where the right-hand side terms of the last equation are the DFT total energy, the exchange-correlation DFT energy, the exact exchange, and the RPA correlation energy, respectively. The sum of the first three terms is referred to as the Hartree−Fock (HF) energy calculated employing the DFT orbitals, and in the following will be denoted simply as HF@DFT. The last term corresponds to the RPA correlation energy computed using the DFT orbitals and orbital energies and will be referred to as RPA@DFT.

The calculation of the $E_{\mathrm{tot}}^{\mathrm{EXX/RPA}}$ for a given system is thus performed by first converging the self-consistent field (SCF) procedure with a given DFT method. Then the ground state single-particle wave functions and orbital energies are used as input to compute the EXX energy and the RPA correlation energy.

All EXX/RPA calculations reported in this work have been performed employing Kohn−Sham Perdew-Burke-Ernzerhof (PBE)[110] orbitals as input unless otherwise stated.

Basis Sets, Thresholds, and Pseudopotentials. The RI-GPW methods as implemented in CP2K[78] have been employed for all calculations in this manuscript. The correlation energy calculations are based on pseudopotentials of the form suggested by Goedecker, Teter, and Hutter (GTH) in ref 111 but specifically parametrized for Hartree−Fock (HF) and DFT calculations. In this way, core states do not need to be represented and valence orbitals are smooth, as required by the GPW method. The same basis used in our previous GPW-MP2 work has been adopted.[63] They consist in valence-only correlation consistent type[112,113] basis sets, generated for being used with these pseudopotentials. The basis sets have been labeled as cc-DZVP, cc-TZVP, and cc-QZVP, denoting double, triple, and quadruple-ζ quality, respectively. For each primary basis set, the relative auxiliary RI basis has been generated according to the procedure proposed by Weigend et al.[41] Using uncontracted Gaussian functions only, the size of the auxiliary basis is typically between 2.5 and 3 times larger than the corresponding primary basis and includes functions with angular momentum quantum number up to (l_{max} + 1), where l_{max} is the maximum l−quantum number of the primary basis. For example, in the case of oxygen, the auxiliary basis associated with the cc-TZVP basis is built with the following pattern of primitives (6s 5p 5d 3f 1g). The condition number of the ($P|Q$) matrix has been computed employing both normalized and unnormalized auxiliary basis functions. In the case of molecular crystals described with the cc-TZVP basis, the largest observed condition number is of the order of 10^8 and 10^{10} for the normalized and unnormalized case, respectively. In the case of the LiH crystal, considering the 3 × 3 × 3 cell, employing the cc-QZVP basis, the condition number is of the order of 10^{10} for both the normalized and unnormalized case. These different normalizations have shown not to give discrepancies for the computed cohesive energies.

The Hartee-Fock exchange calculations have been performed employing our robust Γ-point implementation[114,115] that allows for stable calculations in the condensed phase.[115,116] The Schwarz screening threshold for the HF calculations is in the range 10^{-8}−10^{-10}, periodic calculations used a truncated Coulomb operator,[115] using approximately half the length of the smallest edge (L) of the simulation cell as truncation radius (R_c), typically 5−6 Å. We have verified that the effect on the MP2 energy due to the truncation is usually 1 order of magnitude smaller than the effect on the Hartree−Fock energy. The latter has been shown to converge exponentially,[115] and for

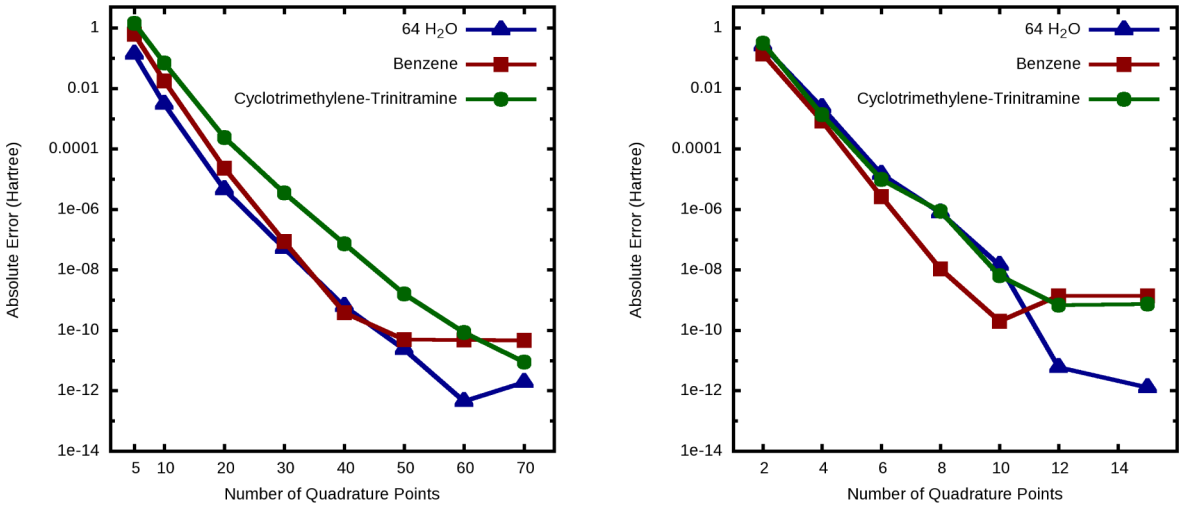


Figure 5. Accuracy of the numerical quadrature for the RI-dRPA (a) and RI-SOS-MP2 (b) methods with respect to the number of quadrature points for three different systems. The reference energy is, in the case of RI-dRPA, the energy obtained employing 100 quadrature points, in the case of RI-SOS-MP2, the exact spin-opposite component of the RI-MP2 energy.

the benzene crystal, the error in the cohesive energy is 1 kJ/mol and 0.2 kJ/mol for an R_c of 5 Å and 7 Å, respectively. Note that, as we discussed in ref 63, the MP2 energy computed at the Γ-point displays a slower convergent (L^{-3}) size-effect.

The threshold for the SCF convergence was 10^{-6} or tighter. The PW cutoff for the HF or DFT part of the calculations was E_{cut} = 1200 Ry to guarantee convergence of the exchange-correlation term, at small cost compared to the post SCF correlation energy calculations. The HF@DFT calculations have been performed retaining the pseudopotential of the DFT method. The correlation energy calculations employed high quality PW cutoffs of E_{cut} = 300 Ry, E_{cut}^{rel} = 50 Ry, $\varepsilon_{filter} = 10^{-7}$, and $\varepsilon_{grid} = 10^{-6}$, unless mentioned otherwise. Gas phase systems have been computed using cluster boundary conditions for solving the Poisson equation.[66]

The HF and PBE pseudopotentials as well as all basis sets employed in this work have been reported in the Supporting Information. Moreover, in order to validate the reported settings, calculations for the S22 set[117] have been performed for the RI-MP2 and RI-dRPA methods. The tested basis sets are cc-TZVP, cc-QZVP, and a slightly modified cc-TZVP basis containing augmented functions (labeled as aug-cc-TZ). The obtained results for the S22 set, reported in the Supporting Information, show good agreement with the previously reported calculations for the basis of comparable quality;[34,117,118] for example, the mean absolute deviation, in the case of the aug-cc-TZ with respect to Dunning's augmented correlation consistent basis aug-cc-pVTZ, is 0.1 kcal/mol for both RI-MP2 and RI-dRPA.

Geometries and Cohesive Energies. For all crystals, supercells have been generated by replicating the unit cell, so that the smallest edge was larger than 9 Å, in order for the Γ-point approximation to be reasonable. The experimental geometries of the molecular crystals have been retrieved from the Cambridge Structural Database (CSD).[119] The positions of the hydrogen atoms of these geometries have been further relaxed at the DFT/BLYP[120,121] level employing the cc-TZVP basis set, unless otherwise stated. The main features of the structure of each crystal, together with the supercell used in the calculation and the CSD refcode, are reported in the Supporting Information or can be recovered from our previous work on the GPW-MP2 method.[63] The LiH crystal geometry is based on the experimental value of the lattice parameter (a = 4.084 Å) .

The counterpoise (CP) corrected cohesive energy per molecule at a given volume V has been computed as[122,123]

$$E_{coh}^{CP}(V) = \frac{E_{supercell}(V)}{N_{mol}} - E_{mol}^{gas} - E_{mol+ghost}^{crystal}(V) + E_{mol}^{crystal}(V) \quad (38)$$

Here, N_{mol} is the number of molecules per supercell, $E_{supercell}(V)$ is the total energy of the supercell, and $E_{mol+ghost}^{crystal}(V)$, $E_{mol}^{crystal}(V)$, and E_{mol}^{gas} are the total energy of an isolated molecule in either the crystal geometry ($E_{mol+ghost}^{crystal}(V)$ and $E_{mol}^{crystal}(V)$) or a gas phase geometry (E_{mol}^{gas}). $E_{mol+ghost}^{crystal}(V)$ includes ghost atoms from the molecules of first coordination shell, while the gas phase geometries have been obtained by relaxation at the B3LYP/cc-TZVP level.[121,124,125] To assess the accuracy of computed cohesive energies, these values have been compared to the experimental sublimation enthalpies ($\Delta H(s)$).

4.2. Accuracy of the Methods. In this section the accuracy of the presented methods is discussed. Three significant approximations can be identified as possible sources of error.

The lowest level approximation is related to the GPW method for which an auxiliary Plane-Waves basis is introduced in order to express the fitting densities in the reciprocal space. We have shown that the GPW approach for calculation the MP2 energy introduces negligible error, and the settings specified in 4.1 are sufficient to provide an error below 10^{-7} au per heavy atom.[63]

The second approximation refers to the RI methods, for which an auxiliary basis of localized Gaussians is introduced in order to factorize the two electron repulsion integrals. We have tested the accuracy of the RI approximation only in the MP2 energy case, since no reference dRPA energies have been calculated due to the high $O(N^6)$ computational cost. However,

dx.doi.org/10.1021/ct4002202 | J. Chem. Theory Comput. 2013, 9, 2654−2671

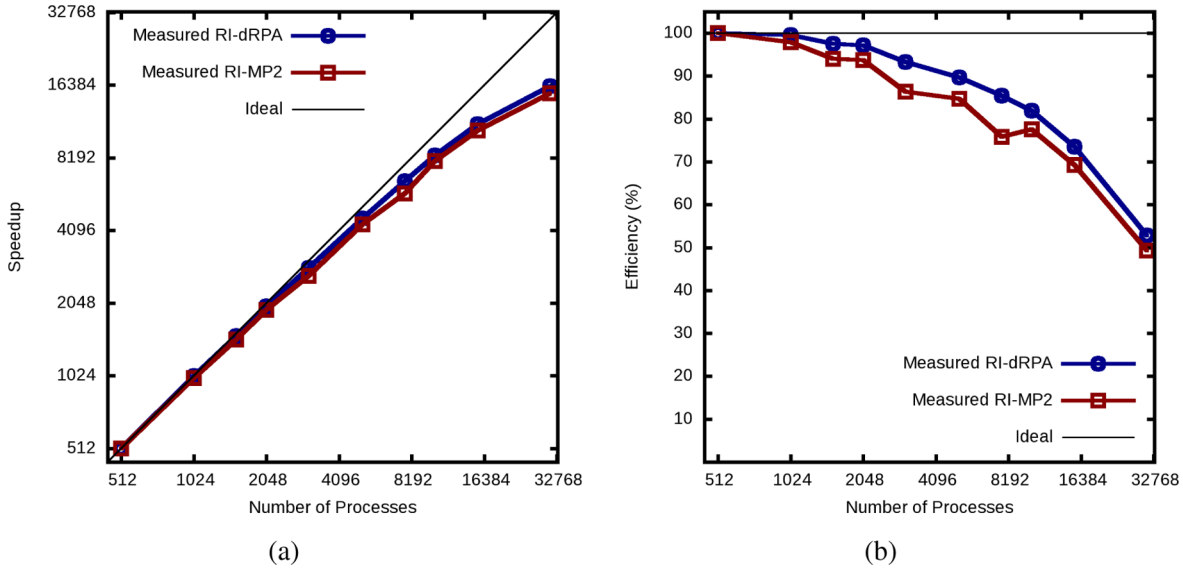


Figure 6. Measured speedup (a) and efficiency (b) with respect to 512 processes for the calculation of the RI-MP2 and RI-dRPA energy of 64 bulk water molecules. RI-dRPA calculation performed employing 60 quadrature points for the numerical integration.

Table 3. Benchmark Calculations for the RI-MP2, RI-dRPA, and RI-SOS-MP2 Methods[a]

	A	n	o	N_a	t^{RI}_{MP2}	t^{RI}_{dRPA}	$t^{RI}_{SOS\text{-}MP2}$	$t^{SCF}_{1^{st}cycle}$	t_{MP2}/t^{RI}_{MP2}
U	128	2752	192	6784	2.4	2.5	1.3	2.1	12.9
D	128	2992	192	7520	3.3	3.5	1.9	4.9	14.3
FA	120	2760	216	6912	2.8	2.8	1.5	2.0	12.1
64 H_2O	192	3648	256	8704	7.5	6.4	3.4	1.0	17.3
B	192	4128	240	10176	10.4	8.8	4.2	5.5	12.7
PD	144	3936	312	10208	13.7	10.3	4.5	6.9	10.4
SA	176	4144	304	10432	14.6	11.1	4.9	5.0	10.5
CT	168	4152	336	10560	17.2	12.3	5.2	4.5	9.8

[a]Time in min.@CRAY-XK7, 3200 cores. U = urea, B = benzene, FA = formic acid, SA = succinic anhydride, D = 2,3-diazanaphthalene, PD = pyromellitic dianhydride, CT = cyclotrimethylene-trinitramine, 64 H_2O = 64 bulk water molecules. A, n, o, and N_a represent the number of atoms, basis functions, occupied orbitals, and auxiliary basis functions, respectively. t^{RI}_{MP2}, t^{RI}_{dRPA}, $t^{RI}_{SOS\text{-}MP2}$, and $t^{SCF}_{1^{st}cycle}$ are the time necessary for the RI-MP2, RI-dRPA, and RI-SOS-MP2 energy calculations and for the first iteration in the Hartree-Fock self-consistent field (SCF) procedure. (t_{MP2}/t^{RI}_{MP2}) is the speedup of the RI method compared to the standard canonical MP2 energy algorithm. All RI-dRPA and RI-SOS-MP2 calculations were performed employing 40 and 10 quadrature points, respectively, for the numerical integration.

it has been shown that the magnitude of the error introduced by RI is of the same order for the MP2 and dRPA energies.[8,49] The largest deviation observed is related to the case of the benzene crystal for which the absolute error is 1.78 milli-Hartree corresponding to approximately 10^{-5} Hartree per atom.

The last possible source of error concerns only the RI-dRPA and RI-SOS-MP2 methods for which an integral quadrature scheme is required. Figure 5 shows the convergence of the absolute error with respect to the number of quadrature points for the two methods for three different condensed phase systems. In both cases, an exponential convergence is observed. For the Clenshaw-Curtis quadrature scheme, employed in the dRPA case, 20−30 quadrature points are enough for micro-Hartree accuracy. For the minimax approximation, used in the Laplace transformed SOS-MP2 method, the same accuracy is obtained with only 6−8 points.

4.3. Performance of the Methods. To assess the performance of the parallel algorithms a system made of 64 bulk water molecules has been chosen. The employed basis set is the cc-TZVP resulting in 256 occupied orbitals and 3648 primary and 8704 auxiliary basis functions. For the RI-dRPA case, 60 integration points have been used for the numerical quadrature.

The speedup and the parallel efficiency for the RI-MP2 and RI-dRPA methods are reported in Figure 6. The Laplace-RI-SOS-MP2 method has not been reported since it is closely related to the RI-dRPA case. Both algorithms show good parallel scalability in a wide range, resulting in an efficiency around 80% for the 10240 processes run. At the full scale-out (30720 processes) the time for computing the RI-MP2 and RI-dRPA energies is 97 and 109 s, respectively.

The main reason of the reduced efficiency in the RI-MP2 case is load imbalance, since the number of independent ij pairs starts to be of the same order as the number of processes for the largest runs. In the RI-dRPA case, the reduced efficiency is mainly related to the replication of the **B** matrix into each integration group R. Even if this step takes only 32 s for the full scale-out run it is significant for the total time.

In Table 3 the timings for different benchmark calculations performed with 3200 processes are reported. Together with the time spent for the computation of the correlation energy for the three methods, also the time spent in the first Hartree−Fock SCF cycle and the speedup of the RI-MP2 compared to the canonical MP2 are shown. The time for the first SCF cycle is

dx.doi.org/10.1021/ct4002202 | J. Chem. Theory Comput. 2013, 9, 2654−2671

Table 4. Benchmark Calculations for the Hybrid CPU/GPU Implementation of the RI-MP2 and RI-dRPA Methods[a]

	RI-MP2					RI-dRPA				
	t_{mul}	t^H_{mul}	t_{mul}/t^H_{mul}	t^H_{tot}	t_{tot}/t^H_{tot}	t_{mul}	t^H_{mul}	t_{mul}/t^H_{mul}	t^H_{tot}	t_{tot}/t^H_{tot}
U	1.37	0.40	3.4	1.49	1.6	1.39	0.32	4.3	1.39	1.8
D	1.80	0.61	3.0	2.10	1.6	1.87	0.44	4.2	2.01	1.7
FA	1.72	0.58	3.0	1.70	1.6	1.61	0.40	4.0	1.58	1.8
64 H_2O	5.00	1.47	3.4	4.19	1.8	4.03	0.91	4.4	3.20	2.0
B	7.14	1.93	3.7	5.16	2.0	5.86	1.36	4.3	4.20	2.1
PD	10.5	2.90	3.6	6.20	2.2	7.46	1.79	4.2	4.65	2.2
SA	11.2	2.62	4.3	6.14	2.4	7.93	1.84	4.3	4.95	2.2
CT	13.7	3.78	3.6	7.44	2.3	9.03	2.01	4.5	5.24	2.3

[a]Time in min.@CRAY-XK7, 3200 cores, 200 GPU. The meaning of the row entries is given in Table 3. In the table the superscript H refers to the hybrid CPU/GPU implementations; if no superscript is specified, the standard (only CPU) implementation is implied. In the case of RI-MP2 the subscript "mul" refers to the $O(N^5)$ step of the algorithm, that is the $(ialjb)_{RI}$ integrals generation (eq 5). In the case of RI-dRPA the subscript "mul" refers to the $O(N^4)$ step in the algorithm, that is the calculation of $\mathbf{Q}(\omega) = 2\mathbf{B}^T\mathbf{B}'(\omega)$ performed as a parallel matrix−matrix multiplication.

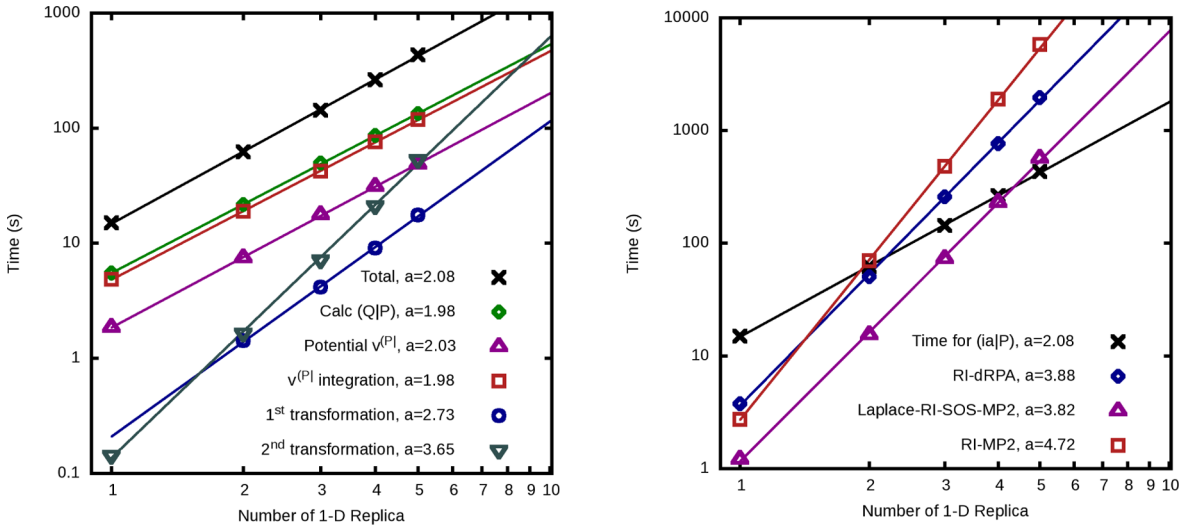


Figure 7. Shown is the time spent in the various significant parts of the energy calculation for the three methods, as a function of the number of replicas of the supercell, containing 32, 64, 96, 128, and 160 molecules of H_2O, respectively. The number of quadrature points employed is 20 and 6, respectively, for the RI-dRPA and RI-SOS-MP2 methods. (a) reports the timing for the $(ia|P)$ integral generation, that is in common for all methods (see Table 1), and (b) reports the total time for the $(ia|P)$ integral generation (black line) together with the timing for the most expensive step for each individual method (see Table 2). Lines represent a linear two-parameter fit of the form $y = bx^a$. The values of a for each task are reported in the legend.

strongly dominated by the calculation of the four-center electron repulsion integrals over atomic orbitals; this time is reported only to give an idea of the relative computational effort between the SCF procedure and the correlation energy calculation.

The calculation of the RI-SOS-MP2 energy is the fastest in all cases due to the combination of the lower $O(N^4)$ scaling and the relatively small number of quadrature points necessary. Moreover, in this case, the timing is of the same order as the first SCF step.

The RI-dRPA method is as expensive as the RI-MP2 method for the smaller systems but becomes cheaper than the latter for larger systems due to the better asymptotic scaling of $O(N^4)$.

Finally the RI-MP2 method shows a speedup compared to the canonical MP2 energy calculation that ranges from 9.8 up to a maximum of 17.3.

4.4. Performance of the Hybrid CPU/GPU Implementations. In order to assess the performance of the hybrid CPU/GPU implementations the same benchmark systems reported in Table 3 for the standard algorithms have been used. The results are shown in Table 4, where the data refer to the total execution time for computing the correlation energies and the time related only to the matrix−matrix multiplication step, labeled for both RI-MP2 and RI-dRPA as "mul". The superscript "H" denotes the case of the hybrid implementations, while the absence of superscript refers to the standard algorithms. The total execution timings in the case of the standard implementations can be found in Table 3.

Comparing the total timings for computing the correlation energies with t_{mul}, for the standard algorithms, it can be noted that this step requires between 50% and 80% of the total time. This sets the boundaries for the impact of the acceleration of this task on the speedup of the overall calculation.

In Table 4 the speedup of the matrix multiplication step is labeled as (t_{mul}/t^H_{mul}). In the case of RI-MP2 the speedups range from 3.0 up to 4.3, while in the case of RI-dRPA the results are better resulting in speedups that are typically greater than 4.

dx.doi.org/10.1021/ct4002202 | J. Chem. Theory Comput. 2013, 9, 2654−2671

The acceleration of the matrix−matrix multiplication by GPUs has a significant impact on the overall timing, giving a speedup of 1.6 for the smaller systems to more than 2 for the larger ones where the matrix multiplication step is clearly dominant. In the limit of very large systems, the speedup can be expected to be even greater than 3.

4.5. System Size Scaling. In order to validate the performance models listed in Table 1 and Table 2, the time for the individual steps of the algorithms has been measured for systems of increasing size. The test system is based on a supercell containing 32 bulk water molecules, with a cc-DZVP basis set. The supercell has been replicated up to five times in one dimension, giving a final supercell containing 160 molecules. All calculations have been performed employing 1200 cores without GPUs. The timings are reported separately for the calculation of the $(ia|P)$ integrals (in common for all methods, Figure 7a) and for the matrix-multiplication part of the correlation energy calculation (different for all methods, Figure 7b). The obtained timings have been fitted with the function $y = bx^a$, yielding the measured scaling exponent a associated with each different step.

In the legend of Figure 7a the label "Calc (Q|P)" includes all the steps necessary to calculate the two center integrals over the auxiliary basis functions; they are all expected to scale roughly $O(N^2)$. For the calculation of the final $(ia|P)$ integrals more details have been reported separating the calculation of the potential, the integration of the potential to give the $(\mu\nu|P)$ integrals, and the index transformations leading to the final required form $(ia|P)$. As shown by the measured values of a the designed models are confirmed giving an $O(N^2)$ scaling for both the potential generation and integration and resulting in $a \simeq 3$ and $a \simeq 4$ for the first and second index transformation, respectively. It is important to note here the Cholesky decomposition and the triangular inversion steps are not reported since, even having an expected scaling of $O(N^3)$, the time associated with these steps is insignificant compared to the total for all sizes tested.

In Figure 7b the reported timings are related to the calculation of the $(ia|jb)_{RI}$ integrals, the $\mathbf{Q}(\omega) = 2\mathbf{B}^T\mathbf{B}'(\omega)$, and the $\bar{\mathbf{Q}} = \mathbf{B}^T\mathbf{B}'$, respectively, for the RI-MP2, RI-dRPA, and RI-SOS-MP2 methods. Also in this case the formal scaling of the individual step is confirmed to be ≃5 for RI-MP2 and ≃4 for the other two methods. The smaller prefactor observed for RI-SOS-MP2 compared to RI-dRPA is due to the lower number of points required in order to reach the same integration accuracy in the numerical quadrature.

4.6. Solid LiH. Due to its favorable features, such as large band gap, simple unit cell, and absence of heavy atoms, the LiH crystal has been widely used as the benchmark system in condensed phase electronic structure calculations.[35,63,116,126−129] In this section the estimated complete basis set (CBS) limit of the RI-MP2 and RI-dRPA contributions to the cohesive energy of the LiH crystal is reported and discussed. In order to do so, two extrapolations have been performed: the first with respect to the cell size going to infinity, the second with respect to the basis set.

The counterpoise corrected contributions to the cohesive energy of LiH for various basis sets and cell sizes together with the extrapolated values are summarized in Table 5. The calculation of the RI-dRPA energies as well as the EXX contributions have been performed with input electronic orbitals and orbital energies obtained from self-consistent PBE calculation. They are denoted respectively as RI-dRPA@PBE and HF@PBE. The calculations have been performed up to the supercell 4 × 4 × 4 for the cc-DZVP and cc-TZVP basis and up to supercell 3 × 3 × 3 for the cc-QZVP. In the following discussion, the energy contribution for a given pair of cell size/basis set is labeled as E_X^n, with X denoting the basis (D,T,Q,∞) and n denoting the number of repeated unit cells $(1,2,3,\infty)$

The estimate for the size converged limit, for a given basis set $(E_X^{n\to\infty})$, has been obtained employing the extrapolation formula $E_X^n = E_X^{n\to\infty} + S(n \times a)^{-3}$ where n is the number of repetitive cells considered and a is the lattice parameter. The choice of the exponent −3 used in the extrapolation has been inspired by the long-range behavior of both MP2 and dRPA pair energy, following the London law C_6/d_{ij}^6, with d_{ij} being the distance between the center of two charge distributions,[2S,130] and integrating over all pairs in the crystal for which $d_{ij} \geq d$. For all the basis sets considered, the $E_X^{n\to\infty}$ value has been obtained by a three points extrapolation for cell sizes ranging from 2 × 2 × 2 to 4 × 4 × 4. In the case of the cc-QZVP basis the additional point E_Q^4 has been obtained as $E_Q^3 \times (E_T^4/E_T^3)$. This extrapolation is justified by the observation that the ratio between the energies per formula unit for two consecutive cell sizes $(E_X^n/E_X^{(n+1)})$ converges quickly with respect to the basis set for both MP2 and dRPA. In fact, as shown in Table 6, the difference of E_X^2/E_X^3 in going from the cc-TZVP to the cc-QZVP basis is of the order of 10^{-4} for both RI-MP2 and RI-dRPA. Moreover, the difference between the computed result E_Q^3 and

Table 5. All-Electron Pseudopotential HF@PBE, RI-dRPA@PBE, and RI-MP2 Contributions to the Counterpoise Corrected Cohesive Energies in mE_h of LiH at the Experimental Geometry (a = 4.084 Å) for Various Basis Set and Cell Sizes[a]

	2 × 2 × 2	3 × 3 × 3	4 × 4 × 4	extr. $(E_X^{n\to\infty})$
		HF@PBE		
cc-DZVP	−131.91	−134.80	−135.00	
cc-TZVP	−124.84	−128.07	−128.31	
cc-QZVP	−124.41	−127.63	−127.88	
		RI-dRPA@PBE		
cc-DZVP	−27.05	−28.53	−28.95	*−29.19*
cc-TZVP	−38.42	−40.11	−40.62	*−40.89*
cc-QZVP	−41.86	−43.73	*−44.28*	*−44.59*
extr. $(E_{X\to\infty}^{n\to\infty})$				*−46.42*
		RI-MP2		
cc-DZVP	−29.25	−30.30	−30.57	*−30.75*
cc-TZVP	−38.00	−39.33	−39.68	*−39.91*
cc-QZVP	−40.57	−41.99	*−42.36*	*−42.60*
extr. $(E_{X\to\infty}^{n\to\infty})$				*−44.10*

[a]The text discusses how the extrapolated numbers (italic type) have been obtained.

Table 6. Ratio between the Energies Per Formula Unit of Bulk LiH for Two Consecutive Cell Sizes $((E_X^n)/(E_X^{(n+1)}))$[a]

	RI-MP2		RI-dRPA@PBE	
	E_X^2/E_X^3	E_X^3/E_X^4	E_X^2/E_X^3	E_X^3/E_X^4
cc-DZVP	0.9794	0.9951	0.9865	0.9961
cc-TZVP	0.9813	0.9954	0.9884	0.9965
cc-QZVP	0.9815		0.9881	

[a]X = D, T, and Q for the cc-DZVP, cc-TZVP, and cc-QZVP, respectively.

dx.doi.org/10.1021/ct4002202 | J. Chem. Theory Comput. 2013, 9, 2654−2671

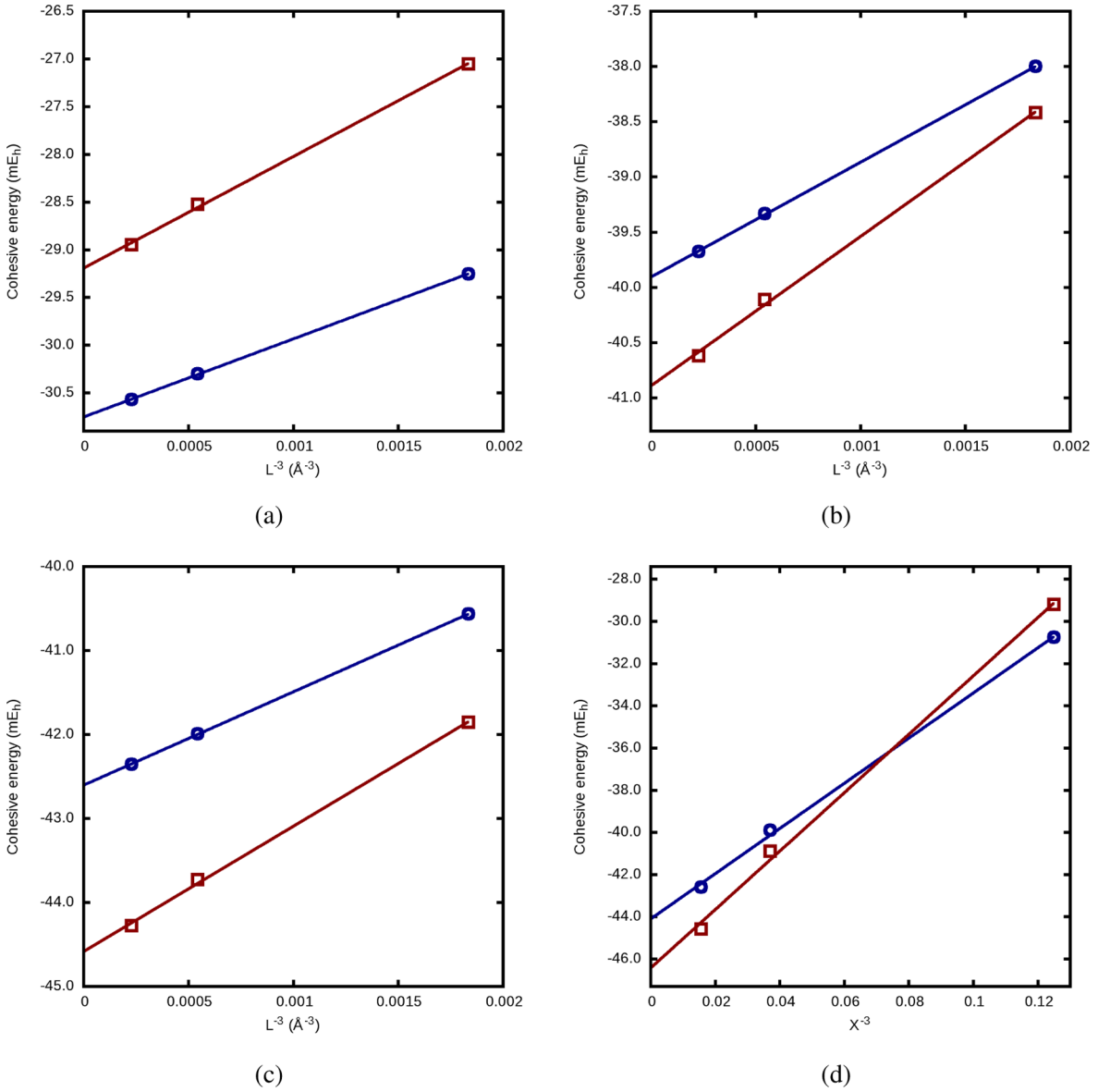


Figure 8. Convergence of the all-electron pseudopotential RI-dRPA@PBE (red squares) and RI-MP2 (blue circles) contribution to the cohesive energy of LiH at the experimental geometry (a = 4.084 Å) with respect to the cell size and basis set. L is the length of the cell edge ($L = n \times a$) and X = 2, 3, and 4 for cc-DZVP, cc-TZVP, and cc-QZVP, respectively. (a) Extrapolation with respect to the cell size for the cc-DZVP basis set. (b) Extrapolation with respect to the cell size for the cc-TZVP basis set. (c) Extrapolation with respect to the cell size for the cc-QZVP basis set. (d) Extrapolation with respect to the basis set for the size-converged cohesive energies.

the extrapolated $E_Q^2 \times (E_T^3/E_T^2)$ is of the order of a few tenths of μE_h for both RI-MP2 and RI-dRPA, validating the approach.

In order to obtain the CBS, the size extrapolated results for each basis set have been finally extrapolated with the cubic interpolation formula $E_X^{n\to\infty} = E_{X\to\infty}^{n\to\infty} + AX^{-3}$ (X = 2, 3, and 4 for cc-DZVP, cc-TZVP, and cc-QZVP, respectively).[33,34] The fits are shown in Figure 8, and the extrapolated values are reported in Table 5.

Finally, the complete basis set limit of the cohesive energy of LiH is estimated to be 44.10 and 46.42 mE_h for RI-MP2 and RI-dRPA, respectively. The obtained result in the RI-MP2 case is in excellent agreement with our previously reported MP2 value of 44.09, showing that the error introduced by the RI approximation is negligibly small. Considering the converged HF@PBE and HF values to be 127.9 and 132.0 mE_h respectively, it can be observed that, while the obtained value for the dRPA@PBE contribution to the cohesive energy is slightly larger that the pure MP2 result, the total cohesive energy given as (HF+dRPA)@PBE is underbinding compared to the HF+MP2 value. This behavior is in agreement with the trend observed for the atomization energies reported in several previous studies.[7,8,10,50]

4.7. Cohesive Energy of Molecular Crystals. The counterpoise corrected cohesive energy for several molecular crystals, calculated at different levels of theory using the experimental crystal cell parameters and employing the cc-TZVP basis, are reported in Table 7.

Together with the RI-MP2 results also the MP2 values, taken from our previous work,[63] have been reported as reference in order to assess the accuracy of the RI approximation. As shown in Table 7, the error introduced by the RI approximation is,

Table 7. Counterpoise Corrected Cohesive Energy ($-E_{coh}^{CP}$) in kJ/mol for the Molecular Crystals of B = Benzene, FA = Formic Acid, SA = Succinic Anhydride, D = 2,3-Diazanaphthalene, PD = Pyromellitic Dianhydride, U = Urea, and CT = Cyclotrimethylene-Trinitramine[a]

	B	FA	SA	D	PD	U	CT	RMSD
exp $\Delta H(s)$	45	68	81	83	83	92	112	
HF	−21.2	26.3	38.6	−5.7	31.3	55.8	49.8	58.1
MP2	58.8	55.5	81.2	79.7	123.4	94.6	113.7	16.9
RI-MP2	58.8	55.6	81.1	79.8	123.2	94.6	114.0	16.8
(HF+dRPA)@HF	29.2	45.0	68.1	45.7	92.0	81.1	91.1	20.6
dRPA@HF	50.3	18.6	29.5	51.4	60.7	25.3	41.3	
(HF+dRPA)@PBE0	36.3	47.0	69.9	49.2	94.1	83.4	83.9	19.9
dRPA@PBE0	60.6	27.7	36.1	54.6	71.3	33.2	36.0	
HF@PBE0	−24.3	19.3	33.8	−5.3	22.8	50.2	47.8	
(HF+dRPA)@B3LYP	37.1	46.9	69.9	49.1	94.1	83.4	82.8	20.1
dRPA@B3LYP	62.4	26.9	34.7	54.7	70.3	32.8	32.5	
HF@B3LYP	−25.4	20.0	35.2	−5.6	23.9	50.6	50.3	
(HF+dRPA)@PBE	37.6	45.0	67.3	44.8	91.0	81.1	75.0	23.2
dRPA@PBE	65.9	30.1	37.2	55.1	72.8	35.5	26.8	
HF@PBE	−28.3	14.8	30.2	−10.3	18.2	45.7	48.2	

[a]The sign of the E_{coh}^{CP} has been changed in order to be compared with the experimental sublimation enthalpies $\Delta H(s)$. $\Delta H(s)$ have been taken from the Supporting Information of ref 123, see also http://webbook.nist.gov/chemistry/.

also in this case, negligibly small, resulting in an error of a few tenths of kJ/mol in the final value of cohesive energy.

In the case of the EXX/dRPA method, the effect of different input orbitals in the computation of the counterpoise corrected cohesive energies has been tested. The tested reference wave functions have been chosen with a decreasing fraction of nonlocal exchange, ranging from the pure Hartree−Fock (100%), to PBE (0%), passing through B3LYP[121,124,125] (20%) and PBE0[131] (25%). As shown in Table 7 the decreasing fraction of nonlocal exchange, in general, results in an increase of the dRPA and a decrease of the EXX contributions to the cohesive energy. As noted in previous studies on vdW bonded molecular crystals,[108,109] these two effects have roughly the same magnitude resulting, in most cases, in a cancellation in the final value of the EXX/dRPA cohesive energy. The smallest root-mean-square deviation (RMSD), compared to the experimental sublimation enthalpies, is observed for the case in which PBE0 is used, even if the difference in RMSD with the other EXX/dRPA cases is quite small.

Finally, the obtained cohesive energies with the EXX/dRPA method are systematically underbinding the molecular crystals compared to the MP2 case, giving a slightly larger RMSD.

5. CONCLUSIONS

With the present work, a novel method for the resolution of identity approximation applied to the calculation of electron repulsion integrals over molecular orbitals, based on the Gaussian and Plane Waves approach, is introduced. The ground foundation of this method lies in the way the three center electron repulsion integrals of the type $(\mu\nu|P)$ are computed, that is, by direct integration between the product of the Gaussian basis function $(\mu\nu|$ and the electrostatic potential arising from the RI fitting densities $(P| = \sum_R (R|L_{PR}^{-1}$. This approach has been shown to be efficient, accurate, and robust for periodic systems; furthermore, it displays a measured scaling of the computational effort that grows only quadratically with the system size. In addition to that, it offers a straightforward way for parallel implementation.

The RI approximation has been applied to the calculation of the correlation energy at the MP2, SOS-MP2, and dRPA level of theory for finite and extended systems. Massively parallel algorithms have been developed for each of these methods, displaying excellent parallel scalability and efficiency up to tens of thousands of processes. Furthermore, it has been shown that a hybrid CPU/GPU implementation can result in speedups to individual steps of the algorithm up to a factor of 4.5 and around a factor of 2 for the global calculation compared to the standard only CPU implementation.

The RI approximation introduces negligible error in the final energy evaluation while giving a speedup, in the case of RI-MP2, of a factor of 10 to 15 compared to the standard GPW-MP2 implementation. In the SOS-MP2 and dRPA cases, the energy evaluation is carried out by a numerical integration procedure that allows for the reduction of the computational effort to $O(N^4)$. The numerical quadrature schemes have been shown to converge exponentially with respect to the number of grid points, such that, for systems with a sizable gap, 6−8 and 20−30 points are enough for micro-Hartree accuracy in the case of SOS-MP2 and dRPA, respectively.

Several benchmark calculations have been reported, showing that correlation energy calculations, at the different level of theory presented, can be performed within minutes for systems containing hundreds of atoms and thousands of basis functions.

■ ASSOCIATED CONTENT

Supporting Information

The pseudopotentials, basis-set parameters, crystal geometries, and S22 benchmark results. This material is available free of charge via the Internet at http://pubs.acs.org.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: delben@pci.uzh.ch (M.D.B.); hutter@pci.uzh.ch (J.H.); Joost.VandeVondele@mat.ethz.ch (J.V.).

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We thank Adrian Tate and Meng-Shiou Wu (Cray) for their support in PDGEMM tuning, Asbjorn Burow and Filipp

Furche for providing reference data for the dRPA calculations, and Ruyman Reyes Castro and Iain Bethune from the EPCC, supported by PRACE, for application analysis and optimization. J.V. acknowledges financial support by the European Union FP7 in the form of an ERC Starting Grant under contract no. 277910. Calculations were enabled by a grant from the Swiss National Supercomputer Centre (CSCS). The research leading to these results has received funding from the Swiss University Conference through the High Performance and High Productivity Computing (HP2C) Programme.

REFERENCES

(1) Perdew, J. P.; Ruzsinszky, A.; Tao, J.; Staroverov, V. N.; Scuseria, G. E.; Csonka, G. I. *J. Chem. Phys.* **2005**, *123*, 062201.
(2) Langreth, D. C.; Perdew, J. P. *Phys. Rev. B* **1977**, *15*, 2884−2901.
(3) Langreth, D.; Perdew, J. *Solid State Commun.* **1975**, *17*, 1425−1429.
(4) Gunnarsson, O.; Lundqvist, B. I. *Phys. Rev. B* **1976**, *13*, 4274−4298.
(5) Fuchs, M.; Gonze, X. *Phys. Rev. B* **2002**, *65*, 235109.
(6) Furche, F.; Voorhis, T. V. *J. Chem. Phys.* **2005**, *122*, 164106.
(7) Heßelmann, A.; Görling, A. *Mol. Phys.* **2011**, *109*, 2473.
(8) Eshuis, H.; Bates, J.; Furche, F. *Theor. Chem. Acc.* **2012**, *131*, 1084.
(9) Paier, J.; Ren, X.; Rinke, P.; Scuseria, G. E.; Grüneis, A.; Kresse, G.; Scheffler, M. *New J. Phys.* **2012**, *14*, 043002.
(10) Ren, X.; Rinke, P.; Joas, C.; Scheffler, M. *J. Mater. Sci.* **2012**, *47*, 7447.
(11) Møller, C.; Plesset, M. S. *Phys. Rev.* **1934**, *46*, 618−622.
(12) Szabo, A.; Ostlund, N. S. *Modern Quantum Chemistry*; McGraw Hill: New York, 1982.
(13) Görling, A.; Levy, M. *Phys. Rev. B* **1993**, *47*, 13105−13113.
(14) Bartlett, R. J.; Grabowski, I.; Hirata, S.; Ivanov, S. J. *Chem. Phys.* **2005**, *122*, 034104.
(15) Grimme, S. *J. Chem. Phys.* **2006**, *124*, 034108.
(16) Schwabe, T.; Grimme, S. *Phys. Chem. Chem. Phys.* **2006**, *8*, 4398−4401.
(17) Goerigk, L.; Grimme, S. *J. Chem. Theory Comput.* **2011**, *7*, 291−309.
(18) Saebø, S.; Pulay, P. *Annu. Rev. Phys. Chem.* **1993**, *44*, 213−236.
(19) Pulay, P.; Saebø, S. *Theor. Chim. Acta* **1986**, *69*, 357−368.
(20) Rauhut, G.; Pulay, P.; Werner, H.-J. *J. Comput. Chem.* **1998**, *19*, 1241−1254.
(21) Schütz, M.; Hetzer, G.; Werner, H.-J. *J. Chem. Phys.* **1999**, *111*, 5691−5705.
(22) Hetzer, G.; Schütz, M.; Stoll, H.; Werner, H.-J. *J. Chem. Phys.* **2000**, *113*, 9443−9455.
(23) Saebø, S.; Pulay, P. *J. Chem. Phys.* **2001**, *115*, 3975−3983.
(24) Pisani, C.; Busso, M.; Capecchi, G.; Casassa, S.; Dovesi, R.; Maschio, L.; Zicovich-Wilson, C.; Schütz, M. *J. Chem. Phys.* **2005**, *122*, 094113.
(25) Pisani, C.; Maschio, L.; Casassa, S.; Halo, M.; Schütz, M.; Usvyat, D. *J. Comput. Chem.* **2008**, *29*, 2113−2124.
(26) Maslen, P. *Chem. Phys. Lett.* **1998**, *283*, 102−108.
(27) Maslen, P. E.; Head-Gordon, M. *J. Chem. Phys.* **1998**, *109*, 7093−7099.
(28) Ayala, P. Y.; Scuseria, G. E. *J. Chem. Phys.* **1999**, *110*, 3660−3671.
(29) Lambrecht, D. S.; Doser, B.; Ochsenfeld, C. *J. Chem. Phys.* **2005**, *123*, 184102.
(30) Doser, B.; Lambrecht, D. S.; Kussmann, J.; Ochsenfeld, C. *J. Chem. Phys.* **2009**, *130*, 064107.
(31) Ayala, P. Y.; Kudin, K. N.; Scuseria, G. E. *J. Chem. Phys.* **2001**, *115*, 9698−9707.
(32) Jung, Y.; Shao, Y.; Head-Gordon, M. *J. Comput. Chem.* **2007**, *28*, 1953−1964.
(33) Halkier, A.; Helgaker, T.; Jørgensen, P.; Klopper, W.; Koch, H.; Olsen, J.; Wilson, A. K. *Chem. Phys. Lett.* **1998**, *286*, 243−252.
(34) Eshuis, H.; Furche, F. J. *Chem. Phys.* **2012**, *136*, 084105.
(35) Shepherd, J. J.; Grüneis, A.; Booth, G. H.; Kresse, G.; Alavi, A. *Phys. Rev. B* **2012**, *86*, 035111.
(36) Klopper, W.; Manby, F. R.; Ten-No, S.; Valeev, E. F. *Int. Rev. Phys. Chem.* **2006**, *25*, 427−468.
(37) Toulouse, J.; Gerber, I. C.; Jansen, G.; Savin, A.; Ángyán, J. G. *Phys. Rev. Lett.* **2009**, *102*, 096404.
(38) Gerber, I. C.; Angyan, J. G. J. *Chem. Phys.* **2007**, *126*, 044103.
(39) Eichkorn, K.; Treutler, O.; Öhm, H.; Häser, M.; Ahlrichs, R. *Chem. Phys. Lett.* **1995**, *240*, 283−290.
(40) Feyereisen, M.; Fitzgerald, G.; Komornicki, A. *Chem. Phys. Lett.* **1993**, *208*, 359−363.
(41) Weigend, F.; Häser, M.; Patzelt, H.; Ahlrichs, R. *Chem. Phys. Lett.* **1998**, *294*, 143−152.
(42) Bernholdt, D. E.; Harrison, R. J. J. *Chem. Phys.* **1998**, *109*, 1593−1600.
(43) Werner, H.-J.; Manby, F. R.; Knowles, P. J. J. *Chem. Phys.* **2003**, *118*, 8149−8160.
(44) Izmaylov, A. F.; Scuseria, G. E. *Phys. Chem. Chem. Phys.* **2008**, *10*, 3421.
(45) Maschio, L.; Usvyat, D.; Manby, F. R.; Casassa, S.; Pisani, C.; Schütz, M. *Phys. Rev. B* **2007**, *76*, 075101.
(46) Usvyat, D.; Maschio, L.; Manby, F. R.; Casassa, S.; Schütz, M.; Pisani, C. *Phys. Rev. B* **2007**, *76*, 075102.
(47) Katouda, M.; Nagase, S. *J. Chem. Phys.* **2010**, *133*, 184103.
(48) Jung, Y.; Lochan, R. C.; Dutoi, A. D.; Head-Gordon, M. *J. Chem. Phys.* **2004**, *121*, 9793−9802.
(49) Eshuis, H.; Yarkony, J.; Furche, F. *J. Chem. Phys.* **2010**, *132*, 234114.
(50) Ren, X.; Rinke, P.; Blum, V.; Wieferink, J.; Tkatchenko, A.; Sanfilippo, A.; Reuter, K.; Scheffler, M. *New J. Phys.* **2012**, *14*, 053020.
(51) Limaye, A. C.; Gadre, S. R. *J. Chem. Phys.* **1994**, *100*, 1303−1307.
(52) Marquez, A. M.; Dupuis, M. *J. Comput. Chem.* **1995**, *16*, 395−404.
(53) Nielsen, I. M. B.; Seidl, E. T. *J. Comput. Chem.* **1995**, *16*, 1301−1313.
(54) Baker, J.; Pulay, P. *J. Comput. Chem.* **2002**, *23*, 1150−1156.
(55) Ishimura, K.; Pulay, P.; Nagase, S. J. *Comput. Chem.* **2006**, *27*, 407−413.
(56) Bernholdt, D. E.; Harrison, R. *J. Chem. Phys. Lett.* **1996**, *250*, 477−484.
(57) Katouda, M.; Nagase, S. *Int. J. Quantum Chem.* **2009**, *109*, 2121−2130.
(58) Doser, B.; Lambrecht, D. S.; Ochsenfeld, C. *Phys. Chem. Chem. Phys.* **2008**, *10*, 3335.
(59) Valeev, E. F.; Janssen, C. L. *J. Chem. Phys.* **2004**, *121*, 1214−1227.
(60) Nakao, Y.; Hirao, K. *J. Chem. Phys.* **2004**, *120*, 6375−6380.
(61) Nielsen, I. M. B.; Janssen, C. L. *J. Chem. Theory Comput.* **2007**, *3*, 71−79.
(62) Maschio, L. *J. Chem. Theory Comput.* **2011**, *7*, 2818−2830.
(63) Del Ben, M.; Hutter, J.; VandeVondele, J. *J. Chem. Theory Comput.* **2012**, *8*, 4177−4188.
(64) Martyna, G. J.; Tuckerman, M. E. *J. Chem. Phys.* **1999**, *110*, 2810−2821.
(65) Blöchl, P. E. *J. Chem. Phys.* **1995**, *103*, 7422−7428.
(66) Genovese, L.; Deutsch, T.; Neelov, A.; Goedecker, S.; Beylkin, G. *J. Chem. Phys.* **2006**, *125*, 074105.
(67) Maschio, L.; Usvyat, D. *Phys. Rev. B* **2008**, *78*, 073102.
(68) Burow, A. M.; Sierka, M.; Mohamed, F. *J. Chem. Phys.* **2009**, *131*, 214101.
(69) Challacombe, M.; White, C.; Head-Gordon, M. *J. Chem. Phys.* **1997**, *107*, 10131−10140.
(70) Kudin, K. N.; Scuseria, G. E. *J. Chem. Phys.* **2004**, *121*, 2886−2890.
(71) Lippert, G.; Hutter, J.; Parrinello, M. *Theor. Chem. Acc.* **1999**, *103*, 124−140.

(72) Krack, M.; Parrinello, M. *Phys. Chem. Chem. Phys.* **2000**, *2*, 2105−2112.
(73) Grimme, S. *J. Chem. Phys.* **2003**, *118*, 9095−9102.
(74) Scuseria, G. E.; Henderson, T. M.; Sorensen, D. C. *J. Chem. Phys.* **2008**, *129*, 231101.
(75) Klopper, W.; Teale, A. M.; Coriani, S.; Pedersen, T. B.; Helgaker, T. *Chem. Phys. Lett.* **2011**, *510*, 147−153.
(76) Almlöf, J. *Chem. Phys. Lett.* **1991**, *181*, 319−320.
(77) Häser, M.; Almlöf, J. *J. Chem. Phys.* **1992**, *96*, 489−494.
(78) The CP2K developers group, 2013. CP2K is freely available from http://www.cp2k.org/ (accessed May 15, 2013).
(79) Whitten, J. L. *J. Chem. Phys.* **1973**, *58*, 4496−4501.
(80) Dunlap, B. I.; Connolly, J. W. D.; Sabin, J. R. *J. Chem. Phys.* **1979**, *71*, 3396−3402.
(81) Vahtras, O.; Almlöf, J.; Feyereisen, M. *Chem. Phys. Lett.* **1993**, *213*, 514−518.
(82) Weigend, F.; Häser, M. *Theor. Chem. Acc.* **1997**, *97*, 331−340.
(83) Head-Gordon, M.; Pople, J. A.; Frisch, M. *J. Chem. Phys. Lett.* **1988**, *153*, 503−506.
(84) Takatsuka, A.; Ten-no, S.; Hackbusch, W. *J. Chem. Phys.* **2008**, *129*, 044112.
(85) Braess, D.; Hackbusch, W. *IMA J. Numer. Anal.* **2005**, *25*, 685−697.
(86) Furche, F. *J. Chem. Phys.* **2008**, *129*, 114105.
(87) Bauernschmitt, R.; Ahlrichs, R. *Chem. Phys. Lett.* **1996**, *256*, 454−464.
(88) Boyd, J. P. *J. Sci. Comput.* **1987**, *2*, 99−109.
(89) Lippert, G.; Hutter, J.; Parinello, M. *Mol. Phys.* **1997**, *92*, 477−488.
(90) VandeVondele, J.; Krack, M.; Mohamed, F.; Parrinello, M.; Chassaing, T.; Hutter, J. *Comput. Phys. Commun.* **2005**, *167*, 103−128.
(91) Ihm, J.; Zunger, A.; Cohen, M. L. *J. Phys. C* **1979**, *12*, 4409.
(92) Aissing, G.; Monkhorst, H. *J. Int. J. Quantum Chem.* **1993**, *48*, 81−89.
(93) Nakajima, T.; Hirao, K. *Chem. Phys. Lett.* **2006**, *427*, 225−229.
(94) Kats, D.; Usvyat, D.; Loibl, S.; Merz, T.; Schütz, M. *J. Chem. Phys.* **2009**, *130*, 127101.
(95) Furche, F. *Phys. Rev. B* **2001**, *64*, 195120.
(96) Aryasetiawan, F.; Miyake, T.; Terakura, K. *Phys. Rev. Lett.* **2002**, *88*, 166401.
(97) Nguyen, H.-V.; de Gironcoli, S. *Phys. Rev. B* **2009**, *79*, 205114.
(98) Miyake, T.; Aryasetiawan, F.; Kotani, T.; van Schilfgaarde, M.; Usuda, M.; Terakura, K. *Phys. Rev. B* **2002**, *66*, 245103.
(99) García-González, P.; Fernández, J. J.; Marini, A.; Rubio, A. *J. Phys. Chem. A* **2007**, *111*, 12458−12465.
(100) Harl, J.; Kresse, G. *Phys. Rev. Lett.* **2009**, *103*, 056401.
(101) Harl, J.; Schimka, L.; Kresse, G. *Phys. Rev. B* **2010**, *81*, 115126.
(102) Xiao, B.; Sun, J.; Ruzsinszky, A.; Feng, J.; Perdew, J. P. *Phys. Rev. B* **2012**, *86*, 094109.
(103) Rohlfing, M.; Bredow, T. *Phys. Rev. Lett.* **2008**, *101*, 266106.
(104) Ren, X.; Rinke, P.; Scheffler, M. *Phys. Rev. B* **2009**, *80*, 045402.
(105) Marini, A.; García-González, P.; Rubio, A. *Phys. Rev. Lett.* **2006**, *96*, 136404.
(106) Mittendorfer, F.; Garhofer, A.; Redinger, J.; Klimeš, J.; Harl, J.; Kresse, G. *Phys. Rev. B* **2011**, *84*, 201401.
(107) Harl, J.; Kresse, G. *Phys. Rev. B* **2008**, *77*, 045136.
(108) Lu, D.; Li, Y.; Rocca, D.; Galli, G. *Phys. Rev. Lett.* **2009**, *102*, 206411.
(109) Li, Y.; Lu, D.; Nguyen, H.-V.; Galli, G. *J. Phys. Chem. A* **2010**, *114*, 1944−1952.
(110) Perdew, J. P.; Burke, K.; Ernzerhof, M. *Phys. Rev. Lett.* **1996**, *77*, 3865−3868.
(111) Goedecker, S.; Teter, M.; Hutter, J. *Phys. Rev. B* **1996**, *54*, 1703−1710.
(112) Dunning, T. H. *J. Chem. Phys.* **1989**, *90*, 1007−1023.
(113) Woon, D. E.; Dunning, T. H. *J. Chem. Phys.* **1993**, *98*, 1358−1371.
(114) Guidon, M.; Schiffmann, F.; Hutter, J.; VandeVondele, J. *J. Chem. Phys.* **2008**, *128*, 214104.
(115) Guidon, M.; Hutter, J.; VandeVondele, J. *J. Chem. Theory Comput.* **2009**, *5*, 3010−3021.
(116) Paier, J.; Diaconu, C. V.; Scuseria, G. E.; Guidon, M.; VandeVondele, J.; Hutter, J. *Phys. Rev. B* **2009**, *80*, 174114.
(117) Jurecka, P.; Sponer, J.; Cerny, J.; Hobza, P. *Phys. Chem. Chem. Phys.* **2006**, *8*, 1985−1993.
(118) Takatani, T.; Hohenstein, E. G.; Malagoli, M.; Marshall, M. S.; Sherrill, C. D. *J. Chem. Phys.* **2010**, *132*, 144104.
(119) Allen, F. H. *Acta Crystallogr., Sect. B: Struct. Sci* **2002**, *58*, 380−388.
(120) Becke, A. D. *Phys. Rev. A* **1988**, *38*, 3098−3100.
(121) Lee, C.; Yang, W.; Parr, R. G. *Phys. Rev. B* **1988**, *37*, 785−789.
(122) Maschio, L.; Usvyat, D.; Schütz, M.; Civalleri, B. *J. Chem. Phys.* **2010**, *132*, 134706.
(123) Maschio, L.; Civalleri, B.; Ugliengo, P.; Gavezzotti, A. *J. Phys. Chem. A* **2011**, *115*, 11179−11186.
(124) Becke, A. D. *J. Chem. Phys.* **1993**, *98*, 5648−5652.
(125) Vosko, S. H.; Wilk, L.; Nusair, M. *Can. J. Phys.* **1980**, *58*, 1200−1211.
(126) Stoll, H.; Doll, K. *J. Chem. Phys.* **2012**, *136*, 074106.
(127) Usvyat, D.; Civalleri, B.; Maschio, L.; Dovesi, R.; Pisani, C.; Schütz, M. *J. Chem. Phys.* **2011**, *134*, 214105.
(128) Marsman, M.; Grüneis, A.; Paier, J.; Kresse, G. *J. Chem. Phys.* **2009**, *130*, 184103.
(129) Nolan, S. J.; Gillan, M. J.; Alfè, D.; Allan, N. L.; Manby, F. R. *Phys. Rev. B* **2009**, *80*, 165109.
(130) Dobson, J. F.; Gould, T. *J. Phys.: Condens. Matter* **2012**, *24*, 073201.
(131) Adamo, C.; Barone, V. *J. Chem. Phys.* **1999**, *110*, 6158−6170.