Enabling Simulation at the Fifth Rung of DFT: Large Scale RPA Calculations with Excellent Time to Solution

Mauro Del Ben^{a,*}, Ole Schütt^b, Tim Wentz^c, Peter Messmer^{d,e}, Jürg Hutter^a, Joost VandeVondele^{b,*}

^aDepartment of Chemistry, University of Zurich, Winterthurerstrasse 190, CH-8057, Zurich, Switzerland

^bDepartment of Materials, ETH Zurich, Wolfgang-Pauli-Strasse 27, CH-8093, Zurich, Switzerland

^cCray Inc., 380 Jackson Street, St Paul, MN 55101, USA

^dNVIDIA Switzerland, Technoparkstr 1, CH-8005 Zurich, Switzerland

^eNVIDIA Co-design lab for hybrid multicore computing, Wolfgang-Pauli-Str. 27, CH-8093 Zurich, Switzerland

Abstract

The Random Phase Approximation (RPA), which represents the fifth rung of accuracy in Density Functional Theory (DFT), is made practical for large systems. Energies of condensed phase systems containing 1000s of explicitly correlated electrons and 1500 atoms can now be computed in minutes and less than one hour, respectively. GPU acceleration is employed for dense and sparse linear algebra, while communication is minimized by a judicious data layout. The performance of the algorithms, implemented in the widely used CP2K simulation package, has been investigated on hybrid Cray XC30 and XK7 architectures, up to 16384 nodes. Our results emphasize the importance of good network performance, in addition to the availability of GPUs and generous on node memory. A new level of predictivity has thus become available for routine application in Monte Carlo and molecular dynamics simulations.

Keywords: Random Phase Approximation, RPA, Density Functional Theory, DFT, Monte Carlo, CP2K

1. Introduction

Density Functional Theory (DFT) is a Nobel prize winning theory that has become the most widely used quantum mechanical tool in chemistry and physics for predicting properties of materials ranging from single molecules to condensed phase systems. DFT is equivalent to the Schrödinger equation and, within the Kohn-Sham (KS) approach,[1, 2] the associated problem can be solved in a convenient and formally exact way. Despite to this, practical KS-DFT calculations replace the unknown exact exchange and correlation energy by one of the various models (functionals) of electron correlation. The success of KS-DFT can be attributed to the excellent 'price-performance ratio' that simple models of electron correlation offer. In particular, the models based on the generalized gradient approximation (GGA) have been a real breakthrough in computational chemistry. GGA KS-DFT requires only the electron density and its gradient as an input to the functional and for this reason they are often referred as semi-local functionals. This simple form

leads to a low computational cost, scaling moderately ($O(N^3)$) with system size (N), while the introduction of the information on the electron density gradient improves the accuracy especially for applications in molecular sciences. Systems containing several hundreds of atoms, necessary and sufficient to describe a wide range of condensed phase phenomena of scientific and technological impact, can now be routinely studied. To optimize atomic structures or to perform first-principles molecular dynamics (MD) and Monte Carlo (MC), time-to-solution is essential, as 10s to 10,000s of sequential energy/force calculations are required. Consequently, only if the time of a single energy/force calculation drops below a few minutes, first-principles MD and MC become practical. Enabling MD and MC has been a prime motivation for massively parallel implementations of GGA DFT in the past, and is, at a more advanced level, also a target for the current work.

To make progress on the way to high accuracy and thus to the predictivity of the theory, we focus on correlation models beyond GGA DFT. To illustrate the need for functionals beyond GGA, the failure of standard GGA functionals to yield van der Waals or dispersive interactions can be used as one example. The absence of these impor-

*Corresponding author

Email addresses: mauro.delben@chem.uzh.ch (Mauro Del Ben), Joost.VandeVondele@mat.ethz.ch (Joost VandeVondele)

tant weak interactions leads to gross errors and qualitatively wrong predictions of seemingly simple properties. For example, the density of water is underestimated by roughly 20% by the GGA BLYP functional[3, 4, 5], and BLYP ice sinks in, rather than floats on, BLYP water. Even if one can argue that this flaw is related to the poor performance of BLYP in describing noncovalent interactions, there are many evidences that this is a general failure at the GGA level.[6, 7] More advanced correlations models take ingredients beyond the electron density into account, and utilize this additional data to improve the quality of the results. The class of functionals that include the virtual orbitals, the so called fifth rung of DFT, is an essential step forward. Indeed, for one such model (MP2 theory), we have recently demonstrated by means of MC simulations that it yields a computed density of water within 2% of experiment, a 10-fold improvement over BLYP.[5, 8] This improvement results in a correct prediction of the most famous anomaly of water, namely that ice floats.

We first clarify the nomenclature and properties of 'fifth rung' functionals, before we discuss the computational aspects. Perdew *et al.*[9, 10] classified the known functionals on a "Jacob's ladder", leading from a 'Hartree-world' to a metaphorical heaven of chemical accuracy. Each rung of the ladder introduces more descriptors of the electronic system and yields models with improved accuracy. This paper deals with the most advanced, fifth rung, which include 1) the density, 2) its gradient, 3) the kinetic energy density, 4) the occupied, and most importantly 5) the virtual orbitals. By capturing physical phenomena not accounted for in semi-local DFT, the fifth rung represents essential progress, including best performance in various benchmark sets. [11, 12, 13, 14, 15, 16, 17, 18] The various functionals on the fifth rung include virtual orbitals either via the random phase approximation (RPA) correlation [19, 20, 21, 22, 23, 24, 13, 25, 26] or via terms derived from second-order Møller-Plesset (MP2) theory, in the form of double hybrids (DH). [27, 28, 29, 30, 31, 11, 12, 15, 16, 32] The focus of this manuscript is on RPA, which has favorable properties both from a computational and physical point of view. Nevertheless, results from MP2 and DH approaches are discussed as well, given the similarity of the methods and their identical location on "Jacob's ladder".

It is not surprising that the benefits of improved accuracy come with increased computational requirements. Unfortunately, this increase from the commonly used second rung to the fifth rung is so large that until our recent work[5] no condensed phase MD or MC had ever been performed. Indeed, the formal computational effort

of RPA and MP2 calculations, in their canonical formulation, scales as $O(N^6)$ and $O(N^5)$ respectively, compared to the $O(N^3)$ of the second rung. An aggravating fact is that fifth rung methods need significantly more basis functions, the proper measure of system size N , than the lower rungs to deliver converged results.[33, 34] Prefactors are such that already for a few dozen atoms second and fifth rung approaches differ by roughly three orders of magnitude in computational demand.

Here, we present recent algorithmic and computational progress that mitigates these requirements and allows for RPA calculations with excellent time to solution on massively parallel hybrid architectures. As we will discuss in detail below, this is based on recent developments:

- An efficient formulation of RPA that exhibits a manageable $O(N^4)$ scaling with small prefactor.[35, 36]
- A novel algorithm, named RI-GPW, for the computation of the needed integrals, which includes a data layout and communication schemes that result in a massively parallel implementation.[37]
- A new implementation that brings all $O(N^4)$ steps to the GPU, including dense and sparse linear algebra.[38]

As a result, using the latest hybrid XC30 computer architecture, energy calculations on typical simulation cells of liquid water (64 H_2O), reach the one minute per step limit already at 512 nodes. This makes MC simulations on this important system now routinely feasible, and represents a breakthrough in the field. The energy of 1000s of correlated electrons (256 H_2O) can be computed in less than ten minutes on 4096 XC30 nodes, and a calculation on 1500 atoms (512 H_2O) is feasible in less than one hour on 16384 XK7 nodes. We therefore anticipate that this level of simulation will be used with increasing frequency to tackle further important open questions with relevance to chemistry, physics, biology, energy and materials science.

2. Current state of the art

In order to perform these calculations with acceptable time to solution, and thus to extend the applicability of these methods to larger systems, the development of efficient algorithms and scalable implementations is of prime interest and an active topic of research in the community. It is now well established that the resolution of identity (RI) is a powerful technique to reduce both

Figure 1: Left: Shown is the exponential convergence of the energy expression Eq.4 with respect to the number of integration points for two different quadrature schemes discussed in Section 3.2.2 (RDX = Cyclotrimethylene-Trinitramine). This greatly contributes to the efficiency of the $O(N^4)$ RI-GPW RPA scheme. Right: Characteristic for the RI-GPW RPA approach is the calculation of the matrix of three-center ERI $(\mu\nu|P)$ using a FFT based Poisson solver, yielding for a given P the sparse matrix in linear scaling time. For each $|P\rangle$, calculations can be performed independently, contributing to excellent scalability.

the prefactor and the scaling of RPA and MP2 calculations. [39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 35, 49]. As discussed below, within direct-RPA, the RI approach speeds-up the calculation of the integrals, reduces the memory requirements, and additionally also allows for reducing the scaling from $O(N^6)$ to $O(N^4)$. [35, 50, 51] This significant reduction in scaling is key for the application to large systems.

Several RPA implementations have been reported in the literature, formulated in term of Gaussian [50, 35, 51], numeric [49, 26] and plane-wave (PW) [22, 52, 53, 54, 55] basis sets. Gaussian basis functions, known for their efficiency for gas phase calculations, but also used by us for the condensed phase, have been used by Es-huis and coworkers [35], reporting as the largest test case the calculation of the RPA correlation energy of the octapeptide Angiotensin II (146 atoms, 1117 atomic orbitals) requiring 18.9 h on a Xeon X7550 2.00 GHz CPU. PW basis sets are most commonly used for condensed phase calculations and RPA has been successfully employed for studying solids [56, 57, 58, 59, 60, 61], surfaces [14, 62, 63, 64, 65] and van der Waals crystals [52, 53, 66]. Also in this basis, the computational cost limits the system size. In a very recent paper that studies low scaling RPA methods, current applications are reported to be limited to $O(100)$ electrons. [36] Although no information about the actual time to solution is reported, Harl *et al.* [58] mentioned that their im-

plementation scales to 100 CPU cores for the $O(N^4)$ computational task. In a recent RPA study on water ice, Macher and coworkers [67] employed 28 H₂O molecules for their largest unit cell.

Contrary to the RPA case, several parallel MP2 and RI-MP2 implementations have been discussed in detail in the past. [68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81]. These methods are briefly discussed here, as they share with RI-RPA an algorithmically similar approach to compute the integrals, but ultimately differ due to an $O(N^5)$ scaling. For gas phase molecules, parallel canonical MP2 calculations have been reported [71, 82, 83], for example requiring 4.8 h on 8192 cores for C₆₀. However, it is now clear that RI-MP2 approaches are superior in performance compared to canonical MP2, without significant loss in accuracy. [37] The largest gas phase RI-MP2 calculation reported, is recent work by Katouda and Nakajima [81], in which, with a new MPI/OpenMP hybrid parallel algorithm, the RI-MP2 energy of interacting nanographene flakes (C₁₅₀H₃₀)₂ (9840 basis functions) was calculated in 65 minutes employing 71288 CPU cores of the K computer. When considering systems in the condensed phase, due to the 3D nature and periodic boundary conditions, the calculations become computationally more demanding, and for large systems alternative methods have been developed. In this respect, Maschio presented a parallel local MP2 implementation for periodic systems making use of density fitting

[80] reporting a calculation on a metal organic framework (MOF-5) crystalline structure (106 atoms and 2884 atomic orbitals per unit cell) performed in less than 24 h on 53 processors. Recently, we have reported[37] on our parallel RI-MP2 implementation, based on the RI-GPW scheme discussed below. In that work we have shown that our RI-MP2 implementation demonstrates excellent time to solution for systems in the condensed phase containing hundreds of atoms and thousands of basis functions. As an example, for the Cyclotrimethylene-Trinitramine molecular crystal, better known as the RDX explosive (168 atoms and 4152 atomic orbitals per unit cell), the RI-MP2 energy has been calculated in 7.4min on 200 nodes of a Cray XK7.[37] The efficiency of this scheme is what has enabled our current RI-GPW RPA implementation, and will be discussed in the following.

3. Recent Innovations

3.1. $O(N^4)$ RI-RPA Theory

In this section, the canonical expression for the direct-RPA correlation energy (E_c^{RPA}) is reformulated in such a way that the scaling changes from $O(N^6)$ to $O(N^4)$. This essential step is based on a resolution of identity (RI) technique, recently introduced in RPA by Furche and co-authors. [35]

The canonical form of the RPA correlation energy is given by

$$E_c^{\text{RPA}} = \frac{1}{2} \text{Tr}(\mathbf{M}^{1/2} - \mathbf{A}). \quad (1)$$

The matrix elements of $\mathbf{A}$ and $\mathbf{M}$ are related to the six-dimensional electron repulsion integrals ($ia|jb$) over canonical molecular orbitals (MO ERIs), and the Kohn-Sham orbital energies ϵ_i and ϵ_a . These ERIs are important quantities in the following, and are indexed by i, j and a, b , which refer to occupied (o) and virtual (v) molecular orbitals (MOs) respectively. Specifically, within the direct-RPA approach [35, 51] considered in this work, the $\mathbf{M}$ matrix is defined according to:

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

where

$$\begin{aligned} (A - B)_{ia,jb} &= (\epsilon_a - \epsilon_i) \delta_{ij} \delta_{ab} \\ (A + B)_{ia,jb} &= (\epsilon_a - \epsilon_i) \delta_{ij} \delta_{ab} + 2(ia|jb). \end{aligned}$$

A straightforward computation of E_c^{RPA} using Eq. 1 implies calculation of $\mathbf{M}^{1/2}$. The dimension of $\mathbf{M}$ is $(ov) \times (ov)$, so that the number of rows scales as $O(N^2)$, the memory requirements as $O(N^4)$, and the flops as $O(N^6)$, since both o and v are proportional to system size

N . For example, for the 512 H₂O system, $\mathbf{M}$ would be a dense, square, $56M \times 56M$ matrix, its diagonalization is beyond the capabilities of current supercomputers.

To avoid this steep scaling, an auxiliary Gaussian basis P, Q of size $N_a \propto N$ is introduced for an RI approximation

$$\begin{aligned} (ia|jb) &\simeq (ia|jb)_{\text{RI}} = \sum_{RQ} (ia|R)(R|Q)^{-1}(Q|jb) \\ &= \sum_P B_P^{ia} B_P^{jb}. \end{aligned} \quad (3)$$

The three center matrix $\mathbf{B}$, given by $B_P^{ia} = \sum_R (ia|R)L_{PR}^{-1}$ with L_{PR}^{-1} obtained from the Cholesky decomposition of the two center $(R|Q)$, is thus sufficient to compute the integrals. This $\mathbf{B}$ requires only $O(N^3)$ memory, for the example given above the dimensions are $56M \times 70K$, a nearly 1000-fold improvement over $\mathbf{M}$, but nevertheless corresponding to 28 TB of RAM.

Most importantly, exploiting the structure of $\mathbf{M}$ and the factorization of ERIs in terms of $\mathbf{B}$, it can be shown [35] that $E_c^{\text{RI-RPA}}$ is identical to the frequency integral

$$E_c^{\text{RI-RPA}} = \frac{1}{2} \int_{-\infty}^{+\infty} \frac{d\omega}{2\pi} \text{Tr}(\log(\mathbf{1} + \mathbf{Q}(\omega)) - \mathbf{Q}(\omega)). \quad (4)$$

Here, the frequency dependent $N_a \times N_a$ matrix $\mathbf{Q}(\omega)$ is given by matrix multiplications as $\mathbf{Q}(\omega) = 2\mathbf{B}^T \mathbf{G}(\omega) \mathbf{B}$, where $\mathbf{G}(\omega)$ is a diagonal matrix that can be computed easily from the Kohn-Sham eigenvalues and frequency. The integral can be computed by numerical quadrature with excellent accuracy using a small number of integration points N_g (independent of system dimension) as shown in Fig. 1. For a given ω , the computation of the integrand requires $O(N_a N_a ov) = O(N^4)$ operations for the matrix multiplications implied by the definition of $\mathbf{Q}$, and $O(N^3)$ for the evaluation of the trace of the logarithm. The total operation count has thus been reduced from $O(N^6)$ to $O(N^4)$, enabling practical large scale calculations. Within this framework, an $O(N^4)$ formalism for forces have also been established.[84]

As a final remark, it has to be noted that there is a close connection between many-body perturbation theory (MBPT) and RPA, such that RPA correlation energy can be equivalently be derived from MBPT.[26] In particular the Taylor expansion of the logarithm in Eq. 4 truncated at the second order leads to an equation that is equivalent to the Opposite-Spin MP2 energy evaluated with KS orbitals within the RI approximation.[35] Additionally, it has to be noted that second-order energy within perturbation theory include contributions from singly-excited configurations. While these contributions

result to be zero for an Hartree-Fock reference they are not if a KS wavefunction is used. Similarly, also the RPA correlation energy can be corrected for single excitation contributions,[85, 86] fixing for example the systematic underbinding of molecules and solids obtained with standard RPA approaches. The inclusion of these advanced features is beyond the scope of this work and the remainder of the paper will be focused on the direct-RPA method.

3.2. The RI-GPW RPA method

The RI Gaussian and Plane Waves (RI-GPW) RPA method, recently introduced by us[37], accomplishes the evaluation of Eq. 4 efficiently and in a highly scalable way. Ref. [37] provides an in-depth description, so here we present the high-level structure, and emphasize recent improvements, implemented to enable the presented calculations. The method consists of two consecutive steps, namely the computation of the matrix elements of $\mathbf{B}$, and the frequency integration of Eq. 4.

3.2.1. The RI-GPW method

The RI-GPW method deals with the computation of the matrix elements of $\mathbf{B}$, and is equivalent to computing $O(N_a o v)$ 6-dimensional integrals of Coulombic nature. Also this step scales as $O(N^4)$, and is, contrary to the evaluation of Eq.4, highly complex in nature and potentially the bottleneck in all but the largest calculations. Referring again to the 512 H₂O system, this step amounts to computing $3.9 \cdot 10^{12}$ 6D integrals, each with non-vanishing contributions from the complete simulation cell.

The parallelization for computing the $(ia|P)$ ERIs within the RI-GPW approach is achieved with a dual-level hybrid OpenMP/MPI scheme, and a careful process layout. To this end, the N_p processes available in total are split in N_G groups, each consisting of N_w processes ($N_p = N_G N_w$). The first level of parallelization corresponds to distributing the work performed for a single given auxiliary basis function ϕ_P or vector $|P\rangle = \sum_R \phi_R L_{PR}^{-1}$ over the group of N_w processes. Since $N_a > N_p$ it is favorable to make N_w small, leading to excellent efficiency for this step. However, the $O(N^2)$ memory requirements on the first level, are such that the availability of parallelism is essential to run the large systems presented in this work. The second level of parallelization corresponds to a distribution of the nearly independent calculations on $|P\rangle$ over N_G groups. The fact that these calculations are independent is a major advantage of the RI-GPW technique.

On the first level, the details of the operations are complicated, involving parallel FFTs, halo-exchanges, and

sparse matrix multiplications, ultimately computing the electrostatic energy of various Gaussian-based charge distributions. Fortunately, it corresponds to the standard parallelization scheme and modus of operation for GGA DFT calculations in our simulation package CP2K[87]. As such, this step has been streamlined over the past 10 years, including interprocess, multi-threading, and single core optimizations, and will not be discussed in detail, except for the newly developed GPU-accelerated sparse matrix library. As sketched in Fig. 1, the initial steps consist of a density $\rho(\vec{R})$ calculation on a real space grid, the computation of the associated electrostatic potential $v(\vec{R})$ using a spectral approach, and the final numerical integration of $v(\vec{R})$ over Gaussian basis functions $(\mu\nu)$. Exploiting sparsity, the calculation of the $(\mu\nu|P)$ matrix is $O(N)$ for each given $|P\rangle$. This first step only contributes $O(N^2)$ to the computational workload, but contributes significantly to the total runtime for smaller systems. Its efficiency is key to the success of the RI-GPW RPA method. This step is followed by two integral transformations from the atomic orbital $(\mu\nu|P)$ to molecular orbital $(ia|P)$ basis, performed as two sparse matrix multiplications with comparatively small ($N \times o$ and $N \times v$) matrices. The first index transformation benefits from sparsity, and contributes $O(N^3)$ to the total workload, while the second index transformation is asymptotically dominating with $O(N^4)$ flops. With the new sparse matrix library, both transformations run efficiently on the GPU. At the end of the RI-GPW integral algorithm each process stores part of the matrix elements B_p^{ia} , and the RPA frequency integration can be performed.

3.2.2. RPA frequency integration

For large systems, the total time necessary for the computation of $E_c^{\text{RI-RPA}}$ depends linearly on the number of quadrature points employed in the numerical quadrature of the integral in equation 4. For the calculation of the integral, two different quadrature schemes have been implemented and tested, one is based on the Clenshaw-Curtis (CC) numerical quadrature [88], the other on the Minimax approximation [89, 90].

The CC scheme has been originally introduced for the calculation of $E_c^{\text{RI-RPA}}$ by Eshuis *et al.* [35] and is based on the mapping $\omega = a \cot(t)$ to the interval $t \in [0, \pi]$, a being a scaling parameter optimized according to a diagonal approximation of the RPA excitation energies

The Minimax approximation has been successfully applied for the evaluation of the spin-opposite MP2 energy [91] within the Laplace Transformation method [92, 93] and has been introduced only recently for the calculation of the RPA energy by Kaltak and coworkers [36]. As reported in detail in their work, given N_q quadrature points,

Figure 2: Left: minimax error function $\eta(x)$ for $R_c = 100$ and 8 quadrature points. Right: absolute maximum and minimum error (ϵ) of $\eta(x)$ as a function of R_c for different number of quadrature points.

the MM weights $\{w_i\}$ and abscissa $\{\omega_i\}$ are obtained by optimizing the error function

$$\eta(x) = \frac{1}{x} - \frac{1}{\pi} \sum_i^{N_q} w_i \left[\frac{2x}{x^2 + \omega_i^2} \right]^2 \quad (5)$$

such that it creates an alternant in the scaled interval $[1, R_c]$ over which the integration is carried out. More specifically, within the Minimax approximation, the error function satisfies

$$\eta(x_i) = (-1)^i \epsilon, \quad \forall i = 1, 2, \dots, 2N_q + 1 \quad (6)$$

for which the $\{x_i\}$ represent the positions of the local maximum and minimum of $\eta(x)$, while ϵ represent the error extremum. In this way $\eta(x)$ guarantees for the best approximation in the given interval due to the alternation of minimum and maximum errors, equal in absolute value and opposite in sign. Fig. 2 (left) shows the resulting $\eta(x)$ for $N_q = 8$ and $R_c = 100$.

There are two other appealing features of the MM scheme. The first is that the range of the approximation R_c , contrary to the CC case, can be obtained solely from the bare orbital energies and thus is calculated prior to the calculation of the correlation energy. The second is related to the property that, for a given number of quadrature points N_q , an upper bound exists $R_c^{\max}$ such that $\eta(x)$ is strictly decaying for $x > R_c^{\max}$. This implies that the Minimax parameters $\{w_i\}$ and $\{\omega_i\}$ for any $R_c > R_c^{\max}$ coincide, since the associated error extremum ϵ is the same in both cases. Fig. 2 (right) shows the variation of ϵ as a function of $R_c \leq R_c^{\max}$ for different N_q . This plot provides a rough estimation of the error in the final

RPA energy for a given pair (R_c, N_q) .

As shown in Fig. 1, both approaches, CC and MM, yield exponential convergence, but with a different rate. As a result, for micro-Hartree accuracy, 20 – 30 quadrature points are required for the CC scheme while 10 – 15 are sufficient for MM.

Two observations can be made that further improve efficiency. First, $\mathbf{B}^T \mathbf{G}(\omega) \mathbf{B}$ can be favorably rewritten as $\mathbf{B}'(\omega)^T \mathbf{B}'(\omega)$ where $\mathbf{B}'(\omega) = \mathbf{G}(\omega)^{1/2} \mathbf{B}$. This allows us to exploit symmetry and to replace, in ScaLAPACK notation, PDGEMM by PDSYRK, not only saving half the flops, but equally important, half the memory. Second, $\text{Tr}(\log(\mathbf{1} + \mathbf{Q}(\omega)))$ is computed efficiently, in particular avoiding the explicit computation of the logarithm (diagonalization), by exploiting the identity $\text{Tr}[\ln \mathbf{A}] = \ln(\text{Det}[\mathbf{A}])$, where the determinant is obtained from the Cholesky decomposition of the positive definite argument. Such a simple observation leads to a roughly 10× speedup of this $O(N^3)$ step.

Finally, the parallel algorithm is again based on a dual-level parallelization. The first level corresponds to the distribution of the quadrature points over subgroups of processes (the integration groups), while the second level corresponds to the parallel work implied by the matrix multiplication needed for each quadrature point, which is discussed in section 3.3.2. The redistribution and replication of the $\mathbf{B}$ matrix can be performed efficiently, as the data layout is the same between the integration groups. Since each integration point is essentially independent, parallelization is highly effective also over the quadrature points. Memory requirements restrict the minimum size of an integration group, for example 512 H₂O require at least 4096 nodes per point, so that with 16 integration

points this $O(N^4)$ step would likely scale beyond 65536 nodes.

3.3. GPU accelerated dense and sparse linear algebra

The $O(N^3)$ and $O(N^4)$ steps of the RI-GPW RPA algorithm are a mixture of sparse and dense linear algebra that benefit significantly from GPU acceleration. During the last year, significant progress has been made in this area, which is summarized here, resulting in all $O(N^4)$ steps being executed efficiently on the GPU.

3.3.1. Sparse Linear algebra: the DBCSR library

As a central part of CP2K, a massively parallel sparse linear algebra library, named DBCSR, has been developed. Its original purpose was linear scaling GGA DFT, enabling calculations with millions of atoms in the condensed phase[94], but it is also useful in the current context of large scale RPA calculations. This multi-year development was needed to accommodate the requirements of the application, and to exploit the internal structure of the matrices that are typical in quantum chemistry. In particular, it provides sparse matrix-matrix multiplication as a key operation, for which it exploits the fact that the sparsity pattern is blocked, with small 'atomic' block sizes that reflect the choice of basis used in the application. The design principles, in particular of its communication scheme, have been summarized in Ref. [95].

Very recently, its GPU implementation has been redesigned and optimized.[38] This implementation provides fully asynchronous operation using double buffered message passing across nodes, and between device and host. Sparse matrix indexing is handled by the host, while most flops are computed on the device, using the CPU as a fall-back. As part of this port, a CUDA library of optimized small matrix matrix multiplication kernels has been generated. This part is essential to deal with the small 'atomic' matrix sub-blocks that are typical for the application, more than 2200 different combinations of block sizes are currently available. The current simulations are characterized by blocks with dimensions that are combinations of 14, 16, and 29. Most important for these sizes is optimal memory transfer within the device, between global, shared and register memory. An off-line auto-tuning framework determines optimal parameters of several kernel variants, for example, for the $29 \times 29 \times 29$ multiplication, 36494 parameter combinations are explored. Within this heuristically pruned search space the median performance is only 160 GF. The performance of the optimal kernel found is close to the bandwidth limited peak performance with 387 GF.

3.3.2. Dense linear algebra: libsci_acc

The most important dense linear algebra step is the computation of $\mathbf{B}'(\omega)^T \mathbf{B}'(\omega)$, which is now performed as a parallel multiplication by the ScaLAPACK routine PDSYRK. The latter routine is implemented in Cray's accelerated version of ScaLAPACK, libsci_acc, and ultimately resolves to GPU accelerated DGEMM. However, to obtain good performance, careful tuning, both from within the CP2K code and libsci_acc, was necessary. This step is challenging due to the shape and size of the matrices. Indeed, for the 512 H₂O calculation, the local part of $\mathbf{B}'$ per MPI rank is close to 8 GB. To accommodate this data and libsci_acc's temporary buffers needed for MPI and GPU communication, careful memory management on CP2K's side was needed. Essential for performance is sizing internal ScaLAPACK buffers, determined by a blocking factor (PILAENV), such that all dimensions of the matrices passed to DGEMM are sufficiently large to execute efficiently on the GPU. Host-device-host transfer bandwidth was increased employing page-locked (pinned) memory for buffers and user data. Due to the very 'skinny' shape of the parent matrix (56Mx70K) the resulting data sets were too large to fit in GPU memory, necessitating the adoption of an out-of-core GPU-accelerated DGEMM. Additionally, legacy code in ScaLAPACK, which relied on 32-bit integers for size calculations, had to be fixed. The same skinny nature requires a similar 2D process layout to reach good performance. To illustrate the benefit and progress made from a 'historical' point of view, a 4.4x speedup per integration point is observed in comparing current performance on Daint (see section 5.2) vs. one year old data obtained on Titan (see section 5.1), for 256 H₂O.

4. Application: Surprising Water

Water is an abundant liquid covering nearly 70% of the surface of Earth, and constituting a roughly equal fraction of the human body. Yet the properties of water, in its liquid and fourteen other known phases, remain a hotly debated topic [96, 97, 98, 99, 100]. Indeed, even the structure of the neat liquid remains disputed, for example if molecules are four-fold coordinated, or if ordinary liquid water is actually a mixture of different liquid phases [101, 102, 103, 104, 105, 106, 107]. The pure material must be record-holding with more than 70 anomalous properties being listed, the most famous one being that ice floats on water.[108] However, going beyond the pure form of water is essential. Indeed, water is an important solvent in chemistry. As one example, the folding, unfolding and aggregation of proteins is driven

Figure 3: Left: For the first time, Monte Carlo simulations of bulk liquid water (64 H_2O) on an RPA potential energy surface can now be performed. Shown is the potential energy evolution, which is characteristic of a well equilibrated system, allowing for a detailed structural analysis. Right: Snapshot of the equilibrated sample of water, with the simulation cell shown embedded in the periodic replicas. Hydrogen bonds, essential for the structure of the liquid and well described by this level of theory, are shown with yellow lines.

by the water solvent, and the natural equilibrium is easily disrupted by changing the properties of the solvent. The role of water is even more prominent when it is active as an educt in chemical reactions, as shown by two examples. First, the process that removes the anthropogenic CO_2 from the atmosphere is dissolution in the oceans where it reacts with water to form carbonic acid $\text{CO}_{2(g)} + \text{H}_2\text{O} \rightarrow \text{HCO}_3^- + \text{H}^+$. This leads to acidification, but ultimately also to its removal by mineralization. Second, to power a future hydrogen economy, the carbon neutral approach is electrochemical water splitting, potentially making water the 'fuel' of the future.

Modeling water has therefore a long tradition. Indeed, already in 1971, soon after the invention of molecular dynamics, Rahman and Stillinger applied this method to liquid water, using a rigid, point charge based description of the molecule [109]. To describe the role of water in chemistry, however, the electronic structure needs to be taken into account. Soon after the development of DFT at its second rung, a pioneering application of the Car-Parrinello technique sampled with a few picoseconds of first-principles MD the potential energy surface of 32 water molecules using months of computing power.[110] This was a pioneering application that started a new field. The delicate nature of the system, related to the directional and subtle hydrogen bonding interaction between molecules, can however not be underestimated. Indeed, at the second rung of DFT, the structure of the liquid depends strongly on the choice of functional[111], a feature which is not clearly corrected by 4th rung functionals.[112] 4th rung functionals are needed to obtain a reasonable dielectric constant of

ice[113], or to describe the electrochemistry relevant in water splitting.[114] While functionals can be parameterised to describe medium-range dispersion-like interactions[115], even at the 4th rung, long-range van der Waals interactions, are absent, and must be added with empirical force-field like terms [4] or alternatively by including into the density functional an explicit non-local kernel.[116, 117]

Density functional theory at the 5th rung models all these effects in a non-empirical way, and is thus an essential step forward. Up to now, the computational cost of this level excluded extensive simulations. With our recent development of the RI-GPW approach[37], we were for the first time able to demonstrate the excellent performance of an approach including the virtual orbitals, more precisely MP2, for the structure and density of bulk liquid water.[5] As needed for a liquid, finite temperature sampling was employed, in our case by mean of Monte Carlo simulation that require thousands of steps,[118] which makes clear why time-to-solution is a major target for our developments. The advantage of RPA over MP2 is at least two-fold, first it is non-perturbative and thus applicable to systems with a small gap, and secondly, it displays a favorable $O(N^4)$ instead of $O(N^5)$ scaling, making it far more suitable for large scale applications. Fig. 3 demonstrates that extended Monte Carlo simulations are indeed possible, yielding a stable and well equilibrated sample. Preliminary analysis shows that both structure and density are in excellent agreement with experiment, but a full analysis will be presented elsewhere.[119] These favorable results, and the computational efficiency with which larger system can be

1
2
3 treated, will lead to a routine application of this tech-
4 nique to a broad range of systems.
5
6

7 **5. System and Environment**

8
9 All calculations presented have been performed with
10 the CP2K simulation package, on two different hybrid
11 computer architectures, namely Cray XK7 (Titan) and
12 Cray XC30 (Piz Daint). In this way, the impact of impor-
13 tant differences between these two architectures can be
14 quantified with a widely used simulation package. At the
15 time of writing, Titan and Piz Daint are the largest super-
16 computers, as measure by their linpack performance, in
17 the USA and Europe, respectively. However, for the cur-
18 rent application, network, memory and compute power
19 are all essential resources. As will be detailed below,
20 Piz Daint is 'minimal in size' to perform the largest
21 calculation presented in this manuscript.
22
23

24 *5.1. Cray XK7: Titan*

25 Part of the calculations presented in this manuscript
26 have been performed on a Cray XK7 supercomputer, lo-
27 cated at the Oak Ridge Leadership Computing Facility
28 (OLCF), named Titan. Titan currently contains 18688
29 compute nodes linked together with a high-speed inter-
30 connect. Each compute node contains one 16-core 2.2
31 GHz AMD Opteron 6274 (Interlagos) processor and 32
32 GB of RAM, and an NVIDIA Kepler K20X acceler-
33 ator with 6 GB of RAM. Two nodes share a Gemini
34 router, which connects the nodes in a three-dimensional
35 torus. The theoretical peak performance of the system
36 is approximately 27.1 Petaflop/s, with a top500 linpack
37 performance of 17.6 Petaflop/s. Its energy efficiency is
38 2.1 Gflops/W as reported in the green500 list. The total
39 host memory available is 598 TB.
40
41

42 *5.2. Cray XC30: Piz Daint*

43 Part of the calculations presented in this manuscript
44 have been performed on a Cray XC30 supercomputer, lo-
45 cated at Swiss National Supercomputing Centre (CSCS),
46 named Piz Daint. Daint currently contains 5272 compute
47 nodes linked together with a high-speed interconnect.
48 Each compute node contains one 8-core 2.6 GHz Intel
49 Xeon E5-2670 (Sandy Bridge) processor and 32 GB of
50 RAM, and an NVIDIA Kepler K20X accelerator with 6
51 GB of RAM. Four nodes share an Aries router, which
52 connects the nodes in a dragonfly topology. The rank-3
53 optical network is fully provisioned, the peak bisection
54 bandwidth is 33 TB/s. The theoretical peak performance
55 of the system is approximately 7.8 Petaflop/s, with a
56 top500 linpack performance of 6.3 Petaflop/s. It is also
57
58
59
60
61
62
63
64
65

the most energy efficient system in the linpack Top-10
(3.2 Gflops/W, as reported in the green500 list). The
total host memory available is 169 TB.

5.3. Simulation package: CP2K

The calculations in this manuscript are based on the
RI-GPW RPA method as implemented in the CP2K sim-
ulation package. CP2K is a freely available software
package for atomistic simulation.[120] It has a wide
range of capabilities, but is best known for its density
functional implementation[87], which is often combined
with molecular dynamics and Monte Carlo. A compre-
hensive overview of the major functionality can be found
in a recent review, Ref. [121]. CP2K has a large user
base and continues to generate a rich scientific output.
The web page <http://www.cp2k.org/science> provides a
selection of high profile publications that feature calcu-
lations performed with CP2K. Being fully open source,
e.g. installable as a package in common Linux distri-
butions and available at computer centers worldwide,
precise tracking of the global use is difficult. An indica-
tion is given by the fact that the CP2K web page sees
approximately 20,000 unique visitors per month. On the
HPC level, CP2K is the most used simulation code at
CSCS, and, for example, the second most used code at
the national center in the UK.

The functionality implemented in CP2K does not
come without complexity at the source level. Indeed,
the code has been actively developed by about a dozen
scientists over more than ten years, with an estimated
yearly growth in code size of about 80K lines. Porting an
application of about 1M lines to a new architecture must
be performed piece-wise, and not all code relevant for
the current calculations has been ported yet. This 'legacy'
part doesn't necessarily imply poor performance. For
example, the Hartree-Fock exchange, a key ingredient
of 4th rung functionals, was implemented more than five
years ago[122] for CPU only, but remains highly com-
petitive. For 1500 atoms (512 H₂O), using 32K cores
of Daint, evaluation of the exchange energy takes 12.5
s. This still compares favorably to a very recent imple-
mentation by a team from IBM, which required 33 s per
MD step for a slightly smaller system with more than
6M threads on an IBM BG/Q.[123]

Finally, CP2K (rev13733) has been built, with local
patches that enable memory pinning, using the GNU
tool-chain and linked against a customized version of
libsci_acc. Both modified versions will be in publically
released in the near future. All runs have been performed
with 2 MPI ranks per node, which implies four and eight
OMP threads per rank on Daint and Titan respectively.

Figure 4: Shown is key data from the simulations on Daint (green lines) and Titan (blue lines). ‘Calculations per day’ (cpd) reflect the ability to do science. At roughly 1000 cpd Monte Carlo simulations become easily possible, at 100 cpd structural relaxation is convenient, while 10 cpd allows for the verification of single point energy differences. These three important targets are easily reached for the three systems presented (64, 256, and 512 H₂O), which span a broad range of computational demands (4096x based on $O(N^4)$ scaling).

6. Performance results

6.1. Benchmark system

In order to quantify the performance, timings obtained for RI-GPW RPA energy calculations will be reported for system sizes ranging from 64 H₂O to 512 H₂O. A good quality cc-TZVP basis[37] has been employed for all calculations, and all configurations employ periodic boundary conditions with unit cells that correspond to the experimental density. 32 integration points have been used for the numerical quadrature, yielding converged results. Already the smallest system is comparable to the largest systems reported in literature, while the largest one is 4096 times larger, as estimated from the $O(N^4)$ scaling of the employed RPA algorithm. In table 1, the sizes of the basis sets and **B** matrix are reported, as well as a conservative estimate of the number of floating point operations needed.

6.2. Time to solution

For a methodology to be practical in the field of atomistic simulation, calculations must be completed with

molecules	64	512
atoms	192	1536
occupied orbitals	256	2048
correlated electrons	512	4096
virtual orbitals	3392	27136
primary basis	3648	29184
auxiliary basis	8704	69632
B dim.	0.9M × 8.7K	55.6M × 69.6K
flop	2.2P	8848.6P

Table 1: System sizes for smallest and largest system studied. The flop estimate is an underestimate as only the $O(N^4)$ terms, with their symmetries, are accounted for.

weeks, or a few months in the worst case, to meet the rapid pace of scientific progress. In the context of the first-principles field, sampling typically requires on the order 20,000 energy evaluations to reach 10 ps of MD or a similar quality MC simulation. To complete such a simulations within three weeks, an energy evaluation must take less than 1.5 minutes per step, so that roughly

1000 calculation per day (cpd) can be performed. With the presented RI-GPW RPA approach, as shown in Fig. 4, the target of 1000 cpd is already exceeded on 512 hybrid nodes of Daint for a system of 64 water molecules. This system was previously considered state-of-the-art for single energy evaluations, with 3648 and 8704 primary and auxiliary basis functions respectively. As a result, a new level of simulation methodology has become easily accessible to scientists world-wide. Pushing the limits for this 'small' system, the fastest calculation, obtained on 4096 nodes, requires only 21.3 seconds, leading to an impressive 4000+ cpd. To provide context from state-of-the-art calculations in literature, Eshuis and co-workers reported 18.9 h for a gas phase RPA calculation with 1117 basis functions, while for condensed phase local MP2 calculations Maschio and co-workers reported timings of 24 h for 2884 basis functions.[80] Yet, since the GPU accelerated $O(N^4)$ terms account for only 9.4 and 2.3 seconds on 512 and 4096 nodes, this system is too small to observe the full benefit of the large scale hybrid machines at our disposal. These large hybrid machines thus allow for increasing model size, essentially matching the models commonly used at lower rungs of DFT, a few hundred up to 1500 atoms. Such model sizes become essential to go beyond the pure substance water, for example to study interfaces, but given the $O(N^4)$ scaling of the theory represent an up to 4096x increase in computational demands. For the largest system sizes, we must therefore aim at single point energy calculations, e.g. to verify calculations performed at a lower level, while for intermediate sizes structural relaxation is the target. Defining 10 and 100 calculations per day as the threshold at which this kind of calculations is enabled, we put the bar high compared to the expectations in the field. Using the full size of either Titan or Daint, both targets can be reached for system sizes of 512 and 256 water molecules respectively. The largest system, which is with 1500 atoms in terms of flops roughly 1000x larger than literature calculations and far beyond the reported $O(100)$ electrons limit[36], can be computed in less than 1 hour.

6.3. Strong scaling

Strong scaling measures, for a fixed problem size, how the time to solution varies with respect to the computational resources. In this context, the performance of the two different architectures, Cray XC30 and Cray XK7, can be most easily compared. Indeed, this type of simulations, with strong coupling between all tasks, challenges the network. However, first we emphasize that time to solution and strong scaling, as measured by parallel efficiency, go hand in hand, but are not the same. Indeed,

running 64 waters on 64 nodes gives more than 90% parallel efficiency, but with 316 s per step would require nearly three months for a trajectory of 20,000 steps. By just doubling the computational investment (48% efficiency), the same trajectory could be obtained at 38 s per step on 1024 nodes in significantly less than two weeks, an eight-fold improvement in time to solution.

As shown in Fig. 4, Titan and Daint have rather different performance, which we attribute to the difference in network performance. Two different aspects are relevant. First, scaling out to larger node counts, the $O(N^4)$ becomes less important, and poorly scaling $O(N^3)$ aspects show up. Prominent is the diagonalization of the Kohn-Sham matrix, which at 16384 nodes is actually more expensive than the $O(N^4)$ terms combined. It does indicate that DFT at the 5th rung, has roughly reached the same time to solution as DFT at the 2nd rung, since they share this common bottleneck. Fortunately, this is an active area of research, with drop-in library replacements being available[124] and in active development[125]. The second aspect is the offset between the scaling curves at low node count. This is directly related to the performance of the PDSYRK call, it is 1.4, 1.7, and 1.8 times faster on Daint than on Titan, for 64, 256, and 512 H_2O respectively. Analysis shows that the underlying DGEMM calls require similar time, as expected since they run on the same GPU, but the MPI calls do not, with a two-fold performance advantage for the Aries over the Gemini network. The importance of data movement is a direct effect of the skinny shape of the matrix, which becomes more pronounced as system size increases. To illustrate this, the $\mathbf{B}'(\omega)^T \mathbf{B}'(\omega)$ multiplication of a 56M x 70K matrix requires the same amount of flops as that of a square 650K x 650K matrix, but requires roughly ten times more data. Hence, linpack performance is only a poor proxy for application performance.

The observed parallel efficiency is nevertheless satisfactory, for 256 H_2O , the parallel efficiency going from 512 to 4096 nodes on Daint is 78% for the overall calculation. The two $O(N^4)$ parts of the code perform at 95% (PDSYRK) and 94% (DBCSR) efficiency. The reason for this is that the parallelization over integration points is highly effective, only beyond 16384 nodes would scaling within the integration group become important. Similar strong scaling is supported by the data obtained on Titan, where 92% efficiency is observed for the $O(N^4)$ terms going from 1024 to 16384 nodes in the case of 256 H_2O , and 95% efficiency is observed in the case of 512 H_2O .

6.4. Weak scaling

Weak scaling measures, for a fixed problem size per computation unit, how the time to solution varies with

respect to the computational resources. Therefore, weak scaling tests must be performed with a constant ratio of resources to system size. For the current method, these must thus take into account the $O(N^3)$ scaling of the required memory and the $O(N^4)$ scaling of the required flops. Results are favorable. For example, 8, 512 and 4096 nodes are required to fit the calculations in memory for the respective system sizes, perfectly in line with the expected $O(N^3)$ behavior, and confirming 512 molecules is close to the limit also for Titan, since 1024 molecules would need 32768 nodes. For 512 H₂O on 4096 nodes, the user processes require per node 28 GB resident host memory and 5.5 GB device memory, emphasizing the importance of this resource for these calculations. Weak scaling for the runtime requires some extrapolation, as even the runs on Titan (8 to 16384 nodes) do not fully cover the required 4096x range. Runtimes are 6675s (extrapolated), 3359s and 3224s, for 64, 256 and 512 H₂O on 4, 1024 and 16384 nodes, respectively. This represents perfect, or even super-linear weak scaling.

6.5. GPU efficiency

Finally, the contribution of GPUs is quantified with runs on Daint with a version of the code and libraries where GPU acceleration is disabled. As only the $O(N^4)$ terms have been ported, the benefit of GPU acceleration on the application level is a moderate 1.8x for the 64 H₂O system, decreasing as the number of nodes is increased. At 256 H₂O a 3.8x speedup is measured on 512 nodes, retaining 3.5x speedup on 4096 nodes. Focussing only on the $O(N^4)$ terms, sparse and dense linear algebra show 3.5x and 5.5x speedups respectively. Based on the latter two numbers, a conservative estimate for the application level speedup for the 512 H₂O system is 4.4x on 4096 nodes.

7. Future systems and science

The calculations performed show the capabilities of current computer systems and hint at what the future will bring. Indeed, models up to 1500 atoms can currently be computed within 1h, or a thousand electrons treated within minutes. In the future, also for such models, time to solution will ultimately be pushed down to less than 1 min. To refer to Fig. 4, the curve for 256 or 512 H₂O will be extended like we did for 64 H₂O. To reach this target, a 20- or 80-fold speedup over the run on 4096 nodes of Daint will be needed, respectively, which puts this type of calculations firmly on the track to exascale. The data indicates that reaching such goals is rather realistic. Indeed, based on the scaling plots and on algorithmic

considerations, already the current combination of hard- and software is likely to show significant speedup up to 65536 nodes, and possibly beyond. Furthermore, the current combination of hard- and software will eventually be upgraded, using the insight gained from this close collaboration as feedback to all parties involved. The requirements of the application as a whole clearly call for a next generation hardware that must be balanced in memory, network and compute. Investments in the full software stack, both system libraries and user code, are required in advance of hardware deployments, to take full advantage of the advanced capabilities.

We have demonstrated the feasibility for RPA calculations of condensed phase systems for systems ranging from 64 to 512 H₂O. The smallest of these systems exceeds what is state-of-the-art for RPA calculations. Yet, most importantly, such calculations can now be performed in seconds rather than hours or days. This enables a new type of scientific calculations, where sampling of condensed phase systems at a finite temperature and using 5th rung functionals becomes easily possible. The RI-GPW RPA approach, implemented in the freely available simulation package CP2K, thus provides the community with a new atomistic simulation tool of improved accuracy and predictivity. Also the larger system sizes can be studied conveniently, so that the prospect of applying this level of theory to more complicated systems, including defects in solids, heterogeneous catalysis, interfaces, etc., is great. We can thus confidently conclude that the importance of the 5th rung of DFT will steadily increase, also in condensed phase simulations.

8. Acknowledgements

We thank Alfio Lazzaro (Cray) and Gilles Fourestey (CSCS) for their help tuning code performance on Daint, Don Maxwell (ORNL) for system support on Titan, Nikolay Markovskiy (NVIDIA) for contributions to the GPU kernels, and Adrian Tate (Cray) and Meng-Shiou Wu (Cray) for PDGEMM tuning used in an early version of the code. We acknowledge that results of this research have been achieved using the PRACE Research Infrastructure resource Hermit based in Germany at Stuttgart (HLRS), using the resources of the Swiss National Supercomputer Centre (CSCS) under project IDs s441 and ch5, and using the resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725, under a DD project ID MAT106. The research leading to these results has received funding from the Swiss University Conference through the High

Performance and High Productivity Computing (HP2C) programme and the Platform for Advanced Scientific Computing (PASC) programme. JV acknowledges financial support by the European Union FP7 in the form of an ERC Starting Grant under contract no. 277910.

- [1] P. Hohenberg, W. Kohn, *Phys. Rev.* 136 (1964) B864–B871.
- [2] W. Kohn, L. J. Sham, *Phys. Rev.* 140 (1965) A1133–A1138.
- [3] M. J. McGrath, J. I. Siepmann, I.-F. W. Kuo, C. J. Mundy, J. VandeVondele, J. Hutter, F. Mohamed, M. Krack, *ChemPhysChem* 6 (2005) 1894–1901.
- [4] J. Schmidt, J. VandeVondele, I.-F. W. Kuo, D. Sebastiani, J. I. Siepmann, J. Hutter, C. J. Mundy, *J. Phys. Chem. B* 113 (2009) 11959–11964.
- [5] M. Del Ben, M. Schönherr, J. Hutter, J. VandeVondele, *J. Phys. Chem. Lett.* 4 (2013) 3753–3759.
- [6] J. Wang, G. Román-Pérez, J. M. Soler, E. Artacho, M.-V. Fernández-Serra, *J. Chem. Phys.* 134 (2011) 024516.
- [7] Z. Ma, Y. Zhang, M. E. Tuckerman, *J. Chem. Phys.* 137 (2012) 044506.
- [8] M. Del Ben, M. Schönherr, J. Hutter, J. VandeVondele, *J. Phys. Chem. Lett.* 5 (2014) 3066–3067.
- [9] J. P. Perdew, S. Kurth, in: C. Fiolhais, F. Nogueira, M. Marques (Eds.), *A Primer in Density Functional Theory*, Springer, Berlin, 2003.
- [10] J. P. Perdew, A. Ruzsinszky, J. Tao, V. N. Staroverov, G. E. Scuseria, G. I. Csonka, *J. Chem. Phys.* 123 (2005) 062201.
- [11] T. Schwabe, S. Grimme, *Phys. Chem. Chem. Phys.* 8 (2006) 4398–4401.
- [12] L. Goerigk, S. Grimme, *J. Chem. Theory Comput.* 7 (2011) 291–309.
- [13] H. Eshuis, J. Bates, F. Furche, *Theor. Chem. Acc.* 131 (2012) 1084.
- [14] L. Schimka, J. Harl, A. Stroppa, A. Grüneis, M. Marsman, F. Mittendorfer, G. Kresse, *Nature Mat.* 9 (2010) 741–744.
- [15] Y. Zhao, B. J. Lynch, D. G. Truhlar, *J. Phys. Chem. A* 108 (2004) 4786–4791.
- [16] Y. Zhao, B. J. Lynch, D. G. Truhlar, *Phys. Chem. Chem. Phys.* 7 (2005) 43–52.
- [17] Y. Zhao, D. G. Truhlar, *J. Chem. Theory Comput.* 2 (2006) 1009–1018.
- [18] J. Zheng, Y. Zhao, D. G. Truhlar, *J. Chem. Theory Comput.* 5 (2009) 808–821.
- [19] D. C. Langreth, J. P. Perdew, *Phys. Rev. B* 15 (1977) 2884–2901.
- [20] D. Langreth, J. Perdew, *Solid State Commun.* 17 (1975) 1425–1429.
- [21] O. Gunnarsson, B. I. Lundqvist, *Phys. Rev. B* 13 (1976) 4274–4298.
- [22] M. Fuchs, X. Gonze, *Phys. Rev. B* 65 (2002) 235109.
- [23] F. Furche, T. V. Voorhis, *J. Chem. Phys.* 122 (2005) 164106.
- [24] A. Heßelmann, A. Görling, *Mol. Phys.* 109 (2011) 2473.
- [25] J. Paier, X. Ren, P. Rinke, G. E. Scuseria, A. Grüneis, G. Kresse, M. Scheffler, *New J. Phys.* 14 (2012) 043002.
- [26] X. Ren, P. Rinke, C. Joas, M. Scheffler, *J. Mater. Sci.* 47 (2012) 7447.
- [27] C. Möller, M. S. Plesset, *Phys. Rev.* 46 (1934) 618–622.
- [28] A. Szabo, N. S. Ostlund, *Modern Quantum Chemistry*, McGraw Hill, New York, 1982.
- [29] A. Görling, M. Levy, *Phys. Rev. B* 47 (1993) 13105–13113.
- [30] R. J. Bartlett, I. Grabowski, S. Hirata, S. Ivanov, *J. Chem. Phys.* 122 (2005) 034104.
- [31] S. Grimme, *J. Chem. Phys.* 124 (2006) 034108.
- [32] N. Q. Su, X. Xu, *J. Chem. Phys.* 140 (2014) 18A512.
- [33] H. Eshuis, F. Furche, *J. Chem. Phys.* 136 (2012) 084105.
- [34] J. J. Shepherd, A. Grüneis, G. H. Booth, G. Kresse, A. Alavi, *Phys. Rev. B* 86 (2012) 035111.
- [35] H. Eshuis, J. Yarkony, F. Furche, *J. Chem. Phys.* 132 (2010) 234114.
- [36] M. Kaltak, J. Klimeš, G. Kresse, *J. Chem. Theory Comput.* 10 (2014) 2498–2507.
- [37] M. Del Ben, J. Hutter, J. VandeVondele, *J. Chem. Theory Comput.* 9 (2013) 2654–2671.
- [38] O. Schütt, P. Messmer, J. Hutter, J. VandeVondele, submitted, 2014.
- [39] K. Eichkorn, O. Treutler, H. Öhm, M. Häser, R. Ahlrichs, *Chem. Phys. Lett.* 240 (1995) 283–290.
- [40] M. Feyereisen, G. Fitzgerald, A. Komornicki, *Chem. Phys. Lett.* 208 (1993) 359–363.
- [41] F. Weigend, M. Häser, H. Patzelt, R. Ahlrichs, *Chem. Phys. Lett.* 294 (1998) 143–152.
- [42] D. E. Bernholdt, R. J. Harrison, *J. Chem. Phys.* 109 (1998) 1593–1600.
- [43] H.-J. Werner, F. R. Manby, P. J. Knowles, *J. Chem. Phys.* 118 (2003) 8149–8160.
- [44] A. F. Izmaylov, G. E. Scuseria, *Phys. Chem. Chem. Phys.* 10 (2008) 3421.
- [45] L. Maschio, D. Usvyat, F. R. Manby, S. Casassa, C. Pisani, M. Schütz, *Phys. Rev. B* 76 (2007) 075101.
- [46] D. Usvyat, L. Maschio, F. R. Manby, S. Casassa, M. Schütz, C. Pisani, *Phys. Rev. B* 76 (2007) 075102.
- [47] M. Katouda, S. Nagase, *J. Chem. Phys.* 133 (2010) 184103.
- [48] Y. Jung, R. C. Lochan, A. D. Dutoi, M. Head-Gordon, *J. Chem. Phys.* 121 (2004) 9793–9802.
- [49] X. Ren, P. Rinke, V. Blum, J. Wierfink, A. Tkatchenko, A. Sanfilippo, K. Reuter, M. Scheffler, *New J. Phys.* 14 (2012) 053020.
- [50] G. E. Scuseria, T. M. Henderson, D. C. Sorensen, *J. Chem. Phys.* 129 (2008) 231101.
- [51] W. Kloppe, A. M. Teale, S. Coriani, T. B. Pedersen, T. Helgaker, *Chem. Phys. Lett.* 510 (2011) 147–153.
- [52] J. Harl, G. Kresse, *Phys. Rev. B* 77 (2008) 045136.
- [53] D. Lu, Y. Li, D. Rocca, G. Galli, *Phys. Rev. Lett.* 102 (2009) 206411.
- [54] H.-V. Nguyen, S. de Gironcoli, *Phys. Rev. B* 79 (2009) 205114.
- [55] T. Olsen, K. S. Thygesen, *Phys. Rev. B* 87 (2013) 075111.
- [56] T. Miyake, F. Aryasetiawan, T. Kotani, M. van Schilfgaarde, M. Usuda, K. Terakura, *Phys. Rev. B* 66 (2002) 245103.
- [57] P. García-González, J. J. Fernández, A. Marini, A. Rubio, *J. Phys. Chem. A* 111 (2007) 12458–12465.
- [58] J. Harl, G. Kresse, *Phys. Rev. Lett.* 103 (2009) 056401.
- [59] J. Harl, L. Schimka, G. Kresse, *Phys. Rev. B* 81 (2010) 115126.
- [60] B. Xiao, J. Sun, A. Ruzsinszky, J. Feng, J. P. Perdew, *Phys. Rev. B* 86 (2012) 094109.
- [61] L. Schimka, R. Gaudoin, J. c. v. Klimeš, M. Marsman, G. Kresse, *Phys. Rev. B* 87 (2013) 214102.
- [62] M. Rohlfing, T. Bredow, *Phys. Rev. Lett.* 101 (2008) 266106.
- [63] X. Ren, P. Rinke, M. Scheffler, *Phys. Rev. B* 80 (2009) 045402.
- [64] A. Marini, P. García-González, A. Rubio, *Phys. Rev. Lett.* 96 (2006) 136404.
- [65] F. Mittendorfer, A. Garhofer, J. Redinger, J. Klimeš, J. Harl, G. Kresse, *Phys. Rev. B* 84 (2011) 201401.
- [66] Y. Li, D. Lu, H.-V. Nguyen, G. Galli, *J. Phys. Chem. A* 114 (2010) 1944–1952.
- [67] M. Macher, J. Klimes, C. Franchini, G. Kresse, *J. Chem. Phys.* 140 (2014) 084502.
- [68] A. C. Limaye, S. R. Gadre, *J. Chem. Phys.* 100 (1994) 1303–1307.
- [69] A. M. Marquez, M. Dupuis, *J. Comput. Chem.* 16 (1995) 395–404.
- [70] I. M. B. Nielsen, E. T. Seidl, *J. Comput. Chem.* 16 (1995)

- 1301–1313.
- [71] J. Baker, P. Pulay, *J. Comput. Chem.* 23 (2002) 1150–1156.
- [72] K. Ishimura, P. Pulay, S. Nagase, *J. Comput. Chem.* 27 (2006) 407–413.
- [73] D. E. Bernholdt, R. J. Harrison, *Chem. Phys. Lett.* 250 (1996) 477–484.
- [74] D. E. Bernholdt, *Parallel Comput.* 26 (2000) 945–963.
- [75] M. Katouda, S. Nagase, *Int. J. Quantum Chem.* 109 (2009) 2121–2130.
- [76] B. Doser, D. S. Lambrecht, C. Ochsenfeld, *Phys. Chem. Chem. Phys.* 10 (2008) 3335.
- [77] E. F. Valeev, C. L. Janssen, *J. Chem. Phys.* 121 (2004) 1214–1227.
- [78] Y. Nakao, K. Hirao, *J. Chem. Phys.* 120 (2004) 6375–6380.
- [79] I. M. B. Nielsen, C. L. Janssen, *J. Chem. Theory Comput.* 3 (2007) 71–79.
- [80] L. Maschio, *J. Chem. Theory Comput.* 7 (2011) 2818–2830.
- [81] M. Katouda, T. Nakajima, *J. Chem. Theory Comput.* 9 (2013) 5373–5380.
- [82] K. Ishimura, S. Ten-no, *Theor. Chem. Acc.* 130 (2011) 317–321.
- [83] M. Del Ben, J. Hutter, J. VandeVondele, *J. Chem. Theory Comput.* 8 (2012) 4177–4188.
- [84] A. M. Burow, J. E. Bates, F. Furche, H. Eshuis, *J. Chem. Theory Comput.* 10 (2014) 180–194.
- [85] X. Ren, A. Tkatchenko, P. Rinke, M. Scheffler, *Phys. Rev. Lett.* 106 (2011) 153003.
- [86] X. Ren, P. Rinke, G. E. Scuseria, M. Scheffler, *Phys. Rev. B* 88 (2013) 035120.
- [87] J. VandeVondele, M. Krack, F. Mohamed, M. Parrinello, T. Chassaing, J. Hutter, *Comput. Phys. Commun.* 167 (2005) 103–128.
- [88] J. P. Boyd, *J. Sci. Comput.* 2 (1987) 99–109.
- [89] D. Braess, W. Hackbusch, *IMA J. Numer. Anal.* 25 (2005) 685–697.
- [90] D. Braess, *Nonlinear Approximation Theory*, Springer Series in Computational Mathematics. Springer London, Limited, 2011.
- [91] A. Takatsuka, S. Ten-no, W. Hackbusch, *J. Chem. Phys.* 129 (2008) 044112.
- [92] J. Almlöf, *Chem. Phys. Lett.* 181 (1991) 319–320.
- [93] M. Häser, J. Almlöf, *J. Chem. Phys.* 96 (1992) 489–494.
- [94] J. VandeVondele, U. Borštnik, J. Hutter, *J. Chem. Theory Comput.* 8 (2012) 3565–3573.
- [95] U. Borštnik, J. VandeVondele, V. Weber, J. Hutter, *Parallel Comput.* 40 (2014) 47–58.
- [96] E. W. Lang, H. D. Lüdemann, *Angew. Chem. Int. Ed. Engl.* 21 (1982) 315–329.
- [97] C. A. Angell, *Science* 319 (2008) 582–587.
- [98] C. A. Tulk, C. J. Benmore, J. Urquidi, D. D. Klug, J. Neufeind, B. Tomberli, P. A. Egelstaff, *Science* 297 (2002) 1320–1323.
- [99] P. H. Poole, F. Sciortino, U. Essmann, H. E. Stanley, *Nature* 360 (1992) 324–328.
- [100] M. Watkins, D. Pan, E. G. Wang, A. Michaelides, J. VandeVondele, B. Slater, *Nature Materials* 10 (2011) 794–798.
- [101] J. R. Errington, P. G. Debenedetti, *Nature* 409 (2001) 318–321.
- [102] O. Mishima, H. E. Stanley, *Nature* 396 (1998) 329–335.
- [103] P. Wernet, D. Nordlund, U. Bergmann, M. Cavalleri, M. Odelius, H. Ogasawara, L. A. Näslund, T. K. Hirsch, L. Ojamäe, P. Glatzel, L. G. M. Pettersson, A. Nilsson, *Science* 304 (2004) 995–999.
- [104] C. Huang, K. T. Wikfeldt, T. Tokushima, D. Nordlund, Y. Harada, U. Bergmann, M. Niebuhr, T. M. Weiss, Y. Horikawa, M. Leetmaa, M. P. Ljungberg, O. Takahashi, A. Lenz, L. Ojamäe, A. P. Lyubartsev, S. Shin, L. G. M. Pettersson, A. Nilsson, *Proc. Natl. Acad. Sci.* 106 (2009) 15214–15218.
- [105] J. D. Smith, C. D. Cappa, K. R. Wilson, B. M. Messer, R. C. Cohen, R. J. Saykally, *Science* 306 (2004) 851–853.
- [106] I. Ohmine, H. Tanaka, *Chem. Rev.* 93 (1993) 2545–2566.
- [107] K. Stokely, M. G. Mazza, H. E. Stanley, G. Franzese, *Proc. Natl. Acad. Sci.* 107 (2010) 1301–1306.
- [108] Chaplin M., <http://www1.lsbu.ac.uk/water/anmlies.html> (2014).
- [109] A. Rahman, F. H. Stillinger, *J. Chem. Phys.* 55 (1971) 3336–3359.
- [110] K. Laasonen, M. Sprik, M. Parrinello, R. Car, *J. Chem. Phys.* 99 (1993) 9080–9089.
- [111] J. VandeVondele, F. Mohamed, M. Krack, J. Hutter, M. Sprik, M. Parrinello, *J. Chem. Phys.* 122 (2005) 014515.
- [112] M. Guidon, F. Schiffmann, J. Hutter, J. VandeVondele, *J. Chem. Phys.* 128 (2008) 214104.
- [113] M. Schönherr, B. Slater, J. Hutter, J. VandeVondele, *J. Phys. Chem. B* 118 (2014) 590–596.
- [114] C. Adriaanse, J. Cheng, V. Chau, M. Sulpizi, J. VandeVondele, M. Sprik, *J. Phys. Chem. Lett.* 3 (2012) 3411–3415.
- [115] *Chem. Phys. Lett.* 502 (2011) 1–13.
- [116] M. Dion, H. Rydberg, E. Schröder, D. C. Langreth, B. I. Lundqvist, *Phys. Rev. Lett.* 92 (2004) 246401.
- [117] M. Dion, H. Rydberg, E. Schröder, D. C. Langreth, B. I. Lundqvist, *Phys. Rev. Lett.* 95 (2005) 109902.
- [118] D. Frenkel, B. Smit, *Understanding Molecular Simulation*, 2nd Edition, Academic Press, San Diego, 2002.
- [119] M. Del Ben, J. Hutter, J. VandeVondele, In preparation.
- [120] The CP2K developers group, CP2K is freely available from: <http://www.cp2k.org/> (2014).
- [121] J. Hutter, M. Iannuzzi, F. Schiffmann, J. VandeVondele, *Wiley Interdiscip. Rev. Comput. Mol. Sci.* 4 (2014) 15–25.
- [122] M. Guidon, J. Hutter, J. VandeVondele, *J. Chem. Theory Comput.* 5 (2009) 3010–3021.
- [123] V. Weber, V. Bekas, T. Laino, A. Curioni, <http://cpmd.org/the-code/performance-scale-out-data/PID3080165.pdf>.
- [124] T. Auckenthaler, V. Blum, H. J. Bungartz, T. Huckle, R. Johanni, L. Krämer, B. Lang, H. Lederer, P. R. Willems, *Parallel Comput.* 37 (2011) 783–794.
- [125] A. Haidar, R. Solcà, M. Gates, S. Tomov, T. Schulthess, J. Dongarra, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7905 LNCS (2013) 67–80.