Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Physics Communications 167 (2005) 103–128
Computer Physics Communications
www.elsevier.com/locate/cpc

QUICKSTEP: Fast and accurate density functional calculations using a mixed Gaussian and plane waves approach

Joost VandeVondele [a,*], Matthias Krack [b], Fawzi Mohamed [b], Michele Parrinello [b], Thomas Chassaing [c], Jürg Hutter [c]

[a] *Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge CB2 1EW, UK*
[b] *Computational Science, Department of Chemistry and Applied Biosciences, ETH Zurich, USI-Campus, via G. Buffi 13, CH-6900 Lugano, Switzerland*
[c] *Physical Chemistry Institute, University of Zurich, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland*

Received 8 October 2004; accepted 21 December 2004

Abstract

We present the Gaussian and plane waves (GPW) method and its implementation in QUICKSTEP which is part of the freely available program package CP2K. The GPW method allows for accurate density functional calculations in gas and condensed phases and can be effectively used for molecular dynamics simulations. We show how derivatives of the GPW energy functional, namely ionic forces and the Kohn–Sham matrix, can be computed in a consistent way. The computational cost of computing the total energy and the Kohn–Sham matrix is scaling linearly with the system size, even for condensed phase systems of just a few tens of atoms. The efficiency of the method allows for the use of large Gaussian basis sets for systems up to 3000 atoms, and we illustrate the accuracy of the method for various basis sets in gas and condensed phases. Agreement with basis set free calculations for single molecules and plane wave based calculations in the condensed phase is excellent. Wave function optimisation with the orbital transformation technique leads to good parallel performance, and outperforms traditional diagonalisation methods. Energy conserving Born–Oppenheimer dynamics can be performed, and a highly efficient scheme is obtained using an extrapolation of the density matrix. We illustrate these findings with calculations using commodity PCs as well as supercomputers.

PACS: 71.15.-m; 71.15.Hx; 71.15.Pd; 31.10.+z

Keywords: Density functional theory (DFT); Gaussian and plane waves method (GPW); Linear scaling electronic structure methods; Ab initio molecular dynamics

* Corresponding author.
E-mail address: jv244@cam.ac.uk (J. VandeVondele).

0010-4655/$ – see front matter
doi:10.1016/j.cpc.2004.12.014

1. Introduction

Density functional theory [1,2] (DFT) is a well established method to perform electronic structure calculations. The accuracy of the method is such that many properties of systems of interest to chemistry, physics, material science, and biology can be predicted in a parameter free way. The standard computational approach to DFT is already efficient and thus appropriate for fairly large systems, currently about 100 atoms. Nevertheless, the computation of the Hartree (Coulomb) energy and the orthogonalisation of the wave functions are not scaling linearly with system size, and these terms therefore dominate the computational cost for larger systems [3]. The hybrid Gaussian and plane waves (GPW) method [4] provides an efficient way to treat these terms accurately at a significantly reduced cost. We present here the implementation of this method in QUICKSTEP, which is part of the freely available program package CP2K [5].

The method uses an atom-centred Gaussian-type basis to describe the wave functions, but uses an auxiliary plane wave basis to describe the density. With a density represented as plane waves or on a regular grid, the efficiency of Fast Fourier Transforms (FFT) can be exploited to solve the Poisson equation and to obtain the Hartree energy in a time that scales linearly with the system size. Fast Fourier Transforms and regular grids are well established in plane wave codes [6] and their efficiency has recently been exploited in a similar method [7–11]. The use of an auxiliary basis set to represent the density goes back to the seventies [12,13] and has become increasingly popular as resolution of the identity (RI) method or density fitting method. Contrary to the GPW method, most RI methods expand the density in an auxiliary basis of the same nature as the primary basis, but optimised specifically for this purpose [14–16]. Furthermore, since a density expanded in plane waves can be represented on a real space grid, there is a direct connection to methods that use numerical calculation of matrix elements [17,18] or grid discretisation and finite element methods (for a recent review see Ref. [19]). The GPW method is most similar to methods that employ auxiliary real space grids but differ by the choice of localised primary basis functions used to represent the wave functions [20–25].

Periodic boundary conditions follow naturally from the FFT based treatment of the Poisson equation, and the GPW method scales linearly for three-dimensional systems with a small prefactor and an early onset. The GPW method seems therefore best suited for the simulation of large and dense systems, such as liquids and solids, and all recent applications of the method fall in this category [26–30]. For these systems, it is important to be able to efficiently perform stable molecular dynamics simulations, in order to address finite temperature effects. Plane wave codes and the basic GPW implementation presented here require that the nuclei are described using pseudopotential. This approximation is highly accurate if, e.g., Goedecker–Teter–Hutter (GTH) pseudopotentials are employed [31,32]. An extension of the GPW method, the Gaussian and augmented-plane-wave (GAPW) method [26,33] allows for all electron calculations.

The extensive experience with Gaussian-type basis sets shows that basis set sequences that increase rapidly in accuracy can be constructed in a systematic way [34]. At the same time, a compact description of the wave functions is maintained, and this opens the way for efficient methods to solve for the self-consistent field (SCF) equations. Furthermore, as Gaussian functions are localised, the representations of the Kohn–Sham, overlap and density matrix in this basis become sparse with increasing system size [3]. This eventually allows for solving the Kohn–Sham (KS) equations using computational resources that scale linearly with system size. We have currently only implemented methods that are scaling cubically with system size, but these have been designed to reach high efficiency for Gaussian basis sets [35].

In this paper, we provide an up-to-date description of the method and its implementation. We review the GPW energy functional and illustrate its linear scaling nature in Section 2, whereas the derivatives of the functional are described in Section 3. In Section 4 details on the program structure and implementation are provided. The construction of a sequence of systematically improving basis sets is discussed in Section 5. In Section 6, two methods to perform wave function optimisation are presented. Molecular dynamics simulations and improved wave function extrapolation method are the subject of Section 7. The accuracy of the method is illustrated for gas phase and condensed phase systems in Section 8, and the efficiency of the

code for serial and parallel calculations is shown in Section 9.

2. Gaussian and plane waves method

2.1. Energy functional

Central in the Gaussian and plane wave (GPW) method [4] is the use of two representations of the electron density. Such a dual representation allows for an efficient treatment of the electrostatic interactions, and leads to a scheme that has a linear scaling cost for the computation of the total energy and Kohn–Sham matrix with respect to the system size. The first representation of the electron density $n(\boldsymbol{r})$ is based on an expansion in atom centred, contracted Gaussian functions

$$n(\boldsymbol{r}) = \sum_{\mu\nu} P^{\mu\nu}\varphi_\mu(\boldsymbol{r})\varphi_\nu(\boldsymbol{r}), \tag{1}$$

where $P^{\mu\nu}$ is a density matrix element, and $\varphi_\mu(\boldsymbol{r}) = \sum_i d_{i\mu} g_i(\boldsymbol{r})$ with primitive Gaussian functions $g_i(\boldsymbol{r})$ and corresponding contraction coefficients $d_{i\mu}$. The second representation employs an auxiliary basis of plane waves, and is given by

$$\tilde{n}(\boldsymbol{r}) = \frac{1}{\Omega}\sum_{\boldsymbol{G}} \tilde{n}(\boldsymbol{G}) \exp(\mathrm{i}\boldsymbol{G}\cdot\boldsymbol{r}), \tag{2}$$

where Ω is the volume of the unit cell, and $\boldsymbol{G}$ are the reciprocal lattice vectors. The expansion coefficients $\tilde{n}(\boldsymbol{G})$ are such that $\tilde{n}(\boldsymbol{r})$ is equal to $n(\boldsymbol{r})$ on a regular grid in the unit cell. This choice allows for a rapid conversion between $n(\boldsymbol{r})$, $\tilde{n}(\boldsymbol{r})$ and $\tilde{n}(\boldsymbol{G})$ using an efficient mapping procedure (Section 4.3.1) and fast Fourier transforms (FFT).

Using this dual representation, the Kohn–Sham DFT energy expression [1,2] as employed within the GPW framework is defined as

$$\begin{aligned} E[n] &= E^{\mathrm{T}}[n] + E^{\mathrm{V}}[n] + E^{\mathrm{H}}[n] + E^{\mathrm{XC}}[n] + E^{\mathrm{II}} \\ &= \sum_{\mu\nu} P^{\mu\nu}\big\langle\varphi_\mu(\boldsymbol{r})\big|-\tfrac{1}{2}\nabla^2\big|\varphi_\nu(\boldsymbol{r})\big\rangle \\ &\quad + \sum_{\mu\nu} P^{\mu\nu}\big\langle\varphi_\mu(\boldsymbol{r})\big|V^{\mathrm{PP}}_{\mathrm{loc}}(r)\big|\varphi_\nu(\boldsymbol{r})\big\rangle \\ &\quad + \sum_{\mu\nu} P^{\mu\nu}\big\langle\varphi_\mu(\boldsymbol{r})\big|V^{\mathrm{PP}}_{\mathrm{nl}}(\boldsymbol{r},\boldsymbol{r}')\big|\varphi_\nu(\boldsymbol{r}')\big\rangle \\ &\quad + 2\pi\Omega\sum_{\boldsymbol{G}}\frac{\tilde{n}^*(\boldsymbol{G})\tilde{n}(\boldsymbol{G})}{\boldsymbol{G}^2} + \int e^{\mathrm{xc}}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} \\ &\quad + \frac{1}{2}\sum_{I\neq J}\frac{Z_I Z_J}{|\boldsymbol{R}_I - \boldsymbol{R}_J|}, \end{aligned} \tag{3}$$

where $E^{\mathrm{T}}[n]$ is the electronic kinetic energy, $E^{\mathrm{V}}[n]$ is the electronic interaction with the ionic cores, $E^{\mathrm{H}}[n]$ is the electronic Hartree energy and $E^{\mathrm{XC}}[n]$ is the exchange–correlation energy. The interaction energies of the ionic cores with charges Z_A and positions $\boldsymbol{R}_A$ is denoted by E^{II}. $E^{\mathrm{V}}[n]$ is described by norm-conserving pseudopotentials with a potential split in a local part $V^{\mathrm{PP}}_{\mathrm{loc}}(r)$ and a fully non-local part $V^{\mathrm{PP}}_{\mathrm{nl}}(\boldsymbol{r},\boldsymbol{r}')$.

The pseudopotential terms are described in more detail in Section 2.2, the electrostatic contributions to the total energy in Section 2.3, and the exchange and correlation term in Section 2.4. In Section 2.5 we illustrate that also for three-dimensional systems of moderate size linear scaling computational cost is observed with the GPW method.

2.2. Pseudopotentials

An expansion of an atomic all-electron density or wave function in plane waves is computationally inefficient. However, to describe a wide range of chemically interesting events, such as bond breaking and formation, an accurate description is required only for the valence electrons. Such an accurate description can be obtained using a pseudopotential description of the nuclei. This technique is well established in the plane wave community. We take advantage of the experience with this scheme, and implement the GPW method using the pseudopotentials of Goedecker, Teter, and Hutter (GTH) [31,32]. These accurate and transferable pseudopotentials (see also Section 8) have an analytic form that allows for an efficient treatment of all terms within the GPW method.

The norm-conserving, separable, dual-space GTH pseudopotentials consist of a local part including a long-ranged (LR) and a short-ranged (SR) term

$$V^{\mathrm{PP}}_{\mathrm{loc}}(r) = V^{\mathrm{LR}}_{\mathrm{loc}}(r) + V^{\mathrm{SR}}_{\mathrm{loc}}(r) \tag{4}$$

$$= -\frac{Z_{\mathrm{ion}}}{r}\operatorname{erf}(\alpha^{\mathrm{PP}}r) + \sum_{i=1}^{4} C_i^{\mathrm{PP}}(\sqrt{2}\alpha^{\mathrm{PP}}r)^{2i-2} \times \exp\big[-(\alpha^{\mathrm{PP}}r)^2\big] \tag{5}$$

with

$$\alpha^{\mathrm{PP}} = \frac{1}{\sqrt{2}\, r_{\mathrm{loc}}^{\mathrm{PP}}}$$

and a non-local part

$$V_{\mathrm{nl}}^{\mathrm{PP}}(\boldsymbol{r},\boldsymbol{r}') = \sum_{lm}\sum_{ij} \langle \boldsymbol{r}|p_i^{lm}\rangle h_{ij}^{l} \langle p_j^{lm}|\boldsymbol{r}'\rangle \tag{6}$$

with the Gaussian-type projectors

$$\langle \boldsymbol{r}|p_i^{lm}\rangle = N_i^l Y^{lm}(\hat{r}) r^{l+2i-2} \exp\left[-\frac{1}{2}\left(\frac{r}{r_l}\right)^2\right],$$

where N_i^l are normalisation constants and $Y^{lm}(\hat{r})$ spherical harmonics. The small set of GTH pseudopotential parameters ($r_{\mathrm{loc}}^{\mathrm{PP}}$, C_i^{PP}, r_l, and h_{ij}^l) have been optimised with respect to atomic all-electron wave functions as obtained from fully relativistic density functional calculations using a numerical atomic program. The optimised pseudopotentials include all scalar relativistic corrections via an averaged potential [32], and improve therefore the accuracy for applications involving heavier elements. The emphasis in the construction of these pseudopotentials has been on accuracy, and hence these pseudopotentials are computationally more demanding for plane wave methods, as a large plane wave basis typically is required. The GPW method is less sensitive to the hardness of the pseudopotential since the kinetic energy (see Eq. (3)) and the short range pseudopotential terms are computed analytically in the Gaussian basis. The long range term can be efficiently treated as part of the electrostatic energy (see Section 2.3), whereas the short range terms can be easily computed as two and three centre overlap integrals. An extended database (H–Rn) with GTH pseudopotential parameters based on the local density approximation is available [5] for use with QUICKSTEP. In addition, parameters for the common elements have been optimised for the gradient-corrected exchange–correlation potentials of Becke, Lee, Yang, and Parr (BLYP) [36–38], Becke and Perdew (BP) [36,39], Hamprecht, Cohen, Tozer and Handy (HCTH/120, HCTH/407) [40] and Perdew, Burke and Ernzerhof (PBE) [41].

2.3. Electrostatic energy

The electrostatic energy in a periodic system is defined by a conditionally converging sum in which the separate contributions of ions and electrons are infinite. All terms of the electrostatic energy are therefore treated simultaneously

$$E^{\mathrm{ES}} = \int V_{\mathrm{loc}}^{\mathrm{PP}}(r) n(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} + 2\pi\Omega \sum_{\boldsymbol{G}} \frac{\tilde{n}^*(\boldsymbol{G})\,\tilde{n}(\boldsymbol{G})}{\boldsymbol{G}^2} + \frac{1}{2}\sum_{I\neq J} \frac{Z_I Z_J}{|\boldsymbol{R}_I - \boldsymbol{R}_J|} \tag{7}$$

using the Ewald sum method [42] as it is commonly implemented in plane wave electronic structure codes [6]. The long range part of all electrostatic interactions is treated in Fourier space, whereas the short range part is treated in real space. This separation is conveniently achieved for the ionic cores if a Gaussian charge distribution ($n_{\mathrm{c}}^I(\boldsymbol{r})$) for each nucleus is introduced and defined as

$$n_{\mathrm{c}}^I(\boldsymbol{r}) = -\frac{Z_I}{(\mathrm{R}_I^{\mathrm{c}})^3}\pi^{-3/2}\exp\left[-\left(\frac{\boldsymbol{r}-\boldsymbol{R}_I}{R_I^{\mathrm{c}}}\right)^2\right], \tag{8}$$

in which the parameter R_I^{c} can be chosen for optimal performance. In QUICKSTEP, this parameter is set to

$$R_I^{\mathrm{c}} = \sqrt{2}\, r_{\mathrm{loc}}^{\mathrm{PP}} \tag{9}$$

so that the corresponding potential of the Gaussian charge distribution

$$V_{\mathrm{core}}^I(\boldsymbol{r}) = \int \mathrm{d}\boldsymbol{r}' \frac{n_{\mathrm{c}}^I(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|} = -\frac{Z_I}{|\boldsymbol{r}-\boldsymbol{R}_I|}\,\mathrm{erf}\left[\frac{|\boldsymbol{r}-\boldsymbol{R}_I|}{R_I^{\mathrm{c}}}\right], \tag{10}$$

cancels exactly the long-ranged term $V_{\mathrm{loc}}^{\mathrm{LR}}(r)$ of the local pseudopotential.

We rewrite the expression for the electrostatic energy, Eq. (7), using Eq. (8) as

$$E^{\mathrm{ES}} = \int V_{\mathrm{loc}}^{\mathrm{SR}}(r) n(\boldsymbol{r})\mathrm{d}\boldsymbol{r} + \frac{\Omega}{2}\sum_{\boldsymbol{G}} \tilde{n}_{\mathrm{tot}}^*(\boldsymbol{G}) v^{\mathrm{H}}(\boldsymbol{G}) + \frac{1}{2}\sum_{I\neq J}\frac{Z_I Z_J}{|\boldsymbol{R}_I-\boldsymbol{R}_J|}\,\mathrm{erfc}\left[\frac{|\boldsymbol{R}_I-\boldsymbol{R}_J|}{\sqrt{R_I^{\mathrm{c}2}+R_J^{\mathrm{c}2}}}\right] - \sum_I \frac{1}{\sqrt{2\pi}}\frac{Z_I^2}{R_I^{\mathrm{c}}}, \tag{11}$$

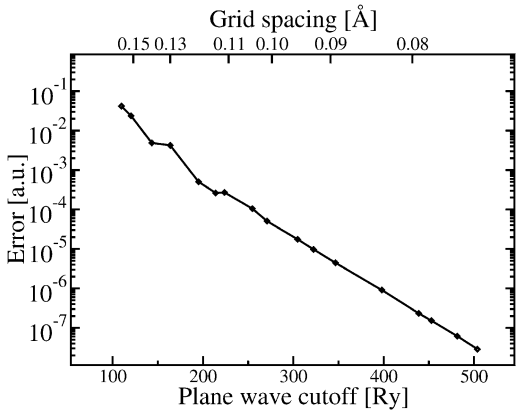

Fig. 1. Shown is the rapid convergence of the absolute error in the electrostatic energy Eq. (11) with respect to plane wave cutoff at fixed density matrix. The system is a single water molecule described with fairly hard GTH pseudopotentials and a TZV2P basis in a 10 Å cubic cell. The relation $E_{\text{cutoff}} = \frac{\pi^2}{2h^2}$ is used throughout this work to convert the grid spacing h to the corresponding plane wave cutoff.

where erfc is the complementary error function, the Hartree potential $v^{\text{H}}(\boldsymbol{G}) = 4\pi\tilde{n}_{\text{tot}}(\boldsymbol{G})/\boldsymbol{G}^2$, and a total charge distribution $\tilde{n}_{\text{tot}}(\boldsymbol{G}) = \tilde{n}(\boldsymbol{G}) + \tilde{n}_{\text{c}}(\boldsymbol{G})$ has been introduced. The last three terms of Eq. (11) define the total Hartree energy ($E^{\text{H}}[n_{\text{tot}}]$), the overlap energy ($E^{\text{ovrl}}$) and self-energy ($E^{\text{self}}$), respectively.

The two representations of the electrostatic energy Eqs. (7) and (11) are strictly equivalent if an infinite sum over $\boldsymbol{G}$ vectors is employed. In practice, a difference between the two energy expressions, due to the use of finite density grids, can be observed, but this difference is rapidly convergent with respect to $\boldsymbol{G}$, i.e. the grid spacing used. The rapid convergence of the electrostatic energy equation (11) with respect to the plane wave cutoff, and thus the size of the auxiliary basis is shown in Fig. 1.

2.4. Exchange–correlation potential

A necessary ingredient in practical applications of DFT is the introduction of an approximate exchange and correlation functional E^{XC}. In the present implementation of QUICKSTEP typical generalised gradient approximations (GGA) and meta-GGAs based on the kinetic energy density τ such as BLYP [36–38], PBE [41], HCTH [40,43,44], OLYP [45], TPSS [46] can be computed efficiently. These functionals have the common general form

$$E^{\text{XC}}[n] = \int e^{\text{XC}}\big(n_\uparrow(\boldsymbol{r}), n_\downarrow(\boldsymbol{r}), \nabla n_\uparrow(\boldsymbol{r}), \nabla n_\downarrow(\boldsymbol{r}), \tau_\uparrow, \tau_\downarrow\big)\,\mathrm{d}\boldsymbol{r}. \tag{12}$$

This form does not cover functionals where the Hartree–Fock exchange term is explicitly introduced [47,48] such as the popular B3LYP functional [49]. Currently, no implementations of Hartree–Fock exchange can approach the efficiency with which Eq. (12) can be evaluated.

To compute Eq. (12) and its derivatives with respect to the density matrix we use a discrete representation of E^{XC} on the same uniform density grid that has been used for the Hartree energy as is common in plane wave based calculations [50] and other grid based methods [51]. This avoids the use of the more accurate techniques employed within the quantum chemistry community [52–54] as these methods would, within the GPW scheme, dominate the total cost of the calculation by a relatively large factor.

For the evaluation of the exchange and correlation contribution, in the spin unpolarised case, the following operations are performed:

(1) collocation of $n(\boldsymbol{r}) = \sum_{\mu\nu} P^{\mu\nu}\varphi_\mu(\boldsymbol{r})\varphi_\nu(\boldsymbol{r})$,
(2) collocation of $\tau(\boldsymbol{r}) = \sum_{\mu\nu}(1/2)P^{\mu\nu}\nabla\varphi_\mu(\boldsymbol{r}) \cdot \nabla\varphi_\nu(\boldsymbol{r})$,
(3) numerical approximation of ∇n based on the values of $n(\boldsymbol{r})$ on the grid,
(4) evaluation of e^{xc} and its derivatives $\partial e^{\text{xc}}/\partial n$, $\partial e^{\text{xc}}/\partial|\nabla n|$, $\partial e^{\text{xc}}/\partial\tau$ on each point of the grid,
(5) computation of v_n^{xc} and v_τ^{xc} on the grid

$$v_n^{\text{xc}} = \frac{\partial e^{\text{xc}}}{\partial n} - \nabla\cdot\left(\frac{\partial e^{\text{xc}}}{\partial|\nabla n|}\frac{\nabla n}{|\nabla n|}\right), \tag{13}$$

$$v_\tau^{\text{xc}} = \frac{\partial e^{\text{xc}}}{\partial\tau}, \tag{14}$$

(6) calculation of the matrix element of the sum of v_n^{xc} and the Hartree potential $v^{\text{H}}(\boldsymbol{r})$ (see Section 2.3) between the Gaussians

$$\int\big(v_n^{\text{xc}}(\boldsymbol{r}) + v^{\text{H}}(\boldsymbol{r})\big)\varphi_\mu(\boldsymbol{r})\varphi_\nu(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}, \tag{15}$$

(7) calculation of the matrix element of v_τ^{xc} between the Gaussians

$$\int v_\tau^{\mathrm{xc}}(\boldsymbol{r})\nabla\varphi_\mu(\boldsymbol{r})\cdot\nabla\varphi_\nu(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}, \tag{16}$$

where the grid based collocation, integration and consistent differentiation are discussed in more detail in Sections 3.1 and 4.3.1.

The presence of terms such as

$$t=-\frac{|\nabla n|^2}{n^\alpha},\qquad \frac{\partial t}{\partial|\nabla n|}=-2\frac{|\nabla n|}{n^\alpha} \tag{17}$$

in GGAs and meta-GGAs leads to very sensitive behaviour in regions of vanishing density such as the tails of the atomic densities. The near singularities encountered in Eq. (17) are in that case customarily resolved by removing the contributions to e^{xc} and v^{xc} of the regions where the density n is lower than a given cutoff ϵ. In addition, care should be taken to fulfil numerically the exact relationship $|\nabla n|<8n\tau$ for functionals that depend on the kinetic energy density. However, using pseudopotentials, the density can also be small in the core region, where gradients are typically larger. This is especially true for the GTH pseudopotentials that by construction have a zero pseudocharge density at the core for all elements apart from H. We illustrate in Fig. 2 that for these pseudopotentials the core region is by far the most problematic part of the exchange and correlation potential. The pronounced spike of v^{xc} at the core gives rise to small variations in the total energy as atoms move relative to the grid.

The $\boldsymbol{G}$ space differentiation is commonly used in plane waves codes but is not the best choice with the GPW method. Whereas $\boldsymbol{G}$ space differentiation of the density on the grid yields the exact derivative $\nabla n(\boldsymbol{r})$ in the former case, the approximate $\nabla\tilde{n}(\boldsymbol{r})$ is obtained in the later case. When used, the differentiation of a small spike of $\partial e/\partial|\nabla n|$ in (13) gives rise to the strong 'ringing' effects illustrated in Fig. 2. Even though integration effectively filters out the highest frequencies, the energy oscillates significantly when the system is translated (see panel (b) of Fig. 3).

We have explored different schemes to compute the exchange and correlation energy more accurately, and to describe them we introduce a nearest neighbour smoothing operator S^q defined as

$$(S^q f)_{i,j,k} = \frac{q^3}{q^3+6q^2+12q+8}\times\sum_{l=-1}^{1}\sum_{m=-1}^{1}\sum_{n=-1}^{1} q^{-|l|-|m|-|n|} f_{i+l,j+m,k+n}, \tag{18}$$

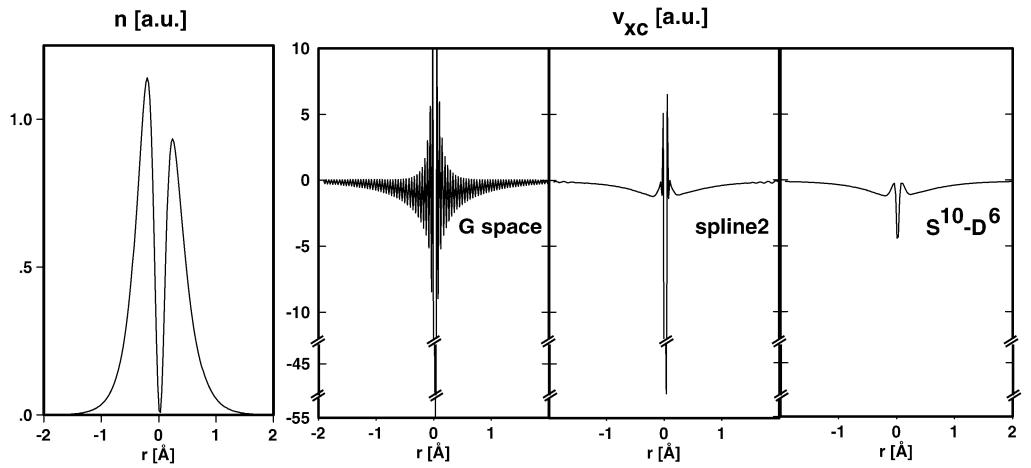

Fig. 2. Behaviour of n and v^{xc} with the BLYP functional close to the core of an O atom in a water molecule along the bisector of the HOH angle with an unusually large cutoff of 5000 Ry. The left panel shows the electron density, whereas the three right panels show v^{xc} as calculated using a derivative in $\boldsymbol{G}$ space, using a quadratic spline ($D^6(S^6)^{-1}$) and using the operators S^{10}–D^6 as defined in the text. It can be observed that the latter methods lead to a more physical exchange and correlation potential surface.

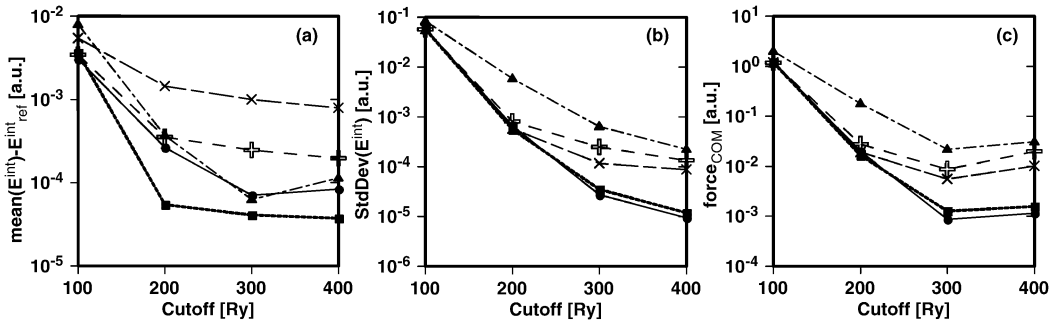

Fig. 3. The performance of the $\boldsymbol{G}$ space, $(D^6(S^6)^{-1})$, D^6, S^{50}–$D^6(S^6)^{-1}$ and S^{10}–D^6 operators as defined in the text (triangles, +, ×, squares and circles) are compared at different cutoffs. Reference calculations employed the usual $\boldsymbol{G}$ space derivative at 2000 Ry. Panel (a) shows the average systematic error in the interaction energy with the BLYP functional for water dimer configurations, panel (b) the oscillations of the interaction energy due to imperfect translational invariance, and panel (c) the magnitude of the forces on the centre of mass.

and a smoothed finite differences operator D^q that for the x derivative is

$$(D_x^q f)_{i,j,k} = \frac{q^2}{2(q^2+4q+4)} \times \sum_{l=-1}^{1}\sum_{m=-1}^{1} q^{-|l|-|m|} \times (f_{i-1,j+l,k+m} - f_{i+1,j+l,k+m}) \quad (19)$$

and likewise for the other directions.

To avoid the 'ringing' a numerical derivative that assumes less continuity can be used. $D^6(S^6)^{-1}$ calculates the derivative of the quadratic spline interpolating n on the grid. It behaves better than the $\boldsymbol{G}$ derivative, but the energy oscillations are not sufficiently reduced. D^6 alone, i.e. without sharpening step $(S^6)^{-1}$, gives information on the neighbourhood rather than on the grid point itself, and damps the oscillations more, at a cost in the accuracy of the energies (see panels (a) and (b) of Fig. 3).

For a translationally invariant evaluation of the integral of a function f over the grid points (i,j,k) it is appropriate to associate to each mesh point not the value of the function itself, but rather an estimate of its average value in a neighbourhood of (i,j,k). For a highly non-linear term such as the exchange correlation energy this average cannot easily be estimated. We therefore evaluate the xc functionals using a locally averaged density $\hat{n}(\boldsymbol{r})$ employing the smoothing operator S^q. Typical values for q are 10 or 50 depending on the required amount of smoothing. Such a smoothing is equivalent with a redefinition of E^{XC} that reduces to the identity as the cutoff is increased. $\hat{v}^{\mathrm{xc}}_{\hat{n}}$ can be calculated as function of $\hat{n} = S^q n$ as

$$v^{\mathrm{xc}}_n = \hat{v}^{\mathrm{xc}}_{\hat{n}} \frac{\delta \hat{n}}{\delta n} = S^q \hat{v}^{\mathrm{xc}}_{\hat{n}}. \quad (20)$$

It is shown in Fig. 3 that the combination of the quadratic spline and D^6 derivatives with the smoothing on n brings the oscillations of the energy and the magnitude of the forces on the centre of mass to an acceptable level for cutoffs of about 300 Ry. S^{50}–$D^6(S^6)^{-1}$ has good convergence characteristics, and implies only a small grid spacing dependent renormalisation of E^{XC}. The operator S^{10}–D^6 implies a significant amount of smoothing, resulting in even less grid dependence in the forces, and is fast to calculate since an inversion step is not necessary, but might be less appropriate to study systems where significant charge reorganisation takes place. The exchange and correlation potentials obtained with these methods are well behaved, which also helps the convergence of the SCF procedure.

Nevertheless, none of the methods presented here is fully satisfactory, as a balance between the different accuracy goals is difficult to achieve. Non-linear core corrected pseudopotentials [55] could provide a more elegant solution as the problematic region of small density would be removed. It is likely that these pseudopotentials can be treated efficiently, and they would bring additional benefits for strongly spin polarised systems. The Gaussian and augmented-plane-wave (GAPW) method [26,33] could also resolve the issues described here in a more fundamental way.

2.5. Linear scaling of the Kohn–Sham matrix construction

In order to obtain linear scaling of the Kohn–Sham matrix construction with the GPW energy functional, only two straightforward steps are needed. The first step is a screening to eliminate all negligible terms that involve a product of two Gaussian basis functions, and exploits the fact that such products are negligible once the Gaussian centres are sufficiently far apart (see also Section 4.3.1). In this way, the number of remaining non-zero terms scales linearly with system size, and this strategy is employed for most terms in Eq. (3) and in the definition of $n(\boldsymbol{r})$. To this end a list of neighbouring atoms is created for each atom.

The neighbour lists are constructed using a linked-list method in combination with a partitioning of the simulation cell into sub-cells. This is a standard technique employed by Monte-Carlo and classical molecular dynamics codes which allows to build the neighbour lists in a time scaling linearly with the number of particles in the simulation cell [56]. In QUICKSTEP, an individual sub-cell size is defined for each pair of atomic kinds in the system depending on the sum of their interaction radii, which leads to an adaptive cell partition scheme. Moreover, the work of Bekker et al. [57], in which the most inner loop for the neighbour search runs over the atoms in the central unit cell and not over the images in order to avoid the computation of the periodic boundary conditions, was generalised for arbitrary interaction ranges.

A second step is the transformation of the density $\tilde{n}(\boldsymbol{r})$ to $\tilde{n}(\boldsymbol{G})$ using Fast Fourier Transforms (FFT). This efficient step is formally $\mathrm{O}(N \ln N)$, but often still considered linear scaling since $\mathrm{O}(N \ln N) < \mathrm{O}(N^{1+\epsilon})$. We have illustrated this behaviour in Fig. 4 for samples of liquid water in periodically repeated cubic unit cells using a DZVP/DZV basis (17 basis functions per molecule), a 200 Ry cutoff, a LDA functional, a screening threshold of 10^{-10} and converging the energy to better than 10^{-12} a.u./atom using a traditional diagonalisation scheme. The timing data and the high quality of the single parameter fits presented in Fig. 4 confirm convincingly the linear scaling nature of the GPW method and its implementation. Furthermore, we can conclude that the GPW method exhibits linear scaling behaviour with a small prefactor even for three-dimensional and relatively small systems.

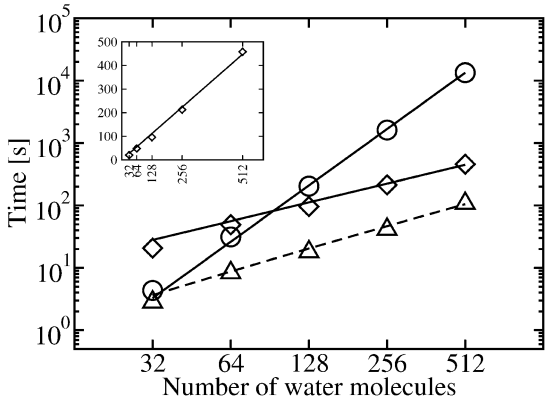

Fig. 4. Shown are the timings of a full wave function optimisation for liquid water samples of increasing size on a single Power4+ CPU using a DZVP/DZV basis. The total execution time is, to very good accuracy, the sum of the linear scaling Kohn–Sham matrix construction (diamonds) and the cubic diagonalisation/DIIS based density matrix update (circles). The solid lines are single parameter fits using the ideal forms $a_1 N$ and $a_3 N^3$, triangles indicate the fraction of time spent in the FFT routines as part of the Kohn–Sham matrix constructions, whereas the dashed line is a fit using $a_{\mathrm{FFT}} N \ln N$. The insert shows the same Kohn–Sham matrix construction data using linear axes.

3. GPW forces

3.1. Deriving the Kohn–Sham matrix from the GPW energy

In this section we present how the *exact* derivative $H_{\mu\nu} = \partial E / \partial P^{\mu\nu}$ of the total energy is computed taking into account all approximations that lead from the Gaussian based density $n(\{P^{\mu\nu}\})$ to the density represented on the grid $\tilde{n}(\{P^{\mu\nu}\})$. This includes the mapping from the Gaussian basis to the grid using finite radii, the use of multi-grids, and of grid based methods to compute $\nabla\tilde{n}(r)$. We use the notation $\tilde{n}_i$ to denote a value of $\tilde{n}(r)$ on a particular grid point with coordinates r_i using a single index i for notational convenience. Truncation to a finite radius of the products $\phi_\mu(r)\phi_\nu(r)$ is equivalent to summing over a subset $\{\mu', \nu'\}$ of $\{\mu, \nu\}$ in the definition

$$\tilde{n}_i = \sum_{\{\mu',\nu'\}} P^{\mu'\nu'} \phi_{\mu'}(r_i)\phi_{\nu'}(r_i). \tag{21}$$

The derivative of $E(\{P^{\mu\nu}\}, \tilde{n}(\{P^{\mu\nu}\}))$ will be obtained explicitly using the chain rule as

$$H_{\mu\nu} = \frac{\partial E}{\partial P^{\mu\nu}} + \frac{\partial E}{\partial \tilde{n}_i}\frac{\partial \tilde{n}_i}{\partial P^{\mu\nu}}, \tag{22}$$

where summation over repeated indices, such as i, is implicit. We refer to $\partial E/\partial\tilde{n}_i$ as the potential on the grid, and use v_i as an abbreviation.

An example term is the Slater exchange energy which we compute as

$$E_x^{\text{slater}} = \sum_j \mathrm{d}r\, C\tilde{n}_j^{4/3}, \tag{23}$$

where $\mathrm{d}r$ is a volume element, and the corresponding derivative is given as

$$\frac{\partial E_x^{\text{slater}}}{\partial\tilde{n}_i}\frac{\partial\tilde{n}_i}{\partial P^{\mu\nu}} = \sum_j \mathrm{d}r\, C\frac{4}{3}\tilde{n}_j^{1/3}\delta_{ij}\frac{\partial\tilde{n}_i}{\partial P^{\mu\nu}} = \mathrm{d}r\, v_i\frac{\partial\tilde{n}_i}{\partial P^{\mu\nu}}. \tag{24}$$

The computation of the term $\partial\tilde{n}_i/\partial P^{\mu\nu}$ needs to respect the fact that the sum in Eq. (21) is only over a subset μ', ν'. This amounts to performing the integration

$$v_i\frac{\partial\tilde{n}_i}{\partial P^{\mu\nu}}\mathrm{d}r = v_i\phi_\mu(r_i)\phi_\nu(r_i)\,\mathrm{d}r \tag{25}$$

over exactly the same grid points that have been used in the mapping of $\phi_\mu(r)\phi_\nu(r)$.

Only slightly more involved is the case where the density functional depends on the density and on the gradient of the density. The gradient is $g_j = D_j(\{\tilde{n}_i\})$, where the derivative operator D_j can be a function of all grid points, e.g., if the discrete Fourier transform is employed for computing the gradient, or, e.g., a local function of the grid for a finite difference approximation $g_j = (\tilde{n}_{j+1} - \tilde{n}_{j-1})/(2\Delta)$. Using the chain rule leads to

$$\frac{\partial E(\tilde{n}_i, g_i)}{\partial P^{\mu\nu}} = \frac{\partial E(\tilde{n}_i, g_i)}{\partial\tilde{n}_i}\frac{\partial\tilde{n}_i}{\partial P^{\mu\nu}} + \frac{\partial E(\tilde{n}_i, g_i)}{\partial g_j}\frac{\partial g_j}{\partial\tilde{n}_i}\frac{\partial\tilde{n}_i}{\partial P^{\mu\nu}}, \tag{26}$$

$$\frac{\partial E(\tilde{n}_i, g_i)}{\partial P^{\mu\nu}} = \frac{\partial E(\tilde{n}_i, g_i)}{\partial\tilde{n}_i}\frac{\partial\tilde{n}_i}{\partial P^{\mu\nu}} + \frac{\partial E(\tilde{n}_i, g_i)}{\partial g_j}\frac{\partial D_j(\{\tilde{n}_i\})}{\partial\tilde{n}_i}\frac{\partial\tilde{n}_i}{\partial P^{\mu\nu}}, \tag{27}$$

valid for all choices of D. Computing $\partial E(\tilde{n}_i, g_i)/\partial g_j$ and $\partial D_j(\{\tilde{n}_i\})/\partial\tilde{n}_i$ is the numerical equivalent of performing a partial integration.

Furthermore, we consider, as an extension of the basic scheme, a multi-grid method in which $\phi_\mu(r)\phi_\nu(r)$ is mapped to a different grid, according to the smoothness of the Gaussian product. The total density needs to be obtained on the finest grid using an operator that performs the interpolation. The value on the fine mesh is obtained as $n_j^f = I_j(\{n_i^c\})$ where the superscripts f and c imply the fine and the coarse mesh, respectively, and the interpolating operator I_j might depend on all grid points. Therefore the derivative will involve terms like

$$v_j\frac{\partial I_j(\{n_i^c\})}{\partial n_i^c}\frac{\partial n_i^c}{\partial P^{\mu\nu}}. \tag{28}$$

The index j in the above expression runs over all grid points of the fine mesh, whereas the index i goes over the coarse mesh. The term $v_j\partial I_j(\{n_i^c\})/\partial n_i^c$ can be interpreted as constructing the potential on the coarse grid.

Finally, we note that in the particular cases of a Fourier space derivative for D_j and a Fourier interpolation for I_j, it is advantageous to use a g-space representation for all densities and operators involved, as all derivatives such as $\partial D_g(\{\tilde{n}_g\})/\partial\tilde{n}_{g'}$ are diagonal (i.e. $\propto\delta_{gg'}$).

3.2. Forces on the ions

The ionic forces can be evaluated by computing explicitly the gradient of the GPW energy as defined by Eqs. (3) and (11) with respect to the atomic positions. This derivative must take the atom centred nature of the Gaussian basis set and the orthogonality constraints on the wave functions into account. We list the required derivatives in the following. These are computed analytically for all terms except for the Coulomb and exchange and correlation terms that are computed on the grids, consistent with their definition.

The derivatives of the density independent terms are given by

$$\nabla_I E^{\text{ovrl}} = \sum_{J\neq I}\frac{\boldsymbol{R}_J - \boldsymbol{R}_I}{|\boldsymbol{R}_I - \boldsymbol{R}_J|^2} \times\left\{\frac{Z_I Z_J}{|\boldsymbol{R}_I - \boldsymbol{R}_J|}\operatorname{erfc}\left[\frac{|\boldsymbol{R}_I - \boldsymbol{R}_J|}{\sqrt{R_I^{c2} + R_J^{c2}}}\right] + \frac{2}{\sqrt{\pi}}\frac{Z_I Z_J}{\sqrt{R_I^{c2} + R_J^{c2}}}\exp\left[-\frac{|\boldsymbol{R}_I - \boldsymbol{R}_J|^2}{R_I^{c2} + R_J^{c2}}\right]\right\}$$

$$\nabla_I E^{\text{self}} = 0.$$

All other terms depend directly on the density matrix $P^{\mu\nu}$, but involve only derivatives of Cartesian Gaussian functions, that can easily be computed, since these are again Cartesian Gaussian functions, but with different l quantum number. In this derivation, we follow closely Ref. [58] and introduce explicitly the derivatives $\nabla_I P^{\mu\nu}$ which are afterwards related to the orthogonality constraints on the wave function. We define E^{core} and $H^{\text{core}}_{\mu\nu}$ as the energy and matrix elements due to the electronic kinetic energy, the short range part of the local pseudopotential, and the non-local pseudopotential to obtain

$$\begin{aligned}
\nabla_I E^{\text{core}} &= \sum_{\mu\nu} (\nabla_I P^{\mu\nu}) H^{\text{core}}_{\mu\nu} + \sum_{\mu\nu} P^{\mu\nu} (\nabla_I H^{\text{core}}_{\mu\nu}) \\
&= \sum_{\mu\nu} (\nabla_I P^{\mu\nu}) H^{\text{core}}_{\mu\nu} \\
&\quad + \sum_{\mu\nu} P^{\mu\nu} \big[2 \big\langle \nabla_I \varphi_\mu(\boldsymbol{r}) \big| - \tfrac{1}{2} \nabla^2 \big| \varphi_\nu(\boldsymbol{r}) \big\rangle \\
&\quad + 2 \big\langle \nabla_I \varphi_\mu(\boldsymbol{r}) \big| V^{\text{SR}}_{\text{loc}}(r) \big| \varphi_\nu(\boldsymbol{r}) \big\rangle \\
&\quad + 2 \big\langle \nabla_I \varphi_\mu(\boldsymbol{r}) \big| V^{\text{PP}}_{\text{nl}}(\boldsymbol{r}, \boldsymbol{r'}) \big| \varphi_\nu(\boldsymbol{r'}) \big\rangle \\
&\quad + \big\langle \varphi_\mu(\boldsymbol{r}) \big| \nabla_I V^{\text{SR}}_{\text{loc}}(r) \big| \varphi_\nu(\boldsymbol{r}) \big\rangle \\
&\quad + \big\langle \varphi_\mu(\boldsymbol{r}) \big| \nabla_I V^{\text{PP}}_{\text{nl}}(\boldsymbol{r}, \boldsymbol{r'}) \big| \varphi_\nu(\boldsymbol{r'}) \big\rangle \big], \qquad (29)
\end{aligned}$$

where the number of terms is already reduced by regrouping terms, exploiting symmetry of $P^{\mu\nu}$ and $H^{\text{core}}_{\mu\nu}$. Furthermore, the translational invariance of the first derivatives is exploited for the force calculation in QUICKSTEP using identities such as e.g.

$$\begin{aligned}
&\big\langle \nabla_I \varphi_\mu(\boldsymbol{r}) \big| - \tfrac{1}{2} \nabla^2 \big| \varphi_\nu(\boldsymbol{r}) \big\rangle \\
&\quad = - \big\langle \varphi_\mu(\boldsymbol{r}) \big| - \tfrac{1}{2} \nabla^2 \big| \nabla_J \varphi_\nu(\boldsymbol{r}) \big\rangle, \\
&\big\langle \nabla_I \varphi_\mu(\boldsymbol{r}) \big| V(\boldsymbol{r}) \big| \varphi_\nu(\boldsymbol{r}) \big\rangle + \big\langle \varphi_\mu(\boldsymbol{r}) \big| V(\boldsymbol{r}) \big| \nabla_J \varphi_\nu(\boldsymbol{r}) \big\rangle \\
&\quad = - \big\langle \varphi_\mu(\boldsymbol{r}) \big| \nabla_K V(\boldsymbol{r}) \big| \varphi_\nu(\boldsymbol{r}) \big\rangle,
\end{aligned}$$

where $\varphi_\mu(\boldsymbol{r})$, $\varphi_\nu(\boldsymbol{r})$ and $V(\boldsymbol{r})$ are located on the atoms I, J and K, respectively.

The density dependent energy terms are computed using the chain rule with the density as an intermediate variable to yield the following derivatives

$$\begin{aligned}
&\nabla_I E^{\text{H}}[n_{\text{tot}}] + \nabla_I E^{\text{xc}}[n] \\
&\quad = \sum_{\mu\nu} (\nabla_I P^{\mu\nu}) V^{\text{tot}}_{\mu\nu} \\
&\qquad + 2 \sum_{\mu\nu} P^{\mu\nu} \int \big(\nabla_I \varphi_\mu(\boldsymbol{r}) \big) v^{\text{tot}}(\boldsymbol{r}) \varphi_\nu(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r} \\
&\qquad + \int \big(\nabla_I n^I_{\text{c}}(\boldsymbol{r}) \big) v^{\text{H}}(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r}, \qquad (30)
\end{aligned}$$

where $v^{\text{tot}}(\boldsymbol{r}) = v^{\text{H}}(\boldsymbol{r}) + v^{\text{xc}}(\boldsymbol{r})$.

In the above equations, the terms involving $\nabla_I P^{\mu\nu}$ can be collected and rewritten using the Kohn–Sham matrix $K_{\mu\nu}$ as

$$\sum_{\mu\nu} \big(\nabla_I P^{\mu\nu} \big) \big(H^{\text{core}}_{\mu\nu} + V^{\text{tot}}_{\mu\nu} \big) = \sum_{\mu\nu} \big(\nabla_I P^{\mu\nu} \big) K_{\mu\nu}. \quad (31)$$

The derivative of the density matrix can be eliminated by expanding the density matrix in terms of the wave function coefficients, inserting the Kohn–Sham equations (Eq. (32)), and simplifying the expression using the derivative of the orthogonality constraints on the wave functions (Eq. (33)).

$$\sum_\nu K_{\mu\nu} c^\nu_i = \varepsilon_i \sum_\nu S_{\mu\nu} c^\nu_i, \qquad (32)$$

$$\begin{aligned}
&\nabla_I \sum_{\mu\nu} c^\mu_i S_{\mu\nu} c^\nu_i = 0, \\
&2 \sum_{\mu\nu} \big(\nabla_I c^\mu_i \big) S_{\mu\nu} c^\nu_i = - \sum_{\mu\nu} c^\mu_i (\nabla_I S_{\mu\nu}) c^\nu_i.
\end{aligned} \qquad (33)$$

This leads to

$$\begin{aligned}
&\sum_{\mu\nu} (\nabla_I P^{\mu\nu}) \, K_{\mu\nu} \\
&\quad = \sum_{\mu\nu} \sum_i^{\text{occ}} \big[\big(\nabla_I c^\mu_i \big) K_{\mu\nu} c^\nu_i + c^\mu_i K_{\mu\nu} \big(\nabla_I c^\nu_i \big) \big] \\
&\quad = 2 \sum_{\mu\nu} \sum_i^{\text{occ}} \varepsilon_i \big(\nabla_I c^\mu_i \big) S_{\mu\nu} c^\nu_i \\
&\quad = - \sum_{\mu\nu} \sum_i^{\text{occ}} \varepsilon_i c^\mu_i c^\nu_i (\nabla_I S_{\mu\nu}) \\
&\quad = - \sum_{\mu\nu} W^{\mu\nu} (\nabla_I S_{\mu\nu}) \\
&\quad = -2 \sum_{\mu\nu} W^{\mu\nu} \big\langle \nabla_I \varphi_\mu(\boldsymbol{r}) \big| \varphi_\nu(\boldsymbol{r}) \big\rangle, \qquad (34)
\end{aligned}$$

where the energy weighted density matrix $W^{\mu\nu}$ is introduced. It can be observed that this force contribution is easily calculated as it only involves derivatives of overlap matrix elements. This term was originally derived by Pulay [58] and is only present if the basis set is atom position dependent.

4. Program structure and implementation

QUICKSTEP is tightly integrated in a larger program named CP2K and the Fortran 95 sources are made freely available [5]. Currently, good portability and efficiency is achieved by relying on a few commonly available libraries such as BLAS [59], LAPACK [60], FFTW [61], MPI [62] and ScaLAPACK [63], and restricting ourselves to standard Fortran with OpenMP [64] extensions. In this section, we present our approach to structural aspects of the code, and describe how good efficiency can be obtained for collocating and integrating Gaussian products on a grid.

4.1. Algorithm environments

In order to deal with the growing complexity of the code in which, e.g., all algorithms must be able to run concurrently and interfaces evolve over time, a more object-oriented approach is employed in CP2K. Each major algorithm has an associated 'environment', i.e. a derived type that contains all necessary data structures to allow for abstracting and simplifying the interfaces and splitting the algorithms. Access to components of the environment is through get/set functions. In this way, a certain abstraction can be introduced so that it is, e.g., possible to hide the actual method used to compute the forces on the atoms (force_env), and provide generic modules that perform the molecular dynamics or geometry optimisation independent on whether forces and energies are calculated ab-initio, classically, or in some other way. The various phases of the algorithm are well separated as follows:

(1) creation of a computational environment c_env, initialising to default values for the components,
(2) optionally, set components of c_env to adapt to a specific task,
(3) call the routine that actually performs the algorithm with a simple interface and passing c_env as parameter,
(4) call the cleanup routine that destroys c_env.

The points (2)–(3) can be repeated more than once, and the point (3) can easily be further split into smaller subroutines. This pattern is quite widespread in the code (qs_env for the whole DFT part, scf_env for the SCF cycle in QUICKSTEP, ks_env for the final assembly of the Kohn–Sham matrix, ...).

4.2. Memory management

Two techniques are employed to address the memory requirements in QUICKSTEP. To safely share instead of copying fairly large objects such as, e.g., descriptors for plane wave grids or matrices, reference counting is employed. In this way, a variable within the objects is used to track how many pointers to the object are in use. Whenever a pointer to the object is assigned (i.e. the object is retained), this counter is increased, and the counter is decreased as soon as the object is released. The object is automatically deallocated as soon as the counter becomes zero.

The second technique is aimed at reducing memory fragmentation and time spent in dynamically allocating and deallocating large objects. Static allocation relies on the knowledge that at a given stage of an algorithm a variable can be overwritten and makes it difficult to change the algorithm, or to allow for variants that use a different sequence of operations. Pools have been introduced as a solution to this problem. Temporarily unused objects are given to a pool that stores these objects in a cache, and objects are created rapidly if the pool can return such a cached element.

4.3. Parallelisation strategy

In order to make a program scale in parallel it is necessary to distribute at least part of the data and tasks between the different processors, without introducing a large amount of communication between the different CPUs. There are currently two layers of parallelisation in CP2K. A standard message passing interface that is available on most parallel machines (MPI [62]) provides the main layer, and this layer is augmented using an application program interface that supports shared-memory parallel programming (OpenMP [64]). In our applications, OpenMP is typically used in addition to MPI with just a few threads per MPI task. The MPI tasks can be addressed through a one-dimensional (1d) rank $n = 0..N-1$ or through two-dimensional (2d) coordinates $(p, q) = (0..P-1, 0..Q-1)$ with $N = P \times Q$.

Most data related to the structure of the whole system, such as lists describing the molecules, atomic

kinds, coordinates, layout of the grids and matrices are replicated. A 1d mapping d that given the atom number i returns the 1d rank $n = d(i)$ of the task that should hold/work on the data is typically used for atom related operations, such as, e.g., computing $\tilde{n}_c(\boldsymbol{r})$.

A two-dimensional mapping that given the indices of the atom pair (i, j) returns the 2d coordinates (p, q) of the task that owns that pair is employed for operations that should be performed on pairs of atoms, e.g., from computing the interaction between atoms i and j. This 2d distribution is built up from two 1d distributions d_1, d_2 distributing over processor rows and columns, respectively. For $i \leqslant j$ the couples (i, j) and (j, i) are local to the processor (p, q) with

$$\begin{aligned} &p = d_1(i) \quad \text{and} \quad q = d_2(j) \\ &\quad \text{if } (i + j \bmod 2 = 0), \\ &p = d_1(j) \quad \text{and} \quad q = d_2(i) \\ &\quad \text{if } (i + j \bmod 2 = 1). \end{aligned} \tag{35}$$

The d_1 and d_2 distributions keep in account the size of the basis of the atoms, and try to build a well balanced 2d distribution. The storage required for the full 1d distribution is small, so that replicating this data enables each processor to calculate rapidly the location of any (i, j) pair. The atomic blocks of the sparse matrices with the same structure as the overlap matrix are distributed using this 2d distribution.

All dense matrix linear algebra is based on ScaLAPACK [63] and thus fully parallelised. These full matrices, e.g., used to represent the coefficients of the orbitals, or the Kohn–Sham matrix during the diagonalisation are distributed using the format imposed by ScaLAPACK, which is block cyclic in both dimensions with blocks of constant size [65]. The sparsity of the overlap and Kohn–Sham matrix is however exploited in the orbital transformation method (Section 6) and an efficient matrix multiply routine has been implemented to multiply a sparse matrix with a dense matrix (i.e. overlap/Kohn–Sham matrix and orbital coefficients). This implementation provides uniformly good performance for the cases where the sparsity of the S matrix is nearly absent or on the order of 10% as is encountered for the largest 3d systems (see also Table 5). The efficiency comes from exploiting the atomic block structure of S, which allows for relatively high peak performance, limiting the number of matrix multiplies by skipping zero atomic blocks, and distributing the blocks according to Eq. (35) which allows for reducing the communication for the orbital coefficients by a factor of $\min(P, Q)$ without introducing the need for communicating blocks of the sparse matrix.

The usual z-slices distribution is employed for plane waves grids, and the parallelisation of the FFT follows the implementation used in CPMD [6,66]. The grids used for the collocation are either fully replicated or are distributed in slices with overlapping borders. These overlapping borders allow for collocating the Gaussian products without the need for treating boundaries in a special manner during the collocation. Currently, the parallelisation of this step appears to be the bottleneck to obtain good scaling on hundreds of CPUs.

4.3.1. *Efficient mapping of product Gaussians on the real space grid*

The transformation of the atomic orbital based density matrix to an electron density on a real space grid, and the complementary operation, i.e. the computation of matrix elements given a potential on a real space grid are important operations in the GPW method. This step needs to be performed at every SCF cycle, and might contribute significantly to the overall execution time of the algorithm. However, a Gaussian basis set allows for techniques that reduce the cost of this operation significantly if regular orthogonal grids are employed. The described algorithm has a computational cost for computing integrals involving Cartesian Gaussian that scales effectively linearly in the l quantum number.

A first step in the algorithm is the transformation of the atomic orbital basis into primitive Cartesian Gaussians

$$\begin{aligned} g_{\eta l^x l^y l^z}(r) = x^{l^x} \exp(-\eta x^2) y^{l^y} \exp(-\eta y^2) z^{l^z} \\ \times \exp(-\eta z^2). \end{aligned} \tag{36}$$

The product of two primitive Gaussians $g_{\eta_a l_a^x l_a^y l_a^z}(r - A)$ and $g_{\eta_b l_b^x l_b^y l_b^z}(r - B)$ is a Cartesian factor multiplied by a single Gaussian with centre

$$P = \frac{\eta_a A + \eta_b B}{\eta_a + \eta_b}, \tag{37}$$

exponent $\eta_p = \eta_a + \eta_b$, and prefactor

$$\exp\left(-\frac{\eta_a \eta_b}{\eta_a + \eta_b} |A - B|^2\right). \tag{38}$$

Based on this exponent, the total l quantum number, and the prefactor, the radius R of a sphere around P is computed where the Gaussian product is non-negligible to within some threshold, typically 10^{-10}–10^{-14}. Only grid points within this sphere are used in the integration and mapping. As shown in Section 3.1, consistency between the Kohn–Sham matrix and the GPW energy can be maintained.

It is exploited that the product can be factorised in three parts that each depend only on a single variable x, y or z. Each part can be precomputed on a 1d grid, the full value of the Gaussian product on any point of the 3d grid is a product of 3 precomputed numbers. In particular, the inner loop on the grid (e.g., x-component) will just be a single multiplication of a constant (depending on y and z) with number stored in a 1d array (independent of y and z). Furthermore, all products that have the same centre, independent of the l quantum number, can be computed simultaneously by expanding for each term the factors $(x-A)^{l_a^x}(x-B)^{l_b^x}$ and summing the prefactors of identical terms. The inner loop contains in that case just $l_a^x + l_b^x + 1$ terms, each one can be precomputed as described above. Therefore $O(l_a^3 l_b^3)$ terms are computed in approximately $O(l_a + l_b)$ time, contributing greatly to the efficiency of the algorithm. In particular, basis sets involving high angular momentum are relatively inexpensive, even if derivatives for, e.g., the kinetic energy density or the forces are required.

As described previously [33], multigrids, i.e. grids with different mesh sizes, are employed to compute matrix elements and densities. We describe in Section 3.1 how to define consistently the Kohn–Sham matrix if multigrids are introduced in the GPW energy functional. In the multigrid method, the exponent η_p of the Gaussian product is used to select a grid so that the number of grid points per $\sigma_p^2 = 1/2\eta_p$ is approximately independent of η_p. The accuracy of the multigrid method is fixed specifying a single number (number of grid points per σ_p) which is conveniently expressed as the plane wave cutoff of the grid used for Gaussians with $\eta_p = 1$. We employ 30 Ry as a default relative cutoff as this is both efficient and accurate. Furthermore, it is not necessary to have multigrids for each exponent, and in our experience it is sufficient to have N grids with

$$E_{\text{cut}}^i = \frac{E_{\text{cut}}^1}{\alpha^{(i-1)}}, \quad i = 1..N, \tag{39}$$

where $\alpha = 3.0$. The necessary N and E_{cut}^1 depend on the smallest and largest exponents of the Gaussian basis set used, but are typically four and 280 Ry, respectively.

5. Basis sets

The Kohn–Sham orbitals are expanded in Gaussian orbital functions in the framework of the GPW method as described in Section 2. Significant experience exists with Gaussian basis sets and they are available in a number of formats [67,68]. Whereas polarisation and diffuse functions can normally be adopted from published basis sets, the valence part of the basis has to be generated for the usage with the GTH pseudopotentials. A systematically improving sequence of basis sets for use with the GTH pseudopotentials was optimised for all first- and second-row elements, using the procedure detailed below.

Exponents of a set of primitive Gaussian functions were optimised to yield the lowest pseudoatom energies for all first- and second-row elements with an atomic DFT code employing the appropriate GTH potential for each element. The atomic DFT code allows for the calculation of first analytic derivatives of the total atomic energy with respect to the Gaussian orbital exponents. A family basis set scheme was adopted using the same set of exponents for each angular momentum quantum number of the occupied valence states, i.e. s and p orbitals for the elements from H to Ar. A growing number of primitive Gaussian functions, typically four to six, was included into these sets to provide an increasingly good description of the pseudoatomic wave function. Finally, these primitive Gaussian functions were contracted using the coefficients of the respective pseudoatomic wave functions. In addition, a split valence scheme was applied to enhance the flexibility of the valence basis part. The splitting was increased in line with the number of primitive Gaussian functions employed from double- (DZV) over triple- (TZV) up to quadruple-zeta valence (QZV). For instance, the basis set sequence for oxygen starts with four primitive Gaussian functions on the DZV level, uses five functions for TZV, and finally six on the QZV level. Moreover, these basis sets were augmented by polarisation functions which were taken from the all-electron basis sets cc-pVXZ

(X = D, T, Q) of Dunning [34,69], but only the first p or d set of polarisation functions was used depending on the actual element. In that way a new sequence of basis sets was created with an increasing number of primitive Gaussian functions and polarisation functions for each first- and second-row element. The basis sets were labelled DZVP, TZVP, TZV2P, QZV2P, and QZV3P according to the applied degree of splitting and the increasing number of provided polarisation functions. As will be shown in Section 8, the quality of the basis improves systematically within this sequence. If required, these basis sets can be further augmented by diffuse functions, analogous to the aug-cc-pVXZ basis sets, resulting in a sequence aug-DZVP, aug-TZVP, aug-TZV2P, aug-QZV2P, and aug-QZV3P. The inclusion of diffuse functions may significantly improve the accuracy of certain molecular properties, especially in the gas phase, but require treatment for the near linear dependencies that typically arise in condensed phase calculations (see Section 6). We will therefore not present any results from calculations involving diffuse functions in the following.

6. Wavefunction optimisation

The calculation of the electronic ground state amounts to the minimisation of the electronic energy (Eq. (3)) with respect to the orthonormal one-particle orbitals or the one-particle density matrix. Even though techniques exist to perform this minimisation in a time scaling linearly with the system size [3], no such technique is currently implemented in QUICKSTEP, and the algorithms show a cubic scaling with respect to the number of atoms in the unit cell. Nevertheless, a careful design of the methods can make them, especially on parallel computers, very efficient for, e.g., molecular or condensed phase systems of up to a few thousand atoms (see Section 9). Two methods are currently available in QUICKSTEP to minimise the total ground state energy of a system by an iterative self-consistent field (SCF) procedure: a traditional diagonalisation (TD) scheme and an efficient orbital transformation (OT) method [35]. In the following, the description is restricted to closed-shell systems, however, the generalisation to open-shell (spin-polarised) systems is straightforward and QUICKSTEP can deal with both types of systems using both the TD and the OT method.

6.1. Traditional diagonalisation

The TD scheme uses an eigensolver from a standard parallel program library (ScaLAPACK) to solve the general eigenvalue problem

$$\boldsymbol{K}\boldsymbol{c} = \boldsymbol{S}\boldsymbol{c}\epsilon, \tag{40}$$

where $\boldsymbol{K}$ is the Kohn–Sham matrix and $\boldsymbol{S}$ is the overlap matrix. The eigenvectors $\boldsymbol{c}$ represent the orbital coefficients and the ϵ are the corresponding eigenvalues. Unfortunately, the overlap matrix $\boldsymbol{S}$ is not the unit matrix, since QUICKSTEP employs an atom-centred basis set of non-orthogonal Gaussian-type orbital functions. The eigenvalue problem is therefore transformed from the generalised to the standard form

$$\boldsymbol{K}\boldsymbol{c} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{c}\epsilon, \tag{41}$$

$$(\boldsymbol{U}^{\mathrm{T}})^{-1}\boldsymbol{K}\boldsymbol{U}^{-1}\boldsymbol{c}' = \boldsymbol{c}'\epsilon \quad (\texttt{pdsygst}), \tag{42}$$

$$\boldsymbol{K}'\boldsymbol{c}' = \boldsymbol{c}'\epsilon \quad (\texttt{pdsyevx} \text{ or } \texttt{pdsyevd}) \tag{43}$$

using a Cholesky decomposition of the overlap matrix

$$\boldsymbol{S} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} \quad (\texttt{pdpotrf}) \tag{44}$$

as the default method for that purpose. Eq. (43) can be solved by diagonalisation of $\boldsymbol{K}'$, and the orbital coefficients $\boldsymbol{c}$ in the non-orthogonal basis are obtained by back transformation

$$\boldsymbol{c}' = \boldsymbol{U}\boldsymbol{c} \quad (\texttt{pdtrsm}). \tag{45}$$

The names in brackets denote the employed ScaLAPACK routines.

Alternatively, a symmetric orthogonalisation instead of a Cholesky decomposition can be applied by using

$$\boldsymbol{U} = \boldsymbol{S}^{1/2}. \tag{46}$$

However, the calculation of $\boldsymbol{S}^{1/2}$ involves a diagonalisation of $\boldsymbol{S}$ which is computationally more expensive than a Cholesky decomposition. On the other hand, linear dependencies in the basis set introduced by small Gaussian function exponents can be detected. Eigenvalues of $\boldsymbol{S}$ smaller than a threshold value (usually 10^{-5}) indicate significant linear dependencies in

the basis set and a filtering of the corresponding eigenvectors might help to circumvent numerical difficulties if they arise during the SCF iteration procedure. Both orthogonalisation schemes are implemented in QUICKSTEP. For small systems the choice of the orthogonalisation has no crucial impact on the performance, since it has to be performed only once for each configuration during the initialisation of the SCF run. By contrast, the eigenvectors and eigenvalues of the full Kohn–Sham matrix $\boldsymbol{K}'$ have to be calculated in each iteration step as indicated by Eq. (43) using a divide-and-conquer (`pdsyevd`) scheme or an expert driver (`pdsyevx`) which allows to request only the build of an eigenvector sub-set. The divide-and-conquer scheme is faster than the expert driver if all eigenvectors have to be computed. However, for the construction of the new density matrix

$$\boldsymbol{P} = 2\boldsymbol{c}_{\mathrm{occ}}\boldsymbol{c}_{\mathrm{occ}}^{\mathrm{T}} \tag{47}$$

only the occupied orbitals are needed. In that case the expert driver is superior, since for standard basis sets only 10–20% of the orbitals are occupied.

The TD scheme is usually combined with methods to improve the convergence of the SCF iteration procedure. The most efficient SCF convergence acceleration is achieved by the direct inversion in the iterative subspace (DIIS) [70,71] exploiting the commutator relation

$$\boldsymbol{e} = \boldsymbol{KPS} - \boldsymbol{SPK} \tag{48}$$

between the Kohn–Sham and the density matrix where the error matrix $\boldsymbol{e}$ is zero for the converged density. The TD/DIIS scheme is an established method for electronic structure calculations. The DIIS method can be very efficient in the number of iterations required to reach convergence starting from a sufficiently preconverged density which is significant if the cost of constructing the Kohn–Sham matrix is larger than the cost of diagonalisation. Nevertheless, the cost for the TD/DIIS scales as $\mathrm{O}(M^3)$, where M is the size of the basis set. This implies that, even at fixed system size, increasing the number of basis functions results in a cubic growth of the computational cost. A further disadvantage of the DIIS is that the method might fail to converge or that a sufficiently pre-converged density cannot be obtained. This happens more frequently for electronically difficult systems such as, e.g., spin-polarised systems with a small energy gap between the highest occupied and the lowest unoccupied orbital.

6.2. Orbital transformation method

The OT method is a direct minimisation method that addresses both deficiencies of the TD/DIIS scheme, as the method is guaranteed to converge, and scales, depending on the preconditioner, as $\mathrm{O}(MN^2)$. As described in detail in Ref. [35], the OT method parametrises the MO coefficients $\boldsymbol{c}$ using new variables $\boldsymbol{x}$ as

$$\boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{c}_0\cos(\boldsymbol{U}) + \boldsymbol{x}\boldsymbol{U}^{-1}\sin(\boldsymbol{U}), \tag{49}$$

where the matrix $\boldsymbol{U} = (\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x})^{1/2}$, $\boldsymbol{x}$ satisfies the linear constraint $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{c}_0 = \boldsymbol{0}$ and $\boldsymbol{c}_0$ are constant initial vectors that satisfy $\boldsymbol{c}_0^{\mathrm{T}}\boldsymbol{S}\boldsymbol{c}_0 = \boldsymbol{I}$. As the constraint on $\boldsymbol{x}$ is linear, standard optimisers such as DIIS or conjugate gradients can be employed. For a robust minimiser, e.g., conjugate gradients in combination with line search, convergence is guaranteed, thus leading to a robust electronic structure method. The computational cost of the OT method is normally dominated by the computation of the $\mathrm{O}(MN)$ terms $\boldsymbol{Hc}$ and $\boldsymbol{Sx}$, but is in principle $\mathrm{O}(MN^2)$ with a sparse preconditioner, and $\mathrm{O}(M^2N)$ if a non-sparse preconditioner is used. In Ref. [35] it was shown how to compute the matrix functionals $\cos(\boldsymbol{U})$, $\boldsymbol{U}^{-1}\sin(\boldsymbol{U})$ and the derivatives with respect to $\boldsymbol{x}$ based on diagonalisation of the $N \times N$ matrix $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x}$. We have found that the performance of the $N \times N$ diagonalisation on a parallel computer can be such that an alternative method based on expanding the matrix functionals is beneficial. In this method, the matrix functionals are Taylor expanded as:

$$\cos(\boldsymbol{U}) = \sum_{i=0}^{K} \frac{(-1)^i}{(2i)!}(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x})^i, \tag{50}$$

$$\boldsymbol{U}^{-1}\sin(\boldsymbol{U}) = \sum_{i=0}^{K} \frac{(-1)^i}{(2i+1)!}(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x})^i. \tag{51}$$

The appropriate order K of the Taylor expansion is determined at every step by estimating an upper bound of the eigenvalue spectrum of $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{x}$ using the maximum absolute row sum norm. Note that each term is quadratic in $\boldsymbol{x}$, and that low order Taylor expansion (e.g., $K = 2, 3$) can be accurate to machine precision,

Table 1
Shown is the execution time in seconds of a single diagonalisation using ScaLAPACK routines, and the time needed by the OT routines for one SCF iteration. The calculations were performed on 32 CPUs of an IBM SP4. A sample of 64 H_2O molecules has been employed, and the basis sets DZVP, TZVP, TZV2P, QZV2P, and QZV3P result in a total of 1472, 1856, 2560, 2944, and 3648 basis functions, respectively

	DZVP	TZVP	TZV2P	QZV2P	QZV3P
OT	0.50	0.60	0.77	0.87	1.06
Diagonalisation	6.02	8.40	13.80	17.34	24.59

in particular, if a good initial guess $\boldsymbol{c}_0$ can be generated. This is frequently the case during molecular dynamics simulations. Example terms needed for the derivative $\frac{\partial E(\boldsymbol{c}(\boldsymbol{x}))}{\partial \boldsymbol{c}} \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{x}}$ are

$$\frac{\partial E(\boldsymbol{c}(\boldsymbol{x}))}{\partial \boldsymbol{c}} \left(\frac{\partial (\boldsymbol{x}^{\mathrm{T}} \boldsymbol{S} \boldsymbol{x})^1}{\partial \boldsymbol{x}} \right) = \boldsymbol{S}\boldsymbol{x}(\boldsymbol{M}), \tag{52}$$

$$\frac{\partial E(\boldsymbol{c}(\boldsymbol{x}))}{\partial \boldsymbol{c}} \left(\frac{\partial (\boldsymbol{x}^{\mathrm{T}} \boldsymbol{S} \boldsymbol{x})^2}{\partial \boldsymbol{x}} \right) = \boldsymbol{S}\boldsymbol{x}\big(\boldsymbol{O}(\boldsymbol{M}) + (\boldsymbol{M})\boldsymbol{O}\big), \tag{53}$$

$$\frac{\partial E(\boldsymbol{c}(\boldsymbol{x}))}{\partial \boldsymbol{c}} \left(\frac{\partial (\boldsymbol{x}^{\mathrm{T}} \boldsymbol{S} \boldsymbol{x})^3}{\partial \boldsymbol{x}} \right) = \boldsymbol{S}\boldsymbol{x}\big(\boldsymbol{O}(\boldsymbol{O}\boldsymbol{M} + \boldsymbol{M}\boldsymbol{O}) + (\boldsymbol{M}\boldsymbol{O})\boldsymbol{O}\big), \tag{54}$$

where $\boldsymbol{O} = \boldsymbol{x}^{\mathrm{T}} \boldsymbol{S} \boldsymbol{x}$ and $\boldsymbol{M} = (\boldsymbol{x}^{\mathrm{T}} \boldsymbol{H} \boldsymbol{c}) + (\boldsymbol{x}^{\mathrm{T}} \boldsymbol{H} \boldsymbol{c})^{\mathrm{T}}$, and terms are grouped to suggest an efficient way of evaluation.

Table 1 compares directly the computational cost of one step of the OT algorithm with the cost of solving the generalised eigenvalue equation for the N lowest eigenvectors using the ScaLAPACK routines `pdsyevx`, `pdsygst`, and `pdtrsm`. It can be observed that the OT algorithm is far superior.

7. Ab initio molecular dynamics

Classical molecular dynamics simulations are well established as a powerful technique to study dynamic and thermodynamic properties of atomic and molecular systems at a finite temperature [42]. Similar studies can be performed with ab initio molecular dynamics in which explicit electronic structure calculations are employed to compute potential energies and forces [6]. A significant advantage of ab initio molecular dynamics is that no parametrisation of an empirical potential is needed and as such a wide range of systems can be simulated, even if unexpected chemical events take place. The length and time scales of typical ab initio molecular dynamics simulations are currently given by approximately 10 to 1000 atoms and 1 to 100 ps.

Born–Oppenheimer (BO) molecular dynamics is a commonly used form of ab initio molecular dynamics simulations, and is implemented in CP2K. In BO MD the atomic coordinates are treated as classical coordinates, and the ions are propagated in time using Newton's equations of motion with the electronic ground state energy as the potential energy surface. The equations of motion can hence be written as

$$\begin{aligned} M_A \ddot{R}_A(t) &= -\nabla_A E\big(\{R_A\}\big) \\ &= -\nabla_A \min_{\rho(r)} E\big(\{R_A\}, \rho(r)\big), \end{aligned} \tag{55}$$

where M_A and R_A are the atomic mass and atomic coordinates of the atom A, respectively. We note that this simple form of the equations of motion introduces a clear separation between the ionic propagation and the electronic structure part, the latter being a "black-box" returning energies and forces for a given ionic configuration. We exploit this in CP2K by introducing sufficiently abstract interfaces to the electronic structure code (see Section 4) so that other methods commonly used in molecular simulation such as, e.g., path integral MD or MD simulations in the isothermal ensemble (NVT) are readily available.

The time propagation of the atomic coordinates is performed with the velocity Verlet algorithm [56], as it is simple and time reversible. We notice that for BO MD, the time step used in the propagation is only dependent on the frequency spectrum of the *atomic* system. The time reversibility guarantees that no long term drift of the constant of motion, i.e. the sum of the atomic kinetic and potential energy, can be observed, provided that the forces are the exact derivatives of the potential energy (see paragraph 4.3.3 of Ref. [56]). As shown in Section 2, the computed forces are exact derivatives of the energy only if $\frac{\partial E}{\partial P^{\mu\nu}} = 0$. In any practical calculation, the wave function optimisation is such that the above criterion is only approximately true, and hence the resulting drift in the constant of motion can be one criterion to judge the quality of the simulation. This is at variance with Car–Parrinello [72] simulations, where the CP constant of motion (i.e. ionic kinetic energy + potential energy + fictitious ki-

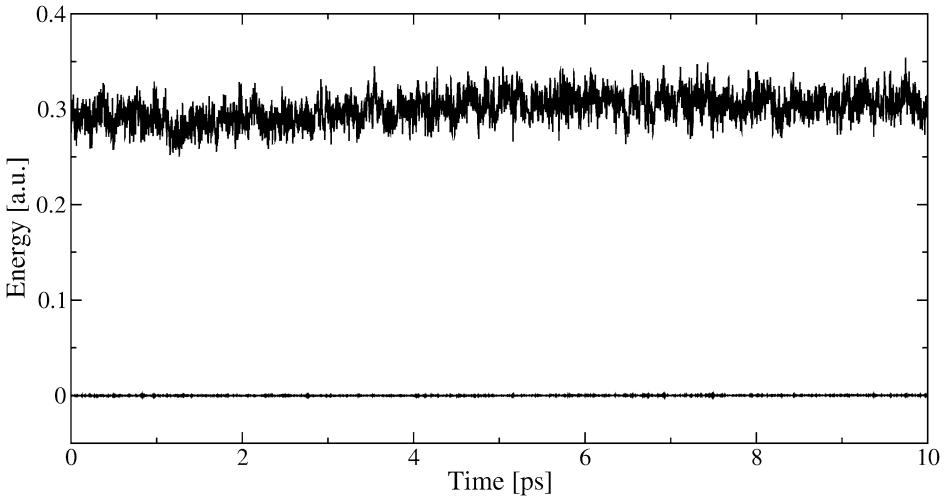

Fig. 5. Shown are the kinetic energy (top), and the constant of motion (bottom) for a liquid of 64 H_2O molecules at approximately 330 K during an NVE simulation. Simulation parameters are: a TZVP//DZVP basis set, a 200 Ry density cutoff, a time step of 0.5 fs, OT DIIS minimiser with $\boldsymbol{PS}$ extrapolation ($K = 3$), and $\epsilon_{\mathrm{SCF}} = 10^{-6}$. The measured drift is as low as $2.0 \cdot 10^{-7}$ a.u. per ps and per atom.

netic energy) is exactly conserved (in the Verlet sense), but where the energy transfer from the ionic system to the fictitious system needs to be controlled.

Shown in Fig. 5 are the results of a simulation on 64 H_2O where the drift in the constant of motion is as low as $2.0 \cdot 10^{-7}$ a.u. per ps and per atom. However, we have found in liquid water simulations with the same convergence criterion, that the drift depends on the basis set and that $1.0 \cdot 10^{-6}$ a.u. per ps and per atom can be expected for $\epsilon_{\mathrm{SCF}} = 10^{-6}$. One advantage of MD is that atomic configurations are generated in a continuous fashion and one can hence predict an initial trial wave function for the SCF calculation by multi-linear extrapolation using the previous wave functions [73]. However, a given eigenfunction ψ_i can change rapidly if the two occupied states ψ_i and ψ_j are energy resonant. We therefore extrapolate the density matrix instead, more precisely its contra-covariant representation $\boldsymbol{PS}$. Since $\boldsymbol{PS}(t_n)$ is approximately a projector onto the occupied subspace it can be multiplied with $\{\psi_i(t_{n-1})\}$ to yield, after orthogonalisation, a initial trial wave function $\{\psi_i(t_n)\}$ for time t_n. Thus, our scheme provides directly an initial wave function, as required for the OT method and, e.g., plane wave schemes, at variance with a similar scheme published recently [74]. A simple formula for $\boldsymbol{C}(t_n)$ that does not require the explicit evaluation of $\boldsymbol{PS}$ is obtained by using Eq. (47):

$$\boldsymbol{C}(t_n) \approx \sum_{m=1}^{K} (-1)^{m+1} \binom{K}{m} \boldsymbol{C}(t_{n-m}) \boldsymbol{C}(t_{n-m})^{\mathrm{T}} \times \boldsymbol{S}(t_{n-m}) \boldsymbol{C}(t_{n-1}). \tag{56}$$

This can be evaluated in $\mathrm{O}(MN^2)$ time. The effect of this extrapolation in terms of the number of iterations needed per MD step and on the error in the energy at the first SCF step is shown in Fig. 6. Our experience has shown that $K = 3$ produces a stable and efficient algorithm. As one application of MD simulation we have computed the hydrogen velocity–velocity auto-correlation function, and the corresponding power spectrum in Fig. 7 for the system described in the caption of Fig. 5. These results can be directly compared to the experimental vibrational spectrum of liquid H_2O. The experimental results for the bending and the two OH stretch modes are 1645, 3280 and 3490 cm^{-1}, and this compares favourably with the well defined peak at 1640 and the broad peak at 3330 cm^{-1} found in our simulations.

8. Accuracy

8.1. Small molecules

As a first accuracy test for QUICKSTEP we have employed the new basis sets described in Section 5

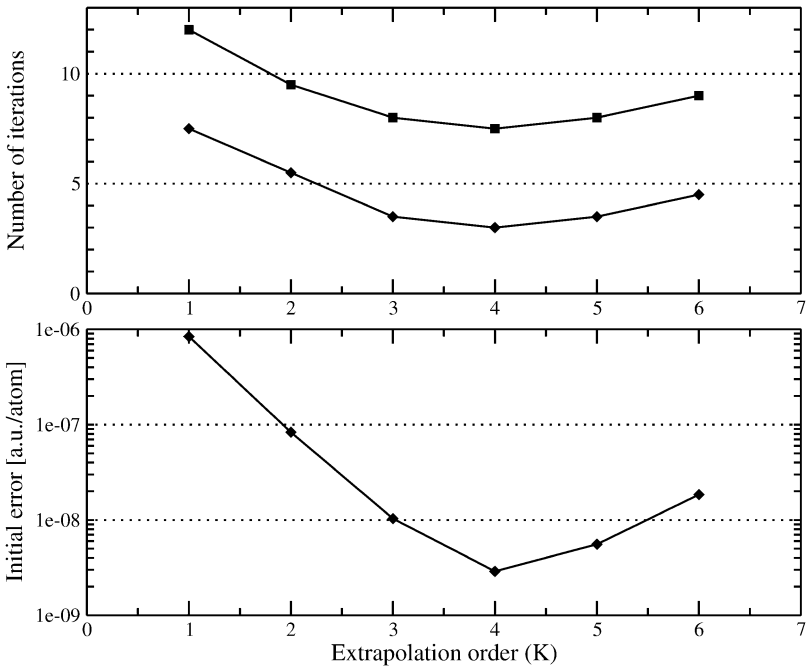

Fig. 6. *Upper panel*: Average number of iterations needed to achieve $\epsilon_{\mathrm{SCF}} = 10^{-6}$ (squares) and convergence ($\epsilon_{\mathrm{SCF}} = 10^{-7}$, diamonds) for an MD simulation of 64 H_2O at 300 K using a TZV2P basis set. *Lower panel*: Error in the total energy per atom of the initial wave function.

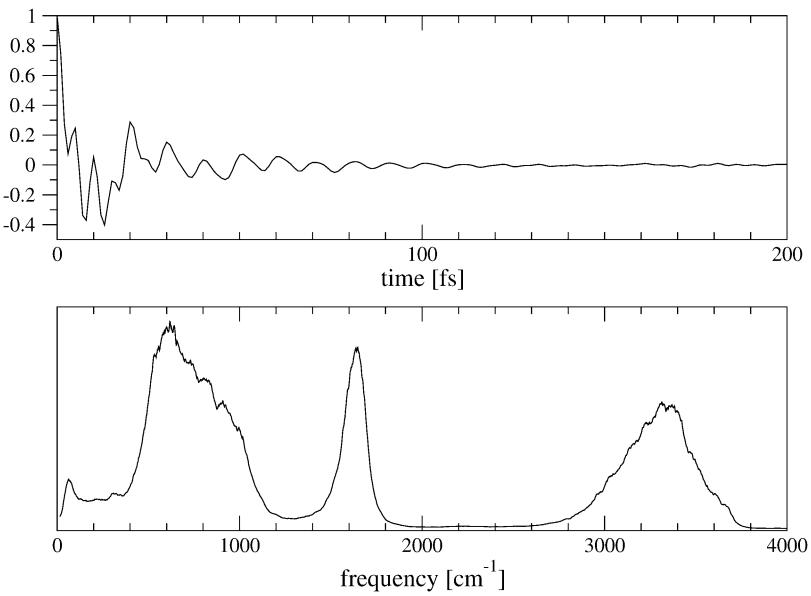

Fig. 7. *Upper panel*: Hydrogen velocity–velocity auto-correlation function. *Lower panel*: Fourier transform of the auto-correlation function. The system is a liquid of 64 H_2O molecules at approximately 330 K during an NVE simulation. Simulation parameters are: a TZVP/DZVP basis set, a 200 Ry density cutoff, a time step of 0.5 fs, OT DIIS minimiser with PS extrapolation ($K = 3$), and $\epsilon_{\mathrm{SCF}} = 10^{-6}$.

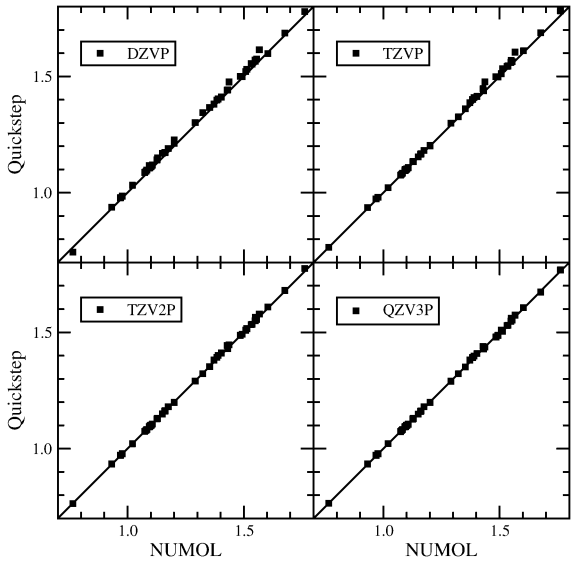

Fig. 8. The optimised bond distances for 39 small molecules calculated with QUICKSTEP using different basis sets are compared to the NUMOL results of Dickson and Becke [75].

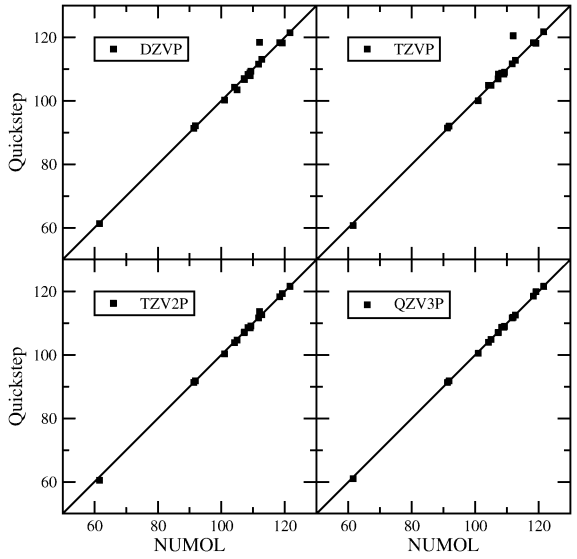

Fig. 9. The optimised bond angles and dihedral angles for 39 small molecules calculated with QUICKSTEP using different basis sets are compared to the NUMOL results of Dickson and Becke [75].

for the geometry optimisation of small molecules using the local density approximation (LDA). The CP2K geometry optimiser works with analytic first derivatives whereas the second derivatives are obtained via an updated Hessian method. In that way, each molecule of the following test set of 39 small molecules:

H_2, Li_2, LiH, BH_3, CH_4, C_2H_2, C_2H_4, C_2H_6, N_2, NH_3, HCN, H_2O, H_2O_2, CO, CO_2, CH_3OH, N_2O, F_2, HF, LiF, CH_3F, OF_2, AlH, SiH_4, SiO, P_2, PH_3, HCP, PN, S_3, H_2S, CS, CS_2, SO_2, COS, SF_6, HCl, CH_3Cl, LiCl

consisting of first- and second-row elements was optimised using Cartesian coordinates. Fig. 8 compares the optimised bond distances obtained with QUICKSTEP using different basis sets with the NUMOL results of Dickson and Becke [75]. NUMOL is a purely numerical DFT code and thus considered to be free of basis set effects. The smallest basis set DZVP gives on average slightly too long bond distances, but already the TZVP basis set works fine for most of the molecules. Finally, the TZV2P, QZV2P, and QZV3P show an satisfactory agreement for all bond distances. Fig. 9 shows the results for the optimised bond and dihedral angles. The agreement for the small DZVP and the TZVP basis set is already excellent. Only one data point is off which corresponds to the dihedral angle of H_2O_2. This angle is known to be very sensitive to the number of employed polarisation functions. However, for the TZV2P basis set the dihedral angle is already very close to the reference value, and for the QZV3P basis a more or less converged result is obtained. A summary of the numerical results of the geometry optimisations is provided in Table 2 which shows the maximum and the root mean square deviation of all bond distances and angle compared to the NUMOL results based on 52 bond distances and 18 angles and dihedral angles. The errors become smaller for growing basis set size as expected. The TZV2P

Table 2
Maximum (Δ_{max}) and root mean square deviation (σ) of bond distances (Å) and bond angles and dihedral angles (°) compared to the NUMOL results for different basis sets

Basis set	Distances [Å]		Angles [°]	
	Δ_{max}	σ	Δ_{max}	σ
DZVP	0.048	0.018	6.4	1.6
TZVP	0.040	0.013	8.5	2.1
TZV2P	0.015	0.006	1.7	0.6
QZV2P	0.012	0.005	2.1	0.6
QZV3P	0.011	0.004	0.7	0.3

basis set gives already an excellent overall agreement and for the QZV3P most distances coincide within the expected errors. Note, that a full agreement with the NUMOL values is not possible, since NUMOL uses a slightly different LDA implementation and it employs a frozen core approximation for the elements beyond Beryllium that differs from the GTH pseudopotentials used by QUICKSTEP. These difference may cause a change of the bond distances of about 0.001 Å. This small error also shows that the effect of the pseudopotential is often negligible compared to typical basis set effects concerning structural properties. Thus a basis set can be chosen according to the accuracy requirements of the actual application. However, the ultimate accuracy of QUICKSTEP is determined by the error of the available exchange and correlation functionals.

8.2. Condensed phase systems

Standard benchmarks to judge the quality of an electronic structure method for the liquid phase do not yet exist. Therefore, in order to assess the accuracy of QUICKSTEP for calculations in the condensed phase, comparisons with the program CPMD [66] have been made. CPMD employs a plane wave basis set to represent the wave functions which is a natural basis set for periodic condensed phase systems. Furthermore, plane waves provide an orthogonal basis that allows to reach conveniently the basis set limit for a given pseudopotential only by increasing a single parameter, i.e. the plane wave cutoff. In principle, identical total energies can be obtained by QUICKSTEP and CPMD, if the same system is described by the same pseudopotentials. Nevertheless, this is a nontrivial and very stringent test for the accuracy of both methods, since the two methodologies and basis sets are very different. We have already mentioned that the GTH pseudopotentials are rather hard (see Section 2.2) and that they require a higher plane wave cutoff than the pseudopotentials of Troullier and Martins [76] that are commonly used in CPMD calculations. Fig. 10 shows a correlation plot between forces

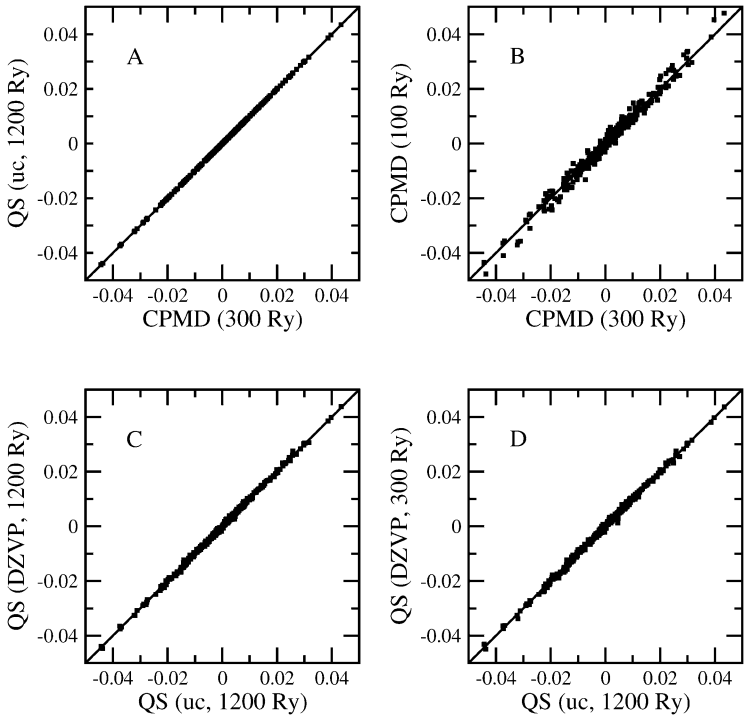

Fig. 10. The four panels show the correlation between all forces F_x, F_y, and F_z calculated for the same configuration of a liquid water sample containing 32 water molecules. Panel A shows that perfect agreement can be obtained between QUICKSTEP (QS) and CPMD using large basis sets (see text) for both methods; panel B compares these converged results to CPMD results using slightly more typical settings; panel C shows the effect of a smaller Gaussian basis set; panel D additionally shows the effect of a smaller density cutoff.

computed with QUICKSTEP and CPMD for the same configuration of a liquid water sample with 32 water molecules in the simulation cell using an LDA density functional [31].

Five different methods are being compared: CPMD with a plane wave cutoff of 300 Ry (implying a density cutoff of 1200 Ry); CPMD with a plane wave cutoff of 100 Ry; QUICKSTEP with a large basis (uc, uncontracted with f and d angular momentum terms, 254 basis functions per water molecule) and a density cutoff of 1200 Ry; QUICKSTEP with a DZVP basis set (23 basis functions per water molecule) and a density cutoff of 1200 Ry; QUICKSTEP with a DZVP basis with a density cutoff of 300 Ry.

Panel A shows that perfect agreement between CPMD and QUICKSTEP can be achieved. The root mean square deviation (RMSD) between the CPMD and the QUICKSTEP forces is below $6 \cdot 10^{-5}$ a.u. The panels B, C, and D, with RMSDs of $2 \cdot 10^{-3}$, $2 \cdot 10^{-4}$, and $7 \cdot 10^{-4}$ a.u., show that the quality of the forces as obtained with QUICKSTEP compares favourably with a typical plane waves calculation. Furthermore, we compared the accuracy of the relative energies obtained from the different methods by computing a potential energy surface of 25 configurations of 64 water molecules sampled during an MD run, using the BLYP density functional [36,38]. Maximum absolute differences (MAD) and root mean square deviations (RMSD) between these potential energy surfaces have been computed as

$$\mathrm{RMSD} = \sqrt{\frac{1}{25}\sum_{i=1}^{25}(E^i - E^i_{\mathrm{ref}})^2}, \tag{57}$$

$$\mathrm{MAD} = \max_i \left|E^i - E^i_{\mathrm{ref}}\right|, \tag{58}$$

where a constant shift is applied to each set of 25 energies so that $\sum_{i=1}^{25} E^i = 0$. The reference potential energy surface is computed using CPMD with a plane wave cutoff of 200 Ry. Results for several basis sets and several density cutoffs are shown in Table 3. The reference potential energy surface itself fluctuates around zero by $9.8 \cdot 10^{-3}$ a.u. RMSD and $18.6 \cdot 10^{-3}$ a.u. MAD. These results show that good agreement can be obtained, but that both a sufficiently good basis set and a sufficiently high density cutoff are needed to reach an accuracy that is, e.g., similar to a 100 Ry CPMD calculation. The best agreement obtained between QUICKSTEP and the reference CPMD calculation was $1.1 \cdot 10^{-3}$ a.u. RMSD and $0.5 \cdot 10^{-3}$ a.u. MAD using a 1200 Ry density cutoff and a 5ZV(3d3f, 3p3d) basis set. We have observed that convergence with respect to the density cutoff is significantly slower for BLYP calculations than for LDA calculations (see also Section 2.4)

Table 3
Root mean square deviation (RMSD) and maximum absolute deviation (MAD) of the total energy in 10^{-3} a.u. (see text for a definition) from a reference BLYP potential energy landscape of 25 configurations of 64 H_2O for a variety of methods. First two rows, using CPMD and different plane wave cutoffs; third and fourth row, using QUICKSTEP with a TZV2P basis and different density cutoffs; fifth and sixth row, using QUICKSTEP and different basis sets and a 340 Ry density cutoff

CPMD	70 Ry	85 Ry	100 Ry	150 Ry	200 Ry
MAD	15.5	10.8	3.6	1.9	0.0
RMSD	7.7	4.2	1.8	0.7	0.0
QUICKSTEP (TZV2P)	200 Ry	240 Ry	280 Ry	340 Ry	380 Ry
MAD	16.7	6.5	5.8	5.3	4.9
RMSD	5.9	2.3	2.2	2.0	1.8
QUICKSTEP (340 Ry)	DZVP	TZVP	TZV2P	QZV2P	QZV3P
MAD	11.1	9.3	5.3	5.2	4.6
RMSD	5.1	4.0	2.0	2.1	1.9

A basis set superposition error (BSSE) is unavoidable if a localised basis is employed. Typically, large basis sets with, e.g., diffuse functions and/or counterpoise corrections [77] are employed to reduce the influence of this error, but these methods are not easy to generalise, e.g., to the case of MD of condensed phase systems. However, the fact that QUICKSTEP reproduces the CPMD forces in the case of liquid water indicates that BSSE is mostly a shift of the potential energy surface. We characterise this shift by looking at the binding energy of a single water molecule in a liquid sample containing 32 molecules.

$$E^{\mathrm{binding}} = E(32\,H_2O) - E(1\,H_2O) - E(31\,H_2O) \tag{59}$$

is sensitive to a proper description of both gas and liquid phase molecules, and might thus be expected to be sensitive to the BSSE error. The BSSE error is defined here approximately for a given basis as the difference in E^{binding} as computed using a testing basis set and a reference counterpoise corrected calculation which is converged with respect to basis set size. We have computed this error for a large number of water molecules

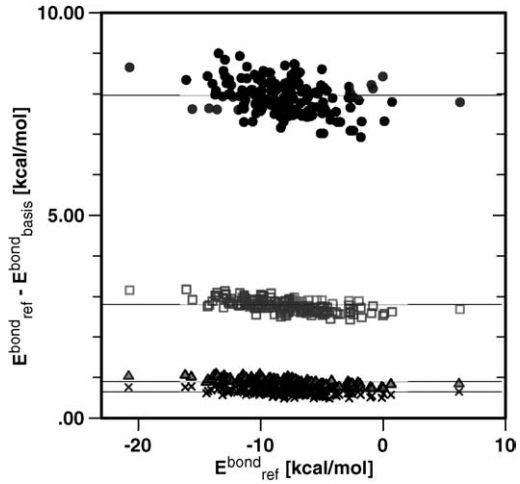

Fig. 11. Computed errors in the binding energy of a single water molecule in the liquid as calculated with DZVP, TZV2P, QZV2P, QZV3P basis sets (circles, squares, triangles and crosses) relative to a reference binding energy that is a counterpoise corrected calculation using a QZV3P basis with added d and f functions (see text for details). The fluctuations around the constant shift have standard deviations of respectively 0.417, 0.157, 0.085, 0.079 kcal/mol.

in liquid samples that were obtained from a MD simulation at around 320 K using the OLYP functional [30] and the results are shown in Fig. 11. As anticipated, a significant shift of this binding energy can be observed, but the error is to a very good extent independent of the configuration. This is consistent with the quality of the potential energy surface observed previously and provides an explanation for the observed rapid convergence with respect to basis set of structural and dynamical properties of liquid water [30].

9. Benchmarks

In this section, we show that the accuracy of QUICKSTEP can be achieved with high computational efficiency. We illustrate both the serial performance and the scalability on parallel computers using a high-end supercomputer as well as a modern cluster based on a PC-like architecture.

9.1. X-ray structure of a dinucleoside monophosphate

In this first section, we illustrate the performance of QUICKSTEP on a desktop computer employing a single Pentium Xeon (3 GHz) CPU for the calculations. The system studied is a dinucleoside monophosphate A2′p5′C that contains a 2′–5′ link, as opposed to the usual 3′–5′. The system has been crystallised and discussed in detail in Ref. [78] and the structure is available in the nucleic acid database [79] under entry URB001. It crystallises with an orthorhombic unit cell (8.631 Å by 18.099 Å by 16.101 Å) with space group $P2_12_12_1$ that contains 280 atoms. The symmetry of the space group has not been exploited in these calculations. Not all hydrogen coordinates are reported in the deposited structure, and as an example application we report here the timings that are relevant in optimising the geometry of the missing hydrogen atoms.

Moderately accurate settings are employed for this task, as our target accuracy for locating the hydrogens is similar to the reported experimental uncertainty of 0.06 Å for hydrogen. We therefore employed a DZVP basis for all atoms (2712 basis functions), a threshold of 10^{-10} for neglecting overlap matrix elements, an auxiliary basis cutoff of 240 Ry ($2.3 \cdot 10^6$ plane waves), and an SCF convergence criterium of $3 \cdot 10^{-6}$, leading to an energy convergence of $\approx 3 \cdot 10^{-8}$ a.u. per atom. Using these settings, combined with the OT method, we obtain single CPU timings of 39 s per SCF iteration, and one BFGS geometry optimisation step takes on average somewhat less than six minutes. The CPU time per SCF iteration increases to only 85 s if the basis is nearly doubled by adopting a TZV2P basis (4652 basis functions). Such a system can thus be studied on a desktop computer.

9.2. Liquid water

Liquid water is a convenient benchmark system since it can easily be scaled by doubling the number of water molecules in the unit cell, followed by classical equilibration to yield a system without additional symmetries. It is a standard benchmark system for the CPMD code [66] to check performance and scalability on various parallel computers. MD runs for pure liquid water at ambient conditions (300 K, 1 bar) have been conducted using input parameters as appropriate for quality production runs, i.e. GTH pseudopotentials, TZV2P basis sets for hydrogen and oxygen, a density cutoff of 280 Ry for the expansion of the electronic density, a threshold of 10^{-12} a.u. for the overlap integral between two primitive Gaussian func-

Table 4
Detailed characteristics of the employed benchmark systems for liquid water at ambient conditions (300 K, 1 bar). The edge length of the cubic simulation cell, the number of atoms, electrons, Gaussian-type orbitals (M), occupied orbitals (N), and plane waves, i.e. grid points, are listed

System	Cell [Å]	Atoms	Electrons	M	N	Grid points ($\times 10^6$)
32 H_2O	9.9	96	256	1280	128	1.3
64 H_2O	12.4	192	512	2560	256	2.0
128 H_2O	15.6	384	1024	5120	512	4.1
256 H_2O	19.7	768	2048	10240	1024	9.3
512 H_2O	24.9	1536	4096	20480	2048	16.0
1024 H_2O	31.3	3072	8192	40960	4096	32.8

Table 5
Occupation of the overlap matrix applying a threshold of 10^{-12} for the overlap contribution of two primitive Gaussian orbital functions

System	Occupation
32 H_2O	100.0%
64 H_2O	99.6%
128 H_2O	85.1%
256 H_2O	51.3%
512 H_2O	25.8%
1024 H_2O	12.9%

tions, and the total energy of the system was converged to 10^{-7} a.u. at every MD time step (0.5 fs).

Table 4 lists the characteristics of the benchmark systems that range in size from 32 to 1024 water molecules, the largest system being several nanometres in all dimensions. The number of contracted Gaussian-type orbital basis functions is growing linearly from 1280 to 40960 functions, and $32.8 \cdot 10^6$ plane waves are required for the auxiliary basis of the largest system. Since matrices like the overlap and Kohn–Sham matrix grow quadratically with system size, it is indispensable to take advantage of the sparsity that is emerging with increasing system size. The fraction of non-zero blocks of the overlap matrix, for a threshold of 10^{-12}, is shown in Table 5. For the smaller systems, all molecules in the unit cell interact with each other, which results in a dense overlap matrix, whereas starting from roughly 200 water molecules, the interaction sphere is completely confined in the unit cell. Consequently, the number of non-zero elements grows only linearly with system size for systems larger than roughly 256 H_2O. As described previously, this matrix sparsity is exploited in QUICKSTEP. A further observation that can be made is that for the employed TZV2P basis the number of occupied orbitals N is only 10% of total number of orbitals M (see Table 4). Operations that deal only with the occupied orbitals (MN) require significantly less memory and time than similar operations on full matrices (M^2). It is therefore possible to test larger systems with the orbital transformation method (OT) than with the traditional diagonalisation (TD) method.

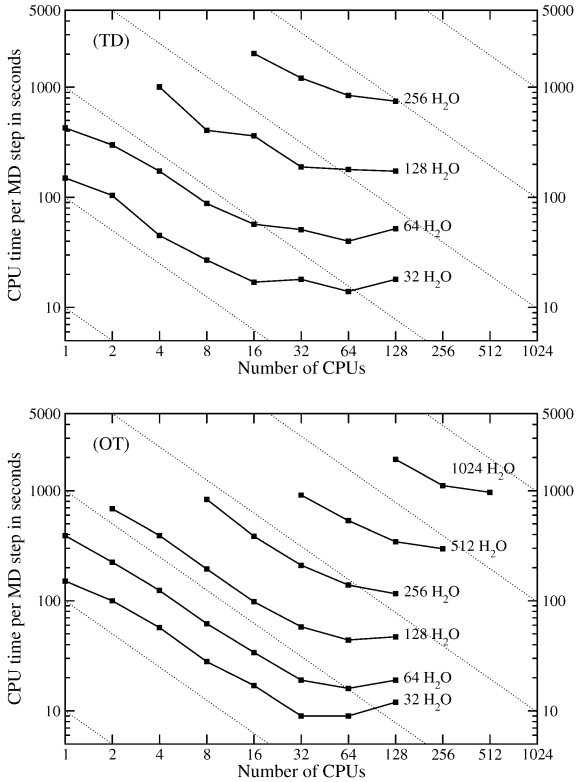

Fig. 12. Scaling of the CPU time per MD step using the traditional diagonalisation (TD) scheme and the orbital transformation (OT) method for the benchmarks systems of Table 4. The calculations were performed on an IBM Regatta p690+ system with 32 Power4+ (1.7 GHz) per node interconnected by an IBM High Performance Switch (HPS). The dotted lines represent ideal scaling.

As a measure of the performance we have reported the average time needed per MD step for the benchmark systems of Table 4 using both traditional diagonalisation and the orbital transformation method. This time covers all aspects of the calculation, including the geometry dependent initialisations, SCF iterations and force calculation. Results are reported in Fig. 12 using an IBM Regatta p690+ system with 32

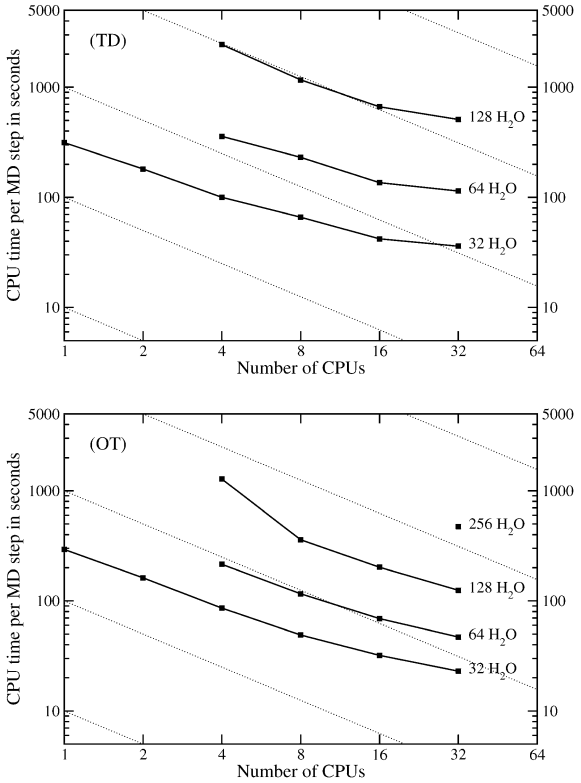

Fig. 13. Scaling of the CPU time per MD step using the traditional diagonalisation (TD) scheme and the orbital transformation (OT) method for the benchmarks systems of Table 4. The calculations were performed on a PC-Cluster with dual AMD Athlon MP2200+ using a SCI interconnect. The dotted lines represent ideal scaling.

Power4+ (1.7 GHz) CPUs per node, 3.6 GB RAM per processor, and interconnected by an IBM High Performance Switch (HPS), and in Fig. 13 for a PC cluster with dual AMD Athlon MP2200+ CPUs and a fast interconnect (SCI). A double logarithmic scale is employed to show the parallel scaling of the methods. From these results, we can conclude that the TD scheme allows for efficient simulation of small systems (32–64 H_2O) on a relatively small number of CPUs, and that medium sized systems (128–256 H_2O) run with fair efficiency on 32–64 CPUs, whereas large systems cannot run with the given memory constraints. The OT method provides significantly improved scaling for all systems, is up to 6 times faster, and the reduced memory requirements allow simulation of up to 1024 water molecules. With the current implementation, at least one or two H_2O molecules (40–80 basis functions) per CPU are needed to maintain efficiency. Whereas short relaxations are the limit for the larger systems, 10 ps of dynamics of a system containing 64 water molecules can be generated in roughly four days using one Regatta node.

10. Summary and outlook

In this paper, we have described the current implementation of the GPW method in the QUICKSTEP program. The dual representation of the charge density allows for an efficient treatment of the Hartree terms, while maintaining the compact representation of the wave functions in the atomic orbital basis. Furthermore, since the description of the wave function is completely in a localised basis, linear scaling Kohn–Sham matrix construction is obtained using screening techniques that lead to sparse matrix structures. Combining pseudopotentials of established quality with a sequence of systematically improving basis sets, we have illustrated that good accuracy can be obtained, both in gas phase and in solution. A number of novel computational techniques make the computational cost of increasing the basis set size rather modest, effectively growing linearly with basis set size. The stability and efficiency of the orbital transformation method, combined with the ***PS*** extrapolation scheme make it possible to perform extended, energy conserving Born–Oppenheimer molecular dynamics simulations. The code shows excellent efficiency on commodity PCs and fair scaling on parallel machines, so that a wide range of physical systems becomes accessible. Nevertheless, a number of challenges remains unsolved. These include, e.g., a more accurate treatment of the exchange–correlation functional, an efficient extension to more general forms such as hybrid functionals and other orbital dependent forms, and a parallelisation strategy that allows for efficient simulations on the coming generation of computers that have thousands of CPUs. Several extensions, e.g., the Gaussian and augmented plane wave (GAPW) method, time dependent density functional theory (TDDFT), and hybrid quantum—classical simulations (QM/MM) are currently being implemented.

Acknowledgements

This work was partially supported by the Bundesministerium für Bildung und Forschung (BMBF) in the framework of the project High-Performance Computing in Chemistry (HPC-Chem) [80]. One of the authors (J.V.) obtained financial support from a Marie Curie Fellowship, and would like to thank M. Sprik for supporting development of the code. T. Chassaing and J. Hutter acknowledge support from the Swiss National Science Foundation (Project 200020-100417). Computer resources were in part provided by the Swiss National Supercomputing Centre (CSCS) in Manno, the Forschungszentrum Jülich, and as part of a EPSRC grant to the UKCP consortium by the HPCx facilities at Daresbury Laboratory.

References

[1] P. Hohenberg, W. Kohn, Phys. Rev. B 136 (1964) B864.
[2] W. Kohn, L.J. Sham, Phys. Rev. 140 (1965) A1133.
[3] S. Goedecker, Rev. Mod. Phys. 71 (1999) 1085.
[4] G. Lippert, J. Hutter, M. Parrinello, Mol. Phys. 92 (1997) 477.
[5] The CP2K developers group, http://cp2k.berlios.de/, 2004.
[6] D. Marx, J. Hutter, in: J. Grotendorst (Ed.), NIC Series, vol. 1, Modern Methods and Algorithms of Quantum Chemistry, FZ Jülich, Germany, 2000, pp. 329–477; see also http://www.fz-juelich.de/nic-series/Volume1/.
[7] L. Füsti-Molnár, P. Pulay, J. Chem. Phys. 116 (2002) 7795.
[8] L. Füsti-Molnár, P. Pulay, J. Chem. Phys. 117 (2002) 7827.
[9] L. Füsti-Molnár, P. Pulay, J. Mol. Struct. (Theochem) 666 (2003) 25.
[10] L. Füsti-Molnár, J. Chem. Phys. 119 (2003) 11080.
[11] J. Baker, L. Füsti-Molnár, P. Pulay, J. Phys. Chem. A 108 (2004) 3040.
[12] J.L. Whitten, J. Chem. Phys. 58 (1973) 4496.
[13] B.I. Dunlap, J.W.D. Connolly, J.R. Sabin, J. Chem. Phys. 71 (1979) 4993.
[14] O. Vahtras, J. Almlöf, M.W. Feyereisen, Chem. Phys. Lett. 213 (1993) 514.
[15] K. Eichorn, O. Treutler, H. Öhm, M. Häser, R. Ahlrichs, Chem. Phys. Lett. 240 (1995) 283.
[16] K. Eichorn, F. Weigend, O. Treutler, R. Ahlrichs, Theor. Chem. Acc. 97 (1997) 119.
[17] B. Delley, J. Chem. Phys. 92 (1989) 508.
[18] V. Termath, N.C. Handy, Chem. Phys. Lett. 230 (1994) 17.
[19] T.L. Beck, Rev. Mod. Phys. 72 (2000) 1041.
[20] X. Chen, J.-M. Langlois, W.A. Goddard III, Phys. Rev. B 52 (1995) 2348.
[21] P. Ordejón, E. Artacho, J.M. Soler, Phys. Rev. B 53 (1996) R10441.
[22] C.M. Goringe, E. Hernández, M.J. Gillan, I.J. Bush, Comput. Phys. Comm. 102 (1997) 1.
[23] S.D. Kenny, A.P. Horsfield, H. Fujitani, Phys. Rev. B 62 (2000) 4899.
[24] A.A. Mostofi, C.-K. Skylaris, P.D. Haynes, M.C. Payne, Comput. Phys. Comm. 147 (2002) 788.
[25] Y. Liu, D.A. Yarne, M.E. Tuckerman, Phys. Rev. B 68 (2003) 125110.
[26] M. Krack, M. Parrinello, Phys. Chem. Chem. Phys. 2 (2000) 2105.
[27] G. Hura, D. Russo, R.M. Glaeser, T. Head-Gordon, M. Krack, M. Parrinello, Phys. Chem. Chem. Phys. 5 (2003) 1981.
[28] I.-F.W. Kuo, C.J. Mundy, Science 303 (2004) 658.
[29] I.-F.W. Kuo, C.J. Mundy, M.J. McGrath, J.I. Siepmann, J. VandeVondele, M. Sprik, J. Hutter, B. Chen, M.L. Klein, F. Mohamed, et al., J. Phys. Chem. B 108 (2004) 12990.
[30] J. VandeVondele, F. Mohamed, M. Krack, J. Hutter, M. Sprik, M. Parrinello, J. Chem. Phys. (2004) 014515.
[31] S. Goedecker, M. Teter, J. Hutter, Phys. Rev. B 54 (1996) 1703.
[32] C. Hartwigsen, S. Goedecker, J. Hutter, Phys. Rev. B 58 (1998) 3641.
[33] G. Lippert, J. Hutter, M. Parrinello, Theor. Chem. Acc. 103 (1999) 124.
[34] T.H. Dunning, J. Chem. Phys. 90 (1989) 1007.
[35] J. VandeVondele, J. Hutter, J. Chem. Phys. 118 (2003) 4365.
[36] A.D. Becke, Phys. Rev. A 38 (1988) 3098.
[37] C.T. Lee, W.T. Yang, R.G. Parr, Phys. Rev. B 37 (1988) 785.
[38] B. Miehlich, A. Savin, H. Stoll, H. Preuss, Chem. Phys. Lett. 157 (1989) 200.
[39] J.P. Perdew, Phys. Rev. B 33 (1986) 8822.
[40] F.A. Hamprecht, A.J. Cohen, D.J. Tozer, N.C. Handy, J. Chem. Phys. 109 (1998) 6264.
[41] J.P. Perdew, K. Burke, M. Ernzerhof, Phys. Rev. Lett. 77 (1996) 3865.
[42] M.P. Allen, D.J. Tildesley, Computer Simulations of Liquids, Claredon Press, Oxford, 1987.
[43] A.D. Boese, N.L. Doltsinis, N.C. Handy, M. Sprik, J. Chem. Phys. 112 (2000) 1670.
[44] A.D. Boese, N.C. Handy, J. Chem. Phys. 114 (2001) 5497.
[45] N.C. Handy, A.J. Cohen, J. Chem. Phys. 116 (2002) 5411.
[46] J.M. Tao, J.P. Perdew, V.N. Staroverov, G.E. Scuseria, Phys. Rev. Lett. 91 (2003) 146401.
[47] A.D. Becke, J. Chem. Phys. 98 (1993) 1372.
[48] A.D. Becke, J. Chem. Phys. 98 (1993) 5648.
[49] P.J. Stephens, F.J. Devlin, C.F. Chabalowski, M.J. Frisch, J. Phys. Chem. 98 (1994) 11623.
[50] J.A. White, D.M. Bird, Phys. Rev. B 50 (1994) 4954.
[51] L.C. Balbás, J.L. Martins, J.M. Soler, Phys. Rev. B 64 (2001) 165110.
[52] A.D. Becke, J. Chem. Phys. 88 (1988) 2543.
[53] C.W. Murray, N.C. Handy, G.J. Laming, Mol. Phys. 78 (1993) 997.
[54] O.T. Treutler, R. Ahlrichs, J. Chem. Phys. 102 (1995) 346.
[55] S.G. Louie, S. Froyen, M.L. Cohen, Phys. Rev. B 26 (1982) 1738.
[56] D. Frenkel, B. Smit, Understanding Molecular Simulation, Computational Science Series, vol. 1, second ed., Academic Press, 2002.
[57] H. Bekker, E.J. Dijkstra, M.K.R. Renardus, H.J.C. Berendsen, Mol. Simul. 14 (1995) 137.

[58] P. Pulay, Mol. Phys. 17 (1969) 197.
[59] BLAS, Basic Linear Algebra Subprograms, http://www.netlib.org/blas/.
[60] LAPACK, Linear Algebra PACKage, http://www.netlib.org/lapack/.
[61] FFTW, Fastest Fourier Transform in the West, http://www.fftw.org/.
[62] MPI, Message Passing Interface, http://www.mpi-forum.org/.
[63] ScaLAPACK, Scalable LAPACK, http://www.netlib.org/scalapack/.
[64] OpenMP, Multi-platform shared-memory parallel programming, http://www.openmp.org/.
[65] L.S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, et al., ScaLAPACK Users' Guide, SIAM, 1997.
[66] CPMD, Version 3.7, copyright IBM Corp. 1990–2003, copyright MPI für Festkörperforschung Stuttgart 1997–2001; http://www.cpmd.org/.
[67] T. Helgaker, P.R. Taylor, Modern Electronic Structure Theory, Part II, World Scientific, Singapore, 1995.
[68] EMSL Gaussian Basis Set Order Form, http://www.emsl.pnl.gov/forms/basisform.html.
[69] D.E. Woon, T.H. Dunning, J. Chem. Phys. 98 (1993) 1358.
[70] P. Pulay, J. Comput. Chem. 3 (1982) 556.
[71] P. Pulay, Chem. Phys. Lett. 73 (1980) 393.
[72] R. Car, M. Parrinello, Phys. Rev. Lett. 55 (1985) 2471.
[73] T.A. Arias, M.C. Payne, J.D. Joannopoulos, Phys. Rev. Lett. 69 (1992) 1077.
[74] P. Pulay, G. Fogarasi, Chem. Phys. Lett. 386 (2004) 272.
[75] R.M. Dickson, A.D. Becke, J. Chem. Phys. 99 (1993) 3898.
[76] N. Troullier, J.L. Martins, Phys. Rev. B 43 (1991) 1993.
[77] S.F. Boys, F. Bernardi, Mol. Phys. 19 (4) (1970) 553; reprinted Mol. Phys. 100 (1) (2002) 65–73.
[78] R. Parthasarathy, M. Malik, S.M. Fridey, Proc. Natl Acad. Sci. USA 79 (1982) 7292.
[79] H.M. Berman, W.K. Olson, D.L. Beveridge, J. Westbrook, A. Gelbin, T. Demeny, S.-H. Hsieh, A.R. Srinivasan, B. Schneider, Biophys. J. 63 (1992) 751.
[80] M. Krack, M. Parrinello, in: J. Grotendorst (Ed.), NIC Series, vol. 25, High Performance Computing in Chemistry, Report of the BMBF Project, Grand Number 01IRA17 A-C, FZ Jülich, Germany, 2004.