JCTC Journal of Chemical Theory and Computation

Article

pubs.acs.org/JCTC

Linear Scaling Self-Consistent Field Calculations with Millions of Atoms in the Condensed Phase

Joost VandeVondele,*[,†] Urban Borštnik,[‡,§] and Jürg Hutter[‡]

[†]Department of Materials, ETH Zurich, Wolfgang-Pauli-Strasse 27, 8093 Zurich, Switzerland

[‡]Physical Chemistry Institute, University of Zurich, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland

ABSTRACT: In this work, the applicability and performance of a linear scaling algorithm is investigated for three-dimensional condensed phase systems. A simple but robust approach based on the matrix sign function is employed together with a thresholding matrix multiplication that does not require a prescribed sparsity pattern. Semiempirical methods and density functional theory have been tested. We demonstrate that self-consistent calculations with 1 million atoms are feasible for simple systems. With this approach, the computational cost of the calculation depends strongly on basis set quality. In the current implementation, high quality calculations for dense systems are limited to a few hundred thousand atoms. We report on the sparsities of the involved matrices as obtained at convergence and for intermediate iterations. We investigate how determining the chemical potential impacts the computational cost for very large systems.

1. INTRODUCTION

The single particle wave functions that are conventionally used in Hartree−Fock or density functional theory (DFT) are orthonormal. If we assume that these orbitals can span all of space, even just verifying the orthonormality of all pairs of orbitals is an operation that has a cost that grows with the third power of system size. This matches the cost of the most commonly used approach to find these orbitals, namely diagonalization of the Fock or Kohn−Sham matrix. However, intuition tells us that for many materials electrons do not interact over long distances. This suggests that methods exploiting this locality could have a cost that grow linearly with system size. Indeed, the development of such linear scaling methods has been an active field of research for several years. The importance of these methods grows as more powerful computational resources become available, resulting in significant activity in recent years. Most of the methods rely on the fact that the orbitals can be effectively localized or that the density matrix has a sparse representation.[1−4]

A large class of methods to obtain a linear scaling density matrix construction is related to the purification of the density matrix as proposed by McWeeny.[5] Starting from a properly scaled Hamiltonian, the density matrix is constructed by a simple iterative procedure.[6−12] Effectively, high order polynomials (or rational functions) that approximate a step function are constructed and evaluated for the given Hamiltonian. Beylkin et al. (ref 7) have expressed the density matrix directly in terms of the matrix sign function and, as such, provided the terminology to easily connect to other fields. Indeed, the matrix sign function has been studied in detail in the mathematical literature (see, e.g., ref 13 for a review), and various iterative schemes with different convergence rates and domains are known. The simplest of these iterative schemes, also known as the Newton−Schulz iteration and equivalent to the McWeeny procedure, has been employed in this and earlier work.[7,8] In the context of electronic structure calculations, higher order purification schemes have been proposed by Holas,[9] and an efficient trace correcting purification method was proposed by Niklasson.[10,11] A detailed analysis of numerical error has been presented in ref 12. Purification can be seen as a limit of the Fermi operator expansion procedures.[14−21] Compared to the purification methods, relatively low order (50) polynomials were used in the early work,[14] but a careful analysis of the Chebyshev representation[15,16,18] shows that good accuracy can be obtained. Novel Fermi operator expansion methods go beyond the Chebyshev expansion.[19−21] An alternative to the expansion methods is a direct minimization of the total energy, using a suitably modified functional that naturally enforces the orthonormality of the orbitals, or the idempotency of the density matrix. Among the first approaches employing this strategy was the method by Li et al.[22] A wide range of variants has been proposed that modify either the functional, the variables employed, or the minimization algorithm.[23−32] For example, a conjugate gradients based density matrix search is employed in refs 25−27 and a direct inversion in the iterative subspace approach in ref 30. Instead of using the density matrix directly, localized orbitals are used in refs 23, 24, and 31 and an exponential parametrization in refs 29, 32, and 33. Also possible are divide-and-conquer strategies, which tackle the problem by splitting the large system into smaller subsystems. More conventional techniques are employed for the smaller subsystems, and the results are patched together to obtain a solution for the full system.[34−41] Methods using Krylov-subspace approaches and the Lanczos algorithm have also been proposed.[42−44] Detailed discussions of implementations of linear scaling codes have been presented by various groups, even though the emphasis is sometimes on the linear scaling construction of the Hamiltonian matrix, rather than on the self-consistent solution of the equations.[45−60]

Special Issue: Wilfred F. van Gunsteren Festschrift

Received: December 15, 2011
Published: February 29, 2012

In this manuscript, we employ a simple iterative scheme based on sign matrix iterations as proposed in refs 7 and 8. The main purpose of our work is to test the linear scaling potential of our implementation, up to millions of atoms, in a massively parallel framework. In order for these calculations to be feasible, both the computational cost and the memory usage must scale linearly with system size. Only recently, DFT calculations on 1 million atoms has been demonstrated, albeit either nonself-consistently[58] or in an orbital free framework.[61] Million atom calculations are an important benchmark for linear scaling implementations, since smaller test systems can easily hide quadratic behavior with a small prefactor. As such, these calculations provide a stringent test for our code, in particular the Gaussian and Plane Waves implementation[62] and the sparse matrix library.[63] With our testing, we focus on systems that are three-dimensional in nature, as these are significantly more challenging but also more realistic than the usual quasi-one-dimensional benchmark systems. These data will provide input for further developments of more advanced methods, including more recent iterative schemes as well as minimization based methods.

The structure of this manuscript is as follows: In section 2, the algorithm used to perform the self-consistent calculations of the electronic structure is described. In section 3, the computational setup is briefly summarized. In section 4, we report on the timings obtained for three-dimensional systems described with periodic boundary conditions. We investigate the performance in the limit both of a gas phase system, which stresses the part of the code which relies on real space grids and fast Fourier transforms, and of a dense condensed phase system, which is heavily dominated by the sparse linear algebra. Both DFT and semiempirical methods are compared, investigating for the DFT calculations the effect of using minimal and more extended basis sets. In section 5, the cost of individual matrix multiplications and the impact of the precise location of the chemical potential is investigated. In section 6, the observed linear scaling complexity is critically investigated, looking at parallel performance and the required screening threshold.

2. ALGORITHMIC DESCRIPTION

The matrix sign function can be defined as

$$\mathrm{sign}(A) = A(A^2)^{-1/2} \tag{1}$$

For diagonalizable A, eigenvectors of A are eigenvectors of $\mathrm{sign}(A)$, with the eigenvalues of $\mathrm{sign}(A)$ being -1 or $+1$ for negative or positive eigenvalues of A, respectively. Fortunately, various simple iterative algorithms are available to compute the matrix sign function, and we refer the reader to ref 13 for a detailed and valuable overview. These algorithms converge superlinearly and thanks to the properties of the sign function are numerically stable. The simplest form, which only requires two matrix multiplications per iteration, is employed here:

$$X_{n+1} = \frac{1}{2}X_n(3I - X_n^2) \tag{2}$$

In the above equation, I is the identity matrix. For $X_0 = cA$, this iteration converges quadratically to $X_\infty = \mathrm{sign}(A)$ for $c < \|A\|^{-1}$. In our implementation, c is obtained from the minimum of the Gershgorin and Frobenius norms of A. The convergence criterion employed terminates the iteration at X_{n+1} if $\|I - X_n^2\|_F < (\varepsilon_{\mathrm{filter}})^{1/2}\,\|X_n^2\|_F$ where $\|.\|_F$ is the Frobenius norm. Since the algorithm is quadratically convergent, near convergence, each iteration will approximately double the number of correct digits in the solution. Furthermore, it requires only matrix multiplications and can thus be implemented in a relatively straightforward way on parallel computers.

A linear scaling cost results from the fact that all matrix operations are performed on sparse matrices, which have a number of nonzero entries that scale linearly with system size. In order to retain sparsity during the iterations of the above algorithm, we employ a threshold ($\varepsilon_{\mathrm{filter}}$) to flush small entries to zero after multiplication, thereby reducing the number of nonzero blocks and speeding up the following multiplications. This filtering is such that the Frobenius norm of the atomic submatrices (atomic blocks) of the result is within the threshold of the exact multiplication. Additionally, during multiplication, computing the product of two atomic blocks (a_{ik} and b_{kj}) is avoided if their norms satisfy $\|a_{ik}\|_F \cdot \|b_{kj}\|_F < \varepsilon_{\mathrm{filter}} N_{ij}$, where N_{ij} is the minimum of the number of blocks in row i of a and column j in b. This scheme greatly reduces the number of blocks newly created and afterward filtered out for sparse matrices and reduces the number of flops for matrices which have matrix blocks of small magnitude, even if they are fully occupied. This filtering during multiplication has previously been used in ref 27, where a 10% speedup has been reported. In this work, speedups on the order of 300% are observed. In ref 27, the threshold during multiplication is 100× tighter than after multiplication, a possibility we have not explored.

On the basis of the sign matrix function, the density matrix P corresponding to a given Hamiltonian (Kohn−Sham or semiempirical) matrix H, overlap matrix S, and chemical potential μ can be computed as

$$P = \frac{1}{2}(I - \mathrm{sign}(S^{-1}H - \mu I))S^{-1} \tag{3}$$

The idempotency ($PSPS = PS$) and commutativity ($SPH - HPS = 0$) conditions are always satisfied. The number of electrons N_{el} is determined by the chemical potential μ and can be obtained from $N_{\mathrm{el}} = \mathrm{trace}(PS)$. Note that equivalent expressions can be obtained using the general identity $Xf(A)X^{-1} = f(XAX^{-1})$ with, for example, $X = S^{1/2}$. S^{-1} is computed conveniently using $S^{-1} = S^{-1/2}S^{-1/2}$ where the square root and inverse square root can be obtained (see, e.g., ref 13) from

$$\mathrm{sign}\left(\begin{bmatrix} 0 & A \\ I & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & A^{1/2} \\ A^{-1/2} & 0 \end{bmatrix} \tag{4}$$

The corresponding iterative scheme has been studied in detail in refs 64 and 65.

A stationary solution of the self-consistent equations is obtained by a simple static mixing[66] approach:

$$P_{n+1} = \frac{1}{2}(I - \mathrm{sign}(S^{-1}\hat{H}_n - \mu_n I))S^{-1}$$

$$\hat{H}_{n+1} = (1 - \alpha)\hat{H}_n + \alpha H_{n+1}$$

α is a mixing parameter between 0 and 1, and $\hat{H}_n$ an auxiliary matrix. The fixed point implies that $\hat{H}_n = H_n$, and thus $SP_nH_n - H_nP_nS = 0$. For each iteration, the total electronic energy (E_n) and Hamiltonian matrix (H_n) are computed from the density matrix P_n. The value of the chemical potential μ_n is determined by bisecting a suitable interval until $|\mathrm{trace}(P_{n+1}\,S) - N_{\mathrm{el}}| < 1/2$, for a given N_{el}. Note that the $\mathrm{trace}(P_{n+1}S)$ is integer-valued unless finite accuracy is employed in the calculation of the sign

dx.doi.org/10.1021/ct200897x | J. Chem. Theory Comput. 2012, 8, 3565−3573

function. The initial $\hat{H}_0$ is obtained from a block diagonal density matrix, with diagonal blocks corresponding to the densities of neutral, spherical atoms. The convergence criterion employed is $E_n - E_{n-1} < \varepsilon_{SCF} N_{el}$. The mixing scheme and convergence criterion are suitable for the homogeneous, simple systems studied here, but they will need refinement for electronically more demanding systems.

Parallel sparse matrix−matrix multiplication is a key operation for the linear scaling algorithms presented here and has triggered significant effort in the field.[67−71] A sparse matrix multiplication library suitable for atomistic systems and aimed at massively parallel architectures has recently been implemented in CP2K.[72] Here, we provide a short high-level description of the implementation, which is freely available from the CP2K Web site.[72] A detailed description of the implementation and an analysis of the various computational and hardware aspects that influence performance is a study in its own right and will be presented elsewhere.[63] The matrix data are stored in blocked form, where a block typically refers to an atom, or a small local cluster of atoms such as a molecule. The parallelization is based on Cannon's algorithm,[73] which assumes a 2D layout (rows/columns) of the processors and a 2D distribution of matrix data. Load balancing is achieved by permuting rows and columns of the matrix to obtain a more even distribution of the data, so that every processor has approximately the same amount of data. This procedure reduces the spatial locality of data but yields similar performance for homogeneous and inhomogeneous systems. The key advantages of the Cannon algorithm are that a given processor communicates only with nearest neighbors on the 2D grid, the total amount of data transferred per processor decreases as the number of processes increases, and only a small number of messages is needed. These factors favor strong scaling and match what is common in dense matrix multiplication libraries. For each communication step in Cannon's algorithm, there is a corresponding local matrix−matrix multiplication. Performance for the local matrix multiplication is improved by using a cache-oblivious recursive multiplication procedure in a first stage, followed by a conversion to compressed sparse row format and block-wise multiplication in a second stage. An autotuned library of small matrix multiplication kernels is used to multiply these small blocks with good performance. The overall performance of the library depends strongly on block size, the fraction of present blocks, and the number of employed processors. On the XT5 system, it ranges between 50% and 1% of processor peak performance.

3. COMPUTATIONAL SETUP

All calculations have been performed with the CP2K/Quickstep program,[62,72] which employs the Gaussian and (augmented) Plane Waves scheme (GPW[74]/GAPW[75]) to compute in linear scaling time the Kohn−Sham matrix for pseudopotential (all-electron) systems. The essence of the GPW method is to combine contracted Gaussians functions as a primary basis, with plane waves (equivalent to regular grids) as an auxiliary basis for the representation of the electronic density. The localized Gaussian basis leads to sparse representations of the Kohn−Sham matrix, and linear scaling analytic calculation of terms such as the kinetic energy. The plane wave representation of the electronic density allows for a linear scaling calculation of the Coulomb energy using Fast Fourier Transforms. The transformation from the atomic orbital representation of the density matrix to a density on a real-space grid can be performed efficiently and in linear scaling time.[62] The calculations reported here are based on pseudopotentials[76] and use the local density approximation (LDA) as parametrized in ref 76. Generalized Gradient Approximations (GGA) can be used without problems and with very similar cost. Our current implementation of Hartree−Fock exchange,[77−79] as needed for hybrid density functionals, is not capable of dealing with systems containing more than 10 000 atoms and is thus not considered in this work. In addition to DFT calculations, linear scaling semiempirical NDDO[80,81] calculations and density functional tight binding (DFTB)[82] calculations have been performed as well. Whereas the basis set is part of the method in the latter two approaches, it is an important parameter in DFT calculations. The basis sets employed for the DFT calculations are highly contracted and molecularly optimized basis sets.[83] These are particularly suitable for linear scaling condensed phase calculations, as they combine accuracy with a favorable condition number of the overlap matrix. For oxygen and hydrogen, the minimal basis set (SZV-GTH-MOLOPT) has four and one function, whereas the better quality basis (DZVP-GTH-MOLOPT) has 13 and 5 functions. The screening threshold in the calculation of the Kohn−Sham matrix is 10^{-10}; the cutoff for the density plane waves grid is 300 Ry. The thresholds ε_{SCF} and ε_{filter} are 10^{-7} unless mentioned explicitly. All energies and energy thresholds are in atomic units. The matrix multiplications are based on atomic blocks, even though molecular clustering is possible, and usually faster. All calculations have been run in double precision; the single precision multiplication is not employed. Calculations have been run on Cray XT5 hardware, except where mentioned explicitly.

4. BENCHMARKS

In this section, we benchmark the current implementation for a number of three-dimensional condensed phase systems. The three-dimensional nature of these systems is an important characteristic, as it directly impacts the sparsity of the matrices involved. While a one-dimensional system (such as an alkane chain) will typically only have tens of atoms interacting with a given atom, in the condensed phase any given atom interacts with hundreds of atoms. The computational cost can be expected to grow quadratically with the number of interacting atoms. Reported timings are the full run time of the program, including initialization and full wave function optimization, to ensure that all parts are linear scaling and, thus, that we have a practical tool for actual large scale calculations. Finally, tested system sizes exceed 1 million atoms in a number of cases; in our experience, tests with systems of this size are essential to identifying and eliminating quadratically scaling loops, even those with very small prefactors.

In order to demonstrate the linear scaling implementation for DFT calculations, a simple system is considered first. A three-dimensional crystal of H_2 molecules, described using a minimal basis set (SZV-GTH-MOLOPT), is employed at a density of 1 molecule per $4 \times 3 \times 3$ Å^3 (approximately corresponding to the density of a gas at 1 kbar). All calculations use an $\varepsilon_{filter} = 10^{-7}$ and are run on 46 656 cores of a Cray XT5 (16 GB/12 cores) for memory reasons. The results are shown in Figure 1, and total execution time is compared with the time spent for matrix multiplication. Linear scaling performance is obtained for systems of up to 1 million molecules. Even for the largest system, the total energy can be computed in just 8 min of wall

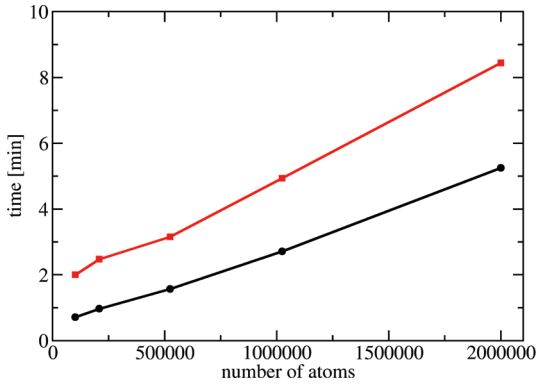

Figure 1. Total walltime (red, squares) and time spent in matrix multiplication (black, circles) for the Kohn–Sham energy calculation of an artificial H_2 crystal on 46 656 cores using a SZV-MOLOPT-GTH basis set. The largest system contains 1 million molecules and has a unit cell of 400 × 300 × 300 Å^3 (41.7 billion grid points at the 300 Ry density cutoff).

time, making such a system practical for geometry optimization or even short MD simulations. A second observation is that the ratio between the time spent in matrix multiplications accounts for somewhat more than half of the total time. The rest of the time is mostly spent in the construction of the Kohn–Sham matrix, which in the GPW algorithm is dominated by operations on the density grid, such as Fast Fourier Transforms (FFT) and halo exchanges. On the basis of these results, it can be expected that most other systems will be very strongly dominated by the matrix multiplications. Indeed, the choice of basis set and atom density for the current test system can be considered extremely biased toward operations on the grid, and away from the matrix multiplication. However, it demonstrates linear scaling up to 1 million molecules and the capability of our GPW implementation to deal with unit cells that exceed 3 × 10^4 nm^3 in volume.

Timings for actual simulations are more likely to be similar to the results for bulk liquid water presented in Figure 2. For these simulations, we employ three different methods, namely, DFT, NDDO, and DFTB, all with $\varepsilon_{\text{filter}} = 10^{-7}$. For DFT, all atoms are described using either a DZVP-MOLOPT-GTH basis set or a minimal SZV-MOLOPT-GTH basis set. Key properties for these systems are summarized in Table 1. For Figure 2, the first

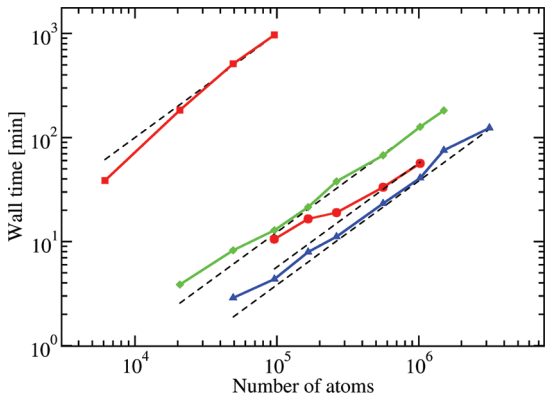

Figure 2. Total walltime for the energy calculation of bulk liquid water using DFT (red, squares and circles), NDDO (green, diamonds), and DFTB (blue, triangles) with 46656, 9216, and 9216 cores, respectively. DFT calculations are presented for two different basis sets, DZVP-MOLOPT-GTH (upper red curve, squares) and SZV-MOLOPT-GTH (lower red curve, circles). Dashed lines represent ideal linear scaling. All calculations employed $\varepsilon_{\text{filter}} = 10^{-7}$.

Table 1. Key Quantities for the Calculations on Bulk Water As Described with Various Methods[a]

	DFT SZV	DFT DZVP	DFTB	NDDO
# BF	6	23	6	12
cond. number	6.7	1800	5.1	1.0
ε_{min}	−0.82	−0.84	−0.89	−1.21
$\varepsilon_{\text{homo}}$	−0.03	−0.11	−0.23	−0.48
$\varepsilon_{\text{lumo}}$	0.46	0.05	0.38	0.16
ε_{max}	0.72	5.34	0.71	36749
μ	0.15	−0.05	−0.20	−0.20
# SCF iter.	8	7	8	18
# sign iter./SCF	9.8	20.7	13.2	33.2
energy error	1.7×10^{-7}	1.9×10^{-7}	0.9×10^{-7}	6.0×10^{-6}
trace error	4.5×10^{-7}	4.5×10^{-7}	4.0×10^{-6}	5.5×10^{-7}

[a]The numbers have been obtained for a system containing 256 water molecules, so that exact diagonalization is unproblematic and refer to the values at SCF convergence. # BF: number of basis functions per water molecule. cond. number: ratio of the largest to smallest eigenvalue of the overlap matrix. ε_{min}: minimum eigenvalue of the Hamiltonian matrix (a.u.). $\varepsilon_{\text{homo}}$: eigenvalue of HOMO eigenstate (a.u.). $\varepsilon_{\text{lumo}}$: eigenvalue of LUMO eigenstate (a.u.). ε_{max}: maximum eigenvalue of the Hamiltonian matrix (a.u.). μ: value of the employed chemical potential (a.u.). # SCF iter.: number of self consistent iterations. # sign iter./SCF: average number of sign iterations per SCF iteration. Energy error: difference per molecule between the total energy (a.u.) as obtained from the linear scaling algorithm and direct diagonalization. Trace error: absolute error in trace(PS).

striking observation is the computational cost of the DZVP-MOLOPT-GTH calculations, the largest calculation (96 000 atoms, 736 000 basis functions) requires approximately 16 h on 46 656 cores. In this case, matrix multiplications account for approximately 99% of the total run time. For our choice of basis and threshold, the density matrix is 17% occupied, $S^{-1/2}$ is 10% occupied, S is 3% occupied, and the code is in the linear scaling regime. The onset of linear scaling is approximately at 20 736 atoms (unit cell with 6 nm edges), in which case the sign matrix is 50% occupied, while for the next smaller system tested (6144 atoms, 4 nm edge) this occupation is nearly 100%. The computational cost with the semiempirical methods and the minimal basis DFT is at least 2 orders of magnitude lower than that of the DFT calculations employing a reasonably accurate basis set. The reason for this is not the reduced cost of computing the Hamiltonian matrix but the fact that the sparsities of the matrices are much more favorable. Compared to the DFT numbers quoted above, the density matrix is 0.4% occupied for NDDO and 0.8% for DFTB. The superior DFTB performance relative to NDDO is related to the fact that the number of SCF and sign function iterations needed is smaller, a feature that is related to the large spectral width of the NDDO Hamiltonian (see Table 1, due to the unusual NDDO parametrization of ref 81). It is interesting to note that the NDDO approach only spends 90% of the time in the matrix multiplications for the largest system. Contrary to the DFT case (GPW) and the DFTB case (SPME), the NDDO code employs a standard Ewald method to compute the electrostatic interactions, which scales as $O(N^{3/2})$. For systems containing tens of millions of atoms, this term will thus dominate, and another approach needs to be adopted. Finally, whereas the largest system size that could be computed with the DZVP-

MOLOPT-GTH basis was determined by the computational cost, the largest system size computable with the other methods was determined by the memory constraints of approximately 1.6 GB/MPI rank. The main memory usage is not related to the actual data of the matrices, which is well distributed, but to replicated information about all atoms, such as coordinates, velocities, forces, and atomic type information. The impact of this replication can be mitigated in a mixed MPI/OMP scheme, where the full memory of the node is available to a single MPI rank, allowing for a roughly 1 order of magnitude increase in system size. Ultimately, only a scheme where the atomic info is also fully distributed will allow for calculations beyond a few tens of millions of particles.

5. MATRIX SPARSITIES AND MULTIPLICATION COSTS

The sparsities of the matrices that occur in the calculation are key factors that determine the cost of the calculation. In our approach, these sparsities are not fixed but vary, subject to the filtering threshold ($\varepsilon_{\text{filter}}$). The cost of individual matrix multiplications can thus vary dramatically depending on the actual sparsity of the matrices, even for the same system, described with the same method. This is illustrated in Figure 3

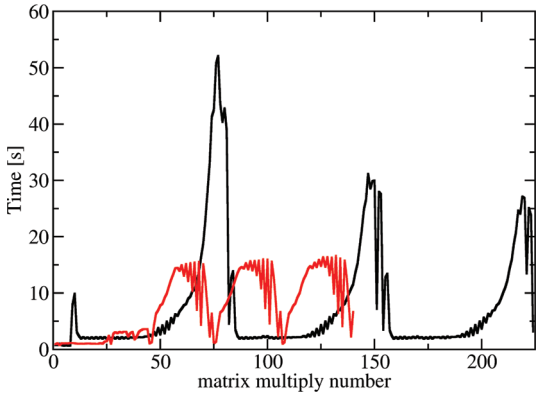

Figure 3. Timings of individual matrix multiplications in the order executed during a wave function optimization, including the initial inversion of S, and the first three SCF steps for a system containing approximately 1 million atoms on 9216 cores. Black data are for the NDDO method, while red data are for the DFTB method. Large variations in timing result from varying sparsity patterns during the Newton–Schulz iterations.

where the timing of individual matrix multiplications is reported from the beginning of the SCF procedure until the third mixing step. These data show that the cost can vary by more than 1 order of magnitude. For example, for the NDDO method, early iterations of the sign iteration are relatively cheap, but a steep increase in cost occurs closer to convergence. The fraction occupation is initially 2×10^{-5} but increases to as much as 1×10^{-3}. As observed previously in ref 8, the two multiplications within the Newton–Schulz iteration can vary significantly in cost as well, the second one being up to 3 times faster. These results suggest that merely counting the number of matrix multiplications is not a good measure for the performance of an algorithm in the sparse regime, even though this is valuable information for the limit in which the matrices are full.

Another example of how the sparsity of intermediate results can affect the cost of a calculation dramatically is related to the problem of fixing the chemical potential. Conventionally, the chemical potential μ is fixed by requiring that the trace of the density matrix equals the number of electrons. The proper value of the chemical potential is fixed by bisection or search in a suitable interval until the correct number of electrons has been obtained. For the benchmarks reported earlier, a value of μ in the middle of the band gap could easily be determined a priori, and bisection was not necessary. The cost of such a bisection can be prohibitive, as the sparsity of the intermediate matrices involved can depend strongly on the value of the chemical potential. For example, if during the bisection procedure the chemical potential is placed in the middle of a dense band of states, the corresponding density matrix will have metal-like properties and be nearly fully dense. However, the precise location within the gap can also critically affect performance. This is illustrated in Figure 4, where the cost of

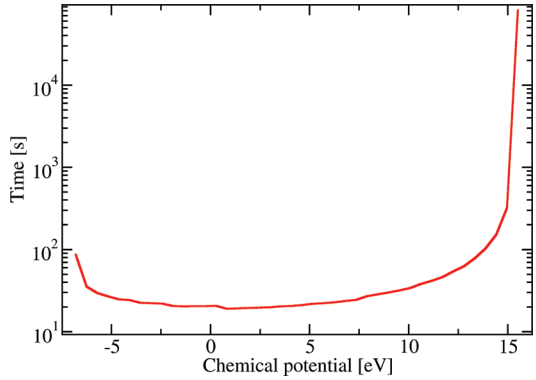

Figure 4. Time required to compute the sign matrix expression as a function of the chemical potential (μ) for the water system containing 32 928 atoms on 3600 cores, as described with DFTB. Except for the two extreme points, all values of μ are within the HOMO–LUMO gap.

the sign matrix iteration is shown as a function of the chemical potential for the liquid water simulations based on DFTB. Approaching either the HOMO or the LUMO of the system, the cost increases steeply. Note the logarithmic scale in Figure 4. In this system, the profile is not symmetric, approaching the LUMO, which is likely more delocalized, impacts the cost more significantly than approaching the HOMO. Calculations with a chemical potential slightly smaller than the LUMO yield the correct density matrix, but at a cost that is almost 2 orders of magnitude larger than for a location in the middle of the gap. For values near the LUMO, the matrices become essentially dense, prohibiting practical calculations. Using knowledge about the location of the HOMO and LUMO can thus be mandatory in efficiently performing calculations. In this context, it would be interesting to investigate the performance of trace correcting purification methods,[10,11] which essentially determine the chemical potential dynamically during the iteration process, and investigate if methods exist that guarantee optimal sparsity throughout the iteration procedure.

Finally, it is interesting to compare the sparsities of four critically important matrices in the SCF procedure, the overlap matrix, the Kohn–Sham matrix, the density matrix, and the inverse overlap matrix. In Figure 5, this information is displayed for the water system containing 96 000 atoms at the DZVP-MOLOPT-GTH level. For various values of a threshold on the norm of atomic blocks, the occupation of the matrix is reported. To investigate the effect of filtering during the iterative calculation of the density and inverse overlap matrix, results obtained with $\varepsilon_{\text{filter}} = 10^{-7}$ and $\varepsilon_{\text{filter}} = 10^{-5}$ are compared. This

dx.doi.org/10.1021/ct200897x | J. Chem. Theory Comput. 2012, 8, 3565–3573

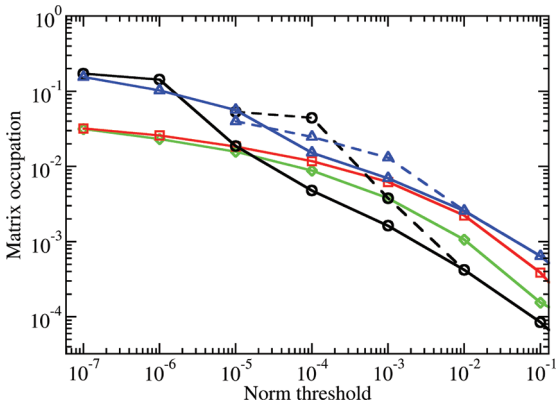

Figure 5. The fraction of matrix blocks of the density matrix (black, circles), the inverse overlap matrix (blue, triangles), the Kohn−Sham matrix (green, diamonds), and the overlap matrix (red, squares) for which the norm exceeds a given threshold. The density and inverse overlap matrix have been obtained iteratively with a filtering threshold of 10^{-7} (solid lines) or 10^{-5} (dashed lines).

comparison reveals that for a threshold close to $\varepsilon_{\text{filter}}$, the occupation is significantly above the expected one. For example, for $\varepsilon_{\text{filter}} = 10^{-5}$, the occupation at a threshold of 10^{-4} is almost 1 order of magnitude above the occupation obtained with the more accurate $\varepsilon_{\text{filter}} = 10^{-7}$ and the same threshold. It thus seems that numerical noise introduced during the iterative procedure leads to a large number of blocks with norms that are within a factor of 100 of the filtering threshold. A procedure that would avoid an accumulation of such blocks would greatly accelerate the calculation. In this context, variable thresholding[12] might be an interesting strategy, which we intend to explore in the future. Focusing on the parts of the curve that are not affected by the noise of the calculation, it is observed that the inverse overlap matrix has the largest occupation, even though it is only marginally more dense than the overlap matrix. Somewhat surprisingly, we see that the density matrix is more sparse than the overlap and Kohn−Sham matrix. This result, of course, is expected to depend on properties of the system (such as the band gap) and the basis set. In particular, we observe this only for the DZVP-MOLOPT-GTH basis, which includes contracted diffuse primitives but is nevertheless rather well conditioned.

6. CRITICAL ASSESSMENT OF LINEAR SCALING PERFORMANCE

In the previous section, it has been demonstrated that linear scaling calculations can be performed with the current methodology. Here, this achievement is critically analyzed, and short-comings of the implementation and algorithms are investigated. It will be shown that nonlinearity is hidden by maintaining parameters of the fixed calculation.

First, all curves shown in Figures 1 and 2 have been generated using a fixed number of parallel computing tasks for all system sizes. This is a reasonable approach, and the only approach for serial codes. However, as the system size is increased, it is natural to increase the computer resources proportionally. Ideally, calculations of arbitrary size should be performed in a constant time to solution. More explicitly, the wall time needed to perform a calculation on N atoms on N cores should be constant. In a weak scaling experiment, system size and the number of parallel processes are increased simultaneously. Results for such an experiment are shown in Figure 6, where we focus on the time spent in the matrix multiplication only. It can be observed that the wall time

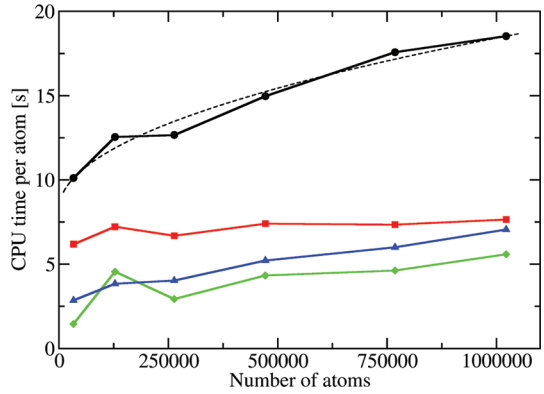

Figure 6. A weak scaling experiment using DFTB for bulk liquid water with approximately 200 atoms/MPI task for systems ranging from 32 928 atoms on 144 cores to 1 022 208 atoms on 5184 cores. Shown is the CPU time per atom spent in total (black, circles), for the atomic block matrix multiplications (red, squares), for the communication between processes (green, diamonds), and for the book-keeping overhead (blue, triangles). The time spent in the multiplications is approximately constant, but communication and book-keeping have a contribution that grows with the square root of the number of MPI tasks (atoms). The dashed line is a fit to the data using $f(N) = a + b\sqrt{N}$.

needed is not constant but can be fitted well as $a + b\sqrt{N}$, and hence total used resources increases as $aN + bN^{3/2}$. Results of a strong scaling experiment, in which the computer resources are increased for a fixed system size, are reported in Figure 7. At the

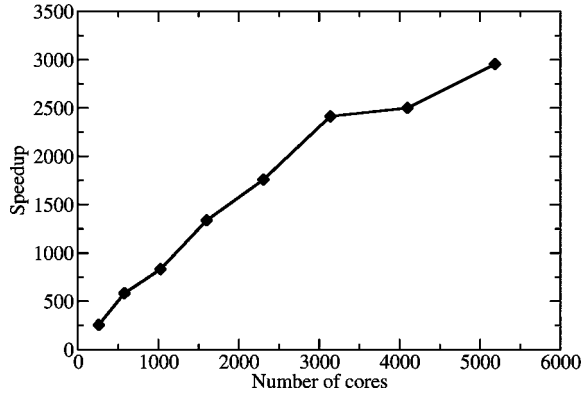

Figure 7. A strong scaling experiment using DFTB for bulk liquid water for a system with 165 888 atoms. Shown is the measured speedup assuming ideal speedup from 1 to 256 cores. These calculations ran on Cray XE6 hardware.

largest number of cores, approximately 3 s is needed per matrix multiplication. As shown in Figure 6, the total time of the parallel matrix multiplication consists of essentially three parts: (1) the local multiplication (the actual flops), (2) book-keeping overhead related to the sparse nature of the matrix, and (3) communication of the matrix data. The actual number of flops shows proper constant cost; the $\sqrt{N}$ scaling comes from the overhead related to book-keeping and communication. To fully explain this behavior, a detailed description of the matrix multiplication library in CP2K would be needed, which is beyond the scope of the current manuscript and will be

dx.doi.org/10.1021/ct200897x | J. Chem. Theory Comput. 2012, 8, 3565−3573

discussed elsewhere.[63] It suffices to say that this behavior is related to Cannon's algorithm,[73] i.e., the parallelization scheme employed for the matrix multiplication, which requires communication of the local matrix data to $\sqrt{N}$ processors. While this scheme is advantageous in the dense limit, where other commonly used schemes would require communication to N processors and thus sacrifice strong scaling, it is suboptimal in the sparse limit, where it is possible to communicate with only a fraction of the total number of processors. To put this result in perspective, it is useful to consider how the constants a and b depend on properties of the system simulated, in particular, the number of basis functions per atom (k). Actual flops will scale as k^3 and the amount of data communicated as k^2, while the book-keeping overhead is constant per block. Furthermore, the ratio of communication to flops also depends strongly on the sparsity of the matrix, being less favorable in the sparse case. The DFTB case considered here, where atoms have only one or four basis functions and the sparsity is significant, should therefore be considered as a worst-case scenario for the computation to communication ratio. Even in this case, for 1 million atoms, the multiplication is only by a factor of 3 slower compared to the case where overhead and communication would be completely eliminated. Nevertheless, it is important to improve this scaling behavior, and new developments in our group aim at combining both good strong and weak scaling behavior for the matrix multiplication.

A second, more important parameter is the employed filtering threshold. In the calculations reported so far, it was fixed at 10^{-7}, which is sufficiently strict to be able to perform an accurate calculation on 1 million atoms. However, it is an important concern that larger systems might need a tighter threshold. Indeed, if an absolute error in the total energy is required, it is clear that the relative accuracy will have to be improved, and thus the threshold will have to be decreased as the system size increases. This problem is commonly avoided by requiring the total energy per electron to be accurate to a given threshold. This is likely to be sufficient for many purposes, such as, for example, relaxing structures or performing molecular dynamics, which require accurate forces, i.e., local quantities only. However, in the current algorithm, one extensive quantity is still needed accurately, namely, the total number of electrons. As in most other codes, the chemical potential is determined requiring that the total number of electrons equals a prescribed number. If the number of electrons is not computed accurately, the bisection procedure could easily lead to a wrong chemical potential, which would lead to ionization of species with high lying occupied or low lying unoccupied orbitals. The trace of the density matrix thus needs to be computed with absolute accuracy if the bisection method is used, requiring a threshold that decreases proportionally to the system size. For systems containing 1 million atoms, using the above-mentioned threshold of 10^{-7} leads to an absolute accuracy in the trace of better than 0.5, while relative quantities such as the number of electrons or energy *per molecule* are accurate to the specified threshold. As shown in Figure 8, decreasing the filtering threshold comes at a cost, and a fit to our timing results suggests that the cost increases as $a\varepsilon^{-1/3}$. If a variable threshold, as required by the system size, is employed, the computation cost will thus scale as $N^{4/3}$. From this result, we can guess that million-atom calculations can be performed at an approximately 5 times smaller cost if a different scheme is developed to determine the chemical potential.

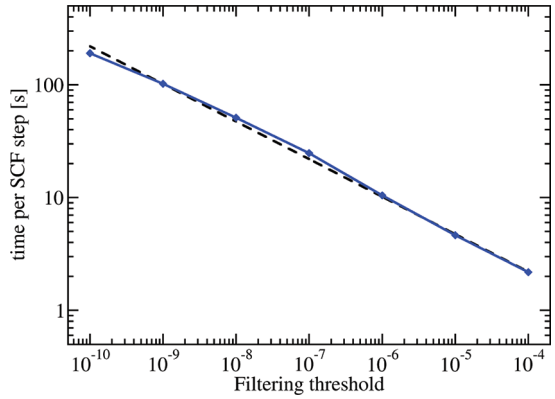

Figure 8. Cost per SCF step for a DFTB calculation on 6912 water molecules as a function of the filtering threshold for the matrix multiplication. The dashed line represents a fit using the functional form $a\varepsilon^{-1/3}$. The error in the trace and the total energy at full SCF convergence is linear in the threshold.

7. CONCLUSIONS

The feasibility of performing self-consistent calculations with millions of atoms has been investigated. We find that these calculations have become possible but remain computationally demanding if accurate basis sets are to be employed. Semiempirical calculations, which have basis sets defined as part of the method, or minimal basis set DFT can be performed relatively easily. We have quantified hidden nonlinearities associated with varying parameters of the calculation, such as the filtering threshold and the number of processes used in parallel calculations. These results suggest that for million-atom calculations, a 10−20 fold speedup remains a realistic target, which however requires more sophisticated methods, further developments, and testing on large scale systems.

AUTHOR INFORMATION

Corresponding Authors

*E-mail: Joost.VandeVondele@mat.ethz.ch.

§On leave from the National Institute of Chemistry, Ljubljana, Slovenia.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We acknowledge useful discussions with Chris Mundy and thank Valery Weber for integrating the sparse matrix library in CP2K. Iain Bethune contributed significantly to the optimization of the code. J.V. acknowledges financial support from the European Union FP7 in the form of an ERC Starting Grant under contract no. 277910. Calculations were enabled by 2008−2010 and 2011 INCITE awards on the CRAY XT5 using resources of the National Center for Computational Sciences at Oak Ridge National Laboratory (ORNL), which is supported by the Office of Science of the U.S. DOE under contract no. DE-AC05-00OR22725 and by the Swiss National Supercomputer Centre (CSCS). The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number RI-261557 and from the Swiss University Conference through the High Performance and High Productivity Computing (HP2C) Programme. This manuscript is dedicated to Wilfred F. van Gunsteren on the occasion of his

65th birthday. J.V. would like to thank Wilfred for introducing him—nearly 15 years ago—to linear scaling molecular mechanics calculations with millions of atoms in the condensed phase.

REFERENCES

(1) Kohn, W. *Phys. Rev. Lett.* **1996**, *76*, 3168−3171.
(2) Galli, G. *Curr. Opin. Solid State Mater. Sci.* **1996**, *1*, 864−874.
(3) Goedecker, S. *Rev. Mod. Phys.* **1999**, *71*, 1085−1123.
(4) Goedecker, S.; Scuseria, G. *Comput. Sci. Eng.* **2003**, *5*, 14−21.
(5) McWeeny, R. *Rev. Mod. Phys.* **1960**, *32*, 335−369.
(6) Palser, A.; Manolopoulos, D. *Phys. Rev. B* **1998**, *58*, 12704−12711.
(7) Beylkin, G.; Coult, N.; Mohlenkamp, M. *J. Comput. Phys.* **1999**, *152*, 32−54.
(8) Nemeth, K.; Scuseria, G. *J. Chem. Phys.* **2000**, *113*, 6035−6041.
(9) Holas, A. *Chem. Phys. Lett.* **2001**, *340*, 552−558.
(10) Niklasson, A. M. N. *Phys. Rev. B* **2002**, *66*, 155115.
(11) Niklasson, A. M. N.; Tymczak, C. J.; Challacombe, M. *J. Chem. Phys.* **2003**, *118*, 8611−8620.
(12) Rubensson, E. H.; Rudberg, E.; Salek, P. *J. Chem. Phys.* **2008**, *128*, 074106.
(13) Higham, N. J. *Functions of Matrices: Theory and Computation*; Society for Industrial and Applied Mathematics: Philadelphia, PA, 2008; pp 107−172.
(14) Goedecker, S.; Colombo, L. *Phys. Rev. Lett.* **1994**, *73*, 122−125.
(15) Baer, R.; Head-Gordon, M. *J. Chem. Phys.* **1997**, *107*, 10003−10013.
(16) Baer, R.; Head-Gordon, M. *Phys. Rev. Lett.* **1997**, *79*, 3962−3965.
(17) Bernstein, N. *Europhys. Lett.* **2001**, *55*, 52−58.
(18) Liang, W.; Saravanan, C.; Shao, Y.; Baer, R.; Bell, A.; Head-Gordon, M. *J. Chem. Phys.* **2003**, *119*, 4117−4125.
(19) Niklasson, A. *Phys. Rev. B* **2003**, *68*, 233104.
(20) Ceriotti, M.; Kuehne, T. D.; Parrinello, M. *J. Chem. Phys.* **2008**, *129*, 024707.
(21) Lin, L.; Lu, J.; Car, R. *Phys. Rev. B* **2009**, *79*, 115133.
(22) Li, X.; Nunes, R.; Vanderbilt, D. *Phys. Rev. B* **1993**, *47*, 10891−10894.
(23) Mauri, F.; Galli, G. *Phys. Rev. B* **1994**, *50*, 4316−4326.
(24) Kim, J.; Mauri, F.; Galli, G. *Phys. Rev. B* **1995**, *52*, 1640−1648.
(25) Millam, J.; Scuseria, G. *J. Chem. Phys.* **1997**, *106*, 5569−5577.
(26) Daniels, A.; Millam, J.; Scuseria, G. *J. Chem. Phys.* **1997**, *107*, 425−431.
(27) Challacombe, M. *J. Chem. Phys.* **1999**, *110*, 2332−2342.
(28) Galli, G. *Phys. Status Solidi B* **2000**, *217*, 231−249.
(29) Helgaker, T.; Larsen, H.; Olsen, J.; Jorgensen, P. *Chem. Phys. Lett.* **2000**, *327*, 397−403.
(30) Li, X.; Millam, J.; Scuseria, G.; Frisch, M.; Schlegel, H. *J. Chem. Phys.* **2003**, *119*, 7651−7658.
(31) Fattebert, J.; Gygi, F. *Comput. Phys. Commun.* **2004**, *162*, 24−36.
(32) Salek, P.; Host, S.; Thogersen, L.; Jorgensen, P.; Manninen, P.; Olsen, J.; Jansik, B.; Reine, S.; Pawlowski, F.; Tellgren, E.; Helgaker, T.; Coriani, S. *J. Chem. Phys.* **2007**, *126*, 114110.
(33) Shao, Y.; Saravanan, C.; Head-Gordon, M.; White, C. *J. Chem. Phys.* **2003**, *118*, 6144−6151.
(34) Dixon, S.; Merz, K. *J. Chem. Phys.* **1996**, *104*, 6643−6649.
(35) Lee, T.; York, D.; Yang, W. *J. Chem. Phys.* **1996**, *105*, 2744−2750.
(36) Van der Vaart, A.; Gogonea, V.; Dixon, S.; Merz, K. *J. Comput. Chem.* **2000**, *21*, 1494−1504.
(37) Li, S.; Li, W.; Fang, T. *J. Am. Chem. Soc.* **2005**, *127*, 7215−7226.
(38) Li, W.; Li, S.; Jiang, Y. *J. Phys. Chem. A* **2007**, *111*, 2193−2199.
(39) Wang, L.-W.; Zhao, Z.; Meza, J. *Phys. Rev. B* **2008**, *77*, 165113.
(40) He, X.; Merz, K. M. Jr. *J. Chem. Theory Comput.* **2010**, *6*, 405−411.
(41) Varga, K. *Phys. Rev. B* **2010**, *81*, 045109.
(42) Ozaki, T.; Terakura, K. *Phys. Rev. B* **2001**, *64*, art. no. 195126.
(43) Ozaki, T. *Phys. Rev. B* **2006**, *74*, 245101.
(44) Ozaki, T. *Phys. Rev. B* **2010**, *82*, 075131.
(45) Maslen, P.; Ochsenfeld, C.; White, C.; Lee, M.; Head-Gordon, M. *J. Phys. Chem. A* **1998**, *102*, 2215−2222.
(46) Daniels, A.; Scuseria, G. *J. Chem. Phys.* **1999**, *110*, 1321−1328.
(47) Scuseria, G. *J. Phys. Chem. A* **1999**, *103*, 4782−4790.
(48) Kudin, K.; Scuseria, G. *Phys. Rev. B* **2000**, *61*, 16440−16453.
(49) Bowler, D.; Miyazaki, T.; Gillan, M. *J. Phys.: Condens. Matter* **2002**, *14*, 2781−2798.
(50) Soler, J.; Artacho, E.; Gale, J.; Garcia, A.; Junquera, J.; Ordejon, P.; Sanchez-Portal, D. *J. Phys.: Condens. Matter* **2002**, *14*, 2745−2779.
(51) Skylaris, C.; Haynes, P.; Mostofi, A.; Payne, M. *J. Chem. Phys.* **2005**, *122*, 084119.
(52) Bowler, D.; Choudhury, R.; Gillan, M.; Miyazaki, T. *Phys. Status Solidi B* **2006**, *243*, 989−1000.
(53) Skylaris, C.-K.; Haynes, P. D. *J. Chem. Phys.* **2007**, *127*, 164712.
(54) Artacho, E.; Anglada, E.; Dieguez, O.; Gale, J. D.; Garcia, A.; Junquera, J.; Martin, R. M.; Ordejon, P.; Pruneda, J. M.; Sanchez-Portal, D.; Soler, J. M. *J. Phys.: Condens. Matter* **2008**, *20*, 064208.
(55) Havu, V.; Blum, V.; Havu, P.; Scheffler, M. *J. Comput. Phys.* **2009**, *228*, 8367−8379.
(56) Hine, N. D. M.; Haynes, P. D.; Mostofi, A. A.; Skylaris, C.-K.; Payne, M. C. *Comput. Phys. Commun.* **2009**, *180*, 1041−1053.
(57) Shang, H.; Xiang, H.; Li, Z.; Yang, J. *Int. Rev. Phys. Chem.* **2010**, *29*, 665−691.
(58) Bowler, D. R.; Miyazaki, T. *J. Phys.: Condens. Matter* **2010**, *22*, 074207.
(59) Iwata, J.-I.; Takahashi, D.; Oshiyama, A.; Boku, T.; Shiraishi, K.; Okada, S.; Yabana, K. *J. Comput. Phys.* **2010**, *229*, 2339−2363.
(60) Rudberg, E.; Rubensson, E. H.; Salek, P. *J. Chem. Theory Comput.* **2011**, *7*, 340−350.
(61) Hung, L.; Carter, E. A. *Chem. Phys. Lett.* **2009**, *475*, 163−170.
(62) VandeVondele, J.; Krack, M.; Mohamed, F.; Parrinello, M.; Chassaing, T.; Hutter, J. *Comput. Phys. Commun.* **2005**, *167*, 103−128.
(63) In preparation , 2012.
(64) Higham, N. *Numer. Algor.* **1997**, *15*, 227−242.
(65) Jansik, B.; Host, S.; Jorgensen, P.; Olsen, J.; Helgaker, T. *J. Chem. Phys.* **2007**, *126*, 124104.
(66) Hartree, D. R. *The Calculation of Atomic Structures*; John Wiley and Sons, Inc.: New York, 1957; pp 87−88.
(67) Challacombe, M. *Comput. Phys. Commun.* **2000**, *128*, 93−107.
(68) Bowler, D.; Miyazaki, T.; Gillan, M. *Comput. Phys. Commun.* **2001**, *137*, 255−273.
(69) Saravanan, C.; Shao, Y.; Baer, R.; Ross, P.; Head-Gordon, M. *J. Comput. Chem.* **2003**, *24*, 618−622.
(70) Rubensson, E. H.; Rudberg, E.; Salek, P. *J. Comput. Chem.* **2007**, *28*, 2531−2537.
(71) Hine, N. D. M.; Haynes, P. D.; Mostofi, A. A.; Payne, M. C. *J. Chem. Phys.* **2010**, *133*, 114111.
(72) The CP2K developers group. http://www.cp2k.org/ (accessed March 2012).
(73) Cannon, L. E. Ph.D. Thesis, 1969. AAI7010025.
(74) Lippert, G.; Hutter, J.; Parrinello, M. *Mol. Phys.* **1997**, *92*, 477−487.
(75) Lippert, G.; Hutter, J.; Parrinello, M. *Theor. Chem. Acc.* **1999**, *103*, 124−140.
(76) Goedecker, S.; Teter, M.; Hutter, J. *Phys. Rev. B* **1996**, *54*, 1703−1710.
(77) Guidon, M.; Schiffmann, F.; Hutter, J.; VandeVondele, J. *J. Chem. Phys.* **2008**, *128*, 214104.
(78) Guidon, M.; Hutter, J.; VandeVondele, J. *J. Chem. Theory Comput.* **2009**, *5*, 3010−3021.
(79) Guidon, M.; Hutter, J.; VandeVondele, J. *J. Chem. Theory Comput.* **2010**, *6*, 2348−2364.
(80) Pople, J.; Beveridge, D. *Approximate Molecular Orbital Theory*; McGraw-Hill: New York, 1970.
(81) Chang, D. T.; Schenter, G. K.; Garrett, B. C. *J. Chem. Phys.* **2008**, *128*, 164111.

(82) Elstner, M.; Porezag, D.; Jungnickel, G.; Elsner, J.; Haugk, M.; Frauenheim, T.; Suhai, S.; Seifert, G. *Phys. Rev. B* **1998**, *58*, 7260−7268.
(83) VandeVondele, J.; Hutter, J. *J. Chem. Phys.* **2007**, *127*, 114105.