

## Kohn–Sham Density Functional Theory Electronic Structure Calculations with Linearly Scaling Computational Time and Memory Usage

Elias Rudberg,<sup>\*,†</sup> Emanuel H. Rubensson,<sup>†</sup> and Paweł Sałek<sup>‡</sup>

*Division of Scientific Computing, Department of Information Technology, Uppsala University, Box 337, SE-751 05 Uppsala, Sweden, and PS Consulting, ul. Zaporoska 8/4, 30-389 Kraków, Poland*

Received October 26, 2010

**Abstract:** We present a complete linear scaling method for hybrid Kohn–Sham density functional theory electronic structure calculations and demonstrate its performance. Particular attention is given to the linear scaling computation of the Kohn–Sham exchange-correlation matrix directly in sparse form within the generalized gradient approximation. The described method makes efficient use of sparse data structures at all times and scales linearly with respect to both computational time and memory usage. Benchmark calculations at the BHandHLYP/3-21G level of theory are presented for polypeptide helix molecules with up to 53 250 atoms. Threshold values for computational approximations were chosen on the basis of their impact on the occupied subspace so that the different parts of the calculations were carried out at balanced levels of accuracy. The largest calculation used 307 204 Gaussian basis functions on a single computer with 72 GB of memory. Benchmarks for three-dimensional water clusters are also included, as well as results using the 6-31G\*\* basis set.

### 1. Introduction

Recent developments of linear scaling algorithms together with the availability of larger computer resources have made it possible to carry out electronic structure calculations for systems with many thousands of atoms,<sup>1–6</sup> using Hartree–Fock (HF) and Kohn–Sham density functional theory (KS-DFT), as well as tight-binding and MNDO-type semiempirical methods. Linear scaling algorithms have been developed for all computationally expensive parts of such calculations, including the computation of the Coulomb,<sup>7–9</sup> HF exchange,<sup>10,11</sup> and KS-DFT exchange-correlation<sup>12,13</sup> matrices and methods for the density matrix construction step.<sup>14–18</sup> However, making efficient use of linear scaling methods is not straightforward, and quadratically or worse scaling methods are still applied even for molecular systems with many thousands of atoms, where the use of linear

scaling algorithms would greatly reduce the computational cost, see, e.g., ref 19.

The development of linear scaling algorithms has mostly focused on achieving linear scaling in time, and the memory usage aspect has often been overlooked. Whereas linear scaling in memory can in principle be achieved by replacing dense matrix storage with a sparse matrix representation, the overhead from locating and modifying elements in the sparse storage can be considerable. Often, linear scaling algorithms have been described without considering the matrix representation, thus implicitly assuming fast access to matrix elements. In practice, efficient use of sparse matrix storage requires that the way that matrix elements are accessed is considered as an integral part of each algorithm and that the different parts of the calculation are combined efficiently while using only sparse data structures. Also, in order to achieve sufficient accuracy in the result, the various threshold values used in different parts of the calculation need to be chosen carefully. Due to such difficulties, linear scaling methods are still far from reaching their full potential.

\* Corresponding author e-mail: elias.rudberg@it.uu.se.

† Uppsala University.

‡ PS Consulting.

This paper builds on our earlier work on linearly scaling HF calculations.<sup>4</sup> We describe the necessary changes needed to efficiently evaluate the exchange-correlation contributions to the Kohn–Sham matrix with linear scaling in both time and memory. In particular, an efficient way of accessing sparse matrix elements is described, with an overhead comparable to that of dense matrix storage. The new algorithm has been implemented in the Ergo program.<sup>20</sup> We also describe how the exchange-correlation contributions can be evaluated at an accuracy level consistent with the other parts of the calculation.

This paper is organized as follows: Section 2 gives an overview of the KS-DFT method from a computational point of view. Our algorithm for linear scaling construction of the exchange-correlation matrix directly in sparse form is described in section 3. Benchmark calculations demonstrating linear scaling behavior are presented in section 4. Finally, a few concluding remarks are given in section 5.

## 2. Method

We consider calculations where the electron density is expanded in a set of  $n$  basis functions  $\{\phi_k(r)\}$  built up by combinations of polynomials and Gaussian functions usually centered at the nuclei of the molecule. These basis sets are commonly referred to as Gaussian type linear combination of atomic orbital (GT-LCAO) basis sets or simply Gaussian basis sets. See ref 21 for a thorough discussion about such basis sets.

The sequence of steps illustrated in Algorithm 1 summarizes how a linear scaling KS-DFT calculation is carried out in the Ergo quantum chemistry program.<sup>20</sup> The Ergo program uses Gaussian basis sets to compute electronic structures with linearly scaling processor time and memory usage.

```

Algorithm 1. Overview of KS-DFT Self-Consistent Field
Program
1:   Read molecule and basis set information from
      input files.
2:   Compute overlap matrix  $S$  and inverse factor  $Z$ 
      such that  $Z^T S Z = I$ .
3:   Compute one-electron Hamiltonian matrix  $H_1$ .
4:   Generate starting guess density matrix  $D$ .
5:   for  $i = 1, 2, \dots$  do
6:     Compute new Coulomb matrix  $J$ .
7:     Compute new HF exchange matrix  $K$ .
8:     Compute new Kohn–Sham exchange-correlation
        matrix  $V_{xc}$  and energy  $E_{xc}$ 
9:     Compute energy  $E = Tr(DH_1) + \frac{1}{2}Tr(D(J + \gamma K))$ 
        +  $E_{xc}$ .
10:    Compute new Kohn–Sham matrix  $F = H_1 + J + \gamma K + V_{xc}$ .
11:    Compute  $\tilde{F}$  as a linear combination of new and
        previous Kohn–Sham matrices.
12:    Compute  $F_\perp = Z^T \tilde{F} Z$ .
13:    Compute new density matrix  $D_\perp$  from  $F_\perp$ .
14:    Compute  $D = Z D_\perp Z^T$ .
15:   end for
```

In the self-consistent field (SCF) procedure given by Algorithm 1, two main operations are repeated: (1) the  $D \rightarrow F$  step, for the construction of the Kohn–Sham matrix for a given density matrix, consisting of steps 6–10 of Algorithm 1, and (2) the  $F \rightarrow D$  step, for the subsequent construction of a new density matrix, consisting of steps 12–14. These two

operations can be employed in a simple fixed point iteration, but usually some convergence enhancing schemes are used to accelerate and hopefully even ensure convergence, see refs 22 and 23 for recent reviews. In our calculations, in each iteration, either damping<sup>24,25</sup> or DIIS<sup>26,27</sup> is used in step 11 to generate  $\tilde{F}$  as a linear combination of new and previous Kohn–Sham matrices.

The Kohn–Sham matrix  $F$  consists of one-electron ( $H_1$ ) and two-electron ( $J$ ,  $K$ ,  $V_{xc}$ ) contributions:  $F = H_1 + J + \gamma K + V_{xc}$ . In the case of a HF calculation,  $V_{xc} = 0$ ,  $E_{xc} = 0$ , and  $\gamma = 1$ . In the case of a pure Kohn–Sham calculation,  $\gamma = 0$ . For so-called hybrid functionals,  $V_{xc}$  and  $E_{xc}$  are both nonzero and  $\gamma \neq 0$ . Becke's half-and-half functional with an LYP correlation part (BHandHLYP),<sup>28</sup> used in the benchmark calculations described in section 4, is a hybrid functional with  $\gamma = 0.5$ .

The Coulomb matrix  $J$  can be efficiently calculated using truncated multipole expansions.<sup>8,9,29–32</sup> An important feature of our implementation is the use of a dynamically selected multipole expansion order, an approach that gives significant speedups compared to always using the same expansion order.<sup>8,9</sup> Truncated multipole expansions are also used for linear scaling computation of the electron-nuclei term of the one-electron Hamiltonian matrix  $H_1$ .

The HF exchange matrix  $K$  can be computed in linear time by exploiting the locality of basis functions together with the sparsity of the density matrix. There has been much research devoted to efficient computation of the exchange matrix.<sup>10,11,33–38</sup> Some details about the exchange matrix evaluation in the Ergo code, including memory usage considerations, can be found in ref 4.

A key result of this article is our algorithm for linear scaling construction of the Kohn–Sham exchange-correlation matrix  $V_{xc}$  directly in sparse form. This algorithm is described in section 3.

The density matrix  $D_\perp$  is computed from  $F_\perp$  using the purification scheme of ref 18 combined with a novel approach for the removal of small matrix elements.<sup>39</sup> The most distinguished feature of this purification scheme, which uses the so-called trace-correcting purification polynomials of the second order,<sup>17</sup> is that it allows for rigorous control of the error in the occupied subspace. This purification procedure is formulated for an orthogonal basis. Therefore, an inverse factor  $Z$  of the overlap matrix is needed for the congruence transformations to and from the orthogonal basis in steps 12 and 14 of Algorithm 1. In the calculations considered in this paper, the inverse factor was computed using inverse Cholesky decomposition,<sup>40–42</sup> although other choices for  $Z$  are possible as well.<sup>43–45</sup>

Matrix operations needed particularly in density matrix purification but also for other operations are performed using sparse matrix algebra. This allows for linear scaling provided that the matrix sparsity is such that the average number of nonzero elements per row does not increase with system size. This is usually the case for large nonmetallic molecular systems.

For simplicity, the description above was given for the common case of a spin-restricted calculation. However, this can be straightforwardly generalized to the spin-unrestricted

case. Then, the electron densities for  $\alpha$  and  $\beta$  spin are represented by separate density matrices  $D_\alpha$  and  $D_\beta$  so that the total electron density matrix is given by  $D = D_\alpha + D_\beta$ , and similarly two Kohn–Sham matrices  $F_\alpha$  and  $F_\beta$  are created. Two Kohn–Sham exchange-correlation matrices  $V_{xc;\alpha}$  and  $V_{xc;\beta}$  are also used, formed from the electron densities  $\rho_\alpha$  and  $\rho_\beta$  as described in the following section.

### 3. Linear Scaling Computation of the Kohn–Sham Exchange-Correlation Matrix

The Kohn–Sham formulation of density functional theory<sup>46</sup> allows one to formulate the framework for density functional theory calculations in a way similar to the HF framework with two important modifications. HF exchange is scaled down or entirely removed. Instead, an exchange-correlation term is added to the energy, and a corresponding contribution is added to the Fock matrix. A Fock matrix with an exchange-correlation contribution is traditionally called a Kohn–Sham matrix.

The exchange-correlation energy  $E_{xc}$  within the generalized gradient approximation (GGA) is given by

$$E_{xc} = \int_{\mathbb{R}^3} \mathcal{F}(\rho_\alpha(r), \rho_\beta(r), q_\alpha(r), q_\beta(r), q_{\alpha\beta}(r)) dr \quad (1)$$

where

$$q_\alpha(r) = |g_\alpha(r)| \quad (2)$$

$$q_{\alpha\beta}(r) = g_\alpha(r) \cdot g_\beta(r) \quad (3)$$

$$g_\alpha(r) = \nabla \rho_\alpha(r) \quad (4)$$

Here,  $\mathcal{F}(\rho_\alpha(r), \rho_\beta(r), q_\alpha(r), q_\beta(r), q_{\alpha\beta}(r)) \equiv \mathcal{F}(r)$  is the density functional which in the case of GGA also depends on the density gradient.

GGA assumes that the nonlocal character of the exchange-correlation contributions can be captured by making the functional dependent on the local value of the spin-dependent density gradient  $g_\alpha(r)$ . It also separately considers electron densities with spin-up  $\alpha$  and -down  $\beta$ , similarly as in the local spin density approximation. The densities  $\rho_\alpha$  and  $\rho_\beta$  are the same only in the special case of a closed shell calculation.

Matrix elements of the exchange-correlation matrix  $V_{xc;\alpha}$  under the GGA approximation are given by

$$V_{xc;\alpha;pq} = \int_{\mathbb{R}^3} [s_{\alpha;p}(r) \phi_q(r) + \phi_p(r) s_{\alpha;q}(r)] dr \quad (5)$$

where

$$\begin{aligned} s_{\alpha;p}(r) &= \phi_p(r) v_\alpha(r) + u_\alpha(r) \sum_{c \in \{x,y,z\}} \frac{\partial \phi_p(r)}{\partial c} \frac{\partial \rho_\alpha}{\partial c} + \\ &\quad t(r) \sum_{c \in \{x,y,z\}} \frac{\partial \phi_p(r)}{\partial c} \frac{\partial \rho_\alpha}{\partial c} \end{aligned} \quad (6)$$

$$v_\alpha(r) = \frac{\partial \mathcal{F}(r)}{\partial \rho_\alpha} \quad (7)$$

$$u_\alpha(r) = \frac{1}{q_\alpha(r)} \frac{\partial \mathcal{F}(r)}{\partial q_\alpha} \quad (8)$$

$$t(r) = \frac{\partial \mathcal{F}(r)}{\partial q_{\alpha\beta}} \quad (9)$$

In contrast to the integrals encountered in calculations of Coulomb repulsion and HF exchange, exchange-correlation integrals cannot be evaluated using a compact analytical expression. Instead, the exchange-correlation energy  $E_{xc}$  and the matrix elements  $V_{xc;\alpha;pq}$  are computed using numerical integration over a grid:

$$E_{xc} = \sum_i w_i \mathcal{F}(r_i) \quad (10)$$

$$V_{xc;\alpha;pq} = \sum_i w_i [s_{\alpha;p}(r_i) \phi_q(r_i) + \phi_p(r_i) s_{\alpha;q}(r_i)] \quad (11)$$

The choice of grid point locations  $\{r_i\}$  and associated grid weights  $\{w_i\}$  determines the quality of the integration grid. The electron density at a given grid point  $r_i$  is computed by contracting the density matrix  $D_\alpha$  with basis functions evaluated at  $r_i$ :

$$\rho_\alpha(r_i) = \sum_{pq} D_{\alpha;pq} \phi_p(r_i) \phi_q(r_i) \quad (12)$$

The calculation of the exchange-correlation matrix formally scales cubically with system size. The scaling can be reduced to linear if basis function screening is implemented.

**3.1. Grids for Numerical Integration.** Traditionally, the entire integration grid is constructed as a union of atomic grids, with grid weights  $\{w_i\}$  adjusted in the overlapping regions.<sup>47</sup> Atomic grids are constructed as outer products of Lebedev grids for angular integration<sup>48</sup> and Gauss–Chebychev radial grids.<sup>47,49</sup> Alternative radial grids have been proposed as well.<sup>50</sup> The weights in overlapping regions are adjusted using Becke partitioning or its variants.<sup>12,51</sup> Smooth switching functions used in the Becke partitioning process in principle stretch out infinitely. This makes the computational cost of the partitioning process scale cubically with system size. In practice, the right choice of multiplication order used in the grid weight scaling process can make the effort per atom roughly independent of the system size. Other partitioning schemes, see, e.g., ref 12, choose the partitioning function in a way that allows for trivial screening of atoms far away from the grid point associated with the weight being adjusted.

While grids constructed as unions of atomic grids are well established, the existence of overlapping regions in multiatom systems introduces errors that are difficult to control. The high accuracy that is possible for the integration of densities or exchange-correlation potentials for spherically symmetric systems like atoms cannot be realized in such cases. The complication of overlapping regions makes the error increase by several orders of magnitude. A grid construction method that in principle allows for integration of the electron density

up to any accuracy is the so-called hierarchical cubature (HiCu).<sup>13</sup> In the HiCu scheme, the quality of the local integration grid in each part of space is measured by comparing the numerical integral of the electron density to the corresponding analytically evaluated integral. The grid is further refined until a predefined integration accuracy criterion is met. Other parameters such as basis function extents are related to the integration accuracy criterion, resulting in a method where a single parameter is used to control the accuracy of the computed exchange-correlation matrix.

In the Ergo program, both space partitioned atomic grids as well as the HiCu scheme are implemented. In the benchmark calculations in section 4, we used the HiCu scheme since it has the advantage that the accuracy is controlled by a single parameter.

**3.2. Evaluation of the Sparse Exchange-Correlation Potential Matrix.** The evaluation of the exchange-correlation matrix as given by eq 5 formally follows the scheme shown in Algorithm 2. An efficient implementation of that algorithm must fulfill a few conditions:

(1) Matrix element magnitudes  $|V_{xc;\alpha;pq}|$  are estimated in advance so that memory for  $V_{xc;\alpha;pq} < \tau$  is not allocated and the sum contributing to that element is not computed. Here,  $\tau$  is a preselected threshold for matrix elements.

(2) Terms  $s_{\alpha;q}(r_i)$ ,  $\phi_p(r_i)$ , and  $\phi_q(r_i)$ , contributing to several matrix elements, are not unnecessarily recomputed.

**Algorithm 2.** Numerical Integration

```

1:   for each ( $p,q$ ) giving rise to a nonvanishing
       $V_{xc;\alpha;pq}$  do
2:      $V_{xc;\alpha;pq} := \sum_i w_i [s_{\alpha;p}(r_i) \phi_q(r_i) + \phi_p(r_i) s_{\alpha;q}(r_i)]$ 
3:   end for

```

Depending on the amount of available memory and other considerations, the operations in Algorithm 2 may be performed in a different order. If memory constraints were not present, we could perform the operations as shown in Algorithm 3. This simple algorithm has significant memory requirements. The sparse matrix  $B$  needs to be available during the entire integration process. Let us consider for example a system with 10 000 atoms, with 10 000 grid points per atom, and where on average 50 basis functions are nonvanishing at a grid point. In that example, the matrix  $B$  would require approximately 60 GB of memory if stored in the compressed sparse row format.<sup>52</sup> One way to reduce the memory demand is to process the grid points in batches. The Ergo implementation follows that approach.

**Algorithm 3.** Linearly Scaling Numerical Integration

```

1:   Compute a sparse matrix  $B$  with elements  $B_{ki} = \phi_k(r_i)$ 
      of basis functions  $\phi_k$  evaluated at grid points  $r_i$ .
2:   Compute  $\rho_\sigma(r_i)$ ,  $\sigma \in \{\alpha, \beta\}$  by contracting the sparse
      density matrix  $D_\sigma$  with sparse  $B$  on each side:  $\rho_\sigma(r_i) = \sum_{pq} B_{pi} D_{\sigma;pq} B_{qi}$ .
3:   Use  $\rho_\sigma(r_i)$  to compute  $v_\sigma(r_i)$ ,  $u_\sigma(r_i)$ ,  $t(r_i)$  and eventually
       $S_{\sigma;pi} = s_{\sigma;p}(r_i)$ , and store the result.
4:   Compute the exchange-correlation matrix by
      performing a matrix scaling and a sparse matrix-
      matrix multiplication as given by eq 11:  $V_{xc;\sigma;pq} = \sum_i w_i (B_{qi} S_{\sigma;pi} + B_{pi} S_{\sigma;qi})$ 

```

At the time of grid generation, grid points are collected in spatial cells. For each cell, we find the basis functions that overlap with that cell. This information is important with

respect to both accuracy and performance. A too cautious estimation may result in a dramatic increase in calculation time. On the other hand, a too sloppy criterion for determination of basis functions relevant for a given cell will inadvertently affect the calculation accuracy. The list of nonvanishing basis functions for a given cell allows us to predict, using eq 11, which exchange-correlation matrix elements may have nonzero values and to determine an exchange-correlation matrix sparsity pattern. This pattern in turn permits preallocation of the resulting exchange-correlation matrix  $V_{xc;pq}$  so that individual contributions computed later can be added quickly without a need to reallocate memory. The numerical integration is then performed one cell at a time. For each cell, we follow Algorithm 3, with the exception that the partial contributions computed according to this algorithm are accumulated for all cells.

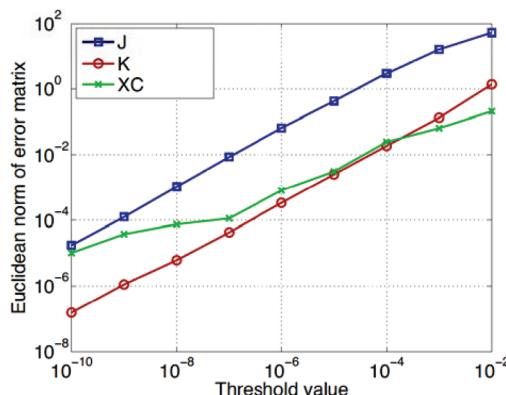
We store the sparse matrix pattern for the exchange-correlation calculation as a list of nonvanishing matrix element ranges for each column of the matrix. For any given column, we choose to number the atoms and associated basis functions following their spatial location, so that the number of ranges is small. This data structure has several properties important from the performance point of view. First, ranges can be rapidly extended as the sparse matrix pattern is built during grid generation. Second, a small number of ranges for any given column, together with an efficient bisection algorithm, makes it possible to efficiently find elements in the sparse matrix. To assess the efficiency of our sparse matrix structure, we have compared the performance to a previous version of the code using full dense matrix storage. The overhead of finding matrix elements in the sparse matrix structure does not exceed 10% of the total exchange-correlation integration time in pessimistic cases, and in many cases, the sparse code is faster by 10% to 20% than the corresponding version operating on full matrices due to increased computer memory locality and improved cache usage.

## 4. Benchmark Calculations

In this section, we present benchmark calculations for two kinds of molecular systems: glutamic acid–alanine (Glu–Ala) helices and water clusters. For each of them, we have performed KS-DFT calculations for varying system sizes using the BHandHLYP functional<sup>28</sup> with two different basis sets: 3-21G and 6-31G\*\*.

**4.1. Molecular Systems Used for Benchmarks.** The Glu–Ala helix systems were generated using the “build sequence” function in the Spartan program,<sup>53</sup> with the  $\alpha$  helix option selected. We refer to these systems as  $[\text{GluAla}]_n$ , where  $n$  is the number of repeating Glu–Ala units. Because generating very large helices using the Spartan program became cumbersome, systems larger than  $[\text{GluAla}]_{448}$  were instead generated from the smaller ones using an elongation procedure.

The water cluster geometries were generated from a large molecular dynamics simulation of bulk water at standard temperature and pressure by including all water molecules within spheres of varying radii. A water molecule was



**Figure 1.** Accuracy scans for BHandHLYP/3-21G calculations on a water cluster containing 471 water molecules, near SCF convergence. These scans are used together with inequality 13 to select threshold values for Coulomb matrix ( $J$ ), HF exchange matrix ( $K$ ), and exchange-correlation matrix ( $XC$ ) evaluations that correspond to a requested level of accuracy in the occupied subspace.

included if its oxygen atom was within the radius, thus making sure that only whole water molecules were included.

The Glu-Ala and water cluster systems used in the benchmarks are comparable to the ones used in ref 4, although the water cluster geometries were now generated from a different, larger molecular dynamics simulation. Thanks to improvements of our implementation as well as the availability of greater computer memory, we are now able to carry out benchmark calculations for significantly larger systems.

To facilitate comparison and verification of results, all molecular geometries used in this paper are available for download at [www.ergoscf.org](http://www.ergoscf.org).

**4.2. Selection of Threshold Values.** The key quantity of interest in electronic structure calculations is the occupied subspace. Computational approximations result in perturbations of this subspace. Errors can be measured by the largest canonical angle  $\theta_1$  between exact and perturbed subspaces. This angle is related to the Euclidean norm of the error matrix  $E$  and the gap  $\xi$  between eigenvalues corresponding to occupied and unoccupied orbitals respectively:<sup>54</sup>

$$\sin \theta_1 \leq \frac{\|E\|_2}{\xi - \|E\|_2} \quad (13)$$

Note that in the case of a Kohn-Sham matrix,  $\xi$  is equal to the HOMO-LUMO gap, and in case of a density matrix,  $\xi = 1$ . For the benchmark calculations, we selected threshold values such that the erroneous rotation of the occupied subspace in each of the  $F \rightarrow D$  and  $D \rightarrow F$  steps as measured by  $\sin \theta_1$  would be below  $1 \times 10^{-2}$ .

For the  $F \rightarrow D$  step, our implementation of density matrix purification allows us to specify the desired accuracy directly in terms of  $\sin \theta_1$ .<sup>18</sup> For the  $D \rightarrow F$  step, we use the accuracy scan information in Figure 1 together with inequality 13 to choose threshold values that correspond to a given error in the occupied subspace. The needed information about the HOMO-LUMO gap is obtained as a byproduct of density matrix purification.<sup>18,55</sup> The resulting threshold values,

**Table 1.** Threshold Values Selected to Give an Accuracy in the Occupied Subspace of  $\sin \theta_1 \leq 1 \times 10^{-2}$  in Each of the  $F \rightarrow D$  and  $D \rightarrow F$  Steps, Where the  $D \rightarrow F$  Step Consists of  $J$ ,  $K$ , and  $V_{xc}$  Matrix Evaluations<sup>a</sup>

calculation	threshold value
$F \rightarrow D$ step	$\tau_D = 1 \times 10^{-2}$
Coulomb matrix $J$	$\tau_J = 5 \times 10^{-9}$
HF exchange matrix $K$	$\tau_K = 2 \times 10^{-6}$
exchange-correlation matrix $V_{xc}$	$\tau_{xc} = 5 \times 10^{-7}$

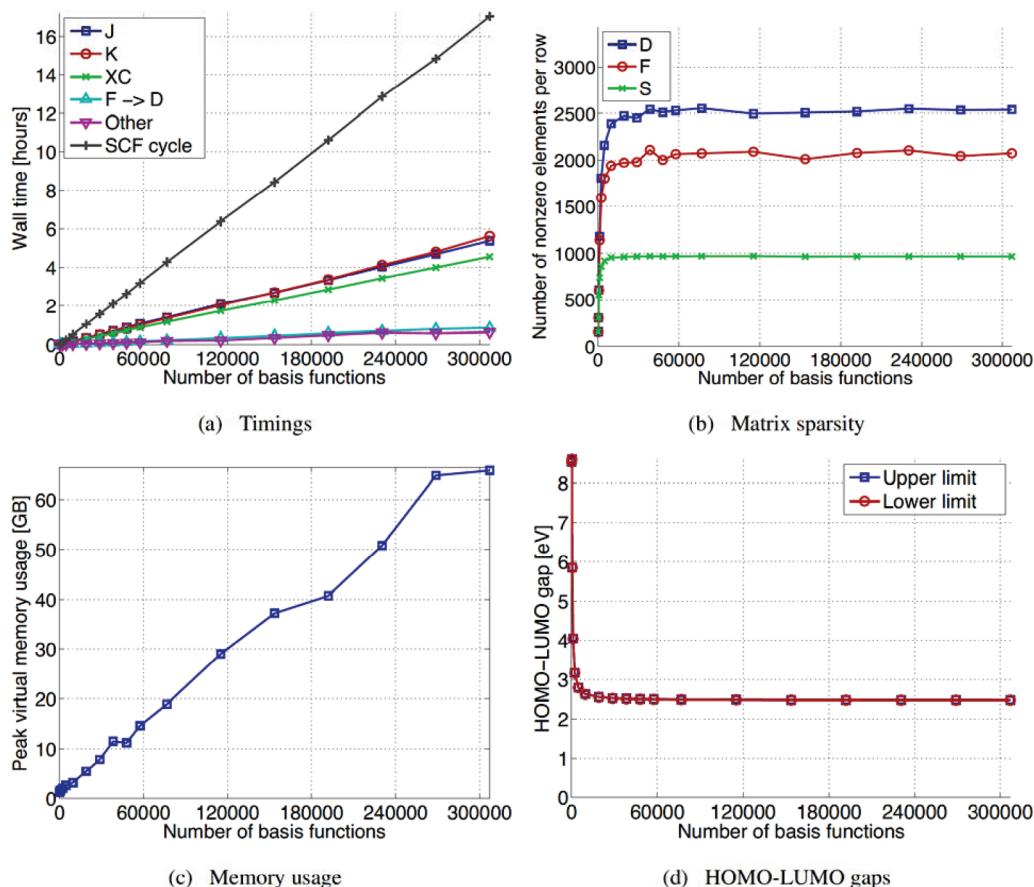
<sup>a</sup> The values were determined using the accuracy scans in Figure 1 together with eq 13. The HOMO-LUMO gap in the BHandHLYP/3-21G calculation was 0.23 au.

chosen so that  $\sin \theta_1 \leq 1 \times 10^{-2}$  in each of the  $F \rightarrow D$  and  $D \rightarrow F$  steps for BHandHLYP water cluster calculations, are shown in Table 1. Note that the threshold values  $\tau_J$  and  $\tau_K$ , for Coulomb and HF exchange matrix construction, respectively, differ by more than 2 orders of magnitude. This is in spite of the fact that these threshold values are used in a similar way, namely, that contributions smaller than the specified value are neglected. Thus, there is no universal relationship between neglect threshold and error matrix norm.

In principle, separate accuracy scans should be performed also for 6-31G\*\* and for the Glu-Ala systems, giving different sets of threshold values for each case. Whereas the different gap value for Glu-Ala is automatically taken into account by the program in the  $F \rightarrow D$  step, this is not the case in the construction of the Kohn-Sham matrix. Ideally, the different gap value for Glu-Ala should have been taken into account in the calculations of  $J$ ,  $K$ , and  $V_{xc}$  as well. However, for simplicity, we have used the set of threshold values in Table 1 for all benchmark calculations reported in this work, with one exception: the parameter  $\tau_D$  determining the accuracy of the  $F \rightarrow D$  step was set to  $1 \times 10^{-3}$  in the Glu-Ala calculations, in order to get more accurate information about HOMO and LUMO eigenvalues. As will be seen below, this parameter change for Glu-Ala did not significantly affect the overall computational time since the Glu-Ala calculations were strongly dominated by the  $D \rightarrow F$  step.

As can be seen in Figure 1, the errors in  $J$  and  $K$  decrease in a very predictable manner as the integral threshold values are decreased. It is therefore possible to implement a program that automatically selects appropriate threshold values by extrapolation after an assessment of the slopes of the lines.<sup>56</sup> We believe that such an automated procedure in principle should be possible also for the exchange-correlation part. However, with our current implementation, the error in the exchange-correlation part decreases in a less predictable manner, see Figure 1, which makes it difficult to automate the procedure of selecting a threshold value for the exchange-correlation matrix evaluation.

**4.3. Results.** Results of the benchmark calculations are shown in Figures 2, 3, 4, and 5. Each figure includes timings, matrix sparsity, memory usage, and the HOMO-LUMO gap plotted against system size. The calculations were performed using the Ergo program,<sup>20</sup> compiled with the Intel C++ compiler (ICC), version 10.1, and linked to the GotoBLAS2 linear algebra library, version 1.11p1.<sup>57,58</sup> Each calculation was run on a HP SL170h G6 compute server with two Quad-



**Figure 2.** Timings, memory usage, matrix sparsity, and computed HOMO–LUMO gaps for BHandHLYP/3-21G calculations on Glu–Ala helix systems of varying size. The right-most points in the graphs are for [GluAla]<sub>2048</sub>, with 53 250 atoms and 307 204 basis functions. The upper and lower limits of the HOMO–LUMO gaps are indistinguishable in the figure.

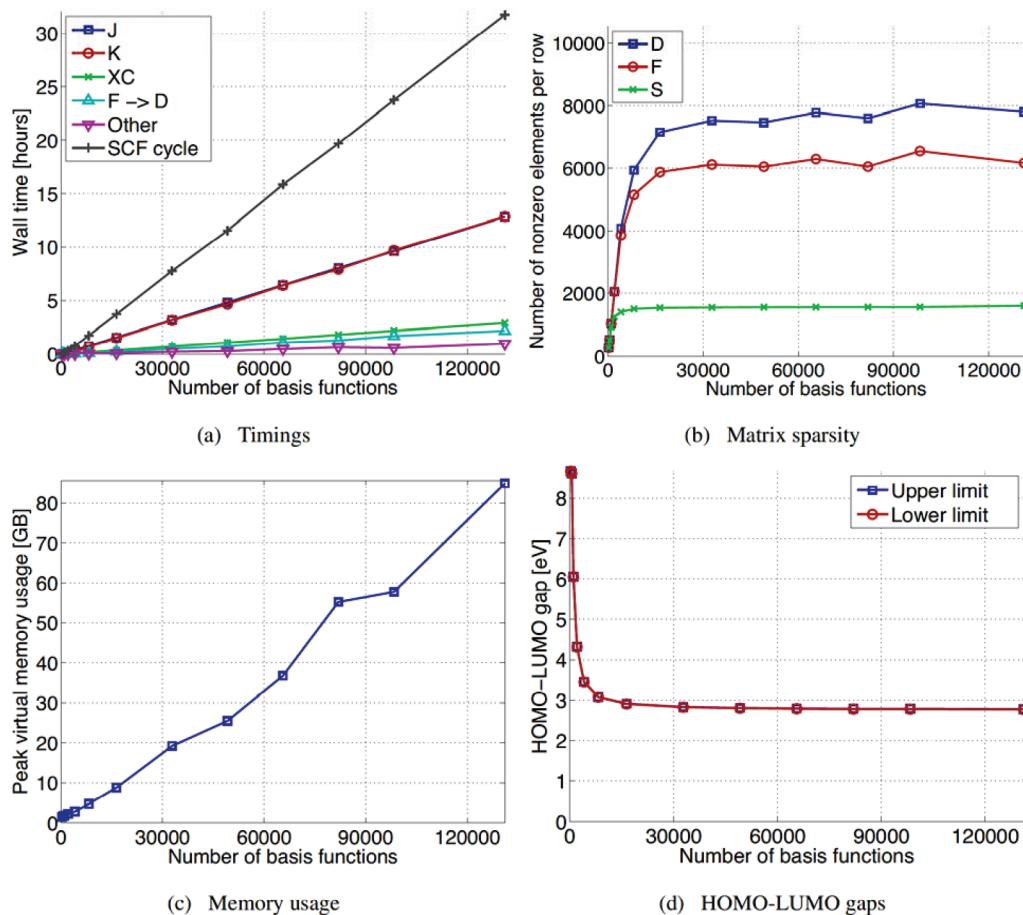
core Intel Xeon 5520 (Nehalem 2.26 GHz, 8 MB cache) processors and 72 GB of shared memory running the Scientific Linux 5.4 operating system. Block-sparse matrix operations were performed using a version of the Hierarchic Matrix Library<sup>42,59</sup> parallelized for shared memory using OpenMP, with a uniform block size of 32 at the lowest level. Other time-critical parts of the code are parallelized using POSIX threads.

To generate starting guess densities, we first performed preliminary calculations at the HF/STO-2G level of theory. The resulting densities were used as starting guesses for calculations at the HF/3-21G level. Finally, the converged HF/3-21G densities were used as starting guesses for the BHandHLYP calculations for both the 3-21G and 6-31G\*\* basis sets. The HF/3-21G guesses were good enough for the BHandHLYP calculations to converge within 4–6 self-consistent field iterations. The calculations were considered converged as soon as the largest magnitude element of  $FDS - SDF$  was smaller than  $1 \times 10^{-3}$ .

The plotted timings do not include the initialization work that is needed before the first SCF cycle, i.e., steps 1–5 in Algorithm 1. Those steps contributed only to a small part of the total calculation time. The whole initialization work, including computation of the overlap and one-electron Hamiltonian matrices, inverse Cholesky decomposition, and starting guess density projection parts, in no case took more than 12% of the total calculation time.

The grid for numerical integration in the KS-DFT exchange-correlation matrix evaluation was created using the HiCu method<sup>13</sup> with threshold value  $\tau_{xc} = 5 \times 10^{-7}$ , which gave on average around 9100 grid points per atom for the Glu–Ala calculations and around 7100 grid points per atom for the water cluster calculations. The choice of basis set did not significantly affect the number of grid points. Even though roughly the same number of grid points were generated for both basis sets, the grid generation for 6-31G\*\* required about 4–5 times as long time as for 3-21G. This is because the description of the electron density used during the HiCu grid generation<sup>13</sup> becomes more expensive when using a larger basis set. The grid was generated only once, in the first SCF cycle. In subsequent cycles, the same grid was reused. Therefore, the exchange-correlation matrix evaluation in the first SCF cycle was computationally more expensive by roughly a factor of 2. Linear scaling was observed for the grid generation, although this extra time is not seen in the figures as the plotted timings are for the third SCF cycle.

The results of the Glu–Ala helix benchmark calculations, shown in Figures 2 and 3 for basis sets 3-21G and 6-31G\*\*, respectively, indicate nearly perfect linear scaling. The timings are strongly dominated by the  $D \rightarrow F$  step, consisting of the  $J$ ,  $K$ , and XC parts, while the  $F \rightarrow D$  step requires less than 10% of the total SCF cycle time. For the larger helices, the number of nonzero elements in the density matrix



**Figure 3.** Timings, memory usage, matrix sparsity, and computed HOMO–LUMO gaps for BHandHLYP/6-31G\*\* calculations on Glu–Ala helix systems of varying size. The right-most points in the graphs are for [GluAla]<sub>512</sub>, with 13 314 atoms and 131 082 basis functions. The timings for the Coulomb (*J*) and HF exchange (*K*) parts in panel a are almost identical.

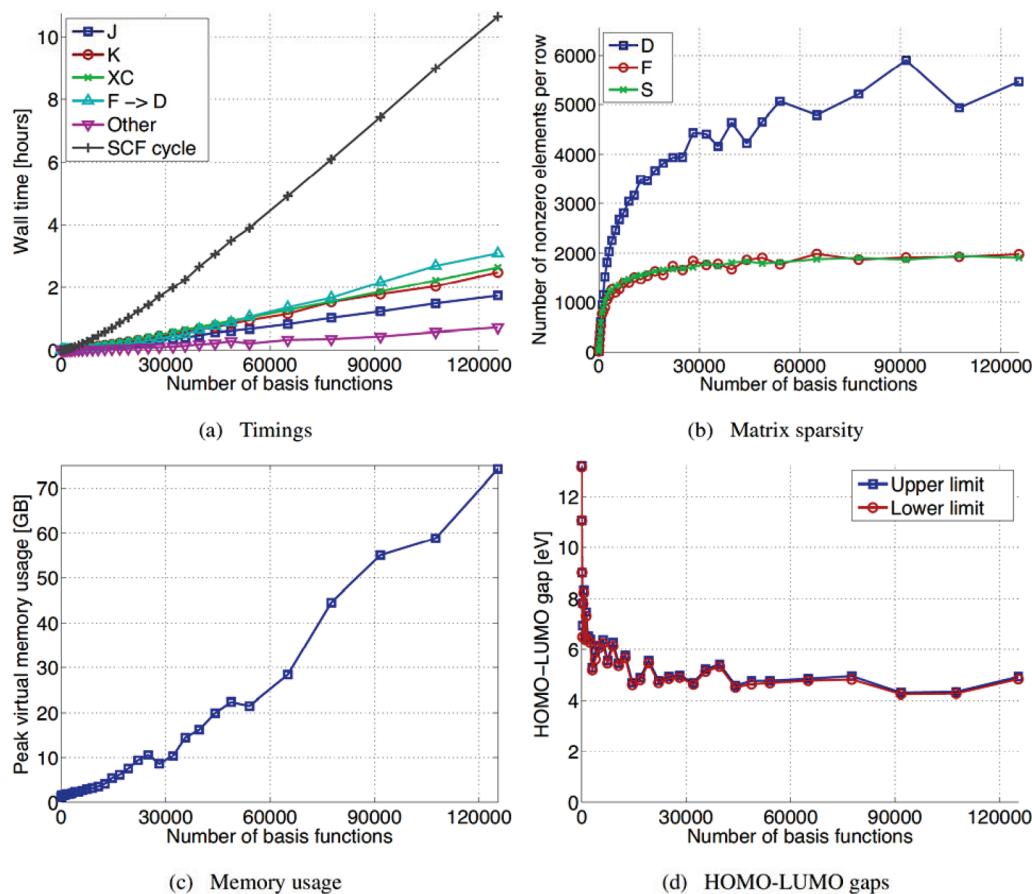
stabilizes at around 2500 and 8000 elements per row for basis sets 3-21G and 6-31G\*\*, respectively. For the 3-21G basis set, the largest helix we could handle was [GluAla]<sub>2048</sub>, C<sub>16384</sub>N<sub>4096</sub>O<sub>8192</sub>H<sub>24578</sub>, corresponding to 307 204 basis functions. For the 6-31G\*\* basis set, the largest helix we could handle was [GluAla]<sub>512</sub>, C<sub>4096</sub>N<sub>1024</sub>O<sub>2048</sub>H<sub>6146</sub>, corresponding to 131 082 basis functions.

In the water cluster benchmark calculations, shown in Figures 4 and 5 for basis sets 3-21G and 6-31G\*\*, respectively, linear scaling behavior is approached for the larger systems. Perfect linear scaling is expected when reaching system sizes where the number of nonzero elements per row in the density matrix no longer increases. We note that for the larger water cluster calculations the *F*→*D* step takes a considerable part of the total SCF cycle time, and it is clearly the last part of the calculation to enter the linear scaling regime. This is in sharp contrast to the one-dimensional Glu–Ala case where the *F*→*D* step takes only a small fraction of the total SCF cycle time. For the larger water clusters, the number of nonzero elements in the density matrix approaches 6000 and 13 000 elements per row for basis sets 3-21G and 6-31G\*\*, respectively. For the 3-21G basis set, the largest water cluster we could handle contained 9644 water molecules, corresponding to 125 372 basis functions. For the 6-31G\*\* basis set, the largest water cluster we could handle contained 3050 water molecules, corresponding to 73 200 basis functions.

The memory usage plotted in panel c of Figures 2, 3, 4, and 5 is the peak virtual memory usage as reported by the operating system. In some of the largest calculations, the virtual memory usage was slightly above the physical memory limit of 72 GB, so that some swapping to disk by the operating system must have occurred. However, this incurred no noticeable overhead.

For both water clusters and Glu–Ala helix systems, the increase in computational cost when changing basis set from 3-21G to 6-31G\*\* was around a factor of 5–7 in computational time and around a factor of 3–4 in memory requirement.

HOMO and LUMO eigenvalues were computed by applying the Lanczos method in intermediate purification iterations as described in ref 18. The resulting HOMO–LUMO gaps are plotted in panel d of Figures 2, 3, 4, and 5. For the larger GluAla systems, the computed HOMO–LUMO gaps are 2.5 and 2.8 eV for 3-21G and 6-31G\*\*, respectively. For the larger water cluster systems, the computed HOMO–LUMO gaps are 4.2–4.9 eV and 5.6–6.4 eV for 3-21G and 6-31G\*\*, respectively. It should be noted that the HOMO–LUMO gaps resulting from this kind of calculation are strongly dependent on the amount of HF exchange in the KS-DFT functional. We chose to use the BHandHLYP functional for these benchmarks in order to get sufficiently large HOMO–LUMO gaps to allow efficient calculations. In fact, for the larger Glu–Ala and



**Figure 4.** Timings, memory usage, matrix sparsity, and computed HOMO–LUMO gaps for BHandHLYP/3-21G calculations on water cluster systems of varying size. The right-most points in the graphs are for a water cluster containing 9644 water molecules, corresponding to 125 372 basis functions.

water cluster systems as well as for large protein molecules in general, our experience is that some fraction of HF exchange in the functional is necessary in order to get a nonvanishing HOMO–LUMO gap. We have been unable reach convergence in attempted calculations using pure functionals such as LDA, BLYP, and PBE for all but the smallest systems.

In the matrix sparsity plots, the curves are somewhat jagged, particularly in Figure 4b. This is an effect of the stepping procedure in the applied truncation scheme which ensures that the norm of the error matrix is below a requested value, see refs 39 and 60. An adjustment of a stepping parameter in the truncation scheme implementation would make the curves more smooth. The effect is more pronounced in the water cluster calculations because, compared to the Glu–Ala case, more matrix elements are removed in each truncation.

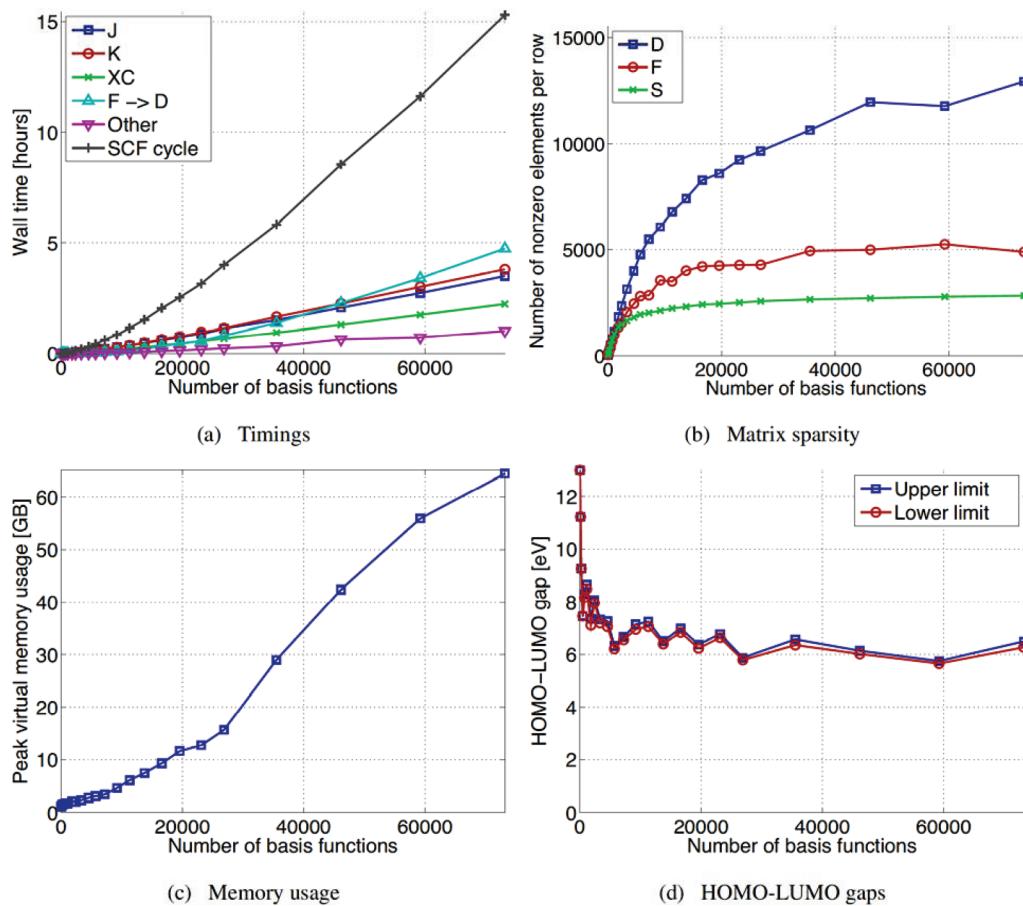
By comparison to higher accuracy results, we have found that, for both water clusters and Glu–Ala, the errors in total energies in the larger calculations were below  $1 \times 10^{-5}$  Hartree/atom for the 3-21G basis set, and below  $3 \times 10^{-5}$  Hartree/atom for the 6-31G\*\* basis set.

## 5. Concluding Remarks

It should be noted that since we employed a hybrid KS-DFT functional, the calculations presented here include all

components needed for HF calculations. Thus, although the presented benchmarks do not include HF calculations, we are able to perform HF calculations for these systems as well. Compared to a BHandHLYP calculation with the same basis set, a HF calculation is typically somewhat faster since the exchange-correlation matrix evaluation is not needed. Also, HF calculations are generally easier to carry out because HF gives a larger HOMO–LUMO gap than DFT.

An important aspect of linear scaling calculations is the selection of threshold values. In the calculations reported in this work, threshold values were chosen to give roughly the same accuracy for the different parts. Threshold values for the different parts of the  $D \rightarrow F$  step were chosen using information from previous accuracy scans. Threshold values in the  $F \rightarrow D$  step were automatically selected by the program to achieve the requested accuracy. Ideally, threshold values for the  $D \rightarrow F$  step should also be chosen automatically, for example using extrapolation.<sup>56</sup> In any case, balancing the accuracy can considerably improve the overall performance. One example of this is the selection of integral screening threshold values for Coulomb and HF exchange matrix evaluation. In our previous study of linear scaling HF calculations,<sup>4</sup> using ad hoc selected threshold values, the HF exchange matrix evaluation was about twice as expensive as the Coulomb part. Now, having adapted the threshold



**Figure 5.** Timings, memory usage, matrix sparsity, and computed HOMO–LUMO gaps for BHandHLYP/6-31G\*\* calculations on water cluster systems of varying size. The right-most points in the graphs are for a water cluster containing 3050 water molecules, corresponding to 73200 basis functions.

values to get balanced accuracy, we find that the Coulomb and HF exchange parts require a similar amount of time.

The benchmark systems used in the present paper are approximately four times larger than the systems of our previous linear scaling HF study,<sup>4</sup> while employing the same basis sets. This increase has been made possible not only because of greater available computer memory but also to a large extent thanks to improvements in the code. The memory usage has been significantly reduced by changing to more economic data structures and avoiding unnecessary matrix copies. Also, we have had to improve the scaling of other previously negligible parts of the code, including the computation of the overlap matrix and preparatory steps in HF exchange and Coulomb matrix construction. The performance of the used density matrix purification method<sup>18</sup> has been considerably improved by use of a novel scheme for the removal of small matrix elements.<sup>39</sup> Other technical issues arising for larger systems include changing data types of several quantities to avoid integer overflow.

One remaining issue is the scaling of the inverse Cholesky algorithm. Whereas the inverse Cholesky operation scales linearly for the Glu–Ala calculations, the scaling for the water cluster systems appears less favorable. We did not pay much attention to this issue here since the inverse Cholesky operation even for the largest water cluster calculations requires less than 5% of the total calculation time, but for larger systems, this issue is likely to become important. An

alternative to the inverse Cholesky algorithm is recursive inverse factorization,<sup>45</sup> a method based on repeated sparse matrix multiplication.

Another important aspect is parallelization. The benchmark calculations presented in this work were performed on a single eight-core computer, using threading to exploit the eight cores. An overall speedup of around 6.5 was achieved compared to the single-core performance. The reason why the perfect speedup of eight for an eight-core machine was not reached is mainly that some parts of the code were not threaded. Clearly, this should be remedied in order to make the best use of computers with larger numbers of cores. Also, considering that many high performance computing resources are distributed memory systems, distributed memory parallelization is desirable. In future work, we aim to use a task-based approach as a way to achieve scalable parallelization of dynamic hierachic algorithms such as the sparse matrix operations and multipole methods used in this kind of calculation.

Finally, we note a change that was needed in our density matrix purification algorithm when handling large systems. Previously, the density matrix purification scheme in the Ergo program used Theorem 3 of ref 18 to strictly ensure the correct occupation number in cases when information about the HOMO and LUMO eigenvalues is not yet available. This strict occupation number requirement has since been removed. Instead, the correct occupation number is assumed

at the end of the purification process. This change, which was necessary to handle large systems, has not caused any problems in practice. The strict occupation number requirement is in conflict with linear scaling methods where the error per electron is fixed as the system size increases. When increasing the system size, one reaches a point where the occupation number cannot be determined from the density matrix within the accuracy of one electron.

**Acknowledgment.** The authors thank Daniel Spångberg at Uppsala University Department of Materials Chemistry for performing the molecular dynamics simulation from which the water cluster geometries could be extracted. Support from the Swedish Research Council under Grant No. 623-2009-803 is gratefully acknowledged. The computations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at Uppsala Multidisciplinary Center for Advanced Computational Sciences (UPPMAX) under Project p2010021.

## References

- (1) Goedecker, S. *Rev. Mod. Phys.* **1999**, *71*, 1085–1123.
- (2) Bowler, D. R.; Miyazaki, T.; Gillan, M. J. *J. Phys.: Condens. Matter* **2002**, *14*, 2781–2798.
- (3) Wu, S. Y.; Jayanthi, C. S. *Phys. Rep.* **2002**, *358*, 1–74.
- (4) Rudberg, E.; Rubensson, E. H.; Sałek, P. *J. Chem. Phys.* **2008**, *128*, 184106.
- (5) Hine, N.; Haynes, P.; Mostofi, A.; Skylaris, C.-K.; Payne, M. *Comput. Phys. Commun.* **2009**, *180*, 1041.
- (6) Bowler, D. R.; Miyazaki, T. *J. Phys.: Condens. Matter* **2010**, *22*, 074207.
- (7) White, C. A.; Head-Gordon, M. *J. Chem. Phys.* **1994**, *101*, 6593.
- (8) Challacombe, M.; Schwegler, E.; Almlöf, J. *J. Chem. Phys.* **1995**, *104*, 4685–4698.
- (9) Rudberg, E.; Sałek, P. *J. Chem. Phys.* **2006**, *125*, 084106.
- (10) Schwegler, E.; Challacombe, M. *J. Chem. Phys.* **1996**, *105*, 2726.
- (11) Ochsenfeld, C.; White, C. A.; Head-Gordon, M. *J. Chem. Phys.* **1998**, *109*, 1663–1669.
- (12) Stratmann, R. E.; Scuseria, G. E.; Frisch, M. J. *Chem. Phys. Lett.* **1996**, *257*, 213–223.
- (13) Challacombe, M. *J. Chem. Phys.* **2000**, *113*, 10037.
- (14) Li, X.-P.; Nunes, R. W.; Vanderbilt, D. *Phys. Rev. B* **1993**, *47*, 10891–10894.
- (15) Goedecker, S.; Colombo, L. *Phys. Rev. Lett.* **1994**, *73*, 122–125.
- (16) Palser, A. H. R.; Manolopoulos, D. E. *Phys. Rev. B* **1998**, *58*, 12704–12711.
- (17) Niklasson, A. M. N. *Phys. Rev. B* **2002**, *66*, 155115.
- (18) Rubensson, E. H.; Rudberg, E.; Sałek, P. *J. Chem. Phys.* **2008**, *128*, 074106.
- (19) Umeda, H.; Inadomi, Y.; Watanabe, T.; Yagi, T.; Ishimoto, T.; Ikegami, T.; Tadano, H.; Sakurai, T.; Nagashima, U. *J. Comput. Chem.* **2010**, *31*, 2381.
- (20) Rudberg, E.; Rubensson, E. H.; Sałek, P. *Ergo*, version 2.1. <http://www.ergoscf.org> (accessed Dec. 9, 2010).
- (21) Helgaker, T.; Jørgensen, P.; Olsen, J. *Molecular electronic-structure theory*; Wiley: Chichester, U. K., 2000.
- (22) Thøgersen, L. Optimization of densities in Hartree-Fock and density-functional theory, atomic orbital based response theory, and benchmarking for radicals. Ph.D. thesis, Department of Chemistry, University of Aarhus, Aarhus, Denmark, 2005.
- (23) Kudin, K. N.; Scuseria, G. E. *Math. Model. Num. Anal.* **2007**, *41*, 281–296.
- (24) Zerner, M. C.; Hehenberger, M. *Chem. Phys. Lett.* **1979**, *62*, 550–554.
- (25) Cancès, E.; Le Bris, C. *Int. J. Quantum Chem.* **2000**, *79*, 82–90.
- (26) Pulay, P. *Chem. Phys. Lett.* **1980**, *73*, 393.
- (27) Pulay, P. *J. Comput. Chem.* **1982**, *3*, 556.
- (28) Becke, A. D. *J. Chem. Phys.* **1993**, *98*, 1372–1377.
- (29) Panas, I.; Almlöf, J.; Feyereisen, M. W. *Int. J. Quantum Chem.* **1991**, *40*, 797–807.
- (30) Panas, I.; Almlöf, J. *Int. J. Quantum Chem.* **1992**, *42*, 1073–1089.
- (31) White, C. A.; Johnson, B. G.; Gill, P. M. W.; Head-Gordon, M. *Chem. Phys. Lett.* **1994**, *230*, 8–16.
- (32) Schwegler, E.; Challacombe, M.; Head-Gordon, M. *J. Chem. Phys.* **1998**, *109*, 8764–8769.
- (33) Burant, J. C.; Scuseria, G. E. *J. Chem. Phys.* **1996**, *105*, 8969.
- (34) Schwegler, E.; Challacombe, M.; Head-Gordon, M. *J. Chem. Phys.* **1997**, *106*, 9708.
- (35) Schwegler, E.; Challacombe, M. *J. Chem. Phys.* **1999**, *111*, 6223.
- (36) Ochsenfeld, C. *Chem. Phys. Lett.* **2000**, *327*, 216.
- (37) Lambrecht, D. S.; Ochsenfeld, C. *J. Chem. Phys.* **2005**, *123*, 184101.
- (38) Aquilante, F.; Pedersen, T. B.; Lindh, R. *J. Chem. Phys.* **2007**, *126*, 194106.
- (39) Rubensson, E. H.; Rudberg, E. *J. Comput. Chem.* **2010**, in press.
- (40) Millam, J. M.; Scuseria, G. E. *J. Chem. Phys.* **1997**, *106*, 5569–5577.
- (41) Benzi, M.; Kouhia, R.; Tuma, M. *Comput. Methods Appl. Mech. Eng.* **2001**, *190*, 6533–6554.
- (42) Rubensson, E. H.; Rudberg, E.; Sałek, P. *J. Comput. Chem.* **2007**, *28*, 2531–2537.
- (43) Niklasson, A. M. N. *Phys. Rev. B* **2004**, *70*, 193102.
- (44) Jansík, B.; Høst, S.; Jørgensen, P.; Olsen, J. *J. Chem. Phys.* **2007**, *126*, 124104.
- (45) Rubensson, E. H.; Bock, N.; Holmström, E.; Niklasson, A. M. N. *J. Chem. Phys.* **2008**, *128*, 104105.
- (46) Kohn, W.; Sham, L. *Phys. Rev.* **1965**, *140*, A1133–A1138.
- (47) Treutler, O.; Ahlrichs, R. *J. Chem. Phys.* **1995**, *102*, 346–354.
- (48) Lebedev, V. I. *Zh. Vychisl. Mat. Mat. Fiz.* **1975**, *45*, 48–54.
- (49) Murray, C. W.; Handy, N. C.; Laming, G. J. *Mol. Phys.* **1993**, *78*, 997.
- (50) Lindh, R.; Malmqvist, P.-Å.; Gagliardi, L. *Theor. Chem. Acc.* **2001**, *106*, 178–187.

- (51) Becke, A. D. *J. Chem. Phys.* **1988**, *88*, 2547.
- (52) Pissanetsky, S. *Sparse Matrix Technology*; Academic Press: New York, 1984.
- (53) *Spartan '02*; Wavefunction, Inc.: Irvine, CA, 2002.
- (54) Rubensson, E. H.; Rudberg, E.; Sałek, P. *J. Math. Phys.* **2008**, *49*, 032103.
- (55) Rubensson, E. H.; Zahedi, S. *J. Chem. Phys.* **2008**, *128*, 176101.
- (56) Rudberg, E.; Rubensson, E. H.; Sałek, P. *J. Chem. Theory Comput.* **2009**, *5*, 80–85.
- (57) Goto, K.; van de Geijn, R. A. *ACM Trans. Math. Software* **2008**, *34*, 12.
- (58) GotoBLAS2. <http://www.tacc.utexas.edu/tacc-projects/gotoblas2> (accessed Jan 21, 2010).
- (59) Rubensson, E. H.; Rudberg, E.; Sałek, P. *Proc. PARA '06, Springer LNCS* **2007**, *4699*, 90–99.
- (60) Rubensson, E. H.; Rudberg, E.; Sałek, P. *J. Comput. Chem.* **2009**, *30*, 974–977.

CT100611Z