

Parallel algorithm for the computation of the Hartree-Fock exchange matrix: Gas phase and periodic parallel ONX

Valéry Weber^{a)}*Department of Chemistry, University of Fribourg, 1700 Fribourg, Switzerland*

Matt Challacombe

Los Alamos National Laboratory, Theoretical Division, Los Alamos, 87545 New Mexico

(Received 16 February 2006; accepted 19 June 2006; published online 13 September 2006)

In this paper we present an efficient parallelization of the ONX algorithm for linear computation of the Hartree-Fock exchange matrix [J. Chem. Phys. **106**, 9708 (1997)]. The method used is based on the equal time (ET) partitioning recently introduced [J. Chem. Phys. **118**, 9128 (2003)] and [J. Chem. Phys. **121**, 6608 (2004)]. ET exploits the slow variation of the density matrix between self-consistent-field iterations to achieve load balance. The method is presented and some benchmark calculations are discussed for gas phase and periodic systems with up to 128 processors. The current parallel ONX code is able to deliver up to 77% overall efficiency for a cluster of 50 water molecules on 128 processors (2.56 processors per heavy atom) and up to 87% for a box of 64 water molecules (two processors per heavy atom) with periodic boundary conditions. © 2006 American Institute of Physics. [DOI: [10.1063/1.2222359](https://doi.org/10.1063/1.2222359)]

I. INTRODUCTION

Density functional theory (DFT) and its variant, the hybrid Hartree-Fock/density functional theory (HF/DFT) are accurate and computationally attractive. Together with linear scaling methods, these advances provide an important tool for large applications of both HF and HF/DFT models to areas such as biochemistry, material science, catalysis, and others.

The most time consuming part of a self-consistent-field (SCF) cycle remains formation of the Fock matrix, even with the appearance of fast linear scaling algorithms that overcome the bottlenecks encountered in conventional methods, including computation of the Hartree-Fock exchange matrix,^{1–6} the Coulomb matrix^{7–14} the exchange-correlation matrix,^{14–18} and iterative alternatives to eigensolution of the SCF equations.^{19–25}

It is well known that linear scaling methods become more advantageous than conventional algorithms only if the system size is bigger than few hundred atoms. This disadvantageous property of linear scaling methods, for small systems, can be overcome by parallelization. Parallel Fock builds (i.e., construction of the Coulomb and exact exchange matrices) are available in most quantum chemistry programs such as GAMESS,²⁶ JAGUAR,²⁷ and many others. The upper limit of a single processor speed is now reached, and massively parallel and distributed architectures such as workstation clusters or supercomputers will be more and more used in the near future for large-scale molecular calculations.

During the past decade conventional parallel algorithms (i.e., algorithms with an $\mathcal{O}(N^4)$ computation of the electron repulsion integrals) for the formation of the Fock matrix have been developed.^{28–34} Most of the current parallel meth-

ods are based on an integral driven strategy to compute the Fock matrix. On one side, the integral driven approach allows an efficient and simple way to use the full permutational symmetry of the two electron integrals, reducing considerably the computational time. On the other side, it requires the contraction of an integral quartet with up to six different density matrix elements. While this is certainly not a problem during a serial Fock build, it becomes much more complicated to distribute the density and/or Fock matrices in parallel.

To achieve load balance as well as data parallelism, these conventional algorithms use complicated schemes for the distribution of the density and/or Fock matrices through the processors during computation. In general, computation time scales as $\mathcal{O}(N^4)$, thus dominating the $\mathcal{O}(N^3)$ inter-processor communication of the density and/or Fock matrices. For example, Foster's algorithm achieves an efficiency, for the Fock matrix construction, up to 97% for $\text{Si}_{16}\text{O}_{25}\text{H}_{14}$ with the 6-31G* on 512 processors.³¹

In this work we present an efficient and simple parallelization of the order N exchange (ONX) algorithm for the $\mathcal{O}(N)$ computation of the HF exchange matrix. The ONX algorithm is fully equivalent to direct SCF (Ref. 35) and uses an ordered pair list of basis set functions to efficiently skip out the innermost loops, avoiding a $\mathcal{O}(N^2)$ logical test. The method is based on the equal time (ET) partitioning and can treat gas phase and periodic systems without increasing the complexity of the code. The equal time partitioning exploits the slow variation of the density matrix between self-consistent-field iterations to achieve load balance. The $\mathcal{O}(N)$ exchange-correlation and Coulomb matrices have been efficiently parallelized through the concept of equal time partition.^{36,37}

It should be pointed out that parallel $\mathcal{O}(N)$ computation of the exact exchange matrix with ONX is highly irregular

^{a)}Electronic mail: valeryw@lanl.gov

relative to conventional parallel $\mathcal{O}(N^4)$ computation of the electron repulsion integrals (ERIs). The ONX algorithm requires an $\mathcal{O}(N)$ communication of the density and Fock matrices, which also contrasts with conventional methods. Our parallel ONX uses a partially distributed-data exchange build. While each processor is assigned a part of the density matrix, the corresponding exchange submatrices are not redistributed during their constructions. This strategy avoids an expensive dynamic redistribution of the exchange submatrices. Also the current ONX is based on the simple “index-driven” algorithm although an “integral-driven” ONX algorithm exists.⁵ The parallelization of the later requires expensive communications of data as do conventional algorithms.³³

The paper is organized as follows: In Sec. II the method will be outlined. In Sec. III we describe a computational implementation of parallel ONX. In Sec. IV we discuss benchmark calculations of water clusters, endothelin, and cells composed of 64 (H₂O) molecules and 32 KCNS with periodic boundary conditions (PBC). In Sec. V we present our conclusions.

II. PARALLELIZATION OF THE ONX ALGORITHM

The ONX algorithm for $\mathcal{O}(N)$ calculation of the exact exchange matrix has been extensively described for gas phase¹⁻⁵ and the periodic Γ -point approximation.⁶ The Γ -point approximation limits \mathbf{k} -space sampling to just the central cell at $\mathbf{k}=0$. Translational invariance is imposed by introducing the minimum image convention (MIC) into the contraction phase of periodically summed two-electron integrals; this ensures that interactions are always calculated between nearest images. In computation of the HF exchange matrix, the MIC Γ -point approximation is given by

$$K_{ab}(D) = -\frac{1}{2} \sum_{\mathbf{m}\mathbf{n}} \sum_{cd} D_{cd} (ac^{\mathbf{m}}|bd^{\mathbf{n}})_{\text{mic}}, \quad (1)$$

where the indices c and d run over basis functions, \mathbf{m} and \mathbf{n} over the Bravais lattice vectors, and D_{cd} is an element of the density matrix. The evaluation of the two electron integrals $(ac^{\mathbf{m}}|bd^{\mathbf{n}})_{\text{mic}}$ is described elsewhere.⁶ In the case of gas phase calculations, the summations over the lattices \mathbf{m} and \mathbf{n} are just removed from Eq. (1).

Our algorithm uses a density driven, completely distributed exchange matrix build. The density matrix, initially stored on the root processor, is partitioned following the scheme proposed in Sec. III. Then subblocks are sent to their respective processors.

The original ONX algorithm has been restructured to include PBC (Ref. 6) and to improve parallel efficiency. In the first step, each processor $i\text{PrC}$ receives a dedicated nonoverlapping subdensity matrix $D(i\text{PrC})$ from root. During the second step, two distribution lists $\{ac^{\mathbf{m}}\}$ and $\{bd^{\mathbf{n}}\}$ are built, where c and d run over the rows and the columns of $D(i\text{PrC})$, respectively. The procedure for structuring the list of distributions for the $\{ac^{\mathbf{m}}\}$ pair is outlined in Fig. 1. This list build

```

DO C ∈ COL(D(iPrC))
DO A = 1, NAtoms
  DO m = 1, NCell
    DO c ∈ C
      DO a ∈ A
        COMPUTE |(acm|acm)mic|1/2
        IF (|(acm|acm)mic|1/2 < Thresh) CYCLE
        ADD acm to {acm}(A, C)
      ENDDO
    ENDDO
  ENDDO
  SORT {acm}(A, C) by decreasing |(acm|acm)mic|1/2
ENDDO
ENDDO

```

FIG. 1. Parallel ordering of significant distributions.

mostly differs from the original version by the presence of the innermost loop over lattice vector \mathbf{m} . For each basis function c , the list $\{ac^{\mathbf{m}}\}$ is created and the two indices a and \mathbf{m} are ordered by decreasing value of the integral estimate $|(ac^{\mathbf{m}}|ac^{\mathbf{m}})_{\text{mic}}|^{1/2}$.

In the third step, the parallel ONX K builds start with the outer loops over the rows and columns of the local density matrix $D(i\text{PrC})$, as shown in Fig. 2. Next, a twofold loop over the distribution lists of $\{ac^{\mathbf{m}}\}$ and $\{bd^{\mathbf{n}}\}$ takes place in which ERIs are computed. Note that in Fig. 2, $bd_i^{\mathbf{n}}$ stands for the first element of the list $\{bd^{\mathbf{n}}\}$ for a B and D given. The contraction of the ERIs with the density matrix is done after the loops over the lists with the help of a LEVEL-2 BLAS matrix-vector multiplication subroutine (DGEMV). The manipulations of the density and exchange matrices are carried out with the FASTMAT data structure.³⁷

In the fourth step every processor, including the root, computes the lower triangular part of the exchange submatrix $K(D(i\text{PrC}))$ without any interprocessor communication. The work corresponding to each atom-block D_{CD} used to compute the exchange submatrix $K(D_{CD})$ is timed and collected in $\text{Time}(C,D)$. At the end of the parallel exchange build, the timing information is collected by the root processor. Thus the new timing matrix Time can be used to partition the density matrix for the next SCF iteration. The distributed exchange submatrices are globally summed through a binary tree based communication to the root processor. The strategy currently used in MONDOSCF (Ref. 38) to store all the matrices on the root processor will be replaced in the near future by a completely distributed scheme.

To avoid excessive interprocessor communications the algorithm does not use permutational symmetry of the ERIs.

```

Received(iPrc) from ROOT

Make list {acm} and {bdn}

DO C ∈ COL(D(iPrc))
  DO D ∈ ROW(D(iPrc), C)

    ti = TIMER()

    DO A = 1, NAToms
      DO B = 1, A

        DO a, c, m ∈ {acm}(A, C)

          IF(max(|Dcd|)(acm|acm)mic)1/2 &
            & |(bdn|bdn)mic)1/2 < Thresh) EXIT

          DO b, d, n ∈ {bdn}(B, D)

            IF(|Dcd|)(acm|acm)mic)1/2 &
              & |(bdn|bdn)mic)1/2 < Thresh) EXIT

            COMPUTE (acm|bdn)mic

          ENDDO

        ENDDO

      Digest D and ERIs

    ENDDO

  ENDDO

  tf = TIMER()

  Time(C, D) = tf - ti

ENDDO

Global summation of K

```

FIG. 2. The parallel ONX loops.

This approach definitely increases the number of ERIs to compute at each Fock build by a factor of 2–3;⁵ however, at the same time it greatly simplifies the coding and avoids complicated synchronization problems between processors to send/receive blocks of the density and/or exchange matrices. It also becomes much easier to control the amount of work given to each processor.

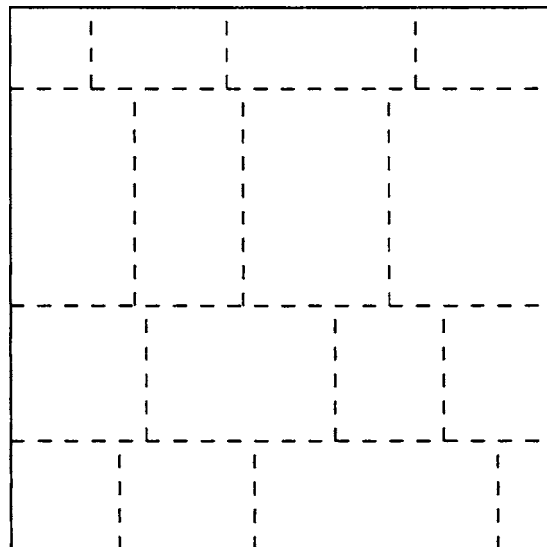


FIG. 3. Example of a horizontal and vertical recursive decompositions for 16 processors.

A. Partition

In this section we present a two level version of the multiple recursive decomposition introduced in Ref. 39. To estimate the time needed to process each block we define a non-negative cost function ϕ on contiguous atom-atom blocks of D . In our case the function ϕ is either the number of nonzero elements of D or the time needed for the computation of the exchange matrix $K(D_{CD})$, where D_{CD} is the CD th atom-atom block of the density matrix (see Fig. 2). We will not discuss the nonzero partitioning; indeed this partition leads to a very poor load balance, and it is only used to generate an initial partition (i.e., when the timing function is not available). The two level recursive partition is obtained through a two step scheme: first a one-dimensional 1D row-wise partition, followed by a second columnwise partition of the rows. In the following we will assume that $\text{NPrc} = p^2$ with p a positive integer. A generalization to any number of processors is straightforward and will not be discussed.

B. 1D partition

During the level 1 partition the 1D array task $T_k = \sum_l \phi_{kl}$ ($k, l = 1, \text{NAtoms}$) is divided into p consecutive intervals $[r_i, r_{i+1}]$ such that the sum of the T_i 's elements for each block $1 \leq i < p$ is equal up to a certain error ε .

C. 2D partition

During the level 2 partition, each partitioned row of the cost function ϕ_i is further sectioned into p columns of approximately equal cost $[c_j, c_{j+1}]$ with $1 \leq j < p$. Figure 3 shows an example of a two-level recursive decomposition for 16 processors.

Of course different approaches to partition the density matrix can be used, as, for example, the recursive bipartitioning algorithm with alternative directions by Vastenhouw and Bisseling.⁴⁰ For more informations about the multiple recursive decomposition, the reader can refer to the original paper.³⁹

D. Global summation of the exchange matrices

The global summation of the local exchange matrices to the master node is an important task. To achieve good performance, a binary tree-based reduction paradigm is used.^{41,42} It requires only $P-1$ communications to globally sum the distributed exchange matrices to the root processor. In a first step, each even node gets the incomplete matrix from the uneven node on the right and adds these two matrices together. In the second step the uneven nodes are left out and the process is repeated for the remaining nodes. This continues until the total matrix is on the root processor.

III. IMPLEMENTATION

All developments were implemented in the MONDOSCF (Ref. 38) suite of programs for linear scaling electronic structure theory and *ab initio* molecular dynamics. The code was compiled using the PORTLAND GROUP F90 COMPILER pgf90 v5.1 (Ref. 43) with the $-O1$ options and with the GNU C COMPILER gcc v3.2.2 using the $-O1$ flag or with the HP FORTRAN COMPILER f95 v5.5A (Ref. 44) and the $-O1$ option and the COMPAQ C COMPILER cc v6.5 (Ref. 45) and the $-O1$ flag. Timings are performed using the message passing interface (MPI)_WTIME function. In some MPI distributions the internal MPI_WTIME routine may be very slow (e.g., globally synchronized), in this case we use the routine interval performance monitoring (IPM)_timer_get_time contained in the IPM timing routine library.⁴⁶ At the present moment, the ERIs are evaluated according to the vertical recurrence relation (VRR) introduced by Obara and Saika⁴⁷ and the modifications proposed by Head-Gordon and Pople.⁴⁸ The density matrix is solved with the help of the quartic trace resetting (TRS4) algorithm.²⁵ The calculations are performed at the GOOD or TIGHT level accuracies⁶ which deliver at least six and eight digits of relative accuracy in the total energy, respectively. More details about the accuracy of the gas and periodic ONX algorithm can be found in Refs. 2 and 6.

The timing for the ONX speedup is taken at the fourth SCF cycle and includes loading and distributing the density matrix to all processors, building the distribution lists, calculation of K , ET partition for the next SCF iteration, global summation of K , and the final I/O.

IV. RESULTS AND DISCUSSIONS

All calculations were carried out on a 1024-node (2048 processors) dual P4 LINUXBIOS/BPROC CLUSTER connected with MYRINET 2000 running RED-HAT LINUX RELEASE 9 (SHRIKE) (Ref. 49) and a cluster of 256 four CPU HP/Compaq Alpha-server ES45s with the QUADRICS QSNET HIGH SPEED INTERCONNECT.

For purpose of performing the scaling tests, we start the calculation with the STO-3G basis set and the GOOD accuracy and switch to the final basis set and accuracy (GOOD or TIGHT) using a mixed integral approach.

We have performed Hartree-Fock parallel calculations on water clusters containing 50 and 110 molecules with the 6-31G and 6-31G** basis sets and on endothelin at the RHF/6-31G** level as well. For the periodic system, we have chosen a box of 64 water molecules at the 6-31G, 6-31G**, and

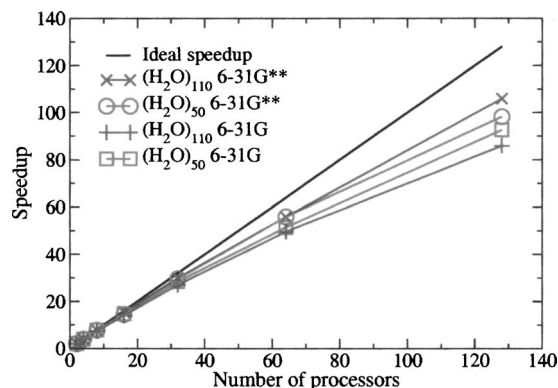


FIG. 4. Scaling of the parallel ONX on $(\text{H}_2\text{O})_{50}$ and $(\text{H}_2\text{O})_{110}$ with the 6-31G and 6-31G** basis functions and the TIGHT threshold. Speedups are relative to two processors.

6-31G*(O)/5-11G*(H) basis sets and a cell of 32 KCNS in the P_{bcm} point group with a 86-511G(K)/86-311G(S)/6-311G(C, N) basis sets. The basis sets for the KCNS calculations were obtained from Ref. 50. These systems are chosen because they are inhomogeneous and three dimensional, thus posing a challenge to parallel ONX.

The number of processors typically varies between 2 to 128. It is worth to mention that our partitioning scheme can be applied to any number of processors, not only of power of 2 as long as the number of processors remains smaller or equal to the number of atom blocks in the density matrix. The speedups and efficiencies are given relative to two processors.

A. Water cluster

The speedup obtained on the BPROC CLUSTER for the $(\text{H}_2\text{O})_{50}$ and $(\text{H}_2\text{O})_{110}$ clusters with different basis sets are shown in Fig. 4. The current partitioning scheme gives an efficiency, for the 50 water molecule cluster (where the number of processors per heavy atom is 2.56), of 72% and 77% with 128 processors, which corresponds to speedups of about 93 and 98 for the 6-31G and 6-31G** basis sets, respectively. At the same number of processors (128), the 110 water molecules gives efficiencies of 67% and 83% and speedups of 86 and 106, respectively. We can also observe in Fig. 4 that the speedup is slightly better for the $(\text{H}_2\text{O})_{50}$ than the $(\text{H}_2\text{O})_{110}$ cluster at the RHF/6-31G level of theory. The reason is that the cluster has been in full operation during all the calculations and the timings have been influenced by the other jobs on a given node. Figure 5 shows the evolution of the relative speedup for the exchange matrix build of the $(\text{H}_2\text{O})_{50}$ cluster along the SCF iterations for 64 and 128 processors and the RHF/6-31G** level of theory. The initial low speedup gained from the use of the ET partitioning reflects a large change in the density matrix during the two first iterations. However, the relative speedup increases very rapidly and already at the third iteration (No. 2) reaches a stable value. This relative speedup is preserved throughout the remaining SCF iterations. Thus the performance of the ET is quite insensitive to the SCF cycle, after a stabilization time.

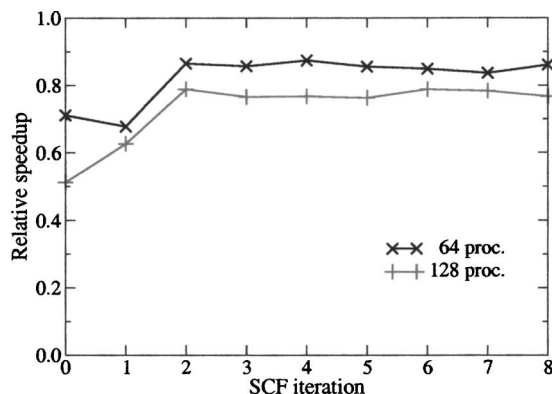


FIG. 5. Relative speedup of the parallel ONX on $(\text{H}_2\text{O})_{50}$ with the RHF/6-31G** along the SCF iterations. Speedups are relative to two processors.

B. Water box with PBC

For the periodic systems, the result of the 64-water molecules obtained on the BPROC CLUSTER is shown in Fig. 6. The calculations are performed at the relativistic Hartree-Fock (RHF) level of theory with the 6-31G*(O)/5-11G*(H), 6-31G and 6-31G** basis sets and the GOOD accuracy. Efficiencies are 87%, 75%, and 79% for the 6-31G*/5-11G*, 6-31G, and 6-31G** basis sets, respectively, with 128 processors. The overall parallel ONX is very good up to 64 processors but degrades slightly at the 128-processor level.

These performances, obtained for the water box, are better compared to the $(\text{H}_2\text{O})_{50}$ cluster. The improved speedup is due to the longer time spent in the exchange build for the periodic cases compared to the gas phase cases. As the computation time spent in the innermost loop increases, due to the double summations over lattice vectors, the timing function gives a more accurate elapsed time for each atom-block of the density matrix.

C. Endothelin and periodic KCNS

Finally, we present, in Fig. 7, speedup obtained on the Compaq machine for endothelin ($\text{C}_{82}\text{H}_{122}\text{O}_{26}\text{N}_{20}\text{S}_5$) and for a orthorhombic supercell composed of 32 KCNS (P_{bcm}) at the RHF/6-31G** and RHF/86-511G(K)/86-311G(S)/6-311G(C,N) level of theories, respectively. These scalings

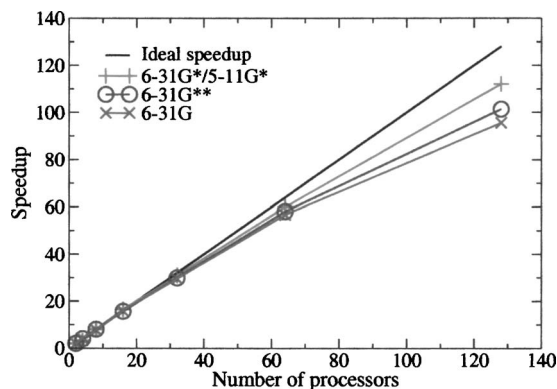


FIG. 6. Scaling of the parallel ONX on $(\text{H}_2\text{O})_{64}$ PBC with the RHF/6-31G*(O)/5-11G*(H), the RHF/6-31G**, and RHF/6-31G level of theories and the GOOD threshold. The speedups are relative to two processors.

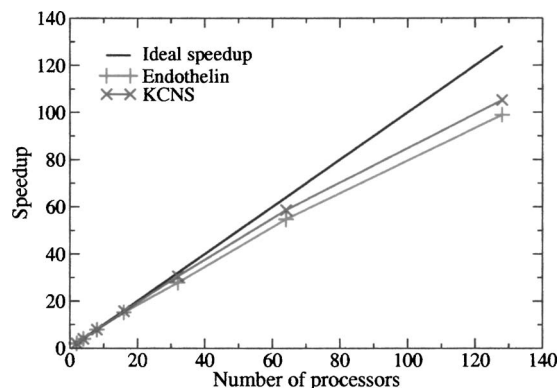


FIG. 7. Scaling of the parallel ONX on endothelin and periodic $(\text{KCNS})_{32}$ with the RHF/6-31G** and RHF/86-511G(K)/86-311G(S)/6-311G(C,N), respectively, and the GOOD threshold. The speedups are relative to two processors.

have been carried out with the GOOD accuracy level. At the 128-processor level, the endothelin (133 heavy atoms) delivers 99-fold speedup, while the $2 \times 2 \times 2$ KCNS (128 atoms per simulation supercell) delivers 105-fold speedup. We also show in Fig. 8 the total speedup of the fourth SCF for endothelin as well as the three most consuming parts of the SCF, i.e., the Coulomb build (quantum chemical tree code¹³ QCTC), the exact exchange build (ONX) and the density matrix solver (TRS4). The timing for the different parts are reported in Table I.

V. CONCLUSION

We have presented a simple and efficient algorithm for parallelization of the exact Hartree-Fock exchange. The method is based on the dynamical distribution of the density matrix to the processors. At each SCF cycle the time needed to build the local exchange matrix corresponding to the local density matrix on a given processor is used to distribute the density matrix in the next cycle.

The concept of ET has proven fruitful for load balancing the ONX algorithm. ET exploits the “slow” variation of the density matrix between SCF iterations to overcome the irregularities arising from the $\mathcal{O}(N)$ methods.

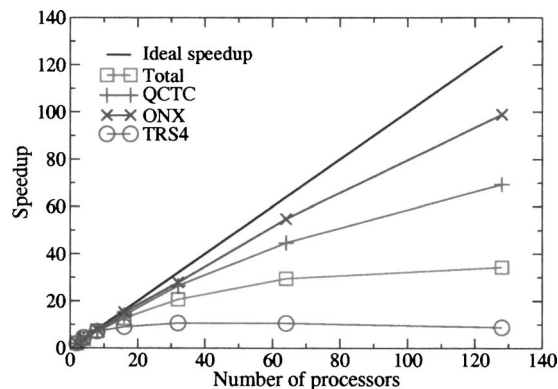


FIG. 8. Total scaling of the parallel SCF and its most consuming parts on endothelin with the RHF/6-31G** and the GOOD threshold. The speedups are relative to two processors.

TABLE I. Timing of the parallel SCF and its most consuming parts on endothelin with the RHF/6-31G** and the GOOD threshold. Times are in seconds.

No. of processors	Total	QCTC	ONX	TRS4
2	19849	10937	6810	1737
4	10284	5669	3432	760
8	5471	2910	1733	484
16	3127	1561	896	380
32	1922	825	490	324
64	1346	489	249	325
128	1156	315	138	391

A finer grain partition of the density matrix (i.e., at the function level) may lead to even better load balance. However, the code may become much slower due to the non-negligible overhead of the timing routine.

The overall efficiency of the ET partition ranges from 74%–87% for all test cases presented in this work with the fine grained (up to 2.56 processors per heavy atom) 128 processor calculations. This high efficiency of the ET retains this property between geometry steps in a geometry optimization or molecular dynamics run after few steps (usually less than 3) of stabilization of the density matrix.

We can also note that the ONX algorithm includes non-parallel step as loading, distributing the density matrix to all processors, global summation, and I/O. Thus we may expect the efficiency to improve with full data parallelism.

ACKNOWLEDGMENTS

This work has been supported by the US Department of Energy under contract W-7405-ENG-36 and the ASCI project. The Advanced Computing Laboratory of Los Alamos National Laboratory is acknowledged.

¹E. Schwegler and M. Challacombe, J. Chem. Phys. **105**, 2726 (1996).

²E. Schwegler, M. Challacombe, and M. Head-Gordon, J. Chem. Phys. **106**, 9708 (1997).

³E. Schwegler, M. Challacombe, and M. Head-Gordon, J. Chem. Phys. **109**, 8764 (1998).

⁴E. Schwegler and M. Challacombe, J. Chem. Phys. **111**, 6223 (1999).

⁵E. Schwegler and M. Challacombe, Theor. Chem. Acc. **104**, 344 (2000).

⁶C. J. Tymczak, V. Weber, E. Schwegler, and M. Challacombe, J. Chem. Phys. **122**, 124105 (2005).

⁷C. A. White, B. Johnson, P. Gill, and M. Head-Gordon, Chem. Phys. Lett. **230**, 8 (1994).

⁸C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Chem. Phys. Lett. **253**, 268 (1996).

⁹M. Challacombe, E. Schwegler, and J. Almlöf, *Computational Chemistry: Review of Current Trends* (World Scientific, Singapore, 1996), pp. 53–107.

¹⁰M. Challacombe, E. Schwegler, and J. Almlöf, J. Chem. Phys. **104**, 4685 (1996).

¹¹M. C. Strain, G. E. Scuseria, and M. J. Frisch, Science **271**, 51 (1996).

- ¹²J. M. Pérez-Jordá and W. T. Yang, J. Chem. Phys. **107**, 1218 (1997).
- ¹³M. Challacombe and E. Schwegler, J. Chem. Phys. **106**, 5526 (1997).
- ¹⁴C. J. Tymczak and M. Challacombe, J. Chem. Phys. **122**, 134102 (2005).
- ¹⁵J. M. Pérez-Jordá and W. Yang, Chem. Phys. Lett. **241**, 469 (1995).
- ¹⁶R. E. Stratmann, G. E. Scuseria, and M. J. Frisch, Chem. Phys. Lett. **257**, 213 (1996).
- ¹⁷C. F. Guerra, J. G. Snijders, G. teVelde, and E. J. Baerends, Theor. Chem. Acc. **99**, 391 (1998).
- ¹⁸M. Challacombe, J. Chem. Phys. **113**, 10037 (2000).
- ¹⁹X. P. Li, R. W. Nunes, and D. Vanderbilt, Phys. Rev. B **47**, 10891 (1993).
- ²⁰M. S. Daw, Phys. Rev. B **47**, 10895 (1993).
- ²¹A. D. Daniels, J. M. Millam, and G. E. Scuseria, J. Chem. Phys. **107**, 425 (1997).
- ²²A. H. R. Palser and D. E. Manolopoulos, Phys. Rev. B **58**, 12704 (1998).
- ²³M. Challacombe, J. Chem. Phys. **110**, 2332 (1999).
- ²⁴A. M. N. Niklasson, Phys. Rev. B **66**, 155115 (2002).
- ²⁵A. M. N. Niklasson, C. J. Tymczak, and M. Challacombe, J. Chem. Phys. **118**, 8611 (2003).
- ²⁶M. W. Schmidt, K. K. Baldrige, J. A. Boatz *et al.*, J. Comput. Chem. **14**, 1347 (1993).
- ²⁷D. Chasman, M. Beachy, L. Wang, and R. Friesner, J. Comput. Chem. **19**, 1017 (1998).
- ²⁸M. E. Colvin, C. L. Janssen, R. A. Whiteside, and C. H. Tong, Theor. Chim. Acta **84**, 301 (1993).
- ²⁹T. R. Furlani and H. F. King, J. Comput. Chem. **16**, 91 (1995).
- ³⁰I. T. Foster, J. L. Tilson, A. F. Wagner, R. L. Shepard, R. J. Harrison, R. A. Kendall, and R. J. Littlefield, J. Comput. Chem. **17**, 109 (1996).
- ³¹R. J. Harrison, M. F. Guest, R. A. Kendall *et al.*, J. Comput. Chem. **17**, 124 (1996).
- ³²Y. Alexeev, R. A. Kendall, and M. S. Gordon, Comput. Phys. Commun. **143**, 69 (2002).
- ³³H. Takashima, S. Yamada, S. Obara, K. Kitamura, S. Inabata, N. Miyakawa, K. Tanabe, and U. Nagashima, J. Comput. Chem. **23**, 1337 (2002).
- ³⁴R. Lindh, J. W. Krogh, M. Schütz, and K. Hirao, Theor. Chem. Acc. **110**, 156 (2003).
- ³⁵J. Almlöf, K. Faegri, and K. Korsell, J. Comput. Chem. **3**, 385 (1982).
- ³⁶C. K. Gan and M. Challacombe, J. Chem. Phys. **118**, 9128 (2003).
- ³⁷C. K. Gan, C. J. Tymczak, and M. Challacombe, J. Chem. Phys. **121**, 6608 (2004).
- ³⁸M. Challacombe, C. Tymczak, C. K. Gan, K. Németh, V. Weber, E. Schwegler, and A. Niklasson, MONDOSCF v1.0a10, A program suite for massively parallel, linear scaling SCF theory, and *ab initio* molecular dynamics, 2001, Los Alamos National Laboratory (LA-CC 01-2), Copyright University of California, <http://www.t12.lanl.gov/home/mchalla/>.
- ³⁹L. F. Romero and E. L. Zapata, Parallel Comput. **21**, 583 (1995).
- ⁴⁰B. Vastenhouw and R. H. Bisseling, SIAM Rev. **47**, 67 (2005).
- ⁴¹G. Fox, M. Johnson, G. Lyzenga, S. Otto, J. Salmon, and D. Walker, *Solving Problems on Concurrent Processors* (Prentice-Hall, Upper Saddle River, NJ, 1988).
- ⁴²R. A. van de Geijn, *LAPACK Working Note 29* (University of Tennessee, Tennessee, 1991).
- ⁴³The Portland Group, PDF90 v5.1, 2002 <http://www.pgroup.com/>.
- ⁴⁴Hewlett-Packard, F95 v5.5A, 2003, <http://www.hp.com/>.
- ⁴⁵Compaq, C v6.5, 2002, <http://www.hp.com/>.
- ⁴⁶K. Asanovic, Interval Performance Monitoring (IPM) package Release 2.0, 1995, <http://www.cag.lcs.mit.edu/~krste/ipm/IPM.html>.
- ⁴⁷S. Obara and A. Saika, J. Chem. Phys. **84**, 3963 (1986).
- ⁴⁸M. Head-Gordon and J. A. Pople, J. Chem. Phys. **89**, 5777 (1988).
- ⁴⁹Redhat, REDHAT v9.0, 2004, <http://www.redhat.com/>.
- ⁵⁰R. Dovesi, V. Saunders, C. Roetti, M. Causà, N. Harrison, R. Orlando, and C. Zicovich-Wilson, CRYSTAL98 *Basis Sets*, 2003, http://www.crystal.unito.it/Basis_Sets/ptable.html.