Parallelization of Four-Component Calculations. I. Integral Generation, SCF, and Four-Index Transformation in the Dirac–Fock Package MOLFDIR

M. PERNPOINTNER,¹ L. VISSCHER,² W. A. DE JONG,³ R. BROER¹

¹Theoretical Chemistry, Materials Science Centre, Rijksuniversiteit Groningen, Nijenborgh 4, 9747 AG Groningen, The Netherlands
²Division of Chemistry, Faculty of Sciences, Vrije Universiteit Amsterdam, De Boelelaan 1083, NL-1081 HV Amsterdam, The Netherlands
³Environmental Molecular Sciences Laboratory, Pacific Northwest National Laboratory, P.O. Box 999, Richland, Washington 99352

Received 7 March 2000; accepted 19 April 2000

ABSTRACT: The treatment of relativity and electron correlation on an equal footing is essential for the computation of systems containing heavy elements. Correlation treatments that are based on four-component Dirac–Hartree–Fock calculations presently provide the most accurate, albeit costly, way of taking relativity into account. The requirement of having two expansion basis sets for the molecular wave function puts a high demand on computer resources. The treatment of larger systems is thereby often prohibited by the very large run times and files that arise in a conventional Dirac–Hartree–Fock approach. A possible solution for this bottleneck is a parallel approach that not only reduces the turnaround time but also spreads out the large files over a number of local disks. Here, we present a distributed-memory parallelization of the program package MOLFDIR for the integral generation, Dirac–Hartree–Fock and four-index MS transformation steps. This implementation scales best for large AO spaces and moderately sized active spaces. © 2000 John Wiley & Sons, Inc.

Keywords: four-component calculations; Dirac–Fock correlations

Correspondence to: L. Visscher; e-mail: visscher@chem.vu.nl
Contract/grant sponsor: Stichting Nationale Computerfaciliteiten (NCF); contract/grant number: NRG.98.06
Contract/grant sponsor: U.S. Department of Energy, Mathematical, Information and Computational Science Division (to W.d.J.); contract/grant number: DE-AC06-76RLO 1830 (with Battelle Memorial Institute)
Introduction

Molecular Dirac–Hartree–Fock (DHF) calculations require substantial computational effort due to the four component character of the wave function. Already, the integral generation needs more time and space requirements than in the nonrelativistic case where we have a computational effort for integral generation that scales like $\frac{1}{2}N^4$. $N$ denotes the number of scalar basis functions in the nonrelativistic case, which is approximately equal to the number of large component basis functions ($N_L$) in the DHF case. For the four-component case and the realistic assumption that the small component basis ($N_S$) is about twice the size of the large component basis ($N_S \approx 2N_L$), we get a scaling for the three Coulomb integral generations in MO basis comprises the fourfold transformation program ROTRAN, which generates two-electron integrals in the MS basis.

We first introduce some notations. The lowercase letters $p, q, r, \text{and } s$ will be used for basis function labels in the AO space, and the letters with a tilde $\tilde{r}, \tilde{s}, \ldots$ mean a whole range of that label. If we want to confine ourselves to a specific range of a basis function label (e.g., function groups only attributed to one node), we add the node index as a subscript like $\tilde{r}_Q$, $\tilde{s}_Q$ for all functions present on node $Q$.

The development of the serial ROTRAN code and its extensions can be found in refs. 3 and 4, and a detailed description of the four-component four-index transformation is given in ref. 7. We, therefore, restrict ourselves to the essential steps of the integral transformation where we focus on the differences to a nonrelativistic MO transformation.

First, we briefly summarize the nonrelativistic four-index transformation. The central task for generating the integrals in MO basis comprises the fourfold transformation

$$ (ij|kl) = \sum_{p=1}^{N} \sum_{q=1}^{N} \sum_{r=1}^{N} \sum_{s=1}^{N} C_{pq}^{rs} C_{ij}^{rs} C_{kl}^{pq} $$

where $N$ is the size of the atomic basis, and $M$ the number of active MOs. Formally, this problem looks as if it scaled like $M^4N^4$ executing the additions/multiplications as given. By a simple rearrangement of the summations one can easily see that the problem reduces to a $N^4M + N^3M^2 + N^2M^3 + NM^4$ process. We distinguish here between the individual quarter transformations to indicate the computational cost associated with each step. If the AO basis is much larger than the space of transformed spinors, it is essential to focus on reduction of the computational cost in the first two steps, which can be achieved by parallelizing these steps as will be discussed below. The condition $M \ll N$ is usually fulfilled because we do not take core states or very high lying virtual spinors into the active space.

A detailed analysis of the computational effort of nonrelativistic four-index transformations was already done at the end of the 1960s by Tang and Edmiston,\cite{8} and 1974 by Diercks,\cite{9} Tang and Edmiston split the total transformation in two two-index transformations called the first- and second half-transformation (here we consider the necessary...
The various approaches and most efficient loop structures of the four-index transformation can be found elsewhere,\textsuperscript{10–17} and we will not discuss the details of these realizations. The loop structure of ROTRAN and the construction of intermediate matrices is closely related to the algorithm by Saunders and van Lenthe.\textsuperscript{17} The sorting steps in the conventional approach need a lot of disk space and fast I/O channels, where an algorithm according to this article is not adapted to the symmetry of the nuclear framework but considerably reducing the number of integrals that have to be stored and processed. Normally, this results in a reduction by a factor of 8. Exploitation of molecular symmetry to the two-electron integrals will further reduce the amount of integrals as outlined by numerous authors.\textsuperscript{22–29}

In the MOLDFIR code one uses the fact that the four-component MS spinors are not only adapted to the symmetry of the nuclear framework but also form degenerate pairs due to time reversal or Kramers’ symmetry. This gives some of the computational saving that is obtained by the decoupling of the spin in nonrelativistic methods. In ROTRAN, both molecular point group symmetry and Kramers symmetry are implemented. We now want to outline the essential steps and differences to the nonrelativistic case for the relativistic four-index transformation.

First of all, the transformation also includes an integration over the spin variable because the AO basis consists of four-component spinors in which only one component is nonzero, while the MS basis consists of spinors with all components nonzero. We can express this integration in the scalar basis and show that only certain classes of two-electron integrals need to be taken into account.

We write down the tensor product of the two one-particle spinors $|\Theta(1)\rangle$ and $|\Xi(2)\rangle$ according to

$$
|\Theta(1)\rangle \otimes |\Xi(2)\rangle = (\theta^L(1), \theta^S(1)) \left(\begin{array}{c}
\theta^\alpha(1) \\
\theta^\beta(1)
\end{array}\right) \\
\otimes \left(\begin{array}{c}
\xi^L(2) \\
\xi^S(2)
\end{array}\right)
$$

The operator $\frac{1}{r_{12}}$ can be viewed as a 16 × 16 diagonal matrix

$$
\frac{1}{r_{12}} \mathbf{1}_{16} = \frac{1}{r_{12}} \mathbf{1}_4 \otimes \frac{1}{r_{12}} \mathbf{1}_4
$$

that couples the four-component bra and ket spinors. The final Coulomb integral then reads as

$$
\langle \Gamma \Lambda | \Theta \Xi \rangle^C = \int \int d\tau_1 d\tau_2 \left(\gamma^\alpha L_a(1)\gamma^\beta L_a(2), \ldots\right) \frac{1}{(\mathbf{r}_{12})^3} \times \mathbf{1}_{16} \left(\gamma^\alpha L_a(1)\gamma^\beta L_a(2), \ldots\right)^T.
$$

After performing the matrix operations, one easily verifies that only the following combinations survive:

$$
\langle \Gamma \Lambda | \Theta \Xi \rangle^C \rightarrow \left(\begin{array}{c}
\langle LL|LL \rangle \\
\langle LL|SS \rangle \\
\langle SS|LL \rangle \\
\langle SS|SS \rangle
\end{array}\right) \otimes \left(\begin{array}{c}
\langle \alpha \alpha | \alpha \alpha \rangle \\
\langle \alpha \alpha | \beta \beta \rangle \\
\langle \beta \beta | \alpha \alpha \rangle \\
\langle \beta \beta | \beta \beta \rangle
\end{array}\right).
$$

The same method is applied to the Gaunt integrals

$$
\langle \Gamma \Lambda | \Theta \Xi \rangle^G = -\left(\Gamma \Lambda \left| \frac{\alpha_1 \cdot \alpha_2}{r_{12}} \right| \Theta \Xi\right),
$$

where the matrices $\alpha_1$ and $\alpha_2$ now refer to the two distinct particle labels. To find the correct matrix representation for the Gaunt operator, we construct the matrix $\alpha(1) \otimes \alpha(2)$, which gives

$$
\alpha_1 \cdot \alpha_2 = \alpha(1) \otimes \alpha(2)
$$

For the Gaunt integral, therefore, the following combinations survive:

$$
\langle \Gamma \Lambda | \Theta \Xi \rangle^G \rightarrow \left(\begin{array}{c}
\langle LL|LS \rangle \\
\langle LL|SL \rangle \\
\langle SL|LS \rangle \\
\langle SL|SL \rangle
\end{array}\right) \otimes \left(\begin{array}{c}
\langle \alpha \alpha | \alpha \alpha \rangle \\
\langle \beta \beta | \beta \beta \rangle \\
\langle \beta \beta | \alpha \alpha \rangle \\
\langle 2\alpha \beta | \alpha \alpha \rangle
\end{array}\right).
$$

Due to permutation symmetry of the spatial parts, only one of the above four has to be calculated [e.g., $\langle SL|SL \rangle$].

Taking these considerations into account, we now formulate the four-component transformation steps explicitly, and determine the numerical effort and storage requirements in comparison to the nonrelativistic case. Again, we denote the number of large-component basis functions by $N_L$ and the number of
The transformation of the first index reads as

\[(ij|rs)^{\alpha \beta} = \sum_{p=1}^{N_i} (pq|rs)C_{\alpha \beta}^{pq}, \]  
\[(ij|rs)^{\alpha \alpha} = \sum_{s=1}^{N_i} (ip|rs)C_{\alpha \alpha}^{is}, \]  
\[(ij|rs)^{\beta \beta} = \sum_{s=1}^{N_i} (ij|rs)C_{\beta \beta}^{js}, \]  
\[(ij|rs)^{\beta \alpha} = \sum_{s=1}^{N_i} (ij|rs)C_{\beta \alpha}^{js}. \]  

In the subsequent transformation step to the \((ij|rs)\) integrals we take only the surviving spin combinations into account:

\[(ij|rs) = (ij|rs)' + ((ij|rs)'), \]  
\[(ij|rs) = (ij|rs)' + ((ij|rs)'), \]  
\[(ij|rs) = (ij|rs)' + ((ij|rs)'), \]  
\[(ij|rs) = (ij|rs)' + ((ij|rs)'), \]  

Because we use the permutation symmetry in the first two indices at the scalar level to reduce the number of scalar integrals we need a symmetrization step after the first half-transformation.

\[(ij|rs) = (ij|rs)' + ((ij|rs)'), \]  
\[(ij|rs) = (ij|rs)' + ((ij|rs)'), \]  
\[(ij|rs) = (ij|rs)' + ((ij|rs)'), \]  
\[(ij|rs) = (ij|rs)' + ((ij|rs)'), \]  

The computational effort is further reduced by exploiting time-reversal (TR) or Kramers' symmetry, which causes a twofold degeneracy of all eigenvalues in the absence of an external magnetic field. The TR operator \(\hat{K}_4\) is represented by

\[\hat{K}_4 = -i \begin{pmatrix} \sigma_y & 0 \\ 0 & \sigma_y \end{pmatrix}. \]

Here we incorporate the permutation symmetry in the second index pair by generating the missing integrals before the third quarter-transformation. This cannot be done after the transformation like in the first half-transformation because we now start with a complex matrix instead of a real matrix.

Finally, we account for particle–particle symmetry (which corresponds to the \((pq|rs) = (rs|pq)\) symmetry at the scalar level) by forming

\[(ij|kl) = (ij|kl)' + (kl|ij)'). \]

With this background we are able to outline the storage requirements and computational cost of a four-component four-index transformation.

As mentioned in the first section, we have a factor of 25 in the amount of scalar integrals to be stored compared to the nonrelativistic case. Taking Kramers' symmetry into account we then generate \(M^2 (\frac{N_{sp}^2}{2} + \frac{N_{sp}^2}{2})\) complex numbers in the relativistic first half-transformation in comparison to \(\frac{1}{2}N_{sp}^2M^2\) real numbers in the nonrelativistic case. Inserting the above relations for the basis set sizes we end up with a ratio rel./nonrel. of approximately 20 in the disk storage requirements for this step. In the second step, we store the integral classes \((ij|kl),\)
plex numbers in contrast to \( 1 \) complex numbers in contrast to \( \frac{1}{2} M \) real numbers in the nonrelativistic case (factor of 16). From these numbers we see a huge increase in the necessary disk space, which can easily become a bottleneck on one node when the systems grow bigger. We note in passing that the last ratio is not specific for four-component calculations but also holds for two-component methods in which spin-orbit coupling is included in the Hartree–Fock stage.

Now we turn to the comparison of the computational effort, where we have to remember that we perform the multiplications only over the \( \alpha \) part due to Kramers’ symmetry and use complex arithmetic. Multiplying a real number with a complex number needs twice as many operations relative to the ordinary real number multiplication. The first half-transformation can undergo an arbitrary symmetry operation \( \hat{R} \) as an element of the allowed molecular point-group operations. This produces either an identical or a different type/center combination. For example, a \( p_x \) function located at the atom 1 of an octahedral molecule transforms under the \( C_3[111] \) operation into a \( p_y \) orbital located at position 2. That means the \( C_3[111] \) operation couples these two function types and centers together. If there exists such a symmetry relation between function types on different centers, two-electron integrals containing these functions contribute to the same extent to the Fock matrix and, therefore, only the symmetry unique integrals should be calculated. These observations materialized in Pitzer’s theorem. According to Dupuis’ terminology, the individual type/center combinations are counted by the so-called Grande list index. This Grande list index does not yet designate individual basis function labels but counts the symmetry-related functions. The individual basis functions resembling the symmetry properties

**Parallelization**

**INTEGRAL GENERATION**

The integrals in the corresponding module RELTWEL are generated according to a scheme outlined by Dacre and Elder. Dupuis and King have reconsidered this scheme in ref. 24, and given a clear and general mathematical framework including the terminology, that makes it easier to refer to different levels in the hierarchy of the computation process. To simplify the outline of the parallel RELTWEL code we will closely follow their terminology.

Taking the correct types of multiplication into account, we end up with a ratio of 1:52 in (1), 1:208 in (2), and 1:40 in (3).

Similar considerations hold for the second half-transformation:

(1) Third index: \( 2N_{1}^{2}M^{3} + 2N_{2}^{3}M^{3} \)

(2) Fourth index: \( 4N_{1}M^{4} + 4N_{3}M^{4} \)

(3) Symmetrization: \( 16M^{4} \)

This amounts to ratios between the nonrelativistic and relativistic transformation of 1:80 for (1), 1:96 for (2), and 1:64 for (3).

We already mentioned that the present implementation performs best in the case of strongly differing sizes of the scalar and transformed spinor space. To corroborate this, we form the ratio of the computational effort for the first and second half-transformation step using the above formulas:

(a) spaces of comparable size \( M = N_{L} \): \( 130 N_{L}^{4} \cdot 0.88 N_{L}^{4} \), including \((SS|SS)\) integrals and \( 50 N_{L}^{3} \cdot 24 N_{L}^{3} \), neglecting the \((SS|SS)\) integrals. (b) Strongly differing orbital spaces \( M = \frac{1}{16} N_{L} : 3.64 N_{L}^{4} \cdot 0.048 N_{L}^{4} \), including \((SS|SS)\) integrals and \( 0.14 N_{L}^{3} \cdot 0.0096 N_{L}^{3} \), neglecting the \((SS|SS)\) integrals.

From these ratios it becomes apparent that for a large scalar basis and a moderately sized active space the first half-transformation certainly represents a bottleneck. On the other hand, the cost of the first half-transformation step can be scaled down linearly with an increasing number of processors as we will see further below. Therefore, parallelization is a perfect means to overcome the problems in this step. Another interesting feature is that reduction of the active spinor space reinforces the savings obtained by neglect of the \((SS|SS)\) integral class.
of such a Grande list index all belong to one such block. A well-known loop construction is to generate the permutationally unique Grande list quadruples with respect to the well-known permutation relations

\[ (IJ|KL) = (JI|KL) = (IJ|LK) = (JI|LK) = (KL|IJ) = (LK|IJ) = (KL|JI) = (LK|JI). \]  

Furthermore, if \( \tilde{R}I = I' \), for example, only one of the blocks \((IJ|KL)\) or \((IJ'|KL)\) needs to be calculated, and the same holds for the other indices in the block. Normally, one sets up the list of symmetry-related Grande list indices at the beginning of the program and takes the highest resulting index as a representative. This reduced list now is both unique with respect to permutational and molecular symmetry, and is called the Petite list. To combine both of these unification processes, one checks during the loop if a given index will occur later on in the loop and leaves it out if this condition is fulfilled. This principle is called “principle of procrastination,” and is fully implemented in RELTWEL. When we remember that for the first transformation step integral streams of the form \((pq|rs)\) are needed, we have to structure the integral code in a way that for a given basis function label \(r, s\) all the symmetry and permutationally allowed labels \(\tilde{p}, \tilde{q}\) are generated, and the corresponding integrals calculated. Because there is always a center and a special function type related to a Petite list index but maybe many basis function labels, it is much more economic to compute the whole block of integrals denoted by a Petite list quadruple \((IJ|KL)\) than to create integrals at the level of individual \((pq|rs)\) indices. Once the whole block of possible basis functions is calculated, it is contracted and stored on a local disk. To facilitate load balancing and use minimal communication at this stage, we chose a master–slave algorithm and communicate the second Petite list index pair to the slaves. Once the node \(Q\) has got the index pair \(1Q, JQ\), the corresponding basis function labels \(\tilde{r}_Q, \tilde{s}_Q\) become available. The nodes themselves now loop automatically over the first index pair, depending on what range of integral generation the user has selected \((LL|LL), (SS|LL), (SS|SS), \) or \((SL|SL))\). This then results in a stream of integrals \(\tilde{p}\tilde{q}|\tilde{r}_Q\tilde{s}_Q\), which now are exclusively grouped with respect to the second index pair. The functionality is outlined in Figure 1.

At the moment, this method of load balancing is based on the time necessary to evaluate a block rather than the resulting file size. The first criterium gives a good balance for the generation step, while the latter criterium is better suited for the DHF step. Imbalances in the DHF step could be remedied by redistributing some integrals after they are generated or by accepting a less optimal load balance in the generation step. It is, however, difficult to find one distribution that is optimal for all three CPU-intensive steps studied here, because the four-index transformation requires yet again a different criterium. The present choice is a pragmatic one that is easy to implement, requires little communication between the nodes, optimizes the generation step, and fulfills the requirement that the integrals be grouped on the second index pair. A secondary problem is that for small systems the coarseness of blocks may give load imbalances already in the generation step. Both effects become less significant for larger systems and integral files.

**DIRAC–HARTREE–FOCK**

As outlined in the section above, after the generation of the two-electron integrals we have sorted integral streams of the form

\[ (\tilde{p}\tilde{q}|\tilde{r}_Q\tilde{s}_Q), \quad \tilde{r}_Q, \tilde{s}_Q \in \{Q\} \]  

available on node \(Q\). This list is completely arbitrary, and only determined by load balance and the master distribution algorithm. We checked if the differently sorted integral lists have any effect on the efficiency of the SCF algorithm, because for the SCF part it is most suitable to have the integrals sorted according to one of the possible 14 types. The inclusion of a bucket sort into the SCF part to generate streams of one single integral type did not lead to a measurable slowdown, because it is completely performed in the core memory. The construction of the

![FIGURE 1. Schematic diagram of the parallel RELTWEL code. On each node we create sorted integral batches that are directly used by the subsequent ROTTRAN program. For the notation conventions, see text.](image-url)
Fock matrix would be very inefficient not having the integrals arranged in these streams. Whenever a bucket has to be emptied, the partial Fock matrices are constructed from this specific stream until all integrals are processed. These partial Fock matrices are then communicated to the master node where they are combined to the full Fock matrix, which is diagonalized. The resulting new densities are communicated back to the nodes where they are needed to create the partial Fock matrices for the subsequent iteration.

**FOUR-INDEX TRANSFORMATION**

Each node can immediately start the first half-transformation locally:

\[(\tilde{p}q|\tilde{r}s)_{Q} \rightarrow (ij|\tilde{r}s)_{Q}\]

where the intermediate file INTTWO contains the half transformed integrals. To continue with the second half transformation it would be necessary to resort the integrals in a way that for a specific \(ij\) all the \(rs\) become available to yield the complete \((ij|kl)\)' integral where the prime indicates that the particle–particle symmetrization is still missing. Because the present implementation does not support a split MS range all the \((ij|kl)\) integrals have to be collected at the end of the second HT to perform the particle–particle symmetrization and produce the final list of MS spinors according to

\[(ij|kl) = (ij|kl)' + (kl|ij)'\]  

We, therefore, have an inevitable communication step at the end of the second HT. This is done efficiently by not reshuffling the integrals from INTTWO, but immediately continuing the second HT according to

\[(ij|\tilde{r}s)_{Q} \rightarrow (ij|\tilde{r}sl)_{Q} = \sum_{r \in \{Q\}} \sum_{s \in \{Q\}} (ij|rs)C_{r}^{s}C_{sl}^{Q}\]  

Because a specific \(ij\) occurs on each node with only a subset of the corresponding \(r \in \{Q\}, s \in \{Q\}\), the second HT indicated above is necessarily incomplete, and this integral batch, therefore, bears the subscript Q. To complete for the missing parts, we add up all the integral batches with matching \(ij\) labels to get the \((ij|kl)\)' integrals. This algorithm is depicted in Figure 2. The master takes over control by looping over the symmetry allowed \(ij\) compound indices and gathering the \((ij|kl)\)' integrals from each node. This implies a synchronization of the nodes to collect the batches for a specific \(ij\), but with a well-balanced distribution of integrals over the nodes

\[(ij|kl) = \sum_{Q=1}^{\text{Nodes}} \sum_{r \in \{Q\}} \sum_{s \in \{Q\}} (ij|rs)C_{r}^{s}C_{sl}^{Q}\]  

synchronization it will not cause significant delays. The local second half-transformation is formulated as

\[(ij|kl)_{Q} = \sum_{r \in \{Q\}} \sum_{s \in \{Q\}} (ij|rs)C_{r}^{s}C_{sl}^{Q}\]  

and leads to a \((ij|kl)_{Q}'\) integral batch. Let the number of spinors be \(\text{NSP}\), then this batch has the length of \(\text{NSP} \times \text{NSP}\) complex numbers (all \(k\) and \(l\)). At this stage we communicate each of the \(Q\) batches to the master and add them up. Formally, we perform the step

\[Q = 1 \sum_{Q=1}^{\text{Nodes}} \sum_{r \in \{Q\}} \sum_{s \in \{Q\}} (ij|rs)C_{r}^{s}C_{sl}^{Q}\]  

leading to a complete non-\(pp\) symmetrized \((ij|kl)\)' batch. The last step now is a symmetrization with respect to \(pp\) symmetry, and is performed on the master node exclusively because it needs all transformed MS spinors to be available. This step cannot be performed in parallel yet, and therefore, does not scale. In this implementation it becomes clear that the scaling should be linear up to the stage of the second HT (see Results and Discussion), and there-
Parallelization of Four-Component Calculations. I

Therefore, this implementation will perform best if the active space is of moderate size but the scalar basis is very large.

Computational Details

We chose the CsF molecule as a test system, which is already large enough to investigate the scaling behavior of the program system. For this purpose we reoptimized a Cs exponent set taken from ref. 31, using a modified version of GRASP\textsuperscript{32,33} which is able to handle Gaussian basis functions in the Dirac–Fock framework. Normally, one also can use a nonrelativistically optimized set and add hard exponents in the $s$ and $p$ region, but we decided for the quite heavy cesium atom to optimize the exponents with respect to the full Dirac Hamiltonian (see Table I). The Dirac–Fock limit for the cesium atom in its $^2S_{1/2}$ ground state is $E_{DF} = -7786.77673329$ a.u. compared to the extended nonrelativistic basis set used in earlier calculations\textsuperscript{31} which gave $E_{DF} = -7786.74200303$ a.u. Careful generation of contraction coefficients is crucial to establish a stable convergence and a good description of kinetic and atomic balance. We took the coefficients produced by the modified GRASP version fully employing $jj$ coupling. For the exponents in Table I the following contraction scheme was applied: $L$: $(24s/18p/14d/2f) \rightarrow [10s/12p/7d/2f]$, $S$: $(18s/24p/18d/14f/2g) \rightarrow [8s/12p/12d/1f]$, and yielded a total energy of $E_{DF} = -7786.7672596$ a.u., which is only 0.07 mHartree above the uncontracted result, showing for the good quality of the contraction.

For fluorine, we did no exponent reoptimization, but used the cc-pVTZ basis by Dunning\textsuperscript{34} and added one hard $p$ exponent of 128.1854 and one soft $s$ function with the exponent 0.09158 yielding a $L$: $(11s/7p/2d/1f) \rightarrow [6s/4p/2d/1f]$, $S$: $(7s/11p/7d/2f/1g) \rightarrow [4s/6p/5d/2f/1g]$ fluorine basis. The $d$ and $f$ exponents provided by Dunning were adapted so that they fulfill the requirement of the

<table>
<thead>
<tr>
<th>Exponents for Cesium and Fluorine as Used for the Calculations.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cs Dual Exponent Set Primitives:</strong> 24s18p14d2f</td>
</tr>
<tr>
<td><strong>F Dual Exponent Set Primitives:</strong> 11s7p2d1f</td>
</tr>
<tr>
<td>73469388.280000</td>
</tr>
<tr>
<td>14464020.390000</td>
</tr>
<tr>
<td>3589737.495000</td>
</tr>
<tr>
<td>1033921.097000</td>
</tr>
<tr>
<td>324455.274600</td>
</tr>
<tr>
<td>109971.252800</td>
</tr>
<tr>
<td>39959.066730</td>
</tr>
<tr>
<td>15432.828390</td>
</tr>
<tr>
<td>6263.7769890</td>
</tr>
<tr>
<td>2636.0674440 d</td>
</tr>
<tr>
<td>1142.4506550 d</td>
</tr>
<tr>
<td>508.7378264 d</td>
</tr>
<tr>
<td>229.8476387 d</td>
</tr>
<tr>
<td>106.7604291 d</td>
</tr>
<tr>
<td>51.1885245 d</td>
</tr>
<tr>
<td>25.0475962 d</td>
</tr>
<tr>
<td>12.4939831 d</td>
</tr>
<tr>
<td>6.2374358 d</td>
</tr>
<tr>
<td>3.0553578 d</td>
</tr>
<tr>
<td>1.4567408 d</td>
</tr>
<tr>
<td>0.1163099 d</td>
</tr>
<tr>
<td>0.0217972 d</td>
</tr>
</tbody>
</table>
family basis. Therefore, we took as \( d \) exponents the closest ones out of the \( s \) set and did the same for the \( f \) exponent. The total molecular basis is of considerable size now, and should give good results.

For the molecule, we did the calculation at the equilibrium bond distance of 2.345351 Å,\textsuperscript{35} yielding a total DF energy of \( E_{\text{DF}} = -7886.6536138 \) a.u. where convergence in energy was achieved up to \( 1.0 \times 10^{-9} \) a.u. To get a better idea of the problem size, we also give the extensions of the orbital spaces. In the used molecular point group \( C_{4v} \), one has four irreducible representations, where representation 1 and 2 resp. 3 and 4 are Kramers pairs. The number of large (small) basis functions in representation 1 and 2 is 75 (165), and in representation 3 and 4 we have 55 (135) functions, which makes a total of 860 scalar basis functions. For the actual calculation it is justified to leave out the \( \int_{\text{SS}} \) integrals as was mentioned in the literature.\textsuperscript{21, 36, 37} The numbers of symmetry unique integrals to compute resulted in 6 million \( \int_{\text{LL}} \) and 48 million \( \int_{\text{SS}} \) integrals.

### Results and Discussion

For the investigation of the scaling behavior we chose two active spaces of different size. The first (large) one comprises correlation of 34 electrons in 136 molecular spinors. The active space was created by keeping the lowest 10 molecular spinors in representation 1 and the lowest 5 spinors in representation 3 frozen, which mainly resemble the Cs 1s, 2s, 2p, 3s, 3p, 3d and F 1s atomic spinors. Virtual orbitals above 4.1 a.u. were deleted. For the small active space only consisting of 42 MS spinors we correlated the valence shell only (mainly the cesium 5s and 5p and fluorine 2s, 2p orbitals) and kept all lower lying orbitals frozen. All virtuals with an orbital energy lower than +0.5 a.u. were included. This space was chosen to create an illustrative example but would be a little bit too small for a good correlation treatment. The computational demands and the scaling behavior indeed change drastically for the smaller active space as we anticipated.

In Table II we present the timing for the large active space. It should be mentioned that the timing results represent maximum run times because at critical steps in the program flow some synchronization is inevitable. For example, a Fock matrix diagonalization can only take place when each node has completely worked through its local integral file. Therefore, the longest integral file is the limiting factor. A good load balance will ensure that these times do not differ strongly. It can be seen that the integral generation scales linearly with the number of nodes, whereas the slope of the SCF scaling is flatter. This can be ascribed to the overhead necessary for the message passing and the fact that the Fock matrix diagonalization is performed on the master node only. For the CsF test system, we measured a diagonalization time of 138 s on the master node, which corresponds to approximately 10% of the total SCF time for 32 nodes. For a very large number of nodes a parallel diagonalization therefore becomes more and more favorable, but is not a very important factor for the present case. As mentioned before, for this large active space the second half-transformation becomes the limiting step, whereas the local first HT scales linearly. This second HT bottleneck will flatten the overall scaling curve with respect to computer time. On the other hand, a problem that could not have been managed on one node now becomes easily tractable, because the file sizes shrink according to the number of nodes. ROTRAN creates only one intermediate file INTTWO

### Table II

<table>
<thead>
<tr>
<th>Step/No. of Nodes</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELTWEL</td>
<td>6351</td>
<td>3183</td>
<td>1590</td>
<td>797</td>
<td>407</td>
<td>229</td>
</tr>
<tr>
<td>MFDSDF</td>
<td>15376</td>
<td>8149</td>
<td>4541</td>
<td>2547</td>
<td>2182</td>
<td>1453</td>
</tr>
<tr>
<td>ROTRAN 1. HT</td>
<td>24851</td>
<td>12032</td>
<td>6160</td>
<td>3107</td>
<td>1521</td>
<td>760</td>
</tr>
<tr>
<td>ROTRAN 2. HT</td>
<td>11901</td>
<td>11113</td>
<td>10923</td>
<td>10688</td>
<td>10767</td>
<td>10909</td>
</tr>
<tr>
<td>Total</td>
<td>60782</td>
<td>36780</td>
<td>25517</td>
<td>19442</td>
<td>17180</td>
<td>15654</td>
</tr>
</tbody>
</table>

Due to synchronization steps occurring in the different modules, the above times are maximal running times. One clearly sees that in the case of large active spaces the second HT becomes the limiting factor in computer time, but the necessary disk spaces always scales down with increasing number of processors. The symmetrization in each case took 2303 s, and is always included in the total time.
TABLE III.
Wall Clock Time in Seconds for the Small Active Space per Module.

<table>
<thead>
<tr>
<th>Step/No. of Nodes</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELTWEL</td>
<td>6351</td>
<td>3183</td>
<td>1590</td>
<td>797</td>
<td>407</td>
<td>229</td>
</tr>
<tr>
<td>MFSCF</td>
<td>15376</td>
<td>8149</td>
<td>4541</td>
<td>2547</td>
<td>2182</td>
<td>1453</td>
</tr>
<tr>
<td>ROTRAN 1. HT</td>
<td>3351</td>
<td>1521</td>
<td>753</td>
<td>397</td>
<td>186</td>
<td>103</td>
</tr>
<tr>
<td>ROTRAN 2. HT</td>
<td>134</td>
<td>115</td>
<td>220</td>
<td>100</td>
<td>145</td>
<td>144</td>
</tr>
<tr>
<td>Total</td>
<td>25214</td>
<td>12970</td>
<td>7106</td>
<td>3903</td>
<td>2922</td>
<td>1931</td>
</tr>
</tbody>
</table>

Now it is evident that the AO basis set size is the limiting factor, and the scaling is optimal. In this case, the symmetrization step took 2 s, and is included in the total run times.

organized as a direct access file. This file size is controlled by the range of the symmetry-allowed MS labels and by the number of scalar basis functions. Table IV shows that the integral file size scales linearly, but the INTTWO file size decreases slower until it reaches a minimum. This is due to the record structure of the file. Even if there are only a few integrals per MS spinor index, a record is created and allocated for each index. In practice, this is not a serious problem because the file size is governed by two different space dimensions and we can always reduce the record length to account for smaller file sizes. Also, for large active spaces one benefits from the split file organization, and this makes larger calculations manageable.

In the case of a small active space, the size of the scalar basis becomes the limiting factor, and the whole transformation now scales approximately linearly, as can be seen in Table III. Here lies an important application of this implementation that is targeted towards calculations of localized states in solids within the embedded cluster approximation. The upcoming version will include the parallelization of the post-Hartree–Fock code, with an additional splitting of the active MS spinor space. This means that each node exclusively transforms MS integrals of a certain nonoverlapping range. These distinct MS integral ranges then are to be fed into the coupled-cluster program. For this case, the scaling can be predicted to behave linearly in the second half-transformation as well.

Acknowledgments

This research was performed in part using the Molecular Science Computing Facility (MSCF) in the William R. Wiley Environmental Molecular Sciences Laboratory at the Pacific Northwest National Laboratory. The MSCF is funded by the Office of Biological and Environmental Reserach in the U.S. Department of Energy. Pacific Northwest is operated by Battelle for the U.S. Department of Energy under contract DE-AC06-76RLO 1830. The project has benefited from allocation of computation time on the VU beta-cluster, which is part of the Distributed ASCI Supercomputer.

References