The integral-direct coupled cluster singles and doubles model

Henrik Koch[a)] and Alfredo Sánchez de Merás
Department of Physical Chemistry, Valencia University, E-46100 Burjassot, Spain

Trygve Helgaker
Department of Chemistry, University of Oslo, N-0315 Oslo, Norway

Ove Christiansen
Department of Chemistry, Aarhus University, DK-8000 Aarhus C, Denmark

(Received 18 October 1995; accepted 29 November 1995)

An efficient and highly vectorized implementation of the coupled cluster singles and doubles (CCSD) model using a direct atomic integral technique is presented. The minimal number of n^6 processes has been implemented for the most time consuming terms and point group symmetry is used to further reduce operation counts and memory requirements. The significantly increased application range of the CCSD method is illustrated with sample calculations on several systems with more than 500 basis functions. Furthermore, we present the basic trends of an open ended algorithm and discuss the use of integral prescreening. © *1996 American Institute of Physics.* [S0021-9606(96)00310-0]

I. INTRODUCTION

During the last decade the coupled cluster approach[1] and especially the coupled cluster singles and doubles model (CCSD) has proven a valuable method in electronic structure calculations of molecular properties. Since the first formulation and implementation of the CCSD model by Purvis and Bartlett[2] in 1982 several different implementations have been presented.[3–6] Mainly because of the rapid increase in CPU speed of modern supercomputers and workstations CCSD calculations with more than 200 basis functions can now be carried out routinely, even without use of point group symmetry. Nevertheless, there is a major problem with the previous implementations as they require a complete or partial integral transformation of the two-electron integrals and for large calculations with more than 300 basis functions an enormous amount of disk space is required. This has put some serious limitations on the size of molecular systems that can be studied with the conventional CCSD model and a direct treatment of the two-electron integrals is needed. A large scale application of the conventional CCSD model was recently carried out by Taylor *et al.*[7]

With the recently implemented integral-direct CCSD model by Koch *et al.*,[8] the application range of the method has increased significantly and it is now feasible to carry out CCSD calculations with more than 300 basis functions. The key idea in the first implementation is to calculate atomic orbital (AO) integral distributions when they are needed in the construction of the coupled cluster vector function. In this approach Koch *et al.*[8] did not exploit point group symmetry and the above mentioned AO integral distributions, which have one fixed AO index and three free AO indices, were completely transformed to the molecular orbital (MO) basis. Furthermore, the minimum operation count in the so-called *B* term was not implemented and this together with the MO transformation of the AO distributions turned out to be the most time-consuming part of the calculation.

In this paper we present an improved implementation of the integral-direct CCSD model where point group symmetry has been exploited, only partial MO transformations of AO integral distributions are required, the minimal operation count in the *B* term has been implemented, a completely new integral code for calculating the AO integral distributions has been developed and an improved overall vectorization is obtained. With these modifications it is now possible to carry out CCSD calculations with more than 500 basis functions one of the goals put forward in a recent publication.[9] Rendell and Lee[9] have taken the approach of approximating the two-electron integrals in order to reduce the disk space requirements of the CCSD and CCSD(T) methods. However, approximating the two-electron integrals using the *V* approximation[10] gives no reduction in the n^6 or n^7 processes and these will eventually dominate the calculations for large systems. In the present work, we have chosen the approach of keeping the Hamiltonian exact and recalculating the AO integrals when needed. We expect that combined use of integral prescreening in both the integral calculation and the direct CCSD part will give important reductions in the computational effort required by the integral-direct CCSD model.

The increased application range of the CCSD model must be extended beyond the wave function determination in order to prove useful. The analytical calculation of molecular properties must be implemented using integral-direct techniques, as the CCSD approach has proven to be successful in calculating several molecular properties in the framework of the response function formalism.[11] Particularly excitation energies[12] and transition matrix elements[13] have been computed, as well as frequency-dependent polarizabilities[14] and magnetic shielding tensors.[15] We have already implemented the calculation of the coupled cluster linear response (CCLR) excitation energies[16] directly from the AO integral distributions. Furthermore, it will also become important to include

[a)] Permanent address: Department of Chemistry, Aarhus University, DK-8000 Aarhus C, Denmark.

Downloaded¬29¬Jan¬2010¬to¬147.156.182.23.¬Redistribution¬subject¬to¬AIP¬license¬or¬copyright;¬see¬http://jcp.aip.org/jcp/copyright.jsp

the effect of connected triples excitations[17] in order to get reliable accuracy in chemical predictions. Integral-direct techniques are now also used in the coupled cluster based $R12$ methods.[18]

This paper is organized as follows. In Sec. II we introduce the necessary notation and present some fundamental aspects of the CCSD equations in the MO basis. In Sec. III we discuss the integral-direct implementation of the coupled cluster singles and doubles model and in Sec. IV we introduce an alternative algorithm intended for massively parallel systems. The new integral program is described in Sec. V together with a discussion of integral prescreening. Section VI contains the results of some sample calculations and a performance analysis of the code. Finally, our concluding remarks are given in Sec. VII.

II. THE CCSD FORMALISM

The coupled cluster singles and doubles (CCSD) wave function for a closed shell system is given by the *ansatz*

$$|\mathrm{CC}\rangle = \exp(T)|\mathrm{HF}\rangle, \tag{1}$$

where the cluster operator T is

$$T = T_1 + T_2 \tag{2}$$

with the connected singles $\{t_i^a\}$ and doubles $\{t_{ij}^{ab}\}$ cluster amplitudes entering the cluster operator as

$$T_1 = \sum_{ai} t_i^a E_{ai}, \tag{3}$$

$$T_2 = \frac{1}{2} \sum_{aibj} t_{ij}^{ab} E_{ai} E_{bj}. \tag{4}$$

The operators $\{E_{ai}\}$ are the unitary group generators and indices $ijkl$ and $abcd$ label occupied and unoccupied orbitals in the Hartree–Fock reference state $|\mathrm{HF}\rangle$. The cluster amplitudes are determined by solving the coupled cluster equations

$$\Omega_\mu = \langle \mu | \exp(-T) H \exp(T) | \mathrm{HF}\rangle = 0, \tag{5}$$

where Ω_μ is the so-called coupled cluster vector function. In Eq. (5) H is the electronic Hamiltonian

$$H = \sum_{pq} h_{pq} E_{pq} + \frac{1}{2} \sum_{pqrs} (pq|rs)(E_{pq}E_{rs} - \delta_{qr}E_{ps}) \tag{6}$$

with $pqrs$ labeling general orbital indices. The projection manifold $\{\langle\mu|\}$ is the set of single and double excited determinants with respect to $|\mathrm{HF}\rangle$

$$\{\langle \mu_1 |\} = \{\langle \mathrm{HF}| E_{ia}\}, \tag{7}$$

$$\{\langle \mu_2 |\} = \{\langle \mathrm{HF}|(2E_{ia}E_{jb} + E_{ja}E_{ib})/3 | ai \geqslant bj\}, \tag{8}$$

where we have used the biorthogonal basis for the double excitation manifold.[19] Introducing the transformed Hamiltonian

$$\hat{H} = \exp(-T_1) H \exp(T_1), \tag{9}$$

the coupled cluster vector function may be written as

$$\Omega_\mu = \langle \mu | \exp(-T_2) \hat{H} \exp(T_2) | \mathrm{HF}\rangle. \tag{10}$$

In this way the CCSD equations transform into coupled cluster doubles (CCD) equations with modified integrals and thereby offer the possibility of a compact and efficient implementation. These modified integrals may be obtained from the transformation of the creation and annihilation operators in the Hamiltonian in Eq. (9) and this leads to[8]

$$\hat{H} = \sum_{pq} \hat{h}_{pq} E_{pq} + \frac{1}{2} \sum_{pqrs} (pq\hat{|}rs)(E_{pq}E_{rs} - \delta_{qr}E_{ps}), \tag{11}$$

which is expressed in terms of the modified one- and two-electron integrals defined as

$$\hat{h}_{pq} = \sum_{\alpha\beta} h_{\alpha\beta} \Lambda^p_{\alpha p} \Lambda^h_{\beta q}, \tag{12}$$

$$(pq\hat{|}rs) = \sum_{\alpha\beta\gamma\delta} (\alpha\beta|\gamma\delta) \Lambda^p_{\alpha p} \Lambda^h_{\beta q} \Lambda^p_{\gamma r} \Lambda^h_{\delta s}. \tag{13}$$

In Eqs. (12) and (13) $\alpha\beta\gamma\delta$ label atomic orbitals and we have introduced the transformation matrices $\mathbf{\Lambda}^p$ and $\mathbf{\Lambda}^h$ for particle and hole operators, respectively,

$$\mathbf{\Lambda}^p = \mathbf{C}(\mathbf{I} - \mathbf{t}_1^T), \tag{14}$$

$$\mathbf{\Lambda}^h = \mathbf{C}(\mathbf{I} + \mathbf{t}_1), \tag{15}$$

where we have used the auxiliary matrix

$$\mathbf{t}_1 = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \{t_{ai}\} & \mathbf{0} \end{pmatrix} \tag{16}$$

and expanded the molecular orbitals $\{\phi_p\}$ in terms of the atomic orbitals $\{\chi_\alpha\}$

$$\phi_p = \sum_\alpha \chi_\alpha C_{\alpha p}. \tag{17}$$

After some manipulations and using the transformed Hamiltonian in Eq. (11), the double excitation part of the CCSD vector function may be written in the molecular orbital basis as[8]

$$\Omega_{ij}^{ab} = \Omega^A_{aibj} + \Omega^B_{aibj} + 2P_{ij}^{ab}\{\Omega^C_{aibj} + \Omega^D_{aibj} + \Omega^E_{aibj}\} + \Omega^F_{aibj}, \tag{18}$$

where the permutation operator P_{ij}^{ab} is given by

$$P_{ij}^{ab}\begin{pmatrix} a & b \\ i & j \end{pmatrix} = \begin{pmatrix} a & b \\ i & j \end{pmatrix} + \begin{pmatrix} b & a \\ j & i \end{pmatrix}. \tag{19}$$

The different contributions to the Ω vector in Eq. (18) are

$$\Omega^A_{aibj} = 2\sum_{kl} t_{kl}^{ab} \left((ki\hat{|}lj) + \sum_{cd} t_{ij}^{cd} (kc|ld) \right), \tag{20}$$

$$\Omega^B_{aibj} = 2\sum_{cd} t_{ij}^{cd} (ac\hat{|}bd), \tag{21}$$

Downloaded¬29¬Jan¬2010¬to¬147.156.182.23.¬Redistribution¬subject¬to¬AIP¬license¬or¬copyright;¬see¬http://jcp.aip.org/jcp/copyright.jsp

$$\Omega^{C}_{aibj}=-\frac{1}{2}\sum_{ck}t^{cb}_{jk}\left((ki\hat{|}ac)-\frac{1}{2}\sum_{dl}t^{ad}_{li}(kd|lc)\right.$$

$$\left.-\sum_{ck}t^{cb}_{ik}\left((kj\hat{|}ac)-\frac{1}{2}\sum_{dl}t^{ad}_{lj}(kd|lc)\right)\right), \tag{22}$$

$$\Omega^{D}_{aibj}=\frac{1}{2}\sum_{ck}(2t^{cb}_{jk}-t^{cb}_{kj})$$

$$\times\left(\hat{L}_{aikc}-\frac{1}{2}\sum_{dl}(2t^{ad}_{il}-t^{ad}_{li})L_{ldkc}\right), \tag{23}$$

$$\Omega^{E}_{aibj}=\sum_{c}t^{ac}_{ij}\left(\hat{F}_{bc}-\sum_{dlm}t^{db}_{lm}L_{ldmc}\right)$$

$$-\sum_{k}t^{ab}_{ik}\left(\hat{F}_{kj}+\sum_{dem}t^{de}_{jm}L_{mekd}\right), \tag{24}$$

$$\Omega^{F}_{aibj}=2(ai\hat{|}bj), \tag{25}$$

where the inactive Fock matrix with transformed integrals has been introduced,

$$\hat{F}_{pq}=\hat{h}_{pq}+\sum_{k}\hat{L}_{kkpq}, \tag{26}$$

together with the modified integral

$$\hat{L}_{pqrs}=2(pq\hat{|}rs)-(ps\hat{|}rq). \tag{27}$$

Similarly, the single excitation part of the vector function may be written as

$$\Omega_{ai}=\Omega^{G}_{ai}+\Omega^{H}_{ai}+\Omega^{I}_{ai}+\Omega^{J}_{ai} \tag{28}$$

with

$$\Omega^{G}_{ai}=2\sum_{cdk}t^{cd}_{ik}\hat{L}_{kdac}, \tag{29}$$

$$\Omega^{H}_{ai}=-2\sum_{dkl}t^{ad}_{kl}\hat{L}_{ldki}, \tag{30}$$

$$\Omega^{I}_{ai}=2\sum_{ck}(2t^{ac}_{ik}-t^{ac}_{ki})\hat{F}_{kc}, \tag{31}$$

$$\Omega^{J}_{ai}=2\hat{F}_{ai}. \tag{32}$$

When exploring the construction of the CCSD vector function by direct use of AO integrals we encounter the complications caused by the MO integrals entering the equations, as it is not evident how the calculation should be organized. Nevertheless, by inspection of Eqs. (18)–(32) we observe that each term may be evaluated from integrals with one or more atomic indices. We therefore reformulate the equations using integrals with three MO indices and one AO index and in this way a suitable algorithm may be designed. This was the basic idea in the first implementation.[8] In the following sections we describe improved algorithms where we in each term treat as many indices in the AO basis as possible.

III. IMPLEMENTATION

In this section we describe in detail the algorithms used in the computer code developed for the integral-direct calculation of the CCSD vector function in Eqs. (18) and (28). It is significant at this point to notice that all the equations presented in the previous and following Sections also apply to the Brueckner coupled cluster doubles model as only the Λ matrices need to be changed and thus the same implementation may be used.

Maximum performance on a super computer like the Cray C90 can only be obtained if the algorithms in question are based on large matrix multiplications. This may naively be achieved by keeping in main memory all relevant amplitude and result vectors sorted as they enter the matrix multiplications. However, this would imply some strong limitations on the size of calculations that could be carried out. We therefore make the following compromise, and in the case where enough main memory is available, options to use different algorithms are also implemented.

The basic assumption is that three vectors of the same length as the symmetry-packed double excitation amplitudes $(t^{ab}_{ij}, ai\geqslant bj)$ can be kept in main memory. Additionally, we allocate an array for one symmetry-packed AO integral distribution as well as some work arrays of shorter length. With this memory allocation we can permanently keep the symmetry-packed $(\Omega^{ab}_{ij}, ai\geqslant bj)$ result vector and the full square of double excitation amplitudes $(t_{ai,bj})$ in main memory. When more memory is available we have implemented the option to keep the result vector squared and/or the full square of double excitation amplitudes with the occupied indices transposed.

A second important aspect to be considered is the way in which the new integral program is integrated into our direct CCSD code. We drive the calculation with an outer loop over the atomic shells in the molecule and the integral code has been designed to calculate an AO integral distribution for a given AO index δ

$$I^{\delta}_{\alpha\beta,\gamma}=(\alpha\beta|\gamma\delta), \tag{33}$$

where $(\alpha\geqslant\beta)$. This implies that the AO integrals are reevaluated four times in each iteration compared to direct self-consistent field (SCF). In order to save time in the integral evaluation we calculate all the δ distributions that belong to the same shell in the molecule and these are written to disk and subsequently read back in one at a time as needed. The γ index is transformed to the occupied space after the B and F terms and the inactive Fock matrix have been calculated

$$I^{\delta}_{\alpha\beta,k}=\sum_{\gamma}I^{\delta}_{\alpha\beta,\gamma}\Lambda^{p}_{\gamma k}. \tag{34}$$

This transformation is carried out in a matrix multiplication over the compound index $\alpha\beta$. Furthermore, for each δ one virtual index of the double excitation amplitudes is transformed to the contravariant AO basis

$$N^{\delta}_{ci,j}=\sum_{d}t_{ci,dj}\Lambda^{h}_{\delta d} \tag{35}$$

Downloaded¬29¬Jan¬2010¬to¬147.156.182.23.¬Redistribution¬subject¬to¬AIP¬license¬or¬copyright;¬see¬http://jcp.aip.org/jcp/copyright.jsp

and this matrix is only needed for one δ at a time.

Having described the outer loop structure we now turn to the individual terms in the coupled cluster vector function. For the sake of conciseness, we only discuss the dominant terms in Eqs. (18) and (28). Initially we consider the A term given in Eq. (20) and write this as

$$\Omega^A_{aibj}=2\sum_{kl} t_{ak,bl}\Gamma_{ki,lj}, \tag{36}$$

where we have introduced the auxiliary matrix

$$\Gamma_{ki,lj}=\sum_{\delta}\left(I^{\delta}_{ki,l}\Lambda^{h}_{\delta j}+\sum_{c} N^{\delta}_{ci,j}I^{\delta}_{kc,l}\right)=\Gamma_{lj,ki}. \tag{37}$$

The Γ matrix may be kept in memory during the calculation of the integrals and for each δ we transform the remaining AO indices of the integrals in Eq. (34) to the appropriate MO indices. The calculation is driven with outer loops over l and j indices with a matrix multiplication of dimension VO^2 (V and O are the number of virtual and occupied orbitals, respectively, in the Hartree–Fock reference state) inside. Once the Γ matrix has been constructed we contract with the amplitudes according to Eq. (36). As we shall see later, the G matrix may be obtained more easily in a ($\mathrm{N^2O^3+NO^4}$) process, as a byproduct of our algorithm for the B term.

The B term is the computationally most demanding term in a CCSD calculation because it involves the contraction of integrals with four virtual indices and double excitation amplitudes. The minimal operations count[3] in the MO basis is $\frac{1}{4}V^4O^2$ and we here demonstrate that a similar operation count $\frac{1}{4}N^4O^2$ (N is the number of atomic orbitals) may be obtained using the AO integral distributions directly without transforming any indices to the MO basis. This is different from the first implementation[8] where two indices in the AO integral distribution were transformed to the virtual space and then contracted with double excitation amplitudes resulting in an operation count of $\frac{1}{2}N^2V^2O^2$.

When we wish to use the AO integrals directly in the accumulation of the different δ contributions, a vector with two AO and two occupied indices ($\Omega^B_{\alpha i,\beta j},\alpha i\geqslant\beta j$) must be used. After the integral calculation this vector is transformed to the MO basis to give the final contribution. We thus write the B term as

$$\Omega^B_{\alpha i,\beta j}=2\sum_{\delta\gamma} M^{\delta}_{\gamma,ij}(\alpha\gamma|\beta\delta), \tag{38}$$

where the auxiliary matrix is defined as

$$M^{\delta}_{\gamma,ij}=\sum_{c} N^{\delta}_{ci,j}\Lambda^{h}_{\gamma c}. \tag{39}$$

Restricting the summation in Eq. (38) to $\gamma\geqslant\delta$ would result in the operation count $\frac{1}{2}N^4O^2$. However, further reduction is achieved by introducing the matrices[3]

$$M^{\delta\pm}_{\gamma,ij}=M^{\delta}_{\gamma,ij}\pm M^{\delta}_{\gamma,ji}, \tag{40}$$

$$J^{\delta\pm}_{\alpha\beta,\gamma}=(I^{\delta}_{\alpha\gamma,\beta}\pm I^{\delta}_{\beta\gamma,\alpha})(1+\delta_{\gamma\delta})^{-1} \tag{41}$$

and write the B term as

$$\Omega^{B\pm}_{\alpha\beta,ij}=\frac{1}{2}\sum_{\gamma\geqslant\delta} J^{\delta\pm}_{\alpha\beta,\gamma}M^{\delta\pm}_{\gamma,ij} \tag{42}$$

where the indices now are restricted as $\alpha\geqslant\beta$, $\gamma\geqslant\delta$, and $i\geqslant j$.

For a given δ distribution we implement the calculation of the $\Omega^{B\pm}_{\alpha\beta,ij}$ vector with a batching over the γ index. In this way we only need to construct as many $J^{\delta\pm}_{\alpha\beta,\gamma}$ integrals as we can hold in the remaining part of memory and the matrix multiplication in Eq. (42) is carried out with the maximum dimension of $\frac{1}{8}N^3O^2$ for both the plus and minus vectors separately. Using this algorithm we obtain the minimal operation count in the AO basis previously mentioned with maximum vectorization over the compound indices $\alpha\beta$ and ij. When the plus and minus vectors have been constructed we transform to the MO basis

$$\Omega^B_{aibj}=\sum_{\alpha\beta}\{\Omega^{B+}_{\alpha\beta,ij}+\Omega^{B-}_{\alpha\beta,ij}\}\Lambda^{p}_{\alpha a}\Lambda^{p}_{\beta b}. \tag{43}$$

in a process that requires $2N^3O^2$ operations.

One should note that by modifying the definition of the auxiliary matrix $M^{\delta}_{\gamma,ij}$ in Eq. (39) as

$$M^{\delta}_{\gamma,ij}=\sum_{c}(N^{\delta}_{ci,j}\Lambda^{h}_{\gamma c})+\Lambda^{p}_{\gamma i}\Lambda^{p}_{\delta j} \tag{44}$$

the F term in Eq. (25) is automatically added through the subsequent matrix multiplication in Eq. (42) with only $\frac{1}{2}N^2O^2$ additional operations. Furthermore, when the B term is calculated with the modified matrix in Eq. (44) the Γ matrix in Eq. (37) may be expressed in terms of the $\Omega^B_{\alpha i\beta j}$ as

$$\Gamma_{ki,lj}=\sum_{\alpha\beta}\Lambda^{p}_{\alpha k}\Lambda^{p}_{\beta l}\Omega^{B}_{\alpha i\beta j}. \tag{45}$$

In this way, the Γ matrix is obtained in ($N^2O^3+NO^4$) operations compared to the NVO^4 operations required in the implementation discussed earlier. We have also implemented this simple reduction in the number of n^6 processes and thus the Γ matrix is not needed in main memory during the integral calculation.

The C and D terms have the same structure, only differing in the effective double excitation amplitudes and integrals entering the expressions. Thus both terms have the same operation count and are implemented in a similar way. Therefore we only discuss the evaluation of the C term which is carried out through the construction of an auxiliary matrix

$$P^{\delta}_{k,ai}=I^{\delta}_{ki,a}-\frac{1}{2}\sum_{dl} I^{\delta}_{k,dl}\hat{t}_{dl,ai} \tag{46}$$

for one δ at a time. The matrix $\hat{t}_{dl,ai}$ contains the double excitation amplitudes with the occupied indices transposed. Expressed in terms of $P^{\delta}_{k,ai}$ the C term may be written as

$$\Omega^C_{ai,bj}=-\frac{1}{2}\sum_{\delta k} N^{\delta}_{bk,j}P^{\delta}_{k,ai}-\sum_{\delta k} N^{\delta}_{bk,i}P^{\delta}_{k,aj}. \tag{47}$$

An implementation[8] of Eq. (47) shows that driving the calculation with an outer loop over δ with the summation over k in a matrix multiplication inside, gives insufficient perfor-

Downloaded¬29¬Jan¬2010¬to¬147.156.182.23.¬Redistribution¬subject¬to¬AIP¬license¬or¬copyright;¬see¬http://jcp.aip.org/jcp/copyright.jsp

mance because of the small number of floating point operations compared to memory references. We therefore choose a different algorithm where the $P^{\delta}_{k,ai}$ is written to disk for each δ and then read in a batched loop over the compound index ai after the integrals have been evaluated. For each batch we transform the δ index to the virtual space

$$\Delta_{ck,\#ai}=\sum_{\delta} P^{\delta}_{k,\#ai}\Lambda^{h}_{\delta c}, \tag{48}$$

where $\#ai$ denotes the number of ai indices handled in the batch. The final contraction is carried out in a matrix multiplication over the compound indices bj, ck, and $\#ai$

$$C_{bj,\#ai}=\sum_{ck} \hat{t}_{bj,ck}\Delta_{ck,\#ai}. \tag{49}$$

The temporary $C_{bj,\#ai}$ matrix is then added into the packed result vector in the MO basis with appropriate weights and this results in a total operation count of $(NV^2O^3+V^3O^3)$ for the C term.

We now turn our attention to the implementation of the E terms. Using the density

$$D_{\alpha\beta}=\sum_{k} \Lambda^{p}_{\alpha k}\Lambda^{h}_{\beta k}, \tag{50}$$

the inactive Fock matrix entering these terms is constructed in the AO basis from standard expressions and subsequently transformed to the MO basis. Furthermore, inside the δ loop we construct the auxiliary matrices

$$R_{b\delta}=\sum_{dlm} \tilde{t}_{dl,bm}I^{\delta}_{ld,m}, \tag{51}$$

$$S_{kj}=\sum_{\delta em} \tilde{N}^{\delta}_{em,j}I^{\delta}_{me,k}, \tag{52}$$

where we have used the notation

$$\tilde{t}_{dl,bm}=2t_{dl,bm}-t_{dm,bl}, \tag{53}$$

$$\tilde{N}^{\delta}_{em,j}=2N^{\delta}_{em,j}-N^{\delta}_{ej,m}. \tag{54}$$

Each δ contribution to the S_{kj} matrix is accumulated in one matrix multiplication once the $I^{\delta}_{me,k}$ integrals have been constructed. The contraction of effective amplitudes with appropriate integrals in the $R_{b\delta}$ matrix is driven with a matrix-vector multiplication of dimension V^2O inside a loop over index m and is transformed to the MO basis after the δ loop. The auxiliary matrices are then added to the Fock matrix and contracted with amplitudes according to Eq. (24).

Having discussed the Ω_{aibj} vector, we now focus on the G and H terms entering the Ω_{ai} vector. The calculation of the G term is facilitated through the construction of the auxiliary matrix

$$G^{\delta}_{\alpha i}=\sum_{\gamma} I^{\delta}_{\alpha\gamma,k}\tilde{M}^{\delta}_{\gamma i,k} \tag{55}$$

in a matrix multiplication of dimension N^2O inside a loop over the k index. In Eq. (55) we have introduced the matrix $\tilde{M}^{\delta}_{\gamma i,k}$ which results from the back transformation of the virtual index in matrix $\tilde{N}^{\delta}_{em,j}$ in Eq. (54) to the contravariant AO basis. For each δ we add the above contribution to the result vector

$$\Omega^{G}_{ai}=\sum_{\delta\alpha} \Lambda^{p}_{\alpha a}G^{\delta}_{\alpha i}, \tag{56}$$

and this procedure gives the total operation count $(N^3O^2+N^2VO)$.

The H term in Eq. (30) is implemented with a loop over the l index, where the effective amplitudes $\tilde{N}^{\delta}_{ak,l}$ are contracted with the corresponding integrals in a matrix multiplication

$$\Omega^{H}_{ai}=\sum_{\delta kl} \tilde{N}^{\delta}_{ak,l}I^{\delta}_{ki,l}, \tag{57}$$

of dimension NVO^3 once the $\alpha\beta$ indices of the integrals in Eq. (34) have been transformed to the occupied space.

The algorithm discussed earlier represents an efficient implementation of the CCSD model and good performance with respect to vectorization is obtained. The dominant operation count for the algorithm may be summarized as

$$\tfrac{1}{4}N^4O^2+2(NV^2O^3+V^3O^3)+\tfrac{1}{4}V^2O^4$$

for the terms scaling as n^6. In Sec. V we discuss these aspects in detail. At this point it is important to point out that although the disk space problem of the CCSD procedure has been eliminated, the memory requirements of the present algorithm now constitute the major bottleneck in our code. This calls for yet another algorithm where the double excitation amplitudes are stored on disk and read on request. In this sense we present, in Sec. IV an alternative algorithm with reduced memory requirements.

IV. AN ALTERNATIVE ALGORITHM

In order to motivate the necessity of an open ended algorithm let us consider the example of The Buckminster fullerene C_{60} using for instance a *cc*-pVDZ basis set.[20] There are more than 800 million double excitation amplitudes when correlating all 360 electrons using D_{2h} symmetry only and the evaluation of the B term alone will approximately require 1.3×10^{14} floating point operations or roughly 40 h of CPU time on a single processor Cray C90. This calculation is not easily carried out using today's computer technology, but the example shows that an algorithm is needed where all double excitations are not required in main memory. Furthermore, the algorithm must offer an efficient implementation on a massively parallel distributed memory computer to take advantage of the latest technological developments. In the following we briefly describe an algorithm where the memory requirement has been significantly reduced and which can be implemented on a parallel architecture.

The basic idea of the algorithm is to require that the integral program on request will return a set of AO integrals

$$I^{\gamma\delta}_{\alpha\beta}=(\alpha\gamma|\beta\delta)=I^{\alpha\beta}_{\gamma\delta} \tag{58}$$

for a fixed $\gamma\geqslant\delta$. The integral program may easily be modified to comply with this requirement if we simultaneously

Downloaded¬29¬Jan¬2010¬to¬147.156.182.23.¬Redistribution¬subject¬to¬AIP¬license¬or¬copyright;¬see¬http://jcp.aip.org/jcp/copyright.jsp

calculate all the integrals where γ and δ belong to the same shells in the molecule. The restriction on the γ and δ indices may trivially be handled noting that

$$I_{\alpha\beta}^{\delta\gamma}=I_{\beta\alpha}^{\gamma\delta} \tag{59}$$

and thus only requires a transposition of the integral matrix. We drive the calculation with two batched loops over δ and γ indices in order to determine the fraction of the auxiliary matrices that may be kept in main memory. This will also dictate how many times the $N_{al,i}^{\gamma}$ and $M_{\gamma,ij}^{\delta}$ matrices should be read from disk as these matrices enter the expressions of the auxiliary matrices that have to be calculated inside the γ and δ loops.

To describe the algorithm some of the expressions given in the preceding section must be rewritten in a suitable way. The B and F terms together with the Γ matrix entering the A term should be calculated according to

$$\Omega_{\alpha i,\beta j}^{BF}=\sum_{\delta\gamma} (\Lambda_{\gamma i}^{h}\Lambda_{\delta j}^{h}+M_{\gamma,ij}^{\delta})I_{\alpha\beta}^{\gamma\delta} \tag{60}$$

and Eq. (45). We need also to modify the way in which the $P_{k,ai}^{\delta}$ matrix in the C term is calculated

$$P_{k,ai}^{\delta}=\sum_{\beta\gamma}\left(\Lambda_{\beta a}^{p}\Lambda_{\gamma i}^{h}-\frac{1}{2}\sum_{l} N_{al,i}^{\gamma}\Lambda_{\beta l}^{p}\right)I_{k\beta}^{\gamma\delta} \tag{61}$$

as well as the equivalent $Q_{k,ai}^{\delta}$ matrix used in the D term

$$Q_{k,ai}^{\delta}=\sum_{\beta\gamma}\left(\Lambda_{\beta a}^{p}\Lambda_{\gamma i}^{h}+\frac{1}{2}\sum_{l} \tilde{N}_{ai,l}^{\gamma}\Lambda_{\beta l}^{p}\right)L_{\beta\gamma k\delta}. \tag{62}$$

The auxiliary matrices entering the E term must also be reformulated as indicated

$$R_{b\gamma}=\sum_{\delta lm} \tilde{N}_{bm,l}^{\delta}I_{ml}^{\gamma\delta}, \tag{63}$$

$$S_{kj}=\sum_{\delta\gamma m} \tilde{M}_{\gamma,mj}^{\delta}I_{mk}^{\gamma\delta}, \tag{64}$$

and the contributions to the single excitation part of the vector function take the following form

$$\Omega_{\alpha i}^{G}=2\sum_{\delta\gamma k} \tilde{M}_{\gamma,ik}^{\delta}I_{\alpha k}^{\gamma\delta}, \tag{65}$$

$$\Omega_{a\gamma}^{H}=-2\sum_{\delta kl} \tilde{N}_{ak,l}^{\delta}I_{kl}^{\gamma\delta}, \tag{66}$$

where we accumulate into different vectors which are finally transformed to the MO basis and added together. After the construction of the auxiliary matrices the contraction with the appropriate amplitudes may be done using the same technique as described in the preceding section keeping in mind that the back transformed double excitation amplitudes are read from disk in a batched loop.

The aforementioned algorithm is intended for super computers like the Cray C90 or workstations with less main memory, but it will not vectorize as well as the algorithm in Sec. III. This is due to the fact that vectors entering matrix multiplications are shorter as only parts of the amplitudes are read from disk with their size depending on the available memory. When we seek an implementation on a massively parallel distributed memory computer the integral program must have the flexibility to return on request a set of integrals

$$I_{AB}^{CD}=(AC|BD), \tag{67}$$

where $ABCD$ denote shells in the molecule. This set of integrals may be discarded using prescreening techniques to give a reduction in both the integral calculation and in the contraction process. Several different algorithms can now be designed and we may, for instance, choose an algorithm with minimal communication where the computational requirements will depend on the amount of memory on a single processor, but with inherent good scaling to a large number of processors. On the other hand, we may also choose an algorithm with minimal computation, but the amount of communication will then depend on the number of processors and total amount of memory on the system. The actual choice of algorithm will depend on the computer system in question. We have undertaken the work of developing a computer code based on the above described algorithms and will report our observations in a future publication.

V. INTEGRAL EVALUATION AND PRESCREENING

In our present direct CCSD code, the integrals are calculated using the McMurchie–Davidson scheme. The present implementation represents a great improvement on the HERMIT (Ref. 21) code, in particular, for vector machines. The integral evaluation is initiated by a sorting of all AO overlap distributions. The integrals are subsequently calculated batchwise, treating as many overlap distributions of the same sort as possible in each batch. This leads to long vector lengths, greatly improving the performance of the code on most computer systems. Thus, a speedup of a factor of 10 or more has been observed in many applications.

In our implementation, the two-electron AO integrals are transformed to the symmetry orbital basis before each integral distribution is written to disk. In this respect, our direct CCSD scheme differs from direct SCF codes, which work in terms of nonsymmetrized symmetry unique integrals. Symmetrization of integrals is necessary for the efficient implementation of the CCSD algorithms discussed earlier, although a skeleton-type approach probably may be developed.

The present code cannot generally handle contracted basis sets except by duplication of the primitive functions. This restriction, which reduces the efficiency of the code for the generally contracted correlation-consistent basis sets considerably, will be lifted in the future and at that time we will publish the details about implementation and performance.

Of particular concern is the interface between the integral and the wave function parts of the CCSD code. A poorly designed interface can severely degrade the overall performance of the code. Ideally, the integral code should be able to return on the request all integrals belonging to a single AO index δ. In practice, such a scheme is inefficient since mod-

Downloaded¬29¬Jan¬2010¬to¬147.156.182.23.¬Redistribution¬subject¬to¬AIP¬license¬or¬copyright;¬see¬http://jcp.aip.org/jcp/copyright.jsp

TABLE I. Molecular geometries used in the calculations where the Cartesian coordinates are given in a.u.

	X	*Y*	*Z*
		C_2H_4	
H	2.328 289 6	1.755 415 6	0.0
O	1.265 172 1	0.0	0.0
		$C_{10}H_8$	
H	2.372 750	4.688 138	0.0
H	6.337 613	2.353 840	0.0
C	0.0	1.332 257	0.0
C	2.349 894	2.647 362	0.0
C	4.565 554	1.341 360	0.0
		HFCO	
H	1.646 033 47	0.0	−1.253 941 47
F	−2.125 333 46	0.0	−1.369 683 95
C	0.0	0.0	0.0
O	0.0	0.0	2.231 766 39

ern integral techniques invariably generate integrals in batches with several related indices δ at a time. For example, indices δ related by point group symmetry, belonging to the same shell or to the same contraction group are best calculated and returned simultaneously. Thus, in our implementation, the integral code returns batches of distributions, where all indices δ that are related by point group symmetry, belong to the same shell or originate from the same contraction group are returned simultaneously. The number of batches of distributions can be relatively large—ten to twenty—and all integrals are therefore written to disk before being read back in on request by the CCSD part of the code. To reduce IO, the integrals are therefore sorted before they are written to a random access device, each record containing only integrals with the same index δ. When requested by the CCSD code, only records containing integrals with the requested index δ are read in.

Our present implementation of the direct CCSD method does not exploit the fact that for extended systems many integrals make a negligible contribution to the wave function. Thus, for a given fixed integral index δ we require—for the sake of efficient vectorization—all integrals of the form $(\alpha\beta|\gamma\delta)$, irrespective of the magnitude of the individual integrals. This constitutes a deficiency of our current implementation since in direct SCF calculations major savings are usually gained by referencing only those integrals that are greater than some chosen threshold. We would like to emphasize that the integral part of the current CCSD code calculates only those integrals that are larger than a given threshold. It is only in the wave function part of the code that small integrals (in practice zero integrals) are not discarded.

It is possible, however, to develop a direct CCSD method that exploits the smallness of many integrals in extended molecular systems. Thus, for a given distribution δ, many AO overlap distributions $\gamma\delta$ are sufficiently small to eliminate all integrals of the form $(\alpha\beta|\gamma\delta)$ from the calculation. Such a systematic sparseness is easily incorporated in the present CCSD scheme by a simple reordering of indices γ for each distribution δ. Further reductions are also possible in cases where all integrals of the form $(\alpha\beta|\gamma\delta)$ vanish for a given distribution δ and a given index α (or a given index β) belonging to the other electron, although such cases should be less common.

TABLE II. The SCF, MP2, and CCSD total energies in E_h for ethylene, naphtalene, and formalfluoride using various correlation consistent basis sets.

Basis set	No. of functions	SCF	MP2	CCSD
		C_2H_4		
cc-pVDZ	48	−78.039 716	−78.320 075	−78.349 900
cc-pVTZ	116	−78.063 239	−78.429 612	−78.455 007
cc-pVQZ	230	−78.068 509	−78.480 798	−78.503 397
cc-pV5Z	402	−78.069 850	−78.501 380	−78.521 824
cc-pV5Z(*uc*)[a]	448	−78.069 878	−78.539 217	−78.551 998
		$C_{10}H_8$		
cc-pVDZ	180	−383.384 688	−384.702 254	−384.750 619
cc-pVTZ	412	−383.476 997	−385.200 880	−385.230 742
cc-pVTZ(*uc*)[a]	548	−383.478 603	−385.493 962	−385.536 925
		HFCO		
cc-pVDZ	47	−212.762 402	−213.266 820	−213.275 461
cc-pVTZ	104	−212.831 991	−213.511 235	−213.513 221
cc-pVQZ	195	−212.849 019	−213.628 126	−213.627 068
cc-pV5Z	328	−212.853 357	−213.673 046	−213.668 007

[a] Completely uncontracted basis set.

V. SAMPLE CALCULATIONS

The implemented algorithm has been tested on several molecular systems. In the following we analyze the performance of the code for calculations using formalfluoride, ethylene, and naphthalene as sample molecules. All the calculations have been made at the experimental geometry and the symmetry-independent coordinates are reported in Table I. The energies obtained at different levels of approximation and using several correlation consistent basis sets[20] are presented in Table II.

In Table III we report the timings of the 328 basis functions formalfluoride calculation and compared to those published together with the first implementation of the integral-direct CCSD model.[8] As seen from Table III, a substantial

TABLE III. Timings (in seconds) for the construction of the different contributions to the CCSD vector function in one iteration of the 328 basis functions HFCO calculation.

	Old implementation	New implementation	Ratio (Old/New)
A	7	6	1
B+*F*	7881	346	21
C	2190	350	6
D	1052	354	3
E	65	33	2
G	780	8	97
H	51	5	10
I	0	0	1
J	0	0	1
Total	16581	1429	12

Downloaded¬29¬Jan¬2010¬to¬147.156.182.23.¬Redistribution¬subject¬to¬AIP¬license¬or¬copyright;¬see¬http://jcp.aip.org/jcp/copyright.jsp

reduction in CPU time is obtained when using the new implementation, due to the use of symmetry and the improvements made in the construction of the B term. The savings obtained by eliminating the transformation of one atomic index to the virtual space are also significant. The theoretical reduction in CPU time due to the C_s symmetry of formalfluoride is a factor of 4, but we obtain an additional factor between 2 and 3 which is ascribed improved algorithms and implementation. However, we must emphasize that the symmetry-adapted version vectorizes less efficiently than the nonsymmetry one because of symmetry shortness in the matrix multiplications.

Having compared the efficiency of the old implementation and the present one, now we demonstrate the capabilities of the new algorithm. Medium sized molecular systems may be studied using very large basis sets and larger systems may be studied using medium-sized basis sets.

As an example of medium-sized systems, we have considered ethylene and carried out several CCSD calculations using the correlation consistent basis sets[20] of Dunning ranging from cc-pVDZ to cc-pV5Z. Even for the cc-pV5Z basis set the total energy is not fully converged and when uncontracting the basis set we observe a change in total energy of $30mE_h$. This is clearly indicating that special attention to core–core and core-valence correlation is required. From the results in Table II we observe that the most significant changes in energy is found going from cc-pVDZ to cc-pVTZ. Thus we conclude that the cc-pVTZ basis set is a reasonable compromise on basis sets in correlated calculations. As previously observed,[8] it is important to mention that basis limit is first reached at the SCF method, when correlated approaches are still far from basis set saturation.

We have also studied naphthalene using cc-pVDZ and cc-pVTZ basis sets. As pointed out above, cc-pVTZ is the smallest basis that should be used in correlated calculations. Here we demonstrate that it is now feasible to reach this level of description for large systems as naphthalene. Thus, with the new computer code we may carry out CCSD calculations using more than 500 basis functions. Anyway, we must emphasize that the cc-pVTZ basis set is still far from the basis set limit, even for the SCF level of approximation. The CCSD calculation in the uncontracted basis also shows the possibility of treating systems as anthracene using the cc-pVTZ basis, as when freezing the 1s core electrons the CPU requirements are about the same as the all electrons naphthalene calculation using 548 basis functions.

VI. CONCLUDING REMARKS

In this paper we have presented an improved implementation of the previously introduced integral-direct coupled cluster singles and doubles model of Koch *et al.*[8] Two main aspects in the developed computer algorithm should be noticed. First, point group symmetry has been used in the entire program in order to minimize the number of floating point operations and memory requirements. Second, we have treated as many indices in the AO basis as possible, reducing in this way the computational cost of transforming the AO integral distributions to the MO basis. Furthermore, the minimal operation count for the so-called B term has also been implemented treating all the integrals in the AO basis. In addition, we have achieved an additional reduction in the number of n^6 processes in the construction of the A term in the AO basis.

In the implemented algorithm we require the CCSD vector function and the full square of double excitation amplitudes to be kept in main memory, as well as one integral distribution for a fixed atomic index. This requirement introduces a memory limitation on the size of the systems that can be studied. To eliminate as much as possible this limitation, we have outlined an alternative algorithm with a smaller memory requirement, making CCSD calculations feasible for large systems on moderately sized workstations. Furthermore, this new algorithm is expected to allow for efficient parallelization on a parallel system if the integral program on request can provide a set of integrals $(AB|CD)$ where the capitalized indices run over the different atomic shells in the molecule under study. It will be fairly easy to modify the new integral program accordingly.

The developed code represents a highly vectorized and efficient implementation of the CCSD model and has been tested in several calculations on the naphtalene molecule. The obtained results demonstrate that CCSD calculations with more than 500 basis functions are now possible. We believe that the parallel implementation of the alternative algorithm will move the limits for CCSD calculations to more than a thousand basis functions.

Finally, we would like to draw your attention to a recent CCSD calculation[17] on the ferrocene molecule correlating all 96 electrons. This calculation, containing 373 basis functions, is an example of chemical applications now feasible using the new implementation presented here.

ACKNOWLEDGMENTS

We would like to thank Poul Jørgensen for helpful suggestions and discussions. H.K. acknowledges Generalitat Valenciana for a grant. Part of this work was supported by the Spanish CICYT project PB94-0993.

[1] F. Coester, Nucl. Phys. **7**, 421 (1958); F. Coester and H. Kümmel, *ibid*. **17**, 477 (1960); J. Cizek, J. Chem. Phys. **45**, 4256 (1966); J. Cizek and J. Paldus, Int. J. Quantum Chem. **5**, 359 (1971).
[2] G. D. Purvis and R. J. Bartlett, J. Chem. Phys. **76**, 1910 (1982).
[3] T. J. Lee and J. E. Rice, Chem. Phys. Lett. **150**, 406 (1988).
[4] G. E. Scuseria, C. L. Janssen, and H. F. Schaefer III, J. Chem. Phys. **89**, 7382 (1988).
[5] J. F. Stanton, J. Gauss, J. D. Watts, and R. J. Bartlett, J. Chem. Phys. **94**, 4334 (1991).
[6] C. Hampel, K. A. Peterson, and H.-J. Werner, Chem. Phys. Lett. **190**, 1 (1992); Chem. Phys. Lett. **192**, 332 (1992).
[7] P. R. Taylor, E. Bylaska, J. H. Weare, and R. Kawai, Chem. Phys. Lett. **235**, 558 (1995).
[8] H. Koch, O. Christiansen, R. Kobayashi, P. Jørgensen, and T. Helgaker, Chem. Phys. Lett. **228**, 233 (1994).
[9] A. P. Rendell and T. J. Lee, J. Chem. Phys. **101**, 400 (1994).

Downloaded¬29¬Jan¬2010¬to¬147.156.182.23.¬Redistribution¬subject¬to¬AIP¬license¬or¬copyright;¬see¬http://jcp.aip.org/jcp/copyright.jsp

[10] O. Vahtras, J. Almlöf, and M. Feyereisen, Chem. Phys. Lett. **208**, 359 (1993).

[11] H. Koch and P. Jørgensen, J. Chem. Phys. **93**, 3333 (1990).

[12] H. Koch, H. J. Aa. Jensen, P. Jørgensen, and T. Helgaker, J. Chem. Phys. **93**, 3345 (1990).

[13] H. Koch, R. Kobayashi, A. Sánchez de Merás, and P. Jørgensen, J. Chem. Phys. **100**, 4393 (1994).

[14] R. Kobayashi, H. Koch, and P. Jørgensen, Chem. Phys. Lett. **219**, 30 (1994).

[15] J. Gauss and J. F. Stanton, J. Chem. Phys. **102**, 251 (1995).

[16] O. Christiansen, A. Halkier, H. Koch, P. Jørgensen, and T. Helgaker (to be published).

[17] H. Koch, P. Jørgensen, and T. Helgaker (to be published).

[18] J. Noga, W. Kutzelnigg, and W. Klopper, Chem. Phys. Lett. **199**, 497 (1992).

[19] H. Koch, H. J. Aa. Jensen, P. Jørgensen, T. Helgaker, G. E. Scuseria and H. F. Schaefer III, J. Chem. Phys. **92**, 4924 (1990).

[20] T. H. Dunning, J. Chem. Phys. **90**, 1007 (1989).

[21] HERMIT, a molecular integral program, T. Helgaker, P. R. Taylor, K. Ruud, O. Vahtras, and H. Koch.

Downloaded¬29¬Jan¬2010¬to¬147.156.182.23.¬Redistribution¬subject¬to¬AIP¬license¬or¬copyright;¬see¬http://jcp.aip.org/jcp/copyright.jsp