

A NEW DIRECT CI METHOD FOR LARGE CI EXPANSIONS IN A SMALL ORBITAL SPACE

Per E.M. SIEGBAHN

Institute of Theoretical Physics, University of Stockholm, S-11346 Stockholm, Sweden

Received 8 June 1984; in final form 18 July 1984

A new direct CI method is presented, which is particularly suited for large CI expansions in a small orbital space. These are the type of expansions which are common in the CAS SCF method. Only one-electron coupling coefficients are stored, which leads to reduced elapsed times and storage requirements compared to earlier approaches. The two-electron coupling coefficients are implicitly created in the diagonalization step. The algorithm for updating the CI vector is formulated as the trace of a product of three matrices, $\text{Tr}(\mathbf{A} \cdot \mathbf{D} \cdot \mathbf{I})$. By ordering the one-electron coupling coefficients (\mathbf{A}) in a certain way the matrix \mathbf{D} is easily created as a sparse scalar product between these coefficients and the trial CI vector. The main computational step is then a simple matrix multiplication between the matrix \mathbf{D} and the symmetry blocked integral matrix (\mathbf{I}). This operation vectorizes very well on most vector processors. Another sparse scalar product between the resultant matrix and the coupling coefficients leads to the update of the CI coefficients. In a calculation on CRAY-1 with 30700 configurations, the two-electron part in a CI iteration required 10 s of which half went into the handling of the one-electron formula tape.

1. Introduction

In a conventional CI calculation the Hamiltonian matrix elements, or equivalent formulas, have to be saved from iteration to iteration. This will quickly lead to storage and input-output problems even for rather short CI expansions (10^4 terms and more). With modern vector processors, which have a very high efficiency for certain floating point operations but with much less efficient input-output capacities, this problem is even more pronounced today. To circumvent the input-output and storage problems, ways of directly constructing the update of the CI vector from a limited list of formulas, the old CI vector and the integrals have to be found. Such direct CI methods exist today for the case of single and double excitations from a multi-reference CI expansion [1]. Even though the vector processor can be made to run at nearly optimal CPU efficiency through matrix formulations of the algorithms [2], input-output will still be a modest fraction of the elapsed time. The problem can therefore be considered to be solved for these types of CI expansions. Another type of CI expansion which is frequently used in our research group in connection

with the CAS SCF (complete active space SCF) method [3] is a CI expansion, often complete CI, in a rather small active orbital space. The most efficient way of performing the CI step in a CAS SCF calculation is today still an essentially conventional type of CI. This means that the CI step will often dominate a CAS SCF calculation and there will be a large amount of input-output. This paper describes a very simple algorithm for performing such a CI calculation from a substantially reduced number of formulas. The algorithm will further vectorize well on most vector processors.

The theoretical background to the method is extremely simple and not new. In fact most CI methods today are built on the same general formulation. The coupling coefficients for the two-electron integrals are written as products of one-electron coupling coefficients. The possibility to do this follows immediately from the expression of the Hamiltonian operator in second-quantized form. One of the first methods where this formulation was explicitly used is described by Wetmore and Segal [4], who reference earlier work by Salmon and Ruedenberg [5], Cooper and McWeeny [6] and Paldus [7]. The more familiar unitary group approaches in the formulation of Paldus [8] and

Shavitt [9] are also built with the same starting point. The method in this paper is, however, much simpler than these formulations and in the case of a CI in a small orbital space it is also more efficient. The only formulas stored are one-electron coupling coefficients, generated by the Shavitt technique [9]. A crucial point in the new method is that these formulas are ordered after the intermediate index from the projection operator. This leads to a simple update formula for the CI vector using straight matrix products. The configuration selection is general as long as all required MO integrals can be held in core storage. The efficiency goes down, however, as more and more configurations are deleted. In the case of the CI expansions which are of interest in a CAS SCF, or a general MC SCF method, it is usually no problem to keep all the integrals in core. The details of the method are given in section 2 and examples from applications on NiC_4H_8 on a CRAY-1 are in section 3.

In the context of complete CI calculations in a small orbital space, the method of Handy [10] should also be mentioned. This is a direct CI method capable of treating large CI expansions of up to 10^6 terms. Many important benchmark calculations have been performed with this method [11]. The major problem with the method is that it is rather slow and the algorithms do not vectorize very well. The method has therefore not been used in CAS SCF calculations where the CI step has to be repeated many times. An efficient direct CI method has also been constructed for the special case of complete CI for only three electrons [12].

2. Method

The present CI method is designed to be used in connection with the CAS SCF method. The requirements are therefore that the method should be able to perform a large CI in a small orbital space with the use of a minimum number of formulas. Since most of the applications will be on vector processors, the algorithm should preferably vectorize rather well. It has recently become more common to use the CAS SCF method with restrictions in the configuration set, such as a limitation on the number of electrons in each spatial symmetry. The method should therefore allow the use of general configuration selection schemes.

Since we want to avoid all references to explicit Hamiltonian matrix elements and instead refer directly to the integrals, a proper starting point is the Hamiltonian in second-quantized language,

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

where p, q, r, s are molecular orbitals and E_{pq} are the generators of the unitary group. In the CI problem the Hamiltonian matrix is diagonalized which is done either by an iterative method or by perturbation theory. In both cases the main computational step in each iteration is the construction of the vector σ ,

$$\sigma_\mu = \sum_\nu H_{\mu\nu} c_\nu,$$

where μ, ν are labels for the chosen configuration basis and c is the wavefunction from the previous iteration. The main working equation in the direct CI method can then be written,

$$\sigma_\mu = \sum_{\nu, pq} A_{pq}^{\mu\nu} (p|h|q) c_\nu + \frac{1}{2} \sum_{\nu, pqrs} A_{pqrs}^{\mu\nu} (pq|rs) c_\nu, \quad (1)$$

where

$$A_{pq}^{\mu\nu} = \langle \mu | E_{pq} | \nu \rangle, \quad A_{pqrs}^{\mu\nu} = \langle \mu | E_{pq} E_{rs} - \delta_{qr} E_{ps} | \nu \rangle$$

are the direct CI coupling coefficients.

From the expressions for these coupling coefficients, it is obvious that the two-electron coefficients can be written as products of one-electron coefficients. The exact expression is obtained by inserting the projection operator over the complete configuration space, the resolution of identity,

$$A_{pqrs}^{\mu\nu} = \sum_\kappa A_{pq}^{\mu\kappa} A_{rs}^{\kappa\nu} - \delta_{qr} A_{ps}^{\mu\nu}. \quad (2)$$

It is in the treatment of the product term where the unitary group methods differ from each other. Wetmore and Segal [3] used (2) as a way to construct formulas for the CI step, which was then performed

in the conventional way by construction of the Hamiltonian matrix. Paldus [8] viewed (2) as a sparse matrix product in the configuration indices. This method was developed further and tested by Robb and Hegarty [13], who found it most convenient to construct a formula tape for the CI step. In the Shavitt method [9] the product is performed by going through tables of special formulas. The special cases are identified by a graph. In all of these methods the products are performed once and for all before the CI step. The methods are not efficient enough to be used to evaluate the products in each CI iteration. The storage problems of the formulas are therefore not avoided.

In the present method we use the fact that in the product in (2) the same intermediate index κ appears in both one-electron coupling coefficients. Only the one-electron coefficients are calculated and stored and these coefficients are ordered after κ . This means that when the two-electron coefficients are constructed in the update formula (1), they can be formed directly from multiplications within the same group of one-electron coefficients, which are ordered sequentially after each other. When (2) is inserted into (1), we obtain for the term involving the product of one-electron coupling coefficients,

$$\Delta\sigma_\mu = \frac{1}{2} \sum_{\nu, pqrs} \sum_{\kappa} A_{pq}^{\mu\kappa} A_{rs}^{\kappa\nu} (pq|rs) c_\nu.$$

This term can then be written after reorganisation,

$$\Delta\sigma_\mu = \frac{1}{2} \sum_{\kappa} \sum_{pq} A_{\kappa, pq}^{\mu} \sum_{rs} I_{pq, rs} D_{rs, \kappa}, \quad (3)$$

which has been written to emphasize the matrix structure with

$$A_{\kappa, pq}^{\mu} = A_{pq}^{\mu\kappa}, \quad I_{pq, rs} = (pq|rs), \quad D_{rs, \kappa} = \sum_{\nu} A_{rs}^{\kappa\nu} c_\nu.$$

Eq. (3) in matrix notation is

$$\Delta\sigma_\mu = \frac{1}{2} \text{Tr}(A^\mu \cdot \mathbf{I} \cdot \mathbf{D}).$$

The corresponding contribution $\Delta\mathbf{P}$ to the second-order density matrix \mathbf{P} , which is needed in the CAS SCF method, is written

$$\Delta\mathbf{P} = \frac{1}{2} \mathbf{D} \cdot \mathbf{D}^T.$$

The three matrices in eq. (3) have different character. The symmetry blocked integral matrix \mathbf{I} is completely dense, i.e. there are normally not any zero elements. The matrix \mathbf{A}^μ on the other hand is very sparse. Since only one generator E_{pq} acting on the row index configuration κ will generate configuration μ , there will only be one element in each row. We note that E_{pq} can be either a lowering or a raising generator. The matrix \mathbf{D} is in a usual CAS SCF calculation rather dense. With N electrons in N orbitals the density of matrix \mathbf{D} can be estimated with simple arguments. If the generator E_{rs} should give a non-zero result when it acts on configuration κ , orbital s has to be occupied and r cannot be doubly occupied. With an equal distribution of empty, singly occupied and doubly occupied orbitals in κ the chance for this should be $4/9$. Since the result of acting with E_{sr} on κ is also entered in matrix element $D_{\kappa, rs}$ the total density should be $8/9$. With many spin coupling possibilities, the chance for obtaining a zero D element increases and the density is therefore probably at most $1/2$ in a usual case.

The trace of a product of three matrices requires one matrix multiplication and a scalar product. The matrix product can be performed between any two of the three matrices. Since the matrices \mathbf{D} and \mathbf{I} are independent of the index μ it is clearly an advantage to perform the product between these two matrices. With many rows rs the product should be organized so that a test on the sparsity of \mathbf{D} is possible. Otherwise the inner loop has to run over κ which is always much larger than rs . Both options have to be present in the program.

The matrices \mathbf{D} and \mathbf{A}^μ in (3) are easily formed with a formula organisation like

$$\begin{aligned} \kappa 1: & (pq)_1, \mu 1, A_{(pq)_1}^{\kappa 1 \mu 1}; (pq)_2, \mu 2, A_{(pq)_2}^{\kappa 1 \mu 2}; \dots, \\ \kappa 2: & (pq)_1, \mu 2, A_{(pq)_1}^{\kappa 2 \mu 1}; (pq)_2, \mu 2, A_{(pq)_2}^{\kappa 2 \mu 2}; \dots, \\ & \vdots \\ \kappa i: & \dots, \text{etc.} \end{aligned} \quad (4)$$

This means that all one-electron coupling coefficients involving a particular intermediate state κi are written sequentially after each other. For each coupling coefficient the generator index $(pq)_j$ and the index of the other (besides κi) interacting configuration μj have to be written on the tape as above.

It should be noted that the $A_{(pq)}^{\kappa\mu}$ are the matrix elements of both E_{pq} and E_{qp} .

A few properties of the **D** matrix are worth noting. First, the coupling coefficients required to form one matrix element of **D** are found sequentially on the formula tape with the above organization, which makes the formation of the **D** matrix easy. Second, when there are many different spin coupling possibilities the same value of $(pq)_i$ will appear many times for the same κ with different coupling coefficients. All of these coefficients will go into the same matrix element of **D**. This summation over spin couplings will therefore drastically reduce the number of necessary multiplications required to form **σ** . This is an advantage which is not present in conventional CI techniques where each matrix element is multiplied by only one CI coefficient during the iterations. In direct CI methods this summation over spin coupling has been used by Saunders and van Lenthe [2] in their internal spin driven scheme, and by the present author in the internally contracted CI method [14].

The matrix products in (3) should be made in symmetry blocks. As usual only one-dimensional point groups will be used. For a particular κ all the (pq) values will then automatically have the same symmetry, namely the product symmetry of κ and μ . The configurations μ belong to the selected configuration set and are therefore all of the same symmetry. For each κ one column of the **D** matrix will be formed, which should then be multiplied with a particular symmetry block of integrals. To symmetry block the matrix **D**, κi of the same symmetry are therefore required sequentially on the formula tape. This suborganization of the tape is easily performed when the coupling coefficients are sorted.

The formulas which are used in the CI step for updating the **σ** vector are organized in two different data sets. In the first data set the coupling coefficients $A_{pq}^{\mu\nu}$ ($p \neq q$) are written, with both configurations μ and ν belonging to the selected configuration set. This means that the orbitals p and q are of the same symmetry. This data set is the same as the one-electron data set in the ordinary unitary group method, with certain slight modifications due to the omission of the coupling coefficients over the weight generators E_{pp} in the two-electron data set (see below). For this data set a sorting as in (4) is not required. The second data set, which is organized as in (4), is used for the two-

electron integrals. This data set contains the same type of one-electron coupling coefficients $A_{pq}^{\mu\nu}$ ($p \neq q$) as in the first data set with the important difference that one of the configurations μ or ν may now be outside the selected configuration set. If the molecule has any symmetry, this means that the orbitals p and q can belong to different irreducible representations. For a fixed number of configurations, it is clear that the most compact formula tape is obtained for the case of a complete CI for a molecule without symmetry. In this case the formula tape is only three times as large as the one-electron part of the formula tape in a conventional CI. With a very restricted CI expansion the number of intermediate states can be much larger than the number of selected configurations, which will lead to a much longer second data set. In the applications so far, however, the size of the data set in the present method has been much shorter in all cases than it was before with all two-electron coupling coefficients explicitly written on the tape. To reduce the size of the second data set further, the coupling coefficients are not written as real numbers but are given an integer label by which they are identified in the CI section. The label is then packed into the integer number which identifies the indices of the coupling coefficient. For this procedure to work the number of different values of the coupling coefficients must not be too large. The largest number found so far was 360 for a singlet with 30700 configurations. The largest number allowed in the present program is $16383 (2^{14} - 1)$.

The data set organized as above will be used in the CI step to perform the updating according to (3) in the following way. The formulas for one κi are read into core. In the process of reading these formulas the vector elements $D_{rs,\kappa}$ are formed from multiplying the trial CI coefficients $c_{\mu j}$ with the coupling coefficients $A^{\kappa\mu j}$. The next κi is then read into core if it has the same symmetry as the previous κi and another column of the **D** matrix can be calculated. As many columns of **D** as can be fitted into the available core space are constructed in this way. The matrix product between **D** and the symmetry blocked integral matrix, where all integrals are divided by two, is then performed. The resultant matrix is organized as a super-vector which enters a sparse scalar product with the same **A^μ** coupling coefficients that were used to build the **D** matrix. The result of this scalar product is scat-

tered and added to the positions of the update vector σ . The formula tape is then read again to construct a new \mathbf{D} matrix, etc.

In the two-electron contribution to σ there are some special cases. Detailed formulas for each integral type are given by Shavitt [9] and these expressions are essentially followed in the present version of the method. First, the diagonal elements of the Hamiltonian are required separately in the diagonalization step. The contributions from the Coulomb integrals and the δ term of the exchange integrals are obtained from

$$\Delta H_{\mu\mu} = \sum_i O_i^\mu \sum_j J_{ij} O_j^\mu - \sum_i \sum_j K_{ij} O_j^\mu,$$

where O^μ is the occupation vector of configuration μ and \mathbf{J} and \mathbf{K} are the Coulomb and exchange integral matrices divided by two. The diagonal elements are set to zero in these matrices. The integrals $(ii|ii)$ are added afterwards to $H_{\mu\mu}$ if orbital i is doubly occupied. The one-electron Fock matrix elements are added as usual. These half-way constructed diagonal elements are stored as one data set and used in the update of the σ vector in each iteration. The diagonal elements are completed by an update as in eq. (3) but only considering the generator product contribution from the exchange integrals. The reason for the use of the only half-way constructed diagonal elements in the updating is that the remaining contribution is automatically obtained from the general matrix product (3). All tests in the matrix multiplication must obviously be avoided.

All δ terms in the formulas for the integral types, resulting from the insertion of expression (2) into eq. (1), are handled by adding the corresponding integral to the one-electron Fock-type matrix, which also contains the one-electron integrals and the contribution from the frozen orbitals as usual.

Since the formulas over the weight generators E_{pp} are not written on the formula tape, the handling of the integrals $(pp|qr)$ requires a special treatment. This is done by writing also the occupation of configuration ν after the one-electron coupling coefficient $A_{pq}^{\mu\nu}$ in the one-electron data set. The integrals $(pp|pq)$ are special cases of the same integral type and are treated in a similar way (see ref. [9]). The contribution from these two-electron integrals is obtained by performing a scalar product between the occupation vector and the integrals and adding the result to the

corresponding Fock matrix element, F_{qr} for $(pp|qr)$, when the formulas for the one-electron integrals are read.

A detailed timing analysis of the present method as formulated in eq. (3), and a comparison with the conventional approach as formulated in eq. (1), is quite difficult. As mentioned in section 1, the goal with the present method was to reduce the elapsed time and, even more important, the storage requirements. Such a reduction is sometimes worth even a rather large increase in CPU time (compare for example with the philosophy in the integral program DISCO where the integrals are recomputed in each SCF iteration [15]). It may seem as if the conventional approach must have an advantage in terms of CPU time compared to the present approach since the summation over κ in eq. (2) is performed prior to the CI step. This sum, which is over the different spin couplings, can in usual CAS SCF calculations often be quite long due to the large number of configurations with many open shells. There are two arguments against this simplified way of reasoning. First, the CPU time spent in reading and unpacking a large number of two-electron coupling coefficients is certainly not negligible. Second, in the present method, a presumption over spin couplings before the multiplication with the integrals is done anyway when the \mathbf{D} matrix is formed. Conclusive timing comparisons between the two methods have not been done yet since optimal versions of the two methods are not available on the same type of computer. The preliminary experience shows, however, that on a vector processor the present method is usually much faster. On scalar machines the present method is also faster in most cases. Exceptions are cases where the number of configurations are substantially reduced due to symmetry and excitation levels between symmetries, in which case the methods are about equally fast.

A final point of interest could be mentioned here. The number of multiplications needed to form a two-electron coupling coefficient is equal to the number of intermediate states κ . The question arises if it is possible to minimize the number of intermediate κ states by a better choice of configuration state functions. In the present program Gelfand states were used. If bonded functions are used, the operation of a single particle generator E_{pq} on a bonded function will generate just another bonded function, not a lin-

ear combination. It would thus seem as if there would be only one intermediate κ state if bonded functions were used. This is unfortunately not correct. Since the bonded functions are not orthogonal, the projection operator that introduces the κ states has a different form, including the overlap between the non-orthogonal states. This means that also for bonded functions there will in principle be as many intermediate states as there are spin couplings of the κ configuration.

3. Applications

The method described in section 2 has been programmed for the CRAY-1 computer and a few preliminary timings are presented in this section. The applications are on NiC_4H_8 arranged in a five-membered metallacycle ring. There are 87 CGTO basis functions and the symmetry is C_{2v} . The CPU times for the different calculations are in table 1. The program is still being optimized and the timings should therefore not be taken as what can optimally be achieved. The diagonal elements and the one-electron part are still run in essentially scalar (and unoptimized) mode for example, but can rather straightforwardly be vectorized after the algorithms given in section 2. Large improvements are expected particularly for the diagonal elements where the main computational step should be two matrix times vector products.

The CPU times in table 1 show that it is possible to do rather large CAS SCF calculations in a reasonable time. The CI iteration times for the medium-size cases are in fact shorter than in the conventional ap-

proach where all two-electron coupling coefficients are precomputed (see discussion in section 2). Preliminary calculations on a VAX 780 show that this seems to hold also on a scalar machine. The main advantage with the new method is, however, that the elapsed times and storage requirements are much reduced. It is in fact impossible on our installation to perform a CAS SCF calculation as large as 30000 configurations with our earlier program which used the unitary group method of Shavitt [9]. Since the reduction of elapsed time was the main objective with the new method it was surprising and remarkable to find that the CPU times were also reduced. The iteration time is, however, much longer than the time required for the same number of configurations in a multi-reference CI expansion, which is somewhat of a disappointment. Part of the reason for this is that the handling of the formula tape is still far from negligible. In the largest calculation with 30700 configurations, 5 of the 10 s in the two-electron part goes into reading the tape and unpacking integers. Even though only one-electron coupling coefficients are saved on the formula tape, this data set is as large as 5 Mwords. It is possible that some time in handling the formula tape can be saved by using non-standard input-output, larger buffers and vectorizing the unpacking of the indices. An ideal solution would clearly be if an algorithm was found which could be used to calculate the coupling coefficients at a speed comparable to the reading of the tape. Again, such an ideal method will probably lead to longer CPU times per iteration but will have much shorter elapsed times and negligible storage requirements, which is much more important on modern computers.

Table 1
CPU times in seconds for NiC_4H_8 on CRAY-1. The total CI iteration time is the sum of the one- and two-electron parts

Orbitals ^{a)} Electrons	Config- urations	Diagonal el. and sorting ^{b)}	One-electron part ^{b)}	Two-electron part
5,3,0,0, 8	900	2	<1	<1
7,3,0,0, 10	9800	7	2	2
8,3,0,0, 10	30744	23	10	10

^{a)} Orbital occupation in C_{2v} symmetry, a_1 , b_2 , b_1 and a_2 .

^{b)} Not yet optimized and vectorized.

As mentioned in section 2, the matrix product in (3) can be done in any order. The timing in table 1 is where the innerloop is over the integral pair index rs , which allows a test on the sparsity of the \mathbf{D} matrix. When the innermost loop was changed to be over the intermediate states κ , which is a much larger number, the time went up to 12 s. In this case the test on sparsity is on the integral matrix, which is very dense. When there are active orbitals in more symmetries than two, the latter order of the matrix product will probably be an advantage for the symmetries with few orbital pairs.

Some data from the construction of the formula tape can also be of interest. In the largest test case with 30744 configurations, there were 60984 intermediate configurations. The coupling coefficients were generated using the Shavitt technique [9] which is a tree search method which is very unfavourable on a vector processor. There were 0.7×10^6 coefficients in the first data set and 3.2×10^6 coefficients in the second data set. The time to calculate these coefficients was 280 s and to sort them 80 s. There is no doubt that these timings can be reduced considerably by using algorithms which are more adapted for vectorization. A final number of interest is the number of different values for the coupling coefficients. With 3.8×10^6 coupling coefficients, there are only 360 different values.

Acknowledgement

The calculations were performed on the CRAY-1

in Linköping using a grant from the Swedish Natural Science Research Council.

References

- [1] P.E.M. Siegbahn, *J. Chem. Phys.* 74 (1980) 1647.
- [2] V.R. Saunders and J.H. van Lenthe, *Mol. Phys.* 48 (1983) 923.
- [3] P.E.M. Siegbahn, J. Almlöf, A. Heiberg and B.O. Roos, *J. Chem. Phys.* 74 (1981) 2384; B.O. Roos, P.R. Taylor and P.E.M. Siegbahn, *Chem. Phys.* 48 (1980) 157.
- [4] R.W. Wetmore and G.A. Segal, *Chem. Phys. Letters* 36 (1975) 478.
- [5] W.I. Salmon and K. Ruedenberg, *J. Chem. Phys.* 57 (1972) 2776.
- [6] I.L. Cooper and R. McWeeny, *J. Chem. Phys.* 45 (1966) 226.
- [7] J. Paldus, in: *Theoretical chemistry: advances and perspectives*, Vol. 2, eds. H. Eyring and D.J. Henderson (Academic Press, New York, 1976) p. 131.
- [8] J. Paldus, *J. Chem. Phys.* 61 (1974) 5321.
- [9] I. Shavitt, *Intern. J. Quantum Chem.* 12S (1978) 5.
- [10] N.C. Handy, *Chem. Phys. Letters* 74 (1980) 280.
- [11] R.J. Harrison and N.C. Handy, *Chem. Phys. Letters* 95 (1983) 386.
- [12] P.E.M. Siegbahn, in: *Quantum Chemistry – The State of the Art*, Proceedings of SRC Atlas Symposium, No. 4, eds. V.R. Saunders and J. Brown (Atlas Computer Laboratory, Chilton Didcot, Oxfordshire, 1975).
- [13] M.A. Robb and D. Hegarty, in: *Correlated wavefunctions*, ed. V.R. Saunders (Science Research Council, 1978).
- [14] P.E.M. Siegbahn, *Intern. J. Quantum Chem.* 18 (1980) 1229.
- [15] J. Almlöf, K. Faegri Jr. and K. Korsell, *J. Comput. Chem.* 3 (1982) 385.