Principles for a Direct SCF Approach to LCAO–MO *Ab-Initio* Calculations

J. Almlöf, K. Faegri, Jr., and K. Korsell

Department of Chemistry, University of Oslo, P.O. Box 1033 Blindern, Oslo 3, Norway

Received 28 September 1981; accepted 23 December 1981

The principles and structure of an LCAO–MO *ab-initio* computer program which recalculates all two-electron integrals needed in each SCF iteration are formulated and discussed. This approach—termed "direct SCF"—is found to be particularly efficient for calculations on very large systems, and also for calculations on small and medium-sized molecules with modern minicomputers. The time requirements for a number of sample calculations are listed, and the distribution of two-electron integrals according to magnitude is investigated for model systems.

I. INTRODUCTION

The basic theory and principles for the LCAO–MO approach to *ab-initio* Hartree–Fock calculations for molecules were formulated by Roothaan and Hall in 1951 (refs. 1,2) for closed-shell systems and extended to the open-shell RHF scheme by Roothaan in 1960.[3] Based on ideas set forth by them, and on the development of increasingly faster electronic computers, the field of computational quantum chemistry has gone through a rapid development up to its present state, where quantitative predictions can be made regarding properties and reactions for molecules of considerable size. During these years, the concept of LCAO Hartree–Fock calculations and computer programs has also undergone an evolution closely coupled to the concomitant development of modern computer hardware and software.

Thus, historically, the LCAO Hartree–Fock procedure has evolved into a two-step computational scheme, where each step is carried out by a separate part of the program. In the first step, executed by the "integral part" of the program, all necessary integrals over the operators of the Hamiltonian are computed in a basis of atomic functions and saved on external storage. The number of such integrals increases as the fourth power of the number of basis functions, and easily reaches 10^6 to 10^7 in medium-size calculations. In the second step, executed by the "SCF part" of the program, the iterative Roothaan–Hall equations are solved; this involves reading through the list of integrals once for every iteration in order to construct the Fock matrix and compute the energy.

For large molecules, the tasks solved by the two parts of such a program system are significantly different from a computational viewpoint. The calculation of integrals is characterized by extensive use of central-processor capacity, and is therefore often described as CPU-bound. The SCF part, on the other hand, also makes heavy use of input/output facilities. of the computer and therefore becomes I/O bound for large molecules. The integral evaluation has been the bottleneck of this computational scheme, mainly because of the large number of integrals that have to be calculated and stored. However, this is the area where the major efforts have been spent on developing faster and more efficient computer programs, and today the CPU requirements of the integral and the SCF sections are often comparable.

Concurrently with the development of improved software, there have been impressive developments in computer hardware. The emergence of powerful minicomputers has made CPU capacity easily accessible at low cost. At present, the price/performance ratio for CPU and memory appears to decrease much faster than that for I/O operations and use of peripheral devices. It has sometimes been pointed out[4] that a favorable I/O-to-CPU price ratio would make it more economical to recompute integrals in every iteration than to store and retrieve these.

CCC 0192-8651/82/030385-15$04.00

Suggestions for reevaluating integrals during the iterative SCF processes have also been expressed by Kaufman and Popkie.[5] Whether such an approach will be advantageous is, of course, dependent on various parameters of the computing environment in which the calculation is performed, as well as on the molecular system under investigation, and the type of calculation attempted. Nevertheless, it is argued here that the recomputation of integrals in every iteration—an approach which we shall term "direct SCF" since it has certain features in common with the "direct CI" approach[6]—offers a number of interesting possibilities and warrants serious consideration.

For two different situations it can immediately be concluded that the direct SCF method has obvious advantages. One of these is in the use of minicomputers for MO–LCAO calculations. Already in 1974, *ab-initio* calculations were carried out on rather small minicomputers by Schaefer and coworkers,[7] and such work is now routinely being performed in a large number of research groups around the world. However, the programs used are largely traditional two-step integral + SCF programs, designed and optimized for use on large mainframe systems, and modified to fit into the smaller environment if necessary. Minicomputers, while having quite powerful CPUs, often have a less developed I/O system, and I/O operations therefore as a rule are more cumbersome on such machines than on large computers. Use of a direct SCF approach will eliminate most of the I/O and is therefore a very interesting alternative for *ab-initio* LCAO–SCF calculations on a minicomputer.

The other situation in which the direct SCF approach will provide definitive advantages is in very large calculations (500 basis functions or more) on medium-sized or large computers. For calculations of this size, the storage and retrieval of two-electron integrals in the traditional two-step procedure would become prohibitive on any present computer system. It is advocated here that for such calculations the direct SCF approach is the only practical solution.

For some time, we have been developing a computer program for performing LCAO–MO calculations in a direct SCF scheme. Some results from applications of this scheme have been published previously.[8] In the present work we discuss our experiences with this new approach. The treatment is naturally divided into two parts. In the first, we discuss the basic principles involved and the problems encountered in the development of a direct SCF computer program. In the second part, the general layout of the program is described, and our computational experience with it is discussed. A final section is devoted to some conclusions and remarks about future possible developments.

II. BASIC PRINCIPLES OF DIRECT SCF CALCULATIONS

A. Integral Evaluation

In a direct SCF approach where integrals are recalculated in every iteration, one is faced with three basic problems: (a) to keep the number of integrals calculated in every iteration to an absolute minimum; (b) to calculate these integrals as efficiently as possible; and (c) to complete the calculation within a minimum number of iterations. All these considerations apply to all LCAO–MO programs, albeit with a different relative importance, and much effort has been spent in developing and improving the software within this area. In developing a program system for the direct SCF approach, one is therefore able to take advantage of existing techniques and algorithms. In this first section we will describe the various schemes available to reduce integral evaluation time.

Several methods have been suggested for improving or simplifying the evaluation of integrals in *ab-initio* calculations. In one approach, the inner shells of the atoms are replaced by a pseudopotential of effective core potential.[9,10] The evaluation of two-electron integrals involving the inner shells is thus avoided. The most obvious drawback of such a method seems to be the fact that the interaction between this inner-shell potential and the valence electrons does not have a physically correct form. This may lead to erroneous results when certain properties of a molecule are investigated with that method. However, it has also been demonstrated that such model potentials give great savings in computer time even in applications to first-row atoms.[11] Also, the most important properties of molecules seem to be reasonably well described with this approximation.

Another aspect, which appears to be particularly interesting in this context, is the use of pseudopotentials to describe effects brought about by relativity in heavy atoms.[12,13] The model potential concept used in this sense is not just a method of approximation, but also a way of incorporating important physical effects into the calculation. This is of special interest in connection with the direct SCF approach, as it involves studying systems with large numbers of electrons. This usually implies very large basis sets, a situation where we claim that our approach has particular advantages.

The pseudopotential concept may easily be incorporated into the direct SCF scheme, and we believe that such a combined approach would make many new areas of chemistry amenable to *ab-initio* studies. In this work, however, we will concentrate on features that are specific for the direct SCF method of calculation.

One principle which is central in most present-day schemes for rapid evaluation of integrals is the batchwise processing technique.[14] In batchwise processing integrals are calculated in groups, where basis functions differing only in the angular part, but having the same center of expansion and orbital exponent, are treated in parallel. This implies, in the case of four sets of p-type functions, for example, that 81 $(pppp)$-type integrals are evaluated in one "batch." Since such integrals have many elements of computation in common, very important time savings—often by orders of magnitudes—can be accomplished by use of the batch-processing technique. Karlström[15] has recently discussed how efficiency in the separate numerical steps involved in integral evaluation may be improved.

The simultaneous evaluation of a whole batch of integrals also has other advantages. Important time savings may often be obtained by making a rapid estimate of the magnitude of an integral before it is calculated. This estimate may be compared with some given threshold, and it can then be decided at an early stage whether it is necessary to calculate the integral. In the batchwise processing of integrals as described above, the test need be made only once for the whole batch of integrals. In particular, a given electron repulsion integral over primitive normalized Gaussians may be written as

$$(ijkl) = I_{ijkl} = T_{ijkl} \sum_m f_m \tag{1}$$

$$T_{ijkl} = S_{ij} S_{kl} \frac{\alpha_{ij}\alpha_{kl}}{\pi(\alpha_{ij} + \alpha_{kl})} \tag{2}$$

S_{ij} and S_{kl} are the radial overlaps:

$$S_{ij} = N_i N_j \left(\frac{\pi}{\alpha_{ij}}\right)^{3/2} \exp\left(-\frac{\alpha_i \alpha_j}{\alpha_{ij}} R_{ij}^2\right) \tag{3}$$

and α_{ij} is the sum of orbital exponents: $\alpha_{ij} = \alpha_i + \alpha_j$. N_i, N_j are the normalization constants. The advantages of the batchwise processing arise, among other things, from the fact that the factor T_{ijkl} is the same for all integrals within one batch, and, further, that the terms in the summation over m in eq. (1) have many elements of computation in common. In the pretesting, when only the order of magnitude of the integral is of interest, the factor T_{ijkl} is used as an estimate of the integral, and the test need only be performed once for the whole batch of integrals.

The above approach is similar in many respects to the approximation scheme suggested by Wilhite and Euwema.[16,17] The basic idea of their method is to neglect *both* one- and two-electron integrals involving small charge distributions, as defined by the product of basis functions. In this way, the neglect of overlap integrals guarantees that the approximated energy corresponds to a wavefunction which is properly normalized in all respects. The neglect of one-electron nuclear attraction integrals and two-electron electron repulsion integrals in a consistent fashion serves to maintain the charge balance of the system. In the original approach by Wilhite and Euwema,[12,13] the magnitude of the orbital product is estimated by the overlap integral between the basis functions involved, after these are reoriented so as to optimize the overlap. An improved version designed by Kaufman and Popkie,[5,18,19] known as the VRDDO scheme, uses instead the radial overlap to estimate the magnitude of the basis function product. This approach is better suited for applications with the batch processing of integrals than the original one. The similarity between the VRDDO technique and our eqs. (1)–(3) is evident. However, in our usual applications the one-electron integrals are left unmodified, and charge is therefore not strictly conserved. With the threshold values of of 10^{-8}–10^{-10} normally used by us, this feature is of no practical importance, as has been confirmed in a large number of applications. The energy and other properties computed with such a low threshold are virtually identical to those obtained with a zero threshold, i.e., when all contributions are computed regardless of their magnitude. Our approximation is thus purely numeric and involves no physical considerations. On the other hand, it is evident that the VRDDO approach—or similar charge-conserving approximation techniques—might easily be implemented in our present calculation scheme with minor modifications. Studies are presently in progress to investigate the efficiency of that method in combination with other special features of our program.

Kalström[15] has also discussed how knowledge of the density matrix may be used to eliminate those integrals which are to be multiplied by density matrix elements below a certain threshold. Contributions to the Fock matrix from a two-electron integral I_{ijkl} are in the closed-shell case:

1—'direct' or Coulomb type contributions

$$F_{ij} = F_{ij} + 4D_{kl}I_{ijkl} \qquad (4a)$$

$$F_{kl} = F_{kl} + 4D_{ij}I_{ijkl} \qquad (4b)$$

2—exchange-type contributions

$$F_{ik} = F_{ik} - D_{jl}I_{ijkl} \qquad (5a)$$

$$F_{jl} = F_{jl} - D_{ik}I_{ijkl} \qquad (5b)$$

$$F_{il} = F_{il} - D_{jk}I_{ijkl} \qquad (5c)$$

$$F_{jk} = F_{jk} - D_{il}I_{ijkl} \qquad (5d)$$

Clearly, if all density matrix elements used to multiply a certain integral are sufficiently small, then the corresponding contributions to the Fock matrix (and the energy) are insignificant, and it is unnecessary to evaluate this integral.

In the traditional SCF approach the density matrix is not known at the time of integral evaluation. Excessive elimination of Fock-matrix contributions which are initially small may lead to a variational collapse in the traditional approach, since the SCF optimization procedure has a tendency to build up charge in such regions of space where the electron repulsion has been neglected. Particularly for very large calculations this can become a problem which is manifested by very large density matrix elements multiplying these small (and therefore neglected) integrals.

In the direct SCF approach, on the contrary, integrals are recalculated in every iteration, at which time the density matrix is known. Full control is therefore maintained in every step of the calculation as to the errors introduced in the Fock matrix and the total energy expressions by the approximations made.

In practice, the above ideas are easily implemented in direct SCF algorithms by multiplying the test criterion for the integral expression by the quantity

$$\tau_1 = \max(4|D_{ij}|, 4|D_{kl}|, |D_{ik}|, |D_{il}|, |D_{jk}|, |D_{jl}|) \qquad (6)$$

The product may then be compared to the given threshold in the usual manner in order to decide whether it is necessary to calculate the integral or not. Any tendency toward a variational collapse will result in an increase in the corresponding density matrix elements, and the problem will, within reasonable limits, be corrected in the next iteration.

Karlström[15] has suggested a slightly different approach, which instead focuses on the contribution to the total energy from a given integral. He suggests the test criterion

$$\tau_2 = T_{ijkl}(|D_{ij}D_{kl}| + |D_{ik}D_{jl}| + |D_{il}D_{jk}|) \qquad (7)$$

Clearly, the criterion (7) would eliminate much more integrals than expression (6), since the product of density matrix elements will become small whenever one of the two elements is small. However, we have frequently encountered convergency problems when using eq. (7) in calculations on large systems. This behavior may be understood and explained if the contributions to the Fock matrix by an individual integral are considered. Assume, for simplicity, that only Coulomb-type contributions are important for a certain integral. The test criterion (7) may then be written as

$$\tau_2' = I_{ijkl}D_{ij}D_{kl} < \tau_0 \qquad (8)$$

and the Coulomb-type contribution [eq. (4a)] to the Fock matrix is then neglected whenever the change in the element (i, j) of the matrix is

$$\Delta F_{ij} = D_{kl}I_{ijkl} < \tau_0/D_{ij} \qquad (9)$$

Clearly, when D_{ij} is sufficiently small, this criterion does not warrant any particular accuracy in the computed Fock matrix element F_{ij}. The iterative SCF procedure may therefore show oscillating or divergent behavior in unfavorable cases. It is worth noting that this problem may occur for any preset value of τ_0 greater than zero. However, a test criterion closely related to eq. (7) may instead be used, which exhibits a similar efficiency but retains the convergence properties of the exact Fock matrix.

In this approach, advantage is taken of the fact that in an MO basis only Fock-matrix elements defining rotations between occupied and virtual orbitals are relevant to the wavefunction, and thus to the total energy and the convergence behavior. If the molecular orbital a is expanded in the basis functions with coefficients c_{ia}, the contribution to a Fock-matrix element in MO basis by an integral I_{ijkl} is given by:

$$\begin{aligned}\Delta F_{ab} = {} & (c_{ia}c_{jb} + c_{ja}c_{ib})\Delta F_{ij} + (c_{ia}c_{kb} \\ & + c_{ka}c_{ib})\Delta F_{ik} + (c_{ia}c_{lb} + c_{la}c_{ib})\Delta F_{il} \\ & + (c_{ja}c_{kb} + c_{ka}c_{jb})\Delta F_{jk} + (c_{ja}c_{lb} \\ & + c_{la}c_{jb})\Delta F_{jl} + (c_{ka}c_{lb} + c_{la}c_{kb})\Delta F_{kl} \\ = {} & I_{ijkl}(4E_{ij}^{ab}D_{kl} + 4D_{ij}E_{kl}^{ab} - E_{ik}^{ab}D_{jl} \\ & - D_{ik}E_{jl}^{ab} - E_{il}^{ab}D_{jk} - D_{il}E_{jk}^{ab}) \quad (10)\end{aligned}$$

where

$$E_{ij}^{ab} = c_{ia}c_{jb} + c_{ja}c_{ib} \quad (11)$$

The appropriate test criterion is then given by

$$\tau_3 = \max(4R_{ijkl}, R_{ikjl}, R_{iljk}) \quad (12)$$

where

$$R_{ijkl} = (f_i g_j + f_j g_i)D_{kl} + (f_k g_l + f_l g_k)D_{ij} \quad (13)$$

and

$$f_i = \max_a |c_{ia}|, \quad g_i = \max_b |c_{ib}| \quad (14)$$

The maxima in eq. (14) are to be taken over all occupied orbitals a and virtual orbitals b, and the construction of the vectors f and g is then a trivial matter from a computational viewpoint. Clearly, the criterion (12) will be particularly efficient whenever the elements of f and/or g involved in eq. (13) are small. This will be the case, e.g., with minimal basis representations of the inner shells, where the core basis functions have small coefficients in the virtual space, rendering the corresponding elements of g small. The savings achieved in such cases are similar in nature to those obtained by using pseudopotentials or effective core model potentials[9,10] which also assume a previously defined shape of the inner shells, usually one taken from atomic calculations. In the present case, this shape is defined by the orbital exponents and the contraction used for the inner shells. It should be noted, however, that contrary to the pseudopotential method, no approximation is introduced for the physical form of interaction between core and valence orbitals.

In addition to minimal basis studies, the criterion (12) is particularly efficient for polarization functions having small occupation, which leads to small f elements. This situation may lead to considerable savings in computation, as the evaluation of integrals over polarization functions is usually quite time-consuming.

A comparison may now be made with the criterion (7) that was previously suggested.[15] Noting that the magnitude of the density matrix element

$$D_{ij} = \sum_a^{occ} c_{ia}c_{ja} \quad (15)$$

is roughly governed by the largest element in the sum, we conclude that, usually,

$$D_{ij} \lesssim f_i f_j \quad (16)$$

Therefore, the criterion (7) is similar to criterion (12) with g replaced by f in the latter expression. Which of these two test criteria gives the largest reduction seems to be case-dependent, but criteria (12) has the clear advantage that full control is maintained over the errors in the Fock matrix, thus warranting a proper convergence behavior. The drawback is, however, that the total energy computed with eq. (12) has no guaranteed accuracy. After convergence (which obviously must be monitored using other quantities than the energy), one final iteration using the criterion (7) should therefore be performed if an accurate value of the total energy is desired. This disadvantage must be balanced against the efficiency gained as compared to the less restrictive criterion (6).

The batchwise processing introduces some additional considerations to this integral testing. The above discussion of contributions from individual integrals becomes less meaningful as the integrals are not evaluated individually. However, the algorithm may be modified to account for this fact. In order for the test to be as safe as possible, the whole batch of integrals is evaluated as soon as any one integral in the batch has a significant contribution to the Fock matrix. Following Karlström,[15] a compressed density matrix is therefore constructed as

$$D_{pq} = \max_{i\epsilon p, j\epsilon q} D_{ij} \quad (17)$$

where p and q denote shells of basis functions and i,j denote individual basis functions within these shells. This compressed density matrix may be constructed at the beginning of each iteration, and it can then be used in eqs. (6), (7), or (12) to obtain a test criterion that pertains to the whole batch of integrals under consideration.

A further development of these ideas can be made by realizing that it is unnecessary to recompute the entire Fock matrix in every iteration. Writing (in matrix notation) the Fock matrix for a closed-shell case in iterations m and $m - 1$ as

$$F_m = h + ID_m \quad (18)$$

$$F_{m-1} = h + ID_{m-1} \quad (19)$$

the Fock matrix F_m is readily obtained from the recursion formula

$$F_m = F_{m-1} + I(D_m - D_{m-1}) \tag{20}$$

by taking $F_0 = h$, $D_0 = 0$. The one-electron Hamiltonian h in the usual SCF procedure is then replaced by the old Fock matrix, and the change in the density matrix from the previous iteration replaces the density matrix. Thus, only changes in density have to be considered, and these will usually be very small when convergency is approached. The test criterion for deciding on the calculation of the integral may therefore also be based on $(D_m - D_{m-1})$ rather than on D_m. In this manner, only changes in density call for recalculation of integrals. The amount of computation in each iteration will therefore be reasonably small, and approaches zero as convergency is reached.

In every iteration, the criteria (6) or (12) will guarantee that the error in each individual contribution to the Fock matrix is less than the given threshold. Since the number of iterations in a direct SCF calculation is necessarily small for other reasons, the accumulation of errors in F through the SCF iterations will not constitute a serious problem.

For a molecule processing symmetry, the number of integrals that need to be evaluated may be further reduced by considering only symmetry-unique integrals. Very efficient schemes for utilizing molecular symmetry in integral evaluation have been developed by Dacre,[20] Elder,[21] and Dupuis and King.[22] Using bit patterns to represent the molecular point group and its subgroups, the group theory employed in the integral evaluation can be handled by Boolean operations NOT, AND, OR, and XOR. This produces very efficient algorithms for utilizing the molecular symmetry, with a minimum of logical IF-tests in the integral evaluation and processing loops. It thus provides for a very fast code on modern computers with "look-ahead" preexecution of instructions.

The Fock matrix may be built from the symmetry-unique integrals in a scheme similar to that employed by Dupuis and King,[22] again providing a fast-flowing code without the extensive testing and sorting or tagging of integrals that is used in many traditional LCAO programs (see ref. 23 and references therein).

B. Convergence Acceleration

For any SCF calculation the importance of minimizing the number of iterations necessary to reach the desired degree of self-consistency is obvious. In the direct SCF approach, where each iteration entails a recalculation of integrals, such considerations are of course even more crucial. There are two main ways to reduce this number: (1) to obtain the best possible estimate of the wavefunction (or of the density matrix) as a starting guess in the first iteration; and (2) to employ an optimally efficient extrapolation scheme to improve the convergency of the iterative process.

The problem of extrapolation in the iterative process has been discussed by a large number of authors,[24,25] and most LCAO-SCF programs routinely incorporate some scheme for convergence acceleration. One of the most efficient schemes for doing this is the procedure recently developed by Pulay,[25] which is especially useful in situations where very good convergence is needed for the calculation of properties that depend critically on the quality of the wavefunction. Any of the schemes in current use may of course be implemented and used also in connection with the direct SCF method.

Very recently, a quadratically convergent Hartree–Fock method has been described.[26] Although its convergency properties appear to be excellent, the method is not an attractive alternative for very large calculations, inasmuch as it is based on a single-excitation CI calculation in every iteration. Instead, we have focused our interest on another interesting approach to the orbital optimization problem which seems to be more suited for our purposes.

The Newton–Raphson method for minimizing the energy expression, using an exponential parametrization of the orbital rotations,[27] has gathered increased interest during the last years. In that method, the change of molecular orbitals (occupied and virtual) from one iteration to the next is described by a unitary matrix U, i.e.,

$$C_m = C_{m-1}U \tag{21}$$

The exponential parametrization implies that U is written as

$$U = \exp(X) \tag{22}$$

where the unitary condition for U is replaced by that of X being skew-symmetric, i.e., $X^+ = -X$.

The energy expression may then be written as a function of the independent parameters X_{ij}, $i<j$; and a Taylor expansion in these parameters can be performed:

$$E(X) = E(0) + \sum_{i<j} X_{ij}\left(\frac{\partial E}{\partial X_{ij}}\right)_0 + \frac{1}{2}\sum_{i<j},\sum_{k<l} X_{ij}X_{kl}\left(\frac{\partial^2 E}{\partial X_{ij}\partial X_{kl}}\right)_0 + \cdots \tag{23}$$

The expression is truncated after the quadratic terms and the minimum of the resulting quadratic function is determined as

$$X = -H^{-1}b \tag{24}$$

with

$$b_{ij} = \left(\frac{\partial E}{\partial X_{ij}}\right)_0, \quad H_{ijkl} = \left(\frac{\partial^2 E}{\partial X_{ij} \partial X_{kl}}\right)_0 \tag{25}$$

This approach has been shown to yield a convergence which is very much superior to that obtained in a traditional SCF approach, based on iterative Fock-matrix diagonalizations.[26] However, the technique is relatively time-consuming, as it requires the computation of first and second derivatives of the total energy with respect to all X_{ij}s, and its major application has hitherto been within the field of MCSCF methods.[28-30] These aspects must be modified somewhat in connection with the direct SCF scheme. As each iteration is relatively more time-consuming than in traditional SCF calculations, it is justified to spend more effort on improving convergence. The possibilities for including a simplified version of the Newton–Raphson approach into the direct SCF scheme appear very promising, and a detailed account of such work will be given in a forthcoming paper.

The selection of a good set of trial vectors, or, equivalently, of a good initial density matrix, often involves a large amount of skilled guesswork. In a scheme where the number of iterations is crucial, it is desirable to avoid the possible mistakes or ambiguities of this guesswork as much as possible. In order to work out an automatic scheme for selecting the trial density, we note that usually one of the following situations prevails:

(a) The density matrix for a close-lying geometry (in a series of pointwise variations of the molecular geometry) or for a similar electronic state (in a series of calculations on several such states at the same geometry) is available and may be used as an initial estimate.

(b) The density matrix from exploratory calculations in a smaller basis set is available and may be expanded to an estimate of the density in the larger basis. The cost for exploratory calculations in smaller basis sets is often so small that it permits the use of a poor guess of the initial density.

(c) Failing (a) or (b), a good starting density matrix may usually be obtained as a sum of the atomic density matrices, calculated with the same basis sets as used in the molecule. Such atomic calculations are often a matter of seconds and are conveniently performed as part of the input processing.

For many cases, it is possible to make an intuitive but reasonable estimate of the charge distribution among the different atomic fragments of the molecule. Such considerations may be based on electronegativities, experimental dipole moments, or on population analyses for similar cases. The trial densities constructed as in (c) may then be improved by modifying the number of electrons filled into the outermost atomic valence shells.

A density matrix obtained by either of these methods is usually not idempotent, and it will therefore not give an upper bound to the final converged energy when used in the calculation of the Hartree–Fock energy. In fact, density matrices obtained according to (c) usually give energies somewhat below the converged Hartree–Fock result. For those cases where this nonidempotency constitutes a problem, it may be overcome by considering the spectral resolution of the density matrix, the natural orbitals. Assuming that the initial density matrix is reasonably close to the converged result, its eigenvalues may be regarded as approximations to the eigenvalues of a correct Hartree–Fock density matrix. Thus, to obtain a density matrix which is compatible with the energy expression in the Roothaan formalism, the nonidempotent estimate may be diagonalized after a transformation to an orthonormal basis. The proper number of eigenvectors corresponding to the highest eigenvalues may then be used as occupied trial molecular orbitals.

This scheme is usually implemented into existing Hartree–Fock programs by substituting the negative of the estimated density matrix for the Fock matrix in the first iteration (one must of course account for the fact that the basis transformations on the density matrix are the inverse of those applying to the Fock matrix).

C. Symmetry Blocking

In an LCAO calculation the molecular symmetry may be used to simplify the computational work in at least two different ways:

(1) Symmetry imposes certain relations among the values of molecular integrals and thus reduces the number of nonredundant integrals that have to be calculated, as already discussed in Section II.

(2) The use of a symmetry-adapted basis set leads to a blocking of various matrices used in the calculation. Such a blocking of the one-electron matrices (density, Fock, and overlap matrices) reduces the requirement

of fast memory and makes various matrix operations less time-consuming (e.g., multiplication, transformation, diagonalization, etc.). For large molecules, the diagonalization of a nonblocked Fock matrix may require a substantial fraction of the total computation time in each iteration, even in the direct SCF approach.

Still more important in ordinary, two-step calculations is the blocking of the two-electron supermatrices, which reduces the time for construction of the Fock matrix and the external storage requirements for the integrals. The latter quantity increases as the fourth power of the number of basis functions, and is often a critical factor in large calculations. However, this blocking requires that the transformation be performed on the individual two-electron integrals, which in itself may be a time-consuming affair due to their large number. The Dacre scheme operates with an atomic orbital basis set, and no improvement in the construction of the Fock matrix will be gained by using a symmetry-adapted basis. In the direct SCF approach the problem of storage is eliminated, and the transformation to a symmetry-adapted basis can therefore be performed directly on the Fock matrix.

One such convenient transformation of the Fock matrix is provided by the orbitals of the previous iteration. These will span the irreducible representations of the full molecular point group, and will, if properly sorted, provide a basis in which the matrix representation of any totally symmetric operator will be fully symmetry blocked. These orbitals also form an orthonormal basis and therefore lead to a form of the Roothaan–Hall equations that can easily be solved by standard numerical methods. Further, since orbitals from the previous iteration and the present one are presumably rather similar, the off-diagonal elements of the transformed matrix are likely to be small (giving rise to the term "prediagonalization" for this transformation). Depending on which diagonalization method is used, this will either speed up the diagonalization or improve the numerical accuracy.

For the first iteration, however, a symmetry-adapted set of orthonormal orbitals is usually not available, so an alternative transformation must be found. Procedures for doing this have been discussed by others previously[31]; here we suggest an approach to the problem which is easily implemented in a program where all symmetry information needed for the Dacre scheme is already available. We start by noting that the eigenvectors V of the overlap matrix S

$$SV = V\epsilon \tag{26}$$

span the irreducible representations of the full molecular symmetry group, provided that the basis set on each atom is isotropic, i.e., that it exhibits local spherical symmetry.[*] The matrix

$$U = V\epsilon^{-1/2} \tag{27}$$

then provides a transformation to an orthonormal, symmetry-adapted basis. Thus, if the columns of U are sorted by irreducible representations, the problem of symmetry blocking is solved for all subsequent iterations. Since the symmetry information is already available, the sorting may be achieved by acting on the functions ϕU (ϕ being the column vector of basis functions) with all the symmetry operations of the molecular point group, computing the characters and comparing with a character table for the group. Such a table is easily generated during input processing when the symmetry operators are generated.

The calculation of the characters for the functions U requires a summation over symmetry degenerate sets, the degeneracies of which may be established by sorting the columns of V according to the eigenvalues ϵ and testing for equalities. In doing so, it is tacitly assumed that any coincidence among these eigenvalues can be ascribed to symmetry degeneracy, i.e., to multidimensional irreducible representations. One thus disregards the possibility of accidental degeneracies. This would normally be a quite reasonable assumption, since all calculations are performed in 60–64 bits precision and the degeneracy may therefore be checked to 14–15 decimal figures of accuracy. Some care must be exercised when very large internuclear distances are used for special purposes,

[*] The requirement for an isotropic basis on each atom is not as trivial as it may first seem. A first, necessary condition is that all functions within a "shell" (i.e., three p-type functions, six Cartesian d-type functions, etc.) have the same orbital exponents, contraction scheme, and center of expansion. This condition must be fulfilled in any case in order to employ the batchwise processing scheme in the integral evaluation step. The complications arise from the fact that Cartesian basis functions within one shell are not necessarily orthogonal. The simplest illustration of this is provided by the set of six Cartesian d functions ($x^2, xy, xz, y^2, yz, z^2$). Because the d functions are not orthogonal (e.g., $<d_{x^2}|d_{y^2} \mp 0$), the functions resulting from a rotation on the basis set by a unitary matrix are not normalized. For instance, a rotation of d_{xy} by 45° in the x,y plane yields $2^{-1/2}(d_{x^2} - d_{y^2})$, a function which has the norm $2/\sqrt{3}$. For d orbitals the problem is most easily circumvented by normalizing the nonorthogonal Cartesian basis functions to 3/2. Similar renormalizations may be carried out on basis functions with higher angular quantum number.

but normally this does not constitute a serious problem.

A somewhat more intricate situation arises, however, in cases when exact degeneracies in the eigenvalues of the overlap matrix are brought about by a certain choice of basis set. In particular this is a problem for the point groups having IRs with complex characters, i.e., T, T_h and C_n, C_{nh}, and S_{2n} with $n > 2$. These groups have complex, one-dimensional representations, which are rendered pairwise degenerate by the choice of real basis functions. Such groups are most conveniently treated using the real, reducible representations; therefore the test for degenerate eigenvalues in the sub-blocking of the overlap matrix discussed below must treat these groups as special cases.

In order to classify the functions ϕU with respect to symmetry, one may operate on the set with each symmetry operator of the point group. A symmetry operator $\hat{k}$ represented by a matrix K, such that

$$\hat{k}\phi = \phi K \tag{28}$$

will have a character

$$\chi = \sum_{i=1}^{l} \langle (\phi U)_i \hat{k} (\phi U)_i \rangle = \sum_{i=1}^{l} (U^+ SKU)_{ii} \tag{29}$$

where the functions $\{(\phi U)_i, i = 1, l\}$ span a one-dimensional irreducible representation. For each vector (or degenerate set of vectors) the characters may be compared row-by-row with the character table, and the vector can then be assigned uniquely to an irreducible representation.

The functions spanning a degenerate representation are, in general, arbitrarily mixed after the diagonalization of the overlap matrix. However, it is desirable, in particular for open-shell calculations, that these degenerate representations be split into subspecies in a well defined and unique way. This further sub-blocking of multidimensional IRs also speeds up diagonalization and reduces memory requirements. Typically for an E representation spanned by, e.g., p_x and p_y orbitals on a central atom, it is desirable to block this into a p_x block and a p_y block. This may be done using the subgroups C_s with the σ_v plane of the full group as the plane reflection, or C_2 with one of the twofold secondary axes as rotation axis. The matrix representation for this operator may be diagonalized, giving eigenvalues of +1 or −1. If the eigenvectors are sorted according to these eigenvalues and the overlap matrix is transformed to this basis, one achieves a preblocking in which the two subspecies of any two-dimensional irreducible representation will always occur in different blocks, and may thus be identified and placed into different sub-blocks after the diagonalization of the overlap matrix.

The procedure outlined here is easily generalized to groups with IRs of dimensions higher than two; in this case more than one nontrivial subgroup operation is necessary to obtain full blocking.

D. RHF Procedures

The SCF equations remain the same for the direct SCF scheme as for the traditional approach, and existing methods for solution may be used without further changes. However, some thought must be given to the open-shell RHF case. Whereas the open-shell calculations in the traditional approach often involve only the extra expense of doing an SCF calculation with integrals that have already been calculated for another state (e.g., when doing SCF calculations for a series of different ionized or excited electronic states), open-shell calculations in the direct SCF scheme involve further calculations of integrals, each iteration as time-consuming as for the previous state.

A number of procedures for open-shell calculations have been proposed since the first formulation of the basic theory by Roothaan in 1960 (ref. 3) (see, e.g., refs. 32 and 33 and references therein). Common to all these schemes is that they require the construction of at least one Fock matrix in AO basis per open shell in addition to the closed-shell matrix. In the direct SCF approach the Fock matrices must all be constructed simultaneously to avoid repeated calculation of integrals for each matrix. Therefore this part of the calculation will determine the space requirements of the program for more than two open shells. Furthermore, Fock-matrix construction may be a very time-consuming step in these calculations, and it is therefore preferable to work with a scheme that requires a minimum of matrices, such as that of Faegri and Manne.[33]

Most open-shell schemes also use a number of density matrices—at least one for each open shell—and the storage of these will further increase the space requirements of routines that have to construct a number of Fock matrices in AO basis simultaneously. Fortunately, the number of orbitals in each open shell is small—usually one—and most of this space may be saved by storing a total density matrix and the open-shell orbital expansion coefficients, and then constructing the appropriate open-shell density matrix element when needed.

Also for traditional SCF programs it may be advantageous to construct the Fock matrices simultaneously in order to reduce the reading of integral tapes; we therefore wish to point out how the space requirements of this particular task may be kept at a minimum. Consider the simple case of one open shell with one molecular orbital ψ_α:

$$\psi_\alpha = \sum_i C_{i\alpha}\phi_i = C_\alpha^+\phi \tag{30}$$

where ϕ is the column vector of basis functions (see Section IIC) and C_α is the column vector of expansion coefficients for ψ_α. The associated open-shell density matrix is

$$D^\alpha = C_\alpha C_\alpha^+ \tag{31}$$

For a given two-electron integral $(ij\ kl)$ the contribution to the AO basis Fock matrix $F^{A\alpha}$ for this open shell is of the form

$$F_{ij}^{A\alpha} := F_{ij}^{A\alpha} + (ij\ kl)D_{kl}^\alpha = F_{ij}^{A\alpha} + (ij\ kl)C_{k\alpha}C_{l\alpha} \tag{32}$$

This matrix is eventually prediagonalized, i.e., transformed to MO basis,

$$F^{M\alpha} = C^+F^{A\alpha}C \tag{33}$$

Only the αth row of this matrix is used; this may be written

$$F^{M\alpha} = C_\alpha^+F^{A\alpha}C \tag{34}$$

By carrying out the left-hand side of this transformation we may construct an intermediate Fock matrix—$F^{I\alpha}$—which is really a row vector

$$F^{I\alpha} = C_\alpha^+F^{A\alpha} \tag{35}$$

The contribution from the integral $(ij\ kl)$ to this matrix is then

$$F_j^{I\alpha} := F_j^{I\alpha} + (ij\ kl)C_{i\alpha}C_{k\alpha}C_{l\alpha} \tag{36}$$

Thus the open-shell Fock matrix may be stored as a row vector. The generalization of this scheme to open shells with several orbitals is straightforward, leading to strips of Fock matrices where the number of row vectors in each strip equals the number of orbitals in the open shell.

III. COMPUTATIONAL EXPERIENCE WITH DIRECT SCF CALCULATIONS

A. Program Description

The principles described above have been implemented in a computer program (DISCO) developed by us. Before discussing the computational experience gained with this program, we give here a brief description of its main parts. The modules are as follows.

1. Input processing routines. In addition to reading information about geometry and basis sets, this part also prepares all the tables needed for the point group operations and the symmetry handling.

2. Atomic SCF. To provide a good starting electron density, an SCF calculation is carried out in most cases for each atomic species in the molecule by a separate atomic SCF program functioning as a subroutine to the input processing section. No separate input is needed for this part of the program which is a dedicated adaption of routines from the atomic SCF program by Roos et al.[34] The atomic ground-state orbital configuration is automatically set by the program. For the purpose of constructing a molecular density for the atom, it suffices to do a calculation for a state average within the lowest-energy orbital configuration. By this simplification the vector coupling coefficients are easily generated automatically, and the only information that is supplied to this part of the program is the basis set and the number of electrons (assumed equal to the nuclear charge).

3. One-electron integrals. In the present version of the program, all one-electron integrals reside on mass storage during execution. No modifications are therefore necessary to adapt standard procedures. If necessary, however, the program may easily be modified such that also the one-electron integrals are recomputed in each SCF cycle, thereby further reducing the I/O requirements.

4. Symmetry-blocked basis. A matrix for transforming the basis set to a symmetry-blocked orthonormal basis is generated as described in Section IIC above. The routines and tables needed for this are already present from other parts of the program. In addition, one needs a routine to carry out a given symmetry operation on a linear combination of atomic orbitals, and a few sections of code handling the logistics of diagonalization, comparing, and sorting.

5. Two-electron integrals. This section basically consists of two parts, one for calculating integrals over shells batchwise, and one for administrating these calculations. The first part may be taken directly from a standard *ab-initio* Hartree–Fock program (we take ours from MOLECULE[35]), the second must be written so as to accommodate the Dacre scheme.

6. P- and Q-matrix generator. The two-electron parts of the skeleton Fock matrices are generated from the nonredundant integrals in a subroutine which is called up immediately after the calculation of a batch of integrals. For a closed-shell system this is a straightforward application of the method outlined by Dupuis and King[22]: however, for open-shell cases some thought must be given to the space requirements, as discussed in Section IID.

7. Solution of the Roothaan equations. This program section sets up the closed-shell Fock matrix and transforms this and the Q matrices—if any—to the orthonormal basis. If necessary, open-shell Fock matrices are created and merged with the closed-shell Fock matrix before diagonalization of the total Fock matrix, following the procedure described by Faegri and Manne.[33] A total density matrix is then constructed and the convergency is checked. The use of two square matrices in the transformation part determines the maximum space requirement for this part of the program.

B. Sample Calculations

The program DISCO described above has been used for studying a number of molecular systems. For the present work we discuss some sample calculations, which are presented in Table I. The pertinent parameters are the size and symmetry of the molecule, the number of primitive and contracted basis functions, and the average time for a full SCF cycle. The cycle times refer to two different types of computers. A number of calculations have been carried out on an IBM 3033, a rather large mainframe machine, whereas others were performed on minicomputers of type NORD 50 or NORD 500. The first series of calculations comprises a number of studies on sandwich compounds, involving a central metal atom and two cyclopentadiene rings. These molecules, which mostly have a D_{5h} or D_{5d} symmetry, are well suited for calculations with this program, which efficiently utilizes the molecular symmetry.

The calculations on ferrocene show the development of the cycle time as a consequence of a systematic increase of the basis: from a double-zeta basis set through one of triple/quadruple-zeta quality, and adding polarization functions on the carbon atoms. The same double-zeta and triple/quadruple-zeta basis sets were also used for decamethylferrocene. The larger of these calculations illustrates our contention that the direct SCF approach is the only practical means by which basis sets of this size can be handled in *ab-initio* calculations. The storage and retrieval of two-electron integrals with indices in a calculation with 500 basis functions would require a mass storage space of several gigabytes. Reading this amount of information in every iteration would also be a heavy burden for any computer system, and would require large amounts of I/O time. At present, we are not aware of any conventional *ab-initio* calculation using basis sets of nearly that size.

The advantages of being able to exploit a high molecular symmetry is readily apparent in the timing data for the two calculations on germanocene. Lowering the symmetry from D_{5d} to C_{2v} leads to an increase in cycle time by more than a factor of 4.

As an example of calculations on systems lacking symmetry, we list the results from a study of rotational barriers in the 2-haloacetamides. These molecules are poorly suited for our program, not only due to the total lack of symmetry, but also because of the compactness of the molecules, all nonhydrogen atoms lying within three bonds from one another. According to our experience, there is little to gain in calculations on such systems by

Table I. Average cycle times (min) for various sample calculations. The times given apply to cases where the ΔF features described by eqs. (18)–(20) have not been used, i.e., the integral threshold has been based on contributions to the total Fock matrix rather than to the increment. The test criterion given by eq. (7) was used, with threshold values of 10^{-8}–10^{-10} for contributions to the Fock matrix in AO basis.

		No of basis functions		Cycle time (min.)		
Molecule	Symmetry	primitive	contr	NORD 50	NORD 500	IBM 3033
Ferrocene	D_{5d}	244	161			8
"	"	425	261			33
"	"	515	351			78
Decamethyl ferrocene	"	464	301			33
"	"	805	501			195
Germanocene	D_{5d}	267	167	119	40	
"	C_{2v}	267	167	540	153	
Magnesocene	D_{5h}	254	174	100		
Hexafluoro-benzene	D_{3d} or D_{6h}	192	120	21[a]		
"	"	288	120	48		
"	"	360	228		130	
Hexachloro-benzene	"	318	222	180	59	
Hexabromo-benzene	"	558	306		276	
2Cl-aceta-mide	C1	134	84	183		
2Br-aceta-mide	"	183	113	779	220	

[a] The time for a corresponding open-shell case is 30 min.

eliminating insignificant integrals from the calculation. Still, it has been possible to carry out the calculations on a rather slow and old-fashioned minicomputer. This places medium-size *ab-initio* calculations within easy reach for the owner of a modern minicomputer.

In judging the efficiency of a computer program such as the one presented here, the timings presented in Table I may not be sufficient. It would undoubtedly be interesting to compare the results to those obtained with other programs. However, such comparison is complicated by the fact that molecules of the size for which the program is primarily designed are not easily amenable to calculation with other programs. In particular, this is the case if calculations on minicomputers are considered. A comparison must therefore be performed on small or medium-sized molecules, or on large mainframe computers, conditions which will bias a comparison in favor of conventional methods. It is interesting to note, therefore, that even in such cases the performance of the direct SCF approach is comparable to that of other methods. Thus, calculations with a 4-31 basis on the benzene molecule with the DISCO and the GAUSSIAN-76 (ref. 36) programs both needed 172 s to converge after seven iterations. The trial MOs in the GAUSSIAN-76 calculations were taken from a converged STO-3G calculation, whereas the DISCO calculation used no trial vector input but started automatically from atomic densities. With an energy-optimized $7s3p$ basis, where the restriction of equal $2s$ and $2p$ exponents is removed, the timings were 122 and 304 s, respectively, in favor of the DISCO program. Similar experience has been gained from comparison with other programs, where the timing data indicate that the direct SCF approach is an interesting alternative to conventional methods. It should be emphasized, however, that performance optimization in this respect has not been the main goal of the present work. The advantages of our method are better illustrated, in our opinion, by the calculations on decamethylferrocene and hexabromobenzene with 501 and 306 basis functions, respectively, the latter being geometry optimized at a reasonable cost on a minicomputer.

C. Distributions of Integrals

It is normally assumed that the time required for integral evaluation in an *ab-initio* calculation is roughly proportional to n^4, where n is the number of basis functions. However, if we consider the two smallest calculations on C_6F_6 shown in Table I, it is seen that a 50% increase in the number of primitives causes the cycle time to increase by only 129%, which is more consistent with an n^2 dependence. It is clear that it will be of great importance to be able to avoid calculation of insigificant integrals as much as possible. For this reason it is of interest to have some insight into the size distribution of two-electron integrals in some different molecules.

As a simple test case we have chosen a model system of nitrogen atoms in a linear array, studied with a double-zeta basis. The number of atoms was varied from 2 to 16, with an N—N distance of 1.10 Å. The distribution of integrals according to their absolute magnitude if shown in Figure 1 for the molecules N_2, N_4, and N_8. (In fact, only the radial part of the integrals is used, since the distribution of the angular factors in a linear molecule is rather atypical for general molecules and would result in an abnormally large number of zero integrals.) Figure 2 shows cumulative diagrams of the same distributions as in Figure 1. The curves are seen to flatten out more and more when the chain length increases, indicating that the fraction of the total number of integrals that need to be calculated should decrease rapidly with increasing size of the molecule. If the elimination criterion is set to 10^{-8} (which has been found to be a reasonable number

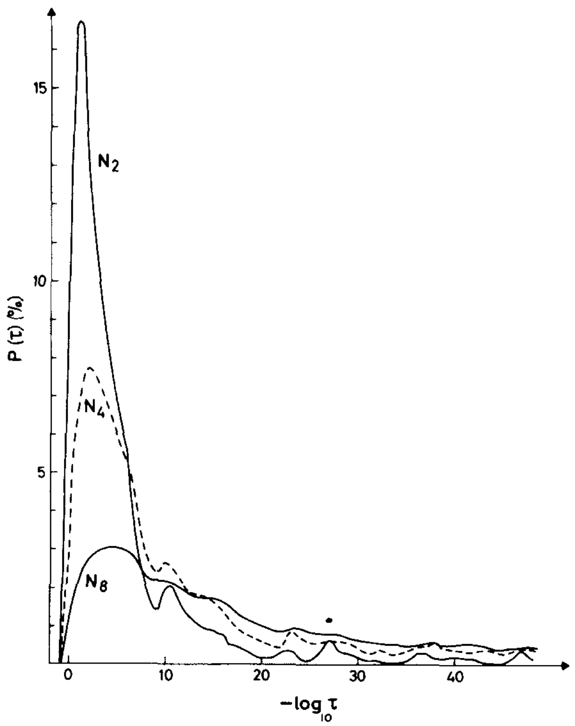

Figure 1. Integral distribution in percent of total for the linear model systems N_2, N_4, and N_8.

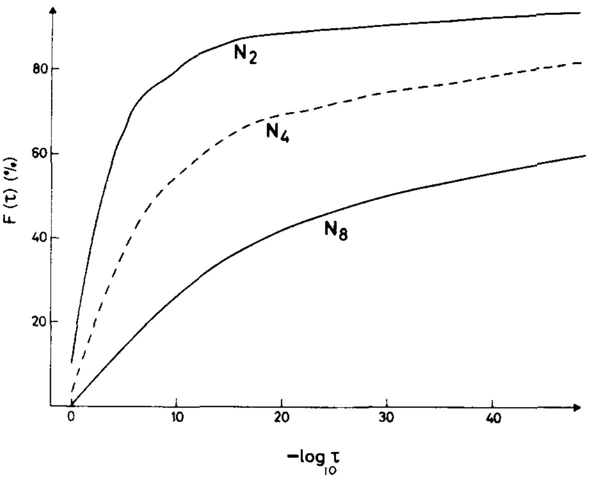

Figure 2. Cumulative integral distributions for the cases in Fig. 1.

in many cases), then the cumulative distribution shows that 77% of the two-electron integrals in N_2 must be calculated, but only 22% for N_8. This implies that if the size increase of the basis set is of the type demonstrated in this chain lengthening, then the cycle time dependence (roughly proportional to the number of integrals evaluated) will certainly be less than n^4. This is further demonstrated in Figure 3, which shows the number of integrals to be calculated as a function of the chain length for some different values of the elimination threshold. The power dependence is obtained from these plots as the slope of the lines. For a threshold of 10^{-5} the calculations suffer a variational collapse for more than eight atoms, but with $\tau = 10^{-8}$ a limiting slope of 2.2 is attained on the outer part of the curve. This value is expected to approach 2.0 for long chains, as the two-center Coulomb integrals are the only ones that will remain significant. It is also interesting to note that

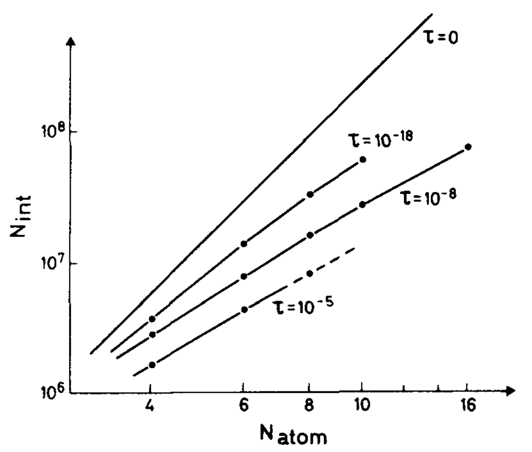

Figure 3. Number of integrals to be calculated as a function of the chain length for different values of the elimination threshold τ.

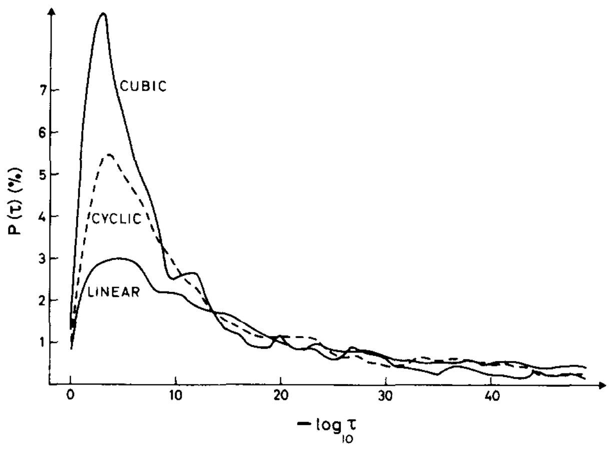

Figure 4. Integral distribution in percent of total for N_8 model systems of three different geometries.

even with a very small threshold, i.e., $\tau_0 = 10^{-18}$, the number of integrals increases slower than n^4. This is of course a feature from which any conventional method will also benefit, to the extent that the time used for evaluation of GTO integrals is the limiting factor.

The choice of a simple linear model system growing in one dimension only is optimal for the integral elimination criterion considered here. Alternatively, the system may grow in two dimensions (planar molecules) or in all three dimensions (clusters). We have investigated the integral distribution for three isomers of the N_8 model system: the linear form considered above, a planar cyclic form (analogous to cyclooctatetraene), and a form equivalent to the cubane molecule, with one atom at each corner of a cube. The resulting distributions are shown in Figure 4, and the cumulative distributions in Figure 5. With an elimination criterion of 10^{-8}, one will have to calculate about twice as many integrals for the

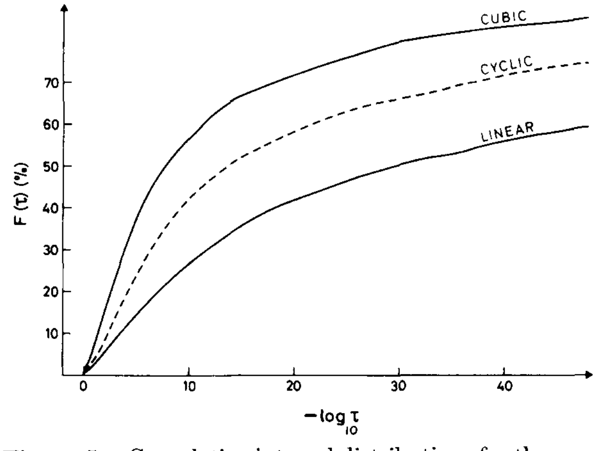

Figure 5. Cumulative integral distributions for the cases in Fig. 4.

cubic form as for the linear one.

From the discussion in this and the previous section, it is evident that the direct SCF approach has definite advantages in calculations on chain-like molecules with large interatomic distances, and for molecules with high symmetry. However, even for smaller molecules the fraction of two-electron integrals that need to be calculated will usually be much less than 50%, if an efficient elimination algorithm is employed. It should also be pointed out that the above considerations apply to cases where the full Fock matrix is calculated, i.e., to one of the first cycles of a calculation. When the ΔF procedure described in Section IIA is employed, the savings are much larger. As convergence is reached, the matrix $(D_m - D_{m-1})$ rapidly approaches zero, and so do, of course, the number of integrals that need to be calculated when this technique is used.

IV. CONCLUSIONS

Our experience with molecular *ab-initio* calculations in a direct SCF scheme shows this method to be competitive with the traditional approach in a modern computing environment. There are two situations where the method has particular advantages. For calculations using very extensive basis sets the storage and handling of large numbers of integrals in the traditional approach is avoided. For minicomputers, which often have ample memory space but less sophisticated mass-storage and peripheral equipment, the direct SCF procedure is also superior. Our calculations show that with a suitable minicomputer it is possible to carry out calculations that would be considered a very heavy task even for a large mainframe system with a traditional approach. The absence of tape-mounting and special disk packs for storing large amounts of integrals also makes the program very comfortable for the user.

Our implementation of the direct SCF approach draws heavily on reduction of the number of integrals by efficient use of molecular symmetry and by elimination of "insignificant" integrals. The use of full molecular symmetry also yields advantages in the diagonalization of the Fock matrix and in the treatment of open-shell systems. For many molecules a subgroup of the full molecular point group will suffice to generate the molecule from the minimum set of unique atoms (e.g., D_{2d} for methane), while the symmetry-blocking and the open-shell calculations are simplified if the full symmetry is used. The program is therefore capable of using two different molecular point groups in cases where this is advantageous.

Finally, we note that the use of atomic densities as a starting point eliminates the guesswork involved in making up a set of trial vectors when no old density matrix from a related geometry or electronic state is available. All these features make for a program which is simple to use and requires a minimum of input and user interaction during execution.

The direct SCF scheme may also be interfaced with an energy-gradient package. The integrals needed for the energy gradient are never stored during the calculation, and conceptually this part of the program will be very similar to the RHF part. The development of such a package is currently in progress in our group.

The authors are indebted to Dr. H. P. Lüthi at the University of Zürich for valuable assistance in the development of the program, improving its performance in several respects, and also for the results obtained on the IBM-3033 computer.

References

1. C. C. J. Roothaan, *Rev. Mod. Phys.*, **23,** 69 (1951).
2. G. G. Hall, *Proc. R. Soc. Sect. A*, **208,** 328 (1951).
3. C. C. J. Roothaan, *Rev. Mod. Phys.*, **32,** 179 (1960).
4. E. Clementi, private communication.
5. J. J. Kaufman, P. C. Hariharan, and H. E. Popkie, in Proceedings of the 1981 Texas Conference on Theoretical Approaches to Chemical Dynamics, 1981, p. 62.
6. B. O. Roos, *Chem. Phys. Lett.*, **15,** 153 (1972); B. O. Roos, in *Computational Techniques in Quantum Chemistry and Molecular Physics*, G. H. F. Diercksen, B. T. Sutcliffe, and A. Veillard, Eds. Reidel, Dordrecht, 1975, p. 251.
7. H. F. Schaefer and W. H. Miller, *Comput. Chem.*, **1,** 85 (1976).
8. See, e.g., H. P. Lüthi, J. Ammeter, J. Almlöf, and K. Korsell, *Chem. Phys. Lett.*, **69,** 540 (1980); S. Sæbo and J. Almlöf, *Acta Chem. Scand. A*, **34,** 650 (1980).
9. J. C. Phillips and L. Kleinmann, *Phys. Rev.*, **116,** 287 (1959); C. S. Ewig, R. Osman, and J. R. van Waser, *J. Chem. Phys.*, **66,** 3557 (1971).
10. L. R. Kahn and W. A. Goddard III, *J. Chem. Phys.*, **56,** 2685 (1972); G. Das and A. C. Wahl, *J. Chem. Phys.*, **64,** 4672 (1976); C. Teichteil, J. P. Malrieu, and J. C. Barthelat, *Mol. Phys.*, **33,** 181 (1977).
11. V. Bonifacic and S. Huzinaga, *J. Chem. Phys.*, **60,** 2779 (1974); V. Bonifacic and S. Huzinaga, *J. Chem. Phys.*, **62,** 1509 (1975); D. McWilliams and S. Huzinaga, *J. Chem. Phys.*, **63,** 4678 (1975).
12. G. Das and A. C. Wahl, *J. Chem. Phys.*, **69,** 53 (1978); P. Hafner and W. H. E. Schwartz, *J. Phys. B*, **11,** 217 (1978).
13. P. J. Hay, W. R. Wadt, L. R. Kahn, and F. W. Bobrowicz, *J. Chem. Phys.*, **69,** 984 (1978); Y. S. Lee, W. C. Ermler, and K. S. Pitzer, *J. Chem. Phys.*, **67,** 5861 (1977); **69,** 976 (1978).
14. R. Ahlrichs, *Theor. Chim. Acta*, **33,** 157 (1974).
15. G. Karlström, *J. Comput. Chem.*, **2,** 33 (1981).
16. D. L. Wilhite and R. N. Euwema, *Chem. Phys. Lett.*,

20, 610 (1973).

17. D. L. Wilhite and R. N. Euwema, *J. Chem. Phys.*, **61,** 375 (1975).
18. J. J. Kaufman, H. E. Popkie, and P. C. Hariharan, in *Computer Assisted Drug Design*, E. C. Olson and R. E. Christoffersen, Eds., ACS Symp. Ser. 112, 1979, p. 415.
19. H. E. Popkie and J. J. Kaufman, *J. Chem. Phys.*, **66,** 4827 (1977); H. E. Popkie and J. J. Kaufman, *Int. J. Quantum Chem.*, **S2,** 279 (1975).
20. P. Dacre, *Chem. Phys. Lett.*, **7,** 47 (1970).
21. M. Elder, *Int. J. Quantum Chem.*, **7,** 75 (1973).
22. M. Dupuis and H. F. King, *Int. J. Quantum Chem.*, **11,** 613 (1977).
23. A. Veillard, in *Computational Techniques in Quantum Chemistry and Molecular Physics*, G. H. F. Diercksen, B. T. Sutcliffe, and A. Veillard, Eds., Reidel, Dordrecht, 1975, p. 201.
24. N. W. Winter and T. H. Dunning, *Chem. Phys. Lett.*, **8,** 169 (1971); T. A. Claxton and N. A. Smith, *Theor. Chem. Acta.*, **22,** 399 (1971); V. R. Saunders and I. H. Hillier, *Int. J. Quantum Chem.*, **7,** 699 (1973); I. Røeggen, *Chem. Phys. Lett.*, **22,** 140 (1973); C. C. J. Roothaan and P. S. Bagus, in *Methods in Computational Physics*, B. Alder, S. Fernbach, and M. Rotenberg, Eds., Academic, New York, 1963, Vol. 2.
25. P. Pulay, *Chem. Phys. Lett.*, **73,** 393 (1980).
26. G. B. Bacskay, *Chem. Phys.*, **61,** 385 (1981).
27. J. Douday, Y. Ellinger, R. Subra, and B. Levy, *J. Chem. Phys.*, **72,** 1452 (1980); B. Levy, *Chem. Phys. Lett.*, **4,** 17 (1969); **18,** 59 (1973); B. Levy, *Int. J. Quantum Chem.*, **4,** 297 (1970).
28. A. Banerjee and F. Grein, *Int. J. Quantum Chem.*, **10,** 123 (1976).
29. E. Dalgaard and P. Jørgensen, *J. Chem. Phys.*, **69,** 3833 (1978).
30. P. Siegbahn, J. Almlöf, A. Heiberg, and B. Roos, *J. Chem. Phys.*, **74,** 2384 (1981).
31. P. Bagus and U. Wahlgren, *Comput. Chem.*, **1,** 95 (1977); A. L. H. Chung and G. L. Goodman, *J. Chem. Phys.*, **56,** 4125 (1972).
32. R. Carbó and O. Gropen, *Adv. Quantum Chem.*, **12,** 159 (1980).
33. K. Faegri and R. Manne, *Mol. Phys.*, **31,** 1037 (1976).
34. B. Roos, E. Salez, A. Veillard, and E. Clementi, Special IBM Technical Report, IBM Research Laboratory, San José, CA, 1968.
35. J. Almlöf, USIP-Report 74-29, University of Stockholm, 1974.
36. W. J. Hehre, W. A. Lathan, R. Ditchfield, M. D. Newton, and J. A. Pople, Quantum Chemistry Program Exchange Program No. 368, Indiana University, Bloomington, IN.