A DETERMINANT BASED FULL CONFIGURATION INTERACTION PROGRAM

Peter J. KNOWLES and Nicholas C. HANDY

University Chemical Laboratory, Lensfield Road, Cambridge CB2 1EW, UK

Received 22 September 1988

The program FCI solves the Full Configuration Interaction (Full CI) problem of quantum chemistry, in which the electronic Schrödinger equation is solved exactly within a given one particle basis set. The Slater determinant based algorithm leads to highly efficient implementation on a vector computer, and has enabled Full CI calculations of dimension more than 10^7 to be performed.

PROGRAM SUMMARY

Title of program: FCI

Catalogue number: ABHV

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue)

Computer for which the program is designed and others on which it is operable: any machine with Fortran 77

Computer: Convex C210; *Installation*: Department of Theoretical Chemistry, University of Cambridge

Operating system: Convex Unix 6.2

Programming language used: Fortran 77

High speed storage required: problem dependent

No. of bits in word: preferably 64 for floating point arithmetic; at least 24 for integers

Peripherals used: disc, printer

No. of lines in combined program and test deck: 2852

Keywords: quantum chemistry, configuration interaction

Nature of physical problem
Benchmark calculations for quantum chemistry.

Method of solution
Diagonalisation of the full CI Hamiltonian matrix. A vectorised determinant-based full CI algorithm [1] is used to compute the actions of the Hamiltonian on a trial wavefunction.

Restrictions on the complexity of the problem
Size of problem limited by available CPU time and the requirement to hold in memory two vectors the length of the CI expansion.

Typical running time
Problem dependent.

Reference
[1] P.J. Knowles and N.C. Handy, Chem. Phys. Lett. 111 (1984) 315.

LONG WRITE-UP

1. Introduction

1.1. *The full CI problem and its application to quantum chemistry*

A full CI calculation represents the determination of the best possible quantum chemical wavefunction for a given one particle basis set. If the one particle basis functions are represented by η_s, $s = 1, 2, \ldots, M$, then one can form M orthonormal orbitals ϕ_i

$$\phi_i = \sum_{s=1}^{M} C_{si}\eta_s, \quad i = 1, 2, \ldots, M. \tag{1}$$

These orbitals can then be used to form N-electron configuration state functions (CSFs) $\{\Phi_I, I = 1, 2, \ldots, N_C\}$, which are antisymmetric and eigenfunctions of the total spin operator S^2. The number of CSFs which can be formed is given by the Weyl formula [1]

$$N_c = \frac{2S+1}{M+1}\binom{M+1}{\frac{1}{2}N - S}\binom{M+1}{M - \frac{1}{2}N - S}, \tag{2}$$

where N is the number of electrons, and $S(S+1)$ is the eigenvalue of S^2. The full CI wavefunction is represented by

$$\Psi = \sum_{I=1}^{N_c} c_I \Phi_I, \tag{3}$$

with the coefficients c_I determined by the variational secular equations

$$\langle \Phi_I | \hat{H} - E | \Psi \rangle = 0, \quad I = 1, 2, \ldots, N_C. \tag{4}$$

Solution of these equations is equivalent to finding an eigensolution of the Hamiltonian matrix **H** which is the representation of the Hamiltonian operator $\hat{H}$ in the complete basis of CSFs. The full CI energy E is invariant to the choice of orbitals $\{\phi_i\}$, although convergence of the iterative scheme used to solve (4) will be most rapid if there is one dominant CSF in the wavefunction, as is frequently the case if self-consistent field orbitals are used.

Since the development in 1984 of our algorithm [2] for the efficient solution of (4), large scale full CI calculations have had considerable impact in quantum chemistry, most notably in three areas:

a) Multiconfiguration SCF calculations. A very special, but extremely useful, type of multiconfiguration SCF wavefunction is that termed Complete Active Space SCF (CASSCF) [4–6]. This involves a full CI expansion within a given set of orbitals. Incorporation of our full CI algorithm into our MCSCF program has enabled very large MCSCF calculations to be performed, and we refer to the literature for further details [3].
b) "Benchmark" full CI calculations. Using the first general full CI program [7], full CI calculations on H_2O with a double zeta basis set [8], and also on Be_2 [9] were performed, and these became established benchmarks for quantum chemistry. That program was very slow. Bauschlicher and coworkers have been successful in using this newer full CI algorithm with larger (but modest) sized basis sets on a wide variety of molecules [10,11]. They have used the program FCI as a calibration tool for other ab initio methods (which can subsequently be used with much larger basis sets). These calibration calculations have served to show that multireference single and double excitation CI calculations can produce rather accurate potential energy and property surfaces, and have been useful in estimating errors in the many different approximate quantum chemical methods. We note that these calculations became possible through a combination of algorithm development and improvements in hardware; many of the benchmark full CI calculations were of dimension greater than 10^7 and required the availability of a large memory machine (a Cray-2 with 256 Mword).
c) Convergence of the Møller–Plesset perturbation series. The principal routine in FCI computes the representation in Slater determinants of the action of the Hamiltonian operator on

some trial wavefunction, also represented as a combination of determinants. This means that it is possible to solve the Møller–Plesset Rayleigh Schrödinger perturbation equations order by order; for each application of the Hamiltonian operator, one may compute the wavefunction to one order higher. In this way, it has been possible to investigate the convergence properties of the Møller–Plesset perturbation series, by computing for a number of different molecules the series up to, in some cases, 48th order. For details we refer to our original papers [12,13].

1.2. Technical considerations for large scale full CI calculations

The matrix elements of the Hamiltonian are given by

$$H_{IJ} = \left\langle \phi_I \middle| \hat{H} \middle| \phi_J \right\rangle = \sum_{ij} \gamma_{ij}^{IJ} h_{ij} + \sum_{\substack{ij\\kl}} \Gamma_{ij,kl}^{IJ} (ij \mid kl), \tag{5}$$

where γ_{ij}^{IJ}, $\Gamma_{ij,kl}^{IJ}$ are the one and two particle coupling coefficients, and h_{ij}, $(ij \mid kl)$ are the usual one and two electron integrals defined by

$$h_{ij} = \left\langle \phi_i(1) \middle| \hat{h}(1) \middle| \phi_j(1) \right\rangle, \tag{6}$$

where $\hat{h}$ is the appropriate one electron operator and

$$(ij \mid kl) = \left\langle \phi_i(1)\phi_k(2) \middle| r_{12}^{-1} \middle| \phi_j(1)\phi_l(2) \right\rangle. \tag{7}$$

In nearly all cases so far observed, the Hamiltonian matrix **H** is strongly diagonally dominant and very sparse. This has allowed quantum chemists to obtain low lying eigensolutions straightforwardly in a small number of iterations using the procedure of Davidson [14]. Instead of the full matrix **H**, one requires only its diagonal elements $d_I = H_{II}$ and an efficient procedure to compute the action of **H** on some trial vector $\boldsymbol{c}$

$$g_I = \sum_{J}^{N_C} H_{IJ} c_J. \tag{8}$$

Thus in each iteration, one must obtain once all of the coupling coefficients γ_{ij}^{IJ}, $\Gamma_{ij,kl}^{IJ}$ and combine them with the appropriate integrals and coefficients. The large number of two particle coupling coefficients are given by

$$\Gamma_{ij,kl}^{IJ} = \tfrac{1}{2} \left\langle \phi_I \middle| \hat{E}_{ij} \hat{E}_{kl} - \delta_{jk} \hat{E}_{il} \middle| \phi_J \right\rangle \tag{9}$$

with $\hat{E}_{ij}$ being the usual single particle excitation operators. In the first implementation of a general full CI procedure [7], the coupling coefficients were obtained by straightforward consideration of all double excitations arising from each configuration Φ_J in turn. A similar procedure has been adopted more recently [15]. Our approach follows that of Siegbahn [16], who recognised the value of using the completeness relation

$$\Gamma_{ij,kl}^{IJ} = \tfrac{1}{2} \sum_{K} \gamma_{ik}^{IK} \gamma_{kl}^{KJ} - \tfrac{1}{2} \delta_{jk} \gamma_{il}^{IJ} \tag{10}$$

(8) is then evaluated as (considering only highest order terms)

$$D_{ij}^{K} = \sum_{J} \gamma_{ij}^{KJ} c_J, \tag{11}$$

$$E_{ij}^{K} = \sum_{kl} (ij \mid kl) D_{kl}^{K}, \tag{12}$$

$$g_I = \sum_{\substack{ij\\K}} \gamma_{ij}^{IK} E_{ij}^{K}. \tag{13}$$

The features of this algorithm as compared with the conventional approach of ref. [7] are

a) Only one particle coefficients γ_{ij}^{IJ} are required, and not the far more numerous two particle quantities $\Gamma_{ij,kl}^{KL}$. This leads to program simplification, and, for the case that the coupling coefficients are stored on a disc rather than being recomputed when needed, reduction in input/output time.

b) For large cases, the cost is entirely dominated by (12). This can be formulated as a matrix multiplication, allowing optimum efficiency on certain classes of supercomputer.

c) Again because (12) is dominant, optimum exploitation of the index permutation symmetry of the two electrons integrals is possible.

Further detail of these considerations may be found in ref. [15].

2. The determinant full CI algorithm

The first use of the scheme given in (1)–(13) was that of Siegbahn [16]. In his approach, the one particle coupling coefficients γ_{ij}^{IJ} are constructed in the Gelfand CSF basis, sorted into the required order, and held on external (disc) storage; later implementations compute the one particle coupling coefficients directly when required [17,18]. Our approach [2] involves replacing the CSF basis by the complete set of Slater determinants with $S_z = S$, the required spin quantum number. This set spans exactly all CSFs with spin quantum numbers S, $S+1, \ldots, \frac{1}{2}N$ and thus in general is larger than the CSF basis. The ratio of the number of such determinants, N_D, to the number of CSFs is given by [19]

$$\frac{N_D}{N_C} = \frac{(M - \frac{1}{2}N + S + 1)(\frac{1}{2}N + S + 1)}{(2S+1)(M+1)}. \tag{14}$$

For example, for $S = 0$, in the limit of large M, this ratio tends to $\frac{1}{2}N + 1$, and for $S = 1$, $\frac{1}{3}(\frac{1}{2}N + 2)$; for finite M, the ratio is somewhat reduced. Even though we use a larger expansion set, there are certain advantages in using determinants; the coupling coefficients are simply 1, -1 or 0, and furthermore it is possible to design and addressing scheme for the configurations which allows construction and use of the coupling coefficients to be vectorized.

2.1. Addressing of the CI expansion

A determinant is constructed from a product of a string of N_α orbitals associated with α spin and a string of N_β orbitals associated with β spin. We define, separately for α and β string, addressing arrays Z, given by

$$Z^\alpha(k, l) = \sum_{m=M-l+1}^{M-k} \left[\binom{m}{N_\alpha - k} - \binom{m-1}{N_\alpha - k - 1} \right]$$

$$(M - N_\alpha + k \geq l \geq k;\ k < N_\alpha), \tag{15}$$

$$Z^\alpha(N_\alpha, l) = l - N_\alpha,$$

where k refers to an electron and l to an orbital. Any string I is identified by a list of occupied orbitals $\{\phi_{I_i},\ i = 1, 2, \ldots, N_\alpha\}$ in strictly ascending order, i.e., $I_1 < I_2 < \cdots$. The address of the string is then given by

$$A_I^\alpha = 1 + \sum_{i=1}^{N_\alpha} Z^\alpha(i, I_i) \tag{16}$$

and the addressing is lexical without gaps. We note that this is similar to the graphical addressing schemes which are common in unitary group and symmetric group formulations of the electron correlation problem [20,21,17,22]. (i, I_i) are equivalent to graph vertices, and $Z(i, I_i)$ corresponds to a partial weight array.

Once the addresses of the α and β strings which form a given Slater determinant have been obtained, the determinant can be addressed as the element of a rectangular array $C(A_J^\beta, A_I^\alpha)$, where I^α, J^β represent the α, β strings. The means that operations on the α string alone can be performed for all β strings in a vector loop and vice versa. Non-degenerate point group symmetry can be easily incorporated into this address scheme by the use of index vectors for the string addresses; we omit details for clarity.

2.2. One particle coupling coefficients

Eqs. (11), (13) require the evaluation together of all non-zero coupling coefficients γ_{ij}^{IK} for a given determinant Φ_K, together with the associated configuration and integral addresses I and ij. Because we use Slater determinants, we are able to take advantage of the fact that the orbital excitation operator $\hat{E}_{ij}$ is just the sum of excitation operators for α and β spin orbitals, $\hat{E}_{ij}^\alpha + \hat{E}_{ij}^\beta$. This in turn means that the coupling coefficients separate into two classes, those where the α string is excited, but the β string unchanged, and vice versa. Having determined a given α spin excitation and its associated numerical value (± 1), we then have coupling coefficients for all possible β strings

$$\left\langle I^\alpha J^\beta \middle| \hat{E}_{ij}^\alpha \middle| K^\alpha J^\beta \right\rangle = \left\langle I^\alpha \middle| \hat{E}_{ij}^\alpha \middle| K^\alpha \right\rangle$$

$$\text{for all } \beta \text{ strings } J^\beta. \tag{17}$$

In our program, we construct and store all possible α and β string excitations, and these are then used in vector loops over β and α strings, respectively, in the construction of (1) and (13), as shown in the next section.

2.3. The determinant CI algorithm

For the sake of completeness, we repeat here the structure of the algorithm used in our program and presented in our earlier publication [2]. Spatial symmetry introduces extra loops outside this structure, but we omit these for clarity.

Split α strings and β strings which will form the intermediate determinants Φ_K of (12) into blocks, according to how much memory is available to hold blocks of the matrices **D**, **E**.

```
Loop over blocks of $\alpha$ strings $I^\alpha$
    Form list of single replacements $\tilde{I}^\alpha = \hat{E}_{ij} I^\alpha$
    Loop over blocks of $\beta$ strings $J^\beta$
        Form list of single replacements $\tilde{J}^\beta = \hat{E}_{ij} J^\beta$
        Loop over $I^\alpha$ in block
            Loop over $\tilde{I}^\alpha$
                Loop over $J^\beta$ in block
                    $D(J^\beta, I^\alpha, ij) \leftarrow \pm C(J^\beta, \tilde{I}^\alpha)$    (18)
        Loop over $J^\beta$ in block
            Loop over $\tilde{J}^\beta$
                Loop over $I^\alpha$ in block
                    $D(J^\beta, I^\alpha, ij) \leftarrow \pm C(\tilde{J}^\beta, I^\alpha)$    (19)
        Loops over $J^\beta$ in block, $I^\alpha$ in block, $ij$, $kl$
            $E(J^\beta, I^\alpha, kl) \leftarrow D(J^\beta, I^\alpha, ij)$
                    $\times (ij \mid kl)$    (20)
        Loop over $I^\alpha$ in block
            Loop over $\tilde{I}^\alpha$
                Loop over $J^\beta$ in block
                    $g(J^\beta, \tilde{I}^\alpha) \leftarrow \pm E(J^\beta, I^\alpha, ij)$    (21)
        Loop over $J^\beta$ in block
            Loop over $\tilde{J}^\beta$
                Loop over $I^\alpha$ in block
                    $g(\tilde{J}^\beta, I^\alpha) \leftarrow \pm E(J^\beta, I^\alpha, ij)$    (22)
```

In the above, (18, 19) correspond to eq. (11), (20) to eq. (12) and (21, 22) to eq. (13).

2.4. Diagonal elements of the Hamiltonian matrix

In each iteration of the Davidson diagonalisation procedure [14], the next correction to the wavefunction is estimated using first order perturbation theory. This involves division by the diagonal matrix elements d_I of the Hamiltonian matrix. As discussed in ref. [2], it is essential that d_I for all determinants with the same orbital occupancy are identical; in general this is not true, since they differ by exchange integrals, but if this condition is not enforced, the correction to the wavefunction is not an eigenfunction of $\hat{S}^2$. The energy of each determinant is given by

$$d_I = \sum_i n_i h_{ii} + \tfrac{1}{2} \sum_{ij} n_i n_j (ii \mid jj) + E_{\text{exch}}, \tag{23}$$

where the exchange energy is

$$E_{\text{exch}} = -\tfrac{1}{2} \sum_{ij} \left(n_i^\alpha n_j^\alpha + n_i^\beta n_j^\beta \right) (ij \mid ji). \tag{24}$$

In the above, $n_i^\alpha = 0$ or 1 represents the occupation number of α spin orbital ϕ_i in determinant Φ_I, and $n_i = n_i^\alpha + n_i^\beta$ is the total occupancy of spatial orbital ϕ_i. In our procedure, we replace the true exchange energy E_{exch} by

$$\tilde{E}_{\text{exch}} = -\tfrac{1}{4} \sum_{ij} n_i n_j (ij \mid ji) - \tfrac{1}{4} \sum_i N_i (ii \mid ii) - \tfrac{1}{4} F \sum_{i \neq j} N_i N_j (ij \mid ji) \tag{25}$$

with $N_i = n_i (2 - n_i)$ being 1 or 0 according to whether or not ϕ_i is single occupied. The averaging factor F is given by

$$F = \frac{4S^2 - N_{\text{open}}}{N_{\text{open}} (N_{\text{open}} - 1)} \tag{26}$$

and $N_{\text{open}} = \Sigma_i N_i$, the number of singly occupied orbitals. This formula (25) is the average exchange energy of all determinants with the same spatial orbital occupancy. This approach has been found to be superior to our original approximate formulation [2].

Given the spin averaged denominators, we have found it possible to obtain correct spin eigenfunction roots of the Hamiltonian matrix, even though there may be lower energy roots having an undesired higher S eigenvalue. It is only necessary to provide a starting estimate for the wavefunction which has the correct spin symmetry; the program ensures that the lowest energy CSF of correct S is used, as described in ref. [2].

3. Description of program and data input

3.1. Installation of the program

The program as supplied consists of a file (fci.f) containing most of the Fortran code, and a number of alternative files (standard.f, cray.f, ibm.asm, convex.c, vax.f) which contain machine dependent code. standard.f contains standard Fortran 77 versions of the routines GMAINV, FMAIN (dynamic memory allocation) and SECOND (elapsed cpu time), which may be used as a template for implementing the program on any machine. cray.f, ibm.asm, convex.c, vax.f contain, respectively, Fortran, IBM assembly, C and Fortran versions of these routines for Cray(COS), IBM(MVS or CMS), Convex(Unix) and VAX(VMS) computers.

Each subroutine contains the declaration IMPLICIT DOUBLE PRECISION (A–H, O–Z); the intention is to use 64 bit arithmetic, and so, for example, on a Cray, where DOUBLE PRECISION is not what is required, one should use the compiler option OFF = P. The only extension to Fortran 77 used (apart from the machine dependent code) is NAMELIST, which occurs in subroutine INPDAT, described below.

The only part of the program which may need adjusting for optimum efficiency on a particular machine is subroutine FCMXM. This performs the matrix multiplication of (12). Depending on the machine architecture, one may take either K or ij as the index of the innermost loop in the middle product form of the matrix multiplication [23]; the K loop is usually longer, but vectorization along ij allows the exploitation of the sparsity of the matrix $\mathbf{D}$. On some machines it is advantageous to explicitly test this sparsity and then make the decision of loop order. Alternatively, on some machines it may be best to implement the inner product algorithm [23]. Finally, fast machine code subroutines may be substituted or called. The supplied version of FCMXM is not necessarily optimum for any machine, but contains code and comments which reflect some of these considerations.

Apart from FCMXM, subroutines FCDIAG and FSIGMA contain Fortran DO loops in which a significant fraction of execution time is spent. On a vector machine, it should be verified that the Fortran compiler has produced vector code for these loops. Where necessary, the code already contains appropriate directives for Cray and Convex Fortran compilers.

3.2. Organisation of the program

As well as various initialisation tasks, the main program calls three important subroutines:

a) INPDAT reads input data.
b) INITC sets up addressing arrays for the CI wavefunction and the integrals, and reads and preprocesses the integrals.
c) DAVIDS is the controlling routine for Davidson diagonalisation of the Hamiltonian matrix. The main program allocates two vectors whose length is the size of the determinant CI expansion, and passes these to DAVIDS. For a large case, this is the principle memory requirement, although some extra space for intermediate arrays (particularly segments of the matrices $\mathbf{D}$ and $\mathbf{E}$) should be allowed in order not to degrade efficiency.

Subroutine DAVIDS calls the following subroutines:

a) FCDIAG returns the diagonal elements of the Hamiltonian matrix.
b) FSIGMA returns the action $\boldsymbol{g} = \mathbf{H}\boldsymbol{c}$ of the Hamiltonian matrix $\mathbf{H}$ on a supplied vector $\boldsymbol{c}$.

This modular structure should allow the replacement of DAVIDS by some other procedure which requires these operations, if required.

3.3. Data input

The date for the program consists of two parts. The first is a specification of the dimensions of the problem, for example numbers of orbitals and electrons, and also any options for the program run; this data is read from Fortran stream 5 by subroutine INPDAT. Secondly, the one and two electron integrals in the molecular orbital basis are read by subroutine INPINT.

3.3.1. Subroutine INPDAT

INPDAT performs a Fortran NAMELIST read from stream 5, which should contain data in the format required by the particular implementation of NAMELIST. The NAMELIST name is FCI. The following variables, which are integers except where marked otherwise, may be specified:

Variable	Default value	Description
NORB	–	Number of orbitals
NELEC	–	Number of electrons
MS2	0	$2S$, where S is the spin quantum number
ISYM	1	Spatial symmetry of wavefunction
ORBSYM-(1:NORB)	1, 1,...	Spatial symmetries of orbitals
IPRTIM	−1	If 0, print additional CPU timing analysis
INT	5	Fortran stream from which integrals will be read
MEMORY	100000	Size of workspace array in floating point words
CORE	0.0	Core energy (may also be given in integral file) (*floating point*)
MAXIT	25	Maximum number of iterations in Davidson diagonalisation
THR	10^{-5}	Convergence threshold for Davidson diagonalisation (*floating point*)
THRRES	10^{-1}	Threshold for printing final CI coefficients (*floating point*)
NROOT	1	Number of eigenvalues of Hamiltonian to be found

3.3.2. Subroutine INPINT

INPINT reads and interprets the integrals according to the following Fortran code:

```
1.    READ (INP, * ,END = 99)Z,I,J,K,L
      IF (K.NE.0) THEN
C···  Z interpreted as two electron integral
C     (IJ|KL)
...
      ELSE IF (I.NE.0) THEN
C···  record contained Z I J 0 0
C···  Z interpreted as one electron integral H(I,J)
...
      ELSE
C···  record contained Z 0 0 0 0
C···  Z interpreted as core energy
...
      END IF
      GOTO 1
99    CONTINUE
```

Only those integrals which are unique (i.e. not equal by permutational symmetry) and non-zero need be specified.

4. Test run

The files hf.dat, hf.out contain, respectively, the data input and resultant output for a full CI calculation for HF with a double zeta basis set [10,24] at a bond length of 1.733 bohr. The $1a_1$ orbital is not correlated, and therefore it appears only through the replacement of the one electron integrals by a core Fock operator. The number of determinants is 27252, equivalent to 8674 CSFs. The input consists of the following NAMELIST data

```
$ FCI NORB = 11, NELEC = 8,
  ORBSYM = 1,1,1,1,1,1,1,2,2,3,3,
  MEMORY = 150000,
$
```

followed by the integrals as described above. The output contains

a) Size of memory used for temporary arrays.
b) Number of orbitals M = NORB, broken down into the number of each symmetry.
c) Number of electrons N = NELEC, and spin quantum number $S = \frac{1}{2}$MS2.
d) Number of orbital pairs ij for each spatial symmetry.

e) Number of α strings of each symmetry, followed by number of β strings.
f) Number of determinants of each symmetry.
g) Core energy, CORE
h) Spatial symmetry of wavefunction, ISYM.
i) Diagonalization parameters MAXIT, THR, THRRES, NROOT.
j) Description of the determinants making up the trial wavefunction. This is the single CSF of lowest energy.
k) For each iteration of the diagonalisation, the iteration number, the current root of the Hamiltonian being improved, the current CPU time, convergence test value and current energy.
l) The final energy and wavefunction; for each determinant with coefficient greater than THRRES, a line is printed containing the coefficient, a list of occupied α orbitals and a list of occupied β orbitals.

References

[1] J. Paldus, J. Chem. Phys. 61 (1974) 5321.
[2] P.J. Knowles and N.C. Handy, Chem. Phys. Lett. 111 (1984) 315.
[3] P.J. Knowles and H.-J. Werner, Chem. Phys. Lett. 115 (1985) 259.
[4] K. Ruedenberg, L.M. Cheung and S.T. Elbert, Intern. J. Quantum Chem. 16 (1979) 1069.
[5] B. Roos, P. Taylor and P.E.M. Siegbahn, Chem. Phys. 48 (1980) 157.
[6] P.E.M. Siegbahn, J. Almlöf, A. Heiberg and B.O. Roos, J. Chem. Phys. 74 (1981) 2384.
[7] N.C. Handy, Chem. Phys. Lett. 74 (1980) 280.
[8] P. Saxe, H.F. Schaefer and N.C. Handy, Chem. Phys. Lett. 79 (1981) 202.
[9] R.J. Harrison and N.C. Handy, Chem. Phys. Lett. 95 (1983) 386.
[10] C.W. Bauschlicher, S.R. Langhoff, P.R. Taylor, N.C. Handy and P.J. Knowles, J. Chem. Phys. 85 (1986) 1469.
[11] C.W. Bauschlicher and P.R. Taylor, J. Chem. Phys. 85 (19860 610.
C.W. Bauschlicher and S.R. Langhoff, J. Chem. Phys. 86 (1987) 5595.
C.W. Bauschlicher and P.R. Taylor, Theor. Chim. Acta 71 (1987) 263.
C.W. Bauschlicher, S.R. Langhoff, H. Partridge and P.R. Taylor, J. Chem. Phys. 85 (1986) 3407.
C.W. Bauschlicher and P.R. Taylor, J. Chem. Phys. 85 (1986) 2779.
C.W. Bauschlicher, S.R. Langhoff and P.R. Taylor, J. Chem. Phys. 88 (1988) 2540.
C.W. Bauschlicher and S.R. Langhoff, J. Chem. Phys. 87 (1987) 4665.
C.W. Bauschlicher and P.R. Taylor, J. Chem. Phys. 86 (1987) 858.
[12] P.J. Knowles, K. Somasundram, N.C. Handy and K. Hirao, Chem. Phys. Lett. 113 (1985) 8.
[13] N.C. Handy, P.J. Knowles and K. Somasundram, Theor. Chim. Acta 68 (1985) 87.
[14] E.R. Davidson, J. Comput. Phys. 17 (1975) 84.
[15] J. Olsen, B.O. Roos, P. Jørgensen and H.J.Aa. Jensen, J. Chem. Phys. 89 (1988) 2185.
[16] P.E.M. Siegbahn, Chem. Phys. Lett. 109 (1984) 417.
[17] W. Duch, Chem. Phys. Lett. 124 (1986) 442.
[18] P.J. Knowles and H.-J. Werner, Chem. Phys. Lett. 145 (1988) 514.
[19] J. Paldus and B. Jeziorski, Theor. Chim. Acta 73 (1988) 81.
[20] I. Shavitt, Intern. J. Quantum Chem. Symp. 11 (1977) 131; Intern. J. Quantum Chem. Symp. 12 (1978) 50.
[21] W. Duch and J. Karwowski, Intern. J. Quantum Chem. 22 (1982) 783.
[22] W. Duch, GRMS or Graphical Representation of Model Spaces (Springer, Berlin, 1986).
[23] R.W. Hockney and C.R. Jesshope, Parallel Computers (Adam Hilger, London, 1981).
[24] T.H. Dunning, J. Chem. Phys. 53 (1970) 2823.
S. Huzinaga, J. Chem. Phys. 42 (1965) 1293.

TEST RUN OUTPUT

```
PROGRAM * FCI  (Determinant based Full CI)     Author: P.J. Knowles, 1984

Variable memory set to     150000 words
Active orbitals:            11( 7 2 2 0 0 0 0 0)
Active electrons:            8
Spin quantum number:         0.0
Orbital pairs:                 34     14     14      4      0      0      0      0
Strings:                       78     84     84     84      0      0      0      0
                               78     84     84     84      0      0      0      0
Determinants:                     27252         27216         27216         27216
                                      0             0             0             0
Core energy:                   -71.436373535000

Space symmetry:              1
Maximum iterations:         25
Convergence threshold:        0.0000100
Output threshold:             0.0500000
Number of roots:             1

Initial configuration generated:
  20197       1.0000000   -100.0219723

It  Tr    CPU  Convergence   Energy

 1   1    6.6  1.00000000   -100.021972267
 2   1   11.3  0.17074489   -100.136780639
 3   1   18.2  0.04621300   -100.146030569
 4   1   23.8  0.01664914   -100.147117321
 5   1   28.5  0.00471317   -100.147194228
 6   1   33.3  0.00121456   -100.147201764
 7   1   38.1  0.00044849   -100.147202456
 8   1   43.0  0.00012907   -100.147202525
 9   1   48.0  0.00003391   -100.147202530
10   1   53.1  0.00001155   -100.147202531
11   1   58.2  0.00000303   -100.147202531
State 1     Energy   -100.147202530670
Correlation energy     -0.125230263724

Final CI vector

      0.981557124440  1  2  8 10  1  2  8 10
/EOF
Variable memory released
```