Theor Chem Acc (2001) 106:339–351
DOI 10.1007/s002140100285

Theoretical
Chemistry Accounts

Regular article

Identification of deadwood in configuration spaces through general direct configuration interaction

Joseph Ivanic, Klaus Ruedenberg

Ames Laboratory USDOE and Department of Chemistry, Iowa State University, Ames, IA 50011, USA

Received: 19 December 2000 / Accepted: 9 May 2001 / Published online: 11 October 2001

Abstract. In order to identify ineffective and, hence, superfluous configurations in algorithmically generated configuration spaces, a direct configuration interaction (CI) method has been developed for determining completely general configurational expansions based on arbitrary determinantal configuration lists. While based on the determinantal ordering scheme of Knowles and Handy, our direct CI algorithm differs from previous ones by the use of the Slater–Condon expressions in direct conjunction with single and double replacements. A full, as well as a completely general selected, CI program has been implemented. With it, full configuration spaces of Ne, C_2, CO and H_2O with up to about 40 million determinants have been investigated. It has been found that, in all cases, fewer than 1% of the configurations in a natural-orbital-based configuration expansion reproduce the exact results within chemical accuracy.

Key words: Configuration interaction – Direct full configuration interaction – Direct selected configuration interaction – Configuration interaction expansion truncation

1 Introduction

Expansions in terms of orbital-based configurations have, as yet, remained the most common representations of molecular electronic wavefunctions. Over the last decades, formal as well as algorithmic advances in conjunction with marked increases in computer power have lead to effective treatments of large expansions so that "chemical accuracy" can be approached more closely, more reliably and for more complex systems. Wavefunctions in full configuration spaces furnish, of course, optimal descriptions in terms of given orbital bases. These spaces, however, typically grow in size roughly with the square of the binomial coefficient of the number of orbitals over the number of electrons, so calculations become impractical as the number of electrons and basis functions increases significantly beyond those typical for moderately sized molecules. Moreover, such expansions typically contain a great number of configurations that are ineffective for any given molecular electronic wavefunction. The separation of "deadwood" from "live wood" in full configurational spaces is, therefore, a relevant problem.

Two different contexts must be distinguished. On the one hand, full spaces were first practically used by one of the authors in the context of full valence spaces, or full "reactive" subspaces thereof, in order to account properly for the electronic structure rearrangements during chemical reactions within the zeroth-order approximation of the wavefunction (FORS model [1])[1]. This kind of configuration mixing, due to so-called nondynamic correlations, is difficult to anticipate and the reason for using full spaces is thus to avoid building any bias into the dominant part of the configurational expansion.

On the other hand, subsequent advances in the ease of performing large full-space calculations owing to the complete-active-space self-consistent-field (SCF) method by Siegbahn and coworkers [2] have led to the use of this approach beyond the zeroth-order approximations, viz. also for the description of dynamic correlations. In this case, it is, however, straightforward to home in on the effective configurations by the consecutive inclusion of single, double, triple, quadruple, etc., excitations from suitable single-configurational or multiconfigurational (MC) zeroth-order ("reference") functions, in many cases the Hartree–Fock approximation; however, even here, the problem of distinguishing live wood from deadwood in the excitation spaces remains.

To address this problem, a variety of approaches have been conceived to generate configurational expansions through iterative extensions of reasonable initial reference spaces, based on various premises, by the successive inclusion of single and double excitations whose im-

Correspondence to: K. Ruedenberg

[1] The configuration spaces treated were of dimensions up to about 1,800.

portance is assessed by means of one or several threshold criteria. In this context, we note the work of Gershgorn and Shavitt [3], Bender and Schaefer and Brookes and Schaefer [4], Huron et al. and Evangelisti et al. [5], Buenker and coworkers [6], Meyer [7], Siegbahn [8], Werner and Knowles [9], Harrison [10], Knowles and Handy [11] and Mitrushenkov [12], which also contain references to other contributors to this subject.

An up-to-date discussion of the variety of existing configuration interaction (CI) methods, including additional original developments, has recently been published by Sherrill and Schaefer [13]. For further specifics and further literature, we refer the reader to this comprehensive review, which came to our attention after completion of the present investigation.

In order to be free to pursue the most unbiased exploration of all promising selection criteria that could lead to shorter configurational expansions, it would seem desirable to be able to use an efficient CI method and standalone code for the solution of CI problems that are based on completely arbitrary large configurational expansions without any built-in constraints, in particular if notions of chemical bonding are to be taken into consideration. It is to this end that we have developed the direct determinantal general CI method discussed here. The code has been incorporated in the GAMESS program system [14] and it has also been linked to MCSCF orbital optimization programs.

The concept of a direct CI technique was introduced by Roos [15]. The direct determinant-based full-space CI (FCI) method goes back to Knowles and Handy [16]. Important further developments are due to Olson and coworkers [17]. Additional modifications were contributed by Zarrabian and coworkers [18], by Bendalozzi, and coworkers [19] and by Sherrill and Schaefer [13]. Our algorithm differs from these methods in that we use neither the factorization of the second-order density in terms of the unitary group generators nor its reduction to first order densities by insertion of a resolution of the identity in terms of a full configurational basis. Rather, we use the standard Slater–Condon expressions directly in conjunction with the generation of genuine single and double excitations. This straightforward approach avoids the addition and subtraction of unneeded terms whose insertion had been motivated by considerations regarding vectorization. No major degradation in efficiency appears to result using current workstations and parallelization looks promising. This methodological difference also distinguishes the present work from the general CI code of Povill and coworkers [20] which is embedded in the CIPSI system of molecular programs [5].

To keep the presentation simple, we forego using the unneeded second quantization notation. Some prerequisite background is briefly collected in Sect. 2. The basic equations and the corresponding generic loop structures on which our codes are based are laid out in Sect. 3. Direct procedures are described for several kinds of configuration spaces in Sect. 4, leading to a general CI procedure based on an arbitrary configuration list. Applications to full configuration spaces of up to about 40 million configurations are reported for the systems Ne, C_2, CO and H_2O in Sect. 5. The results exhibit the performance of the codes as well as the great amount of deadwood that is inherent in full configuration spaces for very large orbital bases.

2 Basic determinantal CI framework

In the present section, prerequisite background basics are briefly collected and choices made in our approach are specified. The reader is referred to Ref. [13] for fuller details and references regarding various topics.

2.1 Configurational expansion

In the CI approach, an electronic wavefunction is expressed as a linear superposition

$$\Psi = \sum_K C_K \Phi_K \tag{2.1}$$

of configurations, Φ_K, that are formed from a given set of configuration generating molecular orbitals (CGOs),

$$f_\nu(\mathbf{x}) = |\nu\rangle, \quad \nu = 1, 2, \ldots, M = \text{dimension of the basis orbital space}\ , \tag{2.2}$$

that are obtained from the atomic orbital basis (χ_a) by some optimizing transformation, typically through an SCF or MCSCF calculation. We assume the CGOs to be orthonormal and use the ket notation for them. Inclusion of all configurations that can be generated results in the FCI wavefunction. Not all orbitals of the full molecular orbital (MO) basis need be included in the CGO set however, nor need all possible configurations be included. The latter can be selected either on the basis of some a priori formulated prescriptive algorithm or by means of an explicitly given, and in principle arbitrary, list of configurations. We shall use the term general CI (GCI) wavefunction to denote the latter option, which is our final focus.

The oldest and simplest antisymmetric basis configurations are the Slater determinants. Since most of them are not eigenfunctions of $\mathbf{S}^2$, spin-adapted linear combinations of determinants, which yield somewhat shorter CI expansions, have come into frequent use, in particular through the Graphical Unitary Group algorithm. Knowles and Handy [16, 21] have, however, revived the use of Slater determinants by showing that the extreme simplicity of the MO-integral coupling coefficients in the Hamiltonian matrix elements entails significant algorithmic advantages in the context of the direct approach and, since then, various modifications of this approach have been developed [17–20].

From the conceptual point of view, determinants moreover have the advantage of automatically yielding information and insight regarding the states of all spin multiplicities that are physically relevant in the energy range of interest (provided that M_s is chosen to be equal to 0 or 1/2 for an even or odd number of electrons,respectively). This feature is assured by always dealing with determinantal basis sets in which each collection of space orbitals is associated in all possible ways with α and β spin functions. Thereby, a reducible representation

basis of the symmetric group is generated and the eigenfunctions of $\mathbf{H}$ will be eigenfunctions of $\mathbf{S}^2$.

2.2 Identification of determinants by string pairs

Since the correlation problem is considered within the context of a spin-independent Hamiltonian, the wavefunction,Ψ, is an eigenfunction of $\mathbf{S}_z$ and so are all determinants, Φ_K, used for the expansion in Eq. (2.1). Each determinant, has, therefore N_α orbitals with α spin and N_β orbitals with β spin such that

$$N_\alpha = (1/2)N + M_s, \quad N_\beta = (1/2)N - M_s \ , \tag{2.3}$$

where N is the total number of electrons and $M_s h$ is the eigenvalue of $\mathbf{S}_z$. (In the subsequent algorithms, M_s is always assumed to be 0 or greater so that $N_\alpha \geq N_\beta$.) Consequently, as pointed out by Handy [21] all determinants of a specific problem can be written with the same spin function in the form

$$\Phi_K = A[\Phi'_K \Phi''_K \alpha(1)\alpha(2)\cdots\alpha(N_\alpha)\beta(N_\alpha+1)\beta(N_\alpha+2)\cdots \times \beta(N_\alpha+N_\beta=N)] \ , \tag{2.4}$$

$$\Phi'_K = \phi'_{K1}\phi'_{K2}(2)\cdots\phi'_{K,N_\alpha}(N_\alpha) \ , \tag{2.5}$$

$$\Phi''_K = \phi''_{K1}(N_\alpha+1)\phi''_{K2}(N_\alpha+2)\cdots\phi''_{K,N_\beta}(N_\alpha+N_\beta=N) \ , \tag{2.6}$$

where Φ'_K and Φ''_K are called the α string and the β string of Φ_K, respectively.

The orbitals in both strings are taken from the same orthonormal basis set (Eq. 2.2). By virtue of the antisymmetrization, all orbitals in the product Φ'_K must be different from each other and can be chosen to occur in the standard order established by Eq. (2.2) and the same is true for the product Φ_K. The number of products containing N_α and N_β orbital factors, respectively, that can be formed from the M basis orbitals of Eq. (2.2) manifestly are

$$\mathrm{NP}_\alpha = \binom{M}{N_\alpha} \quad \mathrm{NP}_\beta = \binom{M}{N_\beta} \ , \tag{2.7}$$

so the full determinantal configuration space has the dimension

$$\mathrm{ND} = \mathrm{NP}_\alpha \times \mathrm{NP}_\beta \ . \tag{2.8}$$

We denote the possible α- and β-string products of the basis orbitals as

$$A_p = |\nu(p1)\rangle|\nu(p2)\rangle|\nu(p3)\rangle\cdots|\nu(pN_\alpha)\rangle, \quad p=1,2,\ldots,\mathrm{NP}_\alpha \ , \tag{2.9}$$

$$B_q = |\nu(q1)\rangle|\nu(q2)\rangle|\nu(q3)\rangle\cdots|\nu(qN_\beta)\rangle, \quad q=1,2,\ldots,\mathrm{NP}_\beta \ , \tag{2.10}$$

where the integer index functions

$$\nu = \nu(pj), \quad j = 1,2,\ldots.N_\alpha \tag{2.11}$$

and

$$\nu = \nu(qn), \quad n = 1,2,\ldots,N_\beta \tag{2.12}$$

furnish, respectively, the index of the orbital for electron j in the product A_p and the index of the orbital for electron $(N_\alpha+n)$ in the product B_q. We reserve the indices p, i, j, k for α strings and q, l, m, n for β strings. Assuming unambiguous associations between the string indices p, q and the corresponding orbital index sets in the strings, every string pair $\{A_p, B_q\}$ determines exactly one determinant Φ_K and, thus, the index couples (p, q) furnish an alternative labeling of all determinants in a full space, so one can write

$$\Phi_K = \mathbf{A}\{A_p \times B_q \times \mathrm{spinfunction}\} = |p,q\rangle, \quad p=1,2,\ldots,\mathrm{NP}_\alpha, \quad q=1,2,\ldots,\mathrm{NP}_\beta \ . \tag{2.13}$$

2.3 Enumeration of determinants

The α- and β-string representation provides the basis for a formalism that establishes the two-way connection between the string labels p, q on the left-hand side of Eqs. (2.9) and (2.10) and the corresponding orbital index sets on the right-hand side of these equations, i.e. an explicit algorithm implementing the index functions (Eqs. 2.11, 2.12). Different algorithms have been used by different authors as discussed in Ref. [13], in particular with reference to vector computers. In view of our objective to create a general direct CI code, we choose to stay with the original enumeration scheme of Knowles and Handy [16].

We describe the ordering for the products A_p, the ordering algorithm for the B_q being analogous. It is illustrated in Table 1 for the case of $M = 6$ and $N_\alpha = 3$ so that $\nu = 1, 2, 3, 4, 5, 6$ and $j = 1, 2, 3$, whence $p = 1, 2, 3, \ldots, \mathrm{NP}_\alpha = 20$. For each string label, p, listed in the first column, the next three columns list the corresponding orbital index values $\nu(p1)$, $\nu(p2)$, $\nu(p3)$. Two algorithms will be needed: One to deduce the orbital index set from the string index p, the other to deduce p

Table 1. Orbital product ordering for $N_\alpha = 3$, $M = 6$

p	$\nu(p1)$	$\nu(p2)$	$\nu(p3)$
1	1	2	3
2	1	2	4
3	1	2	5
4	1	2	6
5	1	3	4
6	1	3	5
7	1	3	6
8	1	4	5
9	1	4	6
10	1	5	6
11	2	3	4
12	2	3	5
13	2	3	6
14	2	4	5
15	2	4	6
16	2	5	6
17	3	4	5
18	3	4	6
19	3	5	6
20	4	5	6

(A) Set $\nu(1,j) = j$ for $j=1,2,\ldots,N_\alpha$

(B) Do for $p=1,2 \ldots . (NP_\alpha - 1)$:

(C) Do for $i=0,1,2,\ldots,(N_\alpha-1)$:

(D) Is $\nu(p, N_\alpha-i) = M-i$?

If Yes :

Advance $i \rightarrow i+1$ and return to (D)

If No :

Determine the $\nu(p+1,j)$ as follows:

For $j=1,2,\ldots,(N_\alpha-i-1)$:

$\nu(p+1,j) = \nu(p,j)$

For $j=(N_\alpha-i),(N_\alpha-i+1),\ldots,N_\alpha$:

$\nu(p+1,j) = \nu(p, N_\alpha-i) + 1 + j - (N_\alpha-i)$

Advance $p \rightarrow p+1$ and return to (C)

Fig. 1. Orbital product ordering algorithm

from the orbital index set. The former consists of going through the recursive sequence of logical steps formulated in Fig. 1, generating the orbital products in the order listed in Table 1, until p has reached the desired value. Conversely, in order to deduce p from the orbital index set, we obtain the formula

$$p[\nu(p,1),\ \nu(p,2),\ldots,\ \nu(p,N_\alpha)] = 1 + \sum_{i=1}^{N_\alpha} \sum_{j=\nu(p,i-1)+1}^{\nu(p,i)-1} \binom{M-j}{N_\alpha-i} , \quad (2.14)$$

where $\nu(p,0)$ is defined to be 0. Equation (2.14) appears to be simpler than Eq. (11) in Ref. [16b].

The linear ordering and addressing of the determinants Φ_K furthermore requires a definition of the sequence in which one goes through the matrix of string pairs (p, q). This enumeration depends on the configuration space considered and is discussed in Sect. 4; the assertion of the existence of such an ordering algorithm, $K(p, q)$, suffices at this point. Through it, in conjunction with the preceding string generation, the address index of each determinant Φ_K is algorithmically connected with the indices of the orbitals in it. Hence, if the values of the CI expansion coefficients, C_K, are listed in the order given by K, no lists of orbital products need be kept.

2.4 Determination of CI coefficients

The vector, **C**, of the CI coefficients in Eq. (2.1) is determined as a solution of the matrix eigenvalue equation

$$\mathbf{HC} = E\mathbf{C} , \quad (2.15)$$

$$H_{KL} = \langle \Phi_K | \mathbf{H} | \Phi_L \rangle , \quad (2.16)$$

where **H** is the electronic Hamilton operator. The eigenvectors of **H** must be determined by an iterative procedure appropriate for very large sparse matrices and various methods available are discussed in Ref. [13]. In quantum chemical contexts, Davidson's algorithm [22] has generally been found to be the most effective and it is also used in our code. In all methods, the essential quantity that has to be calculated at each iterative improvement step is the gradient vector, σ, which is obtained by the matrix multiplication

$$\sigma = \mathbf{HC} . \quad (2.17)$$

In terms of the string-pair labeling of determinants, this equation becomes

$$\sigma(p,q) = \sum_r \sum_s \langle p,q|\mathbf{H}|r,s\rangle C(r,s) , \quad (2.18)$$

where the (p, q) indexing has also been applied to σ.

Operationally, the only MO-related quantities required for the solution of the CI problem are the one- and two-electron energy integrals because it is from these that the elements of the matrix **H** are constructed. The complexities of this coupling process in conjunction with the huge number of matrix elements are the causes for the computational bottlenecks and the focus of the direct algorithms.

If the MOs can be divided into core and active orbitals, the former being doubly occupied in all configurations, then, as is well known, the explicit use of integrals involving core orbitals can be avoided by appropriate modifications of the integrals involving only active orbitals and working out the energy contributions involving only core orbitals ahead of the CI calculation.

A not irrelevant element of the iterative procedure is the construction of a good initial guess for the solution vectors of the various states desired in a particular problem. We construct an initial guess for each CI vector by diagonalizing a Hamiltonian matrix $\mathbf{H}^0$ in a small space, typically of approximately 300 determinants. We identify this space by choosing the determinants with the lowest energies and, then, complementing them by those additional determinants that are required to include, for each space orbital selection, all admissible associations with α and β spin functions.

Our program allows the determination of several of the lowest eigenvalues of **H**. The spin multiplicity of each eigenvalue is determined by calculating the expectation value of $\mathbf{S}^2$, using Dirac's expression in terms of permutations. It is not necessary to find the expectation value of the entire CI vector, one needs only to take a part of each eigenvector that encompasses all determinants derived from one space orbital selection. We make this selection from the determinants with the highest coefficients.

3 Direct determinantal CI framework

3.1 Basic equations

In the direct method, the elements of **H** are computed on the fly from the MO integrals so that they do not have to be stored or repeatedly moved from memory to disk and vice versa. This becomes possible by exploiting the

sparsity of the **H** matrix through explicit insertion of the expressions of the H_{JK} in terms of orbital integrals and considering only nonvanishing matrix elements. We choose the original Slater–Condon expressions [23] to this end. Doing so in the operative Eq. (2.18) yields the following explicit formulas in terms of orbitals:

$$\sigma(p,q) = \sigma_1(p,q) + \sigma_2(p,q) + \sigma_3(p,q) + \sigma_4(p,q) + \sigma_5(p,q) + \sigma_6(p,q) \ , \tag{3.1}$$

with the six terms being the following expressions.

3.1.1 Diagonal elements

$$\sigma_1(p,q) = \langle p,q|\mathbf{H}|p,q\rangle C(p,q) \ , \tag{3.2}$$

$$\begin{aligned}\langle p,q|\mathbf{H}|p,q\rangle &= \sum_j \langle \nu(pj)|h|\nu(pj)\rangle + \sum_n \langle \nu(qn)|h|\nu(qn)\rangle \\ &+ \sum_i \sum_{j<i} \{[\nu(pi)\nu(pi)|\nu(pj)\nu(pj)] \\ &\qquad - [\nu(pi)\nu(pj)|\nu(pi)\nu(pj)]\} \\ &+ \sum_m \sum_{n<n} \{[\nu(qm)\nu(qm)|\nu(qn)\nu(qn)] \\ &\qquad - [\nu(qm)\nu(qn)|\nu(qm)\nu(qn)]\} \\ &+ \sum_j \sum_n [\nu(pj)\nu(pj)|\nu(qn)\nu(qn)] \ ,\end{aligned} \tag{3.3}$$

where the indices i, j, referring to α strings, run from 1 to N_α and the indices m, n, referring to β strings, run from 1 to N_β.

3.1.2 Single replacements in α strings

Here, the following notations are expedient: $|\nu^*(p)\rangle =$ all$(M - N_\alpha)$ basis orbitals not contained in the product A_p, i.e. for which

$$|\nu^*(p)\rangle \neq |\nu(pj)\rangle, \quad j = 1,2,\ldots .N_\alpha \ , \tag{3.4}$$

$r(pj\nu^*)$ is the index of the string $A_{r(pj\nu^*)}$ that is obtained from the string $A_p$by replacing the orbital $|\nu(pj)\rangle$ with the orbital $|\nu^*(p)\rangle$ and then establishing the standard orbital order of Eq. (2.2) in the replaced product (3.5)

$\varepsilon(pj\nu^*) = (\pm 1)$, depending upon the parity of the orbital permutation that brings the simply replaced product into the standard order of $r(pj\nu^*)$ (3.6)

We then have

$$\sigma_2(p,q) = \sum_j \sum_{\nu^*} \varepsilon(pj\nu^*)\langle p,q|\mathbf{H}|r(pj\nu^*),q\rangle C[r(pj\nu^*),q] \ , \tag{3.7}$$

$$\begin{aligned}\langle p,q|\mathbf{H}|r(pj\nu^*),q\rangle &= \langle \nu(pj)|h|\nu^*(p)\rangle \\ &+ \sum_{i\neq j} \{[\nu(pi)\nu(pi)|\nu(pj)\nu^*(p)] \\ &\qquad - [\nu(pi)\nu(pj)|\nu(pi)\nu^*(p)]\} \\ &+ \sum_n [\nu(qn)\nu(qn)|\nu(pj)\nu^*(p)] \ ,\end{aligned} \tag{3.8}$$

where j runs from 1 to N_α, i runs over the same values with the exception of $i = j$, n runs from 1 to N_β, and $\nu^*(p)$ runs over the values specified in Eq. (3.4).

3.1.3 Single replacements in β strings

In this case, we define the β-string symbols $|\nu^*(q)\rangle$, $s(qn\nu^*)$, $\varepsilon(qn\nu^*)$ with meanings analogous to those given in Eqs. (3.4), (3.5) and (3.6) for α strings. We then obtain

$$\sigma_3(p,q) = \sum_n \sum_{\nu^*} \varepsilon(qn\nu^*)\langle p,q|\mathbf{H}|p,s(qn\nu^*)\rangle C[p,s(qn\nu^*)] \ , \tag{3.9}$$

$$\begin{aligned}\langle p,q|\mathbf{H}|p,s(qn\nu^*)\rangle &= \langle \nu(qn)|h|\nu^*(q)\rangle \\ &+ \sum_{m\neq n} \{[\nu(qm)\nu(qm)|\nu(qn)\nu^*(q)] \\ &\qquad - [\nu(qm)\nu(qn)|\nu(qm)\nu^*(q)]\} \\ &+ \sum_j [\nu(pj)\nu(pj)|\nu(qn)\nu^*(q)] \ ,\end{aligned} \tag{3.10}$$

where n runs from 1 to N_β , m runs over the same values with the exception of $m = n$, j runs from 1 to N_α and $\nu^*(q)$ runs over all basis orbitals not contained in the string B_q.

3.1.4 Double replacements in α strings

Here the following notations are useful.

$|\nu^*(p)\rangle, |\mu^*(p)\rangle$ both run over all $(M - N_\alpha)$ basis orbitals not contained in the product A_p . (3.11)

$r(pjk\nu^*\mu^*)$ is the index of the string $A_{r(pjk\nu^*\mu^*)}$ that is obtained from the string A_p by replacing in it the orbitals $\nu(pj), \nu(pk)$ by the orbitals $\nu^*(p), \mu^*(p)$, respectively (where $j > k$ and $\nu^* > \mu^*$) and then establishing the standard orbital order in the replaced product . (3.12)

$\varepsilon(pjk\nu^*\mu^*) = (\pm 1)$, depending upon the parity of the permutation that brings the replaced product into the standard order of $r(pjk\nu^*\mu^*)$. (3.13)

We then obtain

$$\sigma_4(p,q) = \sum_j \sum_{k<j} \sum_{\nu^*} \sum_{\mu^*<\nu^*} \varepsilon(pjk\nu^*\mu^*) \langle p,q|\mathbf{H}|r(pjk\nu^*\mu^*),q\rangle C[r(pjk\nu^*\mu^*),q] \ , \tag{3.14}$$

$$\langle p,q|\mathbf{H}|r(pjk\nu^*\mu^*),q\rangle = [\nu(pj)\nu^*(p)|\nu(pk)\mu^*(p)] - [\nu(pj)\mu^*(p)|\nu(pk)\nu^*(p)] \ , \tag{3.15}$$

where j, k run from 1 to N_α and ν^*, μ^* run over the values specified in Eq. (3.11).

3.1.5 Double replacements in β strings

In this case, we define the β-string symbols $|\nu^*(q)\rangle$, $|\mu^*(q)\rangle$, $s(qnm\nu^*\mu^*)$, $\varepsilon(qnm\nu^*\mu^*)$ with meanings analogous to those given in Eqs. (3.11), (3.12) and (3.13). We then obtain

$$\sigma_5(p,q) = \sum_n \sum_{m<n} \sum_{\nu^*} \sum_{\mu^*<\nu^*} \varepsilon(qnm\nu^*\mu^*) \langle p,q|\mathbf{H}|p,s(qnm\nu^*\mu^*)\rangle C[p,s(qnm\nu^*\mu^*)] \ , \tag{3.16}$$

$$\langle p,q|\mathbf{H}|p,s(qnm\nu^*\mu^*)\rangle = [\nu(qn)\nu^*(q)|\nu(qm)\mu^*(q)] - [\nu(qn)\mu^*(q)|\nu(qm)\nu^*(p)] \ , \tag{3.17}$$

where n, m run from 1 to N_β, and ν^*, μ^* run over all basis orbitals not contained in the string B_q.

3.1.6 Single replacements in both the α and the β strings

Using again the notations employed in Eqs. (3.4), (3.5), (3.6), (3.7), (3.8), (3.9), (3.10), (3.11), (3.12) and (3.13) we have

$$\sigma_6(p,q) = \sum_j \sum_{\nu^*} \sum_n \sum_{\mu^*} \varepsilon(pj\nu^*)\varepsilon(qn\mu^*) \langle p,q|\mathbf{H}|r(pj\nu^*),s(qn\mu^*)\rangle C[r(pj\nu^*),s(qn\mu^*)] \tag{3.18}$$

$$\langle p,q|\mathbf{H}|r(pj\nu^*),s(qn\mu^*)\rangle = [\nu(pj)\nu^*(p)|\nu(qn)\mu^*(q)] \ , \tag{3.19}$$

where the index ranges are the same as in Eqs. (3.7), (3.8), (3.9) and (3.10).

3.2 Generic loop structure

Evaluation of the equations just given calls for the repeated execution of the following operations:

1. Advancing sequentially through the indices p and q, the orbital sets of the strings must be generated. The orbital index sets for $(p+1)$ and $(q+1)$ are obtained from those for p and q by using step C of the overall algorithm displayed in Fig. 1.
2. For orbital products generated by single and double orbital replacements, the appropriate string labels must be found. This is done with Eq. (2.14).
3. In order to pick up the appropriate CI coefficient C_K and to store the $\sigma(K)$, the address $K(pq)$ must be calculated from the string label pair (p, q). These algorithms are discussed in Sect. 4.

The evaluation of Eqs. (3.2), (3.3), (3.4), (3.5), (3.6), (3.7), (3.8), (3.9), (3.10), (3.11), (3.12), (3.13), (3.14), (3.15), (3.16), (3.17), (3.18) and (3.19) is then accomplished through the following sequence of operations, which will serve as the reference for the discussions in Sect. 4.

(0) *Calculation of the diagonal elements* $\langle K|\mathscr{H}|K\rangle = \langle pq|\mathscr{H}|pq\rangle$ *before starting the iterations for solving the eigenvalue problem.*

(0.1) Loop over the alpha strings p, generate and store the orbital index set for A_p. Evaluate and store the first plus the third sum of Eq. (3.3), which do not depend on the beta strings q.

(0.1.1) Loop through all beta strings q, generate the orbital index sets for B_q. Evaluate the fifth sum of Eq. (3.3), add the previously stored value of the first plus the third sum to it.

Add the total as a contribution into the diagonal element $\langle pq|\mathscr{H}|pq\rangle$.

(0.2) Loop over the beta strings q, generate the orbital index sets for B_q. Evaluate and store the second plus the fourth sum of Eq. (3.3), which do not depend on the alpha strings p.

(0.2.1) Loop through the alpha strings p, without generating the orbital index sets for A_p.

Add the just found value of the second plus the fourth sum to the previously found contributions to the diagonal matrix element $\langle pq|\mathscr{H}|pq\rangle = \langle K|\mathscr{H}|K\rangle$, all of which are kept stored throughout all iterations. (They are not only required for the calculation of σ_1 but also in the CI solving algorithm).

(1) *Loop for* σ_1

Loop over all K. Pick up and multiply the stored values of $\langle K|\mathscr{H}|K\rangle$ and $C(K)$ yielding $\sigma_1(K) = \sigma_1(p,q)$ according to Eq. (3.2).

(2) *Loop for* σ_2, σ_4, σ_6

Loop over the alpha strings p, generating and storing the orbital index sets A_p. For each p:

(2.1) *Loop for* σ_2, σ_4

Loop over all single replacements $|\nu(pj)\rangle \rightarrow |\nu^*(p)\rangle$. Determine and store $r(pj\nu^*)$, $\varepsilon(pj\nu^*)$, $|\nu(pj)\rangle$, $|\nu^*(p)\rangle$. Pick up the integrals in the first two lines of Eq. (3.8), which do not depend on q; evaluate and store their sum.

(2.1.1) *Loop* for σ_2

Loop over all q, generating the orbital index set for each B_q. Pick up the integrals in the third line of Eq. (3.8); evaluate the sum and add to it the stored

sum of the first two lines of that equation. Pick up $\varepsilon(pj\nu^*)$ and $C[r(pj\nu^*), q]$, evaluate Eq. (3.7).
Add the resulting contribution into $\sigma_2(pq)$.

(2.1.2) *Loop for* σ_4
Loop over all additional single replacements $|\nu(pk)\rangle \rightarrow |\mu^*(p)\rangle$, with $k > j$, $\nu^* > \mu^*$, thus yielding the double alpha string replacements $|\nu(pj)\rangle$, $|\nu(pk)\rangle \rightarrow |\nu^*(p)\rangle$, $|\mu^*(p)\rangle$. Determine $r(pjk\nu^*\mu^*)$ and $\varepsilon(pjk\nu^*\mu^*)$.
Pickup the integrals in Eq. (3.15), none of which contain any beta-string orbitals, evaluate the expression (3.15), multiply by $\varepsilon(pjk\nu^*\mu^*)$ and store.

(2.1.2.1) Loop over all q without generating the orbital index sets.
Multiply the just stored quantity by $C[r(pjk\nu^*\mu^*), q]$.
Add the resulting contribution into $\sigma_4(pq)$.

(2.2) *Loop for* σ_6
Loop over all q, generating the orbital index set for each B_q. For each q:

(2.2.1) Loop over all single replacements $|\mu(qn)\rangle \rightarrow |\mu^*(q)\rangle$. Determine $s(qn\mu^*)$, $\varepsilon(qn\mu^*)$.

(2.2.1.1) Loop over all single alpha string replacements $r(pj\nu^*)$, information about which was stored at the beginning of loop (2.1). Using this information, pick up the matrix element (3.19). Pick up $C(r(pj\nu^*), s(qn\mu^*))$ and multiply the four factors in Eq. (3.18). Add the resulting contribution into $\sigma_6(p, q)$.

(3) *Loop for* $\sigma_3(p, q)$ *and* $\sigma_5(p, q)$.
The calculation of these contributions is accomplished in a manner analogous to the calculation of $\sigma_2(p, q)$ and $\sigma_3(p, q)$ by appropriate inversion of the roles of p and q and of the corresponding loops.
However, the values of $\sigma_3(q, p)$ and $\sigma_5(q, p)$ do not have to be evaluated explicitly for wavefunctions with $M_s = 0$ because, in this case, one has $N_\alpha = N_\beta$ and

$$C(p,q) = (-1)^S C(q,p) \ , \tag{3.20}$$

where $S(S+1)$ = eigenvalue of $\mathscr{S}^2$, whence, as shown by Olson,[17]

$$\sigma_3(p,q) = (-1)^S \sigma_2(q,p) \ , \tag{3.21}$$

$$\sigma_5(p,q) = (-1)^S \sigma_4(q,p) \ , \tag{3.22}$$

$$\sigma_6(p,q) = (-1)^S \sigma_6(q,p) \ , \tag{3.23}$$

(4) *Exploitation of matrix element symmetries*
In order to reduce the computational effort, additional loop refinements are introduced that take advantage of the symmetry of the **H** matrix and the invariance of the two-electron integrals $[\nu,\nu'|\mu,\mu']$ under eight manifest orbital permutations. Exploitation of these equivalencies allows, for instance, the following savings.

In loop 2.1, the single and double replacements can be restricted to those for which the strings $r(pj\nu^*)$ as well as $r(pjk\nu^*\mu^*)$ *follow* the string p in the sequence established by the algorithm of Fig. 1.

In loop 2.2, the single replacement strings $r(pj\nu^*)$ and $s(qn\mu^*)$ can be restricted to those that follow the strings p and q respectively, and the four contributions

$$\begin{aligned}
&\varepsilon(r)\varepsilon(s)\langle p,q|\mathscr{H}|r,s\rangle C(r,s)+\varepsilon(p)\varepsilon(q)\langle r,s|\mathscr{H}|p,q\rangle . C(p,q)\\
&+\varepsilon(p)\varepsilon(s)\langle r,q|\mathscr{H}|p,s\rangle C(p,s)+\varepsilon(r)\varepsilon(q)\langle p,s|\mathscr{H}|r,q\rangle C(r,q)\\
&=\langle p,q|\mathscr{H}|r,s\rangle[\varepsilon(r)\varepsilon(s)C(r,s)+\varepsilon(p)\varepsilon(q)C(p,q)\\
&\quad+\varepsilon(p)\varepsilon(s)C(p,s)+\varepsilon(r)\varepsilon(q)C(r,q)]
\end{aligned} \tag{3.24}$$

can be calculated at the same time.
In loop 3, similar refinements are introduced.

The computational effort can be further reduced by storing additional information pertaining to single replacements. For larger systems, some of these lists can get quite sizable however, eventually requiring disk storage. With a view to our general CI as well as parallelization objectives, we have chosen to use minimal storage of such information.

4 Direct procedures for specific configuration spaces

We maintain the string generation introduced in Sect. 2.3, as specified in Fig. 1, for all configuration spaces discussed in the following. The differences in the determinant enumerations for the different cases are generated by means of the algorithms $K(p, q)$ that connect the matrix of string index pairs (p, q) with the linear determinant ordering given by the index K. These specifications, which we had postponed until now, are formulated in the present section.

4.1 Full spaces without spatial symmetry

We begin by recalling the prototype of a full configuration space without consideration of symmetry [16]. In this case, a unique ordering of all determinants is established by imagining all index pairs (p, q) arranged in a matrix array and going through this matrix row by row so that the determinant index K is defined by

$$K(p,q) = (p-1)\mathrm{NP}_\beta + q = 1, 2, \ldots, \mathrm{ND} \ , \tag{4.1}$$

$$p = 1, 2, \ldots \mathrm{NP}_\alpha, \ q = 1, 2, \ldots, \mathrm{NP}_\beta \ , \tag{4.2}$$

where NP_α, NP_β and ND are given by Eqs. (2.7) and (2.8), the latter being the dimension of the full determinantal configuration space. Use of this relation in the generic algorithm of Sect. 3.2 yields the direct procedure for a full space with C_1 symmetry. Conversely, the string indices (p, q) can be obtained from the determinant index K by

$$p = \text{nearest integer} > (K/\mathrm{NP}_\beta), q = K-(p-1)\mathrm{NP}_\beta \ , \tag{4.3}$$

a relation not needed for the evaluation of σ, but required in applications where orbital structures have to be established for determinants given by their index.

4.2 Spaces with Abelian space symmetry

The Abelian group D_{2h} and its subgroups can be generated by one, two or three of the generating

elementary groups C_2, σ (reflection by a plane) and i (inversion). Since each of the latter has two one-dimensional irreducible representations with characters ± 1, these Abelian groups have two, four or eight irreps, respectively. The following discussion is couched in terms of D_{2h}, where we label the irreducible representations by $\Gamma = 1,\ 2,\ 3,\ldots,\ 8$. Adaptation to the other groups is trivial. An irrep multiplication table is calculated and stored. A list of the irreps of all basis orbitals is also stored in the sequence established by Eq. (2.2).

Two orderings are used for the orbital strings. One, referred to as the standard order, is the one discussed in Sect. 2.3, which does not take into account symmetry and, as before, is again labeled by p and q respectively. The second ordering, places all α strings of the same irrep into one contiguous sequence, the ordering within each of these sequences being that of their occurrence in the standard ordering and the ordering of the irrep sequences being that of the labels Γ defined (arbitrarily) in the preceding paragraph. An analogous second ordering is used for the β strings. Let these symmetry orderings of the α and β strings be labeled by indices $\hat{p}$ and $\hat{q}$, respectively, and let $\hat{p}(p)$ and $\hat{q}(q)$ denote the integer functions that generate the symmetry orderings from the standard orderings.

For the operational work with the function $\hat{p}(p)$, we store lists of the following quantities

$$\Gamma(p) = \text{the irrep } (1,2,\ldots,8) \text{of the string } p, \text{for each } p = 1,2,\ldots,\mathrm{NP}_\alpha\ , \tag{4.4}$$

$$\hat{P}(\Gamma) = \hat{p} \text{ of the first } \alpha \text{ string in the list of all those } \alpha \text{ strings that belong to irrep } \Gamma, \text{for all } \Gamma = 1,2,\ldots,8, \tag{4.5}$$

$$\delta\hat{p}(p) = \text{the relative position of a string } p \text{ within the list of all those } \alpha \text{ strings that belong to the same irrep } \Gamma(p), \text{ for all } p = 1,2,\ldots,\mathrm{NP}_\alpha\ . \tag{4.6}$$

Hence, we have

$$\hat{p}(p) = \hat{P}[\Gamma(p)] + \delta\hat{p}(p) - 1\ . \tag{4.7}$$

We note that, for a given irrep Γ, the $\hat{p}$ run from $\hat{P}(\Gamma)$ to $[\mathrm{P}(\Gamma+1) - 1]$ so that the corresponding values of $\delta\hat{p}(p)$ run from 1 to $[\hat{P}(\Gamma+1) - \hat{P}(\Gamma)]$. For any standard α string index p, one can manifestly determine the value of $\hat{p}(p)$ by first finding $\Gamma(p)$ from the list (Eq. 4.4) and then $\hat{p}(p)$ from Eq. (4.7) and the lists (Eqs. 4.5, 4.6). Analogous quantities $\Gamma(q)$, $\hat{Q}(\Gamma)$, $\delta\hat{q}(q)$ are similarly defined for β strings.

In order to generate a linearly ordered list of addresses $\hat{K}$ for a symmetry-adapted determinantal basis, we need a symmetry-adapted version of Eq. (4.1). To this end, we define, for the problem at hand, the conjugate irrep $\Gamma^{\#}(\Gamma)$ of a given irrep Γ as that irrep for which the direct product $\Gamma \otimes \Gamma^{\#}$ is the irreducible representation of the wavefunction Ψ, and we store the list of the eight conjugate irreps

$$\Gamma^{\#}(\Gamma),\ \Gamma = 1,2,\ldots,8\ . \tag{4.8}$$

Hence if, for any one determinant, $\hat{p}$ belongs to the irrep Γ then $\hat{q}$ must belong to the irrep $\Gamma^{\#}(\Gamma)$ and we refer to these α and β strings as being mutually conjugate. If we now imagine the symmetry string label pairs $(\hat{p}, \hat{q})$ arranged in a matrix array, then the latter is divided into 64 blocks corresponding to the row and column irreps. Only for eight of them are the row and column irreps conjugate as given by Eq. (4.8), however, and we call them the conjugate blocks. The order of the symmetry-adapted determinants is then defined by going through this matrix row by row and selecting only string pairs from conjugate blocks. By virtue of the meanings of the quantities discussed in Eqs. (4.4), (4.5), (4.6) and (4.7) and the analogous β-string quantities, the address $\hat{K}$ of the symmetry-adapted determinant formed from the conjugate strings p and q is then defined by

$$\begin{aligned}\hat{K}(p,q) &= \hat{K}(\hat{p},\hat{q}) \\ &= \sum_{\gamma}[\hat{P}(\gamma+1) - \hat{P}(\gamma)][\hat{Q}(\Gamma^{\#}+1) - \hat{Q}(\Gamma^{\#})] \\ &\quad + [\delta\hat{p}(p) - 1)][\hat{Q}(\Gamma^{\#}+1) - \hat{Q}(\Gamma^{\#})] + \delta\hat{q}(q)\ ,\end{aligned} \tag{4.9}$$

where $\Gamma = \Gamma(p)$, $\sum_\gamma$ runs over the irreps γ from 1 to $[\Gamma(p) - 1]$ and $\hat{q}(q)$ is restricted to the block conjugate to p.

It is in the order given by the index $\hat{K}$, that the CI coefficients as well as the components of the gradient vector σ are listed, implying the indexing $\sigma(\hat{K})$, $C(\hat{K})$ and a list that is shorter than the full list by approximately a factor of 8 (for D_{2h}).

In the actual calculation of σ, we still go through the generic loop structure of Sect. 3.2 in terms of the standard labels p and q. At each stage, however, certain symmetry-related tests and selections are inserted as follows.

(0) *Diagonal elements*

(0.1.1) *Loop over q*:

• Only those q are selected that are conjugate to p.

(Although these q's are not contiguous in the standard list, they do occur in the *sequence* of the standard order for a given p. Hence, the orbital indices of these beta strings are generated by what amounts, in total, to only one partial sweep through the algorithm of Figure 1 for each p).

(0.2.1) Loop over p:

• Only those p are selected that are conjugate to q.

(1) *Loop for* σ_1

No tests are necessary.

(2) *Loop for* σ_2, σ_4, σ_6.

(2.1) *Loop for* σ_2, σ_4.

• Determine and store the irrep of the product $|\nu(pj)\rangle \times |\nu^*(p)\rangle$.

If it is not the identity skip to loop (2.1.2). Only if it is the identity, calculate the first two lines of Eq. (3.8) and proceed to loop (2.1.1).

(2.1.1) *Loop for* σ_2:

Only those q are chosen for which $\hat{q}$ is conjugate to $\hat{p}$.

(2.1.2) *Loop for* σ_4:

• Only those double replacements are chosen for which the orbital product $|\nu(pk)\rangle \times |\mu^*(p)\rangle$ belongs to the same irrep as the previously stored irrep of the orbital product $|\nu(pj)\rangle \times |\nu^*(p)\rangle$.

(2.1.2,1) • Choose only those q for which $\hat{q}$ is conjugate to $\hat{p}$.

(2.2) *Loop for* σ_6:

• Use of the refinements that exploit the equivalencies discussed in connection with Eq. (3.24) requires looping through single replacements from *all* q.

• In Eq. (3.19), only those two-electron integrals $[\nu(pj)\nu^*(p)\,|\,\nu(qn)\mu^*(q)]$ are selected for which the product of the orbitals for electron 1 belongs to the same irrep as the product of the orbitals for electron 2.

• Accordingly, only the corresponding coefficients $C(r(pj\nu^*), s(qn\mu^*))$ are picked up for Eq. (3.18).

• Only pairs (pq), (ps), (rq), (rs) containing conjugate strings are picked up.

4.3. General configuration spaces

Configuration spaces spanned by arbitrary selections of determinants from full spaces require storage of explicit configuration lists as an input. Such a configuration list can be given by a set of (p, q) pairs and can be viewed as a collection of certain elements in the full (p, q) matrix. It is expedient to use two orderings for the selected given determinants, corresponding to going through the (p, q) matrix by rows or by columns. We denote the two orderings by the determinant indices K and K', respectively and they are illustrated for an arbitrary example in Table 2. For both orderings, a p list and a q list are stored, each one in the order of the pertinent determinant index, viz. K or K', respectively. Also stored is the cross-reference list shown in Table 2 which furnishes the index $K(K')$ for each K'.

It is furthermore useful to define two additional quantities, $K_0(p)$ and $\Delta K(p)$, pertaining to the α strings in the first ordering, where $K_0(p)$ is the position of the first determinant K associated with p and $\Delta K(p)$ is the

Table 2. Example of an arbitrary selection of string pairs

First ordering														
K	1	2	3	4	5	6	7	8	9	10	11	12	13	14
p	1	1	1	2	2	2	2	4	4	7	7	7	7	7
q	4	6	8	2	7	8	9	2	4	1	2	4	6	7
Second ordering														
K'	1	2	3	4	5	6	7	8	9	10	11	12	13	14
p	7	2	4	7	1	4	7	1	7	2	7	1	2	2
q	1	2	2	2	4	4	4	6	6	7	7	8	8	9
Cross-reference list														
K'	1	2	3	4	5	6	7	8	9	10	11	12	13	14
K	10	4	8	11	1	9	12	2	13	5	14	3	6	7

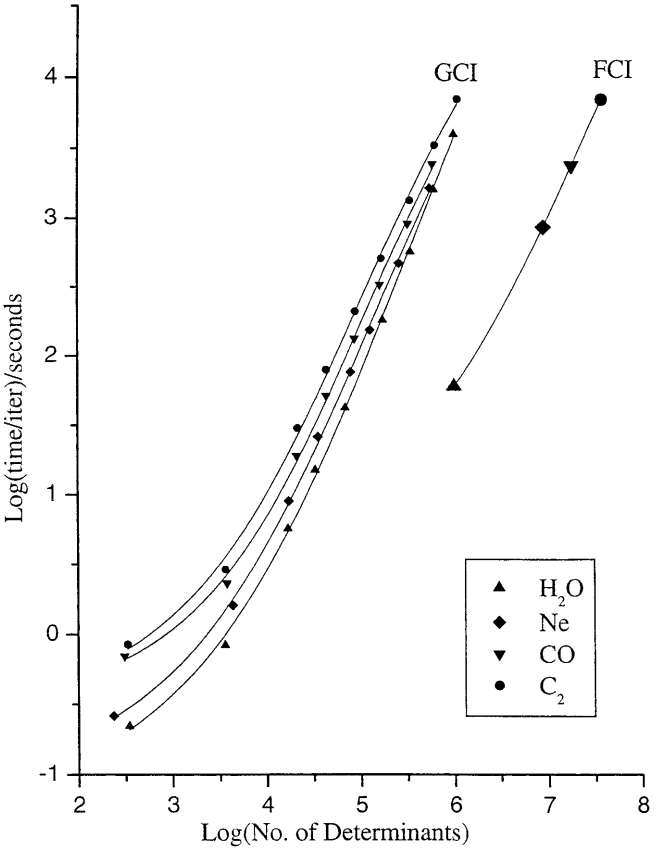

Fig. 2. Scaling of direct configuration interaction (*CI*) algorithm with dimensions of determinantal configuration spaces. Full configuration spaces (*FCI*); truncated configuration spaces (*GCI*)

Table 3. Performance of full-space configuration interaction (*FCI*) code

System	Basis set and number of atomic orbitals	N_C, N_{FV}[a]	Number of determinants	E_{FCI}	E_{CORR}[b]	Time per iteration (s)[c]	Number of iterations[d]
H_2O	DZ[e] (14)	0,0	1,002,708	−76.15572691	−0.14645452	60	12
Ne	cc-pVDZ+[f] (18)	0,0	9,185,280	−128.68519268	−0.19354066	859	11
CO	DZ[e] (20)	2,0	18,367,552	−112.89542617	−0.21037202	2,378	14
C_2	DZP[e] (30)	2,1	38,505,148	−75.72901005	−0.33953192	7,080	23

[a] Number of frozen-core orbitals, Number of omitted highest virtual self-consistent-field orbitals
[b] $E_{CORR} = E_{FCI} - E_{SCF}$
[c] Using an IBM 43P-260 workstation. Subsequent improvements have cut these times in half. For instance, 34 s for H_2O
[d] Number of iterations required for the absolute value of the CI gradient to fall below 1.0×10^{-5}
[e] Ref. [24]
[f] Ref. [25]

Table 4. Performance of general CI code

System	Number of determinants	Percentage full space[a]	$E_{GCI} - E_{SCF}$	$E_{FCI} - E_{GCI}$	Percentage E_{CORR}[b]	Time per iteration (s)[c]	Number of iterations[d]
H_2O	345	0.034	−0.13673502	−0.00971950	93.3635	0.2	6
	3,604	0.36	−0.14559551	−0.00085901	99.4135	1	13
	16,930	1.69	−0.14635340	−0.00010112	99.9310	6	15
	33,125	3.30	−0.14642871	−0.00002581	99.9824	15	16
	68,866	6.89	−0.14644924	−0.00000528	99.9964	42	16
	175,032	17.46	−0.14645410	−0.00000042	99.9997	179	17
	344,558	34.36	−0.14645448	−0.00000004	∼100.00	567	17
	613,230	61.16	−0.14645452	−0.00000000	∼100.00	1,605	17
	1,002,708	100.00	−0.14645452	−0.00000000	∼100.00	3,987	10
Ne	236	0.0026	−0.18668171	−0.00685895	96.4561	0.26[e]	6[e]
	4,361	0.047	−0.19300268	−0.00053798	99.7220	2	8
	17,237	0.19	−0.19350025	−0.00004041	99.9791	10	9
	35,383	0.39	−0.19353057	−0.00001009	99.9948	26	10
	78,654	0.86	−0.19353828	−0.00000238	99.9988	76	10
	127,479	1.39	−0.19353996	−0.00000070	99.9996	152	10
	261,604	2.85	−0.19354055	−0.00000011	99.9999	466	10
	555,539	6.05	−0.19354065	−0.00000001	∼100.00	1,646	10
CO	310	0.0017	−0.18127379	−0.02909823	86.1682	0.7	7
	3,784	0.021	−0.20518499	−0.00518703	97.5344	2	11
	20,975	0.11	−0.20910165	−0.00127037	99.3961	19	13
	43,404	0.24	−0.20981241	−0.00055961	99.7340	51	13
	87,035	0.47	−0.21014620	−0.00022582	99.8927	132	14
	163,544	0.89	−0.21027989	−0.00009213	99.9562	323	14
	324,249	1.77	−0.21034206	−0.00002996	99.9858	908	14
	594,890	3.24	−0.21036276	−0.00000926	99.9956	2,422	14
C_2	335	0.00087	−0.27876569	−0.06076623	82.1029	0.8	9
	3,637	0.0094	−0.32406160	−0.01547032	95.4436	3	14
	21,479	0.055	−0.33518350	−0.00434842	98.7193	30	17
	43,437	0.11	−0.33707515	−0.00245677	99.2764	79	17
	88,747	0.23	−0.33827150	−0.00126042	99.6288	208	17
	169,681	0.44	−0.33889054	−0.00064138	99.8111	504	17
	341,034	0.89	−0.33925925	−0.00027267	99.9197	1,337	18
	625,862	1.62	−0.33941737	−0.00011455	99.9663	3,306	18
	1,090,942	2.83	−0.33948604	−0.00004588	99.9865	7,956	18

[a] Percentage of full-space determinants taken in the general CI
[b] $\%E_{CORR} = (E_{GCI} - E_{SCF})/(E_{FCI} - E_{SCF}) \times 100$
[c] On an IBM 43P-260 workstation. Subsequent improvements have cut these times in half
[d] Number of iterations required for the absolute value of the CI gradient to fall below 5.0×10^{-5}
[e] Initial guess taken with a space of 200 determinants

number of q values associated with p. Similarly, two analogous quantities $K_0'$and $\Delta K'(q)$ are defined pertaining to the β strings in the second ordering.

If the system has symmetry, then a list of the irreducible representations of all orbitals is stored and, by means of prior screening, it is furthermore assured that for each determinant position, K, the β string q is conjugate to the α string p. Care must also be taken that the list spans a representation of the symmetric group so that the eigenfunctions of $\mathbf{H}$ will be eigenfunctions of $\mathbf{S}^2$. If a list has been obtained by configurational truncation, say, the program examines the input list and, if necessary, complements it so that all possible determinants are included for each selection of space orbitals. These additions may enlarge the list markedly.

In the actual calculation of σ, we still use the generic loop structure of Sect. 3.2 in terms of the standard labels p and q; however, in order to ascertain that only those determinants are considered that are in fact in the list, appropriate tests and selections are inserted at each stage of the outline in Sect. 3.2. as follows.

(0) *Diagonal elements*
  (0.1) *Loop over p*
  • Evaluate and store the contribution to Eq. (3.3) only for those p that are in the list of the first ordering.
    (0.1.1) *Loop over q*
    • Evaluate and store the contributions to Eq. (3.3) only for those q that are associated with p in the first ordering.
  (0.2) *Loop over q*
  • Evaluate the contributions to Eq. (3.3) and add them into $\langle pq|\mathscr{H}|pq\rangle$ only for those q that are in the list of the second ordering.
    (0.2.1) *Loop over p*
    • Add the just found contribution only for those p that are associated with q in the list of the second ordering.
(1) *Loop for* σ_1
No tests are necessary.
(2) *Loop for* σ_2, σ_4, σ_6
• Only for those p that are in the p list pertaining to the first ordering:

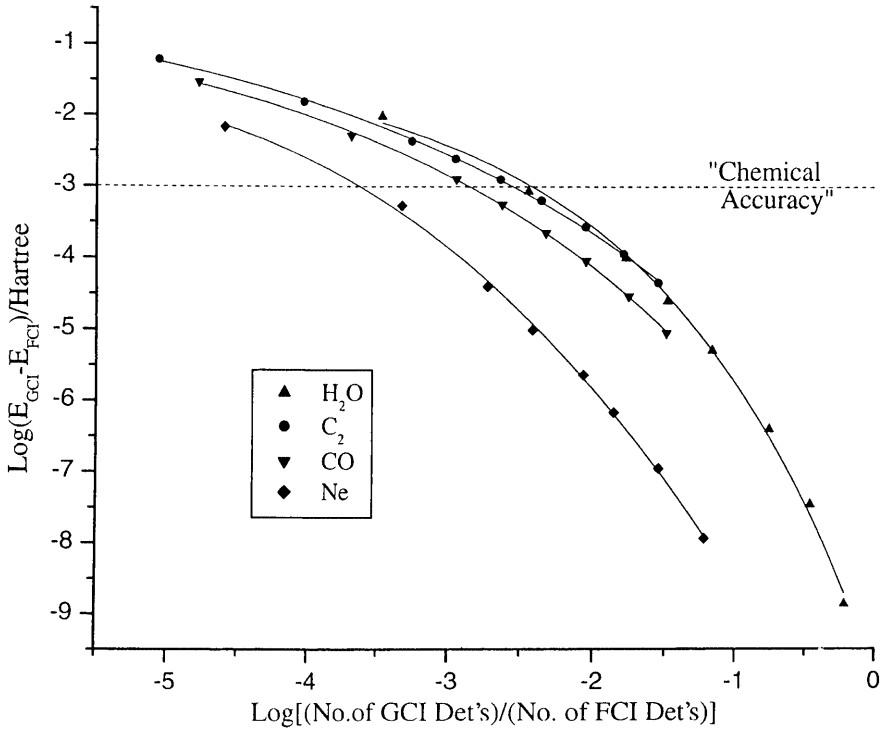

Fig. 3. Energy convergence of truncated CI expansions in terms of natural-orbital-based configurations

Store the orbital index sets for A_p; determine and store $K_0(p)$, $\Delta K(p)$, and proceed to go through the loops (2.1) and (2.2).

(2.1) *Loop for* σ_2, σ_4

• In addition to $r(pj\nu^*)$, $\varepsilon(pj\nu^*)$, $|\nu(pj)\rangle$, $|\nu^*(p)\rangle$, determine and store also the irrep of the product of the two orbitals as well as $K_0[r(pj\nu^*)]$ and $\Delta K[r(pj\nu^*)]$.

• If $r(pj\nu^*)$ is not in the p list of the first ordering or if the product irrep is not the identity, skip to loop (2.1.2) for σ_4, otherwise:

Pick up the first two lines of Eq. (3.8) and proceed to loop (2.1.1) for σ_2.

(2.1.1) *Loop for* σ_2

• Loop only over all those q values that are associated with the alpha string p *as well as* with the alpha string $r(pj\nu^*)$.

(2.1.2) *Loop for* σ_4

• If the irrep of the product $\{|\nu(pk)\rangle \times |\mu^*(p)\rangle\}$ differs from that of the product $\{|\nu(pj)\rangle \times |\nu^*(p)\rangle\}$, go to next additional replacement of p.

If it is identical, determine $r(pjk\nu^*\mu^*)$ and look whether $r(pjk\nu^*\mu^*)$ is in the p-list of the first ordering.

If not, go to the next additional replacement of p.

If it is: Determine $\varepsilon(pjk\nu^*\mu^*)$ and $K_0[r(pjk\nu^*\mu^*)]$.

(2.1.2.1) • Loop only over those q values that are associated with the alpha string p *as well as* with the alpha string $r(pjk\nu^*\mu^*)$.

(2.2) *Loop for* σ_6

• Loop through all q values associated with the alpha string p, as specified by the information stored at the beginning of loop (2).

(2.2.1) • If the irrep of the product $|\mu(qn)\rangle \times |\mu^*(q)\rangle$ differs from that of the product $|\nu(pj)\rangle \times |\nu^*(p)\rangle$, skip to the next single replacement of q. If it is the same, determine $s(qn\mu^*)$, $\varepsilon(qn\mu^*)$.

• By comparison with the information stored at the beginning of loop (2.1), determine whether the beta string $s(qn\mu^*)$ is among those beta strings that are associated with the alpha string $r(pj\nu^*)$ in the first ordering. If it is not, skip to the next single replacement of q.

If it is: Proceed with the evaluation of the contribution to $\sigma_6(p, q)$.

(3) *Loop for* σ_3 *and* σ_5:

These calculations are analogous to those described for σ_2 and σ_4 under loops (2) and (2.1) above, except that the roles of p and q are reversed and the lists associated with the second string ordering are used in conjunction with the cross reference list of Table 2. Eq. (3.20), (3.21), (3.22) and (3.23), still hold so that σ_3 and σ_5 do not have to be evaluated when $M_s = 0$.

5 Identification of deadwood in some large full spaces

The algorithms outlined in Sect. 4 have been implemented in a direct CI code, named JAKAL, which has also been incorporated in the GAMESS program system maintained by M.S. Gordon and M.W. Schmidt [14].

5.1 Full configuration space calculations with C_{2v} and D_{2h} symmetry

So far we have applied the full CI code to problems including up to 140×10^6 determinants. An account of its performance is presented in Table 3, which contains the results for the ground states of the four systems H_2O,

Ne, CO and C_2, with full-space dimensions from 1×10^6 to 40×10^6 determinants. The table lists the specifications for the various cases, the energy results and the execution times using an IBM43P-260 workstation. Refinements of the code since these test cases were run have resulted in a general decrease in the execution times by about a factor of 2. It would appear that the straightforward use of the Slater–Condon [23] expressions for the second-order density, without the factorization in terms of unitary group generators or the insertion of an identity decomposition, does not entail a major degradation in performance.

The scaling of the code with the size of the full configuration space is illustrated by the curve labeled FCI in Fig. 2, which exhibits the decimal logarithm of the execution time per iteration as a function of the decimal logarithm of the number of determinants for these cases. It is very close to a straight line with a least-mean-squares slope of 1.30. The reason for it being less than 2, the anticipated scaling for the eigenvalue problem of a full matrix, is presumably a combination of the sparsity of the **H** matrix, the savings inherent in the direct method and the efficiency of the implementation.

5.2 Truncated configuration space calculations

For each of the four systems, we determined natural orbitals and then redetermined the natural-orbital-based CI coefficients. The determinants were then rearranged in the order of their importance, as given by the absolute value of their CI coefficients and selected lists of configurations were obtained as follows. The X most important determinants were taken and added to them were the least number of determinants needed so that every space orbital selection was coupled in all possible ways with the spin functions α and β, as explained at the end of the third paragraph of Sect. 4.3. For each system, expansions for several values of X were examined. The truncated CI expansions were then determined with the general CI version of the JAKAL code.

Details of the results are presented in Table 4. We note that, in the calculations reported, we did not reduce the basis set of the natural orbitals by deleting those natural orbitals that are unoccupied in all determinants selected for a given choice of X. Since these calculations were performed, improvements in the implementation have resulted in cutting the typical execution time in half and further improvements are currently under way.

The scaling of these calculations is exhibited by the curves labeled GCI in Fig.2, which plot the decimal logarithm of the execution time per iteration versus the decimal logarithm of the number of determinants used. They are seen not only to scale similarly with a relative increase in the number of determinants, but even to lie very close to each other on an absolute scale. With increasing number of determinants, they become close to straight lines, with a common least-mean-squares slope slightly higher than that of the FCI curve.

That GCI calculations are more time consuming than FCI calculation for the same number of determinants is manifestly due to the great amount of checking that is required in every loop, which entails a loss in efficiency. This can be avoided, of course, when it is possible to generate the truncated expansion by an algorithmic enumeration that renders checks unnecessary. Whether the storage of more string information would be beneficial in this context is not obvious. It is conceivable that, for shorter CI expansions, the memory required to do so would exceed that for the CI vector itself.

5.3 Comparison of full and truncated spaces

It is seen from Table 4 that for H_2O, Ne, CO and C_2, respectively, more than 99.9% of the correlation energy is obtained by taking only 1.69, 0.19, 0.89, and 0.89%, respectively, of the full space.

Figure 3 exhibits plots of the decimal logarithm of the energy errors due to truncation (equal to the energy difference between FCI and GCI calculations) versus the decimal logarithm of the truncation fractions (equal to the number of truncated space determinants divided by the number of full space determinants). These curves exhibit the speed of convergence of the natural-orbital-based CI expansions. It is apparent that, for H_2O, Ne, CO and C_2, the truncation error in the energy falls below the chemical accuracy criterion of 1 mhartree with approximately 0.3, 0.03, 0.1 and 0.3% respectively, of the full space.

Of interest is also the size of the truncated space as a percentage of the full space for which the execution time per iteration of the GCI calculations becomes equal to that of FCI calculation. For H_2O, Ne, CO and C_2, this occurs at 6.9, 2.9, 1.8 and 1.6%, of the respective full spaces and results in absolute errors of 5×10^{-3}, 1×10^{-4}, 3×10^{-2} and 1×10^{-1} mhartree, respectively, corresponding to a recovery of 99.9964, 99.9999, 99.9858 and 99.9663% of the correlation energy, respectively,. This is encouraging, especially for a system like C_2, where the basis contains d functions and the dominant determinant has a coefficient of 0.83.

5 Conclusions

A new direct determinantal CI procedure has been developed for full as well as arbitrarily selected configuration expansions and it has been implemented by an efficient workstation code. With it, a quantitative assessment has been obtained for the fraction of large full configuration spaces that must be considered as deadwood for the recovery of dynamic correlation. The program should be useful for experimentation with unconventional CI spaces aiming at the recovery of sufficient parts of the correlation energy with a minimum amount of configurational deadwood. We also plan to use it for the theoretical analysis of molecules in terms of atomic building blocks.

Acknowledgements. The authors wish to thank M.W. Schmidt for stimulating discussions and for his help in our work with the GAMESS program. We also thank M.S. Gordon for his interest and support. This work was supported by the Division of Chemical Sciences, Office of Basic Energy Sciences, U.S. Department of Energy. The Ames Laboratory is operated for the U.S. Department of Energy by Iowa State University under contract no. W-7405-Eng-82.

References

1. (a) Ruedenberg K, Sundberg KR (1978) In: Calais JL, Goscinski O, Linderberg J, Öhrn Y (eds) Quantum science. Plenum, New York, p 505; (b) Sundberg KR (1975) PhD thesis. Iowa State University; (c) Dombeck MG (1977) PhD thesis. Iowa State University; (d) Ruedenberg K (1979) NRCC 1978 Post-Hartree–Fock Quantum Chemistry Workshop Proceedings. Lawrence Berkeley Laboratory, University of California, p 46; (e) Ruedenberg K (1981) NRCC 1980 MCSCF Method Workshop Proceedings. Lawrence Berkeley Laboratory, University of California, p 51; (f) Ruedenberg K, Schmidt MW, Gilbert M, Elbert ST (1982) Chem Phys 71: 51; (g) Ruedenberg K, Schmidt MW, Gilbert M, Elbert ST (1982) Chem Phys 71: 65
2. (a) Siegbahn PEH, Heiberg A, Roos BO, Levy B (1980) Phys Scr 21: 323; (b) Roos BO, Taylor PR, Siegbahn PEM (1980) Chem Phys 48: 157; (c) Roos BO (1980) Int J Quantum Chem 14: 175; (d) Siegbahn PEM, Almlöf J, Heiberg A, Roos BO (1981) J Chem Phys 74: 2384
3. Gershgorn Z, Shavitt I (1968) Int J Quantum Chem 2: 751
4. (a) Bender CF, Schaefer HF (1971) J Chem Phys 55: 4798; (b) Brooks BR, Schaefer HF (1979) J Chem Phys 70: 5092
5. (a) Huron B, Malrieu JP, Rancurel P (1973) J Chem Phys 58: 5745; (b) Evangelisti S, Daudey JP, Malrieu JP (1983) J Chem Phys 75: 91
6. (a) Buenker RJ, Peyerhimoff SD (1974) Theor Chim Acta 35: 33; (b) Buenker RJ, Peyerhimoff SD (1975) Theor Chim Acta 39: 217; (c) Krebs S, Buenker RJ (1995) J Chem Phys 103: 5613
7. Meyer W (1977) In: Schaefer HF (ed) Methods of electronic structure theory. Plenum, New York, p 413
8. Siegbahn PEM (1980) Int J Quantum Chem 18: 1229
9. (a) Werner HJ, Knowles PJ (1988) J Chem Phys 88: 5803; (b) Werner H-J (1987) In: Lawley KP (ed) Ab initio methods in quantum chemistry – II. Wiley, New York, pp 1–61
10. Harrison RJ (1991) J Chem Phys 94: 5021
11. (a) Knowles PJ (1989) Chem Phys Lett 155: 513; (b) Knowles PJ, Handy NC (1989) J Chem Phys 91: 2396
12. Mitrushenkov AO (1994) Chem Phys Lett 217: 559
13. Sherrill CD, Schaefer HF III (1999) Adv Quantum Chem 34: 143
14. (a) Schmidt MW, Baldridge KK, Boatz JA, Elbert ST, Gordon MS, Jensen JH, Koseki S, Matsunaga N, Nguyen KA, Su SJ, Windus TL, Dupuis M, Montgomery JA (1993) J Comput Chem 14: 1347; (b) www.msg.ameslab.gov/GAMESS/GAMESS.html
15. Roos BO (1972) Chem Phys Lett 15: 153
16. (a) Knowles PJ, Handy NC (1984) Chem Phys Lett 111: 315; (b) Knowles PJ, Handy NC (1989) Comput Phys Commun 54: 75
17. (a) Olson J, Roos BO, Jørgensen P, Jensen HJA (1988) J Chem Phys 89: 2185; (b) Olsen J, Jørgensen P, Simons J (1990) Chem Phys Lett 169: 462
18. (a) Zarrabian S, Sarma CR, Paldus J (1989) Chem Phys Lett 155: 183; (b) Harrison RJ, Zarrabian S (1989) Chem Phys Lett 158: 393
19. (a) Bendalozzi GL, Evangelisti S (1993) J Chem Phys 98: 3141; (b) Bendalozzi GL, Evangelisti S (1993) Int J Quantum Chem Symp 27: 287; (c) Evangelisti S, Bendalozzi GL, Ansaloni R, Rossi E (1995) Chem Phys Lett 233: 353; (d) Evangelisti S, Bendalozzi GL, Ansaloni R, Duri F, Rossi E (1995) Chem Phys Lett 252: 437
20. (a) Povill A, Rubio J, Illas F (1992) Theor Chim Acta 82: 229; (b) Povill A, Rubio J (1995) Theor Chim Acta 92: 305
21. Handy NC (1980) Chem Phys Lett 74: 280
22. Davidson ER (1975) J Comput Phys 17: 87
23. (a) Slater JC (1929) Phys Rev 34: 1293; (b) Slater JC (1931) Phys Rev 38: 1109; (c) Condon EU (1930) Phys Rev 36: 1121
24. (a) Dunning TH Jr (1970) J Chem Phys 53: 2823; Dunning TH Jr, Hay PJ (1977) In: Schaeffer HF III (ed) Modern electronic structure theory, vol 4. Plenum, New York, pp 1–27
25. Koch H, Christiansen O, Jørgensen P, Olsen J (1995) Chem Phys Lett 244: 75