

A new efficient approach to the direct restricted active space self-consistent field method

Michael Klene and Michael A. Robb^{a)}
King's College London, Strand, London WC2R 2LS, United Kingdom

Lluís Blancafort
Institut de Química Computacional, Universitat de Girona, 17071 Girona, Spain

Michael J. Frisch
Lorentzian Inc., North Haven, Connecticut 06473

(Received 24 January 2003; accepted 10 April 2003)

We present an implicitly parallel method for integral-block driven restricted active space self-consistent field (RASSCF) algorithms. Our algorithm entirely avoids testing the index space for nonzero contributions to the CI vector, by finding entire blocks of contributions through use of simple algebraic rules (*propagation rules*). The blocks themselves are efficiently identified by introducing a RAS *model space*. Our algorithm is capable of making efficient use of modern supercomputer hardware, supporting both shared and distributed memory architectures and hybrids. Applicability of our method is demonstrated with a RASSCF investigation of the first two excited states of indole. © 2003 American Institute of Physics. [DOI: 10.1063/1.1578620]

I. INTRODUCTION

The restricted active space SCF (RASSCF) variant of the MCSCF method has been first proposed by Olsen *et al.*¹ and many successful applications have since been reported (e.g., Refs. 2–4). The RASSCF method may be considered a logical extension to the CASSCF method, which allows one to tackle two limitations of the latter method.

The practical limit of the system size in a CASSCF computation is around 12–14 active orbitals, at least in cases where the number of electrons is about the same as the number of active orbitals and no symmetry can be exploited. However, the resulting orbital occupancies of several orbitals in the active space are frequently either close to 2 or 0 over the entire range of electronic states and spatial configurations of interest.

Dynamic correlation effects are frequently included in an electronic structure computation in a subsequent step, for example, by adding a calculation treating dynamic correlation perturbatively (e.g., Refs. 5 and 6), or by performing a MRCI calculation, where selected configuration state functions (CSFs) from the CASSCF wave function are used as reference configurations.^{7,8} However, a qualitatively adequate description of some systems sometimes requires the inclusion of dynamical correlation, i.e., an extended active space. Such systems are, for example, negative ions, electronic dipoles and excited states.

In the RASSCF formalism, the configuration space is specified by dividing the active molecular orbitals into three subsets and imposing restrictions on the allowed configurations based upon occupations within those subsets. The first subset, denoted RAS1, typically includes all doubly occupied MOs in some accepted reference, e.g., a CASSCF wave function. Allowed configurations must contain a specified

minimum number of electrons within RAS1. The second subset, denoted RAS2, includes MOs believed to be particularly important for the system under investigation. No occupancy restrictions are imposed on RAS2. The third subset, RAS3, consists of weakly occupied MOs which contribute relatively less to the description of the system of interest. Any allowed configuration can only have a specified maximum number of electrons in RAS3.

The main challenge in large scale CASSCF problems is the efficient computation of coupling coefficients, and this is not different in the RASSCF method. However, the restrictions introduced for the RAS1 and RAS3 subspaces mean that the indexing of the CSFs is more complicated. This in turn means that highly efficient strategies such as the method of reduced excitation strings⁹ cannot be employed.

Our aim in this paper is to propose an efficient, integral block driven algorithm of the RASSCF method, and to describe its implementation on a massively parallel computer. Our approach is based on a model space representation of the RAS active orbitals, and an efficient expansion of the model subspaces. This idea is inspired by the classic paper of Saunders and van Lenthe.⁸ The contributions of a block of integrals (with an identical representation in the model space) are computed together. The reconstruction of binary strings is avoided and the full list of substring addresses is constructed directly, using simple recursive algebraic rules (*propagation rules*). Finally, the proof of applicability of our RAS implementation is given in the form of an application in Sec. V.

II. THEORY

As in any other MCSCF method, we seek the solution of the CI eigenvalue problem,

$$\mathbf{HC} = \mathbf{CE}, \quad (1)$$

^{a)}Electronic mail: mike.rob主@kcl.ac.uk

where

$$\mathbf{H} = \{ \langle K | \hat{H} | L \rangle \} = \{ H_{KL} \} \quad (2)$$

is the representation matrix in a basis of orthonormal many-particle configuration state functions (CSF), which we denote as $\{|K\rangle\}$. In general, only the lowest eigenvalues and their eigenvectors are of interest. In practice, use is normally made of iterative eigenvector procedures, such as the methods of Lanczos¹⁰ or Davidson.¹¹ The time consuming step in these methods is a linear transformation of the CI vector,

$$\sigma = \mathbf{H}\mathbf{C}, \quad (3)$$

where \mathbf{C} is an approximate eigenvector from the previous iteration. The central practical problem is the evaluation of the $\{H_{KL}\}$. The general result can be expressed as

$$H_{KL} = \sum_{ij} (i|j) A_{ij}^{KL} + \frac{1}{2} \sum_{ijkl} (ij|kl) B_{ijkl}^{KL}, \quad (4)$$

where the summation is over orbital indices, and $(i|j)$ and $(ij|kl)$ are the usual one and two electron repulsion integrals. The A_{ij}^{KL} and B_{ijkl}^{KL} are numerical *vector coupling* coefficients that depend on the nature of $|K\rangle$ and $|L\rangle$. In second quantization, these numerical vector coupling coefficients emerge as matrix elements of creation and annihilation operators, $\hat{a}_{i\rho}^\dagger$ and $\hat{a}_{j\rho}$,

$$A_{ij}^{KL} = \langle K | \sum_{\rho} \hat{a}_{i\rho}^\dagger \hat{a}_{j\rho} | L \rangle, \quad (5)$$

$$B_{ijkl}^{KL} = \langle K | \sum_{\rho\gamma} \hat{a}_{i\rho}^\dagger \hat{a}_{k\gamma}^\dagger \hat{a}_{l\gamma} \hat{a}_{j\rho} | L \rangle, \quad (6)$$

where ρ, γ denote spin.

Each Slater determinant of the CI expansion can be written as the tensor product of two strings, one of the occupied α -orbitals, K_α , and one of the occupied β -orbitals, K_β ,

$$|K\rangle = |K_\alpha K_\beta\rangle. \quad (7)$$

A string K_ρ is an ordered product of creation operators acting on the vacuum, and can be represented by a binary word of length M , where each bit b_i represents a spin orbital. The bit value of b_i indicates whether the orbital i is occupied or empty, i.e., the binary representation of K_ρ , ϕ_{K_ρ} , contains N_ρ 1's and $(M - N_\rho)$ 0's. We denote by \mathcal{L} the list of all strings in a defined (*lexical*) order. Each string K_ρ can be identified by its address, $\mathcal{A}\{K_\rho\}$, which is identical to its position in \mathcal{L} . Knowles and Handy¹² have defined $\mathcal{A}\{K_\rho\}$ in full CI as

$$\mathcal{A}\{K_\rho\} = 1 + \sum_{k=1}^{N_\rho} Z(k, l(k)), \quad (8)$$

using the addressing array Z :

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

$$(M - N_\rho + k \geq l \geq k; \quad k < N_\rho),$$

$$Z(N_\rho, l) = l - N_\rho \quad (M \geq l \geq N_\rho), \quad (9)$$

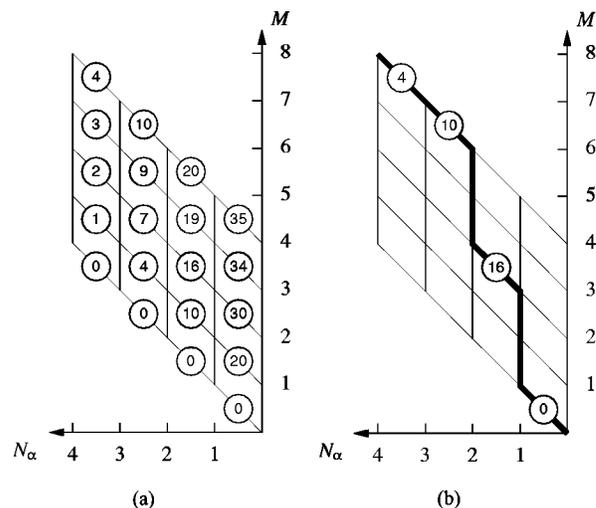


FIG. 1. (a): α -string graph for $N_\alpha=4$ electrons in $M=8$ orbitals. Every possible path from the bottom (foot) to the top (head) of the graph corresponds to an α -string. The orbitals are ordered and at each orbital level an α -electron may be added. At each vertex in the graph there are up to two paths upwards, corresponding to the next α -spin orbital being occupied (sloped path) or unoccupied. A lexical addressing of the strings is achieved by using Handy's addressing array (Ref. 12) [cf. Eqs. (8) and (9)]. The numbers on the slopes are the *arc weights*, $Z(k, l)$, and the address of any string is obtained simply by taking the sum of the arc weights. This addressing scheme corresponds to a strict left-to-right ordering of the strings. (b) The bold line represents the walk corresponding to the binary string 11001001. The corresponding string address is 31.

where k refers to an electron, l to an orbital, M is the number of orbitals, N_ρ the number of electrons and ρ denotes spin. This addressing scheme for strings in full CI is simple. However, in anticipation of the added complexity that will arise when addressing RAS strings we now introduce a graphical representation of Eq. (9). Figure 1(a) illustrates the α -string space for $M=8$ orbitals and $N_\alpha=4$ electrons.¹³ Each string corresponds to one *path* or *walk* on the grid of the graph. All paths begin at the foot of the graph, and finish in its head, advancing one level upwards for each orbital. If an orbital is occupied, the sloped segment step upwards is being used. The elements of the addressing array are added to the graph in Fig. 1(a). According to Eq. (8) the address of a string K_α is thus the sum of all circled values visited by the appropriate walk, plus 1. The β -string address is obtained in the same way. The address of the Slater determinant $|K\rangle$ may then be defined as

$$\mathcal{A}\{K\} = (\mathcal{A}\{K_\alpha\} - 1) \times \binom{M}{N_\beta} + \mathcal{A}\{K_\beta\}. \quad (10)$$

Equation (4) is a simple sum of products. While the one and two electron integrals are generally nonzero quantities, the coupling coefficients are mostly zero. Efficient computer implementations of any CI method take advantage of this fact. One can distinguish between two main types of strategies to deal with Eq. (4), namely the *configuration driven approach* (CDA) and the *integral driven approach* (IDA). In the CDA, given a configuration pair, $|K\rangle, |L\rangle$, all index pairs (i, j) and quadruples (i, j, k, l) that give nonzero coupling coefficients need to be found. In the IDA, by contrast, all configuration pairs $|K\rangle, |L\rangle$ for nonzero coupling coefficients

need to be found, given a set of orbital indices, i, j, k, l . In modern implementations both strategies have found their application. To a degree, the optimum approach will depend on the properties of the computer hardware, and efficient implementations that take advantage of vector processors have been devised. More recently, the development of computer hardware has been towards parallel configurations, where distributed and shared memory architectures are often found together, and vector architectures are less relevant. This should be reflected in the design of any implementation. In particular, methods for a well functioning load balancing should be considered.

The true computational cost of evaluating the matrix multiplication of Eq. (3) consists of finding all nonzero coupling coefficients, A_{ij}^{KL} , B_{ijkl}^{KL} , and performing the SAXPY operations,

$$\begin{aligned}\sigma_K &:= \sigma_K + (i|j)A_{ij}^{KL}C_L, \\ \sigma_K &:= \sigma_K + (ij|kl)B_{ijkl}^{KL}C_L,\end{aligned}\quad (11)$$

for all nonzero coupling coefficients. As the proportion of nonzero coupling coefficients is tiny [for example, $\ll 0.01\%$ for a CASSCF(12,12)] an efficient method of finding nonzero coupling coefficients *implicitly* and without trying or testing the complete space of orbital and/or configuration indices is thus highly desirable. To find a solution to this *indexing problem*, we start by rewriting the σ vector of Eq. (3) as a sum of 2 terms:

$$\sigma = {}^1e\sigma + {}^2e\sigma. \quad (12)$$

The one-electron term, ${}^1e\sigma_K$, may be written with the outer sum over orbital indices i and j and the inner sums over α -strings and β -strings,

$$\begin{aligned}{}^1e\sigma_{K_\alpha K_\beta} &= \sum_{ij} (i|j) \left(\sum_{L_\alpha} \langle K_\alpha | \hat{a}_{i\alpha}^\dagger \hat{a}_{j\alpha} | L_\alpha \rangle C_{L_\alpha K_\beta} \right. \\ &\quad \left. + \sum_{L_\beta} \langle K_\beta | \hat{a}_{i\beta}^\dagger \hat{a}_{j\beta} | L_\beta \rangle C_{K_\alpha L_\beta} \right), \quad \forall K_\alpha, K_\beta.\end{aligned}\quad (13)$$

Likewise, the two-electron term, ${}^2e\sigma_K$, may be rewritten similarly, yielding

$$\begin{aligned}{}^2e\sigma_{K_\alpha K_\beta} &= \sum_{ijkl} (ij|kl) \left(\sum_{L_\alpha} \langle K_\alpha | \hat{a}_{i\alpha}^\dagger \hat{a}_{k\alpha}^\dagger \hat{a}_{l\alpha} \hat{a}_{j\alpha} | L_\alpha \rangle C_{L_\alpha K_\beta} \right. \\ &\quad + \sum_{L_\beta} \langle K_\beta | \hat{a}_{i\beta}^\dagger \hat{a}_{k\beta}^\dagger \hat{a}_{l\beta} \hat{a}_{j\beta} | L_\beta \rangle C_{K_\alpha L_\beta} \\ &\quad + \sum_{L_\alpha} \langle K_\alpha | \hat{a}_{i\alpha}^\dagger \hat{a}_{j\alpha} | L_\alpha \rangle \sum_{L_\beta} \langle K_\beta | \hat{a}_{k\beta}^\dagger \hat{a}_{l\beta} | L_\beta \rangle C_{L_\alpha L_\beta} \\ &\quad + \sum_{L_\alpha} \langle K_\alpha | \hat{a}_{k\alpha}^\dagger \hat{a}_{l\alpha} | L_\alpha \rangle \\ &\quad \left. \times \sum_{L_\beta} \langle K_\beta | \hat{a}_{i\beta}^\dagger \hat{a}_{j\beta} | L_\beta \rangle C_{L_\alpha L_\beta} \right), \quad \forall K_\alpha, K_\beta.\end{aligned}\quad (14)$$

Thus, in Eqs. (13) and (14) the outer summation is over orbital indices i, j and i, j, k, l , respectively. This implies the use of an IDA, where the orbital indices and the repulsion

integrals are given, and all string pairs K_α, L_α and K_β, L_β that give a nonzero contribution to ${}^1e\sigma$ and ${}^2e\sigma$ need to be found.

In restricted CI methods in general, and in the RASSCF method in particular, the restrictions imposed on the active orbital space give rise to certain complications concerning the construction and indexing of CSFs and their strings. However, an efficient solution to the indexing problem should, nevertheless, as in the case of full CI/CASSCF implementations, aim to minimize or eliminate redundant index space testing. We propose in this paper a solution to this problem, based on two steps: the introduction of a *RAS model orbital space*, and the efficient reconstruction of string pairs K_α, L_α from the model strings in that space, using *propagation rules*.

The graphical representation of CSFs used in this work and outlined in Sec. II A greatly facilitates insight in and treatment of the RASSCF indexing problem. In Sec. II B we introduce the *model space* representation of the RAS space, again aided by a graphical representation. Based on the model space concept in Sec. II C we develop an efficient method for reconstructing the string pairs that give nonzero contributions to Eqs. (13) and (14), using *propagation rules*. The method described in this work entirely avoids potentially costly index space testing.

A. A graphical representation of the RAS configurations

The string space in the RASSCF method is a subset of the corresponding CAS string space, where the total number of active orbitals and electrons are identical. Due to the particle number restriction in the RAS1 and RAS3 orbital subspaces, the construction of CSFs is relatively more complicated. The complication arises on two levels: the indexing of strings and the combination of α -strings with β -strings. The remedy to both problems lies in the classification of certain groups of RAS strings. To this end we will make use of the concept of *string categories*.¹⁴ Within any string category indexing is then simple, as no restrictions apply. Allowed combinations of α -strings with β -strings are found by identifying allowed combinations of string categories.

It is convenient to define several graphs. Each graph corresponds to a subset of the paths on the FCI graph. The space of strings is then described by the sum of all possible walks on all graphs. In addition, there will usually be restrictions on the allowed combination of graphs. If the state of interest is a singlet, then the set of graphs for both spin spaces will be identical.

Figure 2 shows three example α -string graphs (of a total of nine graphs) for the case with $N_\alpha=6$, $M=12$, $M(\text{RAS1})=4$, $M(\text{RAS3})=4$ and a maximum of 2 electrons excited out of RAS1 and into RAS3, respectively. As in the CASSCF case, the graphical representation of the α -strings and β -strings may be used to order the paths. This is best done by assigning consecutive local addresses to strings within one graph. If the graphs themselves are ordered, then the global string address will be the sum of the local string

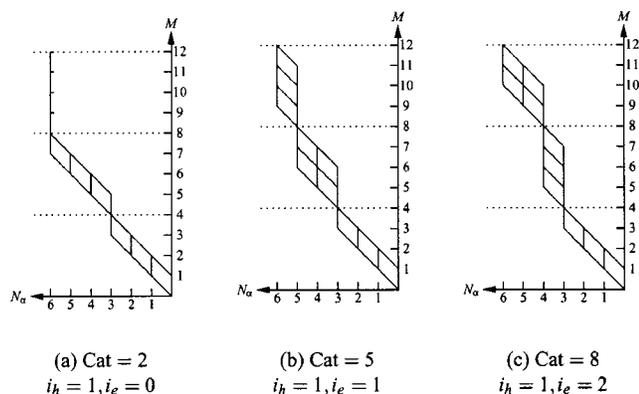


FIG. 2. Three example RAS α -string graphs for $N_\alpha=6$, $M=12$ with RAS subspace definitions $M(\text{RAS1})=4$; $\text{MxHole}=2$ and $M(\text{RAS3})=4$; $\text{MxElec}=2$. The total number of graphs (=categories) is here $(\text{MxHole}+1) \times (\text{MxElec}+1)=9$.

address and an offset accounting for all paths in preceding graphs.

Kozłowski and Pulay¹⁴ proposed a two level addressing scheme, the first level being a string category determined by the number of holes, i_h , in RAS1 and the number of electrons, i_e , in RAS3, while the second level gives the local string address within a given category. A category in this scheme corresponds exactly to a string graph as described above, and we will largely adopt the notation used in Ref. 14. The category $\text{Cat}(i_h, i_e)$ is defined as

$$\text{Cat}(i_h, i_e) = (i_h + 1) + (\text{MxHole} + 1)i_e, \quad (15)$$

and the number of categories (graphs) for each spin space is given by

$$\text{Cat}(\text{MxHole}, \text{MxElec}) = (\text{MxHole} + 1)(\text{MxElec} + 1). \quad (16)$$

The length of that category (number of paths in the corresponding graph) is

$$L[\text{Cat}(i_h, i_e)] = \binom{M(\text{RAS2})}{N_\alpha - M(\text{RAS1}) + i_h - i_e} \times \binom{M(\text{RAS1})}{i_h} \binom{M(\text{RAS3})}{i_e}. \quad (17)$$

The address of an α -string $K_\alpha^{i_h, i_e}$ in category $\text{Cat}(i_h, i_e)$ can then be calculated as

$$\mathcal{A}\{K_\alpha^{i_h, i_e}\} = \mathcal{A}\{K_\alpha\} + \sum_{\text{Cat}=1}^{\text{Cat}(i_h, i_e)-1} L[\text{Cat}], \quad (18)$$

where $\mathcal{A}\{K_\alpha\}$ denotes the local string address and the second term is the sum of lengths of all previous categories. The local address can be assigned in the following way: for a given category, $\text{Cat}(i_h, i_e)$, any string can be considered as a combination of appropriate subspace strings,

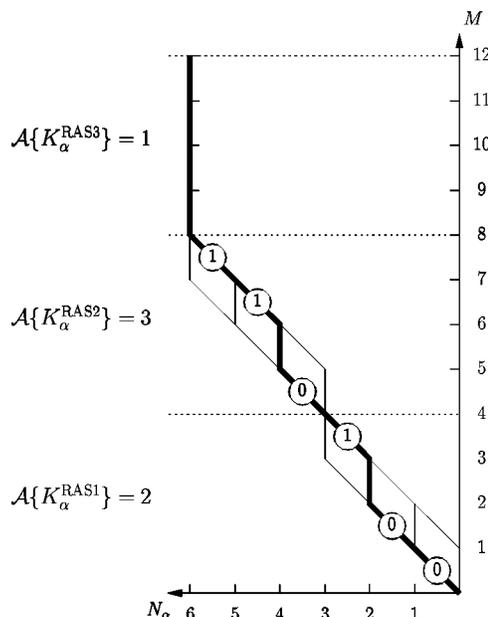


FIG. 3. An example of the addressing of an RAS α -string: $M(\text{RAS1})=M(\text{RAS2})=M(\text{RAS3})=4$, $K_\alpha=124578$, $\phi_{K_\alpha}=0000\ 1101\ 1011$, $i_h=1$, $i_e=0$. The numbers on the slopes are the arc weights, Z , for each subspace, as obtained from Eq. (9), and the resulting substring addresses are shown next to the corresponding graph segment. The local string address, $\mathcal{A}\{K_\alpha\}$, is then obtained from Eq. (19).

$$\begin{aligned} \mathcal{A}\{K_\alpha\} = & (\mathcal{A}\{K_\alpha^{\text{RAS2}}\} - 1) \binom{M(\text{RAS1})}{i_h} \binom{M(\text{RAS3})}{i_e} \\ & + (\mathcal{A}\{K_\alpha^{\text{RAS1}}\} - 1) \binom{M(\text{RAS3})}{i_e} + \mathcal{A}\{K_\alpha^{\text{RAS3}}\}, \end{aligned} \quad (19)$$

where each term K_α^{RAS1} , K_α^{RAS2} , K_α^{RAS3} denote the corresponding subspace strings. This is the crucial difference to the addressing scheme used by Olsen *et al.*,¹ who chose to not logically separate the addressing of the individual RAS subspaces. The advantage of defining the local string address as in Eq. (19) will become clear shortly.

A choice must be made as to how the subspace strings themselves should be addressed. Since the string subspaces resemble CASSCF string spaces, it is straightforward to again use the indexing formula of Knowles and Handy¹² [Eqs. (8) and (9)], i.e.,

$$\mathcal{A}\{K_\rho^{\text{RASX}}\} = 1 + \sum_{k=1}^{N_\rho^{\text{RASX}}} Z(k, l(k)). \quad (20)$$

The number of orbitals, M , and electrons N_ρ in Eq. (9) are substituted with the corresponding RAS subspace variables, $M(\text{RASX})$ and N_ρ^{RASX} , respectively, for the RASX subspace ($X \in \{1, 2, 3\}$).

An example may serve to clarify this. Consider a RAS problem with $M=12$ orbitals, $N_\alpha=6$ electrons, 4 orbitals in each RAS subspace, and maximum numbers of holes in RAS1 and particles in RAS3 both equal 2. One α -string that observes these restrictions is $K_\alpha=124578$. Its binary representation is $\phi_{K_\alpha}=0000\ 1101\ 1011$. The string category is $\text{Cat}(1,0)=2$ (i.e., $i_h=1$, $i_e=0$). Figure 3 shows the graphi-

cal representation of K_α as a walk on the corresponding RAS graph [cf. Fig. 2(b)]. The arc weights are obtained individually for each subspace graph [Eq. (9)] and the local string address is

$$\mathcal{A}\{K_\alpha\} = (2-1) \binom{4}{1} \binom{4}{0} + (3-1) \binom{4}{0} + 1$$

$$= 4 + 2 + 1 = 7.$$

Only the graph corresponding to $\text{Cat}=1$ precedes the current RAS graph, i.e., the lexical α -string address $\mathcal{A}\{K_\alpha^{i_h, i_e}\}$ is

$$\mathcal{A}\{K_\alpha^{1,0}\} = 7 + L[\text{Cat}(0,0)] = 7 + \binom{4}{2} \binom{4}{0} = 13.$$

The α -strings cannot be combined freely with all β -strings, because of the restrictions for the allowed number of holes ($i_h + i'_h \leq \text{MxHole}$; the prime ' signifies β -spin) in RAS1 and the allowed number of electrons ($i_e + i'_e \leq \text{MxElec}$) in RAS3. The resulting expression for valid SDs in the RAS expansion can be obtained using once again a two level addressing scheme. The local address for a string pair in a given set of string graphs, $\mathcal{A}\{K_\alpha, K_\beta\}$, is given by

$$\mathcal{A}\{K_\alpha, K_\beta\} = (\mathcal{A}\{K_\alpha\} - 1)L[\text{Cat}(i'_h, i'_e)] + \mathcal{A}\{K_\beta\}, \quad (21)$$

where $L[\text{Cat}(i'_h, i'_e)]$ is the total number of β -string walks on the corresponding β -string graph. The address of CSF $|K\rangle = |K_\alpha, K_\beta\rangle$ is then

$$\mathcal{A}\{K\} \equiv \mathcal{A}\{K_\alpha^{i_h, i_e}, K_\beta^{i'_h, i'_e}\} = \mathcal{A}\{K_\alpha, K_\beta\} + \mathcal{F}_{\text{Cat}}^{\text{Cat}'}, \quad (22)$$

where $\mathcal{F}_{\text{Cat}}^{\text{Cat}'}$ is the offset accounting for all CSFs prior to K . It is defined as

$$\mathcal{F}_{\text{Cat}}^{\text{Cat}'} = \sum_{\text{Cat}=1}^{\text{Cat}(i_h, i_e)-1} \sum_{\text{Cat}'=1}^{\text{Cat}(\text{MxHole}, \text{MxElec})} L[\text{Cat}] \times L[\text{Cat}']$$

$$+ \sum_{\text{Cat}'=1}^{\text{Cat}(i'_h, i'_e)-1} L[\text{Cat}(i_h, i_e)] \times L[\text{Cat}']. \quad (23)$$

Equation (21) assigns a unique local address to any combination of paths in a given α -string graph and a β -string graph (there are $L[\text{Cat}(i_h, i_e)] \times L[\text{Cat}(i'_h, i'_e)]$ such combinations). The offset, $\mathcal{F}_{\text{Cat}}^{\text{Cat}'}$, may be precomputed for every allowed graph combination. It accounts for all previous allowed combinations, giving the global address for the SD. Although we are conducting the discussion of the RAS algorithm on the basis of SDs as basis CSFs, note that it is straightforward to use spin-adapted CSFs, by appropriately redefining $\mathcal{A}\{K_\alpha, K_\beta\}$ and $\mathcal{F}_{\text{Cat}}^{\text{Cat}'}$.

It is apparent from Eq. (19), and perhaps even more so from the graphs in Fig. 2, that the subspace strings K_α^{RAS1} , K_α^{RAS2} , K_α^{RAS3} may be generated as all possible walks on their respective subspace graphs. The task of finding all allowed configurations in the RAS expansion is then simply to find all allowed combinations of subspace graphs.

B. The RAS model space

We will now introduce a *model space* representation of the RAS string space. The RAS model space closely re-

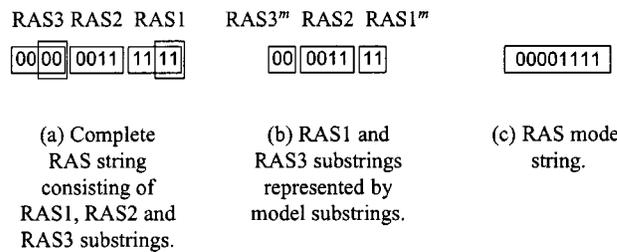


FIG. 4. An example of a RAS α -string consisting of four orbitals in each RAS subspace, and its representing model space string ($\text{MxHole}=2$, $\text{MxElec}=2$). The model string is constructed using the RAS2 substring and parts of the RAS1 and RAS3 substrings, of dimension MxHole and MxElec , respectively. The RAS model string has no occupancy restricted subspaces.

sembles CAS string spaces, thus simply avoiding the problems associated with the restrictions imposed on the RAS1 and RAS3 orbital subspaces.

Suppose we would like to construct a RAS wave function using the determinantal approach, with a maximum number of holes (MxHole) allowed in the RAS1 space and a maximum number of electrons (MxElec) allowed in the RAS3 space. A model RAS space may then be defined with M^m orbitals, given by

$$M^m = M(\text{RAS2}) + \text{MxHole} + \text{MxElec}, \quad (24)$$

and $N^m = N_\alpha^m + N_\beta^m$ electrons where

$$N_\alpha^m = N_\alpha - M(\text{RAS1}) + \text{MxHole} \quad (25)$$

and

$$N_\beta^m = N_\beta - M(\text{RAS1}) + \text{MxHole}. \quad (26)$$

The remaining (spin) orbitals excluded from this model space will always be occupied in the RAS1 subspace and unoccupied in the RAS3 subspace. Figure 4 shows a RAS type α -string consisting of RAS1, RAS2 and RAS3 strings, and the model space string representing it ($\text{MxHole}=2$, $\text{MxElec}=2$). Note that the number of electrons in the model space is constant (other than in the individual RAS subspaces).

It is useful to think of the model space as (1) a compact way of representing the total RAS space, or (2) a combination of the RAS2 space with compacted RAS1 and RAS3 subspaces.

These two interpretations are visualized in Fig. 5 and Fig. 6, respectively. The most striking feature of representation 1 (Fig. 5) of the model string space is the fact that, although representing restricted string spaces, they are entirely unrestricted and are in fact identical to those used in a CASSCF problem with $M(\text{CAS}) = M^m(\text{RAS})$ orbitals and $N(\text{CAS}) = N^m(\text{RAS})$ electrons. This paves the way for an efficient construction of all model strings, since this is a well studied problem. In particular, it is clear that the method of reduced excitation lists presented in Ref. 9 may be applied and we shall return to the details shortly.

Crucial to the usefulness of the model space idea is representation 2: the subdivision of the model space into the RAS2 subspace and compacted RAS1 and RAS3 model subspaces. Because the dimension of these model subspaces is MxHole and MxElec , respectively, the number of model

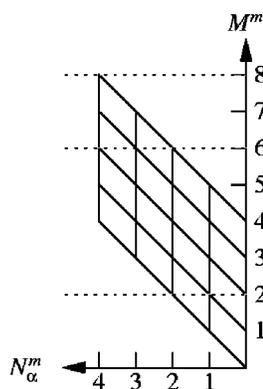


FIG. 5. A graph representing the complete α -model string space; $M^m = M(\text{RAS2}) + \text{MxHole} + \text{MxElec}$; [$M(\text{RAS2})=4$, $\text{MxHole}=\text{MxElec}=2$] and $N_\alpha^m=4$. The model string graph resembles a CAS string graph, due to the absence of occupancy restrictions in the RAS model space.

space graphs in this representation equals the number of categories as given by Eq. (16) for each set of RAS spin graphs. Furthermore, Eq. (15) is not dependent on any absolute number of particles in any RAS subspace, so that there is a one to one correspondence of each *complete* graph with one model graph (cf. Fig. 2 and Fig. 6). This property will be of particular use once a model space string has been found, and is central to the method described here. The idea is simple: once a model string is found, it may be associated with a certain model string space. It is then by implication also associated with a certain string graph in the complete RAS space.

We have already mentioned that the unrestricted character of the model string space may be exploited, since the problem of finding model string pairs $\{K_\rho^m, L_\rho^m\}$ closely resembles the corresponding problem for full CI. This problem is well studied and a number of efficient direct methods have been proposed (e.g., Refs. 1 and 15–17). Here we will use the method of reduced excitation strings, which we have introduced in Ref. 9 to entirely avoid index space testing in direct full CI/CASSCF computations. However, the model space and reduced string method are not interdependent, and any direct method tackling the indexing problem in full CI

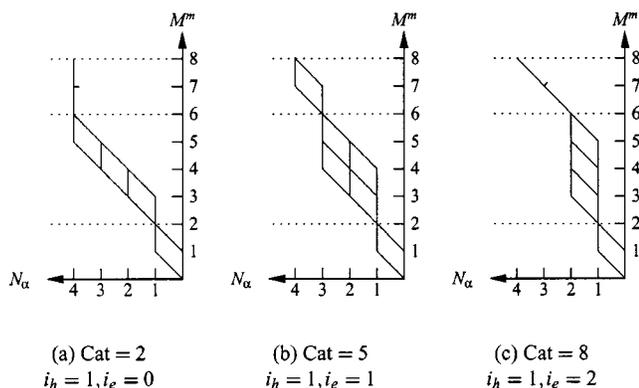


FIG. 6. Three RAS α -model string space graphs for $M(\text{RAS2})=4$, $\text{MxHole}=2$ and $\text{MxElec}=2$. Each graph corresponds to one string graph in Fig. 2, with equal string category. The superposition of all (9) restricted model graphs gives the unrestricted RAS model string graph of Fig. 5.

could be used. The reduced list concept may be applied here without change, and we will now briefly remind ourselves of the method, while placing it in the current context.

In order to avoid confusion between orbital labels of different RAS subspaces, the model space and the full RAS space, we will use separate orbital indices for labeling the respective orbitals: $\{a, b, c, d\} \in \text{RAS1}$; $\{i, j, k, l\} \in \text{RAS2}$; $\{p, q, r, s\} \in \text{RAS3}$; $\{w, x, y, z\} \in \text{model space}$; $\{i, j, k, l\} \in \text{full RAS space}$. Note that we use the same set of indices i, j, k, l for both the full RAS space and the RAS2 subspace. The reason for this is that it is (at different times) useful to liken both sets of orbitals to the active space of a CASSCF. It will always be clear from the context which orbital set is meant.

The set of orbitals under consideration is restricted to those contained in the RAS model space. Thus, given the ρ -model space orbital indices w, x , we define the model space excitation list $\mathcal{X}_{w\rho}^{x\rho}$ containing, in lexical order, all string pair addresses $\mathcal{A}\{K_\rho^m\}$, $\mathcal{A}\{L_\rho^m\}$ and $\text{sgn}_{wx}^{K_\rho^m}$ for which the relation

$$\langle K_\rho^m | \hat{a}_{w\rho}^\dagger \hat{a}_{x\rho} | L_\rho^m \rangle = \text{sgn}_{wx}^{K_\rho^m} \quad (27)$$

holds, with

$$\text{sgn}_{wx}^{K_\rho^m} = \begin{cases} +1, & \text{if } \sum_{n=x+1}^{w-1} b_n \text{ is even,} \\ -1, & \text{if } \sum_{n=x+1}^{w-1} b_n \text{ is odd,} \end{cases} \quad (28)$$

where b_n is the bit value of the n th bit in the binary representation of K_ρ^m , $\phi_{K_\rho^m}$. The sums represent the number of occupied orbitals (i.e., $b_n=1$) between orbitals w and x . K_ρ^m and L_ρ^m are constructed by inserting the appropriate bits b_w, b_x into the binary representation of a *reduced string*, consisting of M^m-2 bits, with N_ρ^m-1 1s. The list of all such reduced strings in lexical order is denoted $\mathcal{L}_{N_\rho^m-1}^{M^m-2}$.

Similarly, for $2e$ excitations a model space excitation list $\mathcal{X}_{w\rho y\rho}^{x\rho z\rho}$ is defined, containing all string pair addresses $\mathcal{A}\{K_\rho^m\}$, $\mathcal{A}\{L_\rho^m\}$ and $\text{sgn}_{wxyz}^{K_\rho^m}$ for which the relation

$$\langle K_\rho^m | \hat{a}_{w\rho}^\dagger \hat{a}_{y\rho}^\dagger \hat{a}_{z\rho} \hat{a}_{x\rho} | L_\rho^m \rangle = \text{sgn}_{wxyz}^{K_\rho^m} \quad (29)$$

holds, with

$$\text{sgn}_{wxyz}^{K_\rho^m} = \begin{cases} -\text{sgn}_{wx}^{K_\rho^m} \times \text{sgn}_{yz}^{K_\rho^m}, & \text{if } w > y > x > z; \\ +\text{sgn}_{wx}^{K_\rho^m} \times \text{sgn}_{yz}^{K_\rho^m}, & \text{otherwise.} \end{cases} \quad (30)$$

The model string pairs K_ρ^m, L_ρ^m are obtained by inserting the bits b_w, b_x, b_y, b_z into the corresponding reduced string. To accommodate for all possible $1e$ and $2e$ excitations we need only five reduced lists, which may be precomputed: $\mathcal{L}_{N_\rho^m-1}^{M^m-1}$, $\mathcal{L}_{N_\rho^m-1}^{M^m-2}$, $\mathcal{L}_{N_\rho^m-2}^{M^m-2}$, $\mathcal{L}_{N_\rho^m-2}^{M^m-3}$, $\mathcal{L}_{N_\rho^m-2}^{M^m-4}$. Thus all model space excitation lists in the model space may be constructed in this fashion, and the efficiency of this method comes from the fact that no redundant index space testing is needed in order to find the needed string pairs.

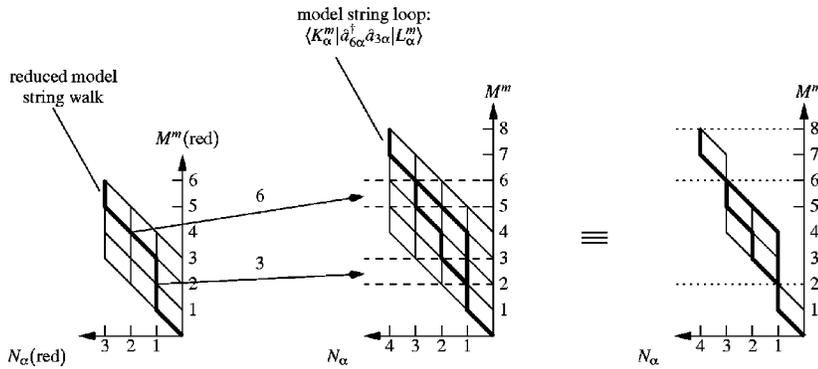


FIG. 7. The construction of the RAS model string pair K_α^m, L_α^m from a reduced model string, for the model space excitation $\hat{a}_{6\alpha}^\dagger \hat{a}_{3\alpha}$. The string K_α^m is represented by the left walk of the loop.

Figure 7 shows an example for the reconstruction of one string pair K_α^m, L_α^m , for the model space excitation $\hat{a}_{6\alpha}^\dagger \hat{a}_{3\alpha}$ and using an arbitrary reduced model string. The resulting walks for K_α^m and L_α^m form a loop on the model string graph. For the assembly of the model excitation list $\mathcal{X}_{3\alpha}^{6\alpha}$ the complete set of walks of reduced model strings (in lexical order), i.e., all entries of $\mathcal{L}_{N_\rho-1}^{M^m-2}$, will be used in the same manner. Once a model string pair K_ρ^m, L_ρ^m is found, each model string may be associated with one category, simply by counting the number of holes/electrons in the RAS1 and RAS3 model substrings. In graphical terms this means that for each walk corresponding to a ρ -model string, we need to find the ρ -model string space graph (cf. Fig. 6) that has the correct dimension to superimpose the walk onto it. If the loop stretches over more than one RAS subspace, the model strings K_ρ^m, L_ρ^m might then belong to different model space graphs (categories). In Sec. II C we will show how to expand the model substrings. However, the RAS2 substrings, K_ρ^{RAS2} and L_ρ^{RAS2} are already in the final form.

At this point we should mention an additional benefit of using the model space approach. In traditional implementations of the RASSCF method (e.g., Refs. 1 and 14), one has to deal with the possibility of *out of space* excitations, i.e., the excitation $\hat{a}_{i\rho}^\dagger \hat{a}_{j\rho} |L_\rho\rangle$ could result in a string $|K_\rho\rangle$ whose number of holes in RAS1 or the number of electrons in RAS3 exceed the limits, namely MxHole and MxElec , respectively. Due to the resemblance of the RAS model space to the string space for a full CI it is, by construction, not possible to produce an *out of space* excitation, and the problem is entirely avoided. This holds true also for double excitations.

C. An efficient expansion technique: Propagation rules

Each walk on a model string graph represents a set of strings on the full graph. Correspondingly, each pair of walks on the model string graph, related by a single ($\hat{a}_{w\rho}^\dagger \hat{a}_{x\rho}$) or double ($\hat{a}_{w\rho}^\dagger \hat{a}_{y\rho}^\dagger \hat{a}_{z\rho} \hat{a}_{x\rho}$) excitation and thus building a loop, represents a set of string pairs on the full graph. We now need to translate each model string pair K_ρ^m, L_ρ^m , together with its associated parity factor, $\text{sgn}_{wx}^{K_\rho^m}$, into a set of RAS string pairs and their parity factor, $\{K_\rho^{i_h, i_e}, L_\rho^{i'_h, i'_e}, \text{sgn}_{ij}^{K_\rho}\}$. While the RAS2 substring is fully determined by the model

string, the RAS1 and RAS3 substrings need to be evaluated separately by expanding the model substrings.

To achieve this, we propose a set of *propagation rules*. The aim of the propagation rules is to efficiently compute all string pairs, $\{K_\rho, L_\rho\}$, that give nonzero contributions for a given excitation term $\langle K_\rho | \hat{a}_{i\rho}^\dagger \hat{a}_{j\rho} | L_\rho \rangle$. All RASX substrings that are represented by a model substring can be constructed by propagating the binary representation of the excitation operator from the right to the left of a binary representation of the remaining orbitals, which we will denote r.s. Importantly, it can be shown that, by defining the address of subspace strings through Eq. (20) the actual construction of the subspace strings is not necessary and the substring addresses can be obtained directly. This is done by means of simple recursive evaluation of blocks of substring addresses $\{\mathcal{A}\{K_\rho^{\text{RASX}}(2)\}, \dots, \mathcal{A}\{K_\rho^{\text{RASX}}(n)\}\}$ from the first substring address $\mathcal{A}\{K_\rho^{\text{RASX}}(1)\}$ of each block.

Propagation Rule 1. Table I shows a series of RASX string pairs. They are constructed from the complete list of strings generated from two 0's and two 1's, ordered according to Handy's index equation [Eq. (9)] and supplemented by the short strings **01** and **10** to the right. The resulting list is, by construction, consecutive elements of a list of strings generated from three 0's and three 1's, also ordered according to Handy's index equation. If we refer to consecutive strings of the short list, $\{0011, \dots, 1100\}$, as r.s. and r.s.', respectively, then, according to Eq. (20) we have the trivial relationship

$$\begin{aligned} \mathcal{A}\{K_\rho^{\text{RASX}}(1, \text{r.s.}')\} &= \mathcal{A}\{K_\rho^{\text{RASX}}(1, \text{r.s.})\} + 1, \\ \mathcal{A}\{L_\rho^{\text{RASX}}(1, \text{r.s.}')\} &= \mathcal{A}\{L_\rho^{\text{RASX}}(1, \text{r.s.})\} + 1, \end{aligned} \tag{31}$$

TABLE I. The addresses of RAS1/RAS3 substrings for consecutive reduced strings are themselves consecutive numbers. This follows directly from the strict left-to-right ordering of strings on the corresponding unrestricted substring graph.

$\phi_{K_\rho^{\text{RASX}}}$	$\phi_{L_\rho^{\text{RASX}}}$	$\mathcal{A}\{K_\rho^{\text{RASX}}\}$	$\mathcal{A}\{L_\rho^{\text{RASX}}\}$
001 101	001 110	5	11
010 101	010 110	6	12
100 101	100 110	7	13
011 001	011 010	8	14
101 001	101 010	9	15
110 001	110 010	10	16

TABLE II. Propagation of the variable bits from left to right through all bit-positions of a fixed reduced string (01011) results in a change of substring address only if the two interchanged bits, b_a and b_{a-1} , differ. The number by which the address changes corresponds to a precomputable binomial coefficient.

a	$\phi_{K_\rho}^{\text{RASX}}$	$\phi_{L_\rho}^{\text{RASX}}$	b_{a-1}	$\binom{M(X)-a}{N_\rho^{\text{left}}(a)}$	$\mathcal{A}\{K_\rho^{\text{RASX}}\}$	ΔK	$\mathcal{A}\{L_\rho^{\text{RASX}}\}$	ΔL
1	010 111	010 110	n/a	10	2	-	12	-
2	010 111	010 101	1	6	2	0	6	-6
3	010 111	010 011	1	3	2	0	3	-3
4	0 11 011	0 10 011	0	2	4	2	3	0
5	0 11 011	001 011	1	1	4	0	2	-1
6	101 011	001 011	0	1	5	1	2	0

where the first argument (1) simply indicates an unchanged position of the added (bold) bits.

Propagation Rule II. Table II shows a series of string pairs. They are constructed from a constant binary string, 01011 and a bit b_a whose position a is variable. The bit value of b_a is 1 for $\phi_{K_\rho}^{\text{RASX}}$ and 0 for $\phi_{L_\rho}^{\text{RASX}}$. As b_a propagates from the right to the left of the strings, the total string addresses change if the bit values in fixed bit positions in the strings change. In particular, if after the propagation of the variable bit to position a (from position $a-1$) we have $b_a = b_{a-1}$, the string and its address have not changed. If in contrast $b_a \neq b_{a-1}$, the string address has changed by a number equal to the number of binary permutations of all bits to the left of position a , represented by the binomial coefficient $\binom{M(\text{RASX})-a}{N_\rho^{\text{left}}(a)}$, where $N_\rho^{\text{left}}(a)$ simply denotes the number of 1's to the left of position a . This is again due to the nature of Handy's index equation. If $b_a = 1$; $b_{a-1} = 0$ the string address rises, if $b_a = 0$; $b_{a-1} = 1$ the string address lowers, i.e.,

$$\Delta a = 1: \Delta \mathcal{A}\{K_\rho^{\text{RASX}}\} = (b_a - b_{a-1}) \binom{M(\text{RASX}) - a}{N_\rho^{\text{left}}(a)}. \quad (32)$$

These two propagation rules indicate an efficient algorithm for the computation of the remaining RAS1/RAS3 string addresses, if the first address of a given list is known. It is necessary to know only a number of reduced RAS1 and RAS3 string lists and some binomial coefficients, both of which may be precomputed at certain stages of the computation. The outlined scheme avoids redundant index space testing in a similar fashion the reduced list approach devel-

oped in Ref. 9 does for unrestricted CI. The algorithm is based on the systematic propagation of substring variables a, b, c, d (or p, q, r, s) over the entries of a reduced string list of the RAS1 (or RAS3) orbital space.

It can be shown that all possible lists of string addresses can be constructed using the *propagator function*,

$$p(u) = \delta_{1\Delta u} \binom{M(\text{RASX}) - u}{N_\gamma(M^{\text{left}}(u))} (b_u - b_{a-\xi}(\text{r.s.})), \quad (33)$$

where $u \in \{a, b, c, d\}$ and the parameter ξ is the number of variable indices in the RASX substring. The propagation rule, \mathcal{P} , for the RASX subspace may then be expressed as

$$\mathcal{A}\{K_\gamma^{\text{RASX}}(n, \text{r.s.})\} = \mathcal{A}\{K_\gamma^{\text{RASX}}(n-1, \text{r.s.})\} + \sum_{v=1}^{\xi} p(u(v)), \quad (34)$$

for a given reduced string (r.s.). When proceeding to the next reduced string (in the case of 0 indices in RASX each entry may be interpreted as a reduced string), the relationship is trivial, as shown above,

$$\mathcal{A}\{K_\gamma^{\text{RASX}}(1, \text{r.s.}')\} = \mathcal{A}\{K_\gamma^{\text{RASX}}(1, \text{r.s.})\} + 1. \quad (35)$$

If the number of holes and electrons in the respective subspaces RAS1 and RAS3 is greater than 2, care must be taken to ensure each term $\langle K_\rho | \hat{a}_{i\rho}^\dagger \hat{a}_{j\rho} | L_\rho \rangle$ is represented by only one model space term $\langle K_\rho^m | \hat{a}_{w\rho}^\dagger \hat{a}_{x\rho} | L_\rho^m \rangle$, in order to avoid multiple contributions. Figure 8 illustrates the problem. The same string pair, K_ρ, L_ρ , is arrived at by expanding the model space string pairs in the complete RAS space, using the appropriate propagation rules. The string pair,

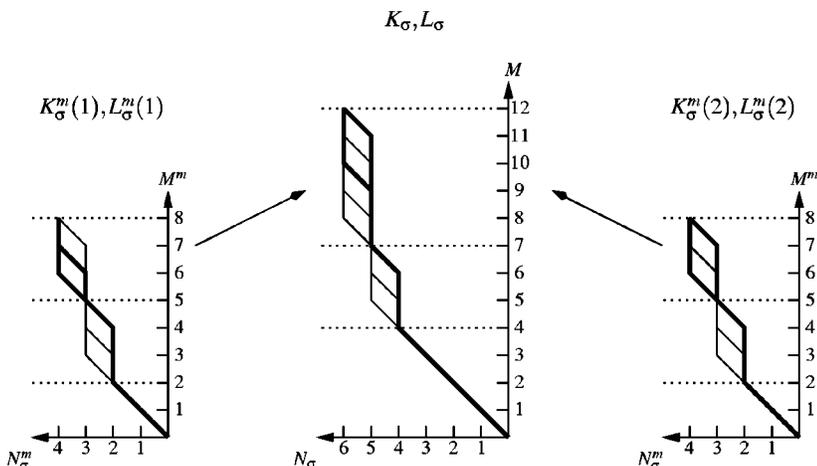


FIG. 8. Unwanted double contributions may arise when two model string pairs generated by different terms $\langle K_\rho^m | \hat{a}_{w\rho}^\dagger \hat{a}_{x\rho} | L_\rho^m \rangle$ result in the same contributing term $\langle K_\rho | \hat{a}_{i\rho}^\dagger \hat{a}_{j\rho} | L_\rho \rangle$, through an application of the propagation rules. The example shown here illustrates the problem: the two model space excitations $\langle K_\rho^m | \hat{a}_{7\rho}^\dagger \hat{a}_{6\rho} | L_\rho^m \rangle$ and $\langle K_\rho^m | \hat{a}_{8\rho}^\dagger \hat{a}_{6\rho} | L_\rho^m \rangle$ both give (among others) the contribution $\langle K_\rho | \hat{a}_{12\rho}^\dagger \hat{a}_{10\rho} | L_\rho \rangle$.

K_ρ, L_ρ , has been constructed using the same propagation rule, but starting from different model string pairs, $K_\rho^m(1), L_\rho^m(1)$ and $K_\rho^m(2), L_\rho^m(2)$. Each model string pair is related to a different excitation operator in the model space, namely $\hat{a}_{7\rho}^\dagger \hat{a}_{6\rho}$ and $\hat{a}_{8\rho}^\dagger \hat{a}_{6\rho}$, respectively. This problem is, however, easily avoided by requiring the model space indices w, x, y, z to be flush-right in the RASX^m model subspaces. In graphical terms this means that loop-opening and loop-closing segments will be concentrated on the lowest possible orbital level in the RASX^m model subspaces (cf. Fig. 8). It is then straightforward to see that the only restrictions on the values of MxHole and MxElec are

$$0 \leq \text{MxHole} \leq 2M(\text{RAS1}),$$

$$0 \leq \text{MxElec} \leq 2M(\text{RAS3}).$$

In the special case of both MxHole and MxElec assuming their maximum values the configuration spaces of CAS and RAS become identical.

The 2e Excitation Contribution. The 2e excitation term [Eq. (14)] of the vector σ [Eq. (3)] is by far the computationally more expensive one. It consists of two parts that differ conceptually. The $\alpha\alpha/\beta\beta$ parts (first two terms) each represent a double excitation involving electrons of one spin type only. All that is needed to accommodate for the double excitations in a single string space, using the model space approach, is an extension of the model subspace by up to two extra model space orbitals for each model subspace. The $\alpha\beta$ -part and $\beta\alpha$ -part [the last two terms in Eq. (14)] are simply combinations of single excitations in both the α -string space and the β -string space. Thus, for the purpose of finding all spin-string pairs, the $\alpha\beta$ part of the 2e excitation may be treated in an identical fashion to the 1e excitation.

As in the case of the 1e excitation, out-of-space excitations within a spin-string are entirely avoided by construction, through usage of the model space approach. The combination of two valid model string pairs for the $\alpha\beta$ -part and $\beta\alpha$ -part may, though, result in an out-of-space excitation. However, such a combination may simply be rejected at the model space level, before any expansion of external subspaces takes place.

The symmetry properties of the two-electron integrals $(ij|kl)$ mean that it is possible to reduce the loops over i, j, k, l to unique integrals and the summation in Eq. (4) over $i \geq j, k \geq l$ and $i(i-1)/2 + j \geq k(k-1)/2 + l$. By introducing the model space we replace summation over orbital indices i, j, k, l with model space orbital indices w, x, y, z . A single set of model space indices potentially represent a set of orbital indices. Nevertheless, the symmetry condition also holds for the RAS model space:

$$w \geq x, \quad y \geq z \quad (wx) \geq (yz),$$

with (36)

$$(wx) = \frac{w(w-1)}{2} + x, \quad (yz) = \frac{y(y-1)}{2} + z.$$

Because of these restrictions we will only find a restricted number of excitation combinations between RAS subspaces.

TABLE III. Possible combinations of excitations between RAS subspaces. (1→2 reads: excitation from RAS1 to RAS2).

y→z	w→x					
	1→1	1→2	2→2	1→3	2→3	3→3
1→1	×	×	×	×	×	×
1→2		×	×	×	×	×
2→2			×	×	×	×
1→3				×	×	×
2→3					×	×
3→3						×

The possible combinations are shown in Table III. Assuming MxHole ≥ 2 and MxElec ≥ 2, how many excitations $x \rightarrow w$ are possible? There are only two possibilities for excitations of type RAS1→RAS1 (1→1), namely $w \rightarrow w$ and $x \rightarrow w$. Likewise there are only two realizations for excitations of type 3→3. Excitations 1→3 are all represented by one model space excitation, $x \rightarrow w$. We have $M(\text{RAS2})$ possible model space excitations $x \rightarrow w$ of type 1→2 (one for each RAS2 orbital), and also $M(\text{RAS2})$ excitations of type 2→3, again, one for each RAS2 orbital. Finally, there are $\binom{M(\text{RAS2})}{2}$ possibilities for excitations of type 2→2. Table IV shows these numbers analytically and numerically for $M(\text{RAS2}) = 3, 6$ and 10. Table V summarizes the number of unique index quadruples, due to the restriction on possible combinations of (wx) with (yz) [cf. Eq. (36)]. The following can be seen.

(1) The contribution of $(w \rightarrow x, y \rightarrow z) = (2 \rightarrow 2, 2 \rightarrow 2)$ is the largest one and its relative importance grows with increasing size of the RAS2 subspace, $M(\text{RAS2})$. This means that in most cases the extension of RAS1 and RAS3 subspaces will simply result in a list of consecutive subspace addresses, and that both sign, $\text{sgn}_{ijkl}^{K_\rho}$, and integral, $(ij|kl)$, are constant for all contributions derived from a model space string pair.

(2) All other relatively large contributions stem from excitations with at least one of the two excitations being a 2→2. The advantage in this case is that the problem of finding relevant pairs $K_\alpha^{i_h \cdot i_e}, L_\alpha^{i_h \cdot i_e}$ is very similar to the 1e case.

In summary, most double excitations are computed from consecutive CSFs, with constant integral $(ij|kl)$ and parity factor $\text{sgn}_{ijkl}^{K_\rho}$. This part is getting relatively more important as the system size grows. The next most relevant part consists logically of a 1e excitation type problem. This is particularly true for the $\alpha\alpha$ 2e part. In Table VI the dimension

TABLE IV. Number of model space excitations $x \rightarrow w$ between RAS model spaces, provided MxHole ≥ 2 and MxElec ≥ 2; analytically and numerically for different RAS2 orbital subspace sizes. (1→2 reads: excitation from RAS1 to RAS2).

	1→1	1→2	2→2	1→3	2→3	3→3
$M(\text{RAS2})$	2	$M(\text{RAS2})$	$\frac{1}{2}M(\text{RAS2})^2 + M(\text{RAS2})$	1	$M(\text{RAS2})$	2
3	2	3	6	1	3	2
6	2	6	21	1	6	2
10	2	10	55	1	10	2

TABLE V. Number of double excitations $x \rightarrow w$, $z \rightarrow y$ between RAS model orbital subspaces; analytically and numerically for different RAS2 orbital subspaces. [Constructed using Tables III and IV and Eq. (36)].

$x \rightarrow w$	$z \rightarrow y$	Number of combinations			
		Analytically	$M(\text{RAS2}) =$		
			3	6	10
1→1	1→1	4	4	4	4
1→2	1→1	$2 \times M(\text{RAS2})$	6	12	20
	1→2	$\frac{1}{2} (M(\text{RAS2})^2 + M(\text{RAS2}))$	6	21	55
2→2	2→2	$\frac{1}{6} (M(\text{RAS2})^3 - M(\text{RAS2}))$	4	35	165
	1→1	$M(\text{RAS2})^2 + M(\text{RAS2})$	12	42	110
	1→2	$\frac{1}{6} (2M(\text{RAS2})^3 + 3M(\text{RAS2})^2 + M(\text{RAS2}))$	14	91	385
1→3	2→2	$\frac{1}{8} (M^4 + 2M^3 + 3M^2 + 2M)$	21	231	1540
	1→1	2	2	2	2
	1→2	$M(\text{RAS2})$	3	6	10
	2→2	$\frac{1}{2} (M(\text{RAS2})^2 + M(\text{RAS2}))$	6	21	55
2→3	1→3	1	1	1	1
	1→1	$2 \times M(\text{RAS2})$	6	12	20
	1→2	$M(\text{RAS2})^2$	9	36	100
	2→2	$\frac{1}{2} (M(\text{RAS2})^3 + M(\text{RAS2})^2)$	18	126	550
	1→3	$M(\text{RAS2})$	3	6	10
3→3	2→3	$\frac{1}{2} (M(\text{RAS2})^2 + M(\text{RAS2}))$	6	21	55
	1→1	2	4	4	4
	1→2	$2 \times M(\text{RAS2})$	6	12	20
	2→2	$M(\text{RAS2})^2 + M(\text{RAS2})$	12	42	110
	1→3	2	2	2	2
	2→3	$2 \times M(\text{RAS2})$	6	12	20
	3→3	4	4	4	4
Σ			155	743	3242

of the discussed cases is summarized analytically and for the same example cases already used above. The two dominating contributions are becoming more important with the growing size of the orbital space, and as a consequence, the efficiency of the propagation rule scheme becomes relatively greater for larger problems.

D. The effect of propagation rules on the order of integrals and the computation of parity factors

In an actual implementation of the propagation rule algorithm, once a model string pair is found for both the

TABLE VI. General overview of the total numbers of distinctive w, x, y, z quadruples for $MxHole=2$, $MxElec=2$, analytically and numerically for a few RAS2 subspace sizes.

Number of indices w, x, y, z in RAS2	Total dimension ...	$M(\text{RAS2}) =$		
	... analytically	3	6	10
4	$\frac{1}{8} (M^4 + 2M^3 + 3M^2 + 2M)$	21	231	1540
3	$M(\text{RAS2})^3 + M(\text{RAS2})^2$	36	252	1100
2 ^a	$\frac{7}{2} (M(\text{RAS2})^2 + M(\text{RAS2}))$	42	147	385
2 ^b	$M(\text{RAS2})^2$	9	36	100
1	$10M(\text{RAS2})$	30	60	100
0	17	17	17	17
Σ		155	743	3242

^aWithout excitations of type 2→3 and 1→2.

^bExcitations of type 2→3 and 1→2 only.

α -spin and β -spin model string space, the RAS1 and RAS3 subspace strings need to be expanded. This will be done in a nested loop over all RAS1 and RAS3 expansions. If all model space indices w, x (w, x, y, z) lie within the RAS2 subspace, then all contributions will be for the same integral ($i|j$) ($(ij|kl)$). If, on the other hand, some or all model space indices fall into one or both of the other subspaces, the contributions will be for a list of integrals. It is therefore useful to assemble the corresponding list of integrals, for every set of model space indices, in the order they will be accessed. This order is entirely given by the propagation rules, and only one list needs to be assembled for any set of model space indices. The integral lists for $\alpha\alpha$ -, $\alpha\beta$ -, $\beta\alpha$ - and $\beta\beta$ -contributions will be identical and need to be assembled only once.

It is by now clear that the value associated with a bra-ket term of the form $\langle K_\rho | \hat{a}_{i\rho}^\dagger \hat{a}_{j\rho} | L_\rho \rangle$ may differ from its model space analog $\text{sgn}_0^{m,\rho} = \langle K_\rho^m | \hat{a}_{w\rho}^\dagger \hat{a}_{x\rho} | L_\rho^m \rangle$ at most by a factor of -1 . If, for example, the index i lies within the RAS3 orbital subspace, then the propagation of the appropriate bit towards the left will result in a sign change every time the bit value of b_{i-1} equals 1. Because of this behavior and the direct association of the sign change with the propagation of variable bits, it is convenient to pre-compute the sign-changing factors for each spin subspace together with the RAS1 and RAS3 subspace string address pairs. The term $\text{sgn}_{ij}^{K,\rho} = \langle K_\rho | \hat{a}_{i\rho}^\dagger \hat{a}_{j\rho} | L_\rho \rangle$ is then computed as

$$\text{sgn}_{ij}^{K_\alpha} = \text{sgn}_0^{m,\alpha} \times \text{sgn}_{ij}^{K_\alpha^{\text{RAS1}}} \times \text{sgn}_{ij}^{K_\alpha^{\text{RAS3}}} \quad (37)$$

The same applies without change to the $2e$ equivalent, $\text{sgn}_{ij}^{K_\rho} = \langle K_\rho | \hat{a}_{i\rho}^\dagger \hat{a}_{k\rho}^\dagger \hat{a}_{i\rho} \hat{a}_{j\rho} | L_\rho \rangle$,

$$\text{sgn}_{ijkl}^{K_\alpha} = \text{sgn}_0^{m,\alpha} \times \text{sgn}_{ijkl}^{K_\alpha^{\text{RAS1}}} \times \text{sgn}_{ijkl}^{K_\alpha^{\text{RAS3}}} \quad (38)$$

III. THE DIRECT RAS ALGORITHM

The algorithm that follows from the use of an integral driven CI is inherently parallel, and the parallel loop may include any combination of orbital indices i, j, k, l . We will now provide the algorithms for one- and two-electron excitation term contributions of Eq. (12). All reduced lists of model space and RAS subspaces and the associated Handy's index matrices are precomputed only once. The outer loop is over the model space orbital indices w, x, y, z . The computation of model space string pairs is analogous to string pair computation as described in Ref. 9 for full CI string pairs. Once the model string pair K_ρ, L_ρ is known, its category and propagation rules for both the RAS1 and RAS3 subspaces are identified. This information is used to assemble the list of integrals into a vector, where they will be accessed in a sequential manner during updating of the σ -vector. Then all excitation lists are assembled, by application of the propagation rules to the RAS subspace model strings, RAS1^m and RAS3^m . We denote these subspace excitation lists $\mathcal{E}_\rho^{\text{RASX}}$. Finally, nested loops over subspace excitation list entries lead to the local address, and by combining α - and β -strings, to the CSF address. The appropriate algorithms for $2e \alpha\alpha$ and $2e \alpha\beta$ parts, respectively, of the σ -vector are summarized in the Appendix.

The dimension of the integral list is a function of the number of orbital indices in RAS1 and RAS3, $n1$ and $n3$, respectively,

$$D(n1, n3) = \begin{pmatrix} M(\text{RAS1}) \\ n1 \end{pmatrix} \begin{pmatrix} M(\text{RAS3}) \\ n3 \end{pmatrix} \begin{cases} 0 \leq n1 \leq 4, \\ 0 \leq n3 \leq 4 - n1 \end{cases} \quad (39)$$

If all orbital indices lie within RAS2, then the integral list contains only one element. The maximum length of the integral vector depends on the relative size of the RAS1 and RAS3 orbital subspaces. In the special case of $M(\text{RAS1}) = M(\text{RAS3}) > 2$ the maximum dimension is for two indices each in the RAS1 and RAS3 subspaces,

$$D(2,2) = \binom{M(\text{RAS1})}{2} |M(\text{RAS1})| \\ = M(\text{RAS3}) > 2 = \left(\frac{M(\text{RAS1})^2 - M(\text{RAS1})}{2} \right)^2$$

For example, with $M(\text{RAS1}) = M(\text{RAS3}) = 10$, the maximum length of the integral vector is 2025. At any stage the length of the integral vector will be small when compared to the CI vector. The memory requirements for the reduced lists of RAS subspace strings and model space strings are equally small compared to the CI vector, as are the partial excitation strings, $\mathcal{E}_\rho^{\text{RASX}}$.

Thus the memory requirements are dominated by the vectors σ and C . However, if the working memory is not large enough to accommodate the entire CI vectors, it is straightforward to adapt the algorithm to use only certain blocks of these vectors, by taking advantage of the blocking of string addresses of all strings that belong to a certain graph. This means that the vectors σ and C with elements $\sigma_{K_\alpha K_\beta}$ and $C_{K_\alpha K_\beta}$ are also blocked with respect to graph combinations (categories). Since the graph combination is fixed for all contributions of a found model string pair, it is possible to adapt the described scheme by either reading in the required block of the CI vectors from distributed memory, or by restricting contributions to a given CI vector block at a time.

Particularly in the first iterations of a typical MCSCF computation, there are only very few nonzero elements $C_{K_\alpha K_\beta}$. This means that in principle, only those contributions,

$$\begin{aligned} \sigma_{L_\alpha L_\beta} &:= \sigma_{L_\alpha L_\beta} + (i|j) A_{ij}^{KL} C_{K_\alpha K_\beta}, \\ \sigma_{L_\alpha L_\beta} &:= \sigma_{L_\alpha L_\beta} + (ij|kl) B_{ijkl}^{KL} C_{K_\alpha K_\beta}, \end{aligned} \quad (40)$$

with $C_{K_\alpha K_\beta} \neq 0$ need to be considered to compute σ . This is an inherent weakness of the integral driven approach, since the resulting algorithm does not provide for efficient exclusion of blocks of the CI vector. However, to increase efficiency in the model space approach proposed here, one could assemble a short list of flags that indicate whether *any* of the α -strings obtained by expanding a α -model string belongs to a block of C with at least one nonzero element. An analogous list would be needed for β -string blocks. The most simple realization of this idea would be to define the blocks as having contributions of a unique α -graph/ β -graph combination. More elaborate methods could include, for example, lists for specific RAS2 substrings, possibly combined with the appropriate propagation rules. Finally, it should be noted that use of the propagation rules means that it is not straightforward to take advantage of point group symmetry.

IV. PARALLELIZATION

We will now describe the parallel implementation of the direct RASSCF algorithm. A thorough discussion of the method of parallelization used and its performance can be found in Ref. 9. We assume a scalable parallel distributed memory computer architecture consisting of nodes, each with local memory. The nodes themselves may be symmetric multiprocessor (SMP) machines with shared memory. As mentioned in the previous section, the integral driven algorithm is already inherently parallel. The main issue to address is thus how to ensure good load balancing. A flexible load balancing mechanism helps to achieve optimum scaling and is essential in heterogeneous environments, where a cluster may consist of nodes of different architecture. It is also important in the realistic situation of uneven preloading of allocated nodes, i.e., in situations where some workers must share resources with concurrently running programs on the same node.

TABLE VII. The number of independent parallel tasks, defined as unique model space index quadruples, w,x,y,z , for various RAS2 orbital subspaces ($MxHole=MxElec=2$).

M (RAS2)	Number of parallel tasks	
	Index pairs (w,x)	Index quadruples (w,x,y,z)
4	23	410
6	38	947
8	57	1947
10	80	3638

Load balancing can be achieved when the total number of independent, parallel tasks is large compared to the number of processing elements (PEs, also: CPUs). The overhead of subdividing the total work into parallel sub-tasks should ideally be kept small compared to total execution time. Communication between workers should be avoided and should be handled by a separate process, in order to avoid synchronization delays.¹⁸

The sub-tasks that are executed on parallel nodes correspond to model space index pairs w,x ($1e$ contribution), or model space index quadruples, w,x,y,z ($2e$ contribution). The relatively small size of the model space (as opposed to the orbital space comprising all active orbitals) means the total number of independent tasks is much reduced. Table VII shows for different sizes of the RAS2 subspace and $MxHole=MxElec=2$ the number of parallel tasks for both the $1e$ and $2e$ contributions. The number of tasks is independent of the size of the RAS1 and RAS3 subspaces, and lies within a useful range, i.e., there are sufficiently many sub-tasks to enable load balancing for a large number of PEs, even for the smallest RAS2 orbital subspace. For larger RAS2 subspaces, in order to reduce overhead due to communication, several subtasks may be grouped together in order to adapt granularity of the parallelism. This may be done dynamically, in order to adapt to the number of requested or allocated CPUs.

The IDA approach followed by us here involves the outer loops over the model space orbital indices w,x,y,z . Inside these nested loops, the assembly of certain blocks of the vector σ takes place. However, with the exception of the CI vector σ itself, all data inside the loops are static. Therefore, we can minimize communication overhead by passing these static data to all worker nodes only once at the beginning of each eigenvector iteration. This is of advantage especially for distributed memory architectures, and, in particular, for low-cost configurations with standard network interprocessor communication. On shared memory architectures (e.g., an SMP node) these static data need to exist only once.

We now discuss the implementation of the general strategy just discussed for distributed memory architectures using Linda.¹⁸ We begin with a few definitions. We assume that each node may be an SMP. The number of processors on each SMP is indicated as $NProcS$ (on single processor nodes $NProcS=1$). The value $NTask$ is computed, with $NTask=NProcS \times H$ and H is chosen to both optimize load-balancing and minimize communication overhead. The main

process is called the master process (master). The process spawned on the first processing node retrieves the index ($wxyz$) (initially $(wxyz)=(wx)_{\max}((wx)_{\max}+1)/2$ with $(wx)_{\max}=M^m(M^m+1)/2$). A list of $NTask$ indices is assembled on the worker, by testing the validity of $(wxyz)$, storing it if valid, testing $(wxyz)-1$, and so forth. Invalid indices are simply dropped when found. The worker then puts a new index $(wxyz)'=(wxyz)-NTask-N_{\text{invalid}}$ into tuple space (the virtual shared memory area in the Linda model), which is then retrieved by the next worker, etc. If $NProcS>1$, then $NProcS-1$ shared memory processes are created, each of which will be assigned a subset of $NTask/NProcS$ tasks. Thus the original loop over model orbital space indices, w,x,y,z , is parallelized. Each processor then carries out the assembly of σ using the original serial algorithm. On finishing, the next available index ($wxyz$) is retrieved from tuple space and this procedure is continued until all model space generator combinations are processed, i.e., the index $(wxyz)=0$ is found. Finally the intermediate results are passed through tuple space to be combined to give the final result of the computation.

SMP architectures are supported in three ways: (i) using Linda, (ii) using shared memory, or (iii) a combination of both. In the first case, (i), one Linda worker is created on each PE. The communication overhead in this scheme is extremely small. Furthermore, using Linda even on SMPs takes full advantage of the load balancing mechanism described above. However, the static data (e.g., the CI vector C , integrals, etc.) must be replicated for each Linda worker.

The alternative (ii) is the usage of shared memory ($NProcS>1$). In this case all static data get replicated only once per SMP and are shared by all PEs. All result vectors (one per PE) are summed before the result of this worker is passed back to tuple space, where it is subsequently retrieved by the master process. This method has the advantage that all static data exist only once in working memory on an SMP node. For example, on an SMP with $NProcS=2$ the memory requirements are for three CI vectors; i.e., one result vector for each PE plus one shared CI vector (C). This is particularly useful if available working memory is limited and/or if the size of the CI vector is very large. A further advantage of this option is a reduction of communication by a factor of $NProcS$, since all static data and also the combined results have to be passed only once per $NProcS$ workers. The price to be paid in this case is that there is no load balancing between PEs on an SMP. The tasks defined by the $NProcS$ indices are of very similar length, but will be slightly different. In practice, this means that efficiency will decrease if $NProcS$ becomes very large.

The third option is any combination of the previous two. In this case the number of PEs using shared memory, $NProcS$, may be chosen to comprise any available number of PEs on each node, i.e., on a 4way SMP node $NProcS$ can assume the values 1, 2, 3 or 4. Case (iii) would then correspond to two Linda workers running on that node, each of them comprising $NProcS=2$ shared memory processes. Taking advantage of this option the mode of parallelism may be tailored to the available architecture.

The approach described above ensures great flexibility in

TABLE VIII. CASSCF and CAS-PT2 (Ref. 19) results for indole. For the source of experimental results see the references in (Ref. 19).

Geom.	State	CASSCF(10,9)/6-31G*		CAS-PT2/ANO	Experiment
		E [hartree]	E_{rel} [eV]	E_{rel} [eV]	E_{rel} [eV]
FC	S2 (1L_a)	-361.3371	6.20 (vert.)	4.73 (vert.)	4.77
FC	S1 (1L_b)	-361.3810	5.00 (vert.)	4.43 (vert.)	4.37
FC	S0	-361.5648	0.00	0.00	0.00
1L_b min.	S1	-361.3940	4.65 (0-0)	4.35 (0-0)	4.37
1L_a min.	S1	-	-	4.66 (0-0)	4.54
	S2	-361.3506	5.83 (0-0)	-	-

choosing the number of processors and leads, due to the relatively high number of tasks, to an effective load balancing, provided the number of PEs is small compared to the number of tasks. The intermediate assembly of a sublist of N_{Task} tasks ensures the optimum granularity, through dynamical adaptation of N_{Task} . The load balancing scheme implemented also allows for the fact that, in many environments, the CPU-time on different nodes of a parallel machine may be shared among a number of running programs and thus automatically uses the resources as they become available.

V. EXAMPLE APPLICATION

The RASSCF program developed and implemented as part of this work, has so far been applied in a small number of projects, including the calculation of the first two excited states of indole. We will in this section denote by $RAS(N, M_I + M_{II} + M_{III})[MxHole, MxElec]$ a RASSCF calculation with N electrons in an active space of M_I RAS1 orbitals, M_{II} RAS2 orbitals and M_{III} RAS3 orbitals. $MxHole$ and $MxElec$ signify the maximum number of particles excited out of the RAS1 space, and into the RAS3 orbital space, respectively.

Indole has two low singlet excited states, 1L_b (covalent) and 1L_a (ionic), which are separated by a small energy gap of approximately 0.4 eV. A CASSCF(10,9)/6-31G* calculation overestimates this gap (1.20 eV), mainly because the energy of the ionic 1L_a state is overestimated. CAS-PT2/ANO calculations by Serrano and Roos¹⁹ reproduce the experimental value. The goal of this study is to give a balanced description

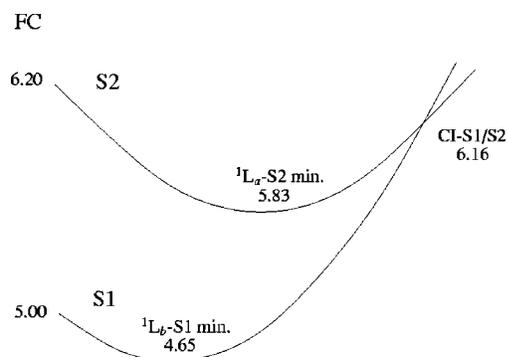


FIG. 9. Potential energy surfaces (S1,S2) of indole, calculated at the CASSCF level [CAS(10,9)/6-31G*].

of the excited states that includes a good approximation to the energy gap of 0.4 eV between the S1 and S2 excited states.

In Table VIII and Fig. 9 we illustrate the results of the CASSCF computation. A comparison with experimental results in Table VIII clearly shows that the results of the CASSCF computation are not only quantitatively, but also qualitatively, incorrect. According to the experimental data, the two states of indole (1L_a and 1L_b) have similar 0-0 transitions (emissions from the minimum for the state to the ground-state minimum). Therefore, the potential energy surface probably has two different minima on the surface of S1: one for each state. Presumably there is a conical intersection (surface crossing) between S1 and S2, because the 1L_a state is S2 at the Franck-Condon geometry. At the CASSCF(10,9)/6-31G* level, the surface is wrong: the 1L_a minimum lies on S2.

A common technique used in order to enhance qualitatively the description of electronic states is to “double” the active space of π -orbitals in a CASSCF computation. The corresponding CASSCF(10,18) in the case of indole is, however, outside the scope of practical computation. The RASSCF treatment of this enlarged active space on the other hand is absolutely practical. A second possibility for qualitatively improving the wavefunction is to include σ -orbitals into the set of active orbitals.

Preliminary computations using the 3-21G basis set were performed in order to identify the most suitable RAS active space definition. In a first test the active space was doubled by including the 9 2p orbitals of A'' symmetry (a π -system used in the CASSCF computation) and the 9 3p orbitals of A'' symmetry. Ten of these orbitals were placed in the RAS3 subspace, and the 3 nearly doubly occupied orbitals from the CASSCF computation in the RAS1 subspace. This corresponds to a CAS(4,5) reference space in RAS2. Two computations were performed, one with $MxHole=MxElec=2$ and one with $MxHole=MxElec=1$. In a second test the active

TABLE IX. Results of preliminary RAS computations of the first two singlet excited states of indole. The 3-21G basis set was used for these computations.

N	M_I	M_{II}	M_{III}	$MxHole$	$MxElec$	ΔE_{S1-S2}
10	3	5	10	1	1	0.90 eV
10	3	5	10	2	2	1.09 eV
30	13	5	20	1	1	0.78 eV

TABLE X. Orbital occupancies for RAS(30,13+5+20)[1,1] and RAS(20,8+5+11)[1,1] computations at the Franck–Condon geometry (6-31G* basis set). Orbitals that have been excluded in the latter RAS computation are shown in italics.

	Sym.	RAS(30,13+5+20)[1,1]	Sym.	RAS(20,8+5+11)[1,1]
RAS1	10A'	1.9986, 1.9984, 1.9983, 1.9979, 1.9978 1.9960, 1.9957, 1.9952, 1.9949, 1.9932	5A'	1.9979, 1.9960, 1.9958, 1.9955, 1.9951
	3A''	1.9866, 1.9842, 1.9731	3A''	1.9847, 1.9808, 1.9716
RAS2	5A''	1.8567, 1.0755, 0.9368, 0.1057, 0.0509	5A''	1.8377, 1.0966, 0.9312, 0.1136, 0.0535
RAS3	1A''	0.0210	1A''	0.0233
	10A'	0.0062, 0.0039, 0.0038, 0.0037, 0.0036 <i>0.0036, 0.0033, 0.0024, 0.0019, 0.0016</i>	5A'	0.0059, 0.0038, 0.0038, 0.0031, 0.0031
	9A''	0.0025, 0.0013, 0.0011, 0.0011, 0.0010 <i>0.0010, 0.0009, 0.0009, 0.0001</i>	5A''	0.0028, 0.0013, 0.0013, 0.0013, 0.0011

orbital space from the first test was augmented by 10 σ -orbitals and σ^* -orbitals corresponding to the 10 CC and CN σ -bonds. The RAS1 subspace now contained 13 orbitals, RAS2 5 orbitals and RAS3 20 orbitals, and excitations out of RAS1 and into RAS3 were limited to 1 particle each. Again the RAS2 subspace translates into a CAS(4,5) reference space. In Table IX we summarize the computations and their results. It is clear from these preliminary results that inclusion of σ -orbitals and single excitations relative to the reference space are particularly important.

The strategy from now on was (a) to increase the basis to 6-31G*, (b) to reduce the RAS space by eliminating the MOs with occupancies closest to 2.00 and 0.00, and vary the RAS2 subspace, and (c) to improve the energies with the 6-311+G* basis set. Table X shows the orbital occupancies resulting from step (a). A number of orbitals that are virtually doubly occupied or unoccupied are excluded from the active space and the orbital occupancies from the resulting RAS(20,8+5+11)[1,1] are also shown in Table X. The excitation energies obtained from these calculations are shown in Table XI. In general, all calculated energies for S1 are 0.7–1.4 eV higher than the experimental value. However, the energy gap between S1 and S2 is much more accurate than the CASSCF value of 1.20 eV (cf. Table VIII). The reduction of the active space has only a small effect, while the use of the larger (6-311+G*) basis set improves the results substantially.

TABLE XI. Results of RASSCF computations of indole, at the FC point. The experimental values are 4.77 eV (S1), 4.37 eV (S2), 0.40 eV (ΔE_{S2-S1}).

Basis set	($N, M_I + M_{II} + M_{III}$)	RAS2	E_{rel} (eV)	ΔE_{S2-S1} (eV)
6-31G*	(30,13+5+20)	(4,5)	S2: 6.41 S1: 5.74	0.67
6-31G*	(20,8+5+11)	(4,5)	S2: 6.01 S1: 5.27	0.74
6-31G*	(20,7+5+12)	(6,5)	S2: 6.74 S1: 6.06	0.68
6-31G*	(20,7+6+11)	(6,6)	S2: 5.98 S1: 5.15	0.83
6-311+G*	(20,7+6+11)	(6,6)	S2: 5.73 S1: 5.08	0.65

Figure 10 shows the potential energy surface as computed at the RAS(20,7+6+11)[1,1]/6-31G* level. The results in the figure are qualitatively correct, although the 0–0 energy for the 1L_a state is off by more than 1 eV. In contrast to the vertical excitations, the potential energy surface corrected with the 6-311+G* basis is worse than the ones with 6-31G*. The reason for this is probably that the points were optimized with the latter basis.

The result of this first application of our RASSCF program is encouraging. The improvements of the obtained excitation energies compared to the CASSCF(10,9)/6-31G* is considerable, but the crucial result is the qualitatively correct potential energy surface. The computational cost increases only moderately, with the number of CSFs increasing from 8001 for the CASSCF(10,9) computation, to 53735 for the RASSCF(20,7+6+11)[1,1]. On the other hand, exploratory calculations have shown that in order to complete the study of the potential energy surface and optimize the transition structure between the two minima on the S1 surface, extension of the RAS2 subspace is necessary.

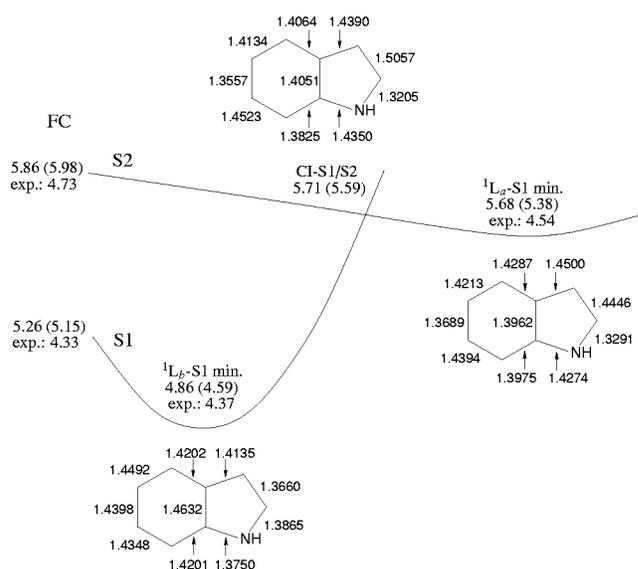


FIG. 10. Potential energy surfaces calculated at the RASSCF level (energies in eV, bond distances in Å). The geometry optimizations were carried out at the RAS(20,7+6+11)[1,1]/6-31G* level (the number in brackets) and the energetics recalculated with a larger basis set, at the RAS(20,7+6+11) × [1,1]/6-311+G* level.

VI. SUMMARY

In this paper we have developed an efficient method for updating the CI vector in a RASSCF computation within the iterative Davidson/Lanczos methods. The string based algorithm arises due to two new concepts: (i) the RAS model space concept ensures a very efficient identification of blocks of contributions to the σ -vector, and we have made use of the reduced list algorithm previously introduced in Ref. 9 for efficient CASSCF computations. The RAS model space allows the representation of a restricted space problem in an unrestricted manner, thus avoiding the complications inherent in the RASSCF method. (ii) Propagation rules are used to efficiently construct the string addresses (as opposed to the strings) in a predefined order. Propagation rules are a simple, but very efficient algebraic tool avoiding table-lookups and the construction of contributing strings themselves.

The resulting method is memory efficient, since it utilizes compressed storage of precomputed model excitation lists in the main memory. The construction of the full excitation strings is fully avoided. The algorithm developed follows an integral driven approach. The resulting outer nested loops over model orbital indices w, x, y, z lends itself to efficient parallelization. Every model space index quadruple w, x, y, z represents a list of integrals $(ij|kl)$, i.e., we have an

integral block driven approach. Clusters of index quadruples with variable cluster size ensure adjustable granularity of the parallelism, thus avoiding scaling problems. Communication overhead (if run in parallel) is minimal through implicit task definition by only one integer. This technique also leads to a natural load balancing.

The algorithm is implemented in the current development version of the Gaussian package of programs.²⁰ The implementation comprises the option of parallel execution. Running in parallel may be done using distributed memory (Linda) or shared memory, or a combination of these. As well as the most general case of Slater determinants, we also implemented the straightforward simplifications for singlets and triplets using Hartree Waller functions.

A test example application has been performed with the current version of the program. The results show that the RASSCF method is a useful tool allowing us to approach problems that are currently not tractable with the CASSCF method.

ACKNOWLEDGMENTS

This work was supported in part by Gaussian Inc., PA, USA. L. B. is funded by the Ramón y Cajal program of the Ministerio de Ciencia y Tecnología (Spain).

APPENDIX: ALGORITHM

1. The 2e excitation ($\alpha\alpha$ part) contribution to the σ -vector

```
-At program initialization, precompute all reduced model string lists
-Do (parallel) loop over model space index quadruples  $w, x, y, z$ , [cf. Eq. (36)]
-Do loop over reduced model  $\alpha$ -string list  $\mathcal{L}_{N^m-2}^{M^m-4}$ 
-Insert bits  $b_w, b_x, b_y, b_z$  into reduced model string and form model excitation string
( $\rightarrow K_\alpha^m, L_\alpha^m$ )
-If  $K_\alpha^m, L_\alpha^m$  are invalid, jump to loop end [this is to avoid double contributions; cf. Sec. IIC]
 $\rightarrow \text{sgn}_\alpha^m, \text{Cat}(i_h, i_e), \mathcal{A}\{K_\alpha^{\text{RAS2}}\}, \text{Cat}(i'_h, i'_e), \mathcal{A}\{L_\alpha^{\text{RAS2}}\}, \mathcal{P}(\text{RAS1}), \mathcal{P}(\text{RAS3})$ 
-Assemble integral list  $\{(ij|kl)\}$ , using the propagation rules  $\mathcal{P}(\text{RAS1})$  and  $\mathcal{P}(\text{RAS3})$ 
-Apply  $\mathcal{P}(\text{RAS1})$  to assemble RAS1 substring excitation list,  $\mathcal{E}_\alpha^{\text{RAS1}}$ ,
( $\rightarrow \mathcal{A}\{K_\alpha^{\text{RAS1}}\}, \mathcal{A}\{L_\alpha^{\text{RAS1}}\}, \text{sgn}_\alpha^{\text{RAS1}}$ )
-Apply  $\mathcal{P}(\text{RAS3})$  to assemble RAS3 substring excitation list,  $\mathcal{E}_\alpha^{\text{RAS3}}$ ,
( $\rightarrow \mathcal{A}\{K_\alpha^{\text{RAS3}}\}, \mathcal{A}\{L_\alpha^{\text{RAS3}}\}, \text{sgn}_\alpha^{\text{RAS3}}$ )
-Do loop over all entries in  $\mathcal{E}_\alpha^{\text{RAS1}}$ 
-Do loop over all entries in  $\mathcal{E}_\alpha^{\text{RAS3}}$ 
 $\rightarrow \mathcal{A}\{K_\alpha\}, \mathcal{A}\{L_\alpha\}, (ij|kl), \text{sgn}_\alpha = \text{sgn}_\alpha^m \text{sgn}_\alpha^{\text{RAS1}} \text{sgn}_\alpha^{\text{RAS3}}$ 
-Do loop over all allowed  $\text{Cat}(i''_h, i''_e)$  [allowed if  $\text{Max}(i_h, i'_h) + i''_h \leq \text{MxHole}$  and  $\text{Max}(i_e, i'_e) + i''_e \leq \text{MxElec}$ ]
-Do loop over all  $K_\beta(\text{Cat}(i''_h, i''_e))$ 
 $-\sigma_{K_\alpha K_\beta} := \sigma_{K_\alpha K_\beta} + \frac{1}{2} \text{sgn}_\alpha(ij|kl) C_{L_\alpha K_\beta}$ 
 $-\sigma_{L_\alpha K_\beta} := \sigma_{L_\alpha K_\beta} + \frac{1}{2} \text{sgn}_\alpha(ij|kl) C_{K_\alpha K_\beta}$ 
-End loops
```

2. The 2e excitation ($\alpha\beta$ part) contribution to the σ -vector

```
-Do (parallel) loop over model space index quadruples  $w, x, y, z$ , [cf. Eq. (36)]
-Integral list  $\{(ij|kl)\}$  is already known from  $\alpha\alpha$  part.
```

-Do loop over reduced model α -string list $\mathcal{L}_{N_\alpha^m-1}^{M^m-2}$

-Insert bits b_w, b_x into reduced model string and form model excitation string ($\rightarrow K_\alpha^m, L_\alpha^m$)

-If K_α^m, L_α^m are invalid, jump to loop end [this is to avoid double contributions; cf. Sec. IIC]

$\rightarrow \text{sgn}_\alpha^m, \text{Cat}(i_h, i_e), \mathcal{A}\{K_\alpha^{\text{RAS2}}\}, \text{Cat}(i'_h, i'_e), \mathcal{A}\{L_\alpha^{\text{RAS2}}\}, \mathcal{P}(\text{RAS1}), \mathcal{P}(\text{RAS3})$

-Apply $\mathcal{P}(\text{RAS1})$ to assemble RAS1 substring excitation list, $\mathcal{E}_\alpha^{\text{RAS1}}$

($\rightarrow \mathcal{A}\{K_\alpha^{\text{RAS1}}\}, \mathcal{A}\{L_\alpha^{\text{RAS1}}\}, \text{sgn}_\alpha^{\text{RAS1}}$)

-Apply $\mathcal{P}(\text{RAS3})$ to assemble RAS3 substring excitation list, $\mathcal{E}_\alpha^{\text{RAS3}}$

($\rightarrow \mathcal{A}\{K_\alpha^{\text{RAS3}}\}, \mathcal{A}\{L_\alpha^{\text{RAS3}}\}, \text{sgn}_\alpha^{\text{RAS3}}$)

-Do loop over reduced model β -string list $\mathcal{L}_{N_\beta^m-1}^{M^m-2}$

-Insert bits b_y, b_z into reduced model string and form model excitation string ($\rightarrow K_\beta^m, L_\beta^m$)

-If K_β^m, L_β^m are invalid, jump to loop end [this is to avoid double contributions; cf. Sec. IIC]

$\rightarrow \text{sgn}_\beta^m, \text{Cat}(i''_h, i''_e), \mathcal{A}\{K_\beta^{\text{RAS2}}\}, \text{Cat}(i'''_h, i'''_e), \mathcal{A}\{L_\beta^{\text{RAS2}}\}$

-If $[(i_h + i''_h) \leq \text{MxHole} \text{ and } (i'_h + i'''_h) \leq \text{MxHole} \text{ and } (i_e + i''_e) \leq \text{MxElec} \text{ and } (i'_e + i'''_e) \leq \text{MxElec}]$ then

-Apply $\mathcal{P}(\text{RAS1})$ to assemble RAS1 substring excitation list, $\mathcal{E}_\beta^{\text{RAS1}}$,

($\rightarrow \mathcal{A}\{K_\beta^{\text{RAS1}}\}, \mathcal{A}\{L_\beta^{\text{RAS1}}\}, \text{sgn}_\beta^{\text{RAS1}}$)

-Apply $\mathcal{P}(\text{RAS3})$ to assemble RAS3 substring excitation list, $\mathcal{E}_\beta^{\text{RAS3}}$,

($\rightarrow \mathcal{A}\{K_\beta^{\text{RAS3}}\}, \mathcal{A}\{L_\beta^{\text{RAS3}}\}, \text{sgn}_\beta^{\text{RAS3}}$)

-Do loop over all entries in $\mathcal{E}_\beta^{\text{RAS1}}$

-Do loop over all entries in $\mathcal{E}_\beta^{\text{RAS3}}$

$\rightarrow \mathcal{A}\{K_\alpha\}, \mathcal{A}\{L_\alpha\}, \text{sgn}_\alpha = \text{sgn}_\alpha^m \text{sgn}_\alpha^{\text{RAS1}} \text{sgn}_\alpha^{\text{RAS3}}$

-Do loop over all entries in $\mathcal{E}_\beta^{\text{RAS1}}$

-Do loop over all entries in $\mathcal{E}_\beta^{\text{RAS3}}$

$\rightarrow \mathcal{A}\{K_\beta\}, \mathcal{A}\{L_\beta\}, (ij|kl), \text{sgn}_\beta = \text{sgn}_\beta^m \text{sgn}_\beta^{\text{RAS1}} \text{sgn}_\beta^{\text{RAS3}}$

$-\sigma_{K_\alpha K_\beta} := \sigma_{K_\alpha K_\beta} + 1/2 \text{sgn}_\alpha \text{sgn}_\beta (ij|kl) C_{L_\alpha L_\beta}$

$-\sigma_{L_\alpha L_\beta} := \sigma_{L_\alpha L_\beta} + 1/2 \text{sgn}_\alpha \text{sgn}_\beta (ij|kl) C_{K_\alpha K_\beta}$

-End loops

¹J. Olsen, B. O. Roos, P. Jørgensen, and H. J. A. Jensen, *J. Chem. Phys.* **89**, 2185 (1988).

²K. Andersson, P. Borowski, P. W. Fowler, P. A. Malmqvist, B. O. Roos, and A. J. Sadlej, *Chem. Phys. Lett.* **190**, 367 (1992).

³M. P. Fülscher, M. Jaszunski, B. O. Roos, and W. P. Kraemer, *J. Chem. Phys.* **96**, 504 (1992).

⁴M. Merchan, E. Orti, and B. O. Roos, *Chem. Phys. Lett.* **221**, 136 (1994).

⁵K. Andersson, P. A. Malmqvist, and B. O. Roos, *J. Chem. Phys.* **96**, 1218 (1992).

⁶B. O. Roos, *Acc. Chem. Res.* **32**, 137 (1999).

⁷P. E. M. Siegbahn, *J. Chem. Phys.* **72**, 1647 (1980).

⁸V. R. Saunders and J. H. van Lenthe, *Mol. Phys.* **48**, 923 (1983).

⁹M. Klene, M. A. Robb, M. J. Frisch, and P. Celani, *J. Chem. Phys.* **113**, 5653 (2000).

¹⁰C. Lanczos, *J. Res. Natl. Bur. Stand.* **45**, 255 (1950).

¹¹E. R. Davidson, *J. Comput. Phys.* **17**, 87 (1975).

¹²P. J. Knowles and N. C. Handy, *Chem. Phys. Lett.* **111**, 315 (1984).

¹³W. Duch, *Chem. Phys. Lett.* **124**, 442 (1986).

¹⁴P. M. Kozłowski and P. Pulay, *Theor. Chem. Acc.* **100**, 12 (1998).

¹⁵S. Zarrabian, C. R. Sarma, and J. Paldus, *Chem. Phys. Lett.* **155**, 183 (1989).

¹⁶N. Bajpai, C. R. Sarma, and J. Paldus, *J. Mol. Struct.: THEOCHEM* **73**, 67 (1991).

¹⁷G. L. Bendazzoli and S. Evangelisti, *J. Chem. Phys.* **98**, 3141 (1993).

¹⁸*Linda, User's Guide & Reference Manual, Version 3.0*, Scientific Computing Associates, 1995.

¹⁹L. Serrano Andrés and B. O. Roos, *J. Am. Chem. Soc.* **118**, 185 (1996).

²⁰M. J. Frisch, G. W. Trucks, H. B. Schlegel *et al.*, GAUSSIAN 99, *Development Version (Revision B.1)*, 1998, Gaussian, Inc., Pittsburgh, PA.