

## SPINS: A collection of algorithms for symbolic generation and transformation of many-electron spin eigenfunctions

Peter B. Karadakov<sup>1</sup>, Joseph Gerratt<sup>1</sup>, David L. Cooper<sup>2</sup>, Mario Raimondi<sup>3</sup>

<sup>1</sup> School of Chemistry, University of Bristol, Cantock's Close, Bristol BS8 1TS, United Kingdom

<sup>2</sup> Department of Chemistry, University of Liverpool, P.O. Box 147, Liverpool L69 3BX, United Kingdom

<sup>3</sup> Dipartimento di Chimica Fisica ed Elettrochimica, Università di Milano, Via Golgi 19, I-20133 Milano, Italy

Received March 9, 1994/Accepted July 9, 1994

**Summary.** SPINS represents a collection of algorithms intended to provide an efficient, robust and easy-to-use quantum-chemical toolbox capable of performing a wide range of operations on spin eigenfunctions in the Rumer, Kotani and Serber spin bases. It includes routines for symbolic generation of the Rumer spin eigenfunctions as linear combinations of elementary spin products, for computing all transformation matrices relating the Rumer, Kotani and Serber spin bases and for calculation of the matrices of the irreducible representations of the symmetric group carried by the Rumer, Kotani and Serber spin eigenfunctions, as well as facilities for interpreting general spin-coupling patterns such as those used in spin-coupled theory. The resulting codes, written in Fortran-77 and available on the Internet (from P.B.Karadakov@Bristol.AC.UK or DLC@Liverpool.AC.UK) are so compact and efficient that they even run on IBM PC-compatible personal computers.

**Key words:** Spin eigenfunctions – Rumer spin basis – Kotani spin basis – Serber spin basis – Spin-coupled theory

### 1 Introduction

Efficient tools for generating and manipulating many-electron spin eigenfunctions are of primary importance to quantum chemistry: The design of the trial wavefunctions (CSFs) such as, for example, valence bond (VB) [1], spin-coupled (SC) [2], generalized VB (GVB) [3], configuration interaction (CI) [4], multiconfiguration self-consistent field (MC SCF) [5], complete-active-space SCF (CAS SCF) [6], requires spin eigenfunctions involving all, or a selected (active) group of the electrons in the system under investigation.

The literature related to the construction of many-electron spin eigenfunctions is extensive. A comprehensive account of its various aspects and a list of further references can be found in the monograph by Pauncz [7].

It is usual to denote  $N$ -electron spin eigenfunctions by  $\Theta_{SM,k}^N$ :

$$\hat{\mathcal{S}}^2 \Theta_{SM,k}^N = S(S+1) \Theta_{SM,k}^N \quad (1)$$

$$\hat{\mathcal{S}}_z \Theta_{SM,k}^N = M \Theta_{SM,k}^N \quad (2)$$

where  $\hat{\mathcal{P}}^2$  and  $\hat{\mathcal{P}}_z$  are the operators for the total spin of the system and its z-projection, respectively (in units of  $\hbar$ ). The index  $k$  distinguishes between different spin functions corresponding to the same pair of  $S, M$  values. Their total number is given by

$$f_S^N = \frac{(2S + 1)N!}{(\frac{1}{2}N + S + 1)!(\frac{1}{2}N - S)!}. \quad (3)$$

The linearly independent set  $\{\Theta_{SM;k}^N | k = 1, 2, \dots, f_S^N\}$  forms a complete spin basis, i.e. any other  $N$ -electron spin eigenfunction characterized by the same  $S$  and  $M$  values can be expanded as

$$\Theta_{SM}^N = \sum_{k=1}^{f_S^N} C_{Sk} \Theta_{SM;k}^N = \Theta_{SM}^N \mathbf{C}_S, \quad (4)$$

where  $\Theta_{SM}^N$  and  $\mathbf{C}_S$  denote a row vector of all spin functions and a column vector of all spin-coupling coefficients, respectively.

It is sufficient to have the spin eigenfunctions  $\Theta_{SM;k}^N$  for only one value of  $M$ : often a convenient choice is the *principal* case  $M = S$ . Spin eigenfunctions, corresponding to other values of  $M$  can be obtained by application of the step-up and step-down operators  $\hat{\mathcal{P}}_+ = \hat{\mathcal{P}}_x + i\hat{\mathcal{P}}_y$  and  $\hat{\mathcal{P}}_- = \hat{\mathcal{P}}_x - i\hat{\mathcal{P}}_y$ :

$$\begin{aligned} \hat{\mathcal{P}}_+ \Theta_{SM}^N &= [(S - M)(S + M + 1)]^{1/2} \Theta_{S, M+1}^N, \\ \hat{\mathcal{P}}_- \Theta_{SM}^N &= [(S + M)(S - M + 1)]^{1/2} \Theta_{S, M-1}^N. \end{aligned} \quad (5)$$

The set  $\{\Theta_{SM}^N\}$  provides a basis for an  $f_S^N$ -dimensional irreducible representation of the symmetric (or permutation) group of  $N!$  permutations  $\mathcal{S}_N$ : If  $\mathcal{P}$  is any permutation acting on the electron spin coordinates

$$\mathcal{P} = \begin{pmatrix} 1 & 2 & \dots & N \\ p_1 & p_2 & \dots & p_N \end{pmatrix}, \quad (6)$$

then

$$\mathcal{P} \Theta_{SM;k}^N = \sum_{l=1}^{f_S^N} \Theta_{SM;l}^N [V_S^N(\mathcal{P})]_{lk} = [\Theta_{SM}^N \mathbf{V}_S^N(\mathcal{P})]_k. \quad (7)$$

Obviously, knowledge of the  $f_S^N$  spin functions  $\Theta_{SM;k}^N$  is sufficient to generate all  $N!$  representation matrices  $\mathbf{V}_S^N(\mathcal{P})$ . Conversely, it is possible to obtain a complete spin basis for an irreducible representation of  $\mathcal{S}_N$  (using group-theory projection operators), if all representation matrices are available.

Quantum-chemical approaches, which represent the trial wavefunction in terms of Slater determinants, make use mainly of the expansions of the many-electron spin eigenfunctions  $\Theta_{SM;k}^N$  as linear combinations of products of one-electron  $\alpha$  and  $\beta$  spin functions. Methods operating in terms of CSFs usually do not require the spin functions explicitly: the corresponding expressions for overlaps or matrix elements of many-electron operators between CSFs involve all or some of the representation matrices  $[\mathbf{V}_S^N(\mathcal{P})]$ , or those of the related *dual* representation  $\mathbf{U}_S^N(\mathcal{P}) = \varepsilon_{\mathcal{P}} \tilde{\mathbf{V}}_S^N(\mathcal{P}^{-1})$ , where  $\varepsilon_{\mathcal{P}} = \pm 1$ , depending on the parity of  $\mathcal{P}$  and  $\tilde{\mathbf{V}}_S^N(\mathcal{P}^{-1})$  denotes the transpose of  $\mathbf{V}_S^N(\mathcal{P}^{-1})$  within a particular spin basis.

The most commonly used complete sets of spin functions are the Rumer [8], Kotani et al. [9] and Serber [10, 11] spin bases. Further in this text we shall denote them as  $\{\mathbf{R}\Theta_{SM}^N\}$ ,  $\{\mathbf{K}\Theta_{SM}^N\}$  and  $\{\mathbf{S}\Theta_{SM}^N\}$ , respectively. The methods for constructing these spin bases are all of the *synthetic* type, i.e. the full set of  $N$ -electron spin

functions is built up consecutively from smaller units (one and two-electron spin functions). As an alternative, spin eigenfunctions can be obtained by projecting out the components with the required values of  $S$  and  $M$  from an arbitrary spin function. Such approaches are known as *analytic*; the most important of these is due to Löwdin [12, 13].

Spin-coupled theory [14] makes a particularly creative use of different spin bases. It employs the most general wavefunction based on a single orbital product:

$$\Psi_{SM} = \hat{\mathcal{A}}(\psi_1 \psi_2 \dots \psi_N \Theta_{SM}^N), \quad (8)$$

where  $\hat{\mathcal{A}}$  stands for the antisymmetrizer,  $\psi_1, \psi_2, \dots, \psi_N$  are singly-occupied nonorthogonal orbitals (approximated by expansions in a suitable basis of atom-centered functions) and  $\Theta_{SM}^N$  is an  $N$ -electron spin eigenfunction in the form of Eq. (4). The optimal  $\psi_1, \psi_2, \dots, \psi_N$  and  $\Theta_{SM}^N$  are determined variationally [15–17]. The interpretation of the spin-coupling pattern within the SC wavefunction (8) [expressed by the values of the spin-coupling coefficients  $C_{Sk}$ , see Eq. (4)] can be often considerably facilitated and enriched by altering the spin basis [18]: generally, different spin bases emphasize different features of the optimal spin arrangement. In order to achieve this, it is necessary to have fast and convenient procedures for transforming between the representations of  $\Theta_{SM}^N$  in the most important spin bases, involving at least those due to Rumer, Kotani and Serber:

$$\Theta_{SM}^N = {}^K\Theta_{SM}^N {}^K\mathbf{C}_S = {}^R\Theta_{SM}^N {}^R\mathbf{C}_S = {}^S\Theta_{SM}^N {}^S\mathbf{C}_S. \quad (9)$$

In the present article we describe SPINS: A collection of algorithms intended to provide a fast, versatile and easy-to-use quantum-chemical toolbox capable of performing a wide range of operations involving spin eigenfunctions from the Rumer, Kotani and Serber spin bases. SPINS includes routines for:

- (a) generation of the expansions of the complete set of Rumer spin functions  $\{{}^R\Theta_{SM}^N\}$  for a system of  $N$  electrons with spin  $S$  (if  $N$  is even,  $S$  can take integer values between 0 and  $N/2$ , and if  $N$  is odd – half-integer values between  $1/2$  and  $N/2$ ) in terms of  $N$ -electron  $\alpha$  and  $\beta$  spin products;
- (b) calculation of the transformation matrices  ${}^{RK}\mathbf{T}_S^N$  and  ${}^{KS}\mathbf{T}_S^N$  relating the Rumer spin basis to the Kotani spin basis  $\{{}^K\Theta_{SM}^N\}$  and  $\{{}^K\Theta_{SM}^N\}$  to the Serber spin basis, respectively;

$${}^R\Theta_{SM}^N {}^{RK}\mathbf{T}_S^N = {}^K\Theta_{SM}^N, \quad (10)$$

$${}^K\Theta_{SM}^N {}^{KS}\mathbf{T}_S^N = {}^S\Theta_{SM}^N; \quad (11)$$

- (c) calculation of the matrices of the irreducible representations of  $\mathcal{S}_N$  formed by the Rumer spin basis [ ${}^R\mathbf{V}_S^N(\mathcal{P})$ , cf. Eq. (7)].

Given the results of these routines, it is straightforward to obtain additional quantities, such as the Kotani-to-Rumer and Serber-to-Kotani transformation matrices:

$${}^{KR}\mathbf{T}_S^N = ({}^{RK}\mathbf{T}_S^N)^{-1}, \quad (12)$$

$${}^{SK}\mathbf{T}_S^N = {}^{KS}\tilde{\mathbf{T}}_S^N \quad (13)$$

(the Rumer spin basis is nonorthogonal, while both the Kotani and Serber spin bases are orthogonal), as well as the matrices of the irreducible representations of  $\mathcal{S}_N$  in the Kotani and Serber spin bases:

$$\mathbf{K}\mathbf{V}_S^N(\mathcal{P}) = \mathbf{K}^R \mathbf{T}_S^N \mathbf{R} \mathbf{V}_S^N(\mathcal{P}) \mathbf{R}^K \mathbf{T}_S^N, \quad (14)$$

$$\mathbf{S}\mathbf{V}_S^N(\mathcal{P}) = \mathbf{S}^K \mathbf{T}_S^N \mathbf{K} \mathbf{V}_S^N(\mathcal{P}) \mathbf{K}^S \mathbf{T}_S^N. \quad (15)$$

The algorithms making up SPINS are based mainly on existing theoretical material. The most important new element is in the compact non-trivial yet transparent realization of those ideas from the theory of many-electron spin functions, which lend themselves most readily to algorithmization for use in computer programs. This, in combination with the bit-wise packing of spin products we introduce, leads to highly efficient and reliable codes.

## 2 Construction of the Rumer spin basis

The Rumer spin basis represents a set of  $f_S^N$  linearly independent spin functions, in which  $N - 2S$  electrons form singlet pairs, and the remaining  $2S$  electrons are assigned spins  $\alpha$  ( $M = S$ ):

$$\begin{aligned} \mathbf{R}\Theta_{SM;k}^N &= 2^{-1/2} [\alpha(\mu_1)\beta(\mu_2) - \alpha(\mu_2)\beta(\mu_1)] \dots 2^{-1/2} [\alpha(\mu_{N-2S-1})\beta(\mu_{N-2S}) \\ &\quad - \alpha(\mu_{N-2S})\beta(\mu_{N-2S-1})] \alpha(\mu_{N-2S+1}) \dots \alpha(\mu_N). \end{aligned} \quad (16)$$

Convenient extended labels for these spin functions are provided by the lists of singlet pairs within each one of them:

$$k \equiv (\mu_1 - \mu_2, \mu_3 - \mu_4, \dots, \mu_{N-2S-1} - \mu_{N-2S}). \quad (17)$$

The number of spin eigenfunctions which can be formed according to Eq. (16) ( $v_S^N$ ) is much larger than  $f_S^N$ :

$$v_S^N = \frac{N!}{2^{N_p} (N - 2N_p)! N_p!} \quad (18)$$

where  $N_p = \frac{1}{2}N - S$  stands for the number of singlet pairs within each Rumer spin function (16). However, only  $f_S^N$  of these Rumer functions are linearly independent. One way of obtaining  $f_S^N$  linearly independent  $\mathbf{R}\Theta_{SM;k}^N$  is to apply the diagrammatic technique suggested by Rumer [8] and extended by Simonetta et al. [19] to nonsinglet states. Another possibility, implemented in SPINS, is to use the *leading term* method [19] which is equivalent to the procedures for constructing bonded functions [4, 20–22].

The first step in this approach is to construct  $f_S^N$  *leading terms*  $\theta_{M;k,1}^N$  ( $M = S$ ). Each of these represents an ordered product of  $\frac{1}{2}N + S$   $\alpha$  and  $\frac{1}{2}N - S$   $\beta$  spin functions.  $\theta_{M;1,1}^N$  always involves  $\frac{1}{2}N - S$   $\alpha\beta$  pairs, followed by  $2S$   $\alpha$  functions:

$$\begin{aligned} \Theta_{M;1,1}^N &= \alpha(1)\beta(2) \dots \alpha(N - 2S - 1)\beta(N - 2S)\alpha(N - 2S + 1) \dots \alpha(N) \\ &= \underbrace{\alpha\beta\alpha\beta \dots \alpha\beta}_{1/2N - S \text{ pairs}} \underbrace{\alpha\alpha \dots \alpha}_{2S}. \end{aligned} \quad (19)$$

The  $i$ th leading term,  $\theta_{M;i,1}^N$  ( $i > 1$ ), is obtained from its predecessor,  $\theta_{M;i-1,1}^N$ , by shifting one place to the right the first  $\beta$  (reading from left to right) followed by an  $\alpha$ , and then bringing all  $\beta$  functions to the left of the shifted one to the positions they held in  $\theta_{M;1,1}^N$ .

In the next step, the leading terms are used to obtain a corresponding number of Rumer functions as follows: In each leading term, a left parenthesis is inserted before every  $\alpha$ , and a right parenthesis – after every  $\beta$  (reading from left to right), then a singlet pair is associated with each combination of  $\alpha$  and  $\beta$  functions enclosed in matching left and right parentheses.

As an example, below we list the leading terms and the resulting Rumer spin functions for  $N = 5$ ,  $S = \frac{1}{2}$  ( $f_{1/2}^5 = 5$ ):

$$\begin{aligned}\theta_{1/2;1,1}^5 &= \alpha\beta\alpha\beta\alpha \rightarrow {}^R\Theta_{1/2\ 1/2;1}^5 = (\alpha\beta)(\alpha\beta)(\alpha) \equiv (1 - 2, 3 - 4), \\ \theta_{1/2;2,1}^5 &= \alpha\alpha\beta\beta\alpha \rightarrow {}^R\Theta_{1/2\ 1/2;2}^5 = (\alpha(\alpha\beta))\beta(\alpha) \equiv (1 - 4, 2 - 3), \\ \theta_{1/2;3,1}^5 &= \alpha\beta\alpha\alpha\beta \rightarrow {}^R\Theta_{1/2\ 1/2;3}^5 = (\alpha\beta)(\alpha(\alpha\beta)) \equiv (1 - 2, 4 - 5), \\ \theta_{1/2;4,1}^5 &= \alpha\alpha\beta\alpha\beta \rightarrow {}^R\Theta_{1/2\ 1/2;4}^5 = (\alpha(\alpha\beta))(\alpha\beta) \equiv (2 - 3, 4 - 5), \\ \theta_{1/2;5,1}^5 &= \alpha\alpha\alpha\beta\beta \rightarrow {}^R\Theta_{1/2\ 1/2;5}^5 = (\alpha(\alpha(\alpha\beta))\beta) \equiv (2 - 5, 3 - 4).\end{aligned}\quad (20)$$

SPINS generates the leading terms by means of the following code fragment (part of subroutine *leadt*; this fragment, as well as the whole program, are written in Fortran-77):

```

      ifns = 1
30   i = index(lt(ifns)(1:n), 'ba')
      if (i.ne.0) then
          ifns = ifns + 1
          lt(ifns) = lt(ifns - 1)
          lt(ifns)(i:i + 1) = 'ab'
          nbeta = 0
          do 40 k = 1, i - 1
              if (lt(ifns)(k:k).eq. 'b') nbeta = nbeta + 1
40   continue
          if (nbeta.gt.0) then
              lt(ifns)(1:2*nbeta) = lt(1)(1:2*nbeta)
              do 50 j = 2*nbeta + 1, i - 1
50   lt(ifns)(j:j) = 'a'
          endif
          go to 30
      endif

```

(21)

The leading terms are recorded in the first  $f_S^N$  elements of the character array *lt*, as strings of  $N$  'a' and 'b' characters, representing  $\alpha$  and  $\beta$  functions, respectively. The

first leading term is expected to have been initialized according to Eq. (19). The number of  $\beta$  functions preceding the shifted  $\beta$  in the current leading term [ $lt(i)$ ] is accumulated in  $nbeta$ . In addition to producing all leading terms, code segment (21) calculates  $f_S^N$  (given by the final value of the counter  $ifns$ ).

The insertion of parentheses to form Rumer spin functions from the leading terms is achieved as follows (excerpt from subroutine *fpairs*):

```

do 30 l = 1, ifns
  k = 1
  kunp = 2 * npairs
  do 20 i = 1, n
    if (lt(l)(i:i).eq.'a') then
      leftp = 0
      do 10 j = i + 1, n
        if (lt(l)(j:j).eq.'a') then
          leftp = leftp + 1
        elseif (lt(l)(j:j).eq.'b') then
          if (leftp.gt.0) then
            leftp = leftp - 1
          else
            rf(k, l) = i
            rf(k + 1, l) = j
            k = k + 2
            go to 20
          endif
        endif
      continue
      kunp = kunp + 1
      rf(kunp, l) = i
    endif
  20 continue
  30 continue

```

(22)

The Rumer spin functions are stored in the two-dimensional integer array  $rf$ : the code shown above fills each of the first  $f_S^N$  ( $ifns$ ) columns of  $rf$  with the numbers  $\mu_1, \mu_2, \dots, \mu_N$ , defining the Rumer function  ${}^R\Theta_{SM;k}^N$  [see Eq. (16)]. Although it is sufficient to keep track only of the singlet pairs in each  ${}^R\Theta_{SM;k}^N$ , according to Eq. (17) (i.e. the ordered sequence  $\mu_1, \mu_2, \dots, \mu_{N-2S}$ ), we find it convenient to record the indices of the unpaired electrons as well, which facilitates a subsequent expansion of each Rumer function in terms of products of  $\alpha$  and  $\beta$  functions.  $npairs$  stands for the number of singlet pairs in every Rumer spin function,  $N_p$ , while  $k$  and  $knup$  index successive locations in the “paired” and “unpaired” parts of the current Rumer spin function. The counter  $leftp$  accumulates the number of unmatched left parentheses, starting from position  $i$  in the current leading term, and proceeding to the right. If  $leftp$  becomes equal to zero at position  $j$ , then  $i$  and  $j$  are singlet paired, otherwise  $i$  is unpaired.

The representation matrix  ${}^R V_S^N(\mathcal{P})$ , corresponding to a permutation  $\mathcal{P}$  can be obtained as [cf. Eq. (7)]

$${}^R V_S^N(\mathcal{P}) = (\langle {}^R \Theta_{SM}^N | {}^R \Theta_{SM}^N \rangle)^{-1} \langle {}^R \Theta_{SM}^N | \mathcal{P} {}^R \Theta_{SM}^N \rangle, \quad (23)$$

where  $\langle {}^R \Theta_{SM}^N | {}^R \Theta_{SM}^N \rangle$  and  $\langle {}^R \Theta_{SM}^N | \mathcal{P} {}^R \Theta_{SM}^N \rangle$  denote the  $f_S^N \times f_S^N$  overlap matrices with elements  $\langle {}^R \Theta_{SM;k}^N | {}^R \Theta_{SM;l}^N \rangle$  and  $\langle {}^R \Theta_{SM;k}^N | \mathcal{P} {}^R \Theta_{SM;l}^N \rangle$ , respectively. One way of evaluating overlap integrals between Rumer functions has been suggested by Cooper and McWeeny [22]. We have adopted a less elegant but, nonetheless, very efficient direct approach employing the expansion of each  ${}^R \Theta_{SM;k}^N$  in terms of  $2^{N_p}$  products of  $\alpha$  and  $\beta$  spin functions:

$$\begin{aligned} {}^R \Theta_{SM;k}^N &= 2^{-N_p/2} (1 - \mathcal{P}_{12})(1 - \mathcal{P}_{34}) \cdots (1 - \mathcal{P}_{2N_p-1, 2N_p}) \theta_{M;k,1}^N \\ &= 2^{-N_p/2} \sum_{r=1}^{2^{N_p}} (-1)^{\lambda_r} \theta_{M;k,r}^N, \end{aligned} \quad (24)$$

which entails evaluation of overlaps between spin products:  $\langle \theta_{M;k,r}^N | \theta_{M;k,s}^N \rangle$  and  $\langle \theta_{M;k,r}^N | \mathcal{P} \theta_{M;k,s}^N \rangle$ . Here  $\mathcal{P}_{\mu\nu}$  denotes the single transposition  $\begin{pmatrix} \mu & \nu \\ \nu & \mu \end{pmatrix}$  and  $\lambda_r$  stands for the number of single transpositions transforming the leading term  $\theta_{M;k,1}^N$  into spin product  $\theta_{M;k,r}^N$ . The calculation of the overlaps  $\langle \theta_{M;k,r}^N | \theta_{M;k,s}^N \rangle$  and  $\langle \theta_{M;k,r}^N | \mathcal{P} \theta_{M;k,s}^N \rangle$  requires just the comparison of the data representations of  $\theta_{M;k,r}^N$  and  $\theta_{M;k,s}^N$ ,  $\theta_{M;k,r}^N$  and  $\mathcal{P}^{-1} \theta_{M;k,s}^N$ , respectively, where  $\mathcal{P}$  is a permutation defined analogously to  $\mathcal{P}$ , but acting on the spin function symbols  $\alpha$  and  $\beta$  (it is easy to verify that  $\mathcal{P} = \mathcal{P}^{-1}$ ). Here we assume that the code uses consistently identical data representations for all  $\theta_{M;k,r}^N$  and  $\mathcal{P}^{-1} \theta_{M;k,r}^N$  throughout.

In an initial version of the program, following the conventions adopted for leading terms [see code fragment (21)], we employed strings of 'a' and 'b' characters to record all required  $\theta_{M;k,r}^N$ . However, use of a whole byte (that is the amount of memory usually occupied by a single character) to hold a one-electron spin function, which can take one of only two values ( $\alpha$  and  $\beta$ ) is certainly excessive. It is much more efficient and economical to represent  $\alpha$  and  $\beta$  functions as *unset* and *set bits*, respectively. Thus, a standard four-byte integer variable can hold a product of up to  $4 \times 8 = 32$  spin functions. In fact, as bit fields are conventionally indexed starting from the least-significant bit, which is assigned an index of 0, we have found it convenient to record a spin product in bits 1 to 31 (reading from the least-significant bit, i.e. from the right-hand side of the 32-bit field), which allows treatment of systems containing up to 31 electrons. This is more than sufficient for most practical purposes. For example, the binary representations of the leading terms from Eq. (20), and the corresponding decimal integer numbers are obtained as (only the underlined five bits from each bit field are affected)

$$\begin{aligned} \theta_{1/2;1,1}^5 &= \alpha\beta\alpha\beta\alpha \rightarrow 00000000 \ 00000000 \ 00000000 \ 00010100 = 20, \\ \theta_{1/2;2,1}^5 &= \alpha\alpha\beta\beta\alpha \rightarrow 00000000 \ 00000000 \ 00000000 \ 00011000 = 24, \\ \theta_{1/2;3,1}^5 &= \alpha\beta\alpha\alpha\beta \rightarrow 00000000 \ 00000000 \ 00000000 \ 00100100 = 36, \\ \theta_{1/2;4,1}^5 &= \alpha\alpha\beta\alpha\beta \rightarrow 00000000 \ 00000000 \ 00000000 \ 00101000 = 40, \\ \theta_{1/2;5,1}^5 &= \alpha\alpha\alpha\beta\beta \rightarrow 00000000 \ 00000000 \ 00000000 \ 00110000 = 48, \end{aligned} \quad (25)$$

We have found that utilization of bit-coded spin products, instead of character strings, speeds up SPINS by approximately a factor of *five* on all platforms which have been used to run the program.

It is most straightforward to explain the algorithm for expanding a Rumer spin function (16) in terms of spin products implemented in SPINS on a particular example. Let us consider one of the singlet Rumer spin functions for a system of 6 electrons (note the use of curly braces to indicate grouping; for convenience, we have omitted the normalization constant):

$${}^R\theta_{00;k}^6 = [\alpha(\mu_1)\beta(\mu_2) - \beta(\mu_1)\alpha(\mu_2)] \{ [\alpha(\mu_3)\beta(\mu_4) - \beta(\mu_3)\alpha(\mu_4)] \\ \times \{ [\alpha(\mu_5)\beta(\mu_6) - \beta(\mu_5)\alpha(\mu_6)] \} \}.$$

Observing the usual rules of precedence, this spin function can be expanded as a sum of spin products:

$${}^R\theta_{00;k}^6 = \theta_{0;k,1}^6 + \theta_{0;k,2}^6 + \dots + \theta_{0;k,8}^6,$$

where

|                      |                                |                                |                                |
|----------------------|--------------------------------|--------------------------------|--------------------------------|
| $\theta_{0;k,1}^6 =$ | $\alpha(\mu_1)\beta(\mu_2)$    | $\alpha(\mu_3)\beta(\mu_4)$    | $\alpha(\mu_5)\beta(\mu_6)$    |
| $\theta_{0;k,2}^6 =$ | $\alpha(\mu_1)\beta(\mu_2)$    | $\alpha(\mu_3)\beta(\mu_4)$    | $[-\beta(\mu_5)\alpha(\mu_6)]$ |
| $\theta_{0;k,3}^6 =$ | $\alpha(\mu_1)\beta(\mu_2)$    | $[-\beta(\mu_3)\alpha(\mu_4)]$ | $\alpha(\mu_5)\beta(\mu_6)$    |
| $\theta_{0;k,4}^6 =$ | $\alpha(\mu_1)\beta(\mu_2)$    | $[-\beta(\mu_3)\alpha(\mu_4)]$ | $[-\beta(\mu_5)\alpha(\mu_6)]$ |
| $\theta_{0;k,5}^6 =$ | $[-\beta(\mu_1)\alpha(\mu_2)]$ | $\alpha(\mu_3)\beta(\mu_4)$    | $\alpha(\mu_5)\beta(\mu_6)$    |
| $\theta_{0;k,6}^6 =$ | $[-\beta(\mu_1)\alpha(\mu_2)]$ | $\alpha(\mu_3)\beta(\mu_4)$    | $[-\beta(\mu_5)\alpha(\mu_6)]$ |
| $\theta_{0;k,7}^6 =$ | $[-\beta(\mu_1)\alpha(\mu_2)]$ | $[-\beta(\mu_3)\alpha(\mu_4)]$ | $\alpha(\mu_5)\beta(\mu_6)$    |
| $\theta_{0;k,8}^6 =$ | $[-\beta(\mu_1)\alpha(\mu_2)]$ | $[-\beta(\mu_3)\alpha(\mu_4)]$ | $[-\beta(\mu_5)\alpha(\mu_6)]$ |

On inspection of the three vertical columns formed by  $\alpha\beta$  and  $-\beta\alpha$  factors from all eight spin products, it is easy to notice that the first column (arising from the expansion of the first singlet pair in the original Rumer spin function) contains four  $\alpha(\mu_1)\beta(\mu_2)$  factors, followed by four  $-\beta(\mu_1)\alpha(\mu_2)$  factors (reading from the top to the bottom). The second column (arising from the expansion of the second singlet pair in the original Rumer spin function) involves a pattern of two  $\alpha(\mu_3)\beta(\mu_4)$  factors, followed by two  $-\beta(\mu_3)\alpha(\mu_4)$  factors, which is repeated twice. The third column exhibits a pattern of one  $\alpha(\mu_5)\beta(\mu_6)$  factor, followed by one  $-\beta(\mu_5)\alpha(\mu_6)$  factor and repeated four times.

This can be easily generalized to the expansion of a Rumer function including  $N_p$  pairs, assuming that the spin products are arranged similarly to the preceding 6-electron example: Then the first column contains  $2^{N_p-1}$   $\alpha(\mu_1)\beta(\mu_2)$  factors, followed by the same number of  $-\beta(\mu_1)\alpha(\mu_2)$  factors, while column  $i_p$  (arising from pair  $i_p$  in the original Rumer spin function  $1 \leq i_p \leq N_p$ ) involves a pattern of  $2^{N_p-i_p}$   $\alpha(\mu_{2i_p-1})\beta(\mu_{2i_p})$  factors, followed by the same number of  $-\beta(\mu_{2i_p-1})\alpha(\mu_{2i_p})$  factors and repeated  $2^{i_p-1}$  times.

It is convenient to obtain the expansion of a Rumer function in terms of spin products by filling sequentially the vertical columns arising from different singlet



pairs. SPINS uses the following code to achieve this (excerpt from subroutine *fdeter*):

```

do 40 l = 1, ifns
  do 5 i = 1, ndet
    rs(i, l) = 1
    rd(i, l) = 0
5 continue
do 30 ipair = 1, npairs
  ip = rf(2 * ipair - 1, l)
  jp = rf(2 * ipair, l)
  lpart = 2 ** (npairs - ipair)
  do 20 j = 1, 2 ** (ipair - 1)
    joffs = (j - 1) * 2 * lpart
    do 10 k = joffs + 1, joffs + lpart
      rd(k, l) = ibset(rd(k, l), jp)
      klp = k + lpart
      rd(klp, l) = ibset(rd(klp, l), ip)
      rs(klp, l) = - rs(klp, l)
10 continue
20 continue
30 continue
40 continue

```

(26)

The spin products are assembled in the two-dimensional integer array *rd*. Its indices (*i*, *l*) range through the  $2^{N_p}$  (*ndet*) spin products within the expansion of a Rumer spin function and through all  $f_S^N$  (*ifns*) Rumer spin functions, respectively. The signs of the spin products  $[(-1)^{\lambda_r}]$ , see Eq. (24)] are recorded in the integer array *rs*. The variables corresponding to the quantities  $i_p$ ,  $N_p$ ,  $\mu_{2i_p-1}$  and  $\mu_{2i_p}$  from the preceding discussion are *ipair*, *npair*, *ip* and *jp*, respectively. Note that it is not necessary to do anything about the unpaired  $\alpha$  spins: They are represented by unset bits, which is handed automatically by the initialization of the elements of *rd* to zero.

This concludes the construction of the Rumer spin basis.

### 3 Kotani spin basis and its relation to the Rumer spin basis

The set of Kotani spin functions  $\{\mathcal{K}\Theta_{SM}^N\}$  is generated by successive coupling of individual electron spins according to the rules for addition of angular momenta. Each  $\mathcal{K}\Theta_{SM,k}^N$  is uniquely defined by the sequence of partial resultant spins obtained after combining the spins of the first 1, 2, ...,  $N - 1$  electrons, which can be used as an extended label for the spin function (it is not necessary to indicate  $S_N$  since it coincides with the total spin  $S$ ):

$$k \equiv (S_1 S_2 \dots S_{N-1}). \quad (27)$$

The gradual formation of all Kotani spin functions and the degeneracy of the individual spin states can be represented in graphical form by means of the Kotani branching diagram (see Fig. 1). This is a plot of the allowed values of the total spin  $S$  against the number of electrons  $N$ . One can associate with each function  $\mathcal{K}\Theta_{SM,k}^N$  a particular path on the branching diagram which follows the arrows connecting

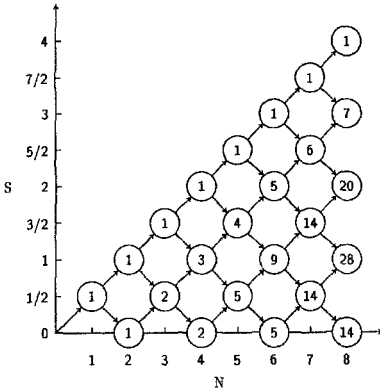


Fig. 1. Kotani branching diagram

the points  $(1, S_1), (2, S_2), \dots, (N, S)$ . The number of all possible paths is, of course,  $f_S^N$ , and it is easy to see that

$$f_S^N = f_{S+1/2}^{N-1} + f_{S-1/2}^{N-1}. \tag{28}$$

The branching diagram allows the introduction of an alternative notation for labeling the different  ${}^K\Theta_{SM;k}^N$ , obtained by starting from the origin  $(0, 0)$  and recording the directions of all upward and downward arrows (say by  $a$  and  $b$ , respectively) leading to the desired  $(N, S)$ , i.e.

$$k \equiv (c_{k_1} c_{k_2} \dots c_{k_N}), \tag{29}$$

where

$$c_{k_\mu} = \begin{cases} a & \text{if } S_{\mu-1} < S_\mu; \\ b & \text{if } S_{\mu-1} > S_\mu. \end{cases}$$

This notation can be used in order to establish an ordering of the Kotani spin functions according to the so-called *last-letter sequence* [7]. Let the first non-identical letters in  $(c_{k_1} c_{k_2} \dots c_{k_N})$  and  $(c_{l_1} c_{l_2} \dots c_{l_N})$ , reading from right to left, be  $c_{k_n}$  and  $c_{l_n}$ .  $(c_{k_1} c_{k_2} \dots c_{k_N})$  is assumed to precede  $(c_{l_1} c_{l_2} \dots c_{l_N})$  if  $c_{k_n} = b$ . For example, the five Kotani spin functions for  $N = 5, S = \frac{1}{2}$  are ordered as

$$1 \equiv (aaabb), \quad 2 \equiv (aabab), \quad 3 \equiv (abaab), \quad 4 \equiv (aabba), \quad 5 \equiv (ababa). \tag{30}$$

Comparison of this example with Eq. (20) shows that the “branching-diagram-path” label for Kotani spin function  $k$  is equivalent to the expression for leading term  $f_{1/2}^3 + 1 - k$ , except for the choice of letters. This holds for any allowed pair of  $N$  and  $S$  values. In fact, the algorithm for constructing leading terms, explained in Sect. 2, simply instructs one how to obtain all paths connecting the origin  $(0, 0)$  to a point  $(N, S)$  in an *inverse* last-letter ordering sequence using  $\alpha$  and  $\beta$  in the place of  $a$  and  $b$ , respectively.

As has been demonstrated by Simonetta et al. [10], there exists a simple relationship between the Rumer spin functions, constructed by means of the leading-term technique, and the Kotani spin functions, ordered in the last-letter sequence: Schmidt orthogonalization of  ${}^R\Theta_{SM;1}^N, {}^R\Theta_{SM;2}^N, \dots, {}^R\Theta_{SM;f_S^N}^N$  yields  ${}^K\Theta_{SM;f_S^N}^N, {}^K\Theta_{SM;f_S^N-1}^N, \dots, {}^R\Theta_{SM;1}^N$ .

SPINS makes use of this relationship in order to calculate the Rumer-to-Kotani transformation matrix  ${}^{\text{RK}}\mathbf{T}_S^N$  [see Eq. (10)]: The Schmidt orthogonalization can be described by an upper-triangular matrix  ${}^{\text{RK}}\bar{\mathbf{T}}_S^N$

$${}^{\text{K}}\Theta_{SM;f_S^N+1-k}^N = \sum_{l=1}^k {}^{\text{R}}\Theta_{SM;l}^N [{}^{\text{RK}}\bar{\mathbf{T}}_S^N]_{lk}, \quad (31)$$

which gives the required  ${}^{\text{RK}}\mathbf{T}_S^N$  upon reordering:

$$[{}^{\text{RK}}\mathbf{T}_S^N]_{kl} = [{}^{\text{RK}}\bar{\mathbf{T}}_S^N]_{k,f_S^N+1-l}. \quad (32)$$

As a consequence,  ${}^{\text{RK}}\mathbf{T}_S^N$  is also triangular, but its non-zero elements are on and above the  $(f_S^N, 1)$ - $(1, f_S^N)$  diagonal. In fact, SPINS uses and prints out (if requested) the upper-triangular matrix  ${}^{\text{RK}}\bar{\mathbf{T}}_S^N$ . The Schmidt-orthogonalization code (subroutine *rumkot*) is fairly standard and follows closely the algorithm outlined by Shavitt [4].

#### 4 Serber spin basis and the Kotani-to-Serber transformation matrix

The Serber spin functions  ${}^{\text{S}}\Theta_{SM;k}^N$  are assembled consecutively from all two-electron singlet and triplet wavefunctions (and, in the odd-electron case, the one-electron spin function for the last electron). The intermediate values of the total spin can be used again, similarly to the case of the Kotani spin basis, in order to derive a compact notation for the individual Serber spin functions:

$$k \equiv \begin{cases} (\dots ((s_{12}s_{34})S_4; s_{56})S_6; \dots S_{N-2}; S_{N-1,N}) & \text{for } N \text{ even;} \\ \underbrace{(\dots ((s_{12}s_{34})S_4; s_{56})S_6; \dots S_{N-3}; S_{N-2,N-1})}_{(N-1)/2} S_{N-1}; 1/2 & \text{for } N \text{ odd,} \end{cases} \quad (33)$$

where  $s_{\mu-1,\mu}$  may be 0 or 1 depending on the coupling of the spins of electrons  $\mu$  and  $\mu - 1$  (singlet or triplet, respectively).

This particular approach to the construction of spin functions can be illustrated by means of a Serber branching diagram (see Fig. 2), conceived similarly to the Kotani branching diagram (see Fig. 1). A two-electron triplet state can be added to

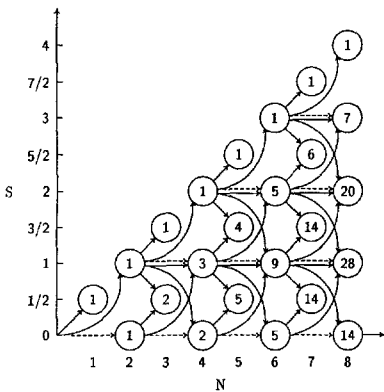


Fig. 2. Serber branching diagram

$(N - 2, S)$  in one of three possible ways: These are denoted by an arrow arc pointing upwards from  $(N - 2, S)$  to  $(N, S + 1)$ , an arrow arc pointing downwards from  $(N - 2, S)$  to  $(N, S - 1)$  and a horizontal arrow pointing from  $(N - 2, S)$  to  $(N, S)$ , respectively (the arrows from the last two cases can be drawn only if  $S > 0$  at  $N - 2$ ). The addition of a two-electron singlet spin state to  $(N - 2, S)$  is marked by a dashed horizontal arrow pointing from  $(N - 2, S)$  to  $(N, S)$ . The odd-electron spin states are connected to their even-electron precursors by upward and downward arrows, just as in the case of the Kotani branching diagram (Fig. 1); as Eq. (33) indicates, they represent termination points on the Serber branching diagram. For  $N$  even, the recursive relations involving the successive values of  $f_S^N$  for the Serber spin basis take the form

$$f_S^N = \begin{cases} f_{S+1}^{N-2} + f_{S-1}^{N-2} + 2f_S^{N-2} & \text{if } S > 0; \\ f_{S+1}^{N-2} + f_S^{N-2} & \text{if } S = 0. \end{cases} \quad (34)$$

The relation for  $N$  odd is identical to its counterpart in the Kotani spin basis [see Eq. (4)].

The Serber spin functions, similarly to the Kotani spin functions, are uniquely defined by the different paths on the Serber branching diagram. These can be used again in order to devise an alternative classification scheme, in which the Serber spin functions for even-electron spin states are denoted by

$$k \equiv (d_{k_1}, d_{k_2} \dots d_{k_{N/2}}), \quad (35)$$

where

$$d_{k_\mu} = \begin{cases} \text{'A'} & \text{if } S_\mu = S_{\mu-2} \text{ and } s_{\mu-1, \mu} = 0; \\ \text{'B'} & \text{if } S_\mu = S_{\mu-2} + 1 \text{ (} s_{\mu-1, \mu} = 1\text{)}; \\ \text{'C'} & \text{if } S_\mu = S_{\mu-2} \text{ and } s_{\mu-1, \mu} = 1; \\ \text{'D'} & \text{if } S_\mu = S_{\mu-2} - 1 \text{ (} s_{\mu-1, \mu} = 1\text{)}. \end{cases}$$

The odd-electron case can be dealt with by appending the symbol  $c_{k_N}$  [see Eq. (29)] to the above index, i.e.

$$k \equiv (d_{k_1}, d_{k_2} \dots d_{k_{(N-1)/2}} c_{k_N}). \quad (36)$$

The distinctive feature of the Serber spin basis is that the representation matrices corresponding to the single interchanges  $\mathcal{P}_{12}, \mathcal{P}_{34}, \dots, \mathcal{P}_{N-1, N}$  for even  $N$ , or  $\mathcal{P}_{N-2, N-1}$  for odd  $N$ , are all diagonal:

$$[{}^S \mathbf{V}_S^N(\mathcal{P}_{\mu-1, \mu})]_{kl} = (-1)^{s_{\mu-1, \mu} + 1} \delta_{kl}. \quad (37)$$

Consequently, this is also known as the “ $\mathcal{P}_{\mu-1, \mu}$ -diagonal basis”.

This property provides a convenient background for the development of algorithms for constructing the transformation matrix  ${}^K \mathbf{S} \mathbf{T}_S^N$  [see Eq. (11)], which links the Kotani and Serber spin bases. Two algorithms of this type have been suggested by Pauncz [7] and by Raos et al. [18]. In SPINS we have realized the first of these algorithms, which has been extended to handle systems containing an odd number of electrons.

Let us start with the case when the number of electrons in the system  $N$  is even. Upon inspection of the Kotani branching diagram (see Fig. 1), it is easy to notice

that, if  $S > 0$ , there exist four different paths connecting the point  $(N, S)$  to its  $N - 2$ -electron precursors:  $(N - 2, S + 1)$ ,  $(N - 2, S)$  and  $(N - 2, S - 1)$ . The last two letters in the labels of the Kotani spin eigenfunctions realizing these paths [see Eq. (29)] are  $bb$ ,  $ab$ ,  $ba$  and  $aa$ , respectively. If all Kotani spin eigenfunctions are ordered in the last-letter sequence, the functions whose labels end in  $bb$ ,  $ab$ ,  $ba$  and  $aa$  appear as separate subsequent subsets of  $f_{S+1}^{N-2}$ ,  $f_S^{N-2}$ ,  $f_S^{N-2}$  and  $f_{S-1}^{N-2}$  functions, respectively. Using this convention, it can be shown that the representation matrix, corresponding to  $\mathcal{P}_{N-1, N}$  in the Kotani basis, has the following block form [7] (all blocks which are not shown explicitly, contain only zeros).

$${}^K\mathbf{V}_S^N(\mathcal{P}_{N-1, N}) = \begin{pmatrix} \boxed{\mathbf{I}_{f_{S+1}^{N-2}}} & & & & \\ & -u\mathbf{I}_{f_S^{N-2}} & v\mathbf{I}_{f_S^{N-2}} & & \\ & v\mathbf{I}_{f_S^{N-2}} & u\mathbf{I}_{f_S^{N-2}} & & \\ & & & & \boxed{\mathbf{I}_{f_{S-1}^{N-2}}} \end{pmatrix}, \quad (38)$$

where  $u = 1/(2S + 1)$ ,  $v = (1 - u^2)^{1/2}$  and  $\mathbf{I}_f$  denotes an  $f \times f$  unit matrix.

It is straightforward to obtain the orthogonal matrix, which brings  ${}^K\mathbf{V}_S^N(\mathcal{P}_{N-1, N})$  into diagonal form:

$${}^K\mathbf{W}_S^N(\mathcal{P}_{N-1, N}) = \begin{pmatrix} \boxed{\mathbf{I}_{f_{S+1}^{N-2}}} & & & & \\ & x\mathbf{I}_{f_S^{N-2}} & y\mathbf{I}_{f_S^{N-2}} & & \\ & -y\mathbf{I}_{f_S^{N-2}} & x\mathbf{I}_{f_S^{N-2}} & & \\ & & & & \boxed{\mathbf{I}_{f_{S-1}^{N-2}}} \end{pmatrix}, \quad (39)$$

where  $x = [(S + 1)/(2S + 1)]^{1/2}$  and  $y = (1 - x^2)^{1/2}$ . The result is

$${}^K\tilde{\mathbf{W}}_S^N(\mathcal{P}_{N-1, N}) {}^K\mathbf{V}_S^N(\mathcal{P}_{N-1, N}) {}^K\mathbf{W}_S^N(\mathcal{P}_{N-1, N}) = \begin{pmatrix} \boxed{\mathbf{I}_{f_{S+1}^{N-2}}} & & & & \\ & -\mathbf{I}_{f_S^{N-2}} & & & \\ & & \mathbf{I}_{f_S^{N-2}} & & \\ & & & \mathbf{I}_{f_S^{N-2}} & \\ & & & & \boxed{\mathbf{I}_{f_{S-1}^{N-2}}} \end{pmatrix}. \quad (40)$$

As a direct consequence, if the row vector of Kotani spin functions  ${}^K\Theta_{SM}^N$  is transformed to  ${}^K\Theta_{SM}^N {}^K\mathbf{W}_S^N(\mathcal{P}_{N-1, N})$ , in each of the resulting spin eigenfunctions the spins of the last two electrons will be combined with the preceding  $S_{N-2}$  values in a Serber-like fashion, i.e. simultaneously, as two-electron spin eigenfunctions, rather than consecutively, as individual one-electron spins. This can be expressed by replacing the last two letters in the branching-diagram symbols for Kotani spin functions (29) by one of the letters used within the branching-diagram symbols for Serber spin functions (35) as follows:

$$ab \rightarrow A, \quad aa \rightarrow B, \quad ba \rightarrow C, \quad bb \rightarrow D. \quad (41)$$

Let us now suppose that we have already calculated the Kotani to Serber transformation matrices [see Eq. (11)]  ${}^{KS}\mathbf{T}_{S+1}^{N-2}$ ,  ${}^{KS}\mathbf{T}_S^{N-2}$  and  ${}^{KS}\mathbf{T}_{S-1}^{N-2}$ . Then, bearing in mind the foregoing discussion,  ${}^{KS}\mathbf{T}_S^N$  is given by

$$\begin{aligned}
 {}^{KS}\mathbf{T}_S^N &= \begin{pmatrix} \boxed{{}^{KS}\mathbf{T}_{S+1}^{N-2}} & & & \\ & \boxed{{}^{KS}\mathbf{T}_S^{N-2}} & & \\ & & \boxed{{}^{KS}\mathbf{T}_S^{N-2}} & \\ & & & \boxed{{}^{KS}\mathbf{T}_{S-1}^{N-2}} \end{pmatrix} \\
 &\times \begin{pmatrix} \boxed{\mathbf{I}_{f_{S+1}^{N-2}}} & & & \\ & \boxed{x\mathbf{I}_{f_S^{N-2}} \quad y\mathbf{I}_{f_S^{N-2}}} & & \\ & \boxed{-y\mathbf{I}_{f_S^{N-2}} \quad x\mathbf{I}_{f_S^{N-2}}} & & \\ & & & \boxed{\mathbf{I}_{f_{S-1}^{N-2}}} \end{pmatrix} \\
 &= \begin{pmatrix} \boxed{\mathbf{T}_{S+1}^{N-2}} & & & \\ & \boxed{x\mathbf{T}_S^{N-2} \quad y\mathbf{T}_S^{N-2}} & & \\ & \boxed{-y\mathbf{T}_S^{N-2} \quad x\mathbf{T}_S^{N-2}} & & \\ & & & \boxed{\mathbf{T}_{S-1}^{N-2}} \end{pmatrix}. \tag{42}
 \end{aligned}$$

This expression enables recursive calculation of the Kotani-to-Serber transformation matrix for systems containing an even number of electrons. Formally, so far we have been dealing with the case of  $S > 0$ . The treatment can be extended to include the  $S = 0$  case in a straightforward manner, bearing in mind that the only  $(N - 2)$ -electron precursors to  $(N, 0)$  are  $(N - 2, 1)$  and  $(N - 2, 0)$ . As a result, the recursive relation producing  ${}^{KS}\mathbf{T}_0^N$  takes the form

$${}^{KS}\mathbf{T}_0^N = \begin{pmatrix} \boxed{{}^{KS}\mathbf{T}_1^{N-2}} & \\ & \boxed{{}^{KS}\mathbf{T}_0^{N-2}} \end{pmatrix}, \tag{43}$$

i.e. no further transformation is necessary. It is also easy to arrive at a similar conclusion in the case when the number of electrons in the system is odd. Then  ${}^{KS}\mathbf{T}_S^N$  is related to its predecessors through

$${}^{KS}\mathbf{T}_S^N = \begin{pmatrix} \boxed{{}^{KS}\mathbf{T}_{S+1/2}^{N-1}} & \\ & \boxed{{}^{KS}\mathbf{T}_{S-1/2}^{N-1}} \end{pmatrix}. \tag{44}$$

The appearance of Eqs. (42)–(44) suggests a recursive implementation. Although recursive functions and subroutines are beginning to make their way into Fortran, it is to be expected that, as a rule, a non-recursive algorithm would be more robust, as well as more efficient in terms of execution speed and memory requirements. A careful examination of Eqs. (42)–(44) shows that it is possible to

evaluate the Kotani-to-Serber transformation matrix without resorting to recursive programming. The resulting algorithm, which is implemented in SPINS, runs as follows (the corresponding Fortran code is included in subroutine *kotsrb*):

(i) Derive the branching-diagram labels for the Serber spin functions from those for the Kotani spin functions (ordered in the last-latter sequence) according to the correspondence rules in Eq. (41).

(ii) Initialize  ${}^{\text{KS}}\mathbf{T}_S^N$  to  $\mathbf{1}_{f_S^N}$ .

(iii) Initialize an array of partial spins  $s_k$  ( $k = 1, 2, \dots, f_S^N$ ) as follows:

Set  $s_k = 0$  if the first letter in the branching diagram label for Serber function  $k$  is  $A$ , otherwise set  $s_k = 1$ .

(iv) For  $i = 2, 3, \dots, N/2$  [or  $(N - 1)/2$ , if  $N$  is odd] do

(iv.i) Update the partial spins  $s_k$ :

if the  $i$ th letter in spin function  $k$  is  $B$  then set  $s_k = s_k + 1$ , else if the  $i$ th letter is  $D$  then set  $s_k = s_k - 1$ , otherwise leave  $s_k$  unchanged.

(iv.ii) For each region of equal partial spins  $s_j = s_{j+1} = \dots = s_{j+n_j-1}$  within  $s_1, s_2, \dots, s_{f_S^N}$ , where  $n_j > 1$  do

(iv.ii.i) Search through spin functions  $j, j + 1, \dots, j + n_j - 1$  for a subgroup of spin functions  $m, m + 1, \dots, m + n_m - 1$ , which contain  $A$  in position  $i$  and are followed by an equal number ( $n_m$ ) of spin functions, which contain  $C$  in position  $i$ . If such a group is found then for all  $k, l = m, m + 1, \dots, m + n_m - 1$  set

$$\begin{aligned} [{}^{\text{KS}}\mathbf{T}_S^N]_{kl} &= x_j [{}^{\text{KS}}\mathbf{T}_S^N]_{kl}, & [{}^{\text{KS}}\mathbf{T}_S^N]_{k,l+n_m} &= y_j [{}^{\text{KS}}\mathbf{T}_S^N]_{kl}, \\ [{}^{\text{KS}}\mathbf{T}_S^N]_{k+n_m,l} &= -y_j [{}^{\text{KS}}\mathbf{T}_S^N]_{kl}, & [{}^{\text{KS}}\mathbf{T}_S^N]_{k+n_m,l+n_m} &= x_j [{}^{\text{KS}}\mathbf{T}_S^N]_{kl}, \end{aligned}$$

where  $x_j = [(s_j + 1)/(2s_j + 1)]^{1/2}$  and  $y_j = (1 - x_j^2)^{1/2}$ .

This method of constructing the Kotani-to-Serber transformation matrix assumes that the Serber spin functions are ordered in the last-latter sequence similarly to the Kotani spin functions, which is ensured by the first step of the preceding algorithm. For example, the ordered branching-diagram labels for all five doublet Serber spin functions for a five-electron system can be obtained from their Kotani counterparts [see Eq. (30)] as

$$1 \equiv (BAb), \quad 2 \equiv (BCb), \quad 3 \equiv (ABb), \quad 4 \equiv (BDa), \quad 5 \equiv (AAa). \quad (45)$$

A distinct advantage of the algorithm for calculating the Kotani-to-Serber transformation matrix presented in this Section over the approach suggested by Raos et al. [18], is in the fact that it does not involve any matrix diagonalizations, which makes the process faster and numerically more stable.

## 5 Interpretation of spin-coupling patterns

The analysis of a spin eigenfunction in its most general form [see Eq. (4)] can be a difficult task, especially when the dimension of the spin degeneracy (3) is large. Transformation to a spin basis which reflects properly the symmetry of the system under investigation often provides a much clearer picture which makes better physical sense. The corresponding change of the spin-coupling coefficients ( $\mathbf{C}_S$ ) can be expressed as

$${}^{\text{R}}\mathbf{C}_S = {}^{\text{R}}\mathbf{K}\mathbf{T}_S^{\text{NK}}\mathbf{C}_S, \quad {}^{\text{S}}\mathbf{C}_S = {}^{\text{S}}\mathbf{K}\mathbf{T}_S^{\text{NK}}\mathbf{C}_S. \quad (46)$$

SPINS is capable of transforming a vector of spin-coupling coefficients, which defines a spin function (4) in the Rumer, Kotani or Serber spin basis, to the corresponding vector in any other of these bases.

Another important point in the analysis of the representation of a spin function within a given spin basis is to establish the relative significance of the individual terms in expansion (4). One way of calculating the weight with which a single spin eigenfunction  $\Theta_{SM,k}^N$  participates in  $\Theta_{SM}^N$  (or, in other words, its occupation number) is due to Chirgwin and Coulson [23] (it is assumed that  $\Theta_{SM}^N$  is normalized to unity):

$$P_{Sk}^{CC} = C_{Sk} \sum_{l=1}^{f_S^N} \langle \Theta_{SM,k}^N | \Theta_{SM,l}^N \rangle C_{Sl}. \quad (47)$$

Computational experience shows that for nonorthogonal spin bases the values of  $P_{Sk}^{CC}$  calculated by means of Eq. (47) can sometimes be either larger than unity or negative. Another expression for  $P_{Sk}$ , which guarantees that its value always stays within the range [0, 1], has been suggested by Gallup and Norbeck [24]:

$$P_{Sk}^{GN} = c(C_{Sk})^2 / (\langle \Theta_{SM}^N | \Theta_{SM}^N \rangle^{-1})_{kk}, \quad c^{-1} = \sum_{k=1}^{f_S^N} (C_{Sk})^2 / (\langle \Theta_{SM}^N | \Theta_{SM}^N \rangle^{-1})_{kk}. \quad (48)$$

In the case of orthogonal spin bases (e.g. the Kotani and Serber spin bases) both Eqs. (47) and (48) reduce to much simpler expressions:  ${}^K P_{Sk} = {}^K C_{Sk}^2$ ,  ${}^S P_{Sk} = {}^S C_{Sk}^2$ .

SPINS calculates and outputs occupation numbers for all components of each spin function it processes. Both Chirgwin–Coulson and Gallup–Norbeck occupation numbers are evaluated for spin functions defined in the Rumer spin basis.

It is often of interest in spin-coupled theory to establish the influence of a reordering of the spatial orbitals  $\psi_1, \psi_2, \dots, \psi_N$  on the spin-coupling pattern within  $\Theta_{SM}^N$  [see Eq. (8)]. If  $\mathcal{P}$  is a permutation defined analogously to Eq. (6), but acting on the orbital indices, such that

$$\mathcal{P}\psi_1\psi_2 \dots \psi_N = \psi_{p_1}\psi_{p_2} \dots \psi_{p_N},$$

then it is easy to show that

$$\mathcal{P}^{-1}\mathcal{A}(\psi_1\psi_2 \dots \psi_N \Theta_{SM}^N) = \varepsilon_{\mathcal{P}}\mathcal{A}(\psi_1\psi_2 \dots \psi_N \Theta_{SM}^N) = \mathcal{A}(\mathcal{P}\psi_1\psi_2 \dots \psi_N \mathcal{P}^{-1}\Theta_{SM}^N).$$

As a direct consequence of these equations, the spin-coupling pattern corresponding to the reordered product of spatial orbitals, is defined by

$$\mathbf{C}_S(\mathcal{P}) = \varepsilon_{\mathcal{P}} \mathbf{V}_S^N(\mathcal{P}^{-1}) \mathbf{C}_S. \quad (49)$$

SPINS implements this expression and thus allows investigation of the effect of any permutation of the spatial orbitals  $\psi_1, \psi_2, \dots, \psi_N$  participating in the spin-coupled wavefunction on the form of the related spin function  $\Theta_{SM}^N$ , which can be expressed in the Rumer, Kotani or Serber spin basis [see Eq. (9)].

## 6 Program control and performance

While developing SPINS, we had two objectives in mind: (i) to produce an efficient and well-documented set of small subroutines, which could later be incorporated in any program requiring spin eigenfunctions from the Rumer, Kotani or Serber spin



bases, the related transformation matrices, or the representation matrices of the symmetric group in these bases and (ii) to organize these routines in one program, which could be used directly to evaluate most of the expressions from the preceding Sections of this article.

Here we describe the operation of SPINS as a standalone program.

SPINS reads its input from Fortran unit 5 and writes its output to Fortran unit 6. On Unix and MS DOS systems these are attached, by default, to the standard input and output file, which allows one to invoke the program simply as

```
spins < input_file > output_file
```

The input file may contain the following keywords:

electrons  $N$

This defines the number of electrons in the system (an integer value, currently from 1 to 16, although the upper limit can be changed by a simple redimensioning of the arrays in the Fortran code); it should be the first directive in the command file. The `electrons` keyword initiates work on a completely new task, i.e., if it is repeated, SPINS ‘forgets’ any information which may have been supplied beforehand.

spin  $S$

Defines the required value of the total spin  $S$ .  $S$  should be compatible with  $N$ , i.e.  $0 \leq S \leq N/2$  and can take integer or half-integer values when  $N$  is even or odd, respectively. If the `spin` keyword is omitted, the default values of  $S$ , assumed by the program for  $N$  even or odd, are 0 (singlet) and 0.5 (doublet). The `spin` keyword, similarly to the `electrons` keyword, instructs SPINS to start a new task. The only difference is that it does not cancel a previously defined permutation (see the `permutation` keyword).

```
input Rumer | Kotani | Serber | Rumer_VB CS1 CS2 ... CSfS
```

This defines an input vector of spin-coupling ( $C_{sk}$ ) coefficients coming from a spin function expanded in the *Rumer*, *Kotani* or *Serber* spin basis [here and further in the text alternatives are separated by a vertical bar (‘|’)]. *Rumer\_VB* can be used to indicate that the input vector corresponds to Rumer spin functions which have the phases defined as in Raimondi’s VB program [25]. The name of the spin basis is expected to be followed by  $f_S^N$  numbers, forming the input vector of spin-coupling coefficients. This input vector is renormalized and printed out, together with the spin-function designations from one of Eqs. (17), (27), or (33). Even if the input has *Rumer\_VB* phases, the output vector is of the standard *Rumer* type (i.e. the signs of the spin-factors corresponding to the leading terms are positive in all spin functions). Then, if a permutation has already been defined (see the `permutation` directive), the input vector is transformed so as to produce  $C_S(\mathcal{P})$  [see Eq. (49)], which replaces  $C_S$ . This means that any subsequent transformations of the spin-coupling coefficients will act on  $C_S(\mathcal{P})$ .

permutation  $p_1 p_2 \dots p_N$

Defines the permutation

$$\begin{pmatrix} 1 & 2 & \dots & N \\ p_1 & p_2 & \dots & p_N \end{pmatrix}.$$

If an input vector of spin-coupling coefficients has already been defined, it is transformed to  $C_S(\mathcal{P})$  [see Eq. (49) and the explanation following the input keyword]. The permutation stays in effect until the next `electrons` or `permutation` keyword, if any, i.e. each new input vector of spin-coupling coefficients will be transformed in an analogous way.  $p_1, p_2, \dots, p_N$  should be unique integer numbers.

transform *Rumer* | *Kotani* | *Serber*

Requests transformation of the vector of spin-coupling coefficients defined previously by an `input` directive to the specified spin basis [see Eq. (46)]. Note that if the initial vector is in the *Rumer* spin basis, then transform *Kotani*, followed by transform *Serber* produces the same final vector as transform *Serber* only.

representation *Rumer* | *Kotani* | *Serber*

Requests calculation of the representation matrix  $\mathbf{V}_S^N(\mathcal{P})$  [see Eq. (7)], which corresponds to a permutation  $\mathcal{P}$ , defined previously by a `permutation` keyword, in the specified spin basis.

```
print leading_terms | Kotani_and_Serber_paths | Rumer_functions |
      Rumer_overlaps | transformations | everything | all
```

Requests increased output detail. *everything* and *all* are synonyms.

A line with a percentage sign (%) in the first position is considered to be a comment line and is copied directly to the output file.

There are very few restrictions on the format of the input file. Keywords and numbers have to be separated by any non-zero number of blanks. A blank is considered to be one of the characters space (' '), comma (','), semicolon (';'), slash ('/'), as well as the end-of-line separator. A keyword and its parameters may span several lines, and a single line may contain several keywords. Keywords may be specified in any mixture of upper and lower case. If it discovers a syntax error, the input parser outputs an appropriate message and terminates the execution of the program.

An example input file and the resulting output file, both coming from spin-coupled investigations of the electronic structure of polyenyl radicals [26], are included in Appendix.

The performance of SPINS can be judged from the following timings: In relation to a recent spin-coupled study of chemical bonding to hypercoordinate second-order atoms [27], we had to find the vectors of spin-coupling coefficients corresponding to reordered orbital products in the *Rumer* spin basis, and then to transform them from the *Rumer* to the *Serber* spin basis for two singlet

spin-coupled wavefunctions, involving 12 valence electrons each ( $f_0^{12} = 132$ ). The total CPU time SPINS took to process this task was 36 s on an IBM RS/6000 model 530H (AIX 3.2.4, xlf 2.3.0) and 54 s on an IBM-compatible PC equipped with a 66 MHz 80486 processor (MS DOS 6.2, Microsoft Fortran Power Station 1.0). Smaller problems run even faster, sometimes it may be even convenient to pipe the output of SPINS to more, i.e. `spins < input_file | more`.

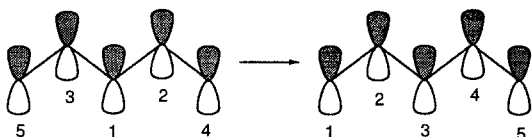
## 7 Concluding remarks

The algorithms incorporated in SPINS present simple, effective and robust solutions to a number of problems associated with the use of Rumer, Kotani and Serber spin eigenfunctions in quantum chemistry. All related Fortran-77 sources, as well as assistance with their use, are available on the Internet from P. B. Karadakov@Bristol.AC.UK or DLC@Liverpool.AC.UK. The comments within the source code, together with the descriptions included in the present article should be sufficient to allow easy incorporation of any of the provided program units into another quantum-chemical package.

As a standalone program, SPINS has already proven to be an essential tool for interpreting spin-coupled wavefunctions. Its recent applications include the comparison of the bent-bond and  $\sigma$ - $\pi$  models for the carbon-carbon double and triple bonds in ethene and ethyne [28], the study of the lowest singlet and triplet states of *o*-benzynes [29], the spin-coupled description of the electronic structure of polyenyl radicals [26], the spin-coupled model for the mechanism of the ethene + ethene cycloaddition reaction [30], the description of chemical bonding to hypercoordinate second-row atoms [27], the comparison between the nature of carbon-carbon bonds in cyclopropane and cyclobutane and the question of  $\sigma$ -aromaticity in cyclopropane [31].

## Appendix

The input and output files included here provide an example of the use of SPINS in order to achieve a VB-style interpretation of the spin-coupled results for  $C_5H_7$ , which form a part of a recent investigation of the electronic structure of polyenyl radicals [26]. The spatial orbitals stemming from the spin-coupled calculation are of  $\pi$ -symmetry and closely resemble distorted carbon  $2p_z$  atomic orbitals. However, they are not appropriately ordered (see the scheme below), and the optimized spin-coupling coefficients refer to the Kotani spin basis. The reordering of the orbitals



and subsequent transformations of the spin-coupling coefficients, first to the Rumer and then to the Serber spin basis, can be carried out by running SPINS on

the following input data:

```
%
% C5H7 (deformed AOs)
%
electrons 5 spin 0.5
permutation 5 3 1 2 4
input kotani
    0.4072934377,    0.0998952744,    -0.5300274516,
    -0.5300274517,    -0.5121277093,
transform rumer
transform serber
```

As a result, SPINS produces the following output (timings refer to an IBM RS/6000 530H):

```
%
% C5H7 (deformed AOs)
%
System of 5 electrons with spin 1/2 (f^N_S = 5)
Permutation:
5 3 1 2 4
Inverse permutation:
3 4 2 5 1
```

Renormalized input spin function coefficients (Kotani spin basis):

| k  | Spin-Coupling pattern | Coefficient | Weight   |
|----|-----------------------|-------------|----------|
| 1: | ((1/2)1(3/2)1)        | 0.407293    | 0.165888 |
| 2: | ((1/2)1(1/2)1)        | 0.099895    | 0.009979 |
| 3: | ((1/2)0(1/2)1)        | -0.530027   | 0.280929 |
| 4: | ((1/2)1(1/2)0)        | -0.530027   | 0.280929 |
| 5: | ((1/2)0(1/2)0)        | -0.512128   | 0.262275 |

Permuted spin function coefficients (Kotani spin basis):

| k  | Spin-Coupling pattern | Coefficient | Weight   |
|----|-----------------------|-------------|----------|
| 1: | ((1/2)1(3/2)1)        | -0.041582   | 0.001729 |
| 2: | ((1/2)1(1/2)1)        | 0.250368    | 0.062684 |
| 3: | ((1/2)0(1/2)1)        | -0.626405   | 0.392383 |
| 4: | ((1/2)1(1/2)0)        | -0.178502   | 0.031863 |
| 5: | ((1/2)0(1/2)0)        | 0.715081    | 0.511341 |

## Kotani to Rumer transformation

Transformed spin function coefficients (Rumer spin basis):

| k  | Spin-Coupling pattern | Coefficient | Weight   | GWeight  |
|----|-----------------------|-------------|----------|----------|
| 1: | (1 - 2, 3 - 4, 5)     | 0.294620    | 0.210677 | 0.138839 |
| 2: | (1 - 4, 2 - 3, 5)     | -0.058806   | 0.030116 | 0.007375 |
| 3: | (1 - 2, 4 - 5, 3)     | -0.576000   | 0.518413 | 0.707572 |
| 4: | (2 - 3, 4 - 5, 1)     | 0.294620    | 0.210677 | 0.138839 |
| 5: | (2 - 5, 3 - 4, 1)     | -0.058806   | 0.030116 | 0.007375 |

## Rumer to Serber transformation

Transformed spin function coefficients (Serber spin basis):

| k  | Spin-Coupling pattern | Coefficient | Weight   |
|----|-----------------------|-------------|----------|
| 1: | ((10)1;(1/2))         | 0.178502    | 0.031863 |
| 2: | ((11)1;(1/2))         | 0.180417    | 0.032550 |
| 3: | ((01)1;(1/2))         | -0.626405   | 0.392383 |
| 4: | ((11)0;(1/2))         | -0.178502   | 0.031863 |
| 5: | ((00)0;(1/2))         | 0.715081    | 0.511341 |

User time: 0.08 s  
 System time: 0.02 s  
 Total time: 0.10 s

The transformed spin-coupling coefficients in the Rumer spin basis reveal a clear resonance pattern, dominated by structures involving spin functions  $3 \equiv (1 - 2, 4 - 5, 3)$ ,  $1 \equiv (1 - 2, 3 - 4, 5)$ ,  $4 \equiv (2 - 3, 4 - 5, 1)$  (the last two have equal weights due to the symmetry of the system).

Insertion of the instruction `print leading_terms` at the beginning of the input data would result in the addition of information about the leading terms of the output:

Leading terms (VB signs):

1: ababa (- 1)  
 2: aabba ( 1)  
 3: abaab ( 1)  
 4: aabab (- 1)  
 5: aaabb ( 1)

The instruction `print rumer_functions` would produce a detailed printout of the Rumer spin basis for the system:

Rumer functions expanded in terms of signed spin factors:

|               |             |             |            |  |
|---------------|-------------|-------------|------------|--|
| Spin function | 1           |             |            |  |
| 1: (ababa)    | 2: -(abbaa) | 3: -(baaba) | 4: (babaa) |  |
| Spin function | 2           |             |            |  |
| 1: (aabba)    | 2: -(ababa) | 3: -(babaa) | 4: (bbaaa) |  |
| Spin function | 3           |             |            |  |
| 1: (abaab)    | 2: -(ababa) | 3: -(baaab) | 4: (baaba) |  |

|               |              |              |            |  |
|---------------|--------------|--------------|------------|--|
| Spin function | 4            |              |            |  |
| 1: (aabab)    | 2: – (aabba) | 3: – (abaab) | 4: (ababa) |  |
| Spin function | 5            |              |            |  |
| 1: (aaabb)    | 2: – (aabab) | 3: – (ababa) | 4: (abbaa) |  |

The directives `print rumer_overlaps` and `print transformations` would instruct SPINS to include in the output the overlap matrix between Rumer spin functions and the Rumer-to-Kotani, Kotani-to-Rumer and Kotani-to-Serber transformation matrices, respectively.

Finally, the instruction `print kotani_and_serber_paths` would add to the output the correspondence between the paths on the Kotani and Rumer branching diagrams:

Kotani and Serber spin functions---paths on the branching diagrams:

| k  | Kotani | Serber |
|----|--------|--------|
| 1: | aaabb  | BAb    |
| 2: | aabab  | BCb    |
| 3: | abaab  | ABb    |
| 4: | aabba  | BDa    |
| 5: | ababa  | AAa    |

## References

- McWeeny R (1992) *Methods of molecular quantum mechanics*, Academic Press, London
- Cooper DL, Gerratt J, Raimondi M (1991) *Chem Rev* 91:929
- Bobrowitz FM, Goddard WA (1977) In: Schaefer III HF (ed) *Modern theoretical chemistry*, Vol 3, *Methods of electronic structure theory* p 79, Plenum Press, New York
- Shavitt I (1977) In: Schaefer III HF (ed) *Modern theoretical chemistry*, Vol 3, *Methods of electronic structure theory* p 189, Plenum Press, New York
- Wahl AC, Das G (1977) In: Schaefer III HF (ed) *Modern theoretical chemistry*, Vol 3, *Methods of electronic structure theory*, Plenum Press, New York
- Roos BO (1987) *Adv Chem Phys* 59:399
- Pauncz R (1979) *Spin eigenfunctions*. Plenum Press, New York
- Rumer G (1932) *Göttinger Nachr* 3:337
- Kotani M, Amemiyama A, Ishiguro E, Kimura T (1963) *Tables of molecular integrals*, Maruzen, Tokyo
- Serber R (1934) *Phys Rev* 45:461
- Serber R (1934) *J Chem Phys* 2:697
- Löwdin PO (1955) *Phys Rev* 6:1509
- Löwdin PO (1964) *Rev Mod Phys* 36:966
- Gerratt J, Lipscomb WN (1968) *Proc Natl Acad Sci (USA)* 59:332
- Pyper NC, Gerratt J (1977) *Proc Roy Soc Lond, Ser A* 355:407
- Gerratt J, Raimondi M (1980) *Proc Roy Soc Lond, Ser A* 371:525
- Karadakov PB, Gerratt J, Cooper DL, Raimondi M (1992) *J Chem Phys* 97:7637
- Raos G, Gerratt J, Cooper DL, Raimondi M (1993) *Mol Phys* 79:197
- Simonetta M, Gianinetti E, Vandoni I (1968) *J Chem Phys* 48:1579
- Reeves CM (1966) *Commun ACM* 9:276
- Sutcliffe BT (1966) *J Chem Phys* 45:235
- Cooper IL, McWeeny R (1966) *J Chem Phys* 45:226, 3484
- Chirgwin BH, Coulson CA (1950) *Proc Roy Soc Lond Ser A* 201:196
- Gallup GA, Norbeck JM (1973) *Chem Phys Lett* 21:495

25. Raimondi M, Sironi M (1990) In: Klein DJ, Trinajstić N (eds) Valence bond theory and chemical structure p 111, Elsevier, Amsterdam
26. Karadakov PB, Gerratt J, Raos G, Cooper DL, Raimondi M (1994) *J Am Chem Soc* 116:2075
27. Cooper DL, Cunningham TP, Gerratt J, Karadakov PB, Raimondi M (1994) *J Am Chem Soc* 116:4414
28. Karadakov PB, Gerratt J, Cooper DL, Raimondi M (1993) *J Am Chem Soc* 115:6863
29. Karadakov PB, Gerratt J, Raos G, Cooper DL, Raimondi M (1993) *Isr J Chem* 33:253
30. Karadakov PB, Gerratt J, Cooper DL, Raimondi M (1994) *J Chem Soc Faraday Trans* 90:1643
31. Karadakov PB, Gerratt J, Cooper DL, Raimondi M (1994) *J Am Chem Soc* 116:7714