CMST 7(1) 27-39 (2001) DOI:10.12921/cmst.2001.07.01.27-39

COMPUTATIONAL METHODS IN SCIENCE AND TECHNOLOGY 7 (1), 27-39 (2001)

COMBINATORIAL STRUCTURES IN SPIN MODELS: A METHOD OF OPERATOR MATRICES GENERATION

SYLWIA BUCIKIEWICZ, WOJCECH FLOREK

Adam Mickiewicz University, Institute of Physics
Umultowska 85, 61-614 Poznań, Poland
e-mail: bucz@spin.amu.edu.pl, florek@spin.amu.edu.pl

Abstract. Finite spin models, applicable to investigations of mesoscopic rings, give rise to eigenproblems of very large dimensions. Efficient, and as exact as possible, solutions of such eigenproblems are very difficult. A method leading to block diagonalization of Hamiltonian matrix is proposed in this paper. For a given symmetry group of a Heisenberg Hamiltonian commuting with the total spin projection (i.e. with the total magnetization being a good quantum number) appropriate combinatorial and group-theoretical structures (partitions, orbits, stabilizers *etc.)* are introduced and briefly discussed. Generation of these structures can be performed by means of algorithms being modifications of standard ones. Main ideas are presented in this paper, whereas the actual form of algorithms will be discussed elsewhere.

1. INTRODUCTION

Finite spin models have been used in condensed matter physics for many years [1] and have been applied to various models of magnetic materials: from the one-dimensional ferromagnetic Ising model to the three-dimensional Heisenberg antiferromagnets. Different techniques for the extrapolation to the thermodynamic limit have been introduced and many different parameters have been calculated: ground state energy, specific heat, spin correlations in excited states etc. Recent developments and discoveries have given rise to interests in finite spin models, since they can be applied to meso- and nanoscopic systems, especially magnetic macromolecules with well-determined symmetry groups [2], In modeling of infinite magnets much more efforts are devoted to small spin numbers (the Ising models or $s = {}^1/_2$, $1, {}^3/_2$), but with a very large number of nodes (up to several millions in the Monte Carlo methods). On the contrary, macromolecules contain several (up to $N = 30$) magnetic ions with relatively large spin numbers (up to $s = {}^5/_2$), what also leads to a veiy large number of states even for a small number of magnetic ions N. Moreover, this number increases rapidly with increasing N. However, group-theoretical and combinatorial methods yield decomposition of a space of states into subspaces with relatively small dimensions and for small and rather simple symmetry groups, like cyclic or dihedral ones, it can be quite easy. Therefore, magnetic properties of macromolecules can be investigated very deeply and one can fit model parameters in more accurate way.

The present paper is motivated by possible applications of combinatorial structures in spin (magnetic) models. However, the described approach, algorithms, suggested solutions etc. can be used in any problem, where a basis of a space of states can be viewed as a set of configurations. Of course, the group-theoretical description implies that a symmetry group of a Hamiltonian can be determined. Moreover, if $|m_0, m_1, ..., m_{N-1})$ denotes a basis vector then it is assumed that an operator

$$A|m_0, m_1, ..., m_{N-1}\rangle = \left(\sum_{j=0}^{N-1} \alpha_j m_j\right) |m_0, m_1, ..., m_{N-1}\rangle$$

commutes with a Hamiltonian. (In this article elements of all arrays are labeled from 0, as it is assumed in C. This is also the standard notation used in Kerber's monograph [3]). In the case of magnetic models A is the total spin projection S^z with $\alpha_j = 1$ for all $0 \leq j < N$. Of course, it implies that, in the case of the Heisenberg model, uniaxial anisotropy is only admissible. However, even in the other cases classification of states with the use of eigenvalues M of S^z can be very helpful.

The paper is devoted to presentation of steps necessary to solve an eigenproblem of a Hamiltonian, with a given symmetry group, acting in a finite space of states. The main purpose of proposed procedure is to decrease dimension of the eigenproblem, whereas its solution can be performed by one of the standard numerical methods and, therefore, it is not discussed here. In the next section main objects are presented in two aspects: as elements of spin models and as combinatorial structures. The problem is formulated in the Sec. 3, where the basic ideas of the proposed solution are presented. The algorithms are presented in Sec. 4. We focus on combinatorial aspects of the presented problem, so most interests is devoted to two algorithms: (i) generation of partitions of a number N into no more than n parts and (ii) generation of all subsets of N-element set - there are n subsets with cardinalities k_0, k_1,..., k_{n-1} The paper ends with the overall summary.

2. FINITE SPIN MODELS AS COMBINATORIAL OBJECTS

One of standard ways of quantum magnets modeling is to consider a finite set of N nodes carrying spins s_j, $0 \leq j < N$, with a spin number s. Each spin can be in one of $2s + 1$ states $m_j = -s, -s + 1, ..., s$. Tensor products of one-spin states $|m_j>$ form a natural basis of the quantum space of states. In the case of finite spin models there are $(2s + 1)^N$ such states written as

$$|m_0, m_1, ..., m_{N-1}\rangle \tag{1}$$

and called *Ising states* or *configurations.* However, these states, except for the special cases, are *not* eigenvectors of the Heisenberg Hamiltonians. In the present considerations bilinear isotropic interactions are taken into account, so a Hamiltonian has the following form:

$$\mathcal{H} = \sum_{\langle jk\rangle} J_{jk} \mathbf{s}_j \cdot \mathbf{s}_k = \sum_{\langle jk\rangle} J_{jk} [(s_j^+ s_k^- + s_j^- s_k^+)/2 + s_j^z s_k^z], \tag{2}$$

where the sum is taken over *all* pairs (jk) and $s^{\pm} = s^x \pm is^y$ are the standard step operators. This Hamiltonian commutes with $\mathbf{S}^2$ and S^z, so the total spin number S and the magnetization M are good quantum numbers. Investigation of the first operator lies out of a scope of this paper, but it is veiy important that due to the relation

$$\mathbf{S}^2 = \left(\sum_{j=0}^{N-1} \mathbf{s}_j\right)^2 = Ns(s+1) + \mathcal{H}(J_{jk} = 2) \tag{3}$$

the methods presented below can be applied to its eigenproblem. From the combinatorial point of view the total magnetization is more interesting, because: (i) the states (1) are eigenvectors of S^z with eigenvalues $M = \Sigma_{j=0}^{N-1} m_j$ and (ii) S^z commutes with *all* permutations of nodes. Moreover, the total spin can be considered as a good quantum number for isotropic Hamiltonians only.

The first property yields a decomposition of states (1) into subsets (sub-bases) with a given magnetization M- $-Ns$, $-Ns + 1, \ldots, Ns$. Moreover, the value of M depends on numbers k_i, $i = 0, 1, 2, \ldots, 2s$, of solutions of equations $m_j = i - s$. Of course

$$\sum_i k_i = N \qquad \text{and} \qquad \sum_i k_i(i-s) = M, \tag{4}$$

so $[k] = [k_0, k_1, \ldots, k_{2s}]$ is a partition of N into no more than $2s + 1$ parts. Partitions are grouped in classes represented by ordered partitions $k_0 \geq k_1 \geq \ldots \geq k_{2s}$. A number of states (configurations) corresponding to an ordered partition, *i.e.* states with k_i projections $m_j = i - s$, is given by a polynomial coefficient N! (k_0! k$_1$!... k_{2s}!). In the further considerations the time-reversal symmetry of the Hamiltonian (2) will be exploited and states with non-negative M will be considered only.

The second property of S^z is related with symmetry properties of the set of nodes. In a crystal lattice, with the Born-von Kármán conditions imposed on it, symmetry is described by a finite space group [4], Most macromolecular magnets can be considered as planar rings [2], so magnetic ions are placed in vertices of a regular polygon and the symmetry group is isomorphic with a dihedral group D_N. However, the presented method can be applied to any other (finite) symmetry group.

The group G, as the symmetiy group of a finite set (of nodes), is the subgroup of a symmetric group Σ_N containing all $N!$ permutations of nodes. On the other hand, we would like to apply the Schur Lemma, so vector representations of G are necessary. We have to determine a relation between these two descriptions (or aspects) of the symmetry group G.

A homomorphism

$$G \to \Sigma_N : g \mapsto \sigma_g = \begin{pmatrix} 0 & 1 & \ldots & N-1 \\ \sigma_g(0) & \sigma_g(1) & \ldots & \sigma_g(N-1) \end{pmatrix} \tag{5}$$

is the so-called *permutation* representation of G [3], This action can be raised to the set of Ising configurations in the following way

$$P(g)|m_0, m_1, \ldots, m_{N-1}\rangle = |m_{\bar{\sigma}_g(0)}, m_{\bar{\sigma}_g(1)}, \ldots, m_{\bar{\sigma}_g(N-1)}\rangle, \tag{6}$$

where $\bar{\sigma}_g$ is the inverse of σ_g determined in (5). In general, the vector representation $P: g \mapsto P(g)$ is reducible and, therefore, can be decomposed into a direct sum of irreducible representations Γ of G [5]:

$$P = \bigoplus_{\Gamma} n(P, \Gamma)\, \Gamma. \tag{7}$$

Since G is a subgroup of Σ_N, then each $P(g)$ commutes with S and, from the definition, with $\mathcal{H}$. Therefore, the irreducible representations Γ can be used as labels of eigenspaces of $\mathcal{H}$ and our task is to determine, in the most efficient way, the symmetry adapted basis, *i.e.* vectors transforming as the standard vectors of Γ. Before presenting the final formula some combinatorial objects have to be introduced.

An action of G on a set X can be described by (5) or, equivalently, as an external multiplication rule $G \times X \to X$: $(g, x) \mapsto gx$ [3], such that

$$\forall g_1, g_2 \in G, x \in X \qquad g_1(g_2 x) = (g_1 g_2)x\,, \quad 1x = x \tag{8}$$

and $gx = \sigma_g(x)$. This shorter notion will be used below. An *orbit* of an element $x \in X$ *is*

$$G(x) := \{gx \mid g \in G\}\,. \tag{9}$$

The corresponding structure in G, called a *stabilizer* of $x \in X$, is defined as

$$G(x) := \{g \mid gx = x\}\,. \tag{10}$$

Stabilizers of elements $gx \in G(x)$ are conjugates of G_x: $G_{gx} = gG_xg^{-1}$ The crucial point is the following bijection between the orbit $G(x)$ and the set of left cosets of G_x [3]:

$$G(x) \to G/G_x : gx \mapsto gG_x\,.$$

This theorem yields the similarity (a formal definition of a similarity can be found in [3]) between actions of G on $G(x)$ and on G/G_x, so all properties of a given action can be determined investigating actions of G on sets of cosets G/G_x determined as

$$(g, g' G_x) \mapsto (gg')G_x\,.$$

Moreover, it may happen that for some x, x', belonging to different (so disjoint) orbits, the stabilizers G_x and $G_{x'}$ are conjugated. Orbits with this property form a type, called a *stratum,* labeled by a subgroup $U \subset G$ representing a class $\tilde{U}$ of conjugated subgroups in G. Let elements $g_0 = 1$, $g_1, \ldots, g_{m-1}$, $m = |G| / |G_x|$, be representatives of cosets g_jU. Then, elements of each orbit in type $\tilde{U}$ can be labeled by these representatives, *i.e.*

$$G(x) = \{1x, g_1 x, \ldots, g_{m-1} x\}\,.$$

If $|g_j x>$ stands for a magnetic configuration $P(g_j) \mid m_0, m_1, \ldots, m_{N-1}>$ then the representation P can be restricted to a subspace spanned over vectors belonging to a given orbit. A decomposition of this restriction, denoted as $P|_{U,\,x}$, into irreducible representations depends only on an orbit type $\tilde{U}$, so Eq. (7) can be rewritten as

$$P = \bigoplus_{U,x,\Gamma} n(P|_{U,x}, \Gamma)\Gamma\,, \tag{11}$$

where the direct sum runs over all representatives U of classes of subgroups, representatives of orbits $G(x)$ *in* the type $\tilde{U}$, and all irreducible representations of G. It has to be underlined that the frequency $n(P\,|_{U,x}, \Gamma)$ does not depend on x and for a finite group G one can easily determine all possible decompositions and corresponding irreducible bases [6, 7]

$$|\Gamma\alpha\gamma\rangle = \sum_{j=0}^{m-1} a_{j,\alpha\gamma}|g_j x\rangle\,, \tag{12}$$

where $0 \leq \alpha < n(P\,|_U, \Gamma)$ is a repetition index and γ labels vectors of Γ.

The Hamiltonian symmetry group G, as a subgroup of Σ_N, decomposes a set of states (1) into orbits. Moreover, this decomposition can be used for all partitions in the same type and, therefore, the same stabilizers are obtained. Since a stabilizer is related to the decomposition into irreducible representations Γ (11), then it suffices to consider ordered partitions.

3. PROBLEM FORMULATION

Since M is a good quantum number, then the basis consisting of the Ising configurations (1) can be decomposed into subsets indexed by magnetization, a stability group type $\tilde{U}$, and an orbit representative x. Moreover, vectors in each subset $B_{M,U,x}$ are labeled by left coset representatives $g_j j = 0,1,\ldots, m - 1$, where $m = |G|/|U|$ is the index of U in G. The representation P (6), restricted to a subspace spanned over these vectors, decomposes into irreducible representations Γ, according with (11). Vectors of the irreducible basis are linear combinations of all vectors $| M\, U\, x\, j >$ (g_j has been replaced by j) for fixed M, U, and x. It can be shown that a general formula reads [6] ($[\Gamma]$ denotes a dimension of Γ)

$$|MUx\Gamma\alpha\gamma\rangle = \sum_{j=0}^{m-1} \left(\frac{[\Gamma]}{m}\right)^{1/2} . \sum_{\beta=0}^{[\Gamma]-1} \Gamma^*_{\gamma\beta}(g_j)\,(A^{U\Gamma}_{\beta\alpha})^*\,|MUxj\rangle\,, \tag{13}$$

where $A^{U\Gamma}_{\beta\alpha}$ are the so-called reduction coefficients [7, 8], Matrix elements of any operator, *e.g.* $\mathcal{H}$, commuting with all $P(g)$ and S depend only on U, x and α (they do not depend on γ due to the Schur Lemma) and are given by the following formula [6]

$$h_{Ux\alpha,\,Vy\beta} = \left(\frac{|U|}{|V|}\right)^{1/2} \sum_{j=0}^{|G|/|U|-1} \langle MUxj|\mathcal{H}|MVy0\rangle\, B^{UV\Gamma}_{\alpha\beta}(g_j)\,, \tag{14}$$

where

$$B^{UV\Gamma}_{\alpha\beta}(g_j) = \sum_{\mu,\nu=0}^{[\Gamma]-1} (A^{V\Gamma}_{\mu\beta})^*\,\Gamma_{\mu\nu}(g_j)A^{U\Gamma}_{\nu\alpha}\,. \tag{15}$$

Note that the irreducible basis is *not* used in this formula in the explicit way. To prove this equation one needs to introduce the concept of double cosets $U\backslash G/V$ [3, 5]. They appear in the natural way, since one considers matrix elements of an operator $\mathcal{H}$ with a symmetry group G between orbits with symmetry U and V. All values $\langle j|\mathcal{H}|k\rangle$, $j = 0,\ 1,\ \ |G|/|U| - 1$, $k = 0,\ 1,\ \ldots,\ |G|/|V| - 1$, are grouped into sets labeled by double cosets $U\backslash G/V$. The relation presented leads to the following algorithm of the determination of matrix elements:

1. Take representatives y of all orbits (for given M and V).
2. Act on $|M\ Vy\ 0> \ \equiv |y>$ with all terms of the Hamiltonian (2). Note that the diagonal part $s^z_j s^z_k$ of the Hamiltonian leaves $|y>$ unchanged, but multiplies it by a factor $\Sigma_{\langle jk\rangle} J_{jk} m_j m_k$ whereas each term of the off-diagonal term $s^\pm_j s^\mp_k$ leads to only one new configuration $|y')$ (or to zero if *trip* $m_k = \pm s$) multiplied by J_{jk} and factors related with the step operators.
3. For each y', found in the previous step, one has to determine its orbit, *i.e.* a stability group U, an orbit representative x, and such g_j that $g_j x = y'$. Coefficients $B^{UV\Gamma}_{\alpha\beta}(g_j)$ do not depend

on the Hamiltonian and on the actual configurations x, y, y', but are the group-theoretical "parameters" of a considered model.

Eigenproblems of obtained matrices are solved numerically, so this task is out of scope of this paper. For a given symmetry group G the reduction coefficients and matrix elements of irreducible representations can be found in many tables or, in simple cases, their analytical form can be easily determined (in the case of dihedral groups D_N, for example). Having all configurations decomposed into subsets labeled by partitions and stability groups one can construct a simple search algorithm to realize the third step presented above. Therefore, the main combinatorial task is devoted to generation of all configurations, decomposition into orbits, and determination of orbit representatives and their stabilizers. If one is interested in states with some possible values of magnetization M, then the following tasks should be realized:

1. Generate all ordered partitions of N into no more than $n = 2s + 1$ parts.
2. For a given partition $[k]$ generate all configurations.
3. Decompose configurations into orbits of G, determine their stabilizers and choose orbit representatives.
4. For all non-negative 0 or $^1/_2 \le M \le Ns$ find *non-ordered* partitions such that $\Sigma_i k_i(i - s) = M$. Orbits, obtained in the previous step for ordered partitions, are "transformed" for those to non-ordered ones.
5. Construct a search algorithm for efficient determination of a stability group U, an orbit representative x, and a label g_j in the orbit $G(x)$ for a given configuration $y = g_jx$ with U being a stabilizer or x.

These steps are discussed in the next section. We will focus on the first two steps since the others can be simply realized.

4. COMBINATORIAL ALGORITHMS

4.1. Ordered partitions

The first, and the simplest, step is to generate all ordered partitions of a number N into no more than $n = 2s + 1$ parts. Several algorithms for partition generation are known [9], but they have been constructed to generate *all* ordered partitions. The algorithm presented below is based on this one, which generates partitions in the antilexicographic order, *i.e.* a partition $[k]$ is generated before a partition $[k']$ if a word $k'_0, k'_1, \ldots$ k'_m, precedes lexicographically a word $k_0, k_1 \ldots k_m$. It means that there exists such $0 \le l \le \min(m, m')$ that $k_i = k'_i$ for $i < l$ and $k'_l < k_l$, It seems that this algorithm can not be applied since the lengths of partitions are not ordered. However, the first elements of partitions are ordered. This fact and the notion of associated partitions [3] allow to apply this algorithm in our problem. Formally, the partition $[k']$ *associated* with $[k]$ is determined by the following formula:

$$k_l' := \sum_{i \mid k_i \ge l} 1 \,.$$

Hence, the length of the associated partition is equal to the first part in the original partitions and, therefore, this numbers (lengths) form a nondecreasing series. Summarizing, to generate partitions with no more than n parts one takes partitions with the first part no greater than n and then construct their associates. It is the only problem to "jump into" the standard procedure in the appropriate place, *i.e.* when the first part equal n appears the first time. It can be easily solved since in this case (almost) all parts are equal. If l = N mod n than the l first parts equal $\lceil N/n \rceil$ and the others - $\lfloor N/n \rfloor$. It is worth noting that in the used basic algorithm [9,10] it is easy to construct and print out associated partitions.

4.2. Configurations

The usual combinatorial problems deal with a very large number of configurations, so the standard procedures are devoted *to evaluate a canonic transversal in an economic way* [3, 330]. In the considered problem our needs are different: acting with a given operator on a configuration $|m_0, m_1, ..., m_{N-1})$ one obtains, as a rule, a linear combinations of many states and, in the next step, it has to be determined to which orbits obtained configurations belong. Therefore, we have to generate all of them or to find an efficient way to determine a stability group, an orbit representative, and a label in the orbit. In this work the first solution is assumed. It follows from the fact that we do not need all $(2s + 1)^N$ configurations at the same time, but only those for a given M and a few partitions, then we can store all of them. Summarizing, our task is: for each ordered partitions $[k]$, $k_0 \geq k_1 \geq ... k_{n-1} \geq 0$, we have to generate all configurations with k_i entries $m_j = i - s$.

To begin with we start with n = 2, *i.e.* s = 1/2. All ordered partition have only two parts $N = N_+ + N_-$, where $N_\pm$ denotes a number of spin projections $\pm 1/2$. For a given partition $[N_+, N_-]$ the generation of all configurations is equivalent to the generation of all N_+-element subsets of N-element set. This problem is very well documented in many books on combinatorial algorithms [9,10], Similarly, the generation of configurations for a given ordered partition $[k_0, k_1, ..., k_{n-1}]$ is equivalent to the generation of all decompositions of an N-element set into / subsets N_i with cardinalities k_0, k_1..., k_{l-1} where l is a number of nonzero parts in $[k]$. Transformation from subsets to configurations is very simply:

$$m_j = i - s \quad \text{if} \; j \in N_i \, .$$

There are some algorithms to decompose sets in subsets or blocks with different conditions imposed on obtained parts [9], however they can not be simply applied to our problem. It is also possible to construct an algorithm with a recurrence with respect to number of subsets, but such an algorithm can not be simply realized in all programming languages. In this section we present an algorithm to generate all decompositions of N-element set into subsets with k_0, k_1 ,..., k_{l-1}, $0 \leq l < n$, elements. A program based on this algorithm generates configurations $|m_0 + s, m_1 + s, ..., m_{N-1+} \; _s)$ starting with the configuration

$$|\underbrace{l-1,\ldots,l-1}_{\times k_{l-1}},\ \underbrace{l-2,\ldots,l-2}_{\times k_{l-2}},\ \ldots,\ \underbrace{0,\ldots,0}_{\times k_0}\rangle\,.$$

The last generated configuration is the above one written in the opposite direction, *i.e.*

$$|\underbrace{0,\ldots,0}_{\times k_0},\underbrace{1,\ldots,1}_{\times k_1},\ldots,\underbrace{l-1,\ldots,l-1}_{\times k_{l-1}}\rangle$$

A number of generated configurations is equal to $N!/(k_0!\ k_1!\ \ldots\ k_{n-1}\ !)$, what may serve as the simplest test of the program validity. The algorithm is based on the standard one generating subsets in the lexicographical order, *i.e.* it starts from $\{0,1,\ldots, k-1\}$ and goes through $\{0, 1,\ldots, k-2, k\},\ldots, \{0, 1,\ldots, k\text{-}2,n)$ to $\{n-k,\ldots, n-1\}$. The recurrence is removed introducing a parameter controlling the current level and some arrays with level-dependent parameters.

For a given partition $[k] = k_0, k_1\ldots, k_{l-1}, 0,\ldots, 0]$ a level-dependent number of elements is calculated as

$$NL[i] = \sum_{i'=0}^{i} k_{i'}\,,$$

so this arrays contains the number of elements at the i^{th} level. Starting from $i = l-1$, a k_i-element subset is generated at the i^{th} level and one goes deeper, *i.e.* i is decreased. Of course, one can stop at level $i = 1$, since at level 0 the solution is trivial: there is only one k_0-element subset of k_0-element set. However, the standard procedure [9,10] works on the set $\{0,1, \ldots,N-1\}$, so actual elements available at the i^{th} level are stored in an array $AE[n][N]$, This array is initialized in the following way ($AE[i]$ is the i^{th} row of this array),

$$\begin{aligned}
AE[0] &= [N-k_0, N-k_0+1,\ldots,N-1,*,\ldots,*]\,,\\
AE[1] &= [N-NL[1], N-NL[1]+1,\ldots,N-1,*,\ldots,*]\,,\\
\ldots\,,&\\
AE[i] &= [N-NL[i], N-NL[i]+1,\ldots,N-1,*,\ldots,*]\,,\\
\ldots\,,&\\
AE[l-1] &= [0, 1\ \ldots, N-1]
\end{aligned}$$

(elements denoted by an asterisk "*" are irrelevant). A program can stop in this moment if $l = 1$. In all other cases program generates all k_1-element subsets of the $NL[1]$-element set. The first subset contains elements $0,1,\ldots, k_1-1$. Similar initialization has to be made for all levels, so an array $AA[n][N]$ is introduced with starting values

$$AA[i] = [0, 1, \ldots, k_i - 1, *, \ldots, *]\,.$$

The procedure generates *labels* of nodes, which cany a given, depending on the level i, projection m_j. To construct (and print out) a configuration $CONF[N]$ from these values the following procedure is used (for a spin number s spin projections are limited to the values $-s$, $-s + 1,...,$ s, so, in fact, the array *CONF* contains values $(m_j + s) = 0, 1,..., 2s$):

- at first a spin projection i is determined; there are i, nodes with this projection;
- relative indices of these nodes, *i.e.* reduced to the range $0 \le j < k_i$, are stored in the array *AA:* we need the i^{th} row of this array;
- using the array *AE*, more precisely its i^{th} row, we determine the absolute, *i. e.* in the range $0, 1,..., N- 1$, indices of these nodes;
- at these nodes spin projections have the value i, so the final formula has a form:

$$CONF[AE[i]\,[AA[i][j]]] = i\,. \tag{16}$$

Therefore, the first printed configuration is

$$CONF[AE[i][j]] = CONF[N - NL[i] + j] = i\,,$$

where $j = 0, 1,..., k_i - 1$. When a next subset is generated then the array *AA* is changed and the array *AE* has to be modified accordingly. For example, in the first step $AA[1]$ is modified to $[0, 1, ..., k_1 - 2, k_1]$ and, therefore, $AE[0] = [N - k_0 - 1, N - k_0 + 1, ..., N - 1,*,...,*]$. Note that after generation of a subset at the i^{th} level the array *AE* has to be modified at all levels $i' < i$. Moreover, the array *AA* at these levels is again initialized and one goes back to / = 1. More generally, let us assume that available elements at the i^{th} level, stored in $AE[i]$, are as follows

$$AE[i] = [e_0, e_1, ..., e_{NL[i]-1}, *, ..., *]$$

and after generation at the i^{th} level the $AA[i]$ has a form

$$AA[i] = [a_0, a_1, ..., a_{ki-1}\}, *, ..., *]\,,$$

where $0 \le e_j < N$ and $0 \le a_j < k_i$. The formula (16) yields

$$CONF[e_{aj}] = i \quad \text{for} \quad j = 0, 1, ..., k_i - 1\,.$$

At levels $0 \le i' < i$ elements different from e_{aj} are available and, moreover, in the starting configuration $AE[i - 1]$ contains k_{i-1}, first elements, $AE[i - 2]$ - k_{i-2} next elements, and so on. The row $AE[0]$ contains k_0 last elements, as it was mentioned earlier,

$$AA[i' < i] = [0, 1, ..., k_{i'} - 1, *, ..., *]\,.$$

The generated configurations is printed out and all k_1-element subset of $NL[1]$-element set are generated. After this step, one element in the set N_2 is changed and one again goes to $i = 1$. The procedure stops when all k_{l-1} -element subsets of N-element set are generated. For example, when $N = 4$, $n = 3$, and $[k] = [2, 1, 1]$ this procedure generates all 4!/(2! 1! 1!) = 12 configurations in the following order: |2,1,0,0>, |2, 0,1,0>, |2, 0, 0, 1>, |1,2, 0, 0>, |0, 2, 1, 0>, |0, 2, 0, 1>, |1, 0, 2, 0>, |0,1,2,0>, |0,0,2,1>, |1,0,0,2>, |0,1,0,2>, |0,0,1,2>. It should be underlined that generating subsets at the i^{th} level one needs four parameters: $NL[i]$ - the cardinality of a set, k_i - the cardinality of a subset, *AA[i]* - elements in a subset, and the so-called "moving point", *i.e.* an index of an element which was moved in the previous step [9, 10], so an array $MP[n]$ is also necessary. The last two parameters are modified during the generation.

4.3. Orbits

In the current applications decomposition into orbits is done by "brute force", *i.e.* having all configurations generated one takes the first one, generates its orbit removing generated elements from the set of all configurations and then goes to the next not removed configurations. There are two problems: (i) efficient implementation of the group action what can be easily done for some groups, such as cyclic or dihedral ones; (ii) generation of orbits in a "canonical" way. Both problems are related to the structure of a group in question. Since the results obtained in the first three steps (partitions, configurations, and orbits) will be used for different values of magnetization, one has to construct algorithms for transforming obtained results from ordered to non-ordered partitions and for searching configurations obtained by the action of,$s_j^{\pm} s_k^{\mp}$.

One of possible ways is to choose generators of *G,* express all elements as products of generators, and to determine action of generators on configurations. In the case of the dihedral group D_N there are two generators: the *N*-fold rotation C_N and one of two-fold rotations, U_0 say. Other elements are expressed as C^J_N and $C^J_N U_0$. In such a simple case it is also possible to determine action of these elements in a direct way. To solve the second problem, one has to fix representatives *U* of classes $\tilde{U}$ of conjugate subgroups. For each *U* the group *G* is decomposed into left cosets *gjU,* and the choice of representatives is also fixed. Moreover, the order of these representatives is also fixed. This order determines the order of configurations in a given orbit (in a type $\tilde{U}$). The reduction coefficients $A^{U\Gamma}_{\gamma\alpha}$ have to be determined, too.

Taking into account the above notes, the decomposition into orbits is done in the following way:

1. Knowing number of all configurations *NC* a boolean array *NU*[*NC*] is initialized to *NU*[*j*] = TRUE.
2. For k = 0 to $k = NC$ - 1 the procedure is started if *NU*[*k*] = TRUE.
3. The stability group G_k of the k^{th} configuration is determined, at first.
4. If G_k is one of the chosen *U's* then the generation of an orbit is started. Otherwise, the next configuration is tested.
5. The first element of this orbit is, of course, the k^{th} configuration, so *NU*[*k*] = FALSE. For all representatives of left cosets G/G_k other elements of the orbit are generated, printed out, and searched in the set of all configurations.
6. During the search, especially when comparison of configurations may take a lot of time, one can use the array *NU.* Of course, for all configurations found the appropriate entry in the array *NU* is changed.
7. Files, the orbits are written to, should be constructed in such a way that it will be easy to determine the ordered partition and the stability group of configurations. It can be done by a proper naming convention, for example.

4.4. Non-ordered partitions

In the previous sections the results for all *ordered* partitions [k] are obtained. Considering a given value *M* of the magnetization one has to consider, in general, non-ordered partitions [k']

in the type $[k]$, *i.e.* $[k']$ contains the same parts k'_i, as *[k]* does, but they are not ordered. Therefore, there exists at least one permutation σ of indices $0 \leq i < n$ such that $k_i = k'_{\sigma(i)}$. All orbits obtained above are transformed into new orbits (with a given *M)* according with this bijection, *i.e.* all entries *i* are replaced by *σi*. In the example presented in the previous section the partition [2, 1, 1] has been considered. This partition corresponds to $M = -1$ since one has $2 \cdot (-1) + 1 \cdot 0 + 1 \cdot 1$. If configurations with $M = 0$ are needed then one has to generate, among others, configurations corresponding to the partition [1,2,1], The results are obtained after the bijection 0 ↔ 1, so the configurations (before the decomposition into orbits) are as follows: |2, 0,1,1>, |2, 1,0,1>, |2, 1,1,0>, |0,2, 1, 1>, |1, 2, 0, 1>, |1,2, 1, 0>, |0, 1, 2, 1>, |1, 0, 2, 1>, |1, 1,2, 0>, |0, 1, 1, 2>, |1,0,1,2>, |1,1,0,2>.

4.5. Matrix elements

Determination of matrix elements of bilinear spin operators, like Hamiltonian or $\mathbf{S}^2$, can be decomposed into two parts. The easiest task is to calculate the diagonal (or "secular") part, *i.e.* corresponding to terms $s^z_j s^z_k$ since the Ising configurations are their eigenstates. The other terms, $s^+_j s^-_k$, transform a given orbit representative to another state, in general an element of another orbit with a different group stabilizer. In the first step one has to determine a stabilizer of a configuration $s^+_j s^-_k$ *|Mvy0>, i.e. U* in Eq. (14). Next the pair of indices *xj, i.e.* an orbit representative and a label within its orbit, has to be determined. The searching procedure can be simplified when one takes into account a relation between partitions $[k]$ and $[k']$ of the starting and resulting configurations. At first, we note that for a given magnetization Meach partition $[k]$ of *N* into $n = 2s + 1$ non-negative parts is determined by $n - 2 = 2s - 1$ numbers k_i, since all parts have to satisfy two (linear) conditions (4). Therefore, one has to "observe" changes in chosen numbers k_j. For example, when $s = 1/2$, then no change in those numbers is admissible, so a partition ss not modified. In the case $s = 1$ one can choose k_1 to be an indexing number and a partition type is changed if and only if a number of projections $m_j = 0$ is affected, *i.e.* when $s^+_j s^-_k$ acts on a configuration with $m_j = -1$ **or** $m_k = 0$. Of course, due to the hermiticity of a Hamiltonian this relation is symmetric. Since during the generation process configurations are decomposed into subsets with respect to stabilizers **and** partitions, then it is convenient to determine the resultant partition $[k']$ and search for the obtained configuration only in this subset. For example, when $s = 1$ and partitions are labeled by k_1, then, in a general case, there are only three possibilities: $k'_1 = k_1$ or $k'_1 = k_1 \pm 2$.

5. FINAL REMARKS

In this work determination of matrix elements of quantum operators for finite systems has been presented. The two main combinatorics problems, generation of ordered partitions and of configurations, are realized by modification of the standard algorithms. The other tasks, especially decomposition of the Ising configurations into orbits and calculation of matrix elements, are performed in the simplest possible way. This solution follows from the fact that the efficiency of the proposed method depends on the first two steps mainly (and on implementation of the group action). The other steps can be improved, and it has been done, by standard methods of storing symmetric matrices, multi-spin coding, searching and ordering of sets *etc.* Programs based on the proposed algorithms were written in FORTRAN (F77) at first and then they have been rewritten in C. They have been compiled and tested under IRIX 6.5 and Linux 2.2 (the Red Hat 6.2 distribution). More detailed discussion is presented in the next paper [12].

In the case of isotropic Heisenberg Hamiltonians one can use the total spin as a good quantum number. To exploit possibility of introducing this additional label of states a set of homogeneous linear equations has to be solved. A matrix of this set can be obtained by the method presented here due to the relation (3). However, even for quite small molecules a dimension of such matrices is large, *e.g.* up to several thousands for twelve spins $s = 1$. Many standard procedures for solving a set of linear equations are unstable in such cases [11], This problem can be solved in some cases by application of libraries with arbitrary precision arithmetics such as the GNU Multiple Precision Arithmetic Library (GMP or GNU MP in short) written and developed by T. Granlund [13]. It is especially useful in investigations of one-dimensional irreducible representations *(e.g.* when one considers the ground state of bipartite antiferromagnets), because all matrix elements can be expressed as $p\sqrt{q}$, where both p and q are integers. Moreover, all q's are divisors of the number of nodes N, so only a few different square roots are necessary. Using the GMP, or any similar library, it is easy to construct procedures realizing the arithmetics for numbers in the form $\Sigma_i\, p_i\sqrt{q_i}$. There are some other simplifications of these procedures in application to the mentioned problem, but their presentation goes very far beyond the scope of this paper and, moreover, they have not been completely documented yet.

Acknowledgments

This work is supported by the Polish State Committee for Scientific Research (KBN) within the project No. 2 P03B 074 19. The author is indebted to Prof. G. Kamieniarz for inspiring and valuable discussions. The presented method and algorithms have been tested on computers in the Supercomputing and Networking Center in Poznań.

References

[1] J. C. Bonner, M. E. Fisher, Phys. Rev., **135**, A640 (1964).

[2] D. Gatteschi, A. Caneschi, L. Pardi, R. Sessoli, Science, **265**, 1054 (1994); A. Lascialfari, D. Gatteschi, A. Cornia, U. Balucani, M. G. Pini, A. Rettori, Phys. Rev. B. **57**, 1115 (1998); A. Caneschi, D. Gatteschi, C. Sangregorio, R. Sessoli, L. Sorace, A. Cornia, M. A. Novak, C. Paulsen, W. Wernsdorfer, J. Mag. Mag. Mater., **200**, 182 (1999).

[3] A. Kerber, *Algebraic Combinatorics via Finite Group Actions,* BI Wissenshaftsverlag, Mannheim-Wien-Zürich, 1991.

[4] W. Florek, T. Lulek, J. Phys. A: Math. Gen., **26**, 2153 (1993).

[5] S. L. Altmann, *Induced Representations in Crystals and Molecules,* Academic Press, London, 1977.

[6] G. Kamieniarz, R. Matysiak, W. Florek, S. Wałcerz, J. Mag. Mag. Mater., **203**, 271 (1999).

[7] W. Florek, Acta Phys. Polon. A, **96**, 699 (1999).

[8] B. Lulek, T. Lulek, J. Phys. A: Math. Gen., **17**, 3077 (1984).

[9] E. N. Reingold, J. Nivergelt, N. Deo, *Combinatorial Algorithms. Theory and Practice,* Prentice Hall, Engelwood Cliffs NJ, 1977.

[10] W. Lipski, *Combinatorics for Programmers.* Polish Sci. Publ. (PWN), Warsaw, 1982 [in Polish].

[11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C. The Art of Scientific Computing,* Cambridge Univ. Press, Cambridge, 1992 (Sec. 2.0).

[12] W. Florek, Comp. Meth. Sci. Tech., this volume.

[13] The GNU MP library has been developed by T. Granlund and now is a standard part in main Linux distributions (it is used to compile SSH2). It is also available at its home page *http://www.swox.com/gmp* maintained by T. Granlund.