Visualisation of many-particle model spaces with application to the shell-model calculations

1987 J. Phys. A: Math. Gen. 20 1633

(http://iopscience.iop.org/0305-4470/20/7/010)

J. Phys. A: Math. Gen. **20** (1987) 1633-1643. Printed in the UK

Visualisation of many-particle model spaces with application to the shell-model calculations

Włodzisław Duch†

Institut für Astrophysik, Max-Planck-Institut für Physik und Astrophysik, Karl-Schwarzschild-Strasse 1, 8046 Garching bei München, West Germany

Received 16 April 1986

Abstract. The bases of model spaces used in many-body theory may be visualised using graphs. The graphs allow a deep insight into the structure of these spaces and facilitate rapid evaluation of matrix elements, making shell-model calculations in the spaces of 10^6 dimensions quite practical. Four examples of graphs are given: representing the spin space, the space of symmetry-adapted configurations of a molecule, representing $\hat{S}^2$ (or $\hat{T}^2$) eigenstates, and representing the space of $(\hat{J}^2, \hat{T}^2)$ eigenstates, including their genealogy. Questions related to the computer analysis of such graphs are discussed and different approaches compared in a practical test. Advantages of the graphical representation of model spaces are summarised.

1. Introduction

Almost every textbook on quantum mechanics starts from the concept of Hilbert spaces, or linear spaces in general. The many-body theory, concerned with the solution of certain equations that are defined in such spaces, has developed from diagrammatic perturbation theory used in quantum electrodynamics. Thanks to Feynman diagrams we know now how to represent and interpret different interactions described by perturbation expansion (cf the marvellous book by Mattuck (1976)). However, no attempt has been made to visualise the many-particle spaces in which all the action is taking place. In this paper I want to point out that such a visualisation is possible and desirable, giving us—similar to the case of many-body diagrammatic approaches—the language and the method of calculations.

Many-particle model spaces appear in the nuclear, atomic and molecular physics. While in molecular or atomic problems Hartree-Fock (or multiconfigurational HF) wavefunctions give a good approximation and allow for a meaningful truncation of the full space of N-particle functions that may be constructed from n orbitals (single-particle functions), such is usually not the case in nuclear problems, where harmonic oscillator wavefunctions are, almost exclusively, taken as orbits (cf Wong 1981). The full spaces have very high dimensions and therefore shell-model calculations in nuclear physics are concentrated mainly in the sd shell, with only modest studies of other shells (McGrory and Wildenthal 1980). The three general purpose computer programs for shell-model calculations used at present were written in the 1960s (Cohen *et al* 1966, French *et al* 1969) or the beginning of the 1970s (Whitehead 1972), with progress in the last decade coming mainly from the development of computer technology. In

† Alexander-von-Humboldt Fellow 1985/86. Permanent address: Instytut Fizyki UMK, ul Grudziądzka 5, 87-100 Toruń, Poland.

0305-4470/87/071633+11$02.50

contrast to nuclear shell-model codes the development of computer programs in molecular physics has been particularly rapid in the last ten years (cf the review by Duch and Karwowski (1985)). The number of basis functions used in configuration interaction calculations has exceeded 10^6 (Saxe *et al* 1982) and, although general programs are most effective for singly or doubly excited configurations out of a set of reference configurations, calculations in the full space of a similar size were also reported for a few molecules (Harrison and Handy 1983). The comparatively slower development of atomic and nuclear physics programs may be only partially due to the complications arising from higher symmetry: the $(\hat{\boldsymbol{J}}^2, \hat{\boldsymbol{T}}^2)$ eigenstates were used only in the Rochester-Oak Ridge code (French *et al* 1969), the two other programs being based on determinants (M scheme). Molecular programs, which work in very large many-particle spaces, do not create the Hamiltonian matrix but rather, using the list of one- and two-particle integrals, recreate matrix elements when they are needed—a similar philosophy lies behind the Glasgow code (Whitehead *et al* 1977). However, the molecular programs do recognise that there are many matrix elements with exactly the same value and these are calculated only once; sometimes the elements differ only in the integrals but have the same coupling coefficients. This is a reflection of the structure of the space, i.e. the way the basis states of this space are constructed and mutually related. It was much easier to notice and use this structure in the case of spaces composed of singly and doubly excited configurations than for the full spaces, although now the structure of determinantal spaces and spin-eigenfunction spaces is known in detail (Duch 1985a, Duch and Karwowski 1985). The recognition of this structure and the insight into the structure of matrices representing operators in the corresponding spaces was made possible by the graphical representation of these spaces. Graphical visualisation of distinct rows in Paldus tableaux (Shavitt 1981, 1983) and subsequent visualisation of the configuration spaces (Duch and Karwowski 1981, 1982) was followed by graphical visualisations of many different model spaces (Duch 1985a, 1986b).

In the next section, after a short introduction of basic concepts, four examples of graphs are presented: the well known branching diagram representing the spin space helps to introduce main concepts, the second graph describes $S = 1$ (or $T = 1$) space for six particles in six orbits, the third graph describes full space of symmetry-adapted configurations for water in a double-zeta basis, and the last represents $J = 2$, $T = 0$ space for two neutrons and two protons in $(s_{1/2} + d_{3/2} + d_{5/2})$ shells. Computer aspects of such a representation are addressed in the next section, with various strategies of using the graphs in practical calculations described. In the final section advantages of the graphical representation of model spaces (or GRMS) are summarised.

2. Examples of a graphical visualisation of model spaces

Graphical representation is introduced here using a simple example (cf also Duch 1985a). Many-particle states are built from simpler objects possessing some symmetry. We may start from a set of one- or two-particle states—let us call them primitive states or just primitives. They are the 'building bricks' of the N-particle states $|\Phi\rangle$. Many-particle states fulfil certain symmetry requirements: permutational symmetry (usually antisymmetry), spatial symmetry ($|\Phi\rangle$ should transform as a basis of a representation of the spatial symmetry group of the system) or spin symmetry ($|\Phi\rangle$ should be the eigenfunction of $\hat{\boldsymbol{S}}_z$, $\hat{\boldsymbol{S}}^2$ or $\hat{\boldsymbol{J}}^2$ operators). From the group-theoretical point of view we

want to construct, using products of primitive states, a tensor product carrier space that carries the representations of various symmetry groups of a system. Eigenstates of $\{\hat{\mathcal{O}}_k\}$ operators will be called $\{\hat{\mathcal{O}}_k\}$-adapted states and the space of such eigenstates $\{\hat{\mathcal{O}}_k\}$-adapted space. In general, if the states transform as a representation of a certain symmetry group Γ, we call these states Γ-adapted and the corresponding graph Γ-adapted. The question that arises now is: how can such a symmetry-adapted space be visualised?

The simplest example one can find is the spin space. Here we have two kinds of primitive objects: spin-up $|\uparrow\rangle$ and spin-down $|\downarrow\rangle$ states. The operators that 'shape' our space (in the sense that graphs adapted to different operators look different) are $\hat{S}_z$ and $\hat{S}^2$ operators. In figure 1 we see the representation of the $\hat{S}_z$- and $\hat{S}^2$-adapted N-particle spaces in the form of the well known spin diagrams (cf Pauncz 1979). These graphs represent the whole spin function spaces. Each path in the graph starts from the $(0, 0)$ vertex and is composed of N segments (arcs), the last one reaching the (N, S) vertex; a path corresponds uniquely to one N-particle spin function. Near each vertex the total number of paths reaching it is written. Spin paths in the branching diagram give 'proper' labels, in the sense that they contain all the information necessary to calculate matrix elements needed in the many-body calculations. On the other hand, listing the quantum numbers of a state does not give that information unless the whole genealogy is specified. Therefore a properly constructed graph should contain the genealogy of all the states within it. To calculate the matrix element $\langle L|\boldsymbol{A}|R\rangle$ we simply have to compare the paths corresponding to the $|L\rangle$ and $|R\rangle$ states. In case of $(\hat{S}_z, \hat{S}^2)$-adapted spin spaces each operator is expressed as a combination of the permutation operators. Very efficient methods of matrix element calculation in the spin spaces were recently published (Duch 1985b, 1986, Rettrup 1986).

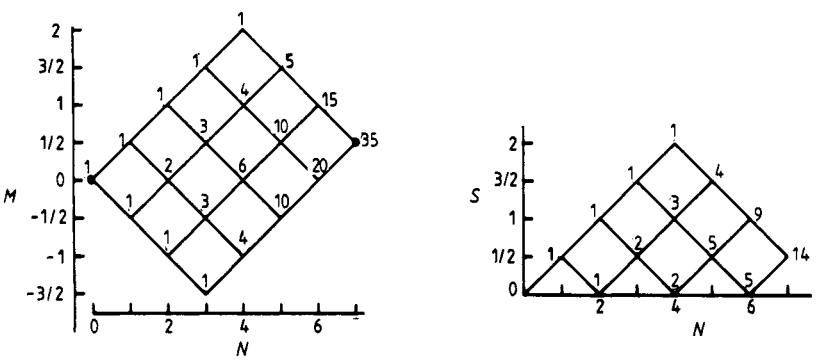

Figure 1. (*a*) 35-dimensional space of $\hat{S}_z$ spin eigenfunctions for seven spins and $M_S = \frac{1}{2}$. (*b*) 14-dimensional space of $\hat{S}^2$ spin eigenfunctions (branching diagram) for seven spins and $S = \frac{1}{2}$.

In the second example let us take $|k\sigma\rangle = |k\uparrow\rangle$ or $|k\downarrow\rangle$ one-particle states, $k = 1, 2, \ldots, n$, as our primitives, where $|k\uparrow\rangle = |k\rangle|\uparrow\rangle$ is a product of a spatial and a spin (isospin) state. We can visualise the resulting space in a few different ways (Duch 1985a, 1986b), for example using a graph shown in figure 2. This graph is similar to the four-slope graph of Shavitt (1983), but instead of n it has $2n$ levels and therefore is more legible. A short vertical line (called an arc) joining two vertices represents an unoccupied primitive state, while two kinds of skew lines are used to differentiate between occupied $|k\uparrow\rangle$ and $|k\downarrow\rangle$ primitives. The position of the vertices within the

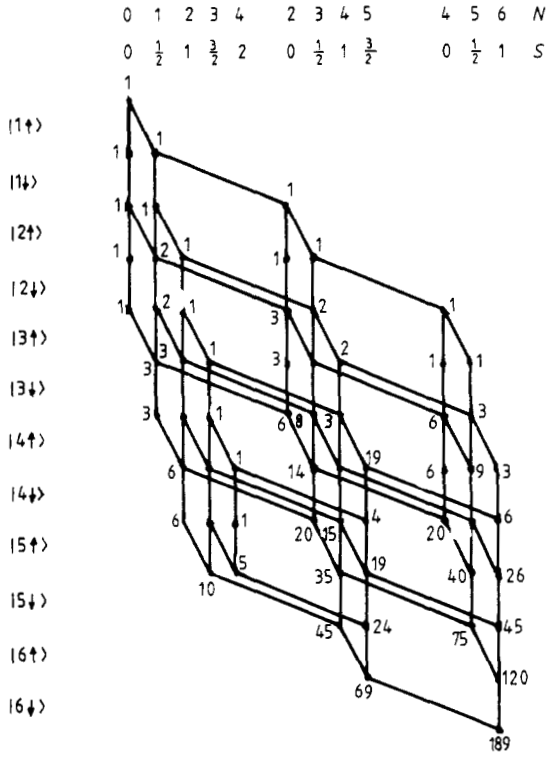

Figure 2. 189-dimensional space of the six-particle functions coupled to the $S = 1$ eigenstate built from eight primitive $|k\uparrow\rangle$ and $|k\downarrow\rangle$ states, visualised using a simple two-slope graph with different slopes of arcs for $|k\uparrow\rangle$ and $|k\downarrow\rangle$ states.

graph is characterised now by three numbers: k, N and S. A path is a collection of $2n$ (12 in this case) arcs, starting from the top arc ($k = 1\uparrow$) to the bottom ($k = n\downarrow$) and represents the six-particle state coupled to $S = 1$. Again, comparing $|L\rangle$ and $|R\rangle$ paths in the graph it is easy to calculate the matrix element of an operator between the two corresponding states.

In the examples discussed above each path of the graph corresponds uniquely to the one of the basis functions. One- and two-electron operators that do not change the number of particles are expressed using the shift operators E_{ij} (Bohr and Mottelson 1969) in the following way:

$$\hat{H} = \sum_{ij}^{n} \langle i|\hat{h}_1|j\rangle E_{ij} + \tfrac{1}{2}\sum_{ijkl}^{n} \langle ij|\hat{h}_2|kl\rangle (E_{ij}E_{kl} - \delta_{jk}E_{il}) + \sum_{ij}^{n} \sum_{\sigma\tau=0,1} \langle i\sigma|\hat{h}_s|j\tau\rangle E_{i\sigma,j\tau} \qquad (2.1)$$

where $\hat{h}_1$, $\hat{h}_2$ are the spin-independent one- and two-particle operators, and $\hat{h}_s$ is the spin-dependent one-particle operator. Similar expressions are obtained using spherical tensor operators. Acting with E_{ij} on a function represented by the path $|R\rangle$ in the graph it is easy to find all the paths $|L\rangle$ giving $\langle L|E_{ij}|R\rangle \neq 0$. The value of this matrix element is given by a product of coefficients assigned to the arcs k, for $i \leqslant k \leqslant j$, belonging to the two paths $|L\rangle$, $|R\rangle$ (cf Shavitt (1981), Duch (1986b) for spin eigenfunctions or Duch (1985a) for determinants).

Another possibility is to represent linear subspaces rather than individual states. In molecules such as water the symmetry group is non-degenerate and each orbital is occupied at most twice. A configuration, like $1a_1^2 2a_1^2 3a_1 4a_1 1b_2 2b_2 3b_2 4b_2$ represents a

linear subspace, in this case (six singly occupied orbitals) corresponding, for example, to five singlet or nine triplet spin eigenstates, or 20 determinants with $M = 0$, or 15 determinants with $M = 1$, etc. In figure 3 we see a representation of a space of orbital configurations of water, a ten-electron system with 14 orbitals (double-zeta basis). There are 76 670 configurations of A_1 symmetry, corresponding to 256 473 $^1A_1$ states. To obtain configurations of the desired symmetry only, some of the vertices were removed from the graph and the orbitals were ordered according to their symmetry. The same graph without removed vertices has 270 270 paths corresponding to 1002 001 singlet functions. The power of the graphical representation is clearly seen in this figure: although the dimension of space is very large the graph is small. The 'bookkeeping problem', a problem in traditional approaches (cf Cohen *et al* 1966, Wong 1981),

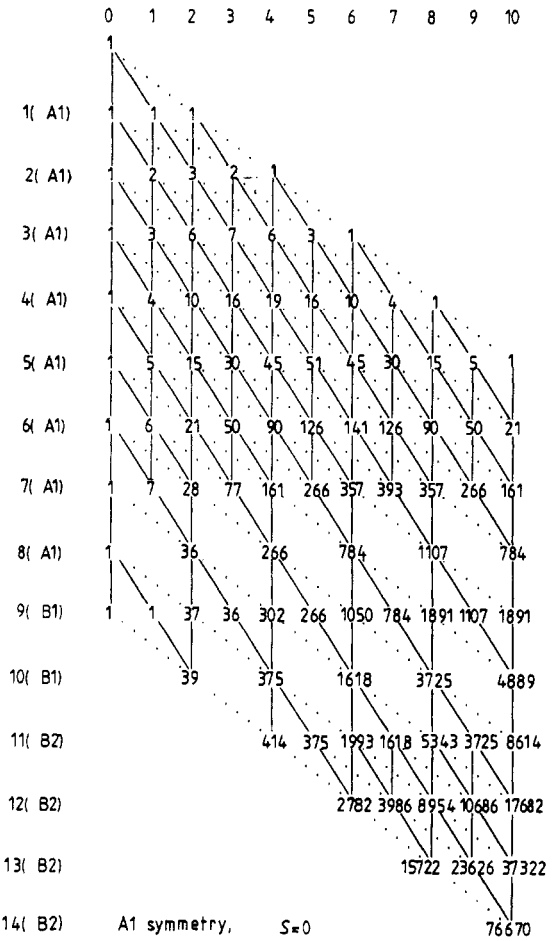

Figure 3. Configurations of water in double-zeta basis and A_1 symmetry. This picture comes as a part of the output from a computer program. For 0 singles, 1 spinfunction, 2002 configurations and 2002 states. For 2 singles, 1 spinfunction, 17 325 configurations and 17 325 states. For 4 singles, 2 spinfunctions, 32 760 configurations and 65 520 states. For 6 singles, 5 spinfunctions, 20 020 configurations and 100 100 states. For 8 singles, 14 spinfunctions, 4290 configurations and 60 060 states. For 10 singles, 42 spinfunctions, 273 configurations and 11 466 states. Total number of A_1 configurations: 76 670. Total number of $S = 0$ states: 256 473.

simply vanishes. Consider now the shift operator $E_{4b_2,3b_2}$ acting on one of the 10 686 configurations whose paths cross the (12, 9) vertex (figure 3) and have the form $\ldots 3b_2^1 4b_2^0$. Then all the 10 686 matrices

$$\langle L|E_{ij}|R\rangle = \langle \ldots 3b_2^0 4b_2^1|E_{4b_2,3b_2}|\ldots 3b_2^1 4b_2^0\rangle \tag{2.2}$$

have the same value (in this case it is a unit matrix of a dimension corresponding to the linear subspace represented by $|L\rangle$ configuration) and the lexical indices corresponding to the $|L\rangle$ and $|R\rangle$ paths in a standard order (cf Duch and Karwowski 1985) are immediately obtained from the graph as: number $L = 37\,322 + K$, number $R = 17\,682 + K$, $K = 1, 2, \ldots, 10\,686$. In the space represented in figure 3 among all 1.15 million matrices $\langle L|H_1|R\rangle$ containing for singlet eigenfunctions 32 million matrix elements (i.e. elements between individual spin eigenfunctions) there are only 432 different matrices. It is not much harder to find an arbitrary contribution from a product of two shift operators.

Our final example, in figure 4, shows the often used basis $(d_{5/2}+s_{1/2}+d_{3/2})$ for the case of two neutrons and two protons, with four-particle states coupled to $J = 2$, $T = 0$. The graph is complicated because each vertex is characterised by the number of particles plus intermediate J and T values. To make the graph more legible the T values are shown only for $N = 2$. The labelling of the $\hat{J}^2$ eigenfunctions always presents a problem in traditional schemes; graphical labelling, showing the whole genealogy, could make all redundancy labels spurious. Unfortunately it would also make the graph much more complicated. The paths in figure 4 do not give proper labels of the states they represent because no attempt has been made to present the parentage within the equivalent groups of particles. Therefore only the mixed groups have proper genealogy.

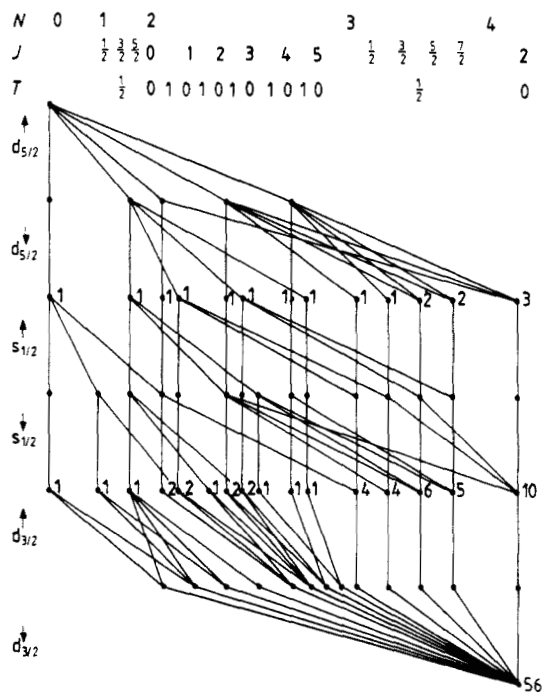

Figure 4. $J = 2$, $T = 0$ states of two neutrons and two protons in $(d_{5/2}+s_{1/2}+d_{3/2})$ basis.

In the case when ambiguity could arise different intermediate couplings, arising for mixed configurations, are shown in the middle of a level to give genealogy of the states. To obtain proper labels supplementary graphs showing the couplings within equivalent configurations ($s^r_{1/2}$, $d^r_{3/2}$, $d^r_{5/2}$, $r = 2, 3, 4$ in this case) are needed. Because these type of graphs are quite complicated, to simplify the analysis of the corresponding spaces we can draw a graph with configurations (there are only 14 in this case) and then resolve separately each subspace corresponding to a configuration into ($\hat{J}^2$, $\hat{T}^2$) eigenstate graphs.

3 Computer-related problems and the strategy of computations

Representation of graphs in a computer does not present any problems and has been a favourite subject of computer scientists for a long time. Representation of graphs with fixed number of slopes at each level is particularly simple (cf Duch and Karwowski 1985, Duch 1985a). The graph of figure 3, for example, is represented using only 330 numbers. The crucial part in the analysis of the graph is the search for pairs of $|L\rangle$, $|R\rangle$ paths. There are many searching algorithms that may be used for this problem. Using the three-slope graph of figure 3 as an example I tested several versions of depth search (DS) algorithms: a simple DS algorithm, DS modified to take advantage of the simple structure of the graph near its borders (when $(k, N-1)$ vertex is reached the remaining arc is on one of $n-k$ levels and these $n-k$ paths form a 'ladder' in the graph that is easily analysed explicitly; similarly for $(k, N-2)$ vertex), DS with fixed number of singly occupied arcs (Duch 1985a). I also tested a global algorithm (cf Duch 1986a) and breadth search algorithm (as described in Duch and Karwowski 1985). For full spaces with a rather small number of orbits all these algorithms are about equally efficient, except that the breadth search and the global algorithms become impractical for large spaces, requiring too much memory. For spaces with a large number of orbits and only few particles the modified depth search algorithm is preferable. However, the best approach seems to be the cutting of a graph at a level where the total number of paths coming from the top is roughly equal to the number of paths coming from the bottom. For each of the vertices at such level the analysis is done separately in the upper part and in the lower part and the results are combined. Using this approach on a scalar computer (Siemens 7880), the 270 270 paths corresponding to the graph in figure 3 without removed vertices were enumerated in 0.34 s, while on a Cray XMP (single processor) it took 0.06 s. The corresponding times for other algorithms were at least an order of magnitude longer. Although the searching algorithms are hard to vectorise because of non-linear indexing and recursive nature of searching, breaking the graph into two parts leaves most of the work for the loops connecting the two parts and makes vectorisation possible. With this approach (it may be pictured as taking the square root of a graph) the logical part of the large scale calculations should take a rather insignificant part of the total time.

To avoid storage problems with a large number of matrix elements Lanczos' scheme is usually advocated (Whitehead *et al* 1977). When the diagonal matrix elements are dominant Davidson's modification of Lanczos' algorithm gives faster convergence (Davidson 1975). There are a few general strategies one may adopt writing a general program for shell-model calculations. The simplest approach is to use the M scheme, or determinants with a proper M_J value. The graph of orbital configurations (cf figure 3) may be readily used in such calculations: the elements of the shift operators in the

determinantal basis are obtained at almost no cost simply by taking them from a small table (Duch 1986c). Another graphical description of determinantal spaces may also be readily employed. The structure of the corresponding spaces is known in all details. The second strategy is the use of the coupled scheme, i.e. full use of the symmetry. The structure of the $(\hat{\boldsymbol{J}}^2, \hat{\boldsymbol{T}}^2)$-adapted spaces is rather complicated (cf figure 4) and the graphical rules for calculation of matrix elements so far are not worked out. The existence of equivalent electrons makes the orbital angular momentum eigenfunctions much more complicated than the spin functions—the results for equivalent electrons have to be 'factored out' and added separately to the graphical rules. Such rules should always be useful, but it is not clear how much there is to be gained from analysis of the structure of $(\hat{\boldsymbol{J}}^2, \hat{\boldsymbol{T}}^2)$-adapted graphs in view of their complication.

Between these two extreme approaches there is a range of possibilities so far little explored. Adding the Abelian symmetry subgroup Γ of a full rotational group reduces the dimension of the space significantly without introducing any essential complications. Using the T scheme is equivalent to the use of $\hat{\boldsymbol{S}}^2$-adapted graphs and the structure of such graphs is known in detail (cf Duch and Karwowski 1982, 1985). The programs based on these graphs, developed at present for molecular calculations, may be adapted to nuclear problems as well. Calculating elements of matrices between two configurations of $(\hat{\boldsymbol{J}}_z, \Gamma, \hat{\boldsymbol{T}}^2)$-adapted graph we may transform these matrices to the $(\hat{\boldsymbol{J}}^2, \hat{\boldsymbol{T}}^2)$-adapted bases, while still being able to use the simpler structure of the $(\hat{\boldsymbol{J}}_z, \Gamma, \hat{\boldsymbol{T}}^2)$-adapted graph. It is enough to know how $(\hat{\boldsymbol{J}}_z, \Gamma)$-adapted configurations are resolved into $\hat{\boldsymbol{J}}^2$-adapted components to find the transformation matrix. Thus if $A_{LR} = \langle L|\hat{A}|R\rangle$ is a matrix between two configurations $|L\rangle$, $|R\rangle$ corresponding to d_M functions of the $(\hat{\boldsymbol{J}}_z, \Gamma)$ space that are resolved into $d_M = d_J + d_{J'} + \ldots$ functions in the $\hat{\boldsymbol{J}}^2$ space, $\tilde{A}_{LR} = \boldsymbol{U}^\dagger A_{LR} \boldsymbol{U}$ is defined in the $(\hat{\boldsymbol{J}}_z, \Gamma, \hat{\boldsymbol{T}}^2)$ space, where $\boldsymbol{U}$ is the rectangular $d_M \times d_J$ transformation matrix. If the Lanczos' algorithm is used and $\tilde{A}_{LR}$ does not appear many time coefficients of the trial wavefunction may be multiplied directly by $\boldsymbol{U}$, A_{LR} and $\boldsymbol{U}^\dagger$, instead of performing the two matrix multiplications (of A_{LR} by $\boldsymbol{U}$ and $\boldsymbol{U}^\dagger$). Because the number of different types of configurations is rather limited the number of transformation matrices, obtained by analytical methods or by a direct diagonalisation of the matrix of $\hat{\boldsymbol{J}}^2$ operator in the $(\hat{\boldsymbol{J}}_z, \Gamma)$-adapted basis, should also be limited. The transformation method is particularly simple when a reduction from $\hat{\boldsymbol{S}}_z$- to $\hat{\boldsymbol{S}}^2$-adapted space is made, because the transformation matrices depend only on the desired multiplicity of the $\hat{\boldsymbol{S}}^2$ eigenfunctions and on the number of singly occupied orbitals (Duch 1986c).

4. Summary

The main points of the graphical representation of model spaces may be summarised as follows.

(1) Spaces of a very high dimension may be represented by small graphs. Each path in a graph corresponds to a one- or more-dimensional subspace of the large space.

(2) Hierarchical representation of spaces sometimes simplifies their visualisation: for example, the $(\hat{\boldsymbol{L}}_z, \hat{\boldsymbol{S}}^2)$-adapted space may first be visualised using an $\hat{\boldsymbol{L}}_z$-adapted n-level graph, each path corresponding to a configuration labelling a subspace of $\hat{\boldsymbol{S}}^2$-adapted states.

(3) Graphs may have different topologically equivalent representations, the simplest being the fixed-slope graphs, like the two-slope graphs of figure 1 or the

variable-slope graphs, the slope at each level depending on the type of its 'building brick', i.e. primitive state, like in the graph of figure 2. Reordering of the levels leads to a new representation of the same space, the new graph being in general not equivalent topologically to the former one.

(4) The calculation of matrix elements does not require the knowledge of the explicit structure of functions represented by a graph. As long as one can construct a 'proper' label all the information necessary for matrix element evaluation may be included in the graph. Thus the graph carries all geometrical or structure information, while all physics is contained in the one- and two-body integrals.

(5) Graphical representation makes the structure of the whole space clearly visible, and the structure of this space is reflected in the structure of the matrices representing operators acting in the space described by a graph. This has great practical value when computer programs are constructed (cf Robb and Niazi 1984, Duch and Karwowski 1985).

(6) In molecular or atomic physics one may develop intuitions based on the shape of the graphs, recognising the relatively unimportant parts of the graph that may be removed reducing the dimension of the whole space without significant loss in accuracy. Selection of configurations is of prime importance in practical applications in view of the rather large one-particle basis sets needed for an accurate description of electron correlation in atoms and molecules and the huge dimensions of the resulting many-particle spaces.

(7) The idea of a graphical representation of model spaces is theory independent; so far graphs were used to describe the bases of the unitary group $U(n)$ and the symmetric group S_N, bases adapted to the $\hat{S}^2$ operator, in a theory-dependent context. This point should be emphasised because there seems to be a lack of understanding of the simple fact that graphs represent spaces, not only 'distinct rows in Paldus tableaux' or likewise theoretical constructs (cf Esser 1984). The excellent results achieved with a four-slope n-level $\hat{S}^2$-adapted graph (cf Shavitt 1983) that was introduced to describe the Gel'fand basis of the $U(n)$ group, are clearly due to the insight gained from the visualisation of the space rather than to the clever way of matrix element calculation based on the unitary group theory. The same expressions for matrix elements may easily be obtained without any reference to the unitary group (Duch 1986b).

(8) While Feynman-type diagrams are natural in many-body methods based on perturbation expansion the graphical representation of model spaces describes these spaces globally and is therefore more natural in variational approaches, although it should also be useful for calculation of matrix elements in perturbation theory.

(9) The labelling of atomic and nuclear states (or other states arising from degenerate point groups) is properly done using complicated chains of subgroups. The use of artificial quantum numbers like seniority is neither general nor elegant. The graphical description of these states may be a useful alternative.

(10) The connection of graphical representation of model spaces with well established branches of mathematics, such as the theory of Diophantine equations (Mordell 1969), the theory of partitions (cf Andrews 1976), graph theory (cf Harary 1969) or operation research and especially integer programming (Garfinkel and Nemhauser 1972) are evident, although so far very little explored.

The ideas described in this paper should be treated more as a research program than a complete theory. Graphs of the fixed-slope type, representing model spaces, are simpler than general graphs used in physics and mathematics. Finding most efficient

computer representation and algorithms to deal with these graphs is one of the problems. Creating the programs that will use full insight into the structure of graphically represented spaces is quite a challenge. So far only the spaces of Abelian symmetry functions and eigenspaces of $\hat{S}^2$ were analysed in detail (Duch 1986b) and the graphical rules of matrix element evaluation derived. Non-Abelian problems are much more complicated. It is hard to include the representation of equivalent particles in a graph or to translate, say, Jahn and van Wieringen (1951) genealogical Young-Yamanouchi decomposition in the LST model into proper graphical terms, but a modest step in this direction has been taken. However, I must admit that since the GRMS idea is rather new there are many problems that were never seriously considered.

Except for technical developments—judging from the molecular physics applications (Shavitt 1983) it should be possible to solve routinely problems in the 10^6-dimensional spaces—GRMS should bring also methodological developments. It should be possible to solve exactly some problems with truncated model Hamiltonians by analysing, with the help of a graph, the structure of the Hamiltonian matrices. Another exciting possibility is a formulation of the space-enlargement perturbation theory, i.e. treating the problem in $(n+1)$-orbital space as a perturbation of the n-orbital space problem. A detailed analaysis of the structure of matrix representations of the operators is necessary to formulate such a theory and it obviously depends on a detailed analysis of the structure of the spaces used. Work in these directions is in progress.

Acknowledgments

I am grateful to the Alexander von Humboldt-Stiftung for supporting my stay in Germany, to G H F Diercksen for his hospitality at the Max-Planck-Institut in Garching, to F Müller-Plathe for reading the manuscript and to J Karwowski for numerous discussions and for using his program for the breadth search algorithm.

References

Andrews G E 1976 *The Theory Of Partitions* (Reading, MA: Addison-Wesley)
Bohr A and Mottelson B R 1969 *Nuclear Structure* vol 1 (New York: Benjamin)
Cohen S, Lawson R D, Macfarlane M H and Soga M 1966 *Methods in Computational Physics* vol 6, ed B Alder (New York: Academic) p 235
Davidson E R 1975 *J. Comput. Phys.* **17** 87
Duch W 1985a *J. Phys. A: Math. Gen.* **18** 3283
—— 1985b *Int. J. Quantum Chem.* **27** 59
—— 1986a *Chem. Phys. Lett.* **124** 442
—— 1986b *GRMS or Graphical Representation of Model Spaces* vol I *Basics* (*Lecture Notes in Chem.* **42**)
—— 1986c *Int. J. Quantum Chem.* **30** 799
Duch W and Karwowski J 1981 *Lecture Notes in Chemistry* **22** 260
—— 1982 *Int. J. Quantum. Chem.* **22** 783
—— 1985 *Comput. Phys. Rep.* **2** 92
Esser M 1984 *Int. J. Quantum Chem.* **26** 313
French J B, Halbert J C, McGrory J B and Wong S S M 1969 *Adv. Nucl. Phys.* **3** 193
Garfinkel R S and Nemhauser G L 1972 *Integer Programming* (New York: Wiley)
Harary F 1969 *Graph Theory* (Reading, MA: Addison-Wesley)
Harrison R J and Handy N C 1983 *Chem. Phys. Lett.* **95** 386
Jahn H A and van Wieringen H 1951 *Proc. R. Soc.* **209** 502

Mattuck R D 1976 *A Guide To Feynman Diagrams In The Many-Body Problem* (New York: McGraw-Hill) 2nd edn
McGrory J B and Wildenthal B H 1980 *Ann. Rev. Nucl. Particle Sci.* **30** 383
Mordell L J 1969 *Diophantine Equations* (New York: Academic)
Pauncz R 1979 *Spin Eigenfunctions: Their Construction And Use* (New York: Plenum)
Rettrup S 1986 *Int. J. Quantum. Chem.* **29** 119
Robb M and Niazi U 1984 *Comput. Phys. Rep.* **1** 127
Saxe P, Fox D J, Schaefer H F and Handy N C 1982 *J. Chem. Phys.* **77** 5584
Shavitt I 1981 *Lecture Notes in Chemistry* **22** 51
—— 1983 *New Horizons in Quantum Chemistry* ed P-O Löwdin and B Pullman (Dordrecht: Reidel) pp 279–93
Whitehead R R 1972 *Nucl. Phys.* A **182** 290
Whitehead R R, Watt A, Cole B J and Morrison I 1977 *Adv. Nucl. Phys.* **9** 123
Wong S S M 1981 *Lecture Notes in Physics* **144** 174