RITSSCHIL, A NEW PROGRAM FOR SHELL-MODEL CALCULATIONS

D. ZWARTS

Robert J. Van de Graafflaboratorium, Rijksuniversiteit Utrecht, P.O. Box 80000, 3508 TA Utrecht, The Netherlands

Received 26 October 1984; in final form 11 March 1985

PROGRAM SUMMARY

Title of program: RITSSCHIL

Catalogue number: AADA

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue)

Computer: CDC 175-100; *Installation*: State University Utrecht

Operating system: NOS/BE 1.5

Programming language used: Fortran V

High speed storage required: 47000 words

No. of bits in a word: 60

Overlay structure: 4 primary overlays

Peripherals used: card reader, disk, line printer

No. of lines in combined program and test deck: 4730

Keywords: shell model, second quantization, general reduction formula

Nature of physical problem
A new program for nuclear (or atomic) shell-model calculations is described. In comparison with other presently available shell-model codes, the new program, called RITSSCHIL, can treat a larger variety of operators and is more flexible. The reduced matrix elements of tensorial shell-model operators are calculated in a multi-shell space for nuclear (or atomic) shell-model states.

Method of solution
With the use of the general reduction formula[1] the matrix elements are expanded in terms of single-shell matrix elements and the elementary matrix elements that describe the operator.

Restrictions on the complexity of the problem
The program is suitable for all tensorial shell-model operators. The number of active shells is limited only by a dimension parameter in the program, which can be set by the user. The order in which the active shells are coupled can be defined by input cards.

Typical running time
The running time depends strongly on the number of active shells and on the dimensions of the computed matrices. For the test run on a CDC 175-100 computer 31 s of CPU time was required.

Reference

[1] J.B. French, E.C. Halbert, J.B. McGrory and S.S.M. Wong, Advan. Nucl. Phys. 3 (1969) 193.

0010-4655/85/$03.30 © Elsevier Science Publishers B.V.
(North-Holland Physics Publishing Division)

LONG WRITE-UP

1. Introduction

The program RITSSCHIL is a FORTRAN77 computer code for large spectroscopic calculations in the microscopic nuclear (or atomic) shell model. It is suitable for all tensorial shell-model operators, such as Hamiltonians (possibly including three- or more-body forces) and operators for electromagnetic transitions, (many-)particle transfer, beta decay, etc. Starting from matrix elements of standard operators in single shells RITSSCHIL calculates the matrix elements of the physical operators in a multishell basis. Since each multishell problem is approached as a recurrent two-shell problem the number of active shells is limited only by a dimension parameter in the program, which can be set by the user. The order in which the active shells are coupled together (the coupling tree) can be defined by input cards. So if one works in a proton–neutron formalism one may couple all proton shells together and all neutron shells together before coupling the proton shells to the neutron shells. In comparison with other presently available shell-model codes program RITSSCHIL can treat a larger variety of operators (e.g. for many-particle transfer) and it is more flexible. In most cases program RITSSCHIL will also run faster and use less space.

To present the program, in this paper the theoretical background is discussed in section 2 and the set-up of the program is described in section 3.

2. Theoretical background of program RITSSCHIL

The essential features of the theory behind the formalism of program RITSSCHIL are the use of second quantization and the representation of all states and operators in terms of irreducible spherical tensors.

In this section a short survey of the theory is presented in order to introduce the formalism to the user. For a more fundamental treatment the reader is referred to refs. [1,2]. After some preliminary remarks in section 2.1, section 2.2 introduces very briefly the state operators and adjoint state operators which are characteristic for the second-quantization formalism. In section 2.3 the transformation of operators to a standard-form expansion is discussed. Finally, section 2.4 offers a presentation of the general reduction formula which expresses matrix elements of operators acting in a multishell in terms of matrix elements of operators acting in single shells.

2.1. Preliminary remarks

The present treatment leans heavily on the approach of French et al. [1] used for the construction of the Oak Ridge–Rochester shell-model code. They chose for the expansion in spherical tensors because it allows the full use of Racah algebra. The advantages of second quantization are the easy handling of particle antisymmetry and the decomposability of operators, i.e. each operator can be expressed as a sum in which each term can be decomposed into single-shell parts, each part operating in a different shell. The extensions in the present treatment concern the iterative and more general set-up of the program (see section 3) which touches the theory for example in the definition of standard operators and in the way physical operators are transformed to a standard-form expansion.

Because the formalism does not depend on the set of quantum numbers considered, a direct-product notation is used. The quantum number n or m stands for all (commuting) quantum numbers that concern the occupation of a state; so in a proton–neutron formalism n stands for n_{p} and n_{n}, the number of protons and the number of neutrons. A single Greek symbol (ρ, λ, Γ, Ω) represents the set of vectorial quantum numbers (final tensorial ranks) considered, for example the set $\{J, T\}$ or $\{L, S\}$ or $\{L, S, T\}$. The further quantum numbers needed to complete the specification of a tensor are represented by the Greek symbol α or β. If in a formula a factor contains only occupation numbers or only vectorial quantum numbers, this factor stands for a product of such factors, one for each represented quantum number. For example a phase factor $(-1)^{nn'}$ will be $(-1)^{n_p n'_p + n_n n'_n}$ in a proton–neutron formalism. In this way the coupling of two tensors, represented by square brackets, can be defined by

$$[V^{\Gamma} \times W^{\Gamma'}]^{\Omega}_{\Omega_z} = \sum_{\Gamma_z \Gamma'_z} \langle \Gamma\Gamma_z \Gamma'\Gamma'_z \,|\, \Omega\Omega_z \rangle V^{\Gamma}_{\Gamma_z} W^{\Gamma'}_{\Gamma'_z}, \tag{1}$$

where the symbol $\langle \Gamma\Gamma_z \Gamma'\Gamma'_z \,|\, \Omega\Omega_z \rangle$ stands for a product of Clebsch–Gordan coefficients, one for each of the commuting vectorial quantum numbers considered. The reduced matrix elements of the Wigner–Eckart theorem are represented by double-bar matrix elements:

$$\langle n\Gamma\Gamma_z\alpha \,|\, W^{\Omega}_{\Omega_z} \,|\, n'\Gamma'\Gamma'_z\alpha' \rangle = (-1)^{2\Omega} \frac{\langle \Gamma'\Gamma'_z\Omega\Omega_z \,|\, \Gamma\Gamma_z \rangle}{(2\Gamma+1)^{1/2}} \langle n\Gamma\alpha \,\|\, W^{\Omega} \,\|\, n'\Gamma'\alpha' \rangle \tag{2}$$

and

$$\langle n\Gamma\alpha \,\|\, W^{\Omega} \,\|\, n'\Gamma'\alpha' \rangle = \frac{(-1)^{2\Omega}}{(2\Gamma+1)^{1/2}} \sum_{\Gamma_z \Omega_z \Gamma'_z} \langle \Gamma'\Gamma'_z\Omega\Omega_z \,|\, \Gamma\Gamma_z \rangle \langle n\Gamma\Gamma_z\alpha \,|\, W^{\Omega}_{\Omega_z} \,|\, n'\Gamma'\Gamma'_z\alpha' \rangle. \tag{3}$$

2.2. State operators and adjoint state operators

This section introduces very briefly the state operators (section 2.2.1), the adjoint state operators (section 2.2.2) and some special matrix elements of these operators (section 2.2.3). The only difference with the treatment in refs. [1,2] is the introduction of a free coupling order of single-shell states (the coupling tree).

2.2.1. State operators

Let the symbol $|\,\rangle$ denote the vacuum state in which no particles are present and let $|n=1\ \rho\rho_z\rangle$ denote the single-particle state with vectorial quantum numbers ρ and projection quantum numbers ρ_z, then the creation operator $a^{\dagger}_{\rho\rho_z}$ can be implicitly defined by the relation

$$|n=1\ \rho\rho_z\ \rangle = a^{\dagger}_{\rho\rho_z}|\ \rangle. \tag{4}$$

The operator $a^{\dagger}_{\rho\rho_z}$ creates a normalized one-particle state and therefore it is called the state operator for the one-particle state. This concept can be extended by defining the n-particle state operator $Z(n\Gamma\Gamma_z\alpha)$ as the operator that creates the antisymmetrized and normalized n-particle state with quantum numbers n, Γ, Γ_z and α:

$$|n\Gamma\Gamma_z\alpha\ \rangle = Z(n\Gamma\Gamma_z\alpha)\,|\ \rangle. \tag{5}$$

The n-particle state operators can be written as coupled products of n creation operators. The single-shell state operators, which create states with all particles in the same orbit ρ, are for $n = 0$, 1, 2 given by:

$$Z_{\rho}(n=0) = 1, \quad Z_{\rho}(n=1\ \rho\rho_z) = a^{\dagger}_{\rho\rho_z}, \quad Z_{\rho}(n=2\ \Gamma\Gamma_z) = (-1/\sqrt{2})\left[a^{\dagger}_{\rho} \times a^{\dagger}_{\rho}\right]^{\Gamma}_{\Gamma_z}. \tag{6}$$

The expression of single-shell state operators in terms of creation operators can be derived more generally with the recurrent relation $(n = n_1 + n_2)$

$$Z_{\rho}(n\Gamma\Gamma_z\alpha) = (-1)^{n_1 n_2}\binom{n}{n_1}^{-1/2} \sum_{\Gamma_1\alpha_1\Gamma_2\alpha_2} \langle \rho^{n}\Gamma\alpha \,|\}\, \rho^{n_1}\Gamma_1\alpha_1(\rho^{n_2}\Gamma_2\alpha_2)\rangle \left[Z_{\rho}(n_1\Gamma_1\alpha_1) \times Z_{\rho}(n_2\Gamma_2\alpha_2)\right]^{\Gamma}_{\Gamma_z}. \tag{7}$$

The symbols $\langle \rho^n \Gamma \alpha | \} \rho^{n_1} \Gamma_1 \alpha_1 (\rho^{n_2} \Gamma_2 \alpha_2) \rangle$ denote the coefficients of fractional parentage (c.f.p.'s) known from first quantization for the antisymmetrization and orthonormalization of the basis states. The binomial coefficient $\binom{n}{n_1}^{-1/2}$ is introduced for normalization, because the particle numbering has vanished in second quantization. The phasefactor $(-1)^{n_1 n_2}$ is required to let the order in which the state operators act agree with the phase convention of the c.f.p.'s in first quantization, i.e. the particles are coupled in the same order as in which they are created. The interchange of the two state operators yields this phase factor as follows from the (anti-)commutation relation for state operators

$$Z(n_2 \Gamma_2 \Gamma_{2z} \alpha_2) Z(n_1 \Gamma_1 \Gamma_{1z} \alpha_1) = (-1)^{n_1 n_2} Z(n_1 \Gamma_1 \Gamma_{1z} \alpha_1) Z(n_2 \Gamma_2 \Gamma_{2z} \alpha_2). \tag{8}$$

For two-shell states, having particles in two different shells, this (anti-)commutation relation provides the antisymmetry, while the Clebsch–Gordan coefficients provide the orthonormality of the two-shell states if the state operators are constructed with the relation

$$Z(n \Gamma \Gamma_z [\alpha_1 \alpha_2]) = (-1)^{n_1 n_2} [Z_1(n_1 \Gamma_1 \alpha_1) \times Z_2(n_2 \Gamma_2 \alpha_2)]^{\Gamma}_{\Gamma_z}. \tag{9}$$

The operator Z_1 acts in one of the two shells and Z_2 in the other. The phase factor $(-1)^{n_1 n_2}$ arises again from the coupling convention mentioned above. Note that this factor vanishes if shell 1 and shell 2 contain inequivalent types of particles, because the state operators for inequivalent types of particles commute.

Because the two-shell state operators thus constructed can be manipulated the same way as singe-shell state operators, one can consider the combined shells as one new shell. So one can use eq. (9) iteratively for the construction of state operators in spaces that consist of more than two single shells. To prevent the appearance of binomial coefficients the two combined groups of shells have to be disjunct (i.e. without equivalent single shells in common). The order in which the different single shells are coupled by iterative application of this relation will be called the coupling tree of the active shells. The resulting configuration space will be referred to as a multishell.

2.2.2. *Adjoint state operators*

The adjoint state operators can be introduced in the same way. If the symbol $|0\rangle$ denotes the closed core and $|n=-1\ \rho\rho_z\rangle$ denotes the single-hole state with quantum numbers ρ and ρ_z, then the annihilation operator $\tilde{a}_{\rho\rho_z}$ is implicitly defined by

$$|n=-1\ \rho\rho_z\ \rangle = \tilde{a}_{\rho\rho_z} |0\ \rangle. \tag{10}$$

The adjoint state operator $\tilde{Z}(n \Gamma \Gamma_z \alpha)$ is likewise defined as the operator that creates an n-hole state with quantum numbers $-n$, Γ, Γ_z and α:

$$|-n\ \Gamma \Gamma_z \alpha\ \rangle = \tilde{Z}(n \Gamma \Gamma_z \alpha) |0\ \rangle. \tag{11}$$

The adjoint state operators are related to the Hermitian conjugates of state operators by

$$\tilde{Z}(n \Gamma \Gamma_z \alpha) = (-1)^{\Gamma + \Gamma_z} \{ Z(n \Gamma - \Gamma_z \alpha) \}^{\dagger}. \tag{12}$$

The phase factor $(-1)^{\Gamma_z}$ is needed in this relation to maintain the tensorial character of the adjoint state operators; $(-1)^{\Gamma}$ is added only to prevent the appearance of imaginary factors for the half-integer values of Γ_z. From this relation it is also clear that an adjoint state operator acting to the left on the vacuum state creates a particle state

$$\langle | \tilde{Z}(n \Gamma \Gamma_z \alpha) = (-1)^{\Gamma + \Gamma_z} \langle n \Gamma - \Gamma_z \alpha |. \tag{13}$$

Each n-hole adjoint state operator can be written as a coupled product of n annihilation operators. For $n = 0, 1, 2$ the adjoint state operators are given by

$$\tilde{Z}(n=0) = 1, \quad \tilde{Z}(n=1\ \rho\rho_z) = \tilde{a}_{\rho\rho_z}, \quad \tilde{Z}(n=2 \Gamma \Gamma_z [\rho\lambda]) = \frac{1}{(1+\delta_{\rho\lambda})^{1/2}} [\tilde{a}_{\rho} \times \tilde{a}_{\lambda}]^{\Gamma}_{\Gamma_z}. \tag{14}$$

For single-shell adjoint state operators the recurrent relation reads

$$\tilde{Z}_p(n\Gamma\Gamma_z\alpha) = \binom{n}{n_1}^{-1/2} \sum_{\Gamma_z\alpha_1\Gamma_2\alpha_2} \langle \rho^n\Gamma\alpha | \} \rho^{n_1}\Gamma_1\alpha_1(\rho^{n_2}\Gamma_2\alpha_2)\rangle \left[\tilde{Z}_\rho(n_1\Gamma_1\alpha_1) \times \tilde{Z}_\rho(n_2\Gamma_2\alpha_2)\right]^{\Gamma}_{\Gamma_z} \tag{15}$$

and for multishell adjoint state operators one has

$$\tilde{Z}(n\Gamma\Gamma_z[\alpha_1\alpha_2]) = \left[\tilde{Z}_1(n_1\Gamma_1\alpha_1) \times \tilde{Z}_2(n_2\Gamma_2\alpha_2)\right]^{\Gamma}_{\Gamma_z}. \tag{16}$$

Note that there is no phasefactor $(-1)^{n_1n_2}$; the particles are annihilated in an order opposite to the order in which they are coupled.

The (anti-)commutation relation for adjoint state operators is the same as for state operators

$$\tilde{Z}(n_2\Gamma_2\Gamma_{2z}\alpha_2)\tilde{Z}(n_1\Gamma_1\Gamma_{1z}\alpha_1) = (-1)^{n_1n_2}\tilde{Z}(n_1\Gamma_1\Gamma_{1z}\alpha_1)\tilde{Z}(n_2\Gamma_2\Gamma_{2z}\alpha_2). \tag{17}$$

If the operators Z and $\tilde{Z}$ do not have any orbit in common one also has

$$\tilde{Z}(n_2\Gamma_2\Gamma_{2z}\alpha_2)Z(n_1\Gamma_1\Gamma_{1z}\alpha) = (-1)^{n_1n_2}Z(n_1\Gamma_1\Gamma_{1z}\alpha_1)\tilde{Z}(n_2\Gamma_2\Gamma_{2z}\alpha_2). \tag{18}$$

2.2.3. *Matrix elements of state operators and adjoint state operators*

Using eq. (7) one can express the reduced matrix elements of single-shell state operators between single-shell states in terms of c.f.p.'s

$$\langle n\Gamma\alpha \| Z(n_2\Gamma_2\alpha_2) \| n_1\Gamma_1\alpha_1\rangle = (-1)^{2\Gamma_2}\binom{n}{n_1}^{1/2}(2\Gamma+1)^{1/2}\langle \rho^n\Gamma\alpha | \} \rho^{n_1}\Gamma_1\alpha_1(\rho^{n_2}\Gamma_2\alpha_2)\rangle. \tag{19}$$

For (multishell) state operators acting on the vacuum state one can derive

$$\begin{aligned}\langle n\Gamma\alpha \| Z(n'\Gamma'\alpha') \| \rangle &= \frac{(-1)^{2\Gamma'}}{(2\Gamma+1)^{1/2}} \sum_{\Gamma_z} \langle 00\Gamma'\Gamma_z | \Gamma\Gamma_z\rangle\langle n\Gamma\Gamma_z\alpha | Z(n'\Gamma'\Gamma_z\alpha') | \rangle \\ &= \frac{(-1)^{2\Gamma'}}{(2\Gamma+1)^{1/2}} \sum_{\Gamma_z} \langle n\Gamma\Gamma_z\alpha | n'\Gamma\Gamma_z\alpha'\rangle \delta_{\Gamma\Gamma'} \\ &= (-1)^{2\Gamma}(2\Gamma+1)^{1/2}\delta_{nn'}\delta_{\Gamma\Gamma'}\delta_{\alpha\alpha'}.\end{aligned} \tag{20}$$

The matrix elements of adjoint state operators can be expressed in those of state operators

$$\begin{aligned}\langle n'\Gamma'\alpha' \| \tilde{Z}(n''\Gamma''\alpha'') \| n\Gamma\alpha\rangle &= \frac{(-1)^{2\Gamma''}}{(2\Gamma'+1)^{1/2}} \sum_{\Gamma_z\Gamma_z'\Gamma_z''} \langle \Gamma\Gamma_z\Gamma''\Gamma_z'' | \Gamma'\Gamma_z'\rangle\langle n'\Gamma'\Gamma_z'\alpha' | \tilde{Z}(n''\Gamma''\Gamma_z''\alpha'') | n\Gamma\Gamma_z\alpha\rangle \\ &= \frac{(-1)^{2\Gamma''}}{(2\Gamma'+1)^{1/2}} \sum_{\Gamma_z\Gamma_z'\Gamma_z''} (-1)^{\Gamma-\Gamma'-\Gamma_z+\Gamma_z'}\left\{\frac{2\Gamma'+1}{2\Gamma+1}\right\}^{1/2} \langle \Gamma'\Gamma_z'\Gamma''-\Gamma_z'' | \Gamma\Gamma_z\rangle \\ &\quad \times \cancel{\langle n'\Gamma'\Gamma_z'\alpha'} | \tilde{Z}(n''\Gamma''\Gamma_z''\alpha'') | n\Gamma\Gamma_z\alpha\rangle \;\rule{2cm}{0.4pt} \\ &= \frac{(-1)^{2\Gamma''}}{(2\Gamma+1)^{1/2}} \sum_{\Gamma\Gamma_z'\Gamma_z''} (-1)^{\Gamma-\Gamma'-\Gamma_z+\Gamma_z'}\langle \Gamma'\Gamma_z'\Gamma''-\Gamma_z'' | \Gamma\Gamma_z\rangle \\ &\quad \times (-1)^{\Gamma''+\Gamma_z''}\langle n\Gamma\Gamma_z\alpha | Z(n''\Gamma''-\Gamma_z''\alpha'') | n'\Gamma'\Gamma_z'\alpha'\rangle \\ &= (-1)^{\Gamma-\Gamma'-\Gamma''}\langle n\Gamma\alpha \| Z(n''\Gamma''\alpha'') \| n'\Gamma'\alpha'\rangle.\end{aligned} \tag{21}$$

So in particular

$$\langle \| \tilde{Z}(n'\Gamma'\alpha') \| n\Gamma\alpha \rangle = \langle n\Gamma\alpha \| Z(n'\Gamma'\alpha') \| \rangle = (-1)^{2\Gamma}(2\Gamma+1)^{1/2}\delta_{nn'}\delta_{\Gamma\Gamma'}\delta_{\alpha\alpha'}. \tag{22}$$

Using eq. (21) and an intermediate-state expansion one can derive a more general conjugation relation:

$$\begin{aligned}
&\langle n'\Gamma'\alpha' \| \left[Z(n_b\Gamma_b\alpha_b) \times \tilde{Z}(n_a\Gamma_a\alpha_a) \right]^{\Omega} \| n\Gamma\alpha \rangle \\
&= (-1)^{\Gamma'+\Omega+\Gamma}(2\Omega+1)^{1/2} \sum_{n''\Gamma''\alpha''} \begin{Bmatrix} \Gamma\Gamma'\Omega \\ \Gamma_b\Gamma_a\Gamma'' \end{Bmatrix} \langle n'\Gamma'\alpha' \| Z(n_b\Gamma_b\alpha_b) \| n''\Gamma''\alpha'' \rangle \\
&\quad \times \langle n''\Gamma''\alpha'' \| \tilde{Z}(n_a\Gamma_{a\alpha a}) \| n\Gamma\alpha \rangle \\
&= (-1)^{\Gamma'+\Omega+\Gamma}(2\Omega+1)^{1/2} \sum_{n''\Gamma''\alpha''} \begin{Bmatrix} \Gamma'\Gamma\Omega \\ \Gamma_a\Gamma_b\Gamma'' \end{Bmatrix} (-1)^{\Gamma'-\Gamma_b-\Gamma''} \langle n''\Gamma''\alpha'' \| \tilde{Z}(n_b\Gamma_b\alpha_b) \| n'\Gamma'\alpha' \rangle \\
&\quad \times (-1)^{\Gamma-\Gamma_a-\Gamma''} \langle n\Gamma\alpha \| Z(n_a\Gamma_a\alpha_a) \| n''\Gamma''\alpha'' \rangle \\
&= (-1)^{\Gamma-\Gamma_a+\Gamma_b-\Gamma'} \langle n\Gamma\alpha \| \left[Z(n_a\Gamma_a\alpha_a) \times \tilde{Z}(n_b\Gamma_b\alpha_b) \right]^{\Omega} \| n'\Gamma'\alpha' \rangle,
\end{aligned} \tag{23}$$

where $\{:::\}$ denotes a $6j$-symbol. A special case of matrix elements of this type can be calculated by use of an intermediate-state expansion

$$\begin{aligned}
&\langle n\Gamma\alpha \| \left[Z(n\Gamma_a\alpha_a) \times \tilde{Z}(n'\Gamma_b\alpha_b) \right]^{\Omega} \| n'\Gamma'\alpha' \rangle \\
&= (-1)^{\Gamma+\Omega+\Gamma'}(2\Omega+1)^{1/2} \begin{Bmatrix} \Gamma'\Gamma\Omega \\ \Gamma_a\Gamma_b 0 \end{Bmatrix} \langle n\Gamma\alpha \| Z(n\Gamma_a\alpha_a) \| \rangle \langle \| \tilde{Z}(n'\Gamma_b\alpha_b) \| n'\Gamma'\alpha' \rangle \\
&= (-1)^{\Gamma+\Omega+\Gamma'}(2\Omega+1)^{1/2} \frac{(-1)^{\Gamma'+\Gamma+\Omega}\delta_{\Gamma\Gamma_a}\delta_{\Gamma'\Gamma_b}}{(2\Gamma'+1)^{1/2}(2\Gamma+1)^{1/2}} (-1)^{2\Gamma}(2\Gamma+1)^{1/2}(-1)^{2\Gamma'}(2\Gamma'+1)^{1/2}\delta_{\alpha\alpha_a}\delta_{\alpha'\alpha_b} \\
&= (-1)^{2\Omega}(2\Omega+1)^{1/2}\delta_{\Gamma\Gamma_a}\delta_{\alpha\alpha_a}\delta_{\Gamma'\Gamma_b}\delta_{\alpha'\alpha_b}.
\end{aligned} \tag{24}$$

2.3. Physical operators in second-quantized form

To use the second-quantization formalism one needs not only a description of many-fermion states in terms of creation and annihilation operators (section 2.2), but also the operators that one wants to use should be represented in the same formalism. Section 2.3.1 shows that this is possible for all usual shell-model operators and in section 2.3.2 the chosen method is discussed by comparing it with the approach of French et al. [1,2].

2.3.1. The standard-form expansion of operators

Let O be any symmetric shell-model operator, i.e. O acts the same way on all indistinguishable particles in a given state. The physical operator O can be decomposed into parts (spherical tensors) that are characterized by a well-defined transformation under rotations of the three-dimensional frame of reference:

$$O = \sum_{\Omega} O^{\Omega} = \sum_{\Omega nn'} O^{\Omega}_{nn'}. \tag{25}$$

In this formula O^{Ω} represents a spherical tensor with well defined rank(s) Ω. Each of those tensors O^{Ω} can be split into tensors $O^{\Omega}_{nn'}$ that annihilate n' particles and create n particles. For scalar operators one has $\Omega = 0$. For $n = n'$ $O^{\Omega}_{nn'}$ is a number-conserving n-body operator. For $n > n'$ $O^{\Omega}_{nn'}$ is an $(n-n')$-particle transfer operator.

In this way the Hamiltonian H is written as $H = H_{11}^{0} + H_{22}^{0} + \dots$ i.e. H is split into its one-body part, its two-body part and possibly its three- or more-body parts. The electromagnetic transition operator $O(\tilde{\Omega} L)$ can be written as $O(\tilde{\omega} L) = O_{11}^{L}(\tilde{\omega} L)$ and the α-transfer operator A is split into its separate L-channels $A = \sum_L A_{40}^{L}$.

The matrix elements of $O_{nn'}^{\Omega}$ between any two states are fully determined by the matrix elements of $O_{nn'}^{\Omega}$ between n-particle final states and n'-particle initial states. So one may define $O_{nn'}^{\Omega}$ by giving the values of its elementary matrix elements, i.e. all matrix elements $(-1)^{2\Omega}\langle n\Gamma\alpha \| O_{nn'}^{\Omega} \| n'\Gamma'\alpha'\rangle/(2\Omega+1)^{1/2}$ for a complete basis of n-particle states and a complete basis of n'-particle states. Therefore the only requirement necessary to define $O_{nn'}^{\Omega}$ in second-quantization formalism is that the elementary matrix elements of the second-quantized operator $\hat{O}_{nn'}^{\Omega}$ are the same as the elementary matrix elements of $O_{nn'}^{\Omega}$. This requirement is met uniquely if one defines

$$\hat{O}_{nn'}^{\Omega\Omega_z} = \sum_{\Gamma\alpha\Gamma'\alpha'} (-1)^{2\Omega} \frac{\langle n\Gamma\alpha \| O_{nn'}^{\Omega} \| n'\Gamma'\alpha'\rangle}{(2\Omega+1)^{1/2}} \left[Z(n\Gamma\alpha) \times \tilde{Z}(n'\Gamma'\alpha') \right]_{\Omega_z}^{\Omega}. \tag{26}$$

The summation runs over a complete basis of n-particle states and over a complete basis of n'-particle states.

In this way the second-quantized operators for H, $O(\tilde{\omega} L)$ and A are:

$$\hat{H} = \sum_{\rho} \langle \rho | H_{11}^{0} \| \rho \rangle \left[a_\rho^\dagger \times \tilde{a}_\rho \right]_0^0 + \sum_{\substack{\Gamma\rho\lambda\sigma\tau \\ \rho\leqslant\lambda\ \sigma\leqslant\tau}} \langle \rho\lambda \| H_{22}^{0} \| \sigma\tau \rangle_\Gamma \left[Z(n=2\ \Gamma[\rho\lambda]) \times \tilde{Z}(n=2\ \Gamma[\sigma\tau]) \right]_0^0 + \dots,$$

$$\hat{O}(\tilde{\omega} LM) = \sum_{\rho\lambda} \frac{\langle \rho \| O(\tilde{\omega} L) \| \lambda \rangle}{(2L+1)^{1/2}} \left[a_\rho^\dagger \times \tilde{a}_\lambda \right]_M^L,$$

$$\hat{A}^{LM} = \sum_{\alpha} \frac{\langle n=4\ L\alpha \| A^L \| \rangle}{(2L+1)^{1/2}} Z(n=4\ LM\alpha) = \sum_{\alpha} \langle n=4\ LM\alpha | A^{LM} \rangle Z(n=4\ LM\alpha).$$

By the use of eq. (24) it is easily shown that this definition meets the requirement:

$$\begin{aligned} \langle n\bar{\Gamma}\bar{\alpha} \| \hat{O}_{nn'}^{\Omega} \| n'\bar{\Gamma}'\bar{\alpha}'\rangle &= \sum_{\Gamma\alpha\Gamma'\alpha'} (-1)^{2\Omega} \frac{\langle n\Gamma\alpha \| O_{nn'}^{\Omega} \| n'\Gamma'\alpha'\rangle}{(2\Omega+1)^{1/2}} \langle n\bar{\Gamma}\bar{\alpha} \| \left[Z(n\Gamma\alpha) \times \tilde{Z}(n'\Gamma'\alpha') \right]^{\omega} \| n'\bar{\Gamma}'\bar{\alpha}'\rangle \\ &= \sum_{\Gamma\alpha\Gamma'\alpha'} (-1)^{2\Omega} \frac{\langle n\Gamma\alpha \| O_{nn'}^{\Omega} \| n'\Gamma'\alpha'\rangle}{(2\Omega+1)^{1/2}} (-1)^{2\Omega} (2\Omega+1)^{1/2} \delta_{\Gamma\bar{\Gamma}} \delta_{\alpha\bar{\alpha}} \delta_{\Gamma'\bar{\Gamma}'} \delta_{\alpha'\bar{\alpha}'} \\ &= \langle n\bar{\Gamma}\bar{\alpha} \| O_{nn'}^{\Omega} \| n'\bar{\Gamma}'\bar{\alpha}'\rangle. \end{aligned} \tag{27}$$

Eq. (26) will be called the standard-form expansion of operator $O_{nn'}^{\Omega}$.

2.3.2. The standard operators

By choosing eq. (26) as the standard-form expansion one has implicitly chosen a set of standard operators to be a basis for the space of all tensorial symmetric operators in the given multishell. These standard operators were defined by

$$F(m\Omega\Omega_z S\beta) = \left[Z(n\Gamma\alpha) \times \tilde{Z}(n'\Gamma'\alpha') \right]_{\Omega_z}^{\Omega} \tag{28}$$

with $m = n - n'$, $S = n + n'$ and β containing the determination of the 'inner' structure (Γ, α, Γ' and α'). With eqs. (25), (26) and (28) one can write the multishell matrix elements of any tensorial symmetric shell-model operator O as a linear combination of multishell matrix elements of these standard operators, using the elementary matrix elements of O as coefficients:

$$\langle \bar{n}\bar{\Gamma}\bar{\alpha} \| O \| \bar{n}'\bar{\Gamma}'\bar{\alpha}' \rangle = \sum_{\Omega n n'} \sum_{\Gamma\alpha\Gamma'\alpha'} (-1)^{2\Omega} \frac{\langle n\Gamma\alpha \| O^{\Omega}_{nn'} \| n'\Gamma'\alpha' \rangle}{(2\Omega+1)^{1/2}} \langle \bar{n}\bar{\Gamma}\bar{\alpha} \| F(m\Omega S\beta) \| \bar{n}'\bar{\Gamma}'\bar{\alpha}' \rangle . \tag{29}$$

The present standard operators are different from the standard-form operators in refs. [1,2]. French et al. define the standard-form operators as coupled products of single-shell standard-form operators (each single-shell standard-form operator being a coupled product of a state operator and an adjoint state operator). In the present approach each standard operator is a coupled product of one state operator and one adjoint state operator (while according to eqs. (9) and (16) the state operators are defined as coupled products of single-shell state operators). E.g. a three-shell standard-form operator of ref. [1] is given in a schematic notation by

$$F = [f_1 \times f_2 \times f_3] = [[z_1 \times \tilde{z}_1] \times [z_2 \times \tilde{z}_2] \times [z_3 \times \tilde{z}_3]],$$

while a three-shell standard operator in the present approach is given by

$$F = [Z \times \tilde{Z}] = [[z_1 \times z_2 \times z_3] \times [\tilde{z}_1 \times \tilde{z}_2 \times \tilde{z}_3]].$$

The approach of French et al. has the disadvantage that for the general case the calculation of the expansion coefficients in the standard-form expansion is rather laborious, whereas in the present approach the expansion coefficients are just the elementary matrix elements. The difficulty has not vanished but is transferred to the calculation of the matrix elements of standard operators where it can be solved elegantly because of the iterative approach.

The standard operators in the present approach obey the recurrent relation:

$$\begin{aligned} F(m\Omega\Omega_z S\beta) &= [Z(n\Gamma\alpha) \times \tilde{Z}(n'\Gamma'\alpha')]^{\Omega}_{\Omega_z} \\ &= \left[(-1)^{n_1 n_2} [Z_1(n_1\Gamma_1\alpha_1) \times Z_2(n_2\Gamma_2\alpha_2)]^{\Gamma} \times [\tilde{Z}_1(n_1'\Gamma_1'\alpha_1') \times \tilde{Z}_2(n_2'\Gamma_2'\alpha_2')]^{\Gamma'}\right]^{\Omega}_{\Omega_z} \\ &= (-1)^{n_1 n_2} (-1)^{n_1' n_2} \sum_{\Omega_1\Omega_2} [(2\Gamma+1)(2\Gamma'+1)(2\Omega_1+1)(2\Omega_2+1)]^{1/2} \begin{Bmatrix} \Gamma_1 \Gamma_2 \Gamma \\ \Gamma_1' \Gamma_2' \Gamma' \\ \Omega_1 \Omega_2 \Omega \end{Bmatrix} \\ &\quad \times \left[[Z_1(n_1\Gamma_1\alpha_1) \times \tilde{Z}_1(n_1'\Gamma_1'\alpha_1')]^{\Omega_1} \times [Z_2(n_2\Gamma_2\alpha_2) \times \tilde{Z}_2(n_2'\Gamma_2'a_2')]^{\Omega_2}\right]^{\Omega}_{\Omega_z} \\ &= (-1)^{(n_1+n_1')n_2} \sum_{\Omega_1\Omega_2} [(2\Gamma+1)(2\Gamma'+1)(2\Omega_1+1)(2\Omega_2+1)]^{1/2} \begin{Bmatrix} \Gamma_1 \Gamma_2 \Gamma \\ \Gamma_1' \Gamma_2' \Gamma' \\ \Omega_1 \Omega_2 \Omega \end{Bmatrix} \\ &\quad \times [F_1(m_1\Omega_1 S_1\beta_1) \times F_2(m_2\Omega_2 S_2\beta_2)]^{\Omega}_{\Omega_z}, \end{aligned} \tag{30}$$

where {:::} denotes a $9j$-symbol. The subscripts $_1$ and $_2$ refer to the two shells that are coupled to get the final multishell. Note that almost all parameters are determined by parameter β; the summation is only over Ω_1 and Ω_2.

For a given coupling tree this recurrent relation (iteratively used) is all one needs to write $F(m\Omega\Omega_z S\beta)$ as a sum of coupled products of single-shell standard operators (and to find in this way the expansion coefficients of French et al. for the general case). But this has not to be done explicitly here, because the programme needs only this relation in the iterative set-up.

Note that for the operators H (up to two-particle parts), $O(\tilde{\omega}L)$ and A in section 2.3.1 the $9j$-symbols in the recurrent relation will contain always at least one zero-argument, so the $9j$-symbols reduce to $6j$-symbols or even further.

2.4. The general reduction formula

In section 2.2 the multishell states are defined in a recurrent way (eqs. (9) and (16)) and in section 2.3 the same is done for multishell operators (eq. (30)). Using these recurrent definitions one can derive also a recurrent relation for multishell matrix elements, i.e. a formula to express matrix elements of operators in a multishell in terms of matrix elements of operators in the two shells that, according to the coupling tree, constitute the total multishell. A review of this derivation is presented in section 2.4.1, while in section 2.4.2 the resulting general reduction formula is discussed.

2.4.1. Derivation of the general reduction formula

Using eqs. (29) and (30) one can reduce the calculation of multishell matrix elements of any symmetric tensorial shell-model operator to the calculation of multishell matrix elements of the coupled operator product $[F_1(m_1\Omega_1S_1\beta_1)\times(F_2(m_2\Omega_2S_2\beta_2)]^{\Omega}_{\Omega_z}$ (for the notation see section 2.3.2).

The first step in the derivation of the reduction formula for the matrix elements of this operator product is to show explicitly all Clebsch–Gordan coefficients involved in these matrix elements (to shorten the notation the quantum numbers α, S and β are suppressed):

$$\begin{aligned}
&\langle n\Gamma\|[F_1(m_1\Omega_1)\times F_2(m_2\Omega_2)]^{\Omega}\|n'\Gamma'\rangle\\
&=\frac{(-1)^{2\Omega}}{(2\Gamma+1)^{1/2}}\sum_{\Gamma_z\Omega_z\Gamma'_z}\langle\Gamma'\Gamma'_z\Omega\Omega_z|\Gamma\Gamma_z\rangle\langle n\Gamma\Gamma_z|[F_1(m_1\Omega_1)\times F_2(m_2\Omega_2)]^{\Omega}_{\Omega_z}|n'\Gamma'\Gamma'_z\rangle\\
&=\frac{(-1)^{2\Omega}}{(2\Gamma+1)^{1/2}}\sum_{\Gamma_z\Omega_z\Gamma'_z}\langle\Gamma'\Gamma'_z\Omega\Omega_z|\Gamma\Gamma_z\rangle\langle|(-1)^{\Gamma-\Gamma_z}\tilde{Z}(n\Gamma-\Gamma_z)[F_1(m_1\Omega_1)\times F_2(m_2\Omega_2)]^{\Omega}_{\Omega_z}Z(n'\Gamma'\Gamma'_z)|\rangle\\
&=\frac{(-1)^{2\Omega}}{(2\Gamma+1)^{1/2}}\sum_{\Gamma_z\Omega_z\Gamma'_z}\langle\Gamma'\Gamma'_z\Omega\Omega_z|\Gamma\Gamma_z\rangle\langle|(-1)^{\Gamma-\Gamma_z}[\tilde{Z}_1(n_1\Gamma_1)\times\tilde{Z}_2(n_2\Gamma_2)]^{\Gamma}_{-\Gamma_z}\\
&\quad\times[F_1(m_1\Omega_1)\times F_2(m_2\Omega_2)]^{\Omega}_{\Omega_z}(-1)^{n'_1n'_2}[Z_1(n'_1\Gamma'_1)\times Z_2(n'_2\Gamma'_2)]^{\Gamma'}_{\Gamma'_z}|\rangle\\
&=\frac{(-1)^{2\Omega}}{(2\Gamma+1)^{1/2}}\sum\sum\sum\sum_{\Gamma_z\Omega_z\Gamma'_z\Gamma_{1z}\Gamma_{2z}\Omega_{1z}\Omega_{2z}\Gamma'_{1z}\Gamma'_{2z}}\langle\Gamma'_1\Gamma'_z\Omega\Omega_z|\Gamma\Gamma_z\rangle\langle\Gamma_1-\Gamma_{1z}\Gamma_2-\Gamma_{2z}|\Gamma-\Gamma_z\rangle\\
&\quad\times\langle\Omega_1\Omega_{1z}\Omega_2\Omega_{2z}|\Omega\Omega_z\rangle\langle\Gamma'_1\Gamma'_{1z}\Gamma'_2\Gamma'_{2z}|\Gamma'\Gamma'_z\rangle(-1)^{\Gamma-\Gamma_z}(-1)^{n'_1n'_2}\\
&\quad\times\langle|\tilde{Z}_1(n_1\Gamma_1-\Gamma_{1z})\tilde{Z}_2(n_2\Gamma_2-\Gamma_{2z})F_1(m_1\Omega_1\Omega_{1z})F_2(m_2\Omega_2\Omega_{2z})Z_1(n'_1\Gamma'_1\Gamma'_{1z})Z_2(n'_2\Gamma'_2\Gamma'_{2z})|\rangle\\
&=\frac{(-1)^{2\Omega}}{(2\Gamma+1)^{1/2}}\sum_{\text{all }z\text{'s}}\langle\Gamma'\Gamma'_z\Omega\Omega_z|\Gamma\Gamma_z\rangle\langle\Gamma_1\Gamma_{1z}\Gamma_2\Gamma_{2z}|\Gamma\Gamma_z\rangle\langle\Omega_1\Omega_{1z}\Omega_2\Omega_{2z}|\Omega\Omega_z\rangle\langle\Gamma'_1\Gamma'_{1z}\Gamma'_2\Gamma'_{2z}|\Gamma'\Gamma'_z\rangle\\
&\quad\times(-1)^{\Gamma_1+\Gamma_2-\Gamma_z}(-1)^{n'_1n'_2}\\
&\quad\times\langle|\tilde{Z}_1(n_1\Gamma_1-\Gamma_{1z})\tilde{Z}_2(n_2\Gamma_2-\Gamma_{2z})F_1(m_1\Omega_1\Omega_{1z})F_2(m_2\Omega_2\Omega_{2z})Z_1(n'_1\Gamma'_1\Gamma'_{1z})Z_2(n'_2\Gamma'_2\Gamma'_{2z})|\rangle.
\end{aligned}\tag{31}$$

To separate the operators acting in shell 1 and those acting in shell 2 one has to interchange the operators

$\tilde{Z}_2$ and F_1, F_2 and Z_1, and finally $\tilde{Z}_2$ and Z_1. Therefore one needs the (anti-)commutation relations for state operators and standard operators acting in disjunct spaces. These relations can be easily derived from the definition of standard operators (28) and the (anti-)commutation relations (8), (17) and (18). One obtains

$$F_1(m_1\Omega_1\Omega_{1z})\tilde{Z}_2(n_2\Gamma_2-\Gamma_{2z})=(-1)^{m_1n_2}Z_2(n_2\Gamma_2-\Gamma_{2z})F_1(m_1\Omega_1\Omega_{1z}), \tag{32a}$$

$$Z_1(n_1'\Gamma_1'\Gamma_{1z}')F_2(m_2\Omega_2\Omega_{2z})=(-1)^{n_1'm_2}F_2(m_2\Omega_2\Omega_{2z})Z_1(n_1'\Gamma_1'\Gamma_{1z}'). \tag{32b}$$

Because F_1 is an m_1-particle transfer operator (creating m_1 particles more than it is annihilating) and F_2 is an m_2-particle transfer operator one should have $n_1'+m_1=n_1$ and $n_2'+m_2=n_2$. So one finds:

$$\begin{aligned}
&(-1)^{\Gamma_1+\Gamma_2-\Gamma_z}(-1)^{n_1'n_2'}\\
&\quad\times\langle|\tilde{Z}_1(n_1\Gamma_1-\Gamma_{1z})\tilde{Z}_2(n_2\Gamma_2-\Gamma_{2z})F_1(m_1\Omega_1\Omega_{1z})F_2(m_2\Omega_2\Omega_{2z})Z_1(n_1'\Gamma_1'\Gamma_{1z}')Z_2(n_2'\Gamma_2'\Gamma_{2z}')|\rangle\\
&\quad=(-1)^{\Gamma_1+\Gamma_2-\Gamma_z}(-1)^{n_1'n_2'+m_1n_2+n_1'm_2+n_1'n_2}\\
&\qquad\times\langle|\tilde{Z}_1(n_1\Gamma_1-\Gamma_{1z})F_1(m_1\Omega_1\Omega_{1z})Z_1(n_1'\Gamma_1'\Gamma_{12}')\tilde{Z}_2(n_2\Gamma_2-\Gamma_{2z})F_2(m_2\Omega_2\Omega_{2z})Z_2(n_2'\Gamma_2'\Gamma_{2z}')|\rangle\\
&\quad=(-1)^{\Gamma_1+\Gamma_2-\Gamma_z}(-1)^{m_1n_2}\langle|\tilde{Z}_1(n_1\Gamma_1-\Gamma_{1z})F_1(m_1\Omega_1\Omega_{1z})Z_1(n_1'\Gamma_1'\Gamma_{1z}')|\rangle\\
&\qquad\times\langle|\tilde{Z}_2(n_2\Gamma_2-\Gamma_{2z})F_2(m_2\Omega_2\Omega_{2z})Z_2(n_2'\Gamma_2'\Gamma_{2z}')|\rangle\\
&\quad=(-1)^{m_1n_2}\langle n_1\Gamma_1\Gamma_{1z}|F_1(m_1\Omega_1\Omega_{1z})|n_1'\Gamma_1'\Gamma_{1z}'\rangle\langle n_2\Gamma_2\Gamma_{2z}|F_2(m_2\Omega_2\Omega_{2z})|n_2'\Gamma_2'\Gamma_{2z}'\rangle\\
&\quad=(-1)^{m_1n_2}(-1)^{2\Omega_1}\frac{\langle\Gamma_1'\Gamma_{1z}'\Omega_1\Omega_{1z}|\Gamma_1\Gamma_{1z}\rangle}{(2\Gamma_1+1)^{1/2}}\langle n_1\Gamma_1\|F_1(m_1\Omega_1)\|n_1'\Gamma_1'\rangle\\
&\qquad\times(-1)^{2\Omega_2}\frac{\langle\Gamma_2'\Gamma_{2z}'\Omega_2\Omega_{2z}|\Gamma_2\Gamma_{2z}\rangle}{(2\Gamma_2+1)^{1/2}}\langle n_2\Gamma_2\|F_2(m_2\Omega_2)\|n_2'\Gamma_2'\rangle.
\end{aligned}\tag{33}$$

Realizing that $2\Omega+2\Omega_1+2\Omega_2$ is always even and that

$$\begin{aligned}
&\sum_{\text{all } z\text{'s}}\langle\Gamma'\Gamma_z'\Omega\Omega_z|\Gamma'\Gamma_z'\Omega\Omega_z|\Gamma\Gamma_z\rangle\langle\Gamma_1\Gamma_{1z}\Gamma_2\Gamma_{2z}|\Gamma\Gamma_z\rangle\langle\Omega_1\Omega_{1z}\Omega_2\Omega_{2z}|\Omega\Omega_z\rangle\langle\Gamma_z'\Gamma_{1z}'\Gamma_2'\Gamma_{2z}'|\Gamma'\Gamma_z\rangle\\
&\quad\times\langle\Gamma_1'\Gamma_{1z}'\Omega_1\Omega_{1z}|\Gamma_1\Gamma_{1z}\rangle\langle\Gamma_2'\Gamma_{2z}'\Omega_2\Omega_{2z}|\Gamma_2\Gamma_{2z}\rangle\\
&\quad=(2\Gamma+1)[(2\Omega+1)(2\Gamma'+1)(2\Gamma_1+1)(2\Gamma_2+1)]^{1/2}\begin{Bmatrix}\Gamma_1\Gamma_2\Gamma\\ \Gamma_1'\Gamma_2'\Gamma'\\ \Omega_1\Omega_2\Omega\end{Bmatrix},
\end{aligned}$$

one obtains after insertion of eq. (33) in relation (31) the desired general reduction formula:

$$\begin{aligned}
\langle n\Gamma\|[F_1(m_1\Omega_1)\times F_2(m_2\Omega_2)]^{\Omega}\|n'\Gamma'\rangle&=(-1)^{m_1n_2}[(2\Gamma+1)(2\Omega+1)(2\Gamma'+1)]^{1/2}\begin{Bmatrix}\Gamma_1\Gamma_2\Gamma\\ \Gamma_1'\Gamma_2'\Gamma'\\ \Omega_1\Omega_2\Omega\end{Bmatrix}\\
&\quad\times\langle n_1\Gamma_1\|F_1(m_1\Omega_1)\|n_1'\Gamma_1'\rangle\langle n_2\Gamma_2\|F_2(m_2\Omega_2)\|n_2'\Gamma_2'\rangle.
\end{aligned}\tag{34}$$

2.4.2. Discussion of the general reduction formula

The recurrent expression (34) derived in the previous section is the pivot on which program RITSSCHIL hinges. It is good to notice the generality of this formula. Because of the consequently used direct-product notation also in this formula each factor should be interpreted as a product of such factors, one for each of the used occupation numbers and vectorial quantum numbers, respectively. So this formula is applicable without any change also in a proton–neutron formalism or in an LST-coupling scheme.

A second point is the arbitrary coupling order of shells, the coupling tree. In eq. (34) shell 1 may be an already coupled group of single shells, but also shell 2 or both. The only relevant restriction is that shell 1 and shell 2 have to be disjunct, which is used in the derivation of (34).

In the present approach the calculation of the multishell matrix elements of an arbitrary shell-model operator in an arbitrary configuration space will involve the following steps. It is assumed that the basis states and the standard operators of single-shell spaces are available. Also the single-shell matrix elements, i.e. the matrix elements of the single-shell standard operators between single-shell basis states, should be calculated in advance (which can be done using eqs. (19) and (21) and an intermediate-state expansion). Then a multishell basis of states can be constructed by iterative application of eq. (9). In this way each multishell basis state is fully defined by its constituent single-shell states and the ranks of the (intermediate) couplings. For each (intermediate) multishell a basis of standard operators can be defined in the same way with eq. (28). By iterative use of eqs. (34) and (30) the multishell matrix elements of standard operators between basis states can be calculated. Finally, eq. (29) offers the possibility to transform these matrix elements to the matrix elements of the shell-model operator one wishes to use. In the program this last transformation is combined with transformation (30), so for each shell coupling only one transformation has to be performed.

3. Set-up of program RITSSCHIL

The program RITSSCHIL is meant to be a very general shell-model code for the calculation of reduced matrix elements of tensorial operators in a multishell configuration space. The user should be able to choose his own set of quantum numbers, to specify the active shells and the order in which they should be coupled together (the coupling tree), and to define the shell-model basis states and the operators he wants to use. To be suitable for an arbitrary number of active shells and for an arbitrary coupling tree the program is set up as a recurrent two-shell code. Each constructed two-shell configuration space is treated again as a new single shell that can be used for further coupling. In order to define any symmetric tensorial shell-model operator, the elementary matrix elements are introduced: each operator is defined by its reduced matrix elements for a complete basis of final and initial states that contain the minimum number of particles for which (a part of) the operator does not yield a trivial zero (see also section 2.3). For example the usual Hamiltonian is defined by its one-body and two-body matrix elements.

The program RITSSCHIL has a modular set-up in order to make it easier for the user to understand the program or to modify it for special purposes. As an aid to a good understanding the set-up of the program is discussed in this section; some terms are introduced and some rather complicated parts are explained.

Section 3.1 surveys the construction of the basis states and operators. In section 3.2 the operator transformations are explained. The principles for the calculation of the matrices are discussed in section 3.3. Finally, section 3.4 contains remarks on the parameters of the program, the error messages and the hardware dependence.

3.1. The construction of the bases

Having defined the active shells and their coupling order (the coupling tree) one must construct for each single shell and each (intermediate) multishell three bases, namely a basis

- for the states that will be used as initial and final states in the matrix elements;
- for the standard operators into which the physical operators are expanded (see section 2.3);
- for the state operators (and implicitly also for the adjoint state operators) in terms of which the standard operators are constructed (see section 2.3).

These three bases are referred to as the state basis, the operator basis and the state-operator basis, respectively. The state-operator basis defines the states that are also used in the definition of the elementary matrix elements of the physical operators one wants to use.

In section 3.1.1 the treatment of quantum numbers and the method of basis definition are introduced. The way by which the basis constructions are controlled is described in section 3.1.2. The selection of single-shell bases is discussed in section 3.1.3 and the construction of multishell bases in section 3.1.4.

3.1.1. Space lists and extended space lists

The basis states, basis operators and basis state operators (for shortness referred to as items) are defined by a set of quantum numbers and an additional index counting the items with the same values of the quantum numbers. Because of their different behaviour with respect to tensor coupling the quantum numbers are divided into four groups: (i) the 'occupation numbers' determining the number of particles (for operators the change in the number of particles) for each type of particles; (ii) the parity or parities of selected groups of particles; (iii) the vectorial quantum numbers giving the multiplicities of the tensorial ranks; (iv) the additive quantum numbers. Group (ii) may be empty and the first element of group (iv) is the quantum number S being the total number of particles. All basis states, basis operators or basis state operators with the same values for the quantum numbers form together a space. The number of items in a space is called the dimension of the space. E.g. the dimension of the space of $J = 0$ two-particle states for the p-shell, which is constituted by the $p_{1/2}$-shell and the $p_{3/2}$-shell, is equal to two, since there are two such states. Because for further coupling only the possible sets of quantum numbers and the number of items contributing to each set are relevant, the main information about a basis can be summarized by giving a list of constructed spaces with their dimensions. This list is called the space list.

The remaining information about the bases, namely the inner structure of the different items in a space hidden in the additional index, is fully determined by specifying for each item its parents, i.e. the two states or state operators that are coupled together to get this item. E.g. one of the two $J = 0$ two-particle states for the p-shell is a two-particle state in the $p_{1/2}$-shell coupled to an empty state in the $p_{3/2}$-shell, while for the other two-particle state it is just the other way around. If an item of space PA (father) and an item of space MA (mother) can couple to get an item of space JR (junior) also all other items of spaces PA and MA can couple to get items of space JR, because they all have the same quantum numbers. So the coupling of space PA and space MA yields a number of items equal to the product of their dimensions. If one puts all these items in a fixed sequence one can specify all their additional indices using only one constant: let jr be the index of a resulting item, pa the index of an item in PA and ma the index of an item in MA, then one may define

$$\mathrm{jr} = \mathrm{C(JR,\ PA,\ MA)} + (\mathrm{pa} - 1)\mathrm{N_{MA}} + \mathrm{ma}, \tag{35}$$

in which $\mathrm{N_{MA}}$ is the dimension of space MA and C(JR, PA, MA) is a constant depending only on the three spaces concerned. C(JR, PA, MA) is called the cumulative dimension, because it counts the number of preceding items of space JR resulting from the coupling of parent space combinations different from PA and MA. So the inner structure of the items in a space can be specified by giving a list of combinations of parent spaces with their cumulative dimensions for this space. This list is called the extended space list. Because the adjoint state operators are implicitly defined at the definition of the state operators (according to eq. (12)), they have the same space numbering and indices as the state operators. In the extended space lists a parent conjugation code NC is added denoting whether the state operators or the adjoint state operators are meant:

$$\mathrm{NC} = 0 \text{ means } [\mathrm{PA} \times \mathrm{MA}], \quad \mathrm{NC} = 1 \text{ means } [\mathrm{PA} \times \tilde{\mathrm{M}}\mathrm{A}], \quad \mathrm{NC} = 2 \text{ means } [\tilde{\mathrm{P}}\mathrm{A} \times \mathrm{MA}]. \tag{36}$$

Finally, one also needs a basis for the coupled operator products $[F_1(m_1\Omega_1S_1\beta_1) \times F_2(m_2\Omega_2S_2\beta_2)]^{\Omega}_{\Omega}$, (see e.g. eq. (30)). Because the quantum numbers of these products are the same as those for the standard operators one may use for both the same space lists. Only the dimensions may be different for the operator products, but this is not important because the operator products are not used for further coupling. So one needs only the extended space lists for the operator products to specify their inner inner structure. Note that in these extended space lists the spaces PA and MA are operator spaces instead of state spaces. The conjugation code NC is used here also because all operators that can be obtained by Hermitian conjugation of other operators are not constructed explicitly. In fact this is the only reason why the conjugation code has been introduced, because for the other extended space lists the conjugation of the parents is already determined by the kind of the list (NC = 0 for states and state operators, NC = 1 for standard operators).

3.1.2. Restrictions for the bases

To enable the program and the user to restrict the construction of the bases to only those items that are really necessary, the system of ranges is introduced. For each single shell or multishell and for each of the three types of bases there are one or more sets of ranges, each set giving a lower limit and an upper limit for each of the defined quantum numbers. The values of the quantum numbers of a constructed item should meet all the requirements given by at least one of the sets of ranges for the shell and list type concerned. The ranges are initialized to default values depending on the kind of the quantum number concerned and can be modified by the user. Before the basis constructions are started each set of ranges is refined as much as possible and reasonable with the strong correlations between the restrictions for different quantum numbers, shells or list types. All items are used to construct items in the complete configuration space. Hence, next to the ranges mentioned (specific for one shell and one basis type), for each basis type there are also common ranges. The latter are derived from the restrictions for the final configuration space but valid for each intermediate shell. These common ranges imply that each intermediate item should either have a value for quantum numbers S (see section 3.1.1) that is lower than the maximum value of S of the final items, or meet all the requirements given by at least one of the sets of ranges for the final items. For operators the common ranges also take care that, if the final operators are pure particle-transfer operators (so $F = Z$), the same holds for each intermediate operator.

3.1.3. Selection of single-shell spaces

It is assumed that for each single shell the bases have been constructed in advance and that the information about these bases is available as space lists on a file. To define the inner structure of the single-shell standard operators the file should contain also extended space lists for the operator spaces. The inner structure of states and state operators is fully defined by the c.f.p.'s in the single-shell matrix elements.

The program has to select from the file those spaces that meet the conditions given by the sets of ranges for the single shells concerned. To avoid a resequencing of the single-shell matrix element labels later on in the program (see section 3.3) only complete spaces of single-shell items are selected. So if one truncates an operator space by prohibiting some of its constituent state operators, the complete operator space is omitted.

The set of quantum numbers on the library file, used to define the bases, has to be roughly the same as the set the user has chosen for the program. The possible conversions are that the program may omit some quantum numbers, it may split up a quantum number into several quantum numbers concerning the different types of particles (e.g. n becomes n_{p} or n_{n}), and it may add a parity quantum number. Because in this part of the program the single-shell quantum numbers are defined, one has to interfere in these routines if one wishes to define single-shell quantum numbers in a different way, e.g. to define an additive quantum number containing the number of oscillator quanta for harmonic-oscillator eigenstates.

3.1.4. Coupling and trimming

Each basis in a (intermediate) multishell is a direct product of two already constructed bases. For the construction of a multishell basis one has only to find the possible quantum numbers to which the two constituent bases can couple. The Racah algebra for the coupling is handled later on in the program in the general reduction formula of the matrix calculation (see section 2.4.1 and 3.3). When coupling two tensors most of the quantum numbers of the product are fully determined by the quantum numbers of the two tensors; only in the resultants of the vectorial ranks there is some freedom, which causes that mostly more than one product can be constructed.

In addition to the ranges restricting the bases an extra restriction is implemented for the operators: the value of the occupation number should be nonnegative (or, in case there are several occupation quantum numbers, the first nonzero value should be positive). This restriction is to prevent the explicit construction of operators that can be seen as Hermitian conjugates of other operators in the same basis, because the matrix elements of such operators can be derived easily from those of the other operators (by the use of eq. (23)).

Having constructed a basis for the operators, one can use its space list for the construction of the extended space lists for the operator products of eq. (30). It is tested whether the four state operators of the parent shells, appearing in eq. (30), are also used to construct a standard operator with the same quantum numbers as this operator product. If not, this operator product is not necessary for the calculations and it is omitted. But also the other way around: each Z-quartet in a standard operator, that is not used for the construction of an operator product, should be removed.

During and after the basis constructions all spaces that are not used for further coupling are eliminated, in order to avoid the calculation of superfluous matrix elements later on in the program.

3.2. *The operator transformation*

The general reduction formula (34) yields reduced matrix elements of the coupled product of two standard operators. For further use these operator products have to be transformed to standard operators in the combined shell with the operator transformation (30). To get physical operators (if opted) the standard operators have to be transformed too by the use of the elementary matrix of the physical operators as described by eq. (26). Both transformations can be taken together into one transformation from the coupled operator product to the desired final operators. All matrix elements calculated with the general reduction formula are converted with this combined transformation to the matrix desired for the shell concerned.

In this section the calculation of the transformation matrix elements is reviewed. The transformation of operator products to standard operators is discussed in section 3.2.1 and the contraction of standard operators to physical operators in section 3.2.2.

3.2.1. *The transformation to standard operators*

In the matrix calculation part of the program RITSSCHIL the multishell matrix elements are calculated with the general reduction formula (34). This formula yields matrix elements of coupled products of two standard operators, one acting in one of the two shells and one acting in the other shell. For further calculations these matrix elements have to be transformed to matrix elements of standard operators in the composite multishell with eq. (30). Because this transformation formula is straightforward the values of the transformation matrix elements can be calculated easily. The problem is hidden in the additional indices of the operator products and the standard operators. In fact the transformation is just a reordering and recoupling of the four constituent parent state operators (the Z-quartet). Hence each operator product is transformed to only those standard operators that have the same constituent Z-quartet.

Let the standard operator with space index JO and additional index jo, denoted as (JO · jo), be a coupled product of state operator (JA · ja) and adjoint state operator (JB · jb). Then, according to eq. (35), one has

$$\mathrm{jo} = \mathrm{C(JO, JA, JB)} + (\mathrm{ja} - 1)\mathrm{N}_{\mathrm{JB}} + \mathrm{jb}. \qquad (37)$$

State operators (JA · ja) and (JB · jb) are themselves coupled products of the state operators (PA · pa) and (MA · ma), and (PB · pb) and (MB · mb), respectively, so one also has

$$\mathrm{ja} = \mathrm{C(JA, PA, MA)} + (\mathrm{pa} - 1)\mathrm{N_{MA}} + \mathrm{ma}, \quad \mathrm{jb} = \mathrm{C(JB, PB, MB)} + (\mathrm{pb} - 1)\mathrm{N_{MB}} + \mathrm{mb}. \tag{38}$$

Insertion of eq. (38) into eq. (37) yields

$$\begin{aligned}\mathrm{jo} = {} & \mathrm{C(JO, JA, JB)} + \mathrm{N_{JB}C(JA, PA, MA)} + \mathrm{C(JB, PB, MB)} + (\mathrm{pa} - 1)\mathrm{N_{MA}N_{JB}} + (\mathrm{ma} - 1)\mathrm{N_{JB}} \\ & + (\mathrm{pb} - 1)\mathrm{N_{MB}} + \mathrm{mb}.\end{aligned} \tag{39}$$

The operator product has the same quantum numbers as the standard operator (JO · jo), so it has also the same space index JO. Its additional index jt is different, however. Depending on the parent conjugation code NC (see section 3.1.1) the operator product (JO · jt) has one of the following structures:

$$\begin{aligned}\mathrm{NC} = 0: &\quad (\mathrm{JO} \cdot \mathrm{jt}) = \left[(\mathrm{PO} \cdot \mathrm{po}) \times (\mathrm{MO} \cdot \mathrm{mo})\right], \\ \mathrm{NC} = 1: &\quad (\mathrm{JO} \cdot \mathrm{jt}) = \left[(\mathrm{PO} \cdot \mathrm{po}) \times (\mathrm{MO} \,\tilde{\cdot}\, \mathrm{mo})\right], \\ \mathrm{NC} = 2: &\quad (\mathrm{JO} \cdot \mathrm{jt}) = \left[(\mathrm{PO} \,\tilde{\cdot}\, \mathrm{po}) \times (\mathrm{MO} \cdot \mathrm{mo})\right].\end{aligned}$$

With the same method as used in the derivation of eq. (39) one can find for its additional index

$$\begin{aligned}\mathrm{jt} = {} & \mathrm{C(JO, PO, MO)} + \mathrm{N_{MO}C(PO, PA, PB)} + \mathrm{C(MO, MA, MB)} + 1 + (\mathrm{pa} - 1)\mathrm{KPA} \\ & + (\mathrm{ma} - 1)\mathrm{KMA} + (\mathrm{pb} - 1)\mathrm{KPB} + (\mathrm{mb} - 1)\mathrm{KMB}\end{aligned} \tag{40}$$

with KPA, KMA, KPB and KMB as in table 1. So the transformation matrix elements have to contain the stipulation that an operator product with index jt as defined by eq. (40) contributes only to a standard operator with the same quantum numbers and with index jo as defined by eq. (39) with the same pa, ma, pb and mb indices.

For single shells the matrix elements of standard operators are read from a file. Hence there is no transformation needed. But for conformity and because also for single shells a contraction may be performed, for singe shells a dummy transformation matrix is calculated.

To be able to perform a conjugation of the multishell matrix elements as described by eq. (23), a conjugation phase factor is added to each matrix element. The operator-dependent part of this conjugation phase factor $(-1)^{\Gamma_b - \Gamma_a}$ is calculated in this part of the program and transferred via the transformation matrix elements, so that in the multiplication part of the program the inner structure of the operator has not to be analysed.

3.2.2. The contraction

In section 2.3 it is shown that physical operators can be written as linear combinations of standard operators. The coefficients of these expansions are the elementary matrix elements of the physical operators. Often the elementary matrix elements also depend on a number of parameters.

The standard option of the program is to calculate the multishell matrix elements of the standard operators that appear in the expansion. In this way for each physical multishell matrix element the

Table 1
Parameters for the calculation of jt (see text)

	KPA	KMA	KPB	KMB
NC = 0	$\mathrm{N_{PB}N_{MO}}$	$\mathrm{N_{MB}}$	$\mathrm{N_{MO}}$	1
NC = 1	$\mathrm{N_{PB}N_{MO}}$	1	$\mathrm{N_{MO}}$	$\mathrm{N_{MA}}$
NC = 2	$\mathrm{N_{MO}}$	$\mathrm{N_{MB}}$	$\mathrm{N_{PA}N_{MO}}$	1

dependence on the separate elementary matrix elements is conserved explicitly. So the user may adjust all elementary matrix elements independently after the geometrical calculations of program RITSSCHIL.

If one does not wish to adjust all elementary matrix elements independently, one may ask the program to perform a contraction. In this case the program performs the transformation of the matrix elements of standard operators to the matrix elements of the physical operators as described by eq. (26). This contraction will be performed if the concerned elementary matrix elements are offered to the program on a file. Because the elementary matrix can be written on this file as linear combinations of a number of parameters, the dependence on the different parameters can be kept separate. So the user may adjust all parameters after program RITSSCHIL has finished.

Although the standard operators in eq. (29) are standard operators in the complete configuration space, the contraction has not to be delayed till after the last shell coupling. As soon as an operator in an intermediate shell has a value for quantum number S that is equal to the maximum value for S in the complete configuration space, it is certain that this operator will be coupled only with unit operators in the remaining shell couplings. At this moment it already has the same quantum numbers as the final operator and it will not be used to construct other operators, so it can be contracted. This contraction changes the dimensions of several operator spaces. Hence the contraction forces a revision of space lists and extended space lists.

For Hermitian physical operators one has

$$\langle n'\Gamma'\alpha' \| O \| n\Gamma\alpha \rangle = (-1)^{\Gamma-\Gamma'} \langle n\Gamma\alpha \| O \| n'\Gamma'\alpha' \rangle. \tag{41}$$

So for contracted operators the operator-dependent part of the conjugation phase factor of the matrix element is put equal to one.

3.3. *The matrix calculation*

The most time-consuming part of the program is the matrix evaluation. To save time and space RITSSCHIL calculates only those matrix elements for which the initial state equals the final state or precedes it in the ordering of the basis states. All other matrix elements can be obtained by interchanging the initial and final state with eq. (23) or (41). In order to be able to perform such a conjugation a phase factor is added to each matrix element giving the change of sign at conjugation. For number-conserving standard operators this conjugation implies that the operator in the matrix element is replaced by an operator with the same quantum numbers, so with the same space index, but in most cases with a different additional index. Hence, if in the program conjugations have to be performed, an inversion table should be available with the change in additional indices for number-conserving operators. Operators that are not number-conserving are replaced by operators that are not explicitly defined at the basis construction. It can be assumed (as is done already in section 3.2.1) that the additional index of nonnumber-conserving operators is not changed at conjugation.

In the approach of the program RITSSCHIL it is assumed that for single-shell spaces all needed matrix elements are offered to the program on a file. The program has to read those matrix elements and to adjust the space indices of the final state, initial state and operator to the space definitions used in the program. Because only complete spaces have been selected (see section 3.1.3) the additional indices have not to be adjusted and the inner structure of the three items is not to be analysed. Maybe a contraction should be performed, so also the single-shell matrix elements pass through a transformation routine. The gathered single-shell matrix elements are finally written to temporary direct-access files. Each shell has its own direct-access file. It is opened and written for every shell in turn, according to the coupling tree, and it is closed as soon as it has been used for further coupling.

For multishells the matrix elements are calculated with the general reduction formula (34) followed by a transformation as discussed in section 3.2. First the required information for the multishell concerned is made available: the transformation matrix elements, the temporary direct-access files with the matrix elements of the two constituent shells, the needed space lists and extended space lists, and some tables. For each final and initial state space a list is made of operator spaces that fit in between. For this selection attention is paid not only to quantum number conditions ($n_f = n_i + n_0$, triangle conditions, etc.), but also to possible future use of the matrix elements. As soon as in a matrix element on an intermediate multishell the final and initial states contain already the maximum number of particles desired for the complete configuration space, they will be coupled only with empty states in the remaining shell couplings. Because only the unit operator fits between empty states, the operator in the matrix element concerned will be coupled only with the unit operator in the remaining shell couplings. So the operator should already have the quantum numbers as desired for the operators in the complete configuration space.

The application of the general reduction formula is straightforward. The multiplication factor in the formula depends only on the quantum numbers of the nine states and operators concerned, so it can be calculated once for all matrix elements with the same 'junior' and 'parent' spaces. The most complicated part of the matrix calculation is the multiplication of the two 'parent' matrix elements. Both the search for the two 'parent' matrix elements on the two (possibly rather large) direct-access files and the calculation of the proper indices for the 'junior' matrix elements need a large amount of administration. The calculated matrix elements are finally transformed into matrix elements of standard operators or physical operators and are written again to a temporary direct-access file.

3.4. Final remarks

The program RITSSCHIL has been developed at a Cyber 175-100 computer. To minimize the dependence on the hardware and to facilitate the adaptation of the programme to special calculations, extensive use is made of the possibility in FORTRAN77 to define parameters. Almost all array dimensions and all operations that have to do with the word length of the machine are defined by parameters, which can be set easily by the user.

To save space or to enable a fast comparison, several times some information is stored in one computer word, although the information consists of more than one integer n_i. These packed labels are constructed as a sum of the contributing integers, each integer multiplied by a constant ($l = \Sigma k_i n_i$). To avoid loss of information the quotient of such a constant and the next higher constant of the same label has to be an integer number larger than the maximum value of the integer that is packed by the use of this constant ($k_{i+1}/k_i > n_i$). The proper choice of the constants depends on the size of the calculations and also on the word length of the machine on which the program runs. So all packing constants are defined with parameters. Also all array dimensions that are not completely fixed by the structure of the program are defined with parameters. The name of a parameter is the same through all of the program and it is exclusively used for this parameter. So it is easy to change a parameter value consistently.

The program has an autodiagnostic system, which takes care that the execution of the program is terminated with a message as soon as an error is detected. With the exception of the array references in the six-j routine (S6J), all array references are tested on overflow by comparing the actual maximum indices with the parameters defining the dimensions. The packing of labels is also tested, but mostly only via the consistency of the packing constants and dimension constants.

Program RITSSCHIL is a link in a chain of programs. Before the program can be run, one has to prepare appropriate files with the definition of the single shells and the single-shell matrix elements and optionally also a file with the elementary matrix elements that define the physical operators one wants to use. The output files will contain the definition of the multishell standard bases and the matrix elements of the physical operators between standard basis states. To get physically relevant quantities (binding energies, transition probabilities, spectroscopic factors, etc.) one has to transform the matrix elements on the standard basis to matrix elements on a basis of eigenstates of the Hamiltonian, obtained by a diagonalization of the Hamiltonian matrix. Hence, accompanied by appropriate preparation programs and programs for finishing off, program RITSSCHIL can be a powerful and flexible tool for microscopic shell-model calculations.

Acknowledgements

The concepts of the present approach have been developed in collaboration with Dr. G.A. Timmer. The implementation of the formalism in the computer code has been performed with assistance of Dr. B.C. Metsch and Dr. A.G.M. van Hees.

This work was performed as part of the research programme of the "Stichting voor Fundamenteel Onderzoek der Materie" (FOM) with financial support from the "Nederlandse Organisatie voor Zuiver-Wetenschappelijk Onderzoek" (ZWO).

References

[1] J.B. French, E.C. Halbert, J.B. McGrory and S.S.M. Wong, Advan. Nucl. Phys. 3 (1969) 193.

[2] P.J. Brussaard and P.W.M. Glaudemans, Shell-model Applications in Nuclear Spectroscopy (North-Holland, Amsterdam, 1977).

[3] D. Zwarts, RITSSCHIL (Internal publ. R.J. Van de Graaff Laboratory, Utrecht, 1984).

TEST RUN OUTPUT

```
PROGRAM  G E N E S I S
----------------------

QUNUM [J], [T]
      [J] / 16       [T] /  2
SHELL 1/2, 3/2

   6 STATE SPACES OF SHELL  1/2
 IDX . DIMENSION      N  [J]  [T]    S    V
   1 .         1 :    0   1    1     0    0
   2 .         1 :    1   2    2     1    1
   3 .         1 :    2   1    3     2    0
   4 .         1 :    2   3    1     2    2
   5 .         1 :    3   2    2     3    1
   6 .         1 :    4   1    1     4    0

   4 Z-OP. SPACES OF SHELL  1/2
 IDX . DIMENSION      N  [J]  [T]    S    V
   1 .         1 :    0   1    1     0    0
   2 .         1 :    1   2    2     1    1
   3 .         1 :    2   1    3     2    0
   4 .         1 :    2   3    1     2    2

  13 OPER. SPACES OF SHELL  1/2
 IDX . DIMENSION      N  [J]  [T]    S    V
   1 .         1 :    0   1    1     0    0
   2 .         1 :    0   1    1     2    2
   3 .         1 :    0   1    1     4    0
   4 .         1 :    0   1    1     4    4
   5 .         1 :    0   1    3     2    2
   6 .         1 :    0   3    1     2    2
   7 .         1 :    0   3    3     2    2
   8 .         1 :    1   2    2     1    1
   9 .         1 :    1   2    2     3    1
  10 .         1 :    1   2    2     3    3
  11 .         1 :    1   4    2     3    3
  12 .         1 :    2   1    3     2    0
  13 .         1 :    2   3    1     2    2

  SSMTAP RECORDS FOR SHELL  1/2
 LENGTH    #LABELS    FIRST     LAST
   135        18    00010001  00060006


  30 STATE SPACES OF SHELL  3/2
 IDX . DIMENSION      N  [J]  [T]    S    V
   1 .         1 :    0   1    1     0    0
   2 .         1 :    1   4    2     1    1
   3 .         1 :    2   1    3     2    0
   4 .         1 :    2   3    1     2    2
   5 .         1 :    2   5    3     2    2
   6 .         1 :    2   7    1     2    2
   7 .         1 :    3   2    2     3    3
   8 .         1 :    3   4    2     3    1
   9 .         1 :    3   4    4     3    1
  10 .         1 :    3   6    2     3    3
  11 .         1 :    3   8    2     3    3
  12 .         1 :    4   1    1     4    0
  13 .         1 :    4   1    5     4    0
  14 .         1 :    4   3    3     4    2
  15 .         1 :    4   5    1     4    2
  16 .         1 :    4   5    1     4    4
  17 .         1 :    4   5    3     4    2
  18 .         1 :    4   7    3     4    2
  19 .         1 :    4   9    1     4    4
  20 .         1 :    5   2    2     5    3
  21 .         1 :    5   4    2     5    1
  22 .         1 :    5   4    4     5    1
  23 .         1 :    5   6    2     5    3
  24 .         1 :    5   8    2     5    3
  25 .         1 :    6   1    3     6    0
  26 .         1 :    6   3    1     6    2
  27 .         1 :    6   5    3     6    2
  28 .         1 :    6   7    1     6    2
  29 .         1 :    7   4    2     7    1
  30 .         1 :    8   1    1     8    0

   6 Z-OP. SPACES OF SHELL  3/2
 IDX . DIMENSION      N  [J]  [T]    S    V
   1 .         1 :    0   1    1     0    0
   2 .         1 :    1   4    2     1    1
   3 .         1 :    2   1    3     2    0
   4 .         1 :    2   3    1     2    2
   5 .         1 :    2   5    3     2    2
   6 .         1 :    2   7    1     2    2

  22 OPER. SPACES OF SHELL  3/2
 IDX . DIMENSION      N  [J]  [T]    S    V
   1 .         1 :    0   1    1     0    0
   2 .         1 :    0   1    1     2    2
   3 .         1 :    0   1    1     4    0
   4 .         3 :    0   1    1     4    4
   5 .         1 :    0   1    3     2    2
   6 .         1 :    0   3    1     2    2
   7 .         1 :    0   3    3     2    2
   8 .         1 :    0   5    1     2    2
   9 .         1 :    0   5    3     2    2
  10 .         1 :    0   7    1     2    2
  11 .         1 :    0   7    3     2    2
  12 .         2 :    1   2    2     3    3
  13 .         1 :    1   4    2     1    1
  14 .         1 :    1   4    2     3    1
  15 .         3 :    1   4    2     3    3
  16 .         3 :    1   6    2     3    3
  17 .         2 :    1   8    2     3    3
  18 .         1 :    1  10    2     3    3
  19 .         1 :    2   1    3     2    0
  20 .         1 :    2   3    1     2    2
  21 .         1 :    2   5    3     2    2
  22 .         1 :    2   7    1     2    2

  SSMTAP RECORDS FOR SHELL  3/2
 LENGTH    #LABELS    FIRST     LAST
  1023        92    00010001  00170013
  1023        89    00170014  00260013
  1023        82    00260015  00340014
   327        30    00340016  00360036
```

```
PROGRAM  R I T S S C H I L
--------------------------


INPUT CARDS:

SHELL OP3
SHELL OP1
SHELL P
STATE P3
OPER. M1


SHELL NAME FAMILY TYPE    PA     MA     JR

   1   OP3      P            0      0      3
   2   OP1      P            0      0      3
   3     P    -              1      2      0


STATE RANGES
NAMSHL       N          P        [J]        [T]         S
  OP3     0 :  3    0 :  1    1 : 99    1 : 99    0 :  3
  OP1     0 :  3    0 :  1    1 : 99    1 : 99    0 :  3
    P     3 :  3    0 :  1    1 : 99    1 : 99    3 :  3


Z-OP. RANGES
NAMSHL       N          P        [J]        [T]         S
  OP3     0 :  1    0 :  1    1 : 99    1 : 99    0 :  1
  OP1     0 :  1    0 :  1    1 : 99    1 : 99    0 :  1
    P     1 :  1    0 :  1    1 : 99    1 : 99    1 :  1


OPER. RANGES
NAMSHL       N          P        [J]        [T]         S
  OP3     0 :  2    0 :  1    1 : 99    1 : 99    0 :  2
  OP1     0 :  2    0 :  1    1 : 99    1 : 99    0 :  2
    P     0 :  0    0 :  0    3 :  3    1 :  3    2 :  2


 11 STATE SPACES OF SHELL  OP3
IDX . DIMENSION      N    P  [J]  [T]    S
  1 .          1 :   0    0    1    1    0
  2 .          1 :   1    1    4    2    1
  3 .          1 :   2    0    1    3    2
  4 .          1 :   2    0    3    1    2
  5 .          1 :   2    0    5    3    2
  6 .          1 :   2    0    7    1    2
  7 .          1 :   3    1    2    2    3
  8 .          1 :   3    1    4    2    3
  9 .          1 :   3    1    4    4    3
 10 .          1 :   3    1    6    2    3
 11 .          1 :   3    1    8    2    3

  2 Z-OP. SPACES OF SHELL  OP3
IDX . DIMENSION      N    P  [J]  [T]    S
  1 .          1 :   0    0    1    1    0
  2 .          1 :   1    1    4    2    1
```

```
   4 OPER. SPACES OF SHELL  OP3
 IDX . DIMENSION      N    P  [J]  [T]    S
   1 .         1 :    0    0    1    1    0
   2 .         1 :    0    0    3    1    2
   3 .         1 :    0    0    3    3    2
   4 .         1 :    1    1    4    2    1



   5 STATE SPACES OF SHELL  OP1
 IDX . DIMENSION      N    P  [J]  [T]    S
   1 .         1 :    0    0    1    1    0
   2 .         1 :    1    1    2    2    1
   3 .         1 :    2    0    1    3    2
   4 .         1 :    2    0    3    1    2
   5 .         1 :    3    1    2    2    3

   2 Z-OP. SPACES OF SHELL  OP1
 IDX . DIMENSION      N    P  [J]  [T]    S
   1 .         1 :    0    0    1    1    0
   2 .         1 :    1    1    2    2    1

   4 OPER. SPACES OF SHELL  OP1
 IDX . DIMENSION      N    P  [J]  [T]    S
   1 .         1 :    0    0    1    1    0
   2 .         1 :    0    0    3    1    2
   3 .         1 :    0    0    3    3    2
   4 .         1 :    1    1    2    2    1



   7 STATE SPACES OF SHELL     P
 IDX . DIMENSION      N    P  [J]  [T]    S
   1 .         5 :    3    1    2    2    3
   2 .         1 :    3    1    2    4    3
   3 .         5 :    3    1    4    2    3
   4 .         3 :    3    1    4    4    3
   5 .         4 :    3    1    6    2    3
   6 .         1 :    3    1    6    4    3
   7 .         2 :    3    1    8    2    3

   2 Z-OP. SPACES OF SHELL     P
 IDX . DIMENSION      N    P  [J]  [T]    S
   1 .         1 :    1    1    2    2    1
   2 .         1 :    1    1    4    2    1

   2 OPER. SPACES OF SHELL     P
 IDX . DIMENSION      N    P  [J]  [T]    S
   1 .         4 :    0    0    3    1    2
   2 .         4 :    0    0    3    3    2

THE MINIMAL NEEDED VALUES OF SOME PARAMETERS:
MOM:*****   MRN:    12   MSP:    13   MSX:     5   MTR:     8   MZ4:     4
```

```
MATRIX LIST OF SHELL  1 =  OP3 = 0 * 0
(LYFTAP NUMBER: 31)
FINAL SPACE  RECORDCOUNT  RECORDLENGTH
      1            1            7
      2            1           19
      3            1           27
      4            1           41
      5            1           57
      6            1           71
      7            1           87
      8            1          115
      9            1          139
     10            1          167
     11            1          187

MATRIX LIST OF SHELL  2 =  OP1 = 0 * 0
(LYFTAP NUMBER: 32)
FINAL SPACE  RECORDCOUNT  RECORDLENGTH
      1            1            7
      2            1           19
      3            1           27
      4            1           41
      5            1           57

MATRIX LIST OF SHELL  3 =    P = 1 * 2
(LYFTAP NUMBER: 33)
FINAL SPACE  RECORDCOUNT  RECORDLENGTH
      1            1           47
      2            1           61
      3            1          199
      4            1          281
      5            1          411
      6            1          465
      7            1          515

THE MINIMAL NEEDED VALUES OF SOME PARAMETERS:
MIV:    0   MJO:    2   MME:    8   MSI:    5   MSO:   11
```

```
SURVEY OF LYFTAP

           1   P
         515              0          28679          20
        4097             48
 34363965448 1.73205080756880        34363998216-2.99999999999980        68723703833-1.41421356237298
 68723736601-2.44948974278299        68727898120-1.15470053837917        68727898144 .912870929175199
 68727930912 1.58113883008402       103087636505 1.22474487139151       103087669273 .707106781186514
103091830792 1.73205080756877       103091863560-.999999999999954       137447374873-.912870929175199
137447407641 1.58113883008407       137451601952 1.41421356237296       137455763464-.577350269189594
137455763488 .730296743340165       137455796232-.999999999999954       171815501849 1.41421356237301
171815534617 2.44948974278304       171819696160 .547722557505168       171819728928 2.84604989415118
        8193             56
 34368192536 1.99999999999986        34372386825-2.82842712474603        34376581153-1.99999999999979
        8194             62
 34363965448 2.44948974278302        34363998216 3.16227766016817
       12289            132
 34363965464-2.12132034355955        34363998232 1.22474487139154        34368192521-1.99999999999982
 34372354064-.612372435695764        34372386832-1.76776695296621        34376548368 1.36930639376282
 34376581136 .790569415042050        68723703832 1.58113883008406        68723736600 2.73861278752560
 68727898121-2.58198889747143        68727898145 .816496580927652        68727930913 1.41421356237293
 68732092432 1.36930639376283        68732125200 .790569415042054        68736286736 .204124145231900
 68736319504-.353553390593222       103087636504-.645497224367833       103087669272 1.11803398874979
103091863585-1.99999999999979       103096025097-1.63299316185535       103096025121 .516397779494277
103096057865-2.82842712474603       103100219408 .499999999999968       103100252176 .866025403784388
137447374872 1.93649167310350       137447407640 1.11803398874979       137455796257-1.78885438199961
137459957776 1.49999999999984       137459990544-.866025403784359       171811307544 1.93649167310361
171811340312-1.11803398874985       171815501848-.866025403784366       171815534616-1.49999999999988
171819728929 1.54919333848274
       12290            142
 34363998225-.999999999999869        68723736593-2.23606797749963       103083474976-2.82842712474584
171802951705 3.16227766016820
       12291            200
 34363965472 1.73205080756873        34363998240-.999999999999925        68723736584 2.82842712474591
 68727898120 1.46059348668034        68727898144 1.27017059221706        68727930912 2.19999999999976
103083442201-1.22474487139151       103083474969-.707106781186507       103087636505-1.27801930084524
103087669273 2.21359436211763       103091830792 1.82574185835043       103091830816 1.15470053837914
103091863560 3.16227766016817       137443180569 1.22474487139151       137443213337 3.53553390593243
137447374873 .547722557505111       137447407641 .316227766016810       137451601952 .799999999999898
137455763464-1.09544511501025       137455763488 2.07846096908239       137455796232 .632455532033639
137455796256 2.39999999999969       171811307545-.774596669241415       171811340313-1.34164078649976
171815501849 .774596669241411       171815534617-.447213595499923       171819696160 1.73205080756878
171819728928-2.19999999999968
       16385            220
 34363998233-3.46410161513755        34368192520-2.82842712474589        34372386833-.999999999999922
 34376581137-2.23606797749962        68727930905 3.16227766016800        68736319520 2.52982212813436
 68740513809-2.44948974278290       103091863577-1.99999999999991       103100252192-3.46410161513703
       16386            230
 34363965456-1.73205080756877        34363998224-2.23606797749960       103083442200 1.73205080756880
103083474968 2.23606797749966
       16387            256
 34363998241 2.82842712474591        34368192521 3.99999999999959        34372386832-1.99999999999984
 34376581136-1.99999999999981        68723736600-1.99999999999975        68727930904 .894427190999817
 68732125217-1.13137084989832        68736319497 1.78885438199970        68736319521 1.69705627484747
 68740513808 1.26491106406723       103096057880 3.99999999999957       103100252193-1.78885438199969
       16388            282
 34363965472 2.44948974278294        34363998240 3.16227766016806        68723703833 3.46410161513744
 68723736601 4.47213595499909        68727898120-1.54919333848285        68727898144 2.93938769133941
 68727930888-1.99999999999983        68727930912 3.79473319220151       103087636505-3.46410161513737
103087669273-4.47213595499903       103091830816 2.44948974278289       103091863584 3.16227766016790
```

```
       20483                346
 34363998217 3.46410161513741        34368159753-2.68328157299952        34368159777 .848528137423770
 34368192545 1.46969384566972        34372354064-.670820393249862        34372386832 1.16189500386210
 34376548368-.670820393249858        34376581136-.387298334620702        68723703832-.612372435695750
 68723736600-1.76776695296621        68727898136 2.46475150877293        68727930904 1.42302494707561
 68732125217-2.39999999999969        68736286729-2.19089023002049        68736286753 .692820323027462
 68736319497 1.26491106406726        68736319521 .799999999999901        68740481040 .387298334620706
 68740513808-.223606797749961       103083442200 2.80624304008023       103083474968 1.62018517460185
103087636504-1.25499003980094       103087669272 2.17370651192815       103096057889-1.83303027798207
103100219408-1.77482393492969       103100252176-3.07408522978758       137451569176 1.77482393492978
137451601944 3.07408522978771       137455763480-1.77482393492963       137455796248 1.02469507659582
137459990561-1.83303027798180
       20484                368
 34363998216-4.89897948556577        34368192529 1.09544511501019        68723736601 .999999999999901
 68727930888-3.57770876399938        68727930912-.565685424949159        68732125201-1.99999999999978
103083474969 4.58257569495544       103087669280-2.59229627936273       137447407641-2.89827534923744
137451601952-4.09878030638276
       20485                412
 34363965448 4.09878030638350        34363965472 1.94422220952215        34363998240 3.36749164809615
 68723703833 2.50998007960193        68723736601 1.44913767461878        68727898120 2.04939015319177
 68727898144 2.59229627936281        68727930888-1.18321595661985        68727930912 2.99332590941877
103083442201 .547722557505089       103083474969-.948683298050394       103087669280 .489897948556564
103091830792-1.46385010942269       103091830816 3.70328039908975       103091863560-2.53546276418537
137447374873-1.89736659610076       137447407641 1.09544511501019       137451569177-1.65615734242140
137451601945-2.86854866240211       137455763488 3.24037034920333       137455796256 1.44321070632708
       24579                426
 34363998233 .999999999999876        34368192537 4.02492235949907        34372386848 3.39411254969494
 34376581128 3.57770876399938        34376581152 .565685424949159        34380775441-.632455532033610
       24580                444
 34363965464-1.73205080756872        34363998232-2.23606797749954        34368159753-3.09838667696567
 34368159777 .979795897113135        34368192521-3.99999999999964        34368192545 1.26491106406716
 34372354064-1.73205080756868        34372386832-2.23606797749953
       24581                456
 34363998232 4.09878030638330        34368192521-3.34664010613604        34368192545 2.11660104885136
 34372386849-.692820323027444        34376581136-3.09838667696550
       24582                466
 34363965448 2.89827534923765        34363965472 3.66606055596417        34363998216 3.74165738677362
 34363998240 4.73286382647896
       28677                494
 34363965464-2.44948974278283        34363998232 4.24264068711869        34368192545-2.19089023002036
 34372354057-2.61861468283170        34372354081 .828078671210733        34372386825-4.53557367611037
 34376548368-.925820099772420        34376581136-1.60356745147434        68727898136 2.99999999999979
 68727930904-1.73205080756879        68732092440-1.96396101212369        68732125208-3.40168025708263
 68736319521-4.05674042269584
       28678                500
 34363998240-3.09838667696546        68723736601 4.89897948556600
       28679                516
 34363965448 2.26778683805522        34363965472 4.30282299360337        34363998216 3.92792202424764
 68723703833-2.26778683805509        68723736601-3.92792202424741        68727898144 5.01996015920375
 68727930912 1.24211800681571
           4                  0     16777215                 1
    16777215                 48
```