Computer Physics Communications 200 (2016) 220–253

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

A computer code for calculations in the algebraic collective model of the atomic nucleus☆

T.A. Welsh*, D.J. Rowe

Department of Physics, University of Toronto, Toronto, Ontario M5S 1A7, Canada

ARTICLE INFO

Article history:
Received 6 April 2015
Received in revised form
28 September 2015
Accepted 18 October 2015
Available online 12 November 2015

Keywords:
Algebraic collective model
Bohr model
SO(5) Clebsch–Gordan coefficients

ABSTRACT

A Maple code is presented for algebraic collective model (ACM) calculations. The ACM is an algebraic version of the Bohr model of the atomic nucleus, in which all required matrix elements are derived by exploiting the model's SU(1, 1) × SO(5) dynamical group. This paper reviews the mathematical formulation of the ACM, and serves as a manual for the code.

The code enables a wide range of model Hamiltonians to be analysed. This range includes essentially all Hamiltonians that are rational functions of the model's quadrupole moments $\hat{q}_M$ and are at most quadratic in the corresponding conjugate momenta $\hat{\pi}_N$ ($-2 \leq M, N \leq 2$). The code makes use of expressions for matrix elements derived elsewhere and newly derived matrix elements of the operators $[\hat{\pi} \otimes \hat{q} \otimes \hat{\pi}]_0$ and $[\hat{\pi} \otimes \hat{\pi}]_{LM}$. The code is made efficient by use of an analytical expression for the needed SO(5)-reduced matrix elements, and use of SO(5) ⊃ SO(3) Clebsch–Gordan coefficients obtained from precomputed data files provided with the code.

Program summary

Program title: ACM

Catalogue identifier: AEYO_v1_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEYO_v1_0.html

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licensing provisions: Standard CPC licence, http://cpc.cs.qub.ac.uk/licence/licence.html

No. of lines in distributed program, including test data, etc.: 3873526

No. of bytes in distributed program, including test data, etc.: 46345414

Distribution format: tar.gz

Programming language: Maple 18 (or versions 17, 16, 15).

Computer: Any.

Operating system: Any which supports Maple; tested under Linux, Max OSX, Windows 7.

RAM: 500Mb

Classification: 17.20.

Nature of problem: The calculation of energy eigenvalues, transition rates and amplitudes of user specified Hamiltonians in the Bohr model of the atomic nucleus.

Solution method: Exploit the model's SU(1, 1) × SO(5) dynamical group to calculate analytic (as far as possible) expressions for matrix elements, making use of extensive files (supplied) of SO(5) ⊃ SO(3) Clebsch–Gordan coefficients. Diagonalisation of the resulting matrices (once the entries are converted to floating point) is carried out using the Maple library procedure `Eigenvectors`. (Maple [1] makes use of the NAG [2] and CLAPACK [3] linear algebra libraries.)

☆ This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Correspondence to: Institute of Mathematics, University of Aberdeen, King's College, Aberdeen, AB24 3UE, United Kingdom.
URLs: http://www.physics.utoronto.ca/~twelsh (T.A. Welsh), http://www.physics.utoronto.ca/~rowe (D.J. Rowe).

http://dx.doi.org/10.1016/j.cpc.2015.10.017

Additional comments:

1. The dimension of the Hilbert space that can be handled is limited only by the available computer memory and the available SO(5) ⊃ SO(3) Clebsch–Gordan coefficients $(v_1\alpha_1 L_1 v_2\alpha_2 L_2 \| v_3\alpha_3 L_3)$.
2. The supplied data files provide coefficients $(v_1\alpha_1 L_1 v_2\alpha_2 L_2 \| v_3\alpha_3 L_3)$ for $1 \leq v_2 \leq 6$, and contain all non-zero coefficients for $v_1 < v_3 \leq 50$ when $v_2 \in 1, 3$, for $v_1 \leq v_3 \leq 30$ when $v_2 \in 2, 4$, and for $v_1 \leq v_3 \leq 25$ when $v_2 \in 5, 6$. (Once calculated, further coefficients can be readily made available to the code without changing the code.) Thus, depending on the model Hamiltonian being analysed, the states in the Hilbert space used are limited in their seniority. For analysis of the more typical types of model Hamiltonian, only the coefficients with $v_2 \in \{1, 3\}$ are required, and therefore, with the supplied files, the seniority limit is 50. More exotic Hamiltonians having terms with seniority $v_2 \in \{2, 4, 5, 6\}$ would have the seniority limited to 30 or 25 accordingly.
3. The code provides lower level procedures that give ready access to the Clebsch–Gordan coefficients and the SU(1, 1) and SO(5) matrix elements. These procedures are described in the manuscript and enable extensions to the code and model to be made easily.
4. The accuracy to which Maple performs numerical calculations is determined by the Maple parameter Digits, which specifies the number of significant decimal digits used. The default value of 10 is more than adequate for most ACM calculations. Note, however, that if Digits is increased beyond a certain value (obtained from the Maple command `evalhf(Digits)`, and usually 15 on modern computers) then the code can no longer take advantage of hardware mathematical operations, and is significantly slower.

Documents included

1. The code makes use of SO(5) ⊃ SO(3) Clebsch–Gordan coefficients which are supplied in zip files, and must be installed by the user.
2. A Maple worksheet that gives various example calculations and tests carried out using procedures from the code is provided.
3. A 162 page PDF file containing everything displayed in the worksheet (input, output and comments, and making use of colour) is also provided.

!!!!! The distribution file for this program is over 46 Mbytes and therefore is not delivered directly when download or Email is requested. Instead a html file giving details of how the program can be obtained is sent. !!!!!

Running time: For a fixed value of the parameter Digits, the running time depends on the dimension of the Hilbert space on which the diagonalisation is performed, and this in turn is governed by the number of eigenvalues required and the accuracy required. Note that diagonalisation is performed separately in each L-space. For typical ACM calculations (such as those carried out in [4]), the matrices being diagonalised are usually of dimension at most a few hundred, and often much smaller. On a modest personal computer, the computation for the smallest cases takes at most a few seconds. The worksheet contains a range of examples for which the calculation time varies between a few seconds and 750s. In the latter case, diagonalisation is performed on L-spaces for $0 \leq L \leq 8$, the dimensions of these spaces being between 154 and 616.

References:

[1] Maplesoft, Waterloo Maple Inc., Waterloo, ON, Canada.
[2] NAG, www.nag.com.
[3] CLAPACK, www.netlib.org/clapack.
[4] D. J. Rowe, T. A. Welsh, M. A. Caprio, Phys. Rev. C 79(2009) 054304.

1. Introduction

The ACM (algebraic collective model) [1–4] is an algebraic version of the Bohr model [5] based on a dynamical group SU(1, 1) × SO(5) for which all the matrix elements needed in applications of the model are calculated analytically. It is a development of the *computationally tractable version of the collective model* [1] that enables collective model calculations to be carried out efficiently by use of wave functions that span modified oscillator series of SU(1, 1) irreps (irreducible representations) [6,7] and complementary SO(5) wave functions. The availability of analytic SU(1, 1) matrix elements and SO(5) Clebsch–Gordan coefficients enable the calculations to bypass expressions for the wave functions entirely, avoiding, in particular, numerical integration. A pedagogical treatment of the geometrical and algebraic foundations of the ACM is given in the recent book by Rowe and Wood [8].

Earlier computer programs that implemented the Bohr model utilised an U(5) ⊃ O(5) ⊃ SO(3) ⊃ SO(2) basis [9–11]. The basis was constructed by starting with a basis of SO(3) coupled polynomials of a given degree in the quadrupole coordinates and diagonalising the O(5) Casimir operator in this basis. The computer program developed by Gneuss and Greiner [12] additionally employed a U(5) ⊃ Sp(4) ⊃ SO(3) × SO(3) ⊃ SO(2) × SO(2) basis, with transformations between the two bases carried out using the methods of [13]. On being further developed, the calculations could be carried out fully in the U(5) ⊃ O(5) ⊃ SO(3) ⊃ SO(2) basis, culminating in the Frankfurt code [14–16]. This was able to analyse model SO(3)-invariant Hamiltonians obtained from three kinetic energy terms and potential energy obtained from various polynomials in β and $\cos 3\gamma$ (the collective model coordinates β and γ are described in Section 2). In the Frankfurt

code, the calculation of matrix elements was partially carried out using numerical integration (in the terminology of Section 2, this numerical calculation was performed on the space $\mathcal{L}^2(S_4, \sin 3\gamma\, d\gamma\, d\Omega)$).

In the Maple [17] code presented here, the use of files of highly accurate precomputed SO(5) ⊃ SO(3) Clebsch–Gordan coefficients together with an exact analytic expression for the SO(5)-reduced matrix elements of SO(5) spherical harmonics enable these computationally intensive methods to be avoided entirely. These files of SO(5) ⊃ SO(3) Clebsch–Gordan coefficients were computed [18,19] using the algorithm developed in [20]. This algorithm, which also calculates SO(5) spherical harmonics, was based on the methods of [1] for calculating model SO(5) wave functions. A Mathematica code for the calculation of SO(5) ⊃ SO(3) Clebsch–Gordan coefficients in exact arithmetic has been published by Caprio et al. [18].

The code presented here also benefits enormously from the use of modified oscillator SU(1, 1) irreps, as used previously by Davidson [21] in molecular physics. As a result, calculations for deformed nuclei converge much more rapidly than in a conventional harmonic oscillator basis, as illustrated in Appendix B.4 and [2], and more general Hamiltonians, such as those involving negative powers of β, can be handled. In addition, the pairing of each SO(5) irrep with a certain modified SU(1, 1) irrep enables all matrix elements of the basic Bohr model observables, and many others, to be computed algebraically.

The code is very versatile and can calculate the spectrum and properties of virtually any Bohr model Hamiltonian one might wish to consider, quickly and easily. In addition to contributing to the stockpile of algebraic models that can be used for exploratory studies of nuclear phenomena, it is intended that this code will also serve as a resource for extensions of the Bohr model. For instance, it is of interest to develop models that include extra degrees of freedom, such as vorticity degrees of freedom, particle-core coupling models, the interacting boson model as recently pursued in [22], and other models in which SO(5) representations, their matrix elements, and SO(5) ⊃ SO(3) Clebsch–Gordan coefficients, can be used with advantage.

This article is organised such that Sections 2–5 describe the theoretical framework of the ACM. The code itself is described in Sections 6–9, which serve as a manual for its use. Appendix A describes how to use the code to perform calculations in the rigid-β limit of the ACM. Appendix B discusses methods for obtaining optimal values of the parameters that specify the bases in which the calculations are performed (see Section 5.3). Appendices C and D provide derivations of certain matrix elements that are not available elsewhere. Some concluding remarks are given in Section 10.

2. The model space

A number of ingredients contribute to the simplicity of the ACM relative to the standard formulation of the Bohr model. The first is the characterisation of the nuclear shape by quadrupole moments instead of surface deformation parameters. This is important because quadrupole moments are well-defined quantum mechanical observables; they also have well-defined microscopic expressions in terms of nucleon coordinates. Thus, the configuration space of the model is expressed as the real five-dimensional space $\mathbb{R}^5$ of nuclear shapes, defined by complex quadrupole moments $\{q_M, M = 0, \pm1, \pm2\}$, for which $q_M^* = (-1)^M q_{-M}$. This space can be assigned a radial $\mathbb{R}_+$ coordinate $\beta \geq 0$, given by $\beta^2 = \sum_{M=-2}^{2} |q_M|^2$, and S_4 spherical coordinates (γ, Ω), where S_4 is the four-dimensional sphere of unit radius. Here $0 \leq \gamma \leq \pi/3$ is an angle coordinate, and Ω labels an SO(3) element, which may be expressed in terms of Euler angles in the standard way (see [23], for example). The quadrupole moments are then expressed as products $q_M = \beta \mathcal{Q}_M$, with $\sum_{M=-2}^{2} |\mathcal{Q}_M|^2 = 1$ and

$$\mathcal{Q}_M(\gamma, \Omega) = \cos\gamma\, \mathscr{D}^2_{0M}(\Omega) + \frac{1}{\sqrt{2}} \sin\gamma \left(\mathscr{D}^2_{2M}(\Omega) + \mathscr{D}^2_{-2M}(\Omega)\right), \tag{1}$$

where $\mathscr{D}^L_{KM}(\Omega)$ is a Wigner $\mathscr{D}$-function [23].

The volume element, d^5x, for $\mathbb{R}^5$ is a product of volume elements for $\mathbb{R}_+$ and S_4 given by

$$d^5x = (\beta^4\, d\beta) \times (\sin 3\gamma\, d\gamma\, d\Omega), \tag{2}$$

where the SO(3) volume element $d\Omega$ is normalised such that $\int_{\mathrm{SO}(3)} d\Omega = 8\pi^2$. Thus, the Hilbert space $\mathbb{H}$ of the ACM is expressed as a tensor product

$$\mathbb{H} = \mathcal{L}^2(\mathbb{R}_+, \beta^4\, d\beta) \otimes \mathcal{L}^2(S_4, \sin 3\gamma\, d\gamma\, d\Omega), \tag{3}$$

where $\mathcal{L}^2(\mathbb{R}_+, \beta^4\, d\beta)$ is the Hilbert space of square-integrable functions on $\mathbb{R}_+$ with respect to the volume element $\beta^4\, d\beta$, and $\mathcal{L}^2(S_4, \sin 3\gamma\, d\gamma\, d\Omega)$ is the Hilbert space of square-integrable functions on S_4 with respect to the volume element $\sin 3\gamma\, d\gamma\, d\Omega$.

The model becomes an algebraic model on introduction of orthonormal bases $\{|(a, \lambda)\, \nu\rangle\}$ and $\{|\nu\alpha LM\rangle\}$ for the factors $\mathcal{L}^2(\mathbb{R}_+, \beta^4\, d\beta)$ and $\mathcal{L}^2(S_4, \sin 3\gamma\, d\gamma\, d\Omega)$ of (3), whose elements are labelled by the quantum numbers of the groups in the respective dynamical subgroup chains

$$\begin{array}{ccc} \mathrm{SU}(1,1) & \supset & \mathrm{U}(1) \\ \lambda & & \nu \end{array}, \tag{4}$$

and

$$\begin{array}{ccccc} \mathrm{SO}(5) & \supset & \mathrm{SO}(3) & \supset & \mathrm{SO}(2) \\ \nu & & \alpha\ \ L & & M \end{array}. \tag{5}$$

The parameter a in the basis $\{|(a, \lambda)\, \nu\rangle\}$ for $\mathcal{L}^2(\mathbb{R}_+, \beta^4\, d\beta)$ is a useful scale parameter that implicitly defines the U(1) ⊂ SU(1, 1) subgroup (see Section 3). The group SO(5) in the chain (5) is the group of linear transformations of the five quadrupole moments $\{q_M\}$ that leave β^2 invariant, and SO(3) is the rotational subgroup that transforms the quadrupole moments as a basis for the 5-dimensional $L = 2$ irrep.

An extra 'missing label' α in the range $1 \leq \alpha \leq d_{\nu L}$ is needed to distinguish the multiplicity, $d_{\nu L}$, of SO(3) irreps of the same angular momentum L in an SO(5) irrep of seniority ν (seniority is the SO(5) analogue of angular momentum). This multiplicity is given [24,18] by

$$d_{\nu L} = (\lfloor \tfrac{1}{3}(\nu - b) \rfloor + 1)\theta_{\nu-b} - \lfloor \tfrac{1}{3}(\nu - L + 2) \rfloor \theta_{\nu-L+2}, \tag{6}$$

where $b = L/2$ for L even and $b = (L+3)/2$ for L odd, $\theta_k = 1$ for $k \geq 0$ and $\theta_k = 0$ for $k < 0$, and $\lfloor x \rfloor$ is the largest integer not greater than x.

As we show in the following, the matrix elements of all operators of interest are given, in the factored basis $\{|(a,\lambda)\ \nu;\ \upsilon\alpha LM\rangle = |(a,\lambda)\ \nu\rangle \otimes |\upsilon\alpha LM\rangle\}$ of $\mathbb{H}$, in terms of easily calculated products of radial and SO(5) matrix elements. However, it must be understood that whereas the set of states $\{|\upsilon\alpha LM\rangle\}$, with seniority υ taking all integer values $\upsilon \geq 0$, is an orthonormal basis for $\mathcal{L}^2(S_4, \sin 3\gamma\ d\gamma\ d\Omega)$, the set $\{|(a,\lambda)\ \nu\rangle\}$ is an orthonormal basis for $\mathcal{L}^2(\mathbb{R}_+, \beta^4\ d\beta)$ for each pair (a,λ). Consequently, when $a' \neq a$ or $\lambda' \neq \lambda$, the overlaps $\langle(a',\lambda')\ \mu|(a,\lambda)\ \nu\rangle$ are not necessarily zero for $\mu \neq \nu$. Thus, to obtain an orthonormal basis for the tensor product space $\mathbb{H}$, and to facilitate exploitation of the SO(5) structure, one must choose a set of states $\{|(a_\upsilon,\lambda_\upsilon)\ \nu;\ \upsilon\alpha LM\rangle\}$ such that a and λ take fixed values a_υ and λ_υ for all states of a given seniority υ. Such a basis then has the overlaps

$$\begin{aligned}\langle(a_{\upsilon'},\lambda_{\upsilon'})\ \mu;\ \upsilon'\alpha'L'M'|(a_\upsilon,\lambda_\upsilon)\ \nu;\ \upsilon\alpha LM\rangle &= \langle(a_\upsilon,\lambda_\upsilon)\ \mu|(a_\upsilon,\lambda_\upsilon)\ \nu\rangle\,\delta_{\upsilon',\upsilon}\delta_{\alpha',\alpha}\delta_{L',L}\delta_{M',M}\\ &= \delta_{\mu,\upsilon}\delta_{\upsilon',\upsilon}\delta_{\alpha',\alpha}\delta_{L',L}\delta_{M',M},\end{aligned} \tag{7}$$

and is therefore indeed an orthonormal basis. In fact, it will be convenient to fix $a_\upsilon = a$ to have a single υ-independent value in any given calculation. The ACM then uses bases

$$\{|(a,\lambda_\upsilon)\ \nu;\ \upsilon\alpha LM\rangle = |(a,\lambda_\upsilon)\ \nu\rangle \otimes |\upsilon\alpha LM\rangle,\ \nu \geq 0,\ \upsilon \geq 0,\ L \geq 0,\ 1 \leq \alpha \leq d_{\upsilon L},\ -L \leq M \leq L\} \tag{8}$$

of orthonormal states for $\mathbb{H}$.

3. SU(1, 1) matrix elements

The basis states $\{|(a,\lambda)\ \nu\rangle,\ \nu = 0, 1, 2, \ldots\}$ for the Hilbert space $\mathcal{L}^2(\mathbb{R}_+, \beta^4\ d\beta)$ correspond to wave functions given by

$$\left\{P^{(a)}_{\nu,\lambda}(\beta) = \frac{1}{\beta^2}\mathcal{R}^{(a,\lambda)}_\nu(\beta),\ \nu = 0, 1, 2, 3, \ldots\right\}, \tag{9}$$

where

$$\mathcal{R}^{(a,\lambda)}_\nu(\beta) = (-1)^\nu\sqrt{\frac{2\nu!\,a}{\Gamma(\lambda+\nu)}}(a\beta)^{\lambda-1/2}e^{-a^2\beta^2/2}L^{(\lambda-1)}_\nu(a^2\beta^2), \tag{10}$$

and $L^{(\lambda-1)}_\nu$ is a generalised Laguerre polynomial [25]. When $\lambda - 5/2$ is a non-negative integer and a is the inverse of a harmonic oscillator unit of length, the wave functions $P^{(a)}_{\nu,\lambda}(\beta)$ are standard radial wave functions for an isotropic five-dimensional harmonic oscillator.[1] However, for arbitrary $\lambda > 0$, they define ordered bases of modified oscillator radial wave functions in terms of which, for optimal choices of a and λ, the expansion of collective model wave functions for deformed nuclei converge more rapidly, often considerably (see Appendix B.4).

The normalisation of the functions $\mathcal{R}^{(a,\lambda)}_\nu(\beta)$ in (10) is such that

$$\langle(a,\lambda)\ \mu|(a,\lambda)\ \nu\rangle = \int_0^\infty P^{(a)}_{\mu,\lambda}(\beta)\,P^{(a)}_{\nu,\lambda}(\beta)\,\beta^4\,d\beta = \int_0^\infty \mathcal{R}^{(a,\lambda)}_\mu(\beta)\,\mathcal{R}^{(a,\lambda)}_\nu(\beta)\,d\beta = \delta_{\mu,\nu}. \tag{11}$$

For each a, $\lambda > 0$, the wave functions (9) then form an orthonormal basis with respect to the volume element $\beta^4\,d\beta$ for $\mathcal{L}^2(\mathbb{R}_+, \beta^4\,d\beta)$, and $\{|(a,\lambda)\ \nu\rangle,\ \nu = 0, 1, 2, 3, \ldots\}$ are orthonormal basis states. Correspondingly, for each a, $\lambda > 0$, the functions (10) form an orthonormal basis with respect to the volume element $d\beta$ for the Hilbert space $\mathcal{L}^2(\mathbb{R}_+, d\beta)$.

Matrix elements of an operator $\hat{X}$ acting on $\mathcal{L}^2(\mathbb{R}_+, \beta^4\,d\beta)$ are given by

$$\langle(a,\lambda')\ \mu|\hat{X}\,|(a,\lambda)\ \nu\rangle = \int_0^\infty P^{(a)}_{\mu,\lambda'}(\beta)\,[\hat{X}\,P^{(a)}_{\nu,\lambda}(\beta)]\,\beta^4\,d\beta = F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\beta^2\hat{X}\frac{1}{\beta^2}\right), \tag{12}$$

where

$$F^{(a)}_{\lambda'\mu;\lambda\nu}(\hat{Z}) = \int_0^\infty \mathcal{R}^{(a,\lambda')}_\mu(\beta)\,[\hat{Z}\,\mathcal{R}^{(a,\lambda)}_\nu(\beta)]\,d\beta. \tag{13}$$

Note that the map $P^{(a)}_{\nu,\lambda}(\beta) \mapsto \mathcal{R}^{(a,\lambda)}_\nu(\beta) = \beta^2 P^{(a)}_{\nu,\lambda}(\beta)$ between basis wave functions generates an isomorphic mapping between the Hilbert spaces $\mathcal{L}^2(\mathbb{R}_+, \beta^4\,d\beta) \to \mathcal{L}^2(\mathbb{R}_+, d\beta)$. This mapping induces the mapping $\hat{X} \mapsto \hat{Z} = \hat{\beta}^2\hat{X}\hat{\beta}^{-2}$ between operators on these spaces.

The space $\mathcal{L}^2(\mathbb{R}_+, d\beta)$ carries representations of the Lie algebra of SU(1, 1), whose complexification has basis elements S_0, S_+ and S_- that satisfy the commutation relations

$$[S_+, S_-] = -2S_0, \qquad [S_0, S_\pm] = \pm S_\pm, \tag{14}$$

and which, in a unitary representation, are realised as operators $\hat{S}_0$, $\hat{S}_+$ and $\hat{S}_-$ respectively, that satisfy the Hermiticity relations

$$\hat{S}_0^\dagger = \hat{S}_0, \qquad \hat{S}_\pm^\dagger = \hat{S}_\mp. \tag{15}$$

[1] More generally, for $n \geq 1$ and $\lambda - n/2$ a non-negative integer, the functions $\beta^{(1-n)/2}\mathcal{R}^{(a,\lambda)}_\nu(\beta)$ are radial wave functions for an isotropic n-dimensional harmonic oscillator.

In fact, for each fixed pair $a, \lambda > 0$, there is a realisation of this Lie algebra in which S_0 and $S_\pm$ map respectively to operators $\hat{S}_0^{(a,\lambda)}$ and $\hat{S}_\pm^{(a,\lambda)}$ on $\mathcal{L}^2(\mathbb{R}_+, d\beta)$, whose matrix elements are

$$F^{(a)}_{\lambda\mu;\lambda\nu}(\hat{S}_0^{(a,\lambda)}) = \frac{1}{2}(\lambda + 2\nu)\,\delta_{\mu,\nu}, \tag{16}$$

$$F^{(a)}_{\lambda\mu;\lambda\nu}(\hat{S}_+^{(a,\lambda)}) = \sqrt{(\lambda+\nu)(\nu+1)}\;\delta_{\mu,\nu+1}, \tag{17}$$

$$F^{(a)}_{\lambda\mu;\lambda\nu}(\hat{S}_-^{(a,\lambda)}) = \sqrt{(\lambda+\nu-1)\nu}\;\delta_{\mu,\nu-1}, \tag{18}$$

relative to the basis $\{\mathcal{R}_\mu^{(a,\lambda)}\}$. Explicitly, these realisations are given by [2][2]

$$\hat{S}_0^{(a,\lambda)} = \frac{1}{4}\left[-\frac{1}{a^2}\frac{d^2}{d\beta^2} + \frac{(\lambda - 3/2)(\lambda - 1/2)}{(a\beta)^2} + (a\beta)^2\right], \tag{19}$$

$$\hat{S}_\pm^{(a,\lambda)} = \frac{1}{4}\left[\frac{1}{a^2}\frac{d^2}{d\beta^2} - \frac{(\lambda - 3/2)(\lambda - 1/2)}{(a\beta)^2} + (a\beta)^2 \mp \left(2\beta\frac{d}{d\beta} + 1\right)\right]. \tag{20}$$

Using these, the following matrix elements can be derived (see [8, Section 4.2.2]):

$$a^2 F^{(a)}_{\lambda\mu;\lambda\nu}\left(\beta^2\right) = \delta_{\mu,\nu+1}\sqrt{(\lambda+\nu)(\nu+1)} + \delta_{\mu,\nu-1}\sqrt{(\lambda+\nu-1)\nu} + \delta_{\mu,\nu}(\lambda+2\nu), \tag{21}$$

$$\frac{1}{a^2}F^{(a)}_{\lambda\mu;\lambda\nu}\left(\frac{1}{\beta^2}\right) = \frac{1}{a^2}F^{(a)}_{\lambda\nu;\lambda\mu}\left(\frac{1}{\beta^2}\right) = \frac{(-1)^{\mu-\nu}}{\lambda-1}\sqrt{\frac{\mu!\,\Gamma(\lambda+\nu)}{\nu!\,\Gamma(\lambda+\mu)}} \qquad \text{for } \mu \geq \nu,\ \lambda > 1, \tag{22}$$

$$\begin{aligned}\frac{1}{a^2}F^{(a)}_{\lambda\mu;\lambda\nu}\left(\frac{d^2}{d\beta^2}\right) &= \delta_{\mu,\nu+1}\sqrt{(\lambda+\nu)(\nu+1)} + \delta_{\mu,\nu-1}\sqrt{(\lambda+\nu-1)\nu} \\ &\quad - \delta_{\mu,\nu}(\lambda+2\nu) + \frac{1}{a^2}\left(\lambda-\frac{3}{2}\right)\left(\lambda-\frac{1}{2}\right)F^{(a)}_{\lambda\mu;\lambda\nu}\left(\frac{1}{\beta^2}\right),\end{aligned} \tag{23}$$

$$F^{(a)}_{\lambda\mu;\lambda\nu}\left(\beta\frac{d}{d\beta}\right) = -\delta_{\mu,\nu+1}\sqrt{(\lambda+\nu)(\nu+1)} + \delta_{\mu,\nu-1}\sqrt{(\lambda+\nu-1)\nu} - \frac{1}{2}\delta_{\mu,\nu}. \tag{24}$$

The dependence on a, given by the factor premultiplying each $F^{(a)}_{\lambda\mu;\lambda\nu}$, exhibits the fact that a is a scale factor in β.

The following matrix elements were derived [3] by expressing $\hat{S}_0^{(a,\lambda)}$ and $\hat{S}_\pm^{(a,\lambda)}$ in terms of products of the operators

$$A(\kappa) := \frac{1}{a}\frac{d}{d\beta} + \frac{\kappa}{a\beta} + a\beta, \qquad A^\dagger(\kappa) := -\frac{1}{a}\frac{d}{d\beta} + \frac{\kappa}{a\beta} + a\beta, \tag{25}$$

with $\kappa = \pm(\lambda - \frac{1}{2})$ and $\kappa = \pm(\lambda - \frac{3}{2})$:

$$aF^{(a)}_{\lambda+1,\mu;\lambda\nu}(\beta) = \delta_{\mu,\nu}\sqrt{\lambda+\nu} + \delta_{\mu,\nu-1}\sqrt{\nu}, \tag{26}$$

$$aF^{(a)}_{\lambda-1,\mu;\lambda\nu}(\beta) = \delta_{\mu,\nu}\sqrt{\lambda+\nu-1} + \delta_{\mu,\nu+1}\sqrt{\nu+1}, \tag{27}$$

$$\frac{1}{a}F^{(a)}_{\lambda+1,\mu;\lambda\nu}\left(\frac{1}{\beta}\right) = \begin{cases} 0 & \text{if } \mu < \nu, \\ (-1)^{\mu-\nu}\sqrt{\dfrac{\mu!\,\Gamma(\lambda+\nu)}{\nu!\,\Gamma(\lambda+\mu+1)}} & \text{if } \mu \geq \nu, \end{cases} \tag{28}$$

$$\frac{1}{a}F^{(a)}_{\lambda-1,\mu;\lambda\nu}\left(\frac{1}{\beta}\right) = \begin{cases} 0 & \text{if } \mu > \nu, \\ (-1)^{\mu-\nu}\sqrt{\dfrac{\nu!\,\Gamma(\lambda+\mu-1)}{\mu!\,\Gamma(\lambda+\nu)}} & \text{if } \mu \leq \nu, \end{cases} \tag{29}$$

$$\frac{1}{a}F^{(a)}_{\lambda+1,\mu;\lambda\nu}\left(\frac{d}{d\beta}\right) = -\delta_{\mu,\nu}\sqrt{\lambda+\nu} + \delta_{\mu,\nu-1}\sqrt{\nu} + \frac{1}{a}\left(\lambda-\frac{1}{2}\right)F^{(a)}_{\lambda+1,\mu;\lambda\nu}\left(\frac{1}{\beta}\right), \tag{30}$$

$$\frac{1}{a}F^{(a)}_{\lambda-1,\mu;\lambda\nu}\left(\frac{d}{d\beta}\right) = -\delta_{\mu,\nu+1}\sqrt{\nu+1} + \delta_{\mu,\nu}\sqrt{\lambda+\nu-1} - \frac{1}{a}\left(\lambda-\frac{3}{2}\right)F^{(a)}_{\lambda-1,\mu;\lambda\nu}\left(\frac{1}{\beta}\right). \tag{31}$$

Note that, with respect to the inner product (13) for the functions $\mathcal{R}_\nu^{(a,\lambda)}(\beta)$, the operators $\hat{\beta}^{\pm1}$, $\hat{\beta}^{\pm2}$ and $d^2/d\beta^2$ are Hermitian, whereas $d/d\beta$ is skew Hermitian. Thus, we have the identities

$$F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\beta^{\pm1}\right) = F^{(a)}_{\lambda\nu;\lambda'\mu}\left(\beta^{\pm1}\right), \qquad F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\beta^{\pm2}\right) = F^{(a)}_{\lambda\nu;\lambda'\mu}\left(\beta^{\pm2}\right), \tag{32a}$$

$$F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\frac{d}{d\beta}\right) = -F^{(a)}_{\lambda\nu;\lambda'\mu}\left(\frac{d}{d\beta}\right), \qquad F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\frac{d^2}{d\beta^2}\right) = F^{(a)}_{\lambda\nu;\lambda'\mu}\left(\frac{d^2}{d\beta^2}\right). \tag{32b}$$

[2] For operators that are clearly multiplicative, we omit their 'hat's when it is typographically convenient.

For maximum flexibility in the choice of λ and λ', the program also makes use of the matrix elements of the identity operator $\hat{1}$ between states $|(a,\lambda)\,\nu\rangle$ and $|(a,\lambda')\,\mu\rangle$ for which $\lambda'-\lambda$ is an even integer. For $r>0$, the calculation given in Appendix C yields

$$F^{(a)}_{\lambda+2r,\mu;\lambda,\nu}(\hat{1}) = (-1)^{\nu-\mu}\sqrt{\frac{\mu!\,\Gamma(\lambda+\nu)}{\nu!\,\Gamma(\lambda+\mu+2r)}}\,c^{(2r)}_{\mu,\nu}, \tag{33}$$

where

$$c^{(2r)}_{\mu,\nu} = \sum_{j=\max\{0,\nu-\mu\}}^{r}(-1)^j\binom{r}{j}\frac{\Gamma(\lambda+\mu+2r)\,\Gamma(\mu+j+1)}{\Gamma(\lambda+\mu+r+j)\,\Gamma(\mu+1)}\binom{\mu-\nu+j+r-1}{\mu-\nu+j}. \tag{34}$$

Note that $c^{(2r)}_{\mu,\nu}$ is a polynomial in λ with integer coefficients and degree at most $\min\{r, r-\nu+\mu\}$. In particular, $c^{(2r)}_{\mu,\nu}=0$ if $\nu>\mu+r$.

For an arbitrary operator $\hat{Z}$ acting on the Hilbert space $\mathcal{L}^2(\mathbb{R}_+,\beta^4\,d\beta)$ and a positive integer r, the matrix elements (33) enable matrix elements $F^{(a)}_{\lambda'\pm2r,\mu;\lambda,\nu}(\hat{Z})$ to be obtained from those of $F^{(a)}_{\lambda',\mu;\lambda,\nu}(\hat{Z})$ using

$$F^{(a)}_{\lambda'+2r,\mu;\lambda,\nu}(\hat{Z}) = \sum_{\xi\geq0}F^{(a)}_{\lambda'+2r,\mu;\lambda',\xi}(\hat{1})\,F^{(a)}_{\lambda',\xi;\lambda,\nu}(\hat{Z}); \tag{35a}$$

$$F^{(a)}_{\lambda'-2r,\mu;\lambda,\nu}(\hat{Z}) = \sum_{\xi\geq0}F^{(a)}_{\lambda'-2r,\mu;\lambda-2r,\xi}(\hat{Z})\,F^{(a)}_{\lambda-2r,\xi;\lambda,\nu}(\hat{1}) = \sum_{\xi\geq0}F^{(a)}_{\lambda'-2r,\mu;\lambda-2r,\xi}(\hat{Z})\,F^{(a)}_{\lambda,\nu;\lambda-2r,\xi}(\hat{1}), \tag{35b}$$

where the final identity follows because $\hat{1}$ is an Hermitian operator. Then, because $c^{(2r)}_{\mu,\xi}=0$ for $\xi>\mu+r$, it follows that a given matrix element $F^{(a)}_{\lambda'\pm2r,\mu;\lambda,\nu}(\hat{Z})$ is obtained precisely by restricting the sum in (35) to $0\leq\xi\leq\mu+r$ or $0\leq\xi\leq\nu+r$ as appropriate. Therefore, when the sum is carried out through the multiplication of matrices, which are necessarily finite-dimensional in a computer implementation, the matrix element $F^{(a)}_{\lambda'\pm2r,\mu;\lambda,\nu}(\hat{Z})$ is computed precisely for matrix dimensions exceeding $\max\{\mu,\nu\}+r$.

4. SO(5) ⊃ SO(3) matrix elements and Clebsch–Gordan coefficients

Consider an SO(5) tensor operator $\hat{T}^v$. This operator has components $\hat{T}^v_{\alpha LM}$ that transform under SO(5) in the same way as the basis states $|v\alpha LM\rangle$. The Wigner–Eckart theorem for SO(3) [23] implies that the matrix elements of these operators can be expressed

$$\langle v_f\alpha_f L_f M_f|\hat{T}^v_{\alpha LM}|v_i\alpha_i L_i M_i\rangle = (L_i M_i\; LM|L_f M_f)\,\frac{\langle v_f\alpha_f L_f\|\hat{T}^v_{\alpha L}\|v_i\alpha_i L_i\rangle}{\sqrt{2L_f+1}}, \tag{36}$$

where $(L_iM_i\;LM|L_fM_f)$ is an SO(3) Clebsch–Gordan coefficient, and $\langle v_f\alpha_fL_f\|\hat{T}^v_{\alpha L}\|v_i\alpha_iL_i\rangle$ is an SO(3)-reduced matrix element. Note that the factor $\sqrt{2L_f+1}$ in (36) is included for symmetry reasons, and some authors use different phase conventions, neither of which are essential to the Wigner–Eckart theorem. In fact, in what follows, we often find it convenient to use the following alternative definition of SO(3)-reduced matrix elements:

$$\langle v_f\alpha_fL_f\|\hat{T}^v_{\alpha L}\|v_i\alpha_iL_i\rangle^\natural = \frac{\langle v_f\alpha_fL_f\|\hat{T}^v_{\alpha L}\|v_i\alpha_iL_i\rangle}{\sqrt{2L_f+1}}. \tag{37}$$

A similar use of the Wigner–Eckart theorem for SO(5) implies that

$$\langle v_f\alpha_fL_fM_f|\hat{T}^v_{\alpha LM}|v_i\alpha_iL_iM_i\rangle = (v_i\alpha_iL_iM_i,\;v\alpha LM|v_f\alpha_fL_fM_f)\,\langle v_f\|\hat{T}^v\|v_i\rangle, \tag{38}$$

where $(v_i\alpha_iL_iM_i,\;v\alpha LM|v_f\alpha_fL_fM_f)$ is an SO(5) Clebsch–Gordan coefficient, and $\langle v_f\|\hat{T}^v\|v_i\rangle$ is an SO(5)-reduced matrix element.

Racah's factorisation lemma [26] combines (36) and (38) to give the identity

$$\langle v_f\alpha_fL_f\|\hat{T}^v_{\alpha L}\|v_i\alpha_iL_i\rangle = \sqrt{2L_f+1}\,(v_i\alpha_iL_i,\;v\alpha L\;\|\;v_f\alpha_fL_f)\,\langle v_f\|\hat{T}^v\|v_i\rangle, \tag{39}$$

where

$$(v_i\alpha_iL_i,\;v\alpha L\;\|\;v_f\alpha_fL_f) = \frac{(v_i\alpha_iL_iM_i,\;v\alpha LM|v_f\alpha_fL_fM_f)}{(L_iM_i\;LM|L_fM_f)}, \tag{40}$$

which is independent of M_i, M and M_f. The quantity $(v_i\alpha_iL_i,\;v\alpha L\;\|\;v_f\alpha_fL_f)$ has been called an SO(5) ⊃ SO(3) Clebsch–Gordan coefficient [18] (it is also known as an SO(3)-reduced SO(5) Clebsch–Gordan coefficient and as an *isoscalar factor*).

The SO(5) ⊃ SO(3) Clebsch–Gordan coefficients have been calculated [19,18] using the algorithm of [20]. In this algorithm, SO(5) spherical harmonics $\{\mathcal{Y}^v_{\alpha LM}(\gamma,\Omega)\}$ are calculated first. These form an orthonormal basis of $\mathcal{L}^2(S_4,\sin3\gamma\;d\gamma\;d\Omega)$, and are wave functions for the basis states $\{|v\alpha LM\rangle\}$. SO(5) spherical harmonics of particular importance are

$$\mathcal{Y}^1_{12M} = \frac{\sqrt{15}}{4\pi}\,\mathcal{Q}_M, \tag{41}$$

and

$$\mathcal{Y}^{3n}_{100}(\gamma,\Omega) = \frac{1}{4\pi}\sqrt{3(2n+1)}\,P_n(\cos3\gamma), \tag{42}$$

for $n \geq 0$, where P_n is a Legendre polynomial [25]. For example,

$$\mathcal{Y}^0_{100}(\gamma, \Omega) = \frac{\sqrt{3}}{4\pi}, \qquad \mathcal{Y}^3_{100}(\gamma, \Omega) = \frac{3}{4\pi}\cos 3\gamma, \qquad \mathcal{Y}^6_{100}(\gamma, \Omega) = \frac{\sqrt{15}}{8\pi}(3\cos^2 3\gamma - 1). \tag{43}$$

To each spherical harmonic there is then a component of a tensor operator $\hat{\mathcal{Y}}^v_{\alpha LM}$ such that the state $\hat{\mathcal{Y}}^v_{\alpha LM}|v_i\alpha_i L_i M_i\rangle$ has wave function $\mathcal{Y}^v_{\alpha LM}(\gamma, \Omega)\mathcal{Y}^{v_i}_{\alpha_i L_i M_i}(\gamma, \Omega)$. It follows from (36) and (39) that evaluation of integrals of the form

$$\langle v_f\alpha_f L_f M_f|\hat{\mathcal{Y}}^v_{\alpha LM}|v_i\alpha_i L_i M_i\rangle = \int_{S_4} \mathcal{Y}^{v_f}_{\alpha_f L_f M_f}(\gamma, \Omega)^*\, \mathcal{Y}^v_{\alpha LM}(\gamma, \Omega)\, \mathcal{Y}^{v_i}_{\alpha_i L_i M_i}(\gamma, \Omega)\, \sin 3\gamma\, d\gamma\, d\Omega \tag{44}$$

enables both the SO(5) $\supset$ SO(3) Clebsch–Gordan coefficients and the SO(5)-reduced matrix elements $\langle v_f\|\hat{\mathcal{Y}}^v\|v_i\rangle$ to be determined. In the process of such calculations, the empirical expression

$$\langle v_f\|\hat{\mathcal{Y}}^v\|v_i\rangle = \frac{1}{4\pi}\frac{(\frac{\sigma}{2}+1)!}{(\frac{\sigma}{2}-v_i)!(\frac{\sigma}{2}-v)!(\frac{\sigma}{2}-v_f)!}\sqrt{\frac{(2v_i+3)(2v+3)}{(v_f+2)(v_f+1)}} \times\sqrt{\frac{(\sigma+4)(\sigma-2v_i+1)!(\sigma-2v+1)!(\sigma-2v_f+1)!}{(\sigma+3)!}}, \tag{45}$$

with $\sigma = v_i + v + v_f$, was obtained by recognising the pattern of results obtained numerically. This expression is consistent with such reduced matrix elements as have been obtained by algebraic methods [1–3] ; for example,

$$\langle v_f\|\hat{\mathcal{Q}}\|v_i\rangle = \delta_{v_f,v_i+1}\sqrt{\frac{v_i+1}{2v_i+5}} + \delta_{v_f,v_i-1}\sqrt{\frac{v_i+2}{2v_i+1}}. \tag{46}$$

It will be shown in the following that the SO(5) tensors that arise in the ACM have matrix elements related to those of corresponding spherical harmonics. Thus, these results prove to be sufficient for present needs. Consequently, the use of (45) and the available tabulations of SO(5) $\supset$ SO(3) Clebsch–Gordan coefficients obviate the need to evaluate integrals such as (44).

5. Collective model observables and their matrix elements

5.1. Operators acting on the tensor product space $\mathbb{H} = \mathcal{L}^2(\mathbb{R}_+, \beta^4\, d\beta) \otimes \mathcal{L}^2(S_4, \sin 3\gamma\, d\gamma\, d\Omega)$

In the bases defined by (8), the matrix elements of any operator $\hat{W} = \hat{X}\hat{T}$ that is a product of operators, $\hat{X}$ and $\hat{T}$, that act independently on the component spaces $\mathcal{L}^2(\mathbb{R}_+, \beta^4\, d\beta)$ and $\mathcal{L}^2(S_4, \sin 3\gamma\, d\gamma\, d\Omega)$ of $\mathbb{H}$, respectively, are given by the products

$$\begin{aligned}\langle(a, \lambda_{v_f})\,\mu;\, v_f\alpha_f L_f M_f|\hat{W}|(a, \lambda_{v_i})\,\nu;\, v_i\alpha_i L_i M_i\rangle &= \langle(a, \lambda_{v_f})\,\mu|\hat{X}|(a, \lambda_{v_i})\,\nu\rangle\,\langle v_f\alpha_f L_f M_f|\hat{T}|v_i\alpha_i L_i M_i\rangle \\ &= F^{(a)}_{\lambda_{v_f}\mu;\lambda_{v_i}\nu}\left(\beta^2\hat{X}\frac{1}{\beta^2}\right)\langle v_f\alpha_f L_f M_f|\hat{T}|v_i\alpha_i L_i M_i\rangle,\end{aligned} \tag{47}$$

after using (12). For observables acting on $\mathbb{H}$ that are products of SU(1, 1) $\times$ SO(5) tensors of the form

$$\hat{W}^v_{\alpha LM} = \hat{X}\hat{\mathcal{Y}}^v_{\alpha LM}, \tag{48}$$

the Wigner–Eckart theorem for each of SO(5) and SO(3) implies that

$$\langle(a, \lambda_{v_f})\,\mu;\, v_f\alpha_f L_f M_f|\hat{W}|(a, \lambda_{v_i})\,\nu;\, v_i\alpha_i L_i M_i\rangle = (v_i\alpha_i L_i M_i,\, v\alpha LM|v_f\alpha_f L_f M_f)\,\langle(a, \lambda_{v_f})\,\mu;\, v_f\|\hat{W}^v\|(a, \lambda_{v_i})\,\nu;\, v_i\rangle \tag{49a}$$

$$= (L_i M_i\, LM|L_f M_f)\,\langle(a, \lambda_{v_f})\,\mu;\, v_f\alpha_f L_f\|\hat{W}^v_{\alpha L}\|(a, \lambda_{v_i})\,\nu;\, v_i\alpha_i L_i\rangle^\natural, \tag{49b}$$

where, using (47) and (38), the SO(5)-reduced matrix elements of $\hat{W}^v_{\alpha LM}$ are given by

$$\langle(a, \lambda_{v_f})\,\mu;\, v_f\|\hat{W}^v\|(a, \lambda_{v_i})\,\nu;\, v_i\rangle = F^{(a)}_{\lambda_{v_f}\mu;\lambda_{v_i}\nu}\left(\beta^2\hat{X}\frac{1}{\beta^2}\right)\langle v_f\|\hat{\mathcal{Y}}^v\|v_i\rangle, \tag{50}$$

and, using first (47), (36) and (37), and then (37), (39) and (50), the adjusted SO(3)-reduced matrix elements of $\hat{W}^v_{\alpha LM}$ are given by

$$\begin{aligned}\langle(a, \lambda_{v_f})\,\mu;\, v_f\alpha_f L_f\|\hat{W}^v_{\alpha L}\|(a, \lambda_{v_i})\,\nu;\, v_i\alpha_i L_i\rangle^\natural &= F^{(a)}_{\lambda_{v_f}\mu;\lambda_{v_i}\nu}\left(\beta^2\hat{X}\frac{1}{\beta^2}\right)\langle v_f\alpha_f L_f\|\hat{\mathcal{Y}}^v_{\alpha L}\|v_i\alpha_i L_i\rangle^\natural \\ &= (v_i\alpha_i L_i,\ v\alpha L\ \|\ v_f\alpha_f L_f)\,\langle(a, \lambda_{v_f})\,\mu;\, v_f\|\hat{W}^v\|(a, \lambda_{v_i})\,\nu;\, v_i\rangle.\end{aligned} \tag{51}$$

Because $q = \beta\mathcal{Q}$, with each $\mathcal{Q}_M$ proportional to the SO(5) spherical harmonic $\mathcal{Y}^1_{12M}$, as in (41), it follows from (46) and (50) that the non-zero SO(5)-reduced matrix elements of the basic collective model observables $\hat{q}_M$ are given by

$$\langle(a, \lambda_{v+1})\,\mu;\, v+1\|\hat{q}\|(a, \lambda_v)\,\nu;\, v\rangle = F^{(a)}_{\lambda_{v+1}\mu;\lambda_v\nu}(\beta)\sqrt{\frac{v+1}{2v+5}}, \tag{52a}$$

$$\langle(a, \lambda_{v-1})\,\mu;\, v-1\|\hat{q}\|(a, \lambda_v)\,\nu;\, v\rangle = F^{(a)}_{\lambda_{v-1}\mu;\lambda_v\nu}(\beta)\sqrt{\frac{v+2}{2v+1}}. \tag{52b}$$

It has also been shown [8] that the non-zero SO(5)-reduced matrix elements of the observables $\hat{\pi}_N$, the momenta conjugate to $\hat{q}_M$, are given by

$$\langle (a, \lambda_{v+1})\, \mu;\, v+1 \| \hat{\pi} \| (a, \lambda_v)\, \nu;\, v \rangle = -\mathrm{i}\hbar F^{(a)}_{\lambda_{v+1}\mu;\lambda_v\nu} \left(\frac{d}{d\beta} - \frac{v+2}{\beta} \right) \sqrt{\frac{v+1}{2v+5}}, \tag{53a}$$

$$\langle (a, \lambda_{v-1})\, \mu;\, v-1 \| \hat{\pi} \| (a, \lambda_v)\, \nu;\, v \rangle = -\mathrm{i}\hbar F^{(a)}_{\lambda_{v-1}\mu;\lambda_v\nu} \left(\frac{d}{d\beta} + \frac{v+1}{\beta} \right) \sqrt{\frac{v+2}{2v+1}}. \tag{53b}$$

Thus, if we choose the v-dependence of λ_v such that

$$\lambda_{v+1} = \lambda_v \pm 1 \tag{54}$$

for all v, explicit algebraic expressions for these collective model matrix elements (52) and (53) are immediately obtained using the expressions for $F^{(a)}_{\lambda\pm1,\mu;\lambda\nu}(\beta)$, $F^{(a)}_{\lambda\pm1,\mu;\lambda\nu}(1/\beta)$ and $F^{(a)}_{\lambda\pm1,\mu;\lambda\nu}(d/d\beta)$ given by (26)–(31). More generally, if the v-dependence of λ_v is such that

$$\lambda_{v+1} - \lambda_v \quad \text{is odd} \tag{55}$$

for all v, then algebraic expressions for (52) and (53) are obtained by using (33) in addition to (26)–(31). One immediate possibility for (55) (and (54)) is to choose

$$\lambda_v = v + \frac{5}{2}, \tag{56}$$

which corresponds to the five-dimensional harmonic oscillator states. Another convenient choice is discussed in Section 5.3.

In $\mathbb{R}^5$, the Laplacian ∇^2 can be expressed (see [8, Section 2.2])

$$\nabla^2 = \frac{1}{\beta^4} \frac{\partial}{\partial\beta} \beta^4 \frac{\partial}{\partial\beta} - \frac{1}{\beta^2} \hat{\Lambda}^2, \tag{57}$$

where $\hat{\Lambda}^2$ is the SO(5) Casimir operator whose eigenfunctions are the SO(5) spherical harmonics $\mathcal{Y}^v_{\alpha LM}$, with

$$\hat{\Lambda}^2 \mathcal{Y}^v_{\alpha LM} = v(v+3) \mathcal{Y}^v_{\alpha LM}. \tag{58}$$

By applying (50) to the two terms of (57), we obtain the SO(5)-reduced matrix elements

$$\langle (a, \lambda_{v_\mathrm{f}})\, \mu;\, v_\mathrm{f} \| \nabla^2 \| (a, \lambda_{v_\mathrm{i}})\, \nu;\, v_\mathrm{i} \rangle = F^{(a)}_{\lambda_{v_\mathrm{i}}\mu;\lambda_{v_\mathrm{i}}\nu} \left(\frac{d^2}{d\beta^2} - \frac{v_\mathrm{i}(v_\mathrm{i}+3)+2}{\beta^2} \right) \delta_{v_\mathrm{f},v_\mathrm{i}}. \tag{59}$$

Explicit analytic expressions for these matrix elements are immediately obtained from those of $F^{(a)}_{\lambda\mu;\lambda\nu}(d^2/d\beta^2)$ and $F^{(a)}_{\lambda\mu;\lambda\nu}(1/\beta^2)$ given in Section 3.

5.2. Collective model observables in the ACM

In our implementation of the ACM, observables of interest are formed by taking sums of products of the following generating operators:

$$\hat{\beta}^2; \quad \hat{\beta}^{-2}; \quad \frac{d^2}{d\beta^2}; \quad \beta\frac{d}{d\beta}; \quad \nabla^2; \quad [\hat{\pi} \otimes \hat{q} \otimes \hat{\pi}]_0; \quad \hat{\mathcal{Y}}^6_{100}; \quad \hat{1}; \tag{60a}$$

$$\hat{\beta}; \quad \hat{\beta}^{-1}; \quad \frac{d}{d\beta}; \quad \cos 3\gamma; \tag{60b}$$

$$\hat{\pi}_M; \quad [\hat{\pi} \otimes \hat{\pi}]_{2M}; \quad [\hat{\pi} \otimes \hat{\pi}]_{4M}; \quad \hat{\mathcal{Y}}^v_{\alpha LM} \quad \text{for } v \in \{2, 4, 6\} \text{ and } L > 0; \tag{60c}$$

$$\hat{\mathcal{Y}}^v_{\alpha LM} \quad \text{for } v \in \{1, 3, 5\} \text{ and } L > 0. \tag{60d}$$

Using (51), SO(3)-reduced matrix elements of $\hat{\pi}_M$ and ∇^2 are obtained from the SO(5)-reduced matrix elements given in (53) and (59) respectively. Those of the spherical harmonics $\hat{\mathcal{Y}}^v_{\alpha LM}$ are obtained from (39) and (45), with those of $\cos 3\gamma$ similarly obtained, by virtue of (43), from those of the spherical harmonic $\hat{\mathcal{Y}}^3_{100}$. The reduced matrix elements of the remaining operators are obtained from the algebraic expressions of Section 3. In the case of the operators $[\hat{\pi} \otimes \hat{q} \otimes \hat{\pi}]_0$ and $[\hat{\pi} \otimes \hat{\pi}]_{LM}$, previously unpublished expressions for their SO(3)-reduced matrix elements are derived in Appendix D. For the operators involving $\hat{\pi}$, the ACM code calculates, for convenience, the renormalised operators $\hbar^{-2}[\hat{\pi} \otimes \hat{q} \otimes \hat{\pi}]_0$, $\mathrm{i}\,\hbar^{-1}\,\hat{\pi}_M$, and $\hbar^{-2}[\hat{\pi} \otimes \hat{\pi}]_{LM}$ for $L \in \{2, 4\}$.

By design, the ACM code analyses Hamiltonians that are SO(3) invariant (i.e. scalar). Thus, for convenience, the list (60) of operators is partitioned such that only those in (60a) and (60b) are SO(3) invariant. The partitioning is also such that there is some redundancy in that some operators in (60a) can be obtained by multiplying two of those from (60b). However, the matrix elements of the former are obtained directly from a single SU(1, 1) representation, thereby avoiding the inefficiency of a sum over intermediate states.

Further SO(3) invariant operators may be obtained by extracting the scalar-coupled product of two operators of equal (non-zero) angular momentum. This enables, in particular, reduced matrix elements of the scalar components of operators $\hat{\pi} \otimes \hat{q} \otimes \hat{q} \otimes \cdots \otimes \hat{q} \otimes \hat{\pi}$ to be obtained (details are given in Section 7.3.3).

The list (60) of operators is also partitioned such that only those in (60a) and (60c) are rational in the quadrupole moments $\{\hat{q}_M\}$ and their conjugate momenta $\{\hat{\pi}_N\}$. In addition, products of operators (60) which contain an even number of factors from (60b) and (60d)

are also rational. That this is so follows because (i) $\hat{\beta}^2 = \hat{q}\cdot\hat{q} \equiv \sum_M (-1)^M \hat{q}_M \hat{q}_{-M}$, $-\mathrm{i}\hbar\hat{\beta}\partial/\partial\beta = \hat{q}\cdot\hat{\pi} \equiv \sum_M (-1)^M \hat{q}_M \hat{\pi}_{-M}$, and $-\hbar^2\nabla^2 = \hat{\pi}\cdot\hat{\pi} \equiv \sum_M (-1)^M \hat{\pi}_M \hat{\pi}_{-M}$ are each quadratic in $\{\hat{q}_M, \hat{\pi}_N\}$; and (ii) any $\hat{\beta}^v \hat{y}^v_{\alpha LM}$ is a polynomial in the quadrupole moments and hence $\hat{\beta}^{v-2n}\hat{y}^v_{\alpha LM}$ is a rational function of the quadrupole moments for any integer n.

The significance of this is that any observable of the ACM that is rational in $\{\hat{q}_M\}$ and $\{\hat{\pi}_N\}$ is expressed efficiently in the code, being readily obtained from the analytical expressions in Sections 3 and 4. Noting that the matrix elements $F^{(a)}_{\lambda\pm1,\mu;\lambda v}(\hat{\beta}^{\pm1})$ and $F^{(a)}_{\lambda\pm1,\mu;\lambda v}(d/d\beta)$ have analytic expressions, we obtain analytic matrix elements for all rational observables in each basis (8) for which (55) holds. The specific choices for (55) that are used in the ACM are discussed in Section 5.3.

Observables that are not rational in $\hat{q}_M$ and $\hat{\pi}_N$ can also be generated. For example, one can include potential energy terms in a Hamiltonian with odd powers of β. However, the matrix elements of such observables are not obtained analytically, and require a larger truncated Hilbert space to achieve accurate results.

The matrix elements of observables that are polynomial in the above generating operators (60), as well as the scalar-coupled products, are now obtained by combining those for the generating operators calculated in a finite-dimensional subspace $\mathbb{H}^{\mathrm{trunc}}$ of $\mathbb{H}$ spanned by a subset of the SU(1, 1) × SO(5) coupled states (8), specified by the user. In particular, we are able to generate algebraic expressions for any ACM Hamiltonian that is SO(3) and time-reversal invariant, is a polynomial function of the basic $\hat{q}_M$, $\hat{\pi}_M$ observables and β^{-2}, and at most quadratic in the momentum operators.

With respect to a truncated basis $\mathbb{H}^{\mathrm{trunc}}$, the SO(3)-reduced matrix elements of all the operators in (60), apart from $[\hat{\pi}\otimes\hat{q}\otimes\hat{\pi}]_0$ and $[\hat{\pi}\otimes\hat{\pi}]_{LM}$, are determined precisely. However, the SO(3)-reduced matrix elements of operators obtained by combining these operators as described above, as well as the operators $[\hat{\pi}\otimes\hat{q}\otimes\hat{\pi}]_0$ and $[\hat{\pi}\otimes\hat{\pi}]_{LM}$, are subject to truncation errors. One consequence is that these matrix elements do not precisely satisfy the required Hermiticity relationships. However, the discrepancy only occurs close to the truncation boundary. Thus, to ensure that diagonalisation of a model Hamiltonian leads to real eigenvalues and eigenvectors in the ACM code, we retain only the Hermitian component of the truncated matrix by taking the average of the matrix and its Hermitian conjugate. This leaves most of the matrix elements unchanged, and does not affect the low-lying eigenvalues or any of their calculated properties. This is ensured by working with a space sufficiently large such that a further increase in size produces no significant change in the calculated quantities.

5.3. Bases in the ACM

The specification of a particular basis (8) for the Hilbert space $\mathbb{H}$ requires the values of a and λ_v to be set. As discussed in the previous section, analytic expressions for the matrix elements of rational observables are efficiently obtained when λ_v is such that (55) holds.

In our main implementation of the ACM, two solutions to (55) are immediately available. In the first, we set

$$\lambda_v = \lambda_0 + v, \tag{61}$$

for a specified value of λ_0. Such a basis generalises (56) which pertains to the five-dimensional harmonic oscillator basis. In the second, we set

$$\lambda_v = \begin{cases} \lambda_0 & \text{if } v \text{ is even,} \\ \lambda_0 + 1 & \text{if } v \text{ is odd.} \end{cases} \tag{62}$$

For these bases, the matrix elements of rational observables are obtained especially efficiently because there is no need for the code to employ (33).

The ACM code also allows the user to set a constant

$$\lambda_v = \lambda_0. \tag{63}$$

In this case, analytic expressions are still obtained for the matrix elements of parity-preserving rational observables, where an observable $\hat{W}$ is said to be parity-preserving if

$$\langle (a, \lambda_{v_\mathrm{f}})\, \mu;\, v_\mathrm{f}\alpha_\mathrm{f}L_\mathrm{f} \| \hat{W} \| (a, \lambda_{v_\mathrm{i}})\, v;\, v_\mathrm{i}\alpha_\mathrm{i}L_\mathrm{i}\rangle = 0 \tag{64}$$

in all instances where $v_\mathrm{f} - v_\mathrm{i}$ is odd. However, if $\hat{W}$ is not parity-preserving then its matrix elements are obtained by the less precise non-analytic means described in Section 5.4.

Having set λ_v to be a function of λ_0 and v, it remains to set the two values a and λ_0 to determine a suitable basis (8) for the Hilbert space $\mathbb{H}$ on which a particular model Hamiltonian acts. Any positive values for these parameters define a basis. However, given that in a computer implementation, it is necessary to use a truncation $\mathbb{H}^{\mathrm{trunc}}$ of $\mathbb{H}$, and that the calculation time increases dramatically with the dimension of $\mathbb{H}^{\mathrm{trunc}}$, it is important to choose values of a and λ_0 that give convergent results for as few basis states as possible. Methods for optimising the parameters a and λ_0 are discussed in Appendix B. Note that the values of these two parameters have a much more significant effect on the efficiency of the calculation than the choice of basis type (61)–(63).

5.4. Non-rational observables

As discussed above, the use of bases (8) for which (55) holds enables the matrix elements of all observables that are rational functions of the quadrupole and conjugate momentum operators to be obtained precisely. Non-rational observables such as $\hat{\beta}^{\pm1}$ and $d/d\beta$ may also be used, with the λ-conserving matrix elements $F^{(a)}_{\lambda\mu;\lambda v}(\hat{\beta})$ obtained from the equation

$$F^{(a)}_{\lambda\mu;\lambda v}(\hat{\beta}^2) = \sum_\xi F^{(a)}_{\lambda\mu;\lambda\xi}(\hat{\beta}) F^{(a)}_{\lambda\xi;\lambda v}(\hat{\beta}), \tag{65}$$

by taking the positive square root of the matrix $F^{(a)}_{\lambda\mu;\lambda v}(\hat{\beta}^2)$ with the knowledge that the operator $\hat{\beta}$ is positive definite.

The matrix elements $F^{(a)}_{\lambda\mu;\lambda\nu}(\hat{\beta}^{-1})$ can then be obtained by taking the inverse of the matrix $F^{(a)}_{\lambda\mu;\lambda\nu}(\hat{\beta})$ obtained above. The matrix elements of $d/d\beta$ cannot be obtained by taking the square root of the matrix representing $d^2/d\beta^2$ because the latter is not positive definite. Instead, they can be determined by use of the identity

$$F^{(a)}_{\lambda\mu;\lambda\nu}(d/d\beta) = \sum_{\xi} F^{(a)}_{\lambda\mu;\lambda\xi}(\hat{\beta}^{-1}) F^{(a)}_{\lambda\xi;\lambda\nu}(\hat{\beta}\, d/d\beta). \tag{66}$$

Matrix elements of other non-rational observables may be obtained by combining those for $\hat{\beta}^{\pm 1}$ and $d/d\beta$, obtained as described above, with those for the rational observables, including those for the identity operator $\hat{1}$ given by (33). The matrix elements of non-parity-preserving rational operators in the basis (63) may also be obtained from such combinations.

It should be noted that matrix elements obtained non-analytically, as described in this section, suffer somewhat from truncation effects. However, when an operator that yields such matrix elements is used as a term in a Hamiltonian, then in general the truncation does not significantly affect the low-lying eigenstates, and thus is not detrimental to obtaining converged results.

5.5. Reduced E2 transition rates

The standard definition (see [8, Section 2.3.1]) of reduced E2 transition rates between sets of levels in nuclear physics is to take the sum of the squared matrix elements of the E2 transition operator $\hat{Q}^{(E)}_m = (Ze/A)\hat{q}_m$ over the final set of states and then average over the initial states. Thus, for a transition between sets of states of angular momenta L_i and L_f, the SO(3)-reduced E2 transition rate is given by

$$B(\mathrm{E2};\, \alpha_\mathrm{i} L_\mathrm{i} \to \alpha_\mathrm{f} L_\mathrm{f}) = \left(\frac{Ze}{A}\right)^2 \frac{\left|\langle \alpha_\mathrm{f} L_\mathrm{f} \| \hat{q} \| \alpha_\mathrm{i} L_\mathrm{i}\rangle\right|^2}{2L_\mathrm{i}+1}. \tag{67}$$

Here, the multiplicity labels α_i and α_f serve to distinguish between different sets of states with identical angular momenta. The ACM code allows the user to multiply the expression (67) by any convenient factor so that the B(E2) values can be obtained in any convenient units.

Applying the above definition to initial and final sets of states that carry SO(5) irreps gives SO(5)-reduced E2 transition rates $\bar{B}(\mathrm{E2};\, v_\mathrm{i} \to v_\mathrm{f})$. Using (39) then results in the following relationship between these transition rates and the above standard SO(3)-reduced transition rates [4, Section VIIC]:

$$B(\mathrm{E2};\, v_\mathrm{i}\alpha_\mathrm{i} L_\mathrm{i} \to v_\mathrm{f}\alpha_\mathrm{f} L_\mathrm{f}) = (v_\mathrm{f}\alpha_\mathrm{f} L_\mathrm{f},\, 112 \parallel v_\mathrm{i}\alpha_\mathrm{i} L_\mathrm{i})^2 \bar{B}(\mathrm{E2};\, v_\mathrm{i} \to v_\mathrm{f}). \tag{68}$$

6. Using the code

In this section, we describe the basic usage of the ACM code. Although basic, the procedures described here enable a vast range of Hamiltonians to be analysed, with their eigenenergies, transition rates and amplitudes calculated and displayed. With a few minor adjustments to the default settings, the procedures of this section may also be used to carry out calculations in the rigid-β limit of the ACM. How this is done is described in Appendix A.

It is anticipated that many users will find the flexibility offered by the procedures of this section to be more than adequate for their purposes. For those who wish to further process the data calculated, use other Hamiltonians, or examine the transition rates of other operators, various ways in which the functionality of the procedures described in this section may be extended are described in Section 7.

More fundamental procedures of the ACM code are described in Sections 8 and 9: these might be of use in models beyond the ACM that make use of the SO(5) and/or SU(1, 1) matrix elements or the SO(5) $\supset$ SO(3) Clebsch–Gordan coefficients. However, it is suggested that the reader skip Sections 7–9 on a first reading, and return to tackle these sections after gaining some experience in using the procedures of the current section.

A Maple worksheet `acm-examples.mw`, supplied with the code, illustrates use of most of the procedures that are described in this and the following sections.

6.1. Preliminaries

6.1.1. Installation

Before the code can be used, it is necessary to install the files that contain the SO(5) $\supset$ SO(3) Clebsch–Gordan coefficients onto the host computer. These files are supplied in three zipped files named `so5cg-data13.zip`, `so5cg-data24.zip` and `so5cg-data56.zip`. The first of these zipped files contains SO(5) $\supset$ SO(3) Clebsch–Gordan coefficients $(v_1\alpha_1 L_1,\ v_2\alpha_2 L_2 \parallel v_3\alpha_3 L_3)$ for $v_2 = 1$ and $v_2 = 3$; the second contains those for $v_2 = 2$ and $v_2 = 4$; and the third contains those for $v_2 = 5$ and $v_2 = 6$. (For the basic usage described in this section, only the data from the first two files is required, with that from the second then only required if the operator $[\hat{\pi} \otimes \hat{q} \otimes \hat{\pi}]_0$ is present in the Hamiltonian.)

Unzipping the file `so5cg-data13.zip` will create the directory `so5cg-data/` (if it does not already exist) and in that directory create subdirectories `v2=1/` and `v2=3/` containing further subdirectories and data files. Likewise, unzipping the file `so5cg-data24.zip` will create subdirectories `v2=2/` and `v2=4/` of `so5cg-data/`, and unzipping `so5cg-data56.zip` will create subdirectories `v2=5/` and `v2=6/` of `so5cg-data/`.[3]

[3] In some instances, such as when the files are automatically unzipped on downloading, or when a file is unzipped by clicking on it, the subdirectories `v2=1/`, `v2=2/`, `v2=3/`, `v2=4/`, `v2=5/` and `v2=6/` might not appear in the desired common `so5cg-data/` directory. It will then be necessary to move those subdirectories.

Once unzipped, and the location of the initial directory `so5cg-data/` is specified to the program (see below), this subdirectory structure is invisible to the user, and all the required SO(5) ⊃ SO(3) Clebsch–Gordan coefficients will be automatically available to the program. However, it may be useful to know that each of the subdirectories `v2=` v_2`/` contains subdirectories `SO5CG_` v_1`_`v_2`_`v_3`/` for various values of v_1 and v_3, and each subdirectory `SO5CG_` v_1`_`v_2`_`v_3`/` contains files named `SO5CG_` v_1`_`v_2`-`α_2`-`L_2`_`v_3, for various values of α_2 and L_2, that contain the SO(5) ⊃ SO(3) Clebsch–Gordan coefficients.

6.1.2. Initialisation

Whenever the code is used, it is necessary to specify the location of the files that contain the SO(5) ⊃ SO(3) Clebsch–Gordan coefficients. This is achieved by specifying the location of the directory `so5cg-data/` whose subdirectories, as described above, contain the data. This location is specified in the global variable `SO5CG_directory`. A typical declaration for a Unix, Linux or Macintosh system would be[4]

```
SO5CG_directory:="/home/username/maple/acm/so5cg-data/":
```
(69)

For a Windows system, a typical declaration would be[5]

```
SO5CG_directory:="C:/Users/Username/maple/acm/so5cg-data/":
```
(70)

The specification of `SO5CG_directory`, using a command of the form (69) or (70), may be done in the Maple session, immediately after reading the main ACM code from `acm.mpl`. Alternatively, it may be more convenient to make this specification in a settings file. An example of such a file, `acm-user.mpl`, is supplied with the ACM code. The supplied file `acm-user.mpl` also specifies various default values that affect the output of eigenvalues and transition rates, as explained in Sections 6.3 and 6.4. These can be altered to suit the user's requirements.

If used, the settings file should be called by using Maple's `read` command immediately after the main code in `acm.mpl` is called:

```
read "acm.mpl":
read "acm-user.mpl":
```
(71)

These two files *should not* be opened as Maple worksheets. Doing so is likely to corrupt them. They should only be used by calling them from a Maple session or Maple worksheet using the `read` command as described above. In addition, the file `"acm-user.mpl"` should only be edited using a text editor. The file `"acm.mpl"` should not be edited at all.

6.2. The main functionality

This subsection describes the framework in which calculations are made with the ACM code. Firstly, it explains a simple way of specifying Hamiltonians. Secondly, it explains how to specify the truncated Hilbert spaces $\mathbb{H}^{\text{trunc}}$ in which the calculations are carried out. It then describes two procedures, `ACM_Scale` and `ACM_Adapt`, which carry out the diagonalisation of a Hamiltonian on a particular $\mathbb{H}^{\text{trunc}}$, and displays the resulting eigenvalues. These two procedures differ only in the way that the results are scaled before being displayed.

The procedures `ACM_Scale` and `ACM_Adapt` are also able to calculate and display E2 transition rates and amplitudes of the quadrupole operator $\hat{q}$. This is described in Section 6.3.

In Section 6.4, various ways to configure the output of `ACM_Scale` and `ACM_Adapt` are described. Section 6.5 shows how to specify the dependence of λ_v on v for $\mathbb{H}^{\text{trunc}}$, while Section 6.6 describes a procedure which enables optimal values of the adjustable parameters a and λ_0 to be obtained for certain types of Hamiltonians.

6.2.1. Specifying Hamiltonians

The Hamiltonians that can be analysed with this code are polynomials in the operators (60a) and (60b). The coefficients in these polynomials may be arbitrary real numbers, or they may be functions of the quantum numbers ν, v and L of the states $|(a, \lambda_v)\ \nu;\ v\alpha LM\rangle$ on which they operate.

In the ACM code, Hamiltonians and other operators are encoded using a particular list structure, which is described in detail in Section 7.3. However, the user does not need to know this encoding when the Hamiltonian is of the form[6]

$$x_1\nabla^2 + x_2 + x_3\beta^2 + x_4\beta^4 + \frac{x_5}{\beta^2} + x_6\beta\cos 3\gamma + x_7\beta^3\cos 3\gamma + x_8\beta^5\cos 3\gamma + \frac{x_9}{\beta}\cos 3\gamma + x_{10}\cos^2 3\gamma$$
$$+ x_{11}\beta^2\cos^2 3\gamma + x_{12}\beta^4\cos^2 3\gamma + \frac{x_{13}}{\beta^2}\cos^2 3\gamma + \frac{x_{14}}{\hbar^2}\,[\hat{\pi}\otimes\hat{q}\otimes\hat{\pi}]_0. \tag{72}$$

For such a Hamiltonian, the encoding is generated by the procedure call

$$\texttt{ACM\_Hamiltonian}(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14})\ : \tag{73}$$

---

[4] Note that the final "/" is necessary.

[5] The slash "/" may be used to separate the subdirectory names here, even though, on Windows systems, this is usually done with the backslash "\". However, in Maple strings, the backslash is an *escape sequence*, and therefore in order to use the usual Windows form, it is necessary to write `SO5CG_directory:= "C:\\Users\\Username\\maple\\acm\\so5cg-data\\":` instead of (70).

[6] These Hamiltonians are rational in the basic observables $\hat{q}_M$ and $\hat{\pi}_M$, and therefore, as explained in Section 5.2, their matrix elements are obtained efficiently and accurately in the code via analytic expressions.

In typical usage, the value returned by this procedure will be assigned to a Maple variable, such as in

$$\texttt{HOp} := \texttt{ACM\_Hamiltonian}(0.5,\, 0,\, 3.5)\,: \tag{74}$$

This variable may then be used as the first argument to the procedures described in Sections 6.2.3 and 6.2.4. Note that, in using the procedure ACM_Hamiltonian, arguments that are not specified are taken to be 0. Thus, in the example (74), all x_i for $4 \leq i \leq 14$ are set to zero.

As mentioned above, each of the arguments x_i in (73) is either a real number or a real function of ν, v or L. This latter case is accommodated by using the symbolic names NUMBER, SENIORITY and ANGMOM, which are then assigned the values ν, v and L, respectively, of the states acted upon by the Hamiltonian.

This is useful, for example, in examining the rigid-β Wilets–Jean model [27], which has Hamiltonian proportional to $\hat{\Lambda}^2$. In view of (58), the ACM code's encoding of this Hamiltonian is the value returned by

$$\texttt{ACM\_Hamiltonian}(0,\, \texttt{SENIORITY} * (\texttt{SENIORITY} + 3))\,: \tag{75}$$

How to use the ACM code to perform rigid-β calculations is discussed in Appendix A.

6.2.2. Specifying the truncated Hilbert space

In using the ACM code, it is necessary to truncate the Hilbert space $\mathbb{H}$ on which the Hamiltonian and other operators act to a finite-dimensional subspace $\mathbb{H}^{\text{trunc}}$. To do this, we specify non-negative integers $\nu_{\min}$, $\nu_{\max}$, $v_{\min}$, $v_{\max}$, $L_{\min}$, $L_{\max}$, in addition to the parameters λ_0 and a. The subspace $\mathbb{H}^{\text{trunc}}$ is then that spanned by

$$\{|(a, \lambda_v)\ \nu;\, v\alpha LM\rangle,\, \nu_{\min} \leq \nu \leq \nu_{\max},\, 1 \leq \alpha \leq d_{vL},\, v_{\min} \leq v \leq v_{\max},\, L_{\min} \leq L \leq L_{\max},\, -L \leq M \leq L\}, \tag{76}$$

where, in the default case, λ_v is determined by λ_0 using (62). However, by using reduced matrix elements, the code avoids reference to the quantum number M entirely.

The range of states (76) available to the code is restricted only by the availability of SO(5) $\supset$ SO(3) Clebsch–Gordan coefficients. There is thus an upper limit on the maximum seniority $v_{\max}$ of $\mathbb{H}^{\text{trunc}}$, with this limit dependent on the Hamiltonian being analysed. Specific values are given in the *Program summary* section.

Techniques to determine good values of a and λ_0 are discussed in Appendix B. Procedures that implement those techniques for certain types of Hamiltonians are described in Section 6.6.

6.2.3. Diagonalisation using ACM_Scale

For a Hamiltonian $\hat{H}$ encoded in the Maple variable HOp, its action on the truncated Hilbert space $\mathbb{H}^{\text{trunc}}$, spanned by (76), is diagonalised by the procedure call

$$\texttt{ACM\_Scale}(\texttt{HOp},\, a,\, \lambda_0,\, \nu_{\min},\, \nu_{\max},\, v_{\min},\, v_{\max},\, L_{\min},\, L_{\max})\,: \tag{77}$$

The final argument $L_{\max}$ of ACM_Scale may be omitted, in which case its value is taken to be $L_{\max} = L_{\min}$.[7] The call (77) diagonalises the action of $\hat{H}$ separately on each subspace of $\mathbb{H}^{\text{trunc}}$ of angular momentum L, for $L_{\min} \leq L \leq L_{\max}$. For each such L-space, the procedure prints a list of eigenvalues in increasing order. These are given relative to their minimum value across all the L-spaces. This minimal value is displayed. The number of eigenvalues output for each L, as well as their displayed precision may be altered, as described in Section 6.4.1.

The procedure ACM_Scale is also able to calculate and display transition rates and amplitudes of the quadrupole operator $\hat{q}$. As described in Section 6.3, these are produced by specifying, beforehand, two lists of pairs of states between which transition rates and amplitudes are required.

The eigenvalues, transition rates and amplitudes are automatically scaled by certain factors stored by the code before being displayed. These scaling factors may be set explicitly using the procedure ACM_set_scales described in Section 6.4.2. Thus, by setting each of these values to 1 (the default), the raw eigenvalues, transition rates and amplitudes are displayed by ACM_Scale. Note that these stored scaling factors change whenever the procedure ACM_Adapt, described next, is used.

As is usually the case with Maple procedures, the procedure ACM_Scale has a return value. This return value contains the information from which the displayed data is obtained, as well as much more. In basic usage, where it is adequate for the user simply to view the energy eigenvalues, transition rates and amplitudes displayed by ACM_Scale, the return value may be ignored. However, if it is required to further process the displayed information or to obtain other results from the calculation, the value returned by (77) should be assigned to a Maple variable. The format of this return value is described in Section 7.1, but because of its voluminous and somewhat cryptic nature, it is probably not usefully displayed directly. Instead, the data it contains may be displayed by using procedures described in Section 7.2.

6.2.4. Diagonalisation using ACM_Adapt

The procedure call

$$\texttt{ACM\_Adapt}(\texttt{HOp},\, a,\, \lambda_0,\, \nu_{\min},\, \nu_{\max},\, v_{\min},\, v_{\max},\, L_{\min},\, L_{\max})\,: \tag{78}$$

performs a similar function to that of ACM_Scale above. The difference here is that a scaling factor is applied to the relative eigenvalues so that a specified eigenvalue takes a prescribed value. All the other relative eigenvalues are scaled accordingly. The quadrupole transition rates and amplitudes are treated similarly, in that a certain transition rate is set to a specified value, and the required scaling is then used for all other transition rates. A corresponding scaling is applied to the amplitudes.

[7] This will be the case for all procedures in this package for which the final two arguments are $L_{\max}$ and $L_{\min}$. This will not be mentioned henceforth.

It is explained in Section 6.4.3 how the specific values used here are set. With the default settings, the scaling factor for the relative eigenvalues obtained from (78) is set so that the lowest for angular momentum $L = 2$ takes the value 6.0. This enables the results to be quickly compared with those of a rigid rotor which has rotational energies proportional to $L(L+1)$. In the case of the quadrupole transition rates, the scaling factor is set so that the transition rate $B(\mathrm{E2}; 2(1) \rightarrow 0(1))$ takes the value 100.0, where $2(1)$ and $0(1)$ label the lowest eigenvalue states of angular momenta 2 and 0 respectively (cf. (80)).

The scaling factors obtained by `ACM_Adapt` for the eigenvalues, transition rates and amplitudes are retained so that they are used in subsequent calls to the procedure `ACM_Scale`.

The value returned by the procedure `ACM_Adapt` has the same format as that returned by `ACM_Scale` and, as in that case, may be disregarded in basic usage.

6.3. Calculation and display of transition rates and amplitudes

The procedure calls

$$\begin{aligned} &\texttt{ACM\_set\_rat\_lst}(\textit{ratelist}) : \\ &\texttt{ACM\_add\_rat\_lst}(\textit{ratelist}) : \end{aligned} \tag{79}$$

determine which quadrupole transition rates are displayed by subsequent calls to the procedures `ACM_Scale` and `ACM_Adapt`. The first of these specifies a list of transition rate designators, and the second augments it. The argument *ratelist* is a list in which each element itself is a list of up to five integers. A four element designator $[L_\mathrm{i}, L_\mathrm{f}, n_\mathrm{i}, n_\mathrm{f}]$ indicates that the SO(3)-reduced transition rate defined by

$$B(\mathrm{E2}; L_\mathrm{i}(n_\mathrm{i}) \rightarrow L_\mathrm{f}(n_\mathrm{f})) = \frac{\langle n_\mathrm{f} L_\mathrm{f} \| \hat{q} \| n_\mathrm{i}\, L_\mathrm{i} \rangle^2}{2L_\mathrm{i} + 1} \tag{80}$$

(omitting the overall factor $(Ze/A)^2$ from (67)), is to be displayed, where $|n_\mathrm{i} L_\mathrm{i}\rangle$ (resp. $|n_\mathrm{f} L_\mathrm{f}\rangle$) denotes the n_ith (resp. n_fth) lowest energy eigenstate of angular momentum L_i (resp. L_f). The zero, one, two and three element designators [ ], $[L_\mathrm{f}]$, $[L_\mathrm{i}, L_\mathrm{f}]$, $[L_\mathrm{i}, L_\mathrm{f}, n_\mathrm{f}]$ indicate that particular lists of values (80) are to be displayed. For these, the stipulated indices are constant while the other indices from $\{L_\mathrm{i}, L_\mathrm{f}, n_\mathrm{i}, n_\mathrm{f}\}$ take a range. For n_i and n_f, this range is as described in Section 6.4.1. The values of L_i and L_f range over all values between L_min and L_max (specified in (77) and (78)) with $|L_\mathrm{i} - L_\mathrm{f}| \leq 2$. For the five element designator $[L_\mathrm{i}^0, L_\mathrm{f}^0, n_\mathrm{i}, n_\mathrm{f}, L_\mathrm{mod}]$, a sequence of the transition rates (80) is displayed for fixed values of n_i and n_f, and all available pairs L_i and L_f given by $L_\mathrm{i} = L_\mathrm{i}^0 + kL_\mathrm{mod}$ and $L_\mathrm{f} = L_\mathrm{f}^0 + kL_\mathrm{mod}$, for $k = 0, 1, 2, \ldots$. Note that $L_\mathrm{mod} < 0$ is permitted.

The transition amplitudes to be displayed are determined in an analogous way. Thus, the procedure calls

$$\begin{aligned} &\texttt{ACM\_set\_amp\_lst}(\textit{amplist}) : \\ &\texttt{ACM\_add\_amp\_lst}(\textit{amplist}) : \end{aligned} \tag{81}$$

specify a list of designators, which is independent of that specified by (79). Each element of *amplist* is a list of up to five integers. A four element designator $[L_\mathrm{i}, L_\mathrm{f}, n_\mathrm{i}, n_\mathrm{f}]$ indicates that the transition amplitude given by

$$(L_\mathrm{i}, L_\mathrm{i},\ 2, L_\mathrm{f} - L_\mathrm{i} | L_\mathrm{f}, L_\mathrm{f}) \frac{\langle n_\mathrm{f} L_\mathrm{f} \| \hat{q} \| n_\mathrm{i} L_\mathrm{i} \rangle}{\sqrt{2L_\mathrm{f} + 1}}, \tag{82}$$

where the first factor is a standard SO(3) Clebsch–Gordan coefficient, is to be displayed (in the most important case where $|n_\mathrm{f} L_\mathrm{f}\rangle = |n_\mathrm{i} L_\mathrm{i}\rangle$, this reduces to the usual expression for the static quadrupole moment $\langle n_\mathrm{i} L_\mathrm{i} L_\mathrm{i} | \hat{q} | n_\mathrm{i} L_\mathrm{i} L_\mathrm{i} \rangle$). The zero, one, two, three and five element designators in *amplist* indicate that lists of transition amplitudes (82) are to be displayed, with these lists determined as for *ratelist* above.

The procedure calls

$$\begin{aligned} &\texttt{ACM\_show\_rat\_lst}(\textit{show}); \\ &\texttt{ACM\_show\_amp\_lst}(\textit{show}); \end{aligned} \tag{83}$$

return, respectively, the currently stored lists of transition rate designators and transition amplitude designators. If $\textit{show} > 0$ or the argument is omitted then a formatted printout of the designators is output. For $\textit{show} \leq 0$, the procedures acts silently.

Note that the values (80) and (82) are displayed after being scaled. These scaling factors are set whenever `ACM_Adapt` is used. Alternatively, they may be set explicitly using the procedure `ACM_set_scales` described in Section 6.4.2.

Also note that the formulae by which the displayed values (80) and (82) are determined from the SO(3)-reduced matrix elements of $\hat{q}$ may be changed. How to do this is described later in Section 7.4.

6.4. Controlling the output

6.4.1. Setting display precision and datum

The format of the values displayed by the procedures `ACM_Scale` and `ACM_Adapt` can be altered by using the procedure

$$\texttt{ACM\_set\_output}(\textit{precision}, \textit{width}, \textit{precision0}, \textit{show}) : \tag{84}$$

Here, *precision* is the maximal number of digits to be used beyond the decimal point to display floating point values; *width* is the maximal total number of digits to be used to display floating point values; *precision0* is the maximal number of digits to be used beyond the decimal point to display the minimal eigenvalue. Here, and for all the other procedures in the ACM code whose names begin `ACM_set`, if the final parameter *show* is positive then the procedure prints a brief summary of the result of its invocation. This is also the case if this parameter is omitted. If $\textit{show} \leq 0$ then the procedure acts silently.

By default, eigenvalues are displayed relative to their overall minimum value. This behaviour can be changed using the procedure

$$\texttt{ACM\_set\_datum}(datum, show) : \tag{85}$$

For $datum = 0$, this specifies that absolute eigenvalues are displayed instead. The default behaviour is restored by using $datum \geq 1$.

The number of values that appear in the horizontal lists of eigenvalues and quadrupole transition rates displayed by the procedures `ACM_Scale` and `ACM_Adapt` can be constrained by issuing the procedure call

$$\texttt{ACM\_set\_listln}(count_{\text{eigs}}, count_{\text{rats}}, show) : \tag{86}$$

Thereafter, the list of eigenvalues that appears for each value of the angular momentum will contain at most the lowest $count_{\text{eigs}}$ eigenvalues. As explained in Section 6.3, a list of transition rates (80) is produced whenever the *ratelist* argument of (79) contains a designator that has three or fewer indices. If n_{i} or n_{f} is not specified in the designator, then, following the call (86), a list of transition rates is displayed for these indices in the ranges $1 \leq n_{\text{i}} \leq count_{\text{rats}}$ or $1 \leq n_{\text{f}} \leq count_{\text{rats}}$ respectively. The output of transition amplitudes (82) is determined from the list *amplist* of transition amplitude designators in exactly the same way.

6.4.2. Explicit setting of display scaling factors

It is often useful to scale the eigenvalues, transition rates and transition amplitudes calculated by the ACM code. For the procedure `ACM_Scale`, such scaling factors can be specified by using the procedure call

$$\texttt{ACM\_set\_scales}(scale_{\text{eigs}}, scale_{\text{rats}}, show) : \tag{87}$$

Then, after the eigenvalues are calculated across all the L-spaces, and their minimal value determined, the relative eigenvalues with respect to this minimal value are each divided by $scale_{\text{eigs}}$ before being displayed (the minimal value is displayed unscaled). Similarly, each transition rate (80) is divided by the scaling factor $scale_{\text{rats}}$ before being displayed.

This call (87) also sets a scaling factor $scale_{\text{amps}}$ which applies similarly to the transition amplitudes (82). This is determined by $scale_{\text{amps}} = \sqrt{scale_{\text{rats}}}$.

The scaling factors set by `ACM_set_scales` apply to all subsequent invocations of `ACM_Scale`, until reset by `ACM_set_scales`, or the procedure `ACM_Adapt` is used. In this latter case, `ACM_Adapt` resets the values of $scale_{\text{eigs}}$ and $scale_{\text{rats}}$, and sets $scale_{\text{amps}} = \sqrt{scale_{\text{rats}}}$, and applies these to the eigenvalues, transition rates and transition amplitudes that it calculates. How these scaling factors are determined in this case is explained in Section 6.4.3.

The scaling factors that are in force at a given time may be obtained using the procedure call

$$\texttt{ACM\_show\_scales}(show) : \tag{88}$$

This returns a list $[scale_{\text{eigs}}, scale_{\text{rats}}, scale_{\text{amps}}]$ of the three current factors by which eigenvalues, transition rates and transition amplitudes are divided. If $show > 0$ or the argument is omitted then a brief description of the values and their purpose is output. For $show \leq 0$, the procedures acts silently.

6.4.3. Settings for implicit determination of display scaling factors

Here we describe two procedures that specify values that are used by the procedure `ACM_Adapt` to adjust the scaling factors $scale_{\text{eigs}}$ and $scale_{\text{rats}}$ so that a certain eigenvalue and a certain transition rate then take specific values.

The procedure call

$$\texttt{ACM\_set\_eig\_fit}(val_{\text{eig}}, L, n, show) : \tag{89}$$

specifies that if the procedure `ACM_Adapt` is invoked subsequently, then the scaling factor $scale_{\text{eigs}}$ applied to the relative eigenvalues would be chosen such that the nth lowest eigenvalue in the space of angular momentum L takes the value val_{eig}. All the other relative eigenvalues would be scaled accordingly, with the minimal value unscaled.

If either of the two final arguments in (89) is omitted, then the value 1 is used.

The procedure call

$$\texttt{ACM\_set\_rat\_fit}(val_{\text{rat}}, L_{\text{i}}, L_{\text{f}}, n_{\text{i}}, n_{\text{f}}, show) : \tag{90}$$

specifies that if the procedure `ACM_Adapt` is invoked subsequently, then the scaling factor $scale_{\text{rats}}$ applied to the transition rates would be chosen such that the specific transition rate $B(\text{E2}; L_{\text{i}}(n_{\text{i}}) \rightarrow L_{\text{f}}(n_{\text{f}}))$ scales to the value val_{rat}. All the other transition rates would be scaled accordingly, as would the transition amplitudes.

If either of the three final arguments in (90) is omitted, then the value 1 is used.

As described above, each invocation of the procedure `ACM_Adapt` sets three scaling factors: one for the eigenvalues, one for the transition rates, and one for the transition amplitudes. These scaling factors are retained for use by any subsequent calls to the procedure `ACM_Scale`. Note that these scaling factors may be reset also by calling `ACM_set_scales`, as described in Section 6.4.2.

6.5. Setting the basis type

As discussed in Section 5.3, specifying the basis (8) requires the dependence of λ_v on v to be set. This is done by using the procedure call

$$\texttt{ACM\_set\_basis\_type}(basistype, pot_{\text{min}}, show) : \tag{91}$$

Here the argument *basistype* takes either of the four values 0, 1, 2 or 3. For all but the last of these, the argument pot_{min} has no effect. For *basistype* set to 1, this call specifies that the harmonic oscillator basis type (61) is to be used. For *basistype* set to 2, this call specifies that the

parity basis type (62) is to be used. This latter is the default. The constant-λ basis type (63) is selected using this procedure when *basistype* is set to 0.

The use of (91) with *basistype* set to 3, specifies the dependence of λ_v on v given by (B.17) with $\beta_* = pot_{\min}$. As with (61) and (62), this dependence respects the condition (54) and thus enables analytic matrix elements to be used for rational operators. The utility of the basis type given by (B.17) is explained in Appendix B.

Note that to completely determine a basis (8), it is necessary, in addition to specifying a basis type, also to specify values of a and λ_0. These two values are passed directly as arguments to the procedures `ACM_Scale` and `ACM_Adapt`, and also to the more basic procedures described later in Section 8.2. A procedure for estimating optimal values of these parameters is given in the following section.

6.6. Optimizing the parameters

The accuracy of the results obtained using the ACM code is dependent not only on the size of the truncated Hilbert space $\mathbb{H}^{\text{trunc}}$ on which the Hamiltonians operate, but also on the values of the two parameters a and λ_0. In fact, for optimal or near-optimal values of these two parameters, a much smaller truncated Hilbert space is required to achieve accurate results, and the calculation can be completed more quickly and reliably.

In Appendix B, a method is described for estimating optimal values of the parameters a and λ_0 for Hamiltonians of the general form (B.1). The ACM code implements this method for the five-parameter Hamiltonians $\hat{H}_{\text{RWC}}(B, c_1, c_2, \chi, \kappa)$ defined by (B.12), and which were considered in [4]. Such Hamiltonians provide a simple alternative to the more general Hamiltonian (72), and their encoding is obtained using the procedure call

$$\texttt{RWC\_Ham}(B, c_1, c_2, \chi, \kappa) : \tag{92}$$

For these Hamiltonians, the method for estimating the optimal values of a and λ_0 is detailed in (B.12)–(B.16), and carried out using the procedure call

$$\texttt{RWC\_alam}(B, c_1, c_2) : \tag{93}$$

This returns a pair $[a_{opt}, \lambda_{opt}]$, whose components are the estimated optimal values of a and λ_0, respectively (they are independent of χ and κ).

Refinements of these estimates, if desired, may be obtained by inspection of the expectation values (B.16) of the Hamiltonian $\hat{H}_{\text{RWC}}(B, c_1, c_2, \chi, \kappa)$ on the $\nu = v = 0$ basis state. These expectation values are obtained, for given values of a and λ_0, using the procedure call

$$\texttt{RWC\_exp}(B, c_1, c_2, \kappa, a, \lambda_0) : \tag{94}$$

(The expectation values are independent of χ.)

In fact, as explained in Appendix B, the method used by the procedure `RWC_alam` to obtain the optimal values a_{opt} and λ_{opt} treats λ_0 as the function of a given in (B.11), when β_0 is obtained from c_1 and c_2 using (B.15). With λ_0 depending on a, c_1 and c_2 in this way, the expectation value of $\hat{H}_{\text{RWC}}(B, c_1, c_2, \chi, \kappa)$ on the $\nu = v = 0$ basis state is returned by the procedure call

$$\texttt{RWC\_exp\_link}(B, c_1, c_2, \kappa, a) : \tag{95}$$

7. Extending the basic functionality

The procedures `ACM_Scale` and `ACM_Adapt`, described in the previous section, provide a simple means to analyse Hamiltonians in the ACM, calculating and displaying their eigenvalues, transition rates and amplitudes. In this section, we describe means to extend the functionality of these two procedures, and describe the format of the data they return. Procedures are described which can use the return value to readily display eigenvalues, transition rates or amplitudes from the calculation, further to those automatically displayed by `ACM_Scale` and `ACM_Adapt`. Moreover, the data in the return value can be readily accessed and further processed in any way the user sees fit.

7.1. Return values

In this section, we describe the return values of the procedures `ACM_Scale` and `ACM_Adapt`. These contain the calculated eigenvalues on the full truncated Hilbert space $\mathbb{H}^{\text{trunc}}$ and all quadrupole amplitudes between the corresponding eigenvectors. These values are the raw unscaled values.

The procedures `ACM_Scale` and `ACM_Adapt`, invoked as in (77) and (78) respectively, for the Hamiltonian $\hat{H}$ encoded in `HOp`, both return a list of three values:

$$[\textit{eigenvals}, \textit{Melements}, \textit{Lvals}]. \tag{96}$$

The first of these, *eigenvals*, contains the energy eigenvalues for $\hat{H}$ acting on the Hilbert space $\mathbb{H}^{\text{trunc}}$ spanned by (76). Specifically, *eigenvals* is a list of lists, with *eigenvals*$[k][n]$ the eigenvalue of $|n\,L\rangle$, the nth lowest energy eigenstate of angular momentum L, where L is obtained from the list *Lvals* through $L = \textit{Lvals}[k]$. The component *Lvals* of (96) contains, in ascending order, all values of L in the range $L_{\min} \le L \le L_{\max}$, specified by (77) or (78), that have non-zero dimension in $\mathbb{H}^{\text{trunc}}$. Note that these values are not necessarily consecutive.

The second component *Melements* of (96) is a Matrix each of whose elements is itself a Matrix.[8] Each internal Matrix corresponds to a pair (L_f, L_i) of angular momenta from the list *Lvals*. If k_i and k_f are such that $L_i = \textit{Lvals}[k_i]$ and $L_f = \textit{Lvals}[k_f]$, then $\textit{Melements}[k_f, k_i][n_f, n_i]$

[8] In our descriptions of ACM procedures, we use a capitalised "Matrix" to refer to the type that Maple uses for matrices processed by its "LinearAlgebra" package.

is the alternative SO(3)-reduced matrix element

$$\langle n_f L_f \| \hat{W} \| n_i L_i \rangle^\natural \equiv \frac{\langle n_f L_f \| \hat{W} \| n_i L_i \rangle}{\sqrt{2L_f + 1}}. \tag{97}$$

In default usage, the procedures `ACM_Scale` and `ACM_Adapt` calculate the matrix elements (97) for $\hat{W} = \hat{q}$, the quadrupole operator. However, this operator can be exchanged for another as explained in Section 7.4.1.

The data contained in the returned values (96) is most conveniently displayed using the procedures described in the next section.

Note that if `ACM_Scale` or `ACM_Adapt` is invoked with the lists *ratelist* and *amplist* of designators both empty, then the component *Melements* of (96) contains no data (because those procedures do not then need to calculate it for their displayed output).

7.2. Display procedures

In this section, we describe three procedures which conveniently display eigenvalues, transition rates and amplitudes. These display procedures are designed to make direct use of the elements *eigenvals*, *Melements* and *Lvals* of the value (96) returned by `ACM_Scale` and `ACM_Adapt`. This is most conveniently done by setting a Maple variable, say `acmvals`, to the value (96), and using `acmvals[1]`, `acmvals[2]` and `acmvals[3]` respectively for the arguments *eigenvals*, *Melements* and *Lvals* required in the display procedures below. In the descriptions of these procedures, the arguments *eigenvals*, *Melements* and *Lvals* have, therefore, the same format as that described in Section 7.1 for the elements having the same names. However, the display procedures here are ignorant of the origin of the data passed to them, being concerned, only, that it has the correct format. Thus these procedures may be used to display data produced by other means, or perhaps by further processing the data (96) returned by `ACM_Scale` and `ACM_Adapt`. They may also be used to display the data calculated by procedures described later in Section 8.2.

7.2.1. Displaying of eigenvalues

The procedure call

$$\texttt{Show\_Eigs}(\mathit{eigenvals}, \mathit{Lvals}, \mathit{count}_{\mathrm{eigs}}, L_{\min}, L_{\max}) : \tag{98}$$

displays, in a convenient format, the eigenvalues stored in the argument *eigenvals*. As in Section 7.1, *eigenvals* is a list, each element of which is a list of eigenvalues that pertains to the corresponding angular momentum value in the list *Lvals*. For each value of the angular momentum L in the list *Lvals* for which $L_{\min} \le L \le L_{\max}$, the eigenvalues are displayed on a horizontal line, with, at most, the lowest $\mathit{count}_{\mathrm{eigs}}$ eigenvalues shown. In the default usage, the eigenvalues are displayed relative to the lowest value across all the angular momenta in *Lvals*. By using `ACM_set_datum`, as described in Section 6.4.1, absolute values can be displayed instead. These eigenvalues, whether relative or absolute, are also divided by the value $\mathit{scale}_{\mathrm{eigs}}$ specified in the most recent call to `ACM_set_scales` as in (87), or as set by a more recent call to `ACM_Adapt`. The format of the values output is affected by the most recent call to `ACM_set_output` as described in Section 6.4.1.

If the final three arguments are omitted, then $\mathit{count}_{\mathrm{eigs}}$ is taken to be the value set by the most recent call to the procedure `ACM_set_listln` described in Section 6.4.1. If the final two arguments are omitted, then eigenvalues for all values of L in *Lvals* are displayed. If only the final argument is omitted then only the eigenvalues for angular momentum $L_{\min}$ are displayed, and then only if this value is present in *Lvals*. The value returned by the procedure is the lowest (unscaled) eigenvalue across all the angular momenta in *Lvals*.

7.2.2. Displaying of transition rates

The procedure call

$$\texttt{Show\_Rats}(\mathit{Melements}, \mathit{Lvals}, \mathit{ratelist}, \mathit{count}_{\mathrm{rats}}) : \tag{99}$$

displays, in a convenient format, selected transition rates obtained from the values stored in *Melements*, with these values taken to be alternative SO(3)-reduced matrix elements $\langle n_f L_f \| \hat{W} \| n_i L_i \rangle^\natural$. In the default usage, the transition rates that are displayed are the values

$$\frac{\langle n_f L_f \| \hat{W} \| n_i L_i \rangle^2}{2L_i + 1}. \tag{100}$$

These values (100) are displayed after being divided by the current value of $\mathit{scale}_{\mathrm{rats}}$.

In the procedure call (99), the first argument *Melements* is precisely of the form specified in Section 7.1, in that it is a Matrix, each element of which is itself a Matrix. Its elements $\mathit{Melements}[k_f, k_i][n_f, n_i]$ are taken to be alternative SO(3)-reduced matrix elements $\langle n_f L_f \| \hat{W} \| n_i L_i \rangle^\natural$, where $L_i = \mathit{Lvals}[k_i]$ and $L_f = \mathit{Lvals}[k_f]$.

The transition rates obtained from these matrix elements are displayed only for those pairs of states specified in the argument *ratelist*. This argument has precisely the format described in Section 6.3, in that it is a list of designators, each of which is a list of up to five integers. A four element designator $[L_i, L_f, n_i, n_f]$ indicates that the SO(3)-reduced transition rate (100) is to be displayed. The zero, one, two and three element designators $[\,]$, $[L_f]$, $[L_i, L_f]$, $[L_i, L_f, n_f]$ indicate that particular lists of values (100) are to be displayed. For these, the stipulated indices are constant while the other indices from $\{L_i, L_f, n_i, n_f\}$ take a range. For n_i and n_f, transition rates (if available in *Melements*) are displayed for indices in the range $1 \le n_i, n_f \le \mathit{count}_{\mathrm{rats}}$. The indices L_i and L_f range over all values in *Lvals*, but subject to $|L_i - L_f| \le L_{\hat{W}}$, where $L_{\hat{W}} = 2$ in the default usage. For the five element designator $[L_i^0, L_f^0, n_i, n_f, L_{\mathrm{mod}}]$, a sequence of transition rates (100) is displayed for fixed values of n_i and n_f, and all available pairs L_i and L_f given by $L_i = L_i^0 + kL_{\mathrm{mod}}$ and $L_f = L_f^0 + kL_{\mathrm{mod}}$, for $k \ge 0$.

When invoking the procedure `Show_Rats`, either of the final two arguments may be omitted. If the third argument *ratelist* is omitted, the current list of transition rate designators, described in Section 6.3, is used in its place. If the fourth argument $\mathit{count}_{\mathrm{rats}}$ is omitted, the value specified in the most recent call to the procedure `ACM_set_listln` is used in its place.

It is possible to change the formula that is used to calculate the displayed values from the Matrix elements $\mathit{Melements}[k_f, k_i][n_f, n_i]$. How this is done is explained in Section 7.4.2.

Table 1
Radial generating operators.

Operator	Symbolic name	Parity	Comment
$\hat{\beta}^2$	`Radial_b2`	+	
$\hat{\beta}^{-2}$	`Radial_bm2`	+	
$\frac{d^2}{d\beta^2}$	`Radial_D2b`	+	
$\hat{\beta}\frac{d}{d\beta}$	`Radial_bDb`	+	
$\hat{\beta}$	`Radial_b`	−	Non-rational
$\hat{\beta}^{-1}$	`Radial_bm`	−	Non-rational
$\frac{d}{d\beta}$	`Radial_Db`	−	Non-rational
$\hat{S}_0$	`Radial_S0`	+	SU(1, 1) operator
$\hat{S}_+$	`Radial_Sp`	+	SU(1, 1) operator
$\hat{S}_-$	`Radial_Sm`	+	SU(1, 1) operator

7.2.3. Displaying of transition amplitudes

The procedure call

$$\texttt{Show\_Amps}(\mathit{Melements}, \mathit{Lvals}, \mathit{amplist}, \mathit{count}_{\text{amps}}) : \tag{101}$$

displays, in a convenient format, selected transition amplitudes obtained from the values stored in *Melements*, with these values assumed to be alternative SO(3)-reduced matrix elements $\langle n_f L_f \| \hat{W} \| n_i L_i \rangle^{\natural}$. In the default usage, the transition amplitudes that are displayed are the values

$$(L_i, L_i,\ L_{\hat{W}}, L_f - L_i | L_f, L_f) \langle n_f L_f \| \hat{W} \| n_i L_i \rangle^{\natural}, \tag{102}$$

where $L_{\hat{W}} = 2$. These values (102) are displayed after being divided by the current value of $\mathit{scale}_{\text{amps}}$ which, as described in Section 6.4.2, is given by $\mathit{scale}_{\text{amps}} = \sqrt{\mathit{scale}_{\text{rats}}}$.

The arguments *Melements* and *Lvals* are exactly as for `Show_Rats` in the previous section, while the arguments *amplist* and $\mathit{count}_{\text{amps}}$ determine the transition amplitudes that are displayed in the same way as the corresponding arguments of (99) determine the values displayed there.

As for `Show_Rats`, either of the final two arguments of `Show_Amps` may be omitted. If the third argument *amplist* is omitted, the current list of transition amplitude designators, described in Section 6.3, is used in its place. If the fourth argument $\mathit{count}_{\text{amps}}$ is omitted, the value $\mathit{count}_{\text{rats}}$ specified in the most recent call to the procedure `ACM_set_listln` is used in its place.

It is possible to change the formula that is used to calculate the displayed values from the Matrix elements $\mathit{Melements}[k_f, k_i][n_f, n_i]$. How this is done is explained in Section 7.4.3.

7.3. Internal representation of Hamiltonians and other operators

Here we describe the format used to encode all operators on the Hilbert space $\mathbb{H}$ that are available in the ACM. In particular, this enables `ACM_Scale` and `ACM_Adapt` to analyse the full range of Hamiltonians available in the ACM, beyond those supplied by the procedure `ACM_Hamiltonian` of Section 6.2.1. It also enables the specification of other operators for which the transition rates and amplitudes are to be calculated.

The Hamiltonian and other operators that are used by the ACM code are encoded using nested Maple lists. Each such operator is formed from the operators given in Tables 1–3.

7.3.1. Products of scalar operators

Consider a tensor operator of the form

$$\hat{W} = \sum_{k=1}^{N} c_k \hat{W}_{k1} \otimes \hat{W}_{k2} \otimes \hat{W}_{k3} \otimes \cdots \otimes \hat{W}_{kM_k}, \tag{103}$$

where $N \geq 0$, each $M_k \geq 0$, each c_k is a constant, and each $\hat{W}_{ki}$ is a zero angular momentum operator from Tables 1, 2 or 3 (each operator in Table 1 has zero angular momentum, while for operators in Tables 2 and 3, the angular momenta are given in the column labelled by A.M.). The restriction to operators of zero angular momentum ensures the operator $\hat{W}$ is itself of zero angular momentum, and thus SO(3)-invariant, permitting its use as a Hamiltonian in the ACM. In the ACM code, this tensor operator $\hat{W}$ is encoded by the nested list

$$[[co_1, [op_{11}, op_{12}, \ldots, op_{1M_1}]], [co_2, [op_{21}, op_{22}, \ldots, op_{2M_2}]], \ldots, [co_N, [op_{N1}, op_{N2}, \ldots, op_{NM_N}]]], \tag{104}$$

where op_{ki} is the symbolic name in the tables that corresponds to the operator $\hat{W}_{ki}$, and co_k corresponds to the constant c_k in (103). These constants can be simple numerical values or they can be functions of the radial quantum number v, the seniority v and the angular momentum L of the state on which the operator (eventually) acts. The constant c_k is encoded in co_k by using the symbolic names `NUMBER`, `SENIORITY` or `ANGMOM` to represent these three values respectively. For example, the Hamiltonian

$$-\frac{1}{80}\nabla^2 + 20(-\beta^2 + \beta^5) + \frac{3}{2}\beta^2 \cos^3 3\gamma \tag{105}$$

Table 2
Spherical tensor operators.

Tensor operator	Symbolic name	A.M.	Parity	Comment
$\hat{y}^0_{10}$	SpHarm_010	0	+	$\hat{y}^0_{100} = \frac{\sqrt{3}}{4\pi}$
$\hat{y}^1_{12}$	SpHarm_112	2	−	$\hat{y}^1_{12M} = \frac{\sqrt{15}}{4\pi}\hat{Q}_M$
$\hat{y}^2_{12}$	SpHarm_212	2	+	$\hat{y}^2_{12M} = -\frac{1}{4\pi}\sqrt{\frac{105}{2}}[\hat{Q}\otimes\hat{Q}]_{2M}$
$\hat{y}^2_{14}$	SpHarm_214	4	+	$\hat{y}^2_{14M} = \frac{1}{4\pi}\sqrt{\frac{105}{2}}[\hat{Q}\otimes\hat{Q}]_{4M}$
$\hat{y}^3_{10}$	SpHarm_310	0	−	$\hat{y}^3_{100} = \frac{3}{4\pi}\cos 3\gamma$
$\hat{y}^3_{13}$	SpHarm_313	3	−	
$\hat{y}^3_{14}$	SpHarm_314	4	−	
$\hat{y}^3_{16}$	SpHarm_316	6	−	$\hat{y}^3_{16M} = \frac{3}{4\pi}\sqrt{\frac{35}{2}}[\hat{Q}\otimes\hat{Q}\otimes\hat{Q}]_{6M}$
$\hat{y}^4_{12}$	SpHarm_412	2	+	
$\hat{y}^4_{14}$	SpHarm_414	4	+	
$\hat{y}^4_{15}$	SpHarm_415	5	+	
$\hat{y}^4_{16}$	SpHarm_416	6	+	
$\hat{y}^4_{18}$	SpHarm_418	8	+	
$\hat{y}^5_{12}$	SpHarm_512	2	−	
$\hat{y}^5_{14}$	SpHarm_514	4	−	
$\hat{y}^5_{15}$	SpHarm_515	5	−	
$\hat{y}^5_{16}$	SpHarm_516	6	−	
$\hat{y}^5_{17}$	SpHarm_517	7	−	
$\hat{y}^5_{18}$	SpHarm_518	8	−	
$\hat{y}^5_{1,10}$	SpHarm_51A	10	−	
$\hat{y}^6_{10}$	SpHarm_610	0	+	$\hat{y}^6_{100} = \frac{\sqrt{15}}{8\pi}(3\cos^2 3\gamma - 1)$
$\hat{y}^6_{13}$	SpHarm_613	3	+	
$\hat{y}^6_{14}$	SpHarm_614	4	+	
$\hat{y}^6_{16}$	SpHarm_616	6	+	
$\hat{y}^6_{26}$	SpHarm_626	6	+	
$\hat{y}^6_{17}$	SpHarm_617	7	+	
$\hat{y}^6_{18}$	SpHarm_618	8	+	
$\hat{y}^6_{19}$	SpHarm_619	9	+	
$\hat{y}^6_{1,10}$	SpHarm_61A	10	+	
$\hat{y}^6_{1,12}$	SpHarm_61C	12	+	

Table 3
Tensor operators on product space $\mathbb{H}$.

Tensor operator	Symbolic name	A.M.	Parity	Comment
$\mathrm{i}\,\hbar^{-1}\,\hat{\pi}$	Xspace_Pi	2	−	See (53)
$\hbar^{-2}\,[\hat{\pi}\otimes\hat{\pi}]_2$	Xspace_PiPi2	2	+	See (D.11)
$\hbar^{-2}\,[\hat{\pi}\otimes\hat{\pi}]_4$	Xspace_PiPi4	4	+	See (D.11)
$\hbar^{-2}\,[\hat{\pi}\otimes\hat{q}\otimes\hat{\pi}]_0$	Xspace_PiqPi	0	+	See (D.12), (D.14)

is encoded

```
[[−1/80, [Radial_D2b]], [(SENIORITY ∗ (SENIORITY + 3) + 2)/80, [Radial_bm2]],
    [−20, [Radial_b2]], [20, [Radial_b2,Radial_b2,Radial_b]],
    [3/2 ∗ Convert_310^3, [Radial_b2,SpHarm_310,SpHarm_310,SpHarm_310]]],
```
(106)

having used the expression (59) for the matrix elements of the Laplacian ∇^2. Here, `Convert_310` is a symbolic name which, as indicated in Table 4, evaluates to $4\pi/3$, this being the factor that, according to (43), converts from y^3_{100} to $\cos 3\gamma$. The particular constants listed in Table 4 are the inverses of those that appear in the last column of Table 2.

Note that the procedure `ACM_Hamiltonian`, described in Section 6.2.1, produces the encoding (104) of rational Hamiltonians in many cases of interest.

7.3.2. *Products involving one non-scalar operator*

Spherical tensor operators $\hat{W}$ of non-zero angular momentum L_1 are encoded in the ACM in a similar way. Here, $\hat{W}$ is of the form (103) as above, with each $\hat{W}_{ki}$ an operator from Tables 1, 2 or 3, restricted such that for each k, there is exactly one operator $\hat{W}_{ki}$ having angular momentum L_1, with all the remaining $\hat{W}_{ki}$ having zero angular momentum. This operator $\hat{W}$ is then also encoded using the nested list (104).

Table 4
Predefined constants.

Symbolic name	Constant
`Convert_112`	$\frac{4}{\sqrt{15}}\pi$
`Convert_212`	$-4\pi\sqrt{\frac{2}{105}}$
`Convert_310`	$\frac{4}{3}\pi$
`Convert_316`	$\frac{4}{3}\sqrt{\frac{2}{35}}\,\pi$
`Convert_610`	$\frac{8}{\sqrt{15}}\pi$
`Convert_red`	$\frac{1}{4\pi}$

A simple example is provided by the quadrupole operator $\hat{q}$, for which the ACM code contains the assignment

$$\texttt{quad\_op} := [[\texttt{Convert\_112}, [\texttt{Radial\_b}, \texttt{SpHarm\_112}]]] : \tag{107}$$

As indicated in Table 4, the symbolic name `Convert_112` evaluates to $4\pi/\sqrt{15}$, this being the factor appearing in the expression $q_M = (4\pi/\sqrt{15})\beta Y^1_{12M}$ (see (41)). Internally, the ACM code makes use of this encoding (107) of the quadrupole operator $\hat{q}$ to obtain values of its transition rates.

The user may define other operators in this way, and then determine their transition rates and amplitudes as described in Section 7.4.

7.3.3. Scalar-coupled products

The specification of operators described above permits at most one operator of non-zero angular momentum in each summand of (103). This is because, on the one hand, a spherical tensor operator $\hat{W}_{ki}$ is represented in the computer as a matrix of alternative reduced elements $\langle n_f L_f \| \hat{W}_{ki} \| n_i L_i \rangle^\natural$ and, on the other, the matrix representation of the tensor product of two non-scalar tensors is not generally a product of their reduced matrices, whether alternative or not. However, with a small adjunct, it is possible to extend the construction to include scalar-coupled products of pairs of tensors of the same angular momentum. The extension makes use of the expression for the matrix elements of a scalar-coupled product of operators $[\hat{A}_{L_1} \otimes \hat{B}_{L_1}]_0$ in an orthonormal basis $\{|\eta LM\rangle\}$[9]:

$$\langle \xi L_f \| [\hat{A}_{L_1} \otimes \hat{B}_{L_1}]_0 \| \eta L_i \rangle^\natural = \delta_{L_f,L_i} \frac{(-1)^{L_1}}{\sqrt{2L_1+1}} \sum_{\zeta L'} \frac{(-1)^{L_f}}{\sqrt{2L_f+1}} \langle \xi L_f \| \hat{A}_{L_1} \| \zeta L' \rangle^\natural (-1)^{L'} \sqrt{2L'+1}\ \langle \zeta L' \| \hat{B}_{L_1} \| \eta L_i \rangle^\natural, \tag{108}$$

which is seen to be a matrix element in the product of a constant and four matrices, two of which are diagonal. For example, the operator $\left[[\hat{\pi} \otimes \hat{\pi}]_2 \otimes \hat{Y}^4_{12}\right]_0$, which is represented as a matrix with elements $\langle (a, \lambda_{v_f})\ \mu;\ v_f \alpha_f L \| \left[[\hat{\pi} \otimes \hat{\pi}]_2 \otimes \hat{Y}^4_{12}\right]_0 \| (a, \lambda_{v_i})\ \nu;\ v_i \alpha_i L \rangle^\natural$ on each L-space, is encoded by the Maple list

$$[[1/\texttt{sqrt}(5), [\texttt{SpDiag\_sqLdiv}, \texttt{Xspace\_PiPi2}, \texttt{SpDiag\_sqLdim}, \texttt{SpHarm\_412}]]]. \tag{109}$$

Here, the use of `SpDiag_sqLdim` and `SpDiag_sqLdiv` effect multiplication by diagonal matrices R and R^{-1} defined by the expectation values of an operator $\hat{R}$ having non-zero matrix elements

$$\langle (a, \lambda)\ \nu;\ v\alpha L \| \hat{R} \| (a, \lambda)\ \nu;\ v\alpha L \rangle^\natural = (-1)^L \sqrt{2L+1}. \tag{110}$$

The facility described in this section enables, in particular, the alternative SO(3)-reduced matrix elements of operators $[\hat{\pi} \otimes \hat{q} \otimes \hat{q} \otimes \cdots \otimes \hat{q} \otimes \hat{\pi}]_0$ to be obtained. The commutation relations $[\hat{q}_M, \hat{\pi}_N] = (-1)^M i\hbar\delta_{-M,N}$ enable this operator to be expressed in terms of the scalar components of $\hat{\pi} \otimes [\hat{q} \otimes \hat{q} \otimes \cdots \otimes \hat{q}]_L$ for $L = 2$ and $[\hat{\pi} \otimes \hat{\pi}]_L \otimes [\hat{q} \otimes \hat{q} \otimes \cdots \otimes \hat{q}]_L$ for $L \in \{0, 2, 4\}$. In both cases, each $[\hat{q} \otimes \hat{q} \otimes \cdots \otimes \hat{q}]_{LM}$ can be written as a linear combination of terms $\hat{\beta}^n \hat{Y}^v_{\alpha LM}$. Then, for the $L = 0$ cases, the required matrix elements are obtained directly after noting that $[\hat{\pi} \otimes \hat{\pi}]_0 = -\hbar^2 \nabla^2$, while for the $L \in \{2, 4\}$ cases, they are obtained by calculating those of $\hat{\beta}^n \left[\hat{\pi} \otimes \hat{Y}^v_{\alpha 2}\right]_0$, and $\hat{\beta}^n \left[[\hat{\pi} \otimes \hat{\pi}]_L \otimes \hat{Y}^v_{\alpha L}\right]_0$, using the means described above.

Note that although the operator $[\hat{\pi} \otimes \hat{q} \otimes \hat{\pi}]_0$ could be tackled in this way, more accurate results are obtained by making use of the symbolic name `Xspace_PiqPi`, which directs the ACM code to use the exact expressions for the matrix elements derived in Appendix D.4.

7.4. Other transition rates

By default, the procedures `ACM_Scale` and `ACM_Adapt` display transition rates and amplitudes of the quadrupole operator $\hat{q}$ that are calculated using (80) and (82) respectively. However, the user might wish to calculate such quantities for another operator. For example, computing the matrix elements of β^2 would furnish information on the beta fluctuations in model states.

In this section, we indicate how to specify that another operator $\hat{W}$ be used in place of $\hat{q}$, and, in addition, how to change the formulae by which the values displayed are obtained from the reduced matrix elements of $\hat{W}$.

[9] This is a re-expression of eqn. (A.48) of [8]. It also follows from eqn. (7.1.1) of [23].

Table 5
Predefined functions of L_i, L_f and M_{el}.

Function name	Dependence on L_i, L_f & M_{el}	Comment
`quad_amp_fun`	$M_{el}(L_i, L_i, L_{\hat{W}}, L_f - L_i \vert L_f, L_f)$	default
`mel_amp_fun`	$M_{el}\sqrt{2L_f+1}$	
`unit_amp_fun`	M_{el}	
`quad_rat_fun`	$M_{el}^2(2L_f+1)/(2L_i+1)$	default
`mel_rat_fun`	$M_{el}^2(2L_f+1)$	
`unit_rat_fun`	M_{el}^2	

7.4.1. Specifying the operator

Let the operator $\hat{W}$ be encoded in the Maple variable *TrOp* as described in Section 7.3. After the procedure call

`ACM_set_transition`(*TrOp*, *show*) : (111)

subsequent use of the procedures `ACM_Scale` and `ACM_Adapt` will calculate and display transition rates for the operator $\hat{W}$. In addition, the component *Melements* of the value (96) returned by `ACM_Scale` and `ACM_Adapt` will then contain the alternative SO(3)-reduced matrix elements $\langle n_f L_f \| \hat{W} \| n_i\, L_i \rangle^\natural$ exactly as described in Section 7.1.

When (111) is invoked, the angular momentum $L_{\hat{W}}$ of the operator $\hat{W}$ is determined and stored. This is used to limit the range of angular momenta for which lists of transition rates and amplitudes are displayed in the procedures `ACM_Scale`, `ACM_Adapt`, `Show_Rats` and `Show_Amps`. The procedures `ACM_Scale`, `ACM_Adapt` and `Show_Amps` also make use of $L_{\hat{W}}$ to calculate the transition amplitudes they display, through the formula (102) (*not* (82)), unless this formula has been changed as described in Section 7.4.3.

7.4.2. Specifying transition rate formula

The procedures `ACM_Scale` and `ACM_Adapt` display two sets of values calculated from the reduced matrix elements $\langle n_f L_f \| \hat{W} \| n_i\, L_i \rangle^\natural$. In the case of the first set, values are displayed for various L_i, n_i, L_f and n_f determined by the set *ratelist* as described in Section 6.3. In the default implementation, these values are calculated using (100). Here, we describe how to use, instead, a different expression, so that values other than transition rates can be displayed.

The procedure call

`ACM_set_rat_form`(*ratfunc*, *ratformat*, *ratdesg*, *show*) : (112)

specifies how, in subsequent use of the procedures `ACM_Scale`, `ACM_Adapt` and `Show_Rats`, displayed values are calculated from the matrix elements $\langle n_f L_f \| \hat{W} \| n_i\, L_i \rangle^\natural$. It also specifies the format used to display these values.

In (112), the argument *ratfunc* should be a Maple procedure which takes three arguments. The value displayed by `ACM_Scale` and `ACM_Adapt` is then *ratfunc*(L_i, L_f, M_{el}) divided by $scale_{rats}$, where $M_{el} = \langle n_f L_f \| \hat{W} \| n_i\, L_i \rangle^\natural$. The value displayed by `Show_Rats` is obtained in the same way, where as described in Section 7.2.2, the elements of the argument *Melements* in (99) are taken to be alternative reduced matrix elements $\langle n_f L_f \| \hat{W} \| n_i\, L_i \rangle^\natural$. The procedure *ratfunc* can be any that takes three numerical arguments. The user can define such a procedure, or make use of one of those that are predefined in the ACM code; these predefined functions are listed in Table 5.

If the quadrupole operator has been exchanged for another as described in Section 7.4.1, or the function used to calculate the transition rates is altered, it would be appropriate to change the format in which each value is displayed. This is done with the second argument *ratform* in (112) which should be a Maple string that contains the desired format in the style of a C programming language format string.[10] Briefly, the string *ratform* should contain two format specifications '`%s`'. For each required L_i, n_i, L_f and n_f, the string *ratform* is printed with the first '`%s`' replaced by '$L_i(n_i) \rightarrow L_f(n_f)$' with the appropriate values substituted, and the second '`%s`' replaced by the calculated value *ratfunc*$(L_i, L_f, M_{el})/scale_{rats}$. As an example, consider the default usage, which is obtained by invoking

```
ACM_set_rat_form( quad_rat_fun, "B(E2; %s) = %s", "transition rates", 0 ):
```
(113)

Thereupon, the display of a typical transition rate takes the form

```
B(E2; 4(1) -> 2(1)) = 149.67.
```
(114)

The third argument *ratdesg* in (112) is a Maple string which contains a phrase, such as `"transition rates"`, that is used in the output to introduce the values being displayed. Having an appropriate such phrase in the output would assist anyone reviewing a calculation, especially if either the quadrupole operator has been exchanged for a different operator, or the function used to calculate the transition rates is altered.

In the call (112), if any argument is omitted, then the previously set value of that parameter is retained.

7.4.3. Specifying transition amplitude formula

The second set of values obtained from the reduced matrix elements $\langle n_f L_f \| \hat{W} \| n_i\, L_i \rangle^\natural$ that are displayed by the procedures `ACM_Scale` and `ACM_Adapt`, are for those L_i, n_i, L_f and n_f determined by the set *amplist* as described in Section 6.3. In the default implementation, these values are calculated using (102). Here, we describe an analogue to the procedure given in the previous subsection, which enables the expression used to calculate these values to be replaced by another.

[10] Consult the Maple help entry for 'printf', or any manual on the C programming language.

The procedure call

`ACM_set_amp_form`(*ampfunc*, *ampformat*, *ampdesg*, *show*) : (115)

specifies how, in subsequent use of the procedures `ACM_Scale`, `ACM_Adapt` and `Show_Amps`, displayed values are calculated from the matrix elements $\langle n_f L_f \| \hat{W} \| n_i L_i \rangle^\natural$. It also specifies the format used to display these values.

The format of the arguments *ampfunc*, *ampformat* and *ampdesg* is precisely that of *ratfunc*, *ratformat* and *ratdesg* described in the previous subsection, and they have the analogous effect on the display of the values designated by *amplist*. As an example, consider the default implementation, which is obtained by invoking

```
ACM_amp_rat_form( quad_amp_fun, "Amp( %s ) = %s", "transition amplitudes", 0 ):
```
(116)

A typical value is then displayed in the form

```
Amp(2(1) -> 2(1)) = 5.25.
```
(117)

Note that having the capacity to change the functions through which the matrix elements designated in the sets *ratelist* and *amplist* are displayed in the procedures `ACM_Scale` and `ACM_Adapt`, means there is little logical distinction between these two sets, other than that those from *ratelist* are displayed before those from *amplist*. In fact, by interchanging the parameters from the default calls to `ACM_set_rat_form` and `ACM_set_amp_form`, transition amplitudes would be displayed before transition rates. However, when using `ACM_Adapt`, there is the distinction that the scaling factor $scale_{\text{rats}}$ is chosen so that one of the alternative reduced matrix elements, processed by the function *ratfunc*, takes a certain value. Thus, if the above interchange is made, this scaling factor would be chosen so that a certain *transition amplitude* takes a specified value. To ensure that the correct scaling factor is then applied to the values designated by *amplist*, the function described in the next subsection would need to be called.

7.4.4. Specifying dependence between scale factors

In the default implementation, the procedures `ACM_Scale` and `ACM_Adapt` display the values of the transition rates (80) and transition amplitudes (82) after dividing them, respectively, by the scale factors $scale_{\text{rats}}$ and $scale_{\text{amps}}$ in force at that time, as described in Section 6.4.2. These scale factors are similarly employed by the procedures `Show_Rats` and `Show_Amps`. Because the transition rates vary as the square of the transition amplitudes, it is appropriate that $scale_{\text{amps}} = \sqrt{scale_{\text{rats}}}$ in this case. However, given the flexibility offered by the procedures of the previous section, this relationship may cease to be appropriate. Here, we provide a means to change it, although it is expected that this would seldom need to be done.

The procedure call

`ACM_set_sft_fun`($scalefunc_{\text{amps}}$, *show*) : (118)

sets the function by which the scale factor $scale_{\text{amps}}$ is obtained from $scale_{\text{rats}}$, to the function given in the argument $scalefunc_{\text{amps}}$. It would be appropriate to invoke `ACM_set_sft_fun` if the arguments *ratfunc* and *ampfunc* supplied to the most recent calls to `ACM_set_rat_form` and `ACM_set_amp_form` are no longer respectively quadratic and linear in the matrix elements $\langle n_f L_f \| \hat{W} \| n_i L_i \rangle^\natural$. The default functionality in which $scale_{\text{amps}} = \sqrt{scale_{\text{rats}}}$ is obtained by using `ACM_set_sft_fun(sqrt_fun,0)` (the procedure `sqrt_fun`, defined in the ACM code, returns the numerical square root of its argument: it has the necessary *type* that allows it to be passed to procedures in the ACM code that expect a procedure argument[11]).

8. Basic procedures for ACM calculations

In this and the following section, we describe a number of other procedures that are in the ACM code. Many of these are called by the procedures `ACM_Scale` and `ACM_Adapt` to perform the default implementation of the ACM described in Sections 6 and 7. This additional information about the code is included for the benefit of anyone wishing to extend the functionality of the code.

8.1. Dimensions and labels

A number of procedures are available to calculate the dimensions of various spaces used by the model. Firstly, the procedure calls

`dimSO3`(L) :
`dimSO5`(v) : (119)

return the dimensions $2L + 1$ and $\frac{1}{6}(v + 1)(v + 2)(2v + 3)$ of the angular momentum L irrep of SO(3) and the seniority v irrep of SO(5) respectively.

On restriction from SO(5) to SO(3), the seniority v irrep of the former decomposes into various irreps of the latter, some of which may have identical SO(3) angular momenta. The multiplicity d_{vL} of the SO(3) irrep of angular momentum L in this restriction is given by (6). This value is returned by the first of the following procedures:

`dimSO5r3`(v, L) :
`dimSO5r3_allL`(v) :
`dimSO5r3_rngVallL`($v_{\min}$, $v_{\max}$) :
`dimSO5r3_rngVvarL`($v_{\min}$, $v_{\max}$, $L_{\min}$, $L_{\max}$) : (120)

[11] Because of its type in recent versions of Maple, the Maple function `sqrt` cannot be used directly in this way.

The other procedures here take the sum of this value over certain ranges of v and L. The second procedure returns $\sum_{L=0}^{\infty} d_{vL}$, which is the total number of SO(3) irreps in the SO(5) irrep of seniority v. The third procedure returns $\sum_{v=v_{\min}}^{v_{\max}} \sum_{L=0}^{\infty} d_{vL}$, and the fourth returns $\sum_{v=v_{\min}}^{v_{\max}} \sum_{L=L_{\min}}^{L_{\max}} d_{vL}$.

The following procedures generate labels for the representations enumerated above.

$$\begin{aligned} &\texttt{lbsSO5r3\_allL}(v) : \\ &\texttt{lbsSO5r3\_rngVallL}(v_{\min}, v_{\max}) : \\ &\texttt{lbsSO5r3\_rngVvarL}(v_{\min}, v_{\max}, L_{\min}, L_{\max}) : \end{aligned} \tag{121}$$

The first of these returns a list of all pairs $[\alpha, L]$ for which $1 \le \alpha \le d_{vL}$. Thus, each element of the returned list corresponds to an SO(3) irrep of angular momentum L in the SO(5) irrep of seniority v. In accordance with (6), L is limited to values $L \le 2v$. The second procedure produces a list of all triples $[v, \alpha, L]$ for the range $v_{\min} \le v \le v_{\max}$ of seniorities, with $1 \le \alpha \le d_{vL}$. The third procedure here produces a list of all labels $[v, \alpha, L]$ for $v_{\min} \le v \le v_{\max}$, $1 \le \alpha \le d_{vL}$ and $L_{\min} \le L \le L_{\max}$.

For the radial space, we have the somewhat trivial analogues of the above procedures:

$$\begin{aligned} &\texttt{dimRadial}(\nu_{\min}, \nu_{\max}) : \\ &\texttt{lbsRadial}(\nu_{\min}, \nu_{\max}) : \end{aligned} \tag{122}$$

The first of these returns the number of radial states ν with $\nu_{\min} \le \nu \le \nu_{\max}$. This number is simply $\nu_{\max} - \nu_{\min} + 1$. The second returns a list of the values ν for which $\nu_{\min} \le \nu \le \nu_{\max}$.

From (76), it is seen that each truncated Hilbert space $\mathbb{H}^{\text{trunc}}$ used in the model is a direct product of truncated spherical and radial spaces. The dimensions of the spaces $\mathbb{H}^{\text{trunc}}$, and labels for their basis states, are obtained using the procedure calls:

$$\begin{aligned} &\texttt{dimXspace}(\nu_{\min}, \nu_{\max}, v_{\min}, v_{\max}, L_{\min}, L_{\max}) : \\ &\texttt{lbsXspace}(\nu_{\min}, \nu_{\max}, v_{\min}, v_{\max}, L_{\min}, L_{\max}) : \end{aligned} \tag{123}$$

The second of these returns a list, each element of which is a length four list $[\nu, v, \alpha, L]$. In this list of states, L varies the slowest, followed by v, then α, with ν varying the fastest. This order of states is used when constructing the matrices that represent the Hamiltonian and other operators on the truncated Hilbert spaces $\mathbb{H}^{\text{trunc}}$.

8.2. Diagonalising Hamiltonians and determining transition rates

In this section, we describe the three procedures which `ACM_Scale` and `ACM_Adapt` use to perform the main calculations. These procedures may, of course, be used independently of one another.

8.2.1. Obtaining matrix representations

Let the operator $\hat{W}$ be encoded in the Maple variable *WOp* as described in Section 7.3. Then, the procedure call

$$\texttt{RepXspace}(WOp, a, \lambda_0, \nu_{\min}, \nu_{\max}, v_{\min}, v_{\max}, L_{\min}, L_{\max}) : \tag{124}$$

obtains the representation of the operator $\hat{W}$ on the truncated Hilbert space $\mathbb{H}^{\text{trunc}}$ that is specified as in Section 6.2.2. The return value is a Matrix whose elements are the alternative SO(3)-reduced matrix elements[12]

$$\langle (a, \lambda_{v_f})\, \mu;\, v_f \alpha_f L_f \| \hat{W} \| (a, \lambda_{v_i})\, \nu;\, v_i \alpha_i L_i \rangle^{\natural} \equiv \frac{\langle (a, \lambda_{v_f})\, \mu;\, v_f \alpha_f L_f \| \hat{W} \| (a, \lambda_{v_i})\, \nu;\, v_i \alpha_i L_i \rangle}{\sqrt{2L_f + 1}}, \tag{125}$$

where λ_{v_i} and λ_{v_f} are determined from λ_0 as described in Section 6.5. The rows and columns of this matrix are labelled by the reduced states $|(a, \lambda_v)\, \nu;\, v \alpha L\rangle$ in such a way that L varies the slowest, followed by v, then α, with ν varying the fastest (this accords with the order returned by the procedure `lbsXspace` which was described in Section 8.1). The procedures `DigXspace` and `AmpXspeig`, described below, both make use of `RepXspace`.

8.2.2. Diagonalising matrix representations

Let the Hamiltonian $\hat{H}$ be encoded in the Maple variable *HOp* as described in Section 7.3 (*HOp* might have been obtained using the procedure `ACM_Hamiltonian` that is described in Section 6.2.1). The procedure call

$$\texttt{DigXspace}(HOp, a, \lambda_0, \nu_{\min}, \nu_{\max}, v_{\min}, v_{\max}, L_{\min}, L_{\max}) : \tag{126}$$

then represents $\hat{H}$ on the truncated Hilbert space $\mathbb{H}^{\text{trunc}}$, specified as in Section 6.2.2, and diagonalises it. Prior to diagonalisation, the truncated matrix for $\hat{H}$ is made Hermitian by averaging it and its transpose.[13] The diagonalisation is performed separately on each L-space (making use of Maple's diagonalisation procedure `Eigenvectors`), and consequently only produces meaningful results if $\hat{H}$ is an SO(3)-invariant operator. This procedure returns a list of four values:

$$[\mathit{eigenvals}, \mathit{eigenbases}, \mathit{Xparams}, \mathit{Lvals}]. \tag{127}$$

---

[12] When evaluating (49b) in the case of a Hamiltonian $\hat{W}$, the Clebsch–Gordan coefficient $(L_i M_i\ 00 | L_f M_f) = \delta_{L_i L_f} \delta_{M_i M_f}$, and thus diagonalising a matrix whose elements are (125) gives the energy eigenvalues of $\hat{W}$ directly.

[13] This is done because the matrix should be Hermitian but might not have come out as such due to working with a truncated Hilbert space. Thus, its Hermiticity is artificially restored to avoid complex eigenvalues.

Table 6
Predefined functions of v.

Function name	Dependence on v	Comment
`lambda_fix_fun`	0	See (63)
`lambda_sho_fun`	v	See (61)
`lambda_acm_fun`	v mod 2	See (62)

The first of these, *eigenvals*, contains the energy eigenvalues for $\hat{H}$ acting on $\mathbb{H}^{\text{trunc}}$. Specifically, *eigenvals* is a list of lists, with *eigenvals*[k][n] the eigenvalue of the nth lowest energy eigenstate $|n\,L\rangle$ of angular momentum L, where L is obtained from the list *Lvals* through $L = Lvals[k]$. The component *Lvals* of (127) contains, in ascending order, all values of L in the range $L_{\min} \le L \le L_{\max}$ that have non-zero dimension in $\mathbb{H}^{\text{trunc}}$. The eigenvalues in *eigenvals* are most conveniently displayed using the procedure `Show_Eigs` described in Section 7.2.1.

The second element *eigenbases* of the returned list (127) is a list of Matrices, one for each angular momentum value given in *Lvals*, that provides the basis transformation from the original basis (76) to the eigenbasis of $\hat{H}$. Specifically, if $L = Lvals[k]$, then the columns of the Matrix *eigenbases*[k] are the eigenvectors of the angular momentum L subspace of $\mathbb{H}^{\text{trunc}}$.

The third element *Xparams* of the returned list (127) is itself a list which contains the original arguments a, λ_0, $\nu_{\min}$, $\nu_{\max}$, $v_{\min}$, $v_{\max}$ to the procedure. This is useful for passing these parameters to the procedure described next.

8.2.3. Transforming basis for matrix representations

Let the operator $\hat{W}$ be encoded in the Maple variable *WOp* as described in Section 7.3. Then, the procedure call

`AmpXspeig`(*WOp*, *eigenbases*, *Xparams*, *Lvals*) : (128)

represents the operator $\hat{W}$ on the truncated Hilbert space $\mathbb{H}^{\text{trunc}}$ specified by the parameters in *Xparams* and *Lvals*, and transforms this to the basis specified by *eigenbases*. This procedure is designed to use, for its final three arguments, the final three elements of the list (127) returned by the procedure `DigXspace`, and thus these arguments have the form described in Section 8.2.2. The procedure `AmpXspeig` returns a Matrix, each element of which is itself a Matrix. These internal Matrices correspond to pairs $(L_{\text{f}}, L_{\text{i}})$ of angular momenta from the list *Lvals*. Specifically, if *Melements* is the value returned by (128), and k_{i} and k_{f} are such that $L_{\text{i}} = Lvals[k_{\text{i}}]$ and $L_{\text{f}} = Lvals[k_{\text{f}}]$, then $Melements[k_{\text{f}}, k_{\text{i}}][n_{\text{f}}, n_{\text{i}}]$ is the alternative SO(3)-reduced matrix element $\langle n_{\text{f}} L_{\text{f}} \| \hat{W} \| n_{\text{i}}\, L_{\text{i}} \rangle^{\natural}$ of the operator $\hat{W}$ between the n_{i}th and n_{f}th states of angular momenta L_{i} and L_{f} respectively in *eigenbases* (when obtained as part of the return value (127) from `DigXspace`, called as in (126), these states are eigenstates of the operator $\hat{H}$).

Note that to obtain the SO(3)-reduced transition rates (100) for the operator $\hat{W}$, these reduced matrix elements $\langle n_{\text{f}} L_{\text{f}} \| \hat{W} \| n_{\text{i}}\, L_{\text{i}} \rangle^{\natural}$ should be squared and multiplied by $(2L_{\text{f}}+1)/(2L_{\text{i}}+1)$. This is readily done using the procedure `Show_Rats` described in Section 7.2.2. Similarly, transition amplitudes (102) are readily obtained using the procedure `Show_Amps` described in Section 7.2.3. Other values obtained from $\langle n_{\text{f}} L_{\text{f}} \| \hat{W} \| n_{\text{i}}\, L_{\text{i}} \rangle^{\natural}$ can be displayed using `Show_Rats` or `Show_Amps`, as described in Sections 7.4.2 and 7.4.3 respectively.

8.3. Further variants on the basis type

The procedure `ACM_set_basis_type`, described in Section 6.5, allows the user to choose between certain ways as to how λ_v depends on λ_0 and v, for use in the procedures `ACM_Scale`, `ACM_Adapt`, `RepXspace`, `DigXspace` and `AmpXspeig`. The procedure call

`ACM_set_lambda_fun`(*func*$_{\text{btype}}$, *show*) : (129)

enables the dependence on v to be more general. Here *func*$_{\text{btype}}$ should be a previously defined Maple procedure that takes a single integer argument, and returns an integer. The call (129) then stipulates that λ_v is given by $\lambda_0 + func_{\text{btype}}(v)$ (the user should ensure that this sum is always positive). Table 6 lists the three such procedures that are predefined in the ACM code, and which are used to give (63), (61) and (62) respectively.

The procedure

`ACM_show_lambda_fun`($v_{\min}$, $v_{\max}$) : (130)

returns a list of the values of the currently set function *func*$_{\text{btype}}$ for its argument taking the range from $v_{\min}$ to $v_{\max}$.

9. Representing operators on the component Hilbert spaces

In this section, we describe procedures by which calculations may be performed independently on the radial Hilbert space $\mathcal{L}^2(\mathbb{R}_+, d\beta)$ and the spherical Hilbert space $\mathcal{L}^2(S_4, \sin 3\gamma\, d\gamma\, d\Omega)$. Together, they are used by the ACM procedures discussed in the previous sections. However, their generality enables them to be used in many other ways, including the construction of models other than the ACM.

9.1. Matrix elements of radial operators

9.1.1. Products of radial operators

Let

$$\hat{Z} = \hat{Z}_N \cdots \hat{Z}_2\, \hat{Z}_1 \tag{131}$$

be a product of the radial operators that appear in Table 1, and let `RadOpList` be the Maple list assigned by

$$\texttt{RadOpList} := [op_N, \ldots, op_2, op_1]: \tag{132}$$

where each op_k is the symbolic name in Table 1 that corresponds to the operator $\hat{Z}_k$. Then, the procedure call

$$\texttt{RepRadial\_Prod}(\texttt{RadOpList}, a, \lambda, r, \nu_{\min}, \nu_{\max}): \tag{133}$$

returns the Matrix, having matrix elements $F^{(a)}_{\lambda+r,\mu;\lambda\nu}(\hat{Z})$, that represents $\hat{Z}$ acting between the truncated subspaces of $\mathcal{L}^2(\mathbb{R}_+, d\beta)$ spanned by

$$\{\mathcal{R}^{(a,\lambda)}_\nu \mid \nu_{\min} \le \nu \le \nu_{\max}\} \quad \text{and} \quad \{\mathcal{R}^{(a,\lambda+r)}_\mu \mid \nu_{\min} \le \mu \le \nu_{\max}\}. \tag{134}$$

Note that if `RadOpList` is an empty list then, in effect, $\hat{Z} = \hat{1}$ and the returned Matrix expresses one set of basis elements in terms of the other.

This procedure is implemented by replacing the list (131) by, depending on the value of r, an equivalent list of operators $\hat{Z} = \hat{Z}'_M \cdots \hat{Z}'_2 \hat{Z}'_1$, and then forming

$$F^{(a)}_{\lambda+r,\mu;\lambda\nu}(\hat{Z}) = \sum_{\mu_0,\mu_1,\ldots,\mu_M} F^{(a)}_{\lambda+r,\mu;\lambda_M\mu_M}(\hat{1})\, F^{(a)}_{\lambda_M\mu_M;\lambda_{M-1}\mu_{M-1}}(\hat{Z}'_M) \cdots F^{(a)}_{\lambda_1\mu_1;\lambda_0\mu_0}(\hat{Z}'_1) F^{(a)}_{\lambda_0\mu_0;\lambda\nu}(\hat{1}). \tag{135}$$

Here, the λ_i are chosen so that as many of the matrices $F^{(a)}_{\lambda_i\mu_i;\lambda_{i-1}\mu_{i-1}}(\hat{Z}'_i)$ as possible are obtained directly from the analytic expressions given in Section 3. In addition, if $r \le 0$ then $\lambda_0 = \lambda$, and if $r \ge 0$ then $\lambda_M = \lambda + r$. This ensures that at least one of the two matrices representing $\hat{1}$ in (135) is the identity matrix, with a non-identity matrix making use of the appropriate case of (35).

For example, the call `RepRadial_Prod([Radial_b,Radial_Db]`, $a, \lambda, 2, \nu_{\min}, \nu_{\max}$) calculates the matrix of elements $F^{(a)}_{\lambda+2,\mu;\lambda\nu}(\hat{\beta}\, d/d\beta)$ by multiplying together the matrices for $F^{(a)}_{\lambda+2,\mu;\lambda+1,\nu}(\hat{\beta})$ and $F^{(a)}_{\lambda+1,\mu;\lambda\nu}(d/d\beta)$ obtained from (26) and (30) respectively. On the other hand, `RepRadial_Prod([Radial_b, Radial_Db]`, $a, \lambda, 0, \nu_{\min}, \nu_{\max}$) obtains the matrix of elements $F^{(a)}_{\lambda\mu;\lambda\nu}(\hat{\beta}\, d/d\beta)$ directly from (24).

In some cases, depending on the parity of r, this calculation makes use of the non-analytic matrix elements of $\hat{\beta}^{\pm 1}$ or $d/d\beta$ obtained as described in Section 5.4. Consequently, the truncation can adversely affect the calculation, and a Hilbert space of larger dimension should be used.

Note that to obtain the matrix representing an operator $\hat{X}$ acting between the truncated subspaces of $\mathcal{L}^2(\mathbb{R}_+, \beta^4\, d\beta)$ spanned by the states

$$\{|(a, \lambda)\ \nu\rangle \mid \nu_{\min} \le \nu \le \nu_{\max}\} \quad \text{and} \quad \{|(a, \lambda + r)\ \mu\rangle \mid \nu_{\min} \le \mu \le \nu_{\max}\} \tag{136}$$

we should, in accordance with (12), obtain the matrix of elements $F^{(a)}_{\lambda+r,\mu;\lambda\nu}(\hat{Z})$ for $\hat{Z} = \hat{\beta}^2 \hat{X} \hat{\beta}^{-2}$.

If *RadOp* is the symbolic name for one of the radial operators $\hat{Z}$ listed in Table 1, then the procedure call

$$\texttt{ME\_Radial}(RadOp, a, \lambda, r, \mu, \nu): \tag{137}$$

returns the single matrix element $F^{(a)}_{\lambda+r,\mu;\lambda\nu}(\hat{Z})$. In addition, if the first argument is set to the symbolic name `Radial_id`, then the matrix element $F^{(a)}_{\lambda+r,\mu;\lambda\nu}(1)$ is returned. Note that, if possible, the value returned by (137) is obtained by directly using one of the expressions from Section 3. Otherwise, the required matrix element is extracted after using `RepRadial_Prod` to calculate a matrix representing $\hat{Z}$ on a certain truncated space. In this latter case, the truncation may affect the accuracy of the returned value.

9.1.2. Linear combinations of radial operators

Linear combinations of radial operators of the type (131) can also be readily obtained using the ACM code. Consider an operator

$$\hat{Z} = \sum_{k=1}^{N} c_k \hat{Z}_{kM_k} \cdots \hat{Z}_{k2} \hat{Z}_{k1}, \tag{138}$$

where $N \ge 0$, each $M_k \ge 0$, each c_k is a constant, and each $\hat{Z}_{ki}$ is an operator from Table 1. Then let `RadOpLC` be the Maple list assigned by

$$\texttt{RadOpLC} := [[co_1, [op_{1M_1}, \ldots, op_{12}, op_{11}]], [co_2, [op_{2M_2}, \ldots, op_{22}, op_{21}]], \ldots, [co_N, [op_{NM_N}, \ldots, op_{N2}, op_{N1}]]], \tag{139}$$

where op_{ki} is the symbolic name in the table that corresponds to the operator $\hat{Z}_{ki}$, and co_k corresponds to the constant c_k in (138). Then, the procedure call

$$\texttt{RepRadial\_LC}(\texttt{RadOpLC}, a, \lambda, r, \nu_{\min}, \nu_{\max}): \tag{140}$$

returns the Matrix, having elements $F^{(a)}_{\lambda+r,\mu;\lambda\nu}(\hat{Z})$, that represents $\hat{Z}$ acting between the two sets of basis states of $\mathcal{L}^2(\mathbb{R}_+, d\beta)$ given in (134).

9.1.3. Improving accuracy

As explained above, the implementation of `RepRadial_Prod` and `RepRadial_LC` sometimes uses matrix multiplication, and this can result in a loss of accuracy. To ameliorate this, these two procedures take an optional seventh parameter which specifies a larger Hilbert space on which the representations are to be calculated, before being truncated to the final space required. Specifically,

$$\begin{aligned}&\texttt{RepRadial\_Prod}(\texttt{RadOpList}, a, \lambda, r, \nu_{\min}, \nu_{\max}, \nu_{\text{lap}}) :\\ &\texttt{RepRadial\_LC}(\texttt{RadOpLC}, a, \lambda, r, \nu_{\min}, \nu_{\max}, \nu_{\text{lap}}) :\end{aligned} \tag{141}$$

each first calculate the matrix representing the operator, `RadOpList` or `RadOpLC` respectively, between the truncated subspaces of $\mathcal{L}^2(\mathbb{R}_+, d\beta)$ spanned by

$$\{\mathcal{R}_\nu^{(a,\lambda)} \mid \nu'_{\min} \le \nu \le \nu'_{\max}\} \quad \text{and} \quad \{\mathcal{R}_\mu^{(a,\lambda+r)} \mid \nu'_{\min} \le \mu \le \nu'_{\max}\}, \tag{142}$$

where $\nu'_{\max} = \nu_{\max} + \nu_{\text{lap}}$ and $\nu'_{\min} = \max\{0, \nu_{\min} - \nu_{\text{lap}}\}$. The resulting matrix is then truncated to provide a representation between the subspaces of $\mathcal{L}^2(\mathbb{R}_+, d\beta)$ spanned by (134).

9.2. SO(5) ⊃ SO(3) *Clebsch–Gordan coefficients and matrix elements of SO(5) spherical harmonics*

The SO(5) ⊃ SO(3) Clebsch–Gordan coefficients (40) are obtained using the procedure call

$$\texttt{CG\_SO5r3}(v_{\text{i}}, \alpha_{\text{i}}, L_{\text{i}}, v, \alpha, L, v_{\text{f}}, \alpha_{\text{f}}, L_{\text{f}}) : \tag{143}$$

These values are obtained from the datafiles (for the sake of efficiency, accessing one value from a particular file loads all the values from that file into memory, from where they are subsequently obtained).

The SO(5)-reduced matrix elements $\langle v_{\text{f}} \| \hat{\mathcal{Y}}^v \| v_{\text{i}} \rangle$, as specified in (45), are obtained by multiplying by $(4\pi)^{-1}$ the value obtained from the procedure call

$$\texttt{ME\_SO5red}(v_{\text{f}}, v, v_{\text{i}}) : \tag{144}$$

The procedure call

$$\texttt{ME\_SO5r3}(v_{\text{f}}, \alpha_{\text{f}}, L_{\text{f}}, v, \alpha, L, v_{\text{i}}, \alpha_{\text{i}}, L_{\text{i}}) : \tag{145}$$

returns the SO(3)-reduced matrix element

$$4\pi \frac{\langle v_{\text{f}} \alpha_{\text{f}} L_{\text{f}} \| \hat{\mathcal{Y}}^v_{\alpha L} \| v_{\text{i}} \alpha_{\text{i}} L_{\text{i}} \rangle}{\sqrt{2L_{\text{f}} + 1}}. \tag{146}$$

For example, because $\mathcal{Y}^3_{100} = (3/4\pi)\cos 3\gamma$ (see (43)), the call $\texttt{ME\_SO5r3}(v_{\text{f}}, \alpha_{\text{f}}, L_{\text{f}}, 3, 1, 0, v_{\text{i}}, \alpha_{\text{i}}, L_{\text{i}})$ returns the value $3\langle v_{\text{f}} \alpha_{\text{f}} L_{\text{f}} \| \cos 3\gamma \| v_{\text{i}} \alpha_{\text{i}} L_{\text{i}} \rangle / \sqrt{2L_{\text{f}} + 1}$.

Let the operator $\hat{Y}$ be the product

$$\hat{Y} = \hat{Y}_N \cdots \hat{Y}_2 \, \hat{Y}_1, \tag{147}$$

where each Y_k is an SO(5) spherical harmonic, and at most one has non-zero SO(3) angular momentum. Then let `SpOpList` be the Maple list assigned by

$$\texttt{SpOpList} := [op_N, \ldots, op_2, op_1] : \tag{148}$$

where each op_k is either the list $[v, \alpha, L]$ that labels (the reduced) Y_k, or alternatively, is the corresponding symbolic name in Table 2. The procedure call

$$\texttt{RepSO5r3\_Prod}(\texttt{SpOpList}, v_{\min}, v_{\max}, L_{\min}, L_{\max}) : \tag{149}$$

then returns the Matrix whose elements are the SO(3)-reduced matrix elements

$$(4\pi)^N \frac{\langle v_{\text{f}} \alpha_{\text{f}} L_{\text{f}} \| \hat{Y} \| v_{\text{i}} \alpha_{\text{i}} L_{\text{i}} \rangle}{\sqrt{2L_{\text{f}} + 1}} \tag{150}$$

of the operator $\hat{Y}$ on the truncated subspace of $\mathcal{L}^2(S_4, \sin 3\gamma \, d\gamma \, d\Omega)$ spanned by

$$\{|v\alpha LM\rangle \mid 1 \le \alpha \le d_{vL}, v_{\min} \le v \le v_{\max}, L_{\min} \le L \le L_{\max}\}. \tag{151}$$

For example, the procedure call $\texttt{RepSO5r3\_Prod}([[3, 1, 0], [3, 1, 0]], v_{\min}, v_{\max}, L_{\min}, L_{\max})$ returns a Matrix whose entries are the alternative SO(3)-reduced matrix elements $9\langle v_{\text{f}} \alpha_{\text{f}} L_{\text{f}} \| \cos^2 3\gamma \| v_{\text{i}} \alpha_{\text{i}} L_{\text{i}} \rangle / (2L_{\text{f}} + 1)$. Alternatively, because, as indicated in Table 2, the SO(3)-reduced SO(5) spherical harmonic $\mathcal{Y}^3_{10}$ is encoded using `SpHarm_310`, we obtain the same Matrix using `RepSO5r3_Prod` ([`SpHarm_310`, `SpHarm_310`], $v_{\min}, v_{\max}, L_{\min}, L_{\max}$).

Note that, because alternative SO(3)-reduced matrix elements for the operator $\hat{Y}$ are obtained from those of the operators $\hat{Y}_k$ using matrix multiplication, meaningful results are obtained from (149) only where at most one of factors $\hat{Y}_k$ of $\hat{Y}$ has non-zero angular momentum. The reason for this is the same as that given in the first two sentences of Section 7.3.3.

9.3. SO(3) Clebsch–Gordan coefficients

Finally, we provide a procedure which gives the usual SO(3) Clebsch–Gordan coefficients. The procedure call

$$\texttt{CG\_SO3}(L_\mathrm{i}, M_\mathrm{i}, L, M, L_\mathrm{f}, M_\mathrm{f}) : \tag{152}$$

returns the SO(3) Clebsch–Gordan coefficient $(L_\mathrm{i}M_\mathrm{i}\ LM|L_\mathrm{f}M_\mathrm{f})$. Here, each argument is a half integer. These Clebsch–Gordan coefficients are calculated using [23, eqn. (3.6.10)].

This procedure is only used in the ACM code to calculate the transition rates (82) and (102) through the procedure `quad_amp_fun` given in Table 5. It can, of course, be used to define other such procedures.

10. Discussion

The ACM code makes use of the SU(1, 1)×SO(5) dynamical group structure of the Bohr model to define a basis on which all operators of interest may be expressed algebraically. This relies, in particular, on the algebraic expressions of some operators between different modified oscillator representations of SU(1, 1). It also relies on the recently available SO(5) ⊃ SO(3) Clebsch–Gordan coefficients and SO(5)-reduced matrix elements of SO(5) spherical harmonics. A wide range of SO(3) and time-reversal invariant Hamiltonians may be defined, and these may be numerically diagonalised on a user-defined finite-dimensional subspace of the Bohr model. Their eigenvalues are output, as well as the quadrupole moments, transition rates and amplitudes between eigenstates of interest to the user. The efficiency of the calculation may be fine-tuned by altering the particular modified oscillator representations being used and the unit a in which the nuclear deformation parameter β is defined. Various parameter settings enable the user to adjust and extend the default implementation.

The code is designed to be flexible, enabling calculations to be implemented in a range of situations, and to be extendable. For example, calculations are easily done in the rigid-β limit, simply by freezing the β degree of freedom, and in the soft-β O(5)-symmetric Wilets–Jean limit [27]. Such calculations were carried out [4] with an early version of the code as well as calculations which explored the approach to the Meyer-ter-Vehn [28] and adiabatic Bohr–Mottelson [29] limits. A significant result of these calculations was that, when the β potential was soft enough to give relatively low β-vibrational states in the model, the centrifugal stretching perturbations to the rotational spectra proved to be stronger than is commonly observed experimentally. This suggests either that such low-energy $L = 0$ excited bands should not be interpreted as β-vibrational bands or that the Bohr model is missing important ingredients. Thus, the ACM code provides powerful tools for investigating the consistency of model interpretations of nuclear data based on the collective model.

We emphasise that we are well aware of the limitations of the Bohr model even when applied with Hamiltonians having freely adjustable parameters. However, we also subscribe to the view that the importance of a model is to reveal its deficiencies as well as its successes so that more realistic models can be constructed.

A major limitation of the Bohr model is that it is fundamentally a liquid drop model. Moreover, the O(5)-invariance of its standard kinetic energy (related to the SO(5) Casimir operator) imposes irrotational-flow relationships between the components of its moments of inertia [8]. To escape from this irrotational-flow limitation, it is necessary to add vorticity degrees of freedom to the model. Thus, it is important to understand the extent to which experimental data favours the addition of such vorticity degrees of freedom to the Bohr model. On the theoretical side, it is also important to understand the effects of adding such degrees of freedom to the Bohr model. For such reasons, the ACM code has been designed to admit extensions as deemed to be desirable. The primary characteristic of the code is that it is based on an SU(1, 1) × SO(5) dynamical group. Thus, we anticipate that it will remain relevant for a model with a larger dynamical group that contains this group as a subgroup.

In this context, it may be noted that the Interacting Boson Model (IBM) [30–32], in its U(6) ⊃ U(5) ⊃ O(5) and U(6) ⊃ O(6) ⊃ O(5) dynamical symmetry limits, is able to make good use of the SU(1, 1) × SO(5) dynamical group [33]. Moreover, as shown in [34–37], the IBM contracts to the Bohr model in these limits and thus IBM calculations can be executed in these limits by simple extensions of the ACM code to include s-boson degrees of freedom. Such calculations have recently been pursued by Thiamova et al. [22].

Acknowledgement

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.

Appendix A. Rigid-β models

A rigid-β model is a sub-model of the Bohr model in which the β-vibrational degree of freedom is suppressed and, in effect, β takes a fixed value β_0. A rigid-β model is also obtained as a limit of a sequence of Bohr models for which the potential increasingly deepens at $\beta = \beta_0$.

Rigid-β calculations are readily carried out using the procedures `ACM_Scale` and `ACM_Adapt`, described in Section 6.2, after a few minor considerations. The first is that the Hamiltonian $\hat{H}$ being analysed should not contain any reference to β (other than its rigid $\beta = \beta_0$ value; if it does, the results will usually be meaningless). Thus it will not be possible to use `ACM_Hamiltonian` to obtain the Maple encoding of $\hat{H}$ in most cases. Instead, the ACM code supplies a procedure which is more convenient for rigid-β Hamiltonians of the form

$$x\hat{\Lambda}^2 + x_0 + x_1\cos 3\gamma + x_2\cos^2 3\gamma + x_3\cos^3 3\gamma + x_4\cos^4 3\gamma + x_5\cos^5 3\gamma + x_6\cos^6 3\gamma. \tag{A.1}$$

For such a Hamiltonian, the Maple encoding is generated by the procedure call

$$\texttt{ACM\_HamRigidBeta}(x, x_0, x_1, x_2, x_3, x_4, x_5, x_6, f) : \tag{A.2}$$

where f is either 0 (the default) or 1, the former indicating that the encoding should use only the spherical harmonic $\mathcal{Y}^3_{100}$, and the latter indicating that $\mathcal{Y}^6_{100}$ should be used as much as possible.[14]

As when using `ACM_Hamiltonian`, the value returned by this procedure should be assigned to a Maple variable, such as in

```
HOp := ACM_HamRigidBeta(1,0,11.75,0.8):
```
(A.3)

This variable is then used as the first argument to the procedures `ACM_Scale` or `ACM_Adapt`. Note that arguments to `ACM_HamRigidBeta` that are not specified are taken to be 0.

The second consideration is that in selecting the basis (76) of the truncated Hilbert space $\mathbb{H}^{\text{trunc}}$, the range of radial states should be suppressed. This is done by setting $\nu_{\min} = \nu_{\max}$ (=0, for example). With the Hamiltonian not referencing β, the values of a and λ_0 become irrelevant. Consequently, for the procedures `ACM_Scale` and `ACM_Adapt`, any values can be used for the second and third arguments.

The third and final consideration is that, in order to properly calculate transition rates, the quadrupole operator (107) should be replaced by one in which the β variable is suppressed. To readily accommodate this, the ACM code contains the assignment

```
quadRigid_op := [[Convert_112, [SpHarm_112]]] :
```
(A.4)

To ensure that transition rates are calculated using this operator, it is necessary, as described in Section 7.4.1, to invoke

```
ACM_set_transition(quadRigid_op):
```
(A.5)

before using (A.3). Note that the transition rates calculated using (A.4) should be multiplied by β_0 to make them compatible with those calculated using (107). If required, this could be accomplished by defining an operator similar to (A.4) with the value of β_0 included, and then invoking `ACM_set_transition` for this operator.

Appendix B. Hilbert space bases parameters

To determine a basis (8) of the Hilbert space $\mathbb{H}$ completely, it is necessary to specify values of the parameters a and λ_v for $v \geq 0$. In the ACM, λ_v is chosen to depend on λ_0 in a certain convenient way, whereupon it is necessary to specify the values of a and λ_0. These values can have a severe effect on the efficiency of a calculation performed using the basis (8). Here, for Hamiltonians of the form

$$\hat{H}_{\text{gen}} = -\frac{1}{2B}\nabla^2 + V_{\text{gen}}(\beta, \gamma), \tag{B.1}$$

we discuss means of obtaining near-optimal values of a and λ_0 based on the energy minimisation variational principle.[15]

B.1. Two parameter minimisation

Define $\Psi^{(a,\lambda_v)}_{v;v\alpha LM}(\beta, \gamma, \Omega) = \beta^{-2}\mathcal{R}^{(a,\lambda_v)}_v(\beta)\mathcal{Y}^v_{\alpha LM}(\gamma, \Omega)$, so that $\Psi^{(a,\lambda_v)}_{v;v\alpha LM}$ is the wave function corresponding to the state $|(a, \lambda_v)\ v; v\alpha LM\rangle$. Then, on using (57) and (58), we obtain

$$\hat{H}_{\text{gen}}\Psi^{(a,\lambda_v)}_{v;v\alpha LM} = \frac{1}{\beta^2}\mathcal{Y}^v_{\alpha LM}\left[-\frac{1}{2B}\frac{d^2}{d\beta^2} + \frac{(v+3/2)^2 - 1/4}{2B\beta^2} + V_{\text{gen}}\right]\mathcal{R}^{(a,\lambda_v)}_v. \tag{B.2}$$

The expectation value of $\hat{H}_{\text{gen}}$ on the basis state for which $\nu = v = 0$ is then given by (using the volume element (2))

$$\begin{aligned}\langle(a, \lambda_0)\ 0; 0100|\hat{H}_{\text{gen}}|(a, \lambda_0)\ 0; 0100\rangle &= \int_{\text{SO(3)}} d\Omega \int_0^\infty \beta^4\, d\beta \int_0^{\pi/3} \sin 3\gamma\, d\gamma\ \Psi^{(a,\lambda_0)*}_{0;0100}\,\hat{H}_{\text{gen}}\,\Psi^{(a,\lambda_0)}_{0;0100} \\ &= \frac{3}{2}\int_0^\infty d\beta \int_0^{\pi/3} \sin 3\gamma\, d\gamma\ \mathcal{R}^{(a,\lambda_0)}_0\left[-\frac{1}{2B}\frac{d^2}{d\beta^2} + \frac{1}{B\beta^2} + V_{\text{gen}}\right]\mathcal{R}^{(a,\lambda_0)}_0,\end{aligned} \tag{B.3}$$

having used the first case of (43). After defining the function $\tilde{V}_{\text{gen}}(\beta)$ of β by

$$\tilde{V}_{\text{gen}}(\beta) = \frac{3}{2}\int_0^{\pi/3} V_{\text{gen}}(\beta, \gamma)\ \sin 3\gamma\, d\gamma, \tag{B.4}$$

and using the definition (13), we then obtain

$$\begin{aligned}\langle(a, \lambda_0)\ 0; 0100|\hat{H}_{\text{gen}}|(a, \lambda_0)\ 0; 0100\rangle &= F^{(a)}_{\lambda_0,0;\lambda_0,0}\left(-\frac{1}{2B}\frac{d^2}{d\beta^2} + \frac{1}{B\beta^2} + \tilde{V}_{\text{gen}}\right) \\ &= \frac{a^2}{2B}\left[1 + \frac{9}{4(\lambda_0 - 1)}\right] + F^{(a)}_{\lambda_0,0;\lambda_0,0}(\tilde{V}_{\text{gen}}),\end{aligned} \tag{B.5}$$

after also making use of (22) and (23). Note that if $V_{\text{gen}}(\beta, \gamma)$ is independent of γ then (B.4) yields $\tilde{V}_{\text{gen}}(\beta) = V_{\text{gen}}(\beta, \gamma)$. Also note that if $\tilde{V}_{\text{gen}}(\beta)$ is a polynomial in $\beta^{\pm 2}$, then the term $F^{(a)}_{\lambda_0,0;\lambda_0,0}(\tilde{V}_{\text{gen}})$ in (B.5) is readily evaluated using the expressions of Section 3.

In accordance with the variational principle, optimal values of a and λ_0 are now obtained by determining where (B.5) is a minimum with respect to these two parameters.

---

[14] Expressing the Hamiltonian in terms of $\mathcal{Y}^6_{100}$ is preferable because, when forming the representation matrices on the truncated Hilbert space, the fewer matrix multiplications required gives greater accuracy. However, in the supplied data files, the SO(5) ⊃ SO(3) Clebsch–Gordan coefficients for $\mathcal{Y}^6_{100}$ are not available to as high seniority as those for $\mathcal{Y}^3_{100}$, and thus for some calculations it might be necessary to restrict usage to the latter.

[15] This variational principle is described in most elementary books on Quantum Mechanics.

B.2. Parameter choices based on physical considerations

In this section, we use physical interpretations of the parameters of a and λ_0, which define the basis (8), to simplify the process of approximating optimal values of these parameters.

Use of (16) and (19) (or [25, (22.6.18)] with $\alpha \to \lambda_v - 1$, $n \to v$ and $x \to a\beta$) shows that the basis wave functions $\mathcal{R}_v^{(a,\lambda_v)}(\beta)$ of $\mathcal{L}^2(\mathbb{R}_+, d\beta)$, defined in (10), satisfy

$$\frac{1}{2a^2}\left[-\frac{d^2}{d\beta^2} + \frac{(\lambda_v-1)^2 - 1/4}{\beta^2} + a^4\beta^2\right]\mathcal{R}_v^{(a,\lambda_v)} = (\lambda_v + 2v)\,\mathcal{R}_v^{(a,\lambda_v)}. \tag{B.6}$$

Consequently, if we set

$$\lambda_v = 1 + \sqrt{(v+3/2)^2 + (a\beta_0)^4}, \tag{B.7}$$

then the Hamiltonian

$$\hat{H}_{\text{rad}}^{(v)} = -\frac{d^2}{d\beta^2} + \frac{(v+3/2)^2 - 1/4}{\beta^2} + V_{\text{rad}}, \tag{B.8}$$

with $V_{\text{rad}}(\beta) = V_{\text{DM}}(\beta)$ given by

$$V_{\text{DM}}(\beta) = a^4\left(\beta^2 + \frac{\beta_0^4}{\beta^2}\right), \tag{B.9}$$

has eigenfunctions $\mathcal{R}_v^{(a,\lambda_v)}(\beta)$ for $v = 0, 1, 2, 3, \ldots$. Note that this holds for all $a > 0$. Also note that the potential $V_{\text{DM}}(\beta)$ has a minimum at $\beta = \beta_0$.

The form of (B.6) shows that, in effect, the parameter a defines the scale of the system determined by $\hat{H}_{\text{rad}}^{(v)}$. This is also reflected in the fact that, apart from an overall normalisation, each eigenfunction $\mathcal{R}_v^{(a,\lambda_v)}(\beta)$, defined by (10), is a function of a and β through the combination $a\beta$ alone. The parameter a is therefore a measure of the *width* of $\mathcal{R}_v^{(a,\lambda_v)}(\beta)$ and, correspondingly, is also related to the width of the potential $V_{\text{DM}}(\beta)$ defined by (B.9). For this form of potential, the parameter λ_v and the scaled value $a\beta_0$ of the potential's minimum determine one another through (B.7). The dependence of $\mathcal{R}_v^{(a,\lambda_v)}(\beta)$ on β_0 is illustrated in Fig. 1 of [4].

For a Hamiltonian $\hat{H}_{\text{rad}}^{(v)}$ of the form (B.8) with an arbitrary potential $V_{\text{rad}}(\beta)$ that has a minimum at $\beta = \beta_0$, we may exploit these properties to obtain estimates of the optimal values of a and λ_v. A robust means of doing this is to use the value of β_0 to define λ_v as the function of a given by (B.7), and then to use the variational method to optimise the value of a. Thus, with the dependence (B.7) imposed, we seek the value of the single parameter a at which

$$\begin{aligned}\int_0^\infty \mathcal{R}_0^{(a,\lambda_v)}\,\hat{H}_{\text{rad}}^{(v)}\,\mathcal{R}_0^{(a,\lambda_v)}\,d\beta &= F^{(a)}_{\lambda_v,0;\lambda_v,0}\left(-\frac{d^2}{d\beta^2} + \frac{(v+3/2)^2-1/4}{\beta^2} + V_{\text{rad}}\right)\\ &= a^2\left[1 + \frac{(v+3/2)^2}{\lambda_v - 1}\right] + F^{(a)}_{\lambda_v,0;\lambda_v,0}(V_{\text{rad}})\end{aligned} \tag{B.10}$$

is a minimum.

For the Hamiltonian $\hat{H}_{\text{gen}}$ of (B.1) acting on $\mathbb{H}$, in the case that $V_{\text{gen}}(\beta,\gamma)$ is independent of γ, comparison of (B.2) with (B.8) shows that in order to apply this method, we should use $V_{\text{rad}}(\beta) = 2B\,\tilde{V}_{\text{gen}}(\beta)$ in (B.10). In fact, in the $v = 0$ case, the same substitution also applies when $V_{\text{gen}}(\beta,\gamma)$ is dependent on γ, because according to (B.5), the optimal values depend only on $\tilde{V}_{\text{gen}}(\beta)$. In this case, we then obtain values of a and λ_0 such that $\Psi^{(a,\lambda_0)}_{0;0100} = \beta^{-2}\mathcal{R}_0^{(a,\lambda_0)}\mathcal{Y}^0_{100}$ is a good approximation to the ground state wave function of $\hat{H}_{\text{gen}}$ by treating λ_0 as dependent on a via

$$\lambda_0(a) = 1 + \sqrt{9/4 + (a\beta_0)^4}, \tag{B.11}$$

and then minimising (B.5) with respect to a.

The above optimisation strategy is used in [4] for Hamiltonians of the form

$$\hat{H}_{\text{RWC}}(B, c_1, c_2, \chi, \kappa) = -\frac{1}{2B}\nabla^2 + \frac{B}{2}(c_1\beta^2 + c_2\beta^4) - \chi\beta\cos 3\gamma + \kappa\cos^2 3\gamma \tag{B.12}$$

for $c_2 \geq 0$, with $c_1 > 0$ if $c_2 = 0$. Here, the potential

$$V_{\text{RWC}}(\beta,\gamma) = \frac{B}{2}(c_1\beta^2 + c_2\beta^4) - \chi\beta\cos 3\gamma + \kappa\cos^2 3\gamma, \tag{B.13}$$

leads to

$$\tilde{V}_{\text{RWC}}(\beta) = \frac{B}{2}(c_1\beta^2 + c_2\beta^4) + \frac{\kappa}{3}, \tag{B.14}$$

via (B.4). Note that the potential $\tilde{V}_{\text{RWC}}(\beta)$ has a minimum at $\beta = \beta_0$, where

$$\beta_0 = \begin{cases}\sqrt{-\dfrac{c_1}{2c_2}} & \text{if } c_1 < 0,\\ 0 & \text{if } c_1 \geq 0.\end{cases} \tag{B.15}$$

Table B.7
Eigenvalues obtained using basis type (62) and optimised parameters.

$\nu_{\max}$	$L=0$				$L=2$			
0	0.00	76.27	149.76	271.21	8.03	45.62	90.54	121.56
1	0.00	72.60	152.22	268.61	3.64	45.40	90.43	116.73
2	0.00	74.44	150.91	270.40	6.01	45.43	90.36	119.28
≥5	0.00	74.43	150.93	270.42	6.00	45.44	90.37	119.27

Table B.8
Eigenvalues obtained using the standard harmonic oscillator basis.

$\nu_{\max}$	$L=0$				$L=2$			
20	0.00	83.48	362.11	901.48	3.98	26.53	147.21	240.59
25	0.00	96.57	118.84	280.33	0.67	7.66	64.82	75.55
30	0.00	76.76	154.48	273.94	8.12	48.37	93.77	122.85
32	0.00	74.28	150.71	270.20	5.86	45.29	90.15	119.05
33	0.00	74.43	150.94	270.42	5.97	45.41	90.37	119.28

In this case where $\tilde{V}_{\text{gen}} = \tilde{V}_{\text{RWC}}$, given by (B.14), use of (B.5) and (21) yields the expectation value

$$\langle (a,\lambda_0)\,0;\,0100|\hat{H}_{\text{RWC}}(B,c_1,c_2,\chi,\kappa)|(a,\lambda_0)\,0;\,0100\rangle = \frac{a^2}{2B}\left[1+\frac{9}{4(\lambda_0-1)}\right]+\frac{B}{2a^2}c_1\lambda_0+\frac{B}{2a^4}c_2\lambda_0(\lambda_0+1)+\frac{\kappa}{3}. \tag{B.16}$$

Expressions (B.11) and (B.15) are then used to define λ_0 as a function of a (and c_1 and c_2), and their optimal values are obtained from where the expectation value (B.16) is minimal with respect to a.[16] The results given in [4] are obtained, after using this method, with λ_v determined from λ_0 using (62).

B.3. Further fine tuning

Having determined optimal values of a and λ_0, the above analysis also suggests that for general seniority v, a more optimal value of λ_v is given by (B.7). However, efficient use of the ACM requires that the values of λ_v should satisfy (54). To achieve this we can thus approximate the optimal value of λ_v by using

$$\lambda_v = \lambda_0 + \left[\sqrt{(v+3/2)^2+\beta_*^4}-\sqrt{9/4+\beta_*^4}\right]_v \tag{B.17}$$

for $v>0$, where $\beta_* = a\beta_0$ and by $[x]_v$, we mean the nearest integer to x with the same parity of v (in the event that $x-v$ is an odd integer, we set $[x]_v = x+1$ for definiteness). That λ_v defined in this way satisfies (54) is guaranteed because $0 < d\lambda_v^{\text{DM}}/dv < 1$.

Note that bases obtained using (B.17) interpolate between those for which $\lambda_v = \lambda_0 + v$ and those for which $\lambda_v = \lambda_0 + (v \bmod 2)$.

B.4. Example computation

Here, we illustrate the efficiencies that can be achieved by optimising the basis parameters. We do this with the Hamiltonian

$$\hat{H}_{\text{EX}} = -\frac{1}{100}\nabla^2 - 100\beta^2 + 25\beta^4 - \frac{1}{10}\beta\cos 3\gamma. \tag{B.18}$$

This is of the form $\hat{H}_{\text{RWC}}$ in (B.12) with $B=50$, $c_1=-4.0$, $c_2=1.0$, $\chi=0.1$ and $\kappa=0.0$. According to (B.15), the spherically averaged potential (B.14) has a minimum at $\beta_0=\sqrt{2}$. Then, with λ_0 the function of a given by (B.11), the expectation value (B.16) is minimal when $a=8.47$ (this is computed using the procedure `RWC_alam` described in Section 6.6). Correspondingly, $\lambda_0 = 144.42$.

After encoding the Hamiltonian $\hat{H}_{\text{EX}}$ using `RWC_Ham`, we compute its eigenvalues using the procedure `ACM_Scale` with arguments $v_{\min}=0$, $v_{\max}=15$, $L_{\min}=0$, $L_{\max}=2$, along with $\nu_{\min}=0$ and various values of $\nu_{\max}$. When the basis type (62) (the default) is used with the optimal values of a and λ_0 given above, the computations produce the lowest four eigenvalues for $L=0$ and $L=2$ listed in Table B.7. Note that the values are given relative to that of the 0(1) state and have been scaled so that the relative value of the converged 2(1) state takes the value 6.0. This has been achieved by first using `ACM_Adapt` for a succession of values $\nu_{\max}$ until convergence is apparent. As explained in Section 6.2.4, use of `ACM_Adapt` sets scaling factors that are applied to the output of subsequent uses of `ACM_Scale`. In this example, we see that using $\nu_{\max}=2$ achieves values better than within 1% of the converged values.

We contrast this with the values obtained using the standard harmonic oscillator basis (61) with $\lambda_0 = 2.5$, and the corresponding scaling $a=\sqrt{B}$. The basis type (61) is enforced by invoking `ACM_set_basis_type(1)`, as described in Section 6.5. Use of `ACM_Scale` for $\nu_{\max}=20, 25, 30, 32, 33$ then yields the eigenvalues listed in Table B.8 (the scaling factors have been retained from the calculations above). We see that here we require $\nu_{\max}=33$ in order that the displayed values are within 1% of the converged values.

For this example, we see that the basis type (62) with optimal parameters converges using far fewer basis states (by a factor of at least 11) than in using the standard harmonic oscillator basis, and is thus significantly quicker. Of course, the contrast in efficiency between the two bases varies significantly between examples. For some calculations, such as for Hamiltonians of the form (B.12) with $c_1 \geq 0$ so that $\beta_0 = 0$, the standard harmonic oscillator basis performs as well as the basis type (62). A systematic study of the efficiencies obtained in computing the ground state eigenvalue for a certain range of model Hamiltonians is given in [2].

[16] The equations (81) and (76) of [4], corresponding to (B.15) and (B.16) here, each contain typographical errors. The account in Section 4.7 of [8] deals (correctly) with the special case in which $c_1 = 1-2\alpha$ and $c_2=\alpha$, although little explicit about the optimal value of a is given. The expressions given here should replace those in [4].

Appendix C. Identity radial matrix elements

The generalised Laguerre polynomials $\{\mathrm{L}_\mu^{(\alpha)}(x),\ \mu = 0, 1, 2, 3, \ldots\}$ satisfy the recurrences [25]:

$$x\,\mathrm{L}_\mu^{(\alpha)}(x) = (\alpha+\mu)\mathrm{L}_\mu^{(\alpha-1)}(x) - (\mu+1)\mathrm{L}_{\mu+1}^{(\alpha-1)}(x), \tag{C.1}$$

$$\mathrm{L}_\mu^{(\alpha)}(x) = \mathrm{L}_\mu^{(\alpha-1)}(x) + \mathrm{L}_{\mu-1}^{(\alpha)}(x), \tag{C.2}$$

with $\mathrm{L}_{-1}^{(\alpha)}(x) = 0$. For a non-negative integer r, the first of these leads (by induction) to

$$x^r\mathrm{L}_\mu^{(\alpha)}(x) = \sum_{j=0}^{r}(-1)^j\binom{r}{j}\frac{\Gamma(\alpha+\mu+1)\,\Gamma(\mu+j+1)}{\Gamma(\alpha+\mu+1-r+j)\,\Gamma(\mu+1)}\mathrm{L}_{\mu+j}^{(\alpha-r)}(x). \tag{C.3}$$

The second (C.2) implies that

$$\mathrm{L}_\mu^{(\alpha)}(x) = \sum_{\xi=0}^{\mu}\mathrm{L}_\xi^{(\alpha-1)}(x). \tag{C.4}$$

Recursively applying this r times leads (by induction) to

$$\mathrm{L}_\mu^{(\alpha)}(x) = \sum_{\xi=0}^{\mu}\binom{\mu-\xi+r-1}{\mu-\xi}\mathrm{L}_\xi^{(\alpha-r)}(x). \tag{C.5}$$

(The case $r = 0$ requires us to use an extended version of the binomial coefficient for which $\binom{n-1}{n} = \delta_{n,0}$.)

Together, (C.1) and (C.5) give:

$$x^r\mathrm{L}_\mu^{(\alpha)}(x) = \sum_{j=0}^{r}(-1)^j\binom{r}{j}\frac{\Gamma(\alpha+\mu+1)\,\Gamma(\mu+j+1)}{\Gamma(\alpha+\mu+1-r+j)\,\Gamma(\mu+1)}\sum_{\xi=0}^{\mu+j}\binom{\mu-\xi+j+r-1}{\mu-\xi+j}\mathrm{L}_\xi^{(\alpha-2r)}(x). \tag{C.6}$$

Thus, for $\alpha = \lambda + 2r - 1$, we obtain

$$x^r\mathrm{L}_\mu^{(\lambda+2r-1)}(x) = \sum_{\xi=0}^{\infty}c_{\mu,\xi}^{(2r)}\,\mathrm{L}_\xi^{(\lambda-1)}(x), \tag{C.7}$$

where we define

$$c_{\mu,\xi}^{(2r)} = \sum_{j=\max\{0,\xi-\mu\}}^{r}(-1)^j\binom{r}{j}\frac{\Gamma(\lambda+2r+\mu)\,\Gamma(\mu+j+1)}{\Gamma(\lambda+r+\mu+j)\,\Gamma(\mu+1)}\binom{\mu-\xi+j+r-1}{\mu-\xi+j}. \tag{C.8}$$

Note, in particular, that $c_{\mu,\xi}^{(2r)} = 0$ if $\xi > \mu + r$.

By virtue of the definition (10), we obtain

$$\begin{aligned}
&\int_0^\infty \mathcal{R}_\mu^{(a,\lambda+2r)}(\beta)\,\mathcal{R}_\nu^{(a,\lambda)}(\beta)\,d\beta \\
&= (-1)^{\nu-\mu}2a\sqrt{\frac{\mu!\,\nu!}{\Gamma(\lambda+\mu+2r)\,\Gamma(\lambda+\nu)}}\int_0^\infty (a\beta)^{2\lambda+2r-1}e^{-a^2\beta^2}\mathrm{L}_\mu^{(\lambda+2r-1)}(a^2\beta^2)\mathrm{L}_\nu^{(\lambda-1)}(a^2\beta^2)\,d\beta \\
&= (-1)^{\nu-\mu}\sqrt{\frac{\mu!\,\nu!}{\Gamma(\lambda+\mu+2r)\,\Gamma(\lambda+\nu)}}\int_0^\infty x^{\lambda+r-1}e^{-x}\mathrm{L}_\mu^{(\lambda+2r-1)}(x)\mathrm{L}_\nu^{(\lambda-1)}(x)\,dx,
\end{aligned} \tag{C.9}$$

where the last line follows on making the substitution $x = a^2\beta^2$, so that $dx = 2a^2\beta\,d\beta$. Then, on using (C.7) and applying the Laguerre orthogonality relationship [25]

$$\int_0^\infty x^{\lambda-1}e^{-x}\mathrm{L}_\xi^{(\lambda-1)}(x)\mathrm{L}_\nu^{(\lambda-1)}(x)\,dx = \delta_{\xi,\nu}\frac{\Gamma(\lambda+\nu)}{\nu!}, \tag{C.10}$$

we obtain:

$$\begin{aligned}
&\int_0^\infty \mathcal{R}_\mu^{(a,\lambda+2r)}(\beta)\,\mathcal{R}_\nu^{(a,\lambda)}(\beta)\,d\beta \\
&= (-1)^{\nu-\mu}\sqrt{\frac{\mu!\,\nu!}{\Gamma(\lambda+\mu+2r)\,\Gamma(\lambda+\nu)}}\int_0^\infty x^{\lambda-1}e^{-x}\sum_{\xi=0}^{\infty}c_{\mu,\xi}^{(2r)}\mathrm{L}_\xi^{(\lambda-1)}(x)\mathrm{L}_\nu^{(\lambda-1)}(x)\,dx \\
&= (-1)^{\nu-\mu}\sqrt{\frac{\mu!\,\Gamma(\lambda+\nu)}{\nu!\,\Gamma(\lambda+\mu+2r)}}c_{\mu,\nu}^{(2r)}.
\end{aligned} \tag{C.11}$$

In fact, similar means could be used to obtain *precise* analytic values of the matrix elements $F_{\lambda+r,\mu;\lambda,\nu}^{(a)}(\hat{Z})$ of all the radial operators considered in Section 3, for r of the same parity as $\hat{Z}$. This would avoid truncation errors that currently arise through using matrix multiplication. Moreover, this would provide, in particular, alternative proofs of the expressions given in Section 3.

Appendix D. Reduced matrix elements of tensors coupled from $\hat{q}$ and $\hat{\pi}$

From (52) and (53), it is evident that analytical expressions exist, in principal, for the matrix elements of any polynomial in the basic $\hat{q}_m$ and $\hat{\pi}_n$ operators. We illustrate how this is done for the SO(3)-coupled tensor operators $[\hat{q}\otimes\hat{q}]_2$, $[\hat{q}\otimes\hat{q}]_4$, $[\hat{\pi}\otimes\hat{\pi}]_2$, $[\hat{\pi}\otimes\hat{\pi}]_4$ and $[\hat{\pi}\otimes\hat{q}\otimes\hat{\pi}]_0$. We also show that the method readily extends to rational operators such as $\beta\cos 3\gamma$.

D.1. Reduced matrix elements of various coupled tensors formed from $\hat{\mathcal{Q}}$

For what follows, we require SO(3)-reduced matrix elements of $\hat{\mathcal{Q}}$, $[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_2$, $[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_4$ and $[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0$. For the first of these, (39) and (46) gives

$$\frac{\langle v_{\rm f}\alpha_{\rm f}L_{\rm f}\|\hat{\mathcal{Q}}\|v_{\rm i}\alpha_{\rm i}L_{\rm i}\rangle}{\sqrt{2L_{\rm f}+1}} = (v_{\rm i}\alpha_{\rm i}L_{\rm i},\ 112\parallel v_{\rm f}\alpha_{\rm f}L_{\rm f})\left(\delta_{v_{\rm f},v_{\rm i}+1}\sqrt{\frac{v_{\rm i}+1}{2v_{\rm i}+5}}+\delta_{v_{\rm f},v_{\rm i}-1}\sqrt{\frac{v_{\rm i}+2}{2v_{\rm i}+1}}\right). \tag{D.1}$$

To obtain the others, we use (1) to infer that (see [18, Table I][17])

$$[\mathcal{Q}\otimes\mathcal{Q}]_{LM} = (-1)^{\frac{L}{2}}4\pi\sqrt{\frac{2}{105}}\,\mathcal{Y}^2_{1LM}\quad (L=2,4), \tag{D.2a}$$

$$[\mathcal{Q}\otimes\mathcal{Q}\otimes\mathcal{Q}]_{LM} = -(-1)^{\frac{L}{2}}\frac{4\pi}{3}\sqrt{\frac{2}{35}}\,\mathcal{Y}^3_{1LM}\quad (L=0,6). \tag{D.2b}$$

For $L\in\{2,4\}$, using first (D.2a) and (39), and then (45), we obtain

$$\begin{aligned}\frac{\langle v_{\rm f}\alpha_{\rm f}L_{\rm f}\|[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_L\|v_{\rm i}\alpha_{\rm i}L_{\rm i}\rangle}{\sqrt{2L_{\rm f}+1}} &= (-1)^{\frac{L}{2}}4\pi\sqrt{\frac{2}{105}}(v_{\rm i}\alpha_{\rm i}L_{\rm i},\ 21L\parallel v_{\rm f}\alpha_{\rm f}L_{\rm f})\langle v_{\rm f}\|\hat{\mathcal{Y}}^2\|v_{\rm i}\rangle\\ &= (-1)^{\frac{L}{2}}(v_{\rm i}\alpha_{\rm i}L_{\rm i},\ 21L\parallel v_{\rm f}\alpha_{\rm f}L_{\rm f})\left(\delta_{v_{\rm f},v_{\rm i}+2}\sqrt{\frac{(v_{\rm i}+1)(v_{\rm i}+2)}{(2v_{\rm i}+5)(2v_{\rm i}+7)}}\right.\\ &\quad\left.+\ \delta_{v_{\rm f},v_{\rm i}}\sqrt{\frac{6v_{\rm i}(v_{\rm i}+3)}{5(2v_{\rm i}+1)(2v_{\rm i}+5)}}+\delta_{v_{\rm f},v_{\rm i}-2}\sqrt{\frac{(v_{\rm i}+1)(v_{\rm i}+2)}{4{v_{\rm i}}^2-1}}\right).\end{aligned} \tag{D.3}$$

Similarly, for $L\in\{0,6\}$, using first (D.2b) and (39), and then (45), we obtain

$$\begin{aligned}\frac{\langle v_{\rm f}\alpha_{\rm f}L_{\rm f}\|[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_L\|v_{\rm i}\alpha_{\rm i}L_{\rm i}\rangle}{\sqrt{2L_{\rm f}+1}} &= -(-1)^{\frac{L}{2}}\frac{4\pi}{3}\sqrt{\frac{2}{35}}(v_{\rm i}\alpha_{\rm i}L_{\rm i},\ 31L\parallel v_{\rm f}\alpha_{\rm f}L_{\rm f})\langle v_{\rm f}\|\hat{\mathcal{Y}}^3\|v_{\rm i}\rangle\\ &= -(-1)^{\frac{L}{2}}(v_{\rm i}\alpha_{\rm i}L_{\rm i},\ 31L\parallel v_{\rm f}\alpha_{\rm f}L_{\rm f})\left(\delta_{v_{\rm f},v_{\rm i}+3}\sqrt{\frac{(v_{\rm i}+1)(v_{\rm i}+2)(v_{\rm i}+3)}{(2v_{\rm i}+5)(2v_{\rm i}+7)(2v_{\rm i}+9)}}\right.\\ &\quad+\ 3\delta_{v_{\rm f},v_{\rm i}+1}\sqrt{\frac{v_{\rm i}(v_{\rm i}+1)(v_{\rm i}+4)}{7(2v_{\rm i}+1)(2v_{\rm i}+5)(2v_{\rm i}+7)}}\\ &\quad\left.+\ 3\delta_{v_{\rm f},v_{\rm i}-1}\sqrt{\frac{(v_{\rm i}-1)(v_{\rm i}+2)(v_{\rm i}+3)}{7(4{v_{\rm i}}^2-1)(2v_{\rm i}+5)}}+\delta_{v_{\rm f},v_{\rm i}-3}\sqrt{\frac{v_{\rm i}(v_{\rm i}+1)(v_{\rm i}+2)}{(2v_{\rm i}-3)(4{v_{\rm i}}^2-1)}}\right).\end{aligned} \tag{D.4}$$

D.2. Reduced matrix elements of $[\hat{q}\otimes\hat{q}]_2$, $[\hat{q}\otimes\hat{q}]_4$ and $\beta\cos 3\gamma$

Because $q=\beta\mathcal{Q}$, we obtain

$$\langle(a,\lambda')\,\mu;\,v_{\rm f}\alpha_{\rm f}L_{\rm f}\|[\hat{q}\otimes\hat{q}]_L\|(a,\lambda)\,\nu;\,v_{\rm i}\alpha_{\rm i}L_{\rm i}\rangle = F^{(a)}_{\lambda'\mu;\lambda\nu}(\beta^2)\,\langle v_{\rm f}\alpha_{\rm f}L_{\rm f}\|[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_L\|v_{\rm i}\alpha_{\rm i}L_{\rm i}\rangle, \tag{D.5}$$

for which explicit analytic expressions are obtained, when $\lambda'-\lambda$ is even, from (D.3) and the expressions of Section 3.

In addition, (D.2b) and (43) imply that

$$\cos 3\gamma = -\sqrt{\tfrac{35}{2}}[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0, \tag{D.6}$$

whereupon

$$\langle(a,\lambda')\,\mu;\,v_{\rm f}\alpha_{\rm f}L_{\rm f}\|\beta\cos 3\gamma\|(a,\lambda)\,\nu;\,v_{\rm i}\alpha_{\rm i}L_{\rm i}\rangle = -\sqrt{\tfrac{35}{2}}\,F^{(a)}_{\lambda'\mu;\lambda\nu}(\beta)\,\langle v_{\rm f}\alpha_{\rm f}L_{\rm f}\|[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v_{\rm i}\alpha_{\rm i}L_{\rm i}\rangle, \tag{D.7}$$

for which explicit analytic expressions are obtained when $\lambda'-\lambda$ is odd, using (D.4) and the expressions of Section 3.

[17] The fifth entry in [18, Table I] contains a typographical error in that the coefficient of $\xi_6^{(6)}$ should be $\frac{3}{16}\sqrt{\frac{35}{2}}(3\sin\gamma-\sin 3\gamma)$.

D.3. Reduced matrix elements of $[\hat{\pi}\otimes\hat{\pi}]_2$ and $[\hat{\pi}\otimes\hat{\pi}]_4$

To obtain results involving the momentum operators $\hat{\pi}_m$, first note that, in view of (46), we can define operators $\hat{\mathcal{Q}}^{\pm}_M$ such that $\hat{\mathcal{Q}}_M = \hat{\mathcal{Q}}^{+}_M + \hat{\mathcal{Q}}^{-}_M$ and

$$\langle v'\|\|\hat{\mathcal{Q}}^{+}\|\|v\rangle = \delta_{v',v+1}\sqrt{\frac{v+1}{2v+5}}, \qquad \langle v'\|\|\hat{\mathcal{Q}}^{-}\|\|v\rangle = \delta_{v',v-1}\sqrt{\frac{v+2}{2v+1}}. \tag{D.8}$$

Let $\Psi^{(a,\lambda)}_{v;v\alpha LM}$ denote the wave function corresponding to $|(a,\lambda)\,v;\,v\alpha LM\rangle$, so that $\Psi^{(a,\lambda)}_{v;v\alpha LM} = \beta^{-2}\mathcal{R}^{(a,\lambda)}_{v}\mathcal{Y}^{v}_{\alpha LM}$. Then, from (D.8) and (53), we obtain

$$\hat{\pi}_m\Psi^{(a,\lambda)}_{v;v\alpha_{\rm i}L_{\rm i}M_{\rm i}} = -\frac{{\rm i}\hbar}{\beta^2}\left[\left(\frac{d}{d\beta}-\frac{v+2}{\beta}\right)\hat{\mathcal{Q}}^{+}_m + \left(\frac{d}{d\beta}+\frac{v+1}{\beta}\right)\hat{\mathcal{Q}}^{-}_m\right]\mathcal{R}^{(a,\lambda)}_{v}\,\mathcal{Y}^{v}_{\alpha_{\rm i}L_{\rm i}M_{\rm i}}. \tag{D.9}$$

It follows that

$$\begin{aligned}
\beta^2\hat{\pi}_m\hat{\pi}_n\Psi^{(a,\lambda)}_{v;v\alpha_{\rm i}L_{\rm i}M_{\rm i}} =& -\hbar^2\left(\frac{d^2}{d\beta^2}+\frac{(v+2)(v+4)}{\beta^2}-\frac{2v+5}{\beta}\frac{d}{d\beta}\right)\mathcal{R}^{(a,\lambda)}_{v}\,\hat{\mathcal{Q}}^{+}_m\hat{\mathcal{Q}}^{+}_n\mathcal{Y}^{v}_{\alpha_{\rm i}L_{\rm i}M_{\rm i}}\\
&-\hbar^2\left(\frac{d^2}{d\beta^2}-\frac{(v+1)(v+2)}{\beta^2}\right)\mathcal{R}^{(a,\lambda)}_{v}\,\hat{\mathcal{Q}}^{+}_m\hat{\mathcal{Q}}^{-}_n\mathcal{Y}^{v}_{\alpha_{\rm i}L_{\rm i}M_{\rm i}}\\
&-\hbar^2\left(\frac{d^2}{d\beta^2}-\frac{(v+1)(v+2)}{\beta^2}\right)\mathcal{R}^{(a,\lambda)}_{v}\,\hat{\mathcal{Q}}^{-}_m\hat{\mathcal{Q}}^{+}_n\mathcal{Y}^{v}_{\alpha_{\rm i}L_{\rm i}M_{\rm i}}\\
&-\hbar^2\left(\frac{d^2}{d\beta^2}+\frac{(v-1)(v+1)}{\beta^2}+\frac{2v+1}{\beta}\frac{d}{d\beta}\right)\mathcal{R}^{(a,\lambda)}_{v}\,\hat{\mathcal{Q}}^{-}_m\hat{\mathcal{Q}}^{-}_n\mathcal{Y}^{v}_{\alpha_{\rm i}L_{\rm i}M_{\rm i}}.
\end{aligned} \tag{D.10}$$

This leads to the following expressions for the non-zero SO(3)-reduced matrix elements of $[\hat{\pi}\otimes\hat{\pi}]_L$:

$$\begin{aligned}
&\langle(a,\lambda')\,\mu;\,v+2,\alpha_{\rm f}L_{\rm f}\|[\hat{\pi}\otimes\hat{\pi}]_L\|(a,\lambda)\,v;\,v\alpha_{\rm i}L_{\rm i}\rangle\\
&\quad=-\hbar^2\,F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\frac{d^2}{d\beta^2}+\frac{(v+2)(v+4)}{\beta^2}-\frac{2v+5}{\beta}\frac{d}{d\beta}\right)\langle v+2,\alpha_{\rm f}L_{\rm f}\|[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_L\|v\alpha_{\rm i}L_{\rm i}\rangle,
\end{aligned} \tag{D.11a}$$

$$\begin{aligned}
&\langle(a,\lambda')\,\mu;\,v-2,\alpha_{\rm f}L_{\rm f}\|[\hat{\pi}\otimes\hat{\pi}]_L\|(a,\lambda)\,v;\,v\alpha_{\rm i}L_{\rm i}\rangle\\
&\quad=-\hbar^2\,F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\frac{d^2}{d\beta^2}+\frac{(v-1)(v+1)}{\beta^2}+\frac{2v+1}{\beta}\frac{d}{d\beta}\right)\langle v-2,\alpha_{\rm f}L_{\rm f}\|[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_L\|v\alpha_{\rm i}L_{\rm i}\rangle,
\end{aligned} \tag{D.11b}$$

and

$$\begin{aligned}
&\langle(a,\lambda')\,\mu;\,v\alpha_{\rm f}L_{\rm f}\|[\hat{\pi}\otimes\hat{\pi}]_L\|(a,\lambda)\,v;\,v\alpha_{\rm i}L_{\rm i}\rangle\\
&\quad=-\hbar^2\,F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\frac{d^2}{d\beta^2}-\frac{(v+1)(v+2)}{\beta^2}\right)\langle v\alpha_{\rm f}L_{\rm f}\|[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_L\|v\alpha_{\rm i}L_{\rm i}\rangle.
\end{aligned} \tag{D.11c}$$

Explicit analytic expressions for these matrix elements in the cases where $\lambda'-\lambda$ is even are then obtained using (D.3) and the expressions of Section 3.

D.4. Reduced matrix elements of $[\hat{\pi}\otimes\hat{q}\otimes\hat{\pi}]_0$

To derive matrix elements of the scalar-coupled product $[\hat{\pi}\otimes\hat{q}\otimes\hat{\pi}]_0$, first note that the momenta $\hat{\pi}_N$ conjugate to the quadrupole moments $\hat{q}_M$ are defined to satisfy the commutation relations $[\hat{q}_M,\hat{\pi}_N] = (-1)^M{\rm i}\hbar\delta_{-M,N}$. This implies that $[\hat{\pi}\otimes\hat{q}\otimes\hat{\pi}]_0 = [\hat{q}\otimes\hat{\pi}\otimes\hat{\pi}]_0$, whereupon Eqs. (D.11a) and (D.11b) (or (D.10)) immediately lead to

$$\begin{aligned}
&\langle(a,\lambda')\,\mu;\,v+3,\alpha_{\rm f}L\|[\hat{\pi}\otimes\hat{q}\otimes\hat{\pi}]_0\|(a,\lambda)\,v;\,v\alpha_{\rm i}L\rangle\\
&\quad=-\hbar^2\,F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\beta\frac{d^2}{d\beta^2}+\frac{(v+2)(v+4)}{\beta}-(2v+5)\frac{d}{d\beta}\right)\langle v+3,\alpha_{\rm f}L\|[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v\alpha_{\rm i}L\rangle
\end{aligned} \tag{D.12a}$$

and

$$\begin{aligned}
&\langle(a,\lambda')\,\mu;\,v-3,\alpha_{\rm f}L\|[\hat{\pi}\otimes\hat{q}\otimes\hat{\pi}]_0\|(a,\lambda)\,v;\,v\alpha_{\rm i}L\rangle\\
&\quad=-\hbar^2\,F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\beta\frac{d^2}{d\beta^2}+\frac{(v-1)(v+1)}{\beta}+(2v+1)\frac{d}{d\beta}\right)\langle v-3,\alpha_{\rm f}L\|[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v\alpha_{\rm i}L\rangle.
\end{aligned} \tag{D.12b}$$

Explicit analytic expressions for these *stretched* matrix elements of $[\hat{\pi}\otimes\hat{q}\otimes\hat{\pi}]_0$ in the cases where $\lambda'-\lambda$ is odd are then obtained using (D.4) and the expressions of Section 3. Expressions for the other non-zero matrix elements of $[\hat{\pi}\otimes\hat{q}\otimes\hat{\pi}]_0$ are obtained, after again noting that $[\hat{\pi}\otimes\hat{q}\otimes\hat{\pi}]_0 = [\hat{q}\otimes\hat{\pi}\otimes\hat{\pi}]_0$, by combining

$$\langle v\pm1,\alpha_{\rm f}L\|[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v\alpha_{\rm i}L\rangle = \langle v\pm1,\alpha_{\rm f}L\|[(\hat{\mathcal{Q}}^{+}+\hat{\mathcal{Q}}^{-})\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v\alpha_{\rm i}L\rangle, \tag{D.13}$$

with (D.11). This yields

$$\begin{aligned}
&\langle (a,\lambda')\,\mu;\, v+1,\alpha_{\rm f}L\|[\hat{\pi}\otimes\hat{q}\otimes\hat{\pi}]_0\|(a,\lambda)\,\nu;\, v\alpha_{\rm i}L\rangle\\
&\quad=-\hbar^2\,F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\beta\frac{d^2}{d\beta^2}-\frac{(v+1)(v+2)}{\beta}\right)\langle v+1,\alpha_{\rm f}L\|[\hat{\mathcal{Q}}^+\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v\alpha_{\rm i}L\rangle\\
&\qquad-\hbar^2\,F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\beta\frac{d^2}{d\beta^2}+\frac{(v+2)(v+4)}{\beta}-(2v+5)\frac{d}{d\beta}\right)\langle v+1,\alpha_{\rm f}L\|[\hat{\mathcal{Q}}^-\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v\alpha_{\rm i}L\rangle\\
&\quad=-\hbar^2\,F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\beta\frac{d^2}{d\beta^2}-\frac{(v+1)(v+2)}{\beta}\right)\langle v+1,\alpha_{\rm f}L\|[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v\alpha_{\rm i}L\rangle\\
&\qquad+\hbar^2(2v+5)F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\frac{d}{d\beta}-\frac{v+2}{\beta}\right)\langle v+1,\alpha_{\rm f}L\|[\hat{\mathcal{Q}}^-\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v\alpha_{\rm i}L\rangle
\end{aligned}\tag{D.14a}$$

and

$$\begin{aligned}
&\langle (a,\lambda')\,\mu;\, v-1,\alpha_{\rm f}L\|[\hat{\pi}\otimes\hat{q}\otimes\hat{\pi}]_0\|(a,\lambda)\,\nu;\, v\alpha_{\rm i}L\rangle\\
&\quad=-\hbar^2\,F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\beta\frac{d^2}{d\beta^2}-\frac{(v+1)(v+2)}{\beta}\right)\langle v-1,\alpha_{\rm f}L\|[\hat{\mathcal{Q}}^-\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v\alpha_{\rm i}L\rangle\\
&\qquad-\hbar^2\,F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\beta\frac{d^2}{d\beta^2}+\frac{(v-1)(v+1)}{\beta}+(2v+1)\frac{d}{d\beta}\right)\langle v-1,\alpha_{\rm f}L\|[\hat{\mathcal{Q}}^+\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v\alpha_{\rm i}L\rangle\\
&\quad=-\hbar^2\,F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\beta\frac{d^2}{d\beta^2}-\frac{(v+1)(v+2)}{\beta}\right)\langle v-1,\alpha_{\rm f}L\|[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v\alpha_{\rm i}L\rangle\\
&\qquad-\hbar^2(2v+1)F^{(a)}_{\lambda'\mu;\lambda\nu}\left(\frac{d}{d\beta}+\frac{v+1}{\beta}\right)\langle v-1,\alpha_{\rm f}L\|[\hat{\mathcal{Q}}^+\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v\alpha_{\rm i}L\rangle.
\end{aligned}\tag{D.14b}$$

To evaluate these, we require, in addition to expressions for the reduced matrix elements $\langle v\pm1,\alpha_{\rm f}L\|[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v\alpha_{\rm i}L\rangle$ provided by (D.4), expressions for the reduced matrix elements $\langle v\pm1,\alpha_{\rm f}L\|[\hat{\mathcal{Q}}^\mp\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v\alpha_{\rm i}L\rangle$. To obtain the latter, first note that (39) implies that

$$\frac{\langle v\pm1,\alpha_{\rm f}L\|[\hat{\mathcal{Q}}^\mp\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v\alpha_{\rm i}L\rangle}{\sqrt{2L+1}\,(v\alpha_{\rm i}L,\ 310\parallel v\pm1,\alpha_{\rm f}L)}=\langle v\pm1\|\|\,[[\hat{\mathcal{Q}}^\mp\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]]^3\|\|v\rangle,\tag{D.15}$$

where $\langle v\pm1\|\|\,[[\hat{\mathcal{Q}}^\mp\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]]^3\|\|v\rangle$ is an SO(5)-reduced matrix element. To evaluate this, we consider the special cases of the left side with $L=\min\{2v,2(v\pm1)\}$ and $\alpha_{\rm i}=\alpha_{\rm f}=1$. For this, we use the standard expression for factoring an SU(2) or SO(3)-reduced matrix element of angular momentum zero (see (108)). In a form applicable here, this reads

$$\langle v_{\rm f}\alpha_{\rm f}L\|[A_{L_1}\otimes B_{L_1}]_0\|v_{\rm i}\alpha_{\rm i}L\rangle=\sum_{v',\alpha',L'}(-1)^{L+L_1-L'}\frac{\langle v_{\rm f}\alpha_{\rm f}L\|A_{L_1}\|v'\alpha'L'\rangle\langle v'\alpha'L'\|B_{L_1}\|v_{\rm i}\alpha_{\rm i}L\rangle}{\sqrt{(2L_1+1)(2L+1)}}.\tag{D.16}$$

For $A_2=\hat{\mathcal{Q}}^\mp$, $B_2=[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_2$, $v_{\rm i}=v$, $v_{\rm f}=v\pm1$ and $L=\min\{2v_{\rm i},2v_{\rm f}\}$, this yields

$$\begin{aligned}
&\langle v+1,1,2v\|[\hat{\mathcal{Q}}^-\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v,1,2v\rangle\\
&\quad=\sum_{\alpha',L'}(-1)^{L'}\frac{\langle v+1,1,2v\|\hat{\mathcal{Q}}\|v+2,\alpha',L'\rangle\langle v+2,\alpha',L'\|[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_2\|v,1,2v\rangle}{\sqrt{5(4v+1)}}
\end{aligned}\tag{D.17a}$$

and

$$\begin{aligned}
&\langle v-1,1,2v-2\|[\hat{\mathcal{Q}}^+\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v,1,2v-2\rangle\\
&\quad=\frac{\langle v-1,1,2v-2\|\hat{\mathcal{Q}}\|v-2,1,2v-4\rangle\langle v-2,1,2v-4\|[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_2\|v,1,2v-2\rangle}{\sqrt{5(4v-3)}}.
\end{aligned}\tag{D.17b}$$

In the second case here, use has been made of the fact that, in accordance with (6), all states of seniority v have SO(3) angular momentum at most $2v$, and this angular momentum has multiplicity one. Thus, because the values of L' being summed over may be restricted to the range $L-2\le L'\le L+2$, with $L=2v-2$ in this case, just one state contributes to the sum from (D.16).

Given that the value of (D.15) is independent of L, $\alpha_{\rm i}$ and $\alpha_{\rm f}$, we conclude that

$$\begin{aligned}
&\langle v+1,\alpha_{\rm f}L\|[\hat{\mathcal{Q}}^-\otimes\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_0\|v\alpha_{\rm i}L\rangle\\
&\quad=\frac{1}{4v+1}\sqrt{\frac{2L+1}{5}}\frac{(v\alpha_{\rm i}L,\ 310\parallel v+1,\alpha_{\rm f}L)}{(v,1,2v,\ 310\parallel v+1,1,2v)}\\
&\qquad\times\sum_{\alpha',L'}(-1)^{L'}\langle v+1,1,2v\|\hat{\mathcal{Q}}\|v+2,\alpha'L'\rangle\langle v+2,\alpha'L'\|[\hat{\mathcal{Q}}\otimes\hat{\mathcal{Q}}]_2\|v,1,2v\rangle
\end{aligned}\tag{D.18a}$$

and

$$\begin{aligned}&\langle v-1, \alpha_f L \| [\hat{\mathcal{Q}}^+ \otimes \hat{\mathcal{Q}} \otimes \hat{\mathcal{Q}}]_0 \| v \alpha_i L \rangle \\ &\quad = \frac{1}{4v-3}\sqrt{\frac{2L+1}{5}} \frac{(v\alpha_i L,\ 310 \parallel v-1, \alpha_f L)}{(v, 1, 2v-2,\ 310 \parallel v-1, 1, 2v-2)} \\ &\qquad \times \langle v-1, 1, 2v-2 \| \hat{\mathcal{Q}} \| v-2, 1, 2v-4 \rangle \langle v-2, 1, 2v-4 \| [\hat{\mathcal{Q}} \otimes \hat{\mathcal{Q}}]_2 \| v, 1, 2v-2 \rangle. \end{aligned} \tag{D.18b}$$

Analytic expressions for the SO(3)-reduced matrix elements $\langle (a, \lambda')\ \mu;\ v \pm 1, \alpha_f L \| [\hat{\pi} \otimes \hat{q} \otimes \hat{\pi}]_0 \| (a, \lambda)\ v;\ v\alpha_i L \rangle$ in the cases where $\lambda' - \lambda$ is odd are then obtained from (D.14), making use of (D.4), (D.18) via (D.1) and (D.3), and the expressions of Section 3.

Appendix E. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.cpc.2015.10.017.

References

[1] D.J. Rowe, Nuclear Phys. A 735 (2004) 372–392.
[2] D.J. Rowe, P.S. Turner, Nuclear Phys. A 753 (2005) 94–105.
[3] D.J. Rowe, J. Phys. A: Math. Gen. 38 (2005) 10181–10201.
[4] D.J. Rowe, T.A. Welsh, M.A. Caprio, Phys. Rev. C 79 (2009) 054304.
[5] A. Bohr, Mat. Fys. Medd. Dan. Vid. Selsk. 26 (14).
[6] J. Čížek, J. Paldus, Int. J. Quantum Chem. 12 (1977) 875–896.
[7] T.H. Cooke, J.L. Wood, Amer. J. Phys. 70 (2002) 945–950.
[8] D.J. Rowe, J.L. Wood, Fundamentals of Nuclear Models; Foundational Models, World Scientific, Singapore, 2010.
[9] E. Chaćon, M. Moshinsky, R. Sharp, J. Math. Phys. 17 (1976) 668–676.
[10] E. Chaćon, M. Moshinsky, J. Math. Phys. 18 (1977) 870–880.
[11] J.M. Eisenberg, W. Greiner, Nuclear Models, third ed., North Holland, Amsterdam, 1987.
[12] G. Gneuss, W. Greiner, Nuclear Phys. A 171 (1971) 449–479.
[13] K.T. Hecht, Nuclear Phys. 63 (1965) 177–213.
[14] P.O. Hess, M. Seiwert, J.A. Maruhn, W. Greiner, Z. Phys. A 296 (1980) 147–163.
[15] P.O. Hess, J.A. Maruhn, W. Greiner, J. Phys. G: Nucl. Phys. 7 (1981) 737–769.
[16] D. Troltenier, J.A. Maruhn, P.O. Hess, in: K. Langanke, J.A. Maruhn, S.E. Koonin (Eds.), Computational Nuclear Physics, vol. 1, Springer, Berlin, 1991.
[17] Maplesoft, Waterloo Maple Inc., Waterloo, ON, Canada.
[18] M.A. Caprio, D.J. Rowe, T.A. Welsh, Compt. Phys. Comm. 180 (2009) 1150–1163.
[19] T.A. Welsh, C++ code, 2008, unpublished.
[20] D.J. Rowe, P.S. Turner, J. Repka, J. Math. Phys. 45 (2004) 2761–2784.
[21] P.M. Davidson, Proc. R. Soc. Lond. Ser. A 135 (1932) 459–472.
[22] G. Thiamova, D.J. Rowe, M. Caprio, Nuclear Phys. A 895 (2012) 20–32.
[23] A.R. Edmonds, Angular Momentum in Quantum Mechanics, second ed., Princeton University Press, New Jersey, 1960.
[24] T. Kishimoto, T. Tamura, Nuclear Phys. A 163 (1971) 100–112.
[25] M. Abramowitz, I.A. Stegun, Handbook of Mathematical Functions, Dover Publications, New York, 1968.
[26] G. Racah, Phys. Rev. 76 (1949) 1352–1365.
[27] L. Wilets, M. Jean, Phys. Rev. 102 (1956) 788–796.
[28] J. Meyer-ter-Vehn, Nuclear Phys. A 249 (1975) 111–140.
[29] A. Bohr, B.R. Mottelson, Nuclear Structure, Benjamin, New York and Reading, Mass., Vol. 1: 1969, Vol. 2: 1975, (republished by World Scientific, Singapore, 1998).
[30] A. Arima, F. Iachello, Ann. Phys. 99 (1976) 253–317.
[31] A. Arima, F. Iachello, Ann. Phys. 111 (1978) 201–238.
[32] F. Iachello, A. Arima, The Interacting Boson Model, Cambridge University Press, Cambridge, 1987.
[33] D.J. Rowe, Nuclear Phys. A 745 (2004) 47–78.
[34] D. Janssen, R.V. Jolos, Y. Dönau, Nuclear Phys. A 224 (1974) 93–115.
[35] J.N. Ginocchio, M.W. Kirson, Phys. Rev. Lett. 44 (1980) 1744–1747.
[36] A.E.L. Dieperink, O. Scholten, F. Iachello, Phys. Rev. Lett. 44 (1980) 1747–1750.
[37] D.J. Rowe, G. Thiamova, Nuclear Phys. A 760 (2005) 59–81.