REPRESENTATIONS OF U(3) IN U(N) *

J.P. DRAAYER, Y. LEISCHER, S.C. PARK and R. LOPEZ

Department of Physics and Astronomy, Louisiana State University, Baton Rouge, LA 70803-4401, USA

Received 28 May 1989

An interactive FORTRAN code for determining the representations of U(3) that occur in a representation of U(N) is introduced. The U(N) \to U(3) chart is the basic group structure of the isotropic oscillator in three dimensions. In particular, \( N = (n + 1)(n + 2)/2 \) is the degeneracy of the shell with \( n \) quanta per level. Since the oscillator potential is a good starting approximation for the self-consistent field that binds nucleons in the nucleus, motivation for the work comes from nuclear physics, and in particular, from studies of quadrupole collective phenomena in deformed nuclear systems. A PASCAL version of the program, which is simpler because the basic algorithm is a recursive one, is also available.

0010-4655/89/$03.50 © Elsevier Science Publishers B.V. (North-Holland)

PROGRAM SUMMARY

Title of program: UNTOUC3

Catalogue number: AB12

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

Computer IBM 360/6500 and DEC VAX-11/750. Installation: Louisiana State University

Operating systems: MV5/VMS, VA-311 and VMS-45, respectively

Programming language used: FORTRAN (version in PASCAL also available)

High level storage required: 231532 and 219221 bytes on the IBM and Dijc systems, respectively. These numbers are for the full package, including the main and binary tree routines and a 10000 node binary tree writer, see below

Peripheral used: none

Number of lines in program: the count is 498 (516 for routines in the interactive subprogram) versions of the UNTOUC3 package

CPC Program Library subprogram used: codes from RHFREE, a binary tree package (360 lines of code) are required [1].

* Work supported in part by a grant from the US National Science Foundation.

Keywords: harmonic oscillator, three-dimensional oscillator, Effort S(U(3)) scheme, symplectic shell model, microscopic collective models, U(3) symmetry, algebraic theory, dynamical symmetry, U(N) \to U(3), unitary group plethysm.

Nature of the physical problem

U(N) \to U(3) plethysm, that is, finding a complete set of irreducible representations (irreps) of U(3) in specific irreps of U(N), where \( N = (n + 1)(n + 2)/2 \) for nonnegative integer \( n \) values.

Method of solution

Solutions are obtained by applying a simple difference algorithm to the U(3) weight distribution function [2]. The latter is generated in three steps: (1) by identifying the \( N \) irreps of U(N) as the distinguishable arrangements of \( n \) oscillator quanta in three Cartesian directions, (2) by distributing the total number of quanta (\( n = m + n \) is the number of oscillator particles) among these levels subject to restrictions (betweenness conditions) of the Giladiad scheme for labelling basis states of U(N), and (3) by iterating over all the \( N \) levels to determine the final distribution of quanta in the three Cartesian directions.

Imprecision on the complexity of the problem

The main limitation is cpu time, see below. Storage can be a problem, but the time constraint usually sets the long before program use becomes a problem. A PC with 640 K of memory is sufficient to run most cases of interest in nuclear physics.
Typical running times:

<table>
<thead>
<tr>
<th></th>
<th>VAX 11/750</th>
<th>IBM 3090 (4MIPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution</td>
<td>1.0</td>
<td>0.7</td>
</tr>
<tr>
<td>time</td>
<td>1.0</td>
<td>1.2</td>
</tr>
<tr>
<td>(normalized)</td>
<td>1.0</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Unusual features of the program:

Return codes set in the subprograms are used to fix branches.

1. Introduction

The purpose of this article is to document a program for calculating the reduction of irreducible representations (irreps) of the unitary group in N dimensions, U(N), into irreps of the unitary group in three dimensions, U(3). The U(N) → U(3) chain is a basic group structure of the many-particle, three-dimensional harmonic oscillator problem [1]. Specifically, the oscillator shell with n quanta, see fig. 1, consists of N = (n + 1)(n + 2)/2 levels that can be identified as the distinguishable distributions of n identical objects (quanta) in the three Cartesian (x, y, z) directions. Hence N = 3, 6, 10, 15, 21, 28, 36, ... for the n = 1, 2, 3, 4, 5, 6, 7, ... shells, respectively.

The physical motivation for considering this problem comes from nuclear physics. The reason is that the three-dimensional harmonic oscillator is a good starting approximation for the self-consistent potential that binds nucleons within the nucleus. While this has been appreciated for many years [2], recent developments have further underscored the importance of the SU(3) picture. In particular, we now have an analytic mapping between the free parameters that characterize a quantum rotor and a simple SU(3) → SO(3) integrity basis interaction [3]. Since many nuclei display rotational characteristics, this is important in efforts aimed at gaining a microscopic description of collective rotations in nuclei.

It is important to point out that although the spin-orbit interaction destroys the simple oscillator picture for heavy nuclei, another realization of the oscillator symmetry emerges in the rare-earth and actinide regions that is even better than the first. This new symmetry is called the pseudo SU(3) model [4]. The prefix "pseudo" comes from earlier theoretical work that demonstrated the physical significance of spin-orbit doublets with j = l + 1/2 and j = (l + 2) - 1. These doublets can be considered partners of a pair of orbitals with pseudo l = l + 1 and pseudo-spin 1/2, that is, j = l 1/2. Though the relabeling only represents a unitary transformation among basis states, it is a transformation that the many-body nuclear Hamiltonian favors [5]. The pseudo spin-orbit doublets of a major shell, taken together, can be identified as members of a pseudo-oscillator scheme of one less quanta than the parent structure, N = n - 1. So the U(N) → U(3) problem applies for heavy nuclei as well as light ones.

Another point that connects to the importance of the U(N) → U(3) group structure is the symplectic model [6] and its extension to heavy nuclei, the pseudo-symplectic scheme [7]. This is just the shell-model scheme referred to above, but extended to include collective excitations of the quadrupole type. The relevant symmetry is Sp(3, R), the dynamical symmetry group (noncompact) of the oscillator: Sp(3, R) → U(3), where the U(3) irreps are those contained in the shell-model reduction U(N) → U(3). This scheme promises to yield a comprehensive understanding of rotations, including enhanced E2 transition rates and the giant quadrupole mode in nuclei.
2. Mathematical algorithm

The Gellett pattern is a convenient labeling scheme for basis states of $U(N)$ representations [8]. If the irrep is labeled in the standard way by a Young shape, $\{ f \} = \{ f_1, f_2, \ldots, f_n \}$, then

$$\rho_{n = 0} = \begin{bmatrix}
    h_{1,3} & h_{2,3} & h_{3,3} & \cdots & h_{N-1,3} & h_{N,3} \\
    h_{1,2} & h_{2,2} & h_{3,2} & \cdots & h_{N-1,2} & h_{N,2} \\
    h_{1,1} & h_{2,1} & h_{3,1} & \cdots & h_{N-1,1} & h_{N,1} \\
    h_{1,0} & h_{2,0} & h_{3,0} & \cdots & h_{N-1,0} & h_{N,0} \\
    \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
    h_{1,0} & h_{2,0} & h_{3,0} & \cdots & h_{N-1,0} & h_{N,0} \\
  \end{bmatrix}$$

(1)

where $h_{a,b} - f_a$ for $a = 1, 2, \ldots, N$. The $h_{a,b}$, $a \leq b \leq N$, are nonnegative integers that satisfy betweenness conditions: $h_{a,b} \geq h_{a,d-1} \geq h_{a+1,b}$. This is a complete labeling scheme, so the number of distinct patterns is equal to the dimension of the $U(N)$ irrep,

$$d[ f ] = \prod_{j=1}^{N-1} \prod_{k=1}^{N-j} (f_j - f_k)(j-k)/(j-k).$$

(2)

The $h_{a,b}$ specify irreps in the canonical reduction of $U(N)$ with respect to its unitary subgroups, $U(N) \rightarrow U(N-1) \rightarrow \cdots \rightarrow U(2) \rightarrow U(1)$. The irrep of $U(M)$ in the chain $U(M) \rightarrow \cdots \rightarrow U(2) \rightarrow U(1)$ to which the state belongs is given by the Young shape $\{ f \} = \{ h_{1,1}, h_{2,2}, \ldots, h_{N,N} \}$. Moving down the pattern by one row, say from $M$ to $M-1$, corresponds to reducing the dimensionality of the space by one.

The weight vector of a basis state of $U(N)$ is the set of $N$ numbers,

$$w_n = \sum [\text{row}(a)] - \sum [\text{row}(a-1)] - \sum h_{a,b} - \sum h_{a,b+1} - s_a - s_{a-1}.$$  

(3)

If the $\{ f \}$ label refers to the distribution of $m$ particles among the $N$ levels of $U(N)$, the $w_n$ can be thought of as specifying the number of particles in the $a$th level, hence $\sum w_n = m - \Sigma h_{a,b}$. The weights, $w_n$, are therefore eigenvalues of operators, $n_a$, that measure level occupancy, $(n_a) \rightarrow w_n$.

This organizational structure yields a simple means for determining the $U(3)$ content of a $U(N)$ irrep. If the $N$ levels of $U(N)$ are taken to be the distinct arrangements of $n$ indistinguishable quanta among the three Cartesian $(z, x, y)$ directions, see fig. 1, the sums,

$$n_i = \sum_n n_{i,n} w_n,$$  

(4)

where $n_{i,n}$ is the number of quanta of the $i = (z, x, y)$ type in the $n$th level, give the total number of $(z, x, y)$ quanta for a specific pattern. The full set, one for each of the allowed patterns, defines the $U(3)$ weight distribution of the $U(N)$ irrep. This, in turn, suffices to determine the $U(3)$ irreps in the $U(N)$ irrep through a simple difference algorithm. Specifically, the following expression yields level dimensionalities, $D[ f ]$, that is, the number of times an irrep $\{ f \}$ of $U(m)$ occurs in a distribution $D[ f ]$ of $U(m)$ weights [9,10].
Here $J$ denotes a determinant and the $O_j$ are shift operators that carry a pattern $[f] = \{f_1, f_2, \ldots\}$ into a new one with the $j$th element increased by an amount $t$, $[f'] = \{f_1, f_2 + t, \ldots\}$. For the U(3) case this general result reduces to a six-term difference equation:

$$D[f; f_1, f_2, f_3] - D[f; f_1 + 1, f_2, f_3] - D[f; f_1 - 1, f_2 + 1, f_3] + D[f; f_1 + 2, f_2 - 1, f_3] - D[f; f_1, f_2 - 2, f_3] + D[f; f_1, f_2, f_3 - 2] = 0,$$

(5)

Applying this result to the U(3) weight distribution of a U(N) irrep, determined by following the procedure outlined above, yields the number of times $[f_1, f_2, f_3]$ of U(3) occurs in the $[f_1, f_2, \ldots, f_N]$ irrep of U(N).

As a simple example, consider the problem of finding the U(6) $\to$ U(3) reduction for the 6 particle $[f] = \{2, 1, 1, 1, 1, 0\}$ irrep of U(6). The dimensionality of this irrep, see (2), is 35. The highest

<table>
<thead>
<tr>
<th>Oscillator</th>
<th>Shells</th>
</tr>
</thead>
<tbody>
<tr>
<td>$6 \cdot N = 35$</td>
<td></td>
</tr>
<tr>
<td>6, 5, 4, 3, 2, 1</td>
<td></td>
</tr>
<tr>
<td>$^0\text{O}_1$, $^1\text{O}_2$, $^2\text{O}_3$, $^3\text{O}_4$, $^4\text{O}_5$, $^5\text{O}_6$</td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 1.** Shell structure of the three-dimensional isotropic harmonic oscillator. The degeneracy of each shell is given in terms of spherical harmonics by $N = \frac{1}{2}(N+1)(N+2)/2$. In nuclear physics the number of particles per shell is 2 for identical nucleons (spin 1 or spin 1/2), 4 for a sum of multiplets (spin 0, 1, 2, 3). Hence, for example, the 6-shell holds a maximum of $6 \cdot 4 = 24$ particles.

**Fig. 2.** The highest-weight state for the $[f] = \{0, 4\}$ distribution of U(6) passes 4 particles in the first two lines: $\Omega_0$, $^4\text{O}_4$, $^3\text{O}_3$, and $^2\text{O}_2$. The U(3) weight vector is therefore $\{4, 0, 0, 0, 0\}$, and the corresponding U(3) weight vector is $\{12, 6, 0\}$. Each $\eta_j = \pm 1$ and $\eta_j = \mp 1$. Since the highest-weight state is unique, so is the corresponding U(3) irrep, see (3). Specifically, for the $[f] = \{0, 4\}$ example, $[f] \to [12, 6, 0] \to \{3, 2, 1\}$, which is a well-known fact that applies for $^{245}\text{Mcg}$. The highest-
weight state is the Gelfand pattern with the $k_{a\beta}$ taking on their maximum values:

\[
\begin{align*}
211110 \\
21111 \\
2111 \\
211 \\
2
\end{align*}
\]

(7)

The corresponding weight vector, see (3), is $(w) = (2, 1, 1, 1, 1, 0)$. States of lower weight include, for example,

\[
\begin{align*}
2111110 & 211110 211110 \\
21110 & 21110 11110 \\
2111 & 2110 1110 \\
211 & 20 10 \\
2 & 1 0 \\
2 & 1 & 0
\end{align*}
\]

(8)

The first of these three has weight $(w) = (2, 1, 1, 1, 0, 1)$, the second has weight $(w) = (1, 1, 1, 1, 1, 1)$, while the weight vector of the last, the lowest-weight state, is $(w) = (0, 1, 1, 1, 1, 2)$. The weight vectors of extremal states, like the highest- and lowest-weight patterns, are always unique. In general, however, the weight vectors are not unique identifiers, so the full Gelfand scheme is needed. A complete list of distinct weight vectors for the $[2, 1^4]$ irrep of $U(6)$ is given in table 1. Opposite each entry is the associated $U(3)$ weight distribution, see (4). In determining the $U(3)$ weights, the assignment of oscillator quanta to levels was done as specified in fig. 1. Note that $(w) = (1, 1, 1, 1, 1, 1)$ occurs five times and is therefore an example of a nonunique weight vector.

An application of the difference algorithm (6) to the data in table 2 suffices for determining the $U(3)$ content of the $[2, 1^4]$ irrep of $U(6)$. Starting at the top of the list and working downward, remembering that for a valid $U(3)$ irrep $f_1 \geq f_2 \geq f_3$, one finds the following nonzero $D_{ij}$ values:

\[
\begin{align*}
D_{[6, 4, 2]} & = D_{[6, 4, 2]} + D_{[7, 5, 0]} + D_{[8, 3, 1]} - D_{[8, 4, 0]} - D_{[7, 3, 2]} - D_{[6, 5, 1]} \\
& = -1 + 0 + 0 - 0 - 0 - 1 = 1, \\
D_{[5, 4, 3]} & = D_{[5, 4, 3]} + D_{[6, 5, 1]} - D_{[7, 4, 1]} - D_{[6, 3, 3]} - D_{[5, 5, 2]} \\
& = -3 + 0 + 0 - 1 - 1 = -1.
\end{align*}
\]

(10a)

(10b)

All other $D_{ij}$'s vanish, for example:

\[
\begin{align*}
D_{[4, 4, 4]} & = D_{[4, 4, 4]} + D_{[5, 5, 2]} + D_{[6, 3, 3]} - D_{[6, 4, 2]} - D_{[5, 3, 4]} - D_{[4, 5, 3]} \\
& = 5 + 1 + 1 - 1 - 3 = 0.
\end{align*}
\]

(11)

Since the dimension of the $[f] = [6, 4, 2]$ irrep of $U(3)$ is 27 and that of $[f] = [5, 4, 3]$ is 8, these are the only $U(3)$ irreps in the $U(3)$ representation:

\[
\begin{align*}
U(6) & \rightarrow U(3) & \leftarrow \text{group reduction}, \\
[2, 1^4] & \rightarrow [6, 4, 2] \cdot [5, 4, 3] & \leftarrow \text{representations}, \\
35 & \rightarrow 27 + 8 & \leftarrow \text{dimension check}
\end{align*}
\]

(12)
Table 1

<table>
<thead>
<tr>
<th>Number</th>
<th>Weights</th>
<th>Number</th>
<th>Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>U(3)</td>
<td>U(3)</td>
<td>U(3)</td>
</tr>
<tr>
<td>1</td>
<td>211110</td>
<td>64^2</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>211101</td>
<td>633</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>211011</td>
<td>624</td>
<td>21*</td>
</tr>
<tr>
<td>4</td>
<td>210111</td>
<td>543</td>
<td>22</td>
</tr>
<tr>
<td>5</td>
<td>201111</td>
<td>534</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>121110</td>
<td>552</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>121101</td>
<td>543</td>
<td>25*</td>
</tr>
<tr>
<td>8</td>
<td>121011</td>
<td>534</td>
<td>26</td>
</tr>
<tr>
<td>9</td>
<td>120111</td>
<td>453</td>
<td>27</td>
</tr>
<tr>
<td>10</td>
<td>121101</td>
<td>534</td>
<td>28</td>
</tr>
<tr>
<td>11</td>
<td>121011</td>
<td>525</td>
<td>29</td>
</tr>
<tr>
<td>12</td>
<td>120111</td>
<td>462</td>
<td>30*</td>
</tr>
<tr>
<td>13</td>
<td>111210</td>
<td>462</td>
<td>31</td>
</tr>
<tr>
<td>14</td>
<td>121201</td>
<td>435</td>
<td>32</td>
</tr>
<tr>
<td>15</td>
<td>111120</td>
<td>453</td>
<td>33</td>
</tr>
<tr>
<td>16*</td>
<td>111111</td>
<td>444</td>
<td>34</td>
</tr>
<tr>
<td>17</td>
<td>111110</td>
<td>435</td>
<td>35</td>
</tr>
<tr>
<td>18*</td>
<td>111111</td>
<td>444</td>
<td>36</td>
</tr>
</tbody>
</table>

The code UNTOU3 carries out this exercise for general irreps of U(N). It gives a complete list of the U(3) irreps, the frequency with which they occur, and values for the dimension of the irreps and the Casimir invariants:

\[
D(\lambda, \mu) = (\lambda + \mu + 2)(\lambda + 1)(\mu + 1)/2, \tag{13a}
\]

\[
C_2(\lambda, \mu) = \lambda^2 + 2\lambda \mu + \mu^2 + 3(\lambda + \mu), \tag{13b}
\]

\[
C_3(\lambda, \mu) = (\lambda - \mu)(\lambda + 2\mu + 3)(2\lambda + \mu + 3). \tag{13c}
\]

Table 2

U(3) weight vectors with multiplicities for the [I] = [2, 1, 1]* group of U(6), see table 1. Only the starred entries are required to determine the U(3) irreps that are represented. This illustrates a general result, namely, since \(I_3 \geq I_2 \geq I_1\) for the D's, only the D's with \(I_3 \geq I_2 \geq I_1 \geq -3\) are needed to determine the U(3) irreps in content of a weight distribution, see (6). An application of the difference algorithm (6) to this data yields: [2, 1, 1]* = [6, 4, 2]* \(I_3(\lambda, \mu) = (2, 2); + [5, 4, 3]* \(I_3(\lambda, \mu) = (1, 3);\) for the U(6) \(-\) U(3) decomposition

<table>
<thead>
<tr>
<th>((I_3, I_2, I_1))</th>
<th>(D(I_3, I_2, I_1))</th>
<th>((I_3, I_2, I_1))</th>
<th>(D(I_3, I_2, I_1))</th>
</tr>
</thead>
<tbody>
<tr>
<td>635</td>
<td>1</td>
<td>435</td>
<td>3</td>
</tr>
<tr>
<td>635*</td>
<td>1</td>
<td>426</td>
<td>3</td>
</tr>
<tr>
<td>624</td>
<td>1</td>
<td>353</td>
<td>3</td>
</tr>
<tr>
<td>512</td>
<td>1</td>
<td>334</td>
<td>3</td>
</tr>
<tr>
<td>512*</td>
<td>1</td>
<td>345</td>
<td>2</td>
</tr>
<tr>
<td>525</td>
<td>1</td>
<td>264</td>
<td>1</td>
</tr>
<tr>
<td>525*</td>
<td>1</td>
<td>253</td>
<td>1</td>
</tr>
<tr>
<td>462</td>
<td>1</td>
<td>246</td>
<td>1</td>
</tr>
<tr>
<td>444*</td>
<td>1</td>
<td>244</td>
<td>1</td>
</tr>
</tbody>
</table>
In these expressions irrep labels of SU(3) rather than U(3) are used, that is, \([f'_1, f_2, f_3] \rightarrow (\lambda, \mu) = (f_1 - f_2, f_2 - f_3)\).

3. Program

The UNTOU3 package consists of a main program, three subroutines, two functions, and routines from BETTKE [11], a balanced binary tree package, that provide the requisite software for dynamic fetch and insert operations on a database where U(3) weight information, the \([f_1, f_2, f_3]\) in (6), is stored. A brief description of each program and the function it serves follows:

**MAIN**

The main program is divided into three parts. The first section is simply a program administration, where the values of various parameters are fixed, arrays are initialized, the oscillator shell number which is equal to the number of quanta per level \((n)\) is read in, the distribution of the quanta within the shell is determined through a call to a subroutine called CONFUN, and the \(N=(n+1)(n+2)/2\) numbers that are required to specify a representation of \(U(N)\) are input. In the second section the required Gelfand patterns are generated and a database that contains the U(3) weight information is built. This involves multiple calls to a subroutine CONFUN that generates the allowed patterns, see below, in the third and final section of the main program the difference algorithm is applied against the database where the U(3) weight information is stored to extract the U(3) content of the \(U(N)\) representation. The U(3) irrep labels together with values for the dimension and the second and third order Casimir invariants of each are output. If the sum of the U(3) dimensions is not equal to the dimension of the \(U(N)\) irrep, a program interrupt is generated.

**CONFUN**

This subroutine generates the distribution of \(n\) quanta in the three Cartesian directions, \((z, x, y)\), see fig. 1. Each of the \(N=(n+1)(n+2)/2\) levels of a major oscillator shell is associated with a unique set of \((n_z, n_x, n_y)\) values. These numbers are the \(n_{ij}\) of (4) that enter into a determination of the U(3) weight vector associated with a specific \(U(N)\) basis state.

**CONVERT**

The allowed Gelfand patterns are generated by successive calls to this routine. The initial highest-weight state for \(U(N)\) requires \(N\) calls to CONFUN, one for each row in the pattern. A new pattern is generated from the previous one by changing pattern entries, the \(n_{ij}\) of (1), in order. For example, if \(n_{i1} \rightarrow n_{i1} + 1\) is allowed, it will be the second pattern generated. If it is not allowed, then \(n_{i2} \rightarrow n_{i2} + 1\) with \(n_{i1} \rightarrow n_{i1} + 1\) and \(n_{i2} \rightarrow n_{i2} + 1\) with \(n_{i1} \rightarrow n_{i1} + 1\) equal to its old value is tested, etc.

The pattern generation procedure is recursive. Since FORTRAN, as compared to PASCAL, for example, is a nonrecursive language, the recursive feature is forced by branching to different locations in the main program depending on the condition code set in CONFUN: normal return if finding a new distribution requires increasing the sum \(n_i=\sum n_{ij}\) by one; return 1 if a new distribution is found by simpler shift operations, for example \(n_{i1} \rightarrow n_{i1} + 1\) and \(n_{i2} \rightarrow n_{i2} + 1\), within a row keeping the sum \(n_i\) constant; and return 2 when all such possibilities are exhausted. The \((N+1)\)-st component of the \(U(N)\) weight vector decreases by one, \(n_{i1} \rightarrow n_{i1} - 1\), under a normal return because \(x_y\) increases by one while \(n_{i1} + 1\) remains uncharged. The weight remains the same for a return 1 condition code. A return 2 means one must move up a row for possible additional patterns. The generation process is complete when a return 2 from the \((N+1)\)-st row is encountered.
EZOFT
This is an elementary subprogram for generating output statements. There are two entries: TKO and OUT. The entry TKO stops execution after giving a fatal error message whereas the entry OUT prints a message and execution continues.

DIMUN
This function evaluates the expression for determining the dimension of an irrep of U(1), see (2).

DTELG
This is a support function for DIMUN that uses logarithm arithmetic so underflow and/or overflow conditions in the evaluation of the products in (2) are avoided.

TSET
This routine is the first of three from the binary tree package called BBTREE. It is used to initialize the linear array where the U(3) weight information is stored.

TREE
A subroutine from BBTREE that is used to build the U(3) database. Fetch and insert times go as $\varepsilon([\log(n)])$, where $n$ is the number of items in the list. Only the $D(f_1, f_2, f_3)$ values with $f_1 \geq f_2 \geq f_3 > 2$ are needed, see (6), so the database array is typically half or less than the dimension of the U(N) irrep in length. This is illustrated by the example cited in tables 1 and 2.

TOUT
This is another subroutine from BBTREE. It allows one to extract information from the tree in ascending order as is needed for a systematic application of the difference algorithm (6). An accumulator in the main program keeps track of the sum of the product of the $D(f_1, f_2, f_3)$ and the corresponding U(3) multiplicity. When this sum equals the dimension of the U(N) irrep, processing stops. If, for some reason,

![Graph](image_url)

Fig. 2. Total cpu times plotted as a function of the dimension of the U(N) irrep for $/ell = [2^s, 1]$ with $s = 1, 2, 3$. The results shown are for the fourth oscillator shell. Similar results apply for other U(N) symmetries. This shows that within a single shell of the oscillator the overall CPU time increases approximately linearly with the dimension of the irrep.
the sum fails to check against the \( U(N) \) dimension, a TKO fatal error statement is generated and processing is stopped.

The CONFUN algorithm is called recursively so total cpu time can become excessive. The dependence on the dimension of the irrep, \( d \), is approximately linear while the dependence on the number of oscillator quanta, \( n \), is close to being exponential. This behavior for a typical case is shown in figs. 2 and 3, respectively. A function that can be used to determine the cpu time to a reasonable degree of accuracy is given by

\[
t(\text{cpu}) = nd \exp(\beta n).
\]

The constants \( n \) and \( \beta \) are roughly \( 13 \times 10^{-7} \) and 2.4 for the VAX 11/750 and \( 0.7 \times 10^{-7} \) and 2.6 on the IBM 3090/608E systems, respectively. As written, the codes work for all integer \( n \leq 7 \) and up to 10000 nodes in the binary tree. The latter is about \( \frac{1}{2} \) of the maximum allowed value for the dimension of the \( U(N) \) irrep, because not all the basis states are needed due to \((z, x, y)\) interchange symmetries. To repeat, if an overflow condition is encountered a warning message is printed and execution stopped. Whereas the number of nodes can be easily increased, cpu time is the more critical limiting factor. While simpler in form because of the recursive nature of the language, the PASCAL version of the code executes less efficiently than the FORTRAN version of the program.

Acknowledgement

The authors acknowledge encouragement from J. Lauritsen and G. Bersch, whose need for \( U(N) \rightarrow U(3) \) results in their work on the nuclear partition function [12] stimulated this product.
References

TEST RUN OUTPUT

DO YOU WANT DETAILED OUTPUT? YES NO [NO] (YES)
RESULT: ONLY WILL BE DISPLAYED
ENTER THE NUMBER OF OSCILLATOR QUANTA NEXT, PLEASE
OSCILLATOR QUANTA NUMBER = 3
... USE -2 TO 33 FOR N=10
... USE THE UP- FRONT REPRESENTATION (A, B, C, D, E, F, G, H, I, J, K)
... USE THE REAR REPRESENTATION (A, B, C, D, E, F, G, H, I, J, K)
... USE THE PATTERN NUMBER
... USE THE PATTERN NAME
... USE THE PATTERN DATA

<table>
<thead>
<tr>
<th>U1</th>
<th>U2</th>
<th>U3</th>
<th>U4</th>
<th>U5</th>
<th>U6</th>
<th>U7</th>
<th>U8</th>
<th>U9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

... USE THE PATTERN NAME
... USE THE PATTERN DATA

<table>
<thead>
<tr>
<th>U1</th>
<th>U2</th>
<th>U3</th>
<th>U4</th>
<th>U5</th>
<th>U6</th>
<th>U7</th>
<th>U8</th>
<th>U9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>