

> restart;

>

## Maple 12

The procedure W3j is derived from the C program described by William J. Thompson.

William J. Thompson, Angular Momentum: An Illustrated Guide to Rotational Symmetries for Physical Systems,  
John Wiley and Sons Inc., 1994

## Addition of Three Angular Momenta

Using Racah's formula to determine Wigner 3j coefficients

$$\begin{bmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{bmatrix} = (-1)^{j_1 - j_2 - m_3} \cdot N \cdot S$$

$$N = \sqrt{\frac{(j_3 + j_1 - j_2)! (j_3 - j_1 + j_2)! (j_1 + j_2 - j_3)! (j_3 - m_3)! (j_3 + m_3)!}{(j_1 + j_2 + j_3 + 1)! (j_1 - m_1)! (j_1 + m_1)! (j_2 - m_2)! (j_2 + m_2)!}}$$

$$S = \sum_{kmin}^{kmax} \frac{(-1)^{(kmin + j_2 + m_2)} (j_2 + j_3 + m_1 - k)! (j_1 - m_1 + k)!}{(k)! (j_3 - j_1 + j_2 - k)! (j_3 - m_3 - k)! (k + j_1 - j_2 + m_3)!}$$

where

kmin = maximum of [0 and j<sub>2</sub> - j<sub>1</sub> - m<sub>3</sub>] and kmax = minimum of [j<sub>3</sub> - j<sub>1</sub> + j<sub>2</sub> and j<sub>3</sub> - m<sub>3</sub>]

Clebsch-Gordan, C-G, Coefficients from Wigner 3j Coefficients

$$\langle j_1, j_2; m_1, m_2 | j_1, j_2; j_3, m_3 \rangle = (-1)^{-j_1 + j_2 - m_3} \sqrt{2j_3 + 1} \begin{bmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & -m_3 \end{bmatrix}$$

\*\*\*\* Note that we are using the following conventions \*\*\*\*

coefficient {a,b,c,m<sub>a</sub>,m<sub>b</sub>,m<sub>c</sub>} ≡ ⟨a,b;m<sub>a</sub>,m<sub>b</sub>|a,b;c,m<sub>c</sub>⟩

C-G value [±N] ≡ ±√N

This procedure determines the Wigner 3j coefficient using Racah's formula

```

> W3j :=proc(j1, j2, j3, m1, m2, m3)
    local k, kmin, kmax, N, S, P;

    kmin := max(0, j2 - j1 - m3);
    kmax := min( j3 - j1 + j2, j3 - m3);
    P := (-1)(j1 - j2 - m3);

    N :=  $\sqrt{\frac{(j3 + j1 - j2)!(j3 - j1 + j2)!(j1 + j2 - j3)!(j3 - m3)!(j3 + m3)!}{(j1 + j2 + j3 + 1)!(j1 - m1)!(j1 + m1)!(j2 - m2)!(j2 + m2)!}}$ ;

    S :=  $\sum_{k=kmin}^{kmax} \frac{(-1)^{(k + j2 + m2)} \cdot (j2 + j3 + m1 - k)!(j1 - m1 + k)!}{(k)!(j3 - j1 + j2 - k)!(j3 - m3 - k)!(k + j1 - j2 + m3)!}$ ;

    return (P·N·S);
end proc:

```

This procedure returns a signed squared coefficient.

$$\pm a\sqrt{n} = \pm a^2 n = \pm N$$

```

> SQ :=proc(n)
    local sign;
    if (evalf(n) < 0) then sign := -1 else sign := 1 end if;
    return (sign·simplify( (n·n) ));
end proc:

```

This procedure determines the value of the C-G coefficients; e.g.,  $\langle j1, j2; m1, m2 | j1, j2; j12, m12 \rangle$

$$\langle j1, j2; m1, m2 | j1, j2; j12, m12 \rangle = [\pm N]$$

$$[\pm N] \equiv \pm \sqrt{N}$$

```

> VCo :=proc(j1, j2, j3, j23, J, M)
    local m1, m2, m3, m23, c1, c2, s;
    s := " = ";
    for m2 from -j2 by 1 to j2 do
        for m3 from -j3 by 1 to j3 do
            for m23 from -j23 by 1 to j23 do
                for m1 from -j1 by 1 to j1 do
                    if (m2 + m3) = m23 and (m23 + m1) = M then # selection rule
                        c1 := (-1)(-j2 + j3 - m23) ·  $\sqrt{2 \cdot j23 + 1}$  · W3j(j2, j3, j23, m2, m3, -m23);
                        c2 := (-1)(-j23 + j1 - M) ·  $\sqrt{2 \cdot J + 1}$  · W3j(j23, j1, J, m23, m1, -M);
                        printf(" %s[%a]", s, SQ(c1·c2) );
                        printf("|%a,%a,%a;%a,%a,%a> \n", j2, j3, j1, m2, m3, m1);
                        s := " + ";
                    end if;
                end do;
            end do;
        end do;
    end do;
    print( );
end proc:

```

This procedure returns the product of the CG coefficients and the state  $|j_2, j_3, j_1; m_2, m_3, m_1\rangle$

$\{j_2, j_3, j_{23}, m_2, m_3, m_{23}\} \{j_{23}, j_1, J, m_{23}, m_1, M\} |j_2, j_3, j_1; m_2, m_3, m_1\rangle$

where  $j_{23} = j_2 + j_3$  and  $J = j_{23} + j_1$   
 $m_{23} = \{j_{23}, j_{23} - 1, \dots, -j_{23}\} = m_2 + m_3$   
 $M = \{J, J - 1, \dots, -J\} = m_{23} + m_1$

```
> SCo := proc(j1, j2, j3, j23, J, M)
    local m1, m2, m3, m23, c, s;
    s := " = ";
    printf(" |%a,%a,%a;%a,%a,%a>\n", j2, j3, j1, j23, J, M);
    for m2 from -j2 by 1 to j2 do
        for m3 from -j3 by 1 to j3 do
            for m23 from -j23 by 1 to j23 do
                for m1 from -j1 by 1 to j1 do
                    if (m2 + m3) = m23 and (m23 + m1) = M then # selection rule
                        printf("      %s{ %a,%a,%a,%a,%a,%a}", s, j2, j3, j23, m2, m3, m23 );
                        printf("{ %a,%a,%a,%a,%a,%a}", j23, j1, J, m23, m1, M);
                        printf(" |%a,%a,%a;%a,%a,%a> \n", j2, j3, j1, m2, m3, m1 );
                        s := " + ";
                    end if;
                end do;
            end do;
        end do;
    end do;
end proc;
```

Main procedure. Add3j adds three angular momenta - Add2j(j1,j2,j3)  
 using the coupling scheme  $j_2 + j_3 = j_{23}$  and  $j_{23} + j_1 = J$

$|j_2 - j_3| \leq j_{23} \leq (j_2 + j_3)$   
 $m_{23} = \{j_{23}, j_{23} - 1, j_{23} - 2, \dots, -j_{23}\}$

$|j_{23} - j_1| \leq J \leq (j_{23} + j_1)$   
 $M = \{J, J - 1, J - 2, \dots, -J\}$

```
> Add3j := proc(j1, j2, j3)
    local c, j23, J, M;
    printf("\n  There are %a |j2,j3,j1;j23,J,M> states where,\n\n", (2*j1 + 1) * (2*j2 + 1) * (2*j3
+ 1) )
    print(" |j2,j3,j1;j23,J,M> = \sum{j2,j3,j23,m2,m3,m23} {j23,j1,J,m23,m1,M} |j2,j3,j1,m2,m3,m1 >
");
    c := 0 : # counter
    for j23 from (j2 + j3) by -1 to |j2 - j3| do
        for J from (j23 + j1) by -1 to |j1 - j23| do
            for M from J by -1 to -J do
                c := c + 1 : printf("%a.", c);
                SCo(j1, j2, j3, j23, J, M);
                VCo(j1, j2, j3, j23, J, M);
            end do;
        end do;
    end do;
end proc;
```

### Example:

$$> \text{Add3j}\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right);$$

There are 8  $|j_2, j_3, j_1; j_{23}, J, M\rangle$  states where,

$$|j_2, j_3, j_1; j_{23}, J, M\rangle = \sum \{j_2, j_3, j_{23}, m_2, m_3, m_{23}\} \{j_{23}, j_1, J, m_{23}, m_1, M\} |j_2, j_3, j_1, m_2, m_3, m_1\rangle$$

$$1. |1/2, 1/2, 1/2; 1, 3/2, 3/2\rangle$$

$$= \{1/2, 1/2, 1, 1/2, 1/2, 1\} \{1, 1/2, 3/2, 1, 1/2, 3/2\} |1/2, 1/2, 1/2; 1/2, 1/2, 1/2\rangle \\ = [1] |1/2, 1/2, 1/2; 1/2, 1/2, 1/2\rangle$$

$$2. |1/2, 1/2, 1/2; 1, 3/2, 1/2\rangle$$

$$= \{1/2, 1/2, 1, -1/2, 1/2, 0\} \{1, 1/2, 3/2, 0, 1/2, 1/2\} |1/2, 1/2, 1/2; -1/2, 1/2, 1/2\rangle \\ + \{1/2, 1/2, 1, 1/2, -1/2, 0\} \{1, 1/2, 3/2, 0, 1/2, 1/2\} |1/2, 1/2, 1/2; 1/2, -1/2, 1/2\rangle \\ + \{1/2, 1/2, 1, 1/2, 1/2, 1\} \{1, 1/2, 3/2, 1, -1/2, 1/2\} |1/2, 1/2, 1/2; 1/2, 1/2, -1/2\rangle \\ = [1/3] |1/2, 1/2, 1/2; -1/2, 1/2, 1/2\rangle \\ + [1/3] |1/2, 1/2, 1/2; 1/2, -1/2, 1/2\rangle \\ + [1/3] |1/2, 1/2, 1/2; 1/2, 1/2, -1/2\rangle$$

$$3. |1/2, 1/2, 1/2; 1, 3/2, -1/2\rangle$$

$$= \{1/2, 1/2, 1, -1/2, -1/2, -1\} \{1, 1/2, 3/2, -1, 1/2, -1/2\} |1/2, 1/2, 1/2; -1/2, -1/2, 1/2\rangle \\ + \{1/2, 1/2, 1, -1/2, 1/2, 0\} \{1, 1/2, 3/2, 0, -1/2, -1/2\} |1/2, 1/2, 1/2; -1/2, 1/2, -1/2\rangle \\ + \{1/2, 1/2, 1, 1/2, -1/2, 0\} \{1, 1/2, 3/2, 0, -1/2, -1/2\} |1/2, 1/2, 1/2; 1/2, -1/2, -1/2\rangle \\ = [1/3] |1/2, 1/2, 1/2; -1/2, -1/2, 1/2\rangle \\ + [1/3] |1/2, 1/2, 1/2; -1/2, 1/2, -1/2\rangle \\ + [1/3] |1/2, 1/2, 1/2; 1/2, -1/2, -1/2\rangle$$

$$4. |1/2, 1/2, 1/2; 1, 3/2, -3/2\rangle$$

$$= \{1/2, 1/2, 1, -1/2, -1/2, -1\} \{1, 1/2, 3/2, -1, -1/2, -3/2\} |1/2, 1/2, 1/2; -1/2, -1/2, -1/2\rangle \\ = [1] |1/2, 1/2, 1/2; -1/2, -1/2, -1/2\rangle$$

$$5. |1/2, 1/2, 1/2; 1, 1/2, 1/2\rangle$$

$$= \{1/2, 1/2, 1, -1/2, 1/2, 0\} \{1, 1/2, 1/2, 0, 1/2, 1/2\} |1/2, 1/2, 1/2; -1/2, 1/2, 1/2\rangle \\ + \{1/2, 1/2, 1, 1/2, -1/2, 0\} \{1, 1/2, 1/2, 0, 1/2, 1/2\} |1/2, 1/2, 1/2; 1/2, -1/2, 1/2\rangle \\ + \{1/2, 1/2, 1, 1/2, 1/2, 1\} \{1, 1/2, 1/2, 1, -1/2, 1/2\} |1/2, 1/2, 1/2; 1/2, 1/2, -1/2\rangle \\ = [-1/6] |1/2, 1/2, 1/2; -1/2, 1/2, 1/2\rangle \\ + [-1/6] |1/2, 1/2, 1/2; 1/2, -1/2, 1/2\rangle \\ + [2/3] |1/2, 1/2, 1/2; 1/2, 1/2, -1/2\rangle$$

$$6. |1/2, 1/2, 1/2; 1, 1/2, -1/2\rangle$$

$$= \{1/2, 1/2, 1, -1/2, -1/2, -1\} \{1, 1/2, 1/2, -1, 1/2, -1/2\} |1/2, 1/2, 1/2; -1/2, -1/2, 1/2\rangle \\ + \{1/2, 1/2, 1, -1/2, 1/2, 0\} \{1, 1/2, 1/2, 0, -1/2, -1/2\} |1/2, 1/2, 1/2; -1/2, 1/2, -1/2\rangle \\ + \{1/2, 1/2, 1, 1/2, -1/2, 0\} \{1, 1/2, 1/2, 0, -1/2, -1/2\} |1/2, 1/2, 1/2; 1/2, -1/2, -1/2\rangle \\ = [-2/3] |1/2, 1/2, 1/2; -1/2, -1/2, 1/2\rangle \\ + [1/6] |1/2, 1/2, 1/2; -1/2, 1/2, -1/2\rangle \\ + [1/6] |1/2, 1/2, 1/2; 1/2, -1/2, -1/2\rangle$$

$$\begin{aligned}
7. & |1/2, 1/2, 1/2; 0, 1/2, 1/2\rangle \\
&= \{1/2, 1/2, 0, -1/2, 1/2, 0\} \{0, 1/2, 1/2, 0, 1/2, 1/2\} |1/2, 1/2, 1/2; -1/2, 1/2, 1/2\rangle \\
&+ \{1/2, 1/2, 0, 1/2, -1/2, 0\} \{0, 1/2, 1/2, 0, 1/2, 1/2\} |1/2, 1/2, 1/2; 1/2, -1/2, 1/2\rangle \\
&= [-1/2] |1/2, 1/2, 1/2; -1/2, 1/2, 1/2\rangle \\
&+ [1/2] |1/2, 1/2, 1/2; 1/2, -1/2, 1/2\rangle
\end{aligned}$$

$$\begin{aligned}
8. & |1/2, 1/2, 1/2; 0, 1/2, -1/2\rangle \\
&= \{1/2, 1/2, 0, -1/2, 1/2, 0\} \{0, 1/2, 1/2, 0, -1/2, -1/2\} |1/2, 1/2, 1/2; -1/2, 1/2, -1/2\rangle \\
&+ \{1/2, 1/2, 0, 1/2, -1/2, 0\} \{0, 1/2, 1/2, 0, -1/2, -1/2\} |1/2, 1/2, 1/2; 1/2, -1/2, -1/2\rangle \\
&= [-1/2] |1/2, 1/2, 1/2; -1/2, 1/2, -1/2\rangle \\
&+ [1/2] |1/2, 1/2, 1/2; 1/2, -1/2, -1/2\rangle
\end{aligned}$$

(1)

