CB–Note 298 (Preliminary)

Using *Mathematica* to Compute Helicity Amplitudes

Rod McCrady
Carnegie Mellon University

July 25, 1996

1 Introduction

When performing partial wave analyses, one typically must compute helicity amplitudes many thousands of times in one run of a fitting program. The program SPIN can perform such calculations for almost any decay chain, but its generality has a cost in terms of speed. It is common, therefore, to develop program modules for computing amplitudes for those specific decay chains which are being analyzed. It is necessary to obtain formulas for these specific cases in order to implement them in a programming language. The computer program *Mathematica* is well suited to this task. Its use in such situations is described in the following sections. Firstly, though, the equations to be used will be presented.

2 Helicity Amplitudes

Consider the decay $\alpha \to 1+2$, where α has spin J, particles 1 and 2 have spins s_1 and s_2 respectively, and particles 1 and 2 have relative orbital angular momentum L. Richman's equation 5.13 (see Reference [1]) gives the transition amplitude for the decay from magnetic substate M to the state with helicites λ_1 and λ_2 for particles 1 and 2, with particle 1 going in a direction defined by the angles θ_1, ϕ_1 relative to the quantization axis for particle α:

$$A(\alpha \to f) = \sqrt{\frac{2J+1}{4\pi}} \mathcal{D}^{J*}_{M,\lambda}(\phi_1, \theta_1, -\phi_1) A_{\lambda_1,\lambda_2} \quad \text{where} \quad \lambda = \lambda_1 - \lambda_2.$$

Taking the third angle in the $\mathcal{D}$ function to be $-\phi$ is a matter of convention; henceforth the third angle will be taken to be zero, with $\mathcal{D}^{J*}_{M,\lambda}(\theta, \phi)$ meaning $\mathcal{D}^{J*}_{M,\lambda}(\phi, \theta, -\phi)$. With the notation changed to a more convenient form, the amplitude is:

$$f^{\alpha}_{\lambda_1\lambda_2,M}(\theta_1, \phi_1) = \sqrt{\frac{2J+1}{4\pi}} \mathcal{D}^{J*}_{M,\lambda}(\theta_1, \phi_1) T^{\alpha\to 12}_{\lambda_1,\lambda_2}, \quad \text{where} \quad T^{\alpha\to 12}_{\lambda_1,\lambda_2} = \langle J;\lambda | L, s; 0, \lambda\rangle \langle s;\lambda | s_1, s_2; \lambda_1, -\lambda_2\rangle.$$

The angular distribution, assuming the final state helicities are not measured, is

$$W_M(\theta_1, \phi_1) = \sum_{\lambda_1,\lambda_2} \left| f^{\alpha}_{\lambda_1\lambda_2,M}(\theta_1, \phi_1) \right|^2$$

for the decay from magnetic substate M. If the various initial magnetic substates M are populated with probabilities P_M, then the total angular distribution is

$$W(\theta_1, \phi_1) = \sum_M P_M \sum_{\lambda_1, \lambda_2} f^{\alpha}_{\lambda_1 \lambda_2, M}(\theta_1, \phi_1) f^{*\alpha}_{\lambda_1 \lambda_2, M}(\theta_1, \phi_1).$$

These P_M are the diagonal elements of the density matrix ρ. If the initial state is described by a density matrix, then

$$W(\theta_1, \phi_1) = \sum_{M,\mu} \sum_{\lambda_1, \lambda_2} f^{\alpha}_{\lambda_1 \lambda_2, M}(\theta_1, \phi_1) \rho_{M,\mu} f^{*\alpha}_{\lambda_1 \lambda_2, \mu}(\theta_1, \phi_1) = \mathrm{Tr}(f^{\alpha} \rho f^{\alpha\dagger}).$$

Now consider the situation where one of the decay products of α decays:

$$\alpha \longrightarrow 1 + 2,\ 1 \longrightarrow 3 + 4.$$

The amplitude for this decay chain is the product of the amplitudes for the individual decays, summed over the unobservable helicity of particle 1:

$$f^{\alpha}_{\lambda_2 \lambda_3 \lambda_4, M}(\Omega) = \sum_{\lambda_1} f^{\alpha \to 12}_{\lambda_1 \lambda_2, M}(\theta_1, \phi_1) f^{1 \to 34}_{\lambda_3 \lambda_4, \lambda_1}(\theta_3, \phi_3)$$

that is,

$$f^{\alpha}_{\lambda_2 \lambda_3 \lambda_4, M} = \sum_{\lambda_1} \sqrt{\frac{2J+1}{4\pi}} T^{\alpha \to 12}_{\lambda_1, \lambda_2} \mathcal{D}^{J*}_{M, \lambda_1 - \lambda_2}(\theta_1, \phi_1) \sqrt{\frac{2s_1 + 1}{4\pi}} T^{1 \to 34}_{\lambda_3, \lambda_4} \mathcal{D}^{s_1 *}_{\lambda_1, \lambda_3 - \lambda_4}(\theta_3, \phi_3).$$

Take note of the fact that, for the decay of the particle α, $\lambda = \lambda_1 - \lambda_2$ and that the angles in the $\mathcal{D}$ function are those of particle 1. This correspondence between how the λ's are combined and which particle's angles are used is important, and will be adressed further in an example.

One can easily see that this can be interpreted as matrix multiplication and tensor products, as in the paper by Amsler and Bizot [2]. For use in computer programs for symbolic mathematics, though, it will be easier to leave the expressions for the amplitudes in the form of sums over products of $\mathcal{D}$ functions. *Mathematica* is such a program, and it has proven to be quite effective in computing simplified analytical expressions for helicity amplitudes for complicated decay chains. Examples of its use follow, beginning with outlines of simple decay chains.

3 Using *Mathematica* to Compute Helicity Amplitudes

Appendix A explains some aspects of *Mathematica* which are used in the following examples; those with no experience with the program may need to refer to that material while reading the following examples. Of particular interest will be the use of such built-in functions as `ClebschGordan[]`, `Table[]`, etc. In the first example, the essential elements concerning the use of *Mathematica* in this type of calculation are shown, without the clutter of details such as output from the program, how to load files, etc. Subsequent examples will present all of the details needed in order to carry out the calculation from beginning to end.

3.1 An Outline of the Procedure

Consider the process

$$\overline{p}p(^3S_1) \longrightarrow \rho^{\circ}\pi^{\circ},\ \rho^{\circ} \longrightarrow \pi^{+}\pi^{-}$$

which is addressed in section 2.3.1 of the paper by Amsler and Bizot [2]. The ρ° and π° must be in a P-wave, and the total spin of that system is $s = s_\rho = 1$. So

$$T^{\overline{p}p\to\rho\pi}_{\lambda_\rho,0} = \langle J_{\overline{p}p};\lambda|l,s;0,\lambda\rangle\langle s;\lambda|s_\rho,s_{\pi^\circ};\lambda_\rho,-\lambda_{\pi^\circ}\rangle = \langle 1;\lambda_\rho|1,1;0,\lambda_\rho\rangle\langle 1;\lambda_\rho|1,0;\lambda_\rho,0\rangle.$$

In *Mathematica*, it is useful to put these values in an array:

```
tpbp = ( Table[ ClebschGordan[ {1,0},    {1,lrho}, {1,lrho} ] *
                ClebschGordan[ {1,lrho}, {0,0},    {1,lrho} ],
                {lrho,1,-1,-1} ] )
```

This creates a *table* (in this case, a 1×3 array) containing the T values for $\lambda_\rho = +1, 0, -1$ in elements 1, 2, and 3, respectively (see Appendix A on the use of `Table[]` and `ClebschGordan`). In order to avoid indexing complications, defining a *function* to return $T^{\overline{p}p\to\rho\pi}_{\lambda_\rho,0}$ given λ_ρ is helpful:

```
tpbpf[ lrho_ ] := tpbp[[ 2-lrho ]]
```

The ρ decay produces two pions in a P-wave, so

$$T^{\rho\to\pi\pi}_{0,0} = \langle 1;0|1,0;0,0\rangle\langle 0;0|0,0;0,0\rangle = 1,$$

so no function needs to be defined for this. Then a table containing the helicity amplitudes for each magnetic substate of the $\overline{p}p$ system can be computed:

$$f_m = \sum_{\lambda_\rho=1,0,-1} \sqrt{\frac{3}{4\pi}} T^{\overline{p}p\to\rho\pi}_{\lambda_\rho} \mathcal{D}^{1*}_{m,\lambda_\rho}(\theta_\rho,\phi_\rho) \sqrt{\frac{3}{4\pi}} \mathcal{D}^{1*}_{\lambda_\rho,0}(\theta_{\pi^+},\phi_{\pi^+})$$

```
  amplitude = Table[ Sum[ tpbpf[ lrho ] * Ds[1,m,lrho, q,r] *
Sqrt[3/(4Pi)]
                                        * Ds[1,lrho,0, s,t] *
Sqrt[3/(4Pi)]
                         {lrho,1,-1,-1} ],
                    {m,1,-1,-1} ]
```

The amplitude is a matrix with one row per final state helicity and one column per initial state helicity. In this case, there is one row (no spin in the final state), and three columns (one for each magnetic substate of the initial state). This expression can then be simplified using the *Mathematica* built-in functions `Simplify[]`, `Expand[]`, `Factor[]`, `Collect[]`, etc., along with the package `Trigonometry`. The function `Ds[J,m,l,q,r]` $= \mathcal{D}^{J*}_{m,l}(q,r)$ must be defined by the user; a sample *Mathematica* file for doing so is shown in Appendix B.

To compute a formula for the angular distribution, one must form $f^\dagger_m$, the complex conjugate of the amplitude:

```
  astar = ( ( Conjugate[ amplitude ] //. {conjsum,conjprod} )
                                      /. {conjsin,conjcos } )
```

This statement takes the complex conjugate of each element of `amplitude`, then *recursively* applies the user-defined rules `conjsum` and `conjprod` (which define how to take complex conjugates of sums and products), then applies (once) the user-defined rules `conjsin` and `conjcos` (which define conjugates of sines and cosines of real variables.) These rules are shown in Appendix B. One can form a density matrix,

$$\rho = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & a \end{pmatrix},$$

with the statement

```
density = { {a,0,0}, {0,b,0}, {0,0,a} }
```

then form the product $W = f\rho f^\dagger$, which is the angular distribution:

```
ang_dist = amplitude . density . astar
```

The dot operator in this statement indicates matrix multiplication; *Mathematica* makes no distinction between row vectors and column vectors. To check the normalization, integrate over all of the angles:

$$N = \int_0^\pi \sin\theta_\rho d\theta_\rho \int_0^{2\pi} d\phi_\rho \int_0^\pi \sin\theta_{\pi+} d\theta_{\pi+} \int_0^{2\pi} d\phi_{\pi+} W(\theta_\rho, \phi_\rho, \theta_{\pi+}, \phi_{\pi+}) = 1$$

```
Integrate[ ang_dist Sin[q] Sin[s], {s,0,Pi}, {t,0,2Pi}, {u,0,Pi},
{v,0,2Pi} ]
```

The steps described above are the main steps involved in producing formulas for helicity amplitudes and angular distributions; some details have been omitted for the sake of clarity. Some illustrative examples are included below, which include all of the statements necessary to make *Mathematica* produce the desired output.

3.2 Complete Examples

Consider the decay chain:

$$\begin{array}{lllll} \overline{p}p(^1S_0) \longrightarrow & \hat{\rho}\pi & & & \\ & \hat{\rho} \longrightarrow & b_1\pi & & \\ & & b_1 \longrightarrow & \omega\pi & \\ & & & \omega \longrightarrow & \pi^\circ\gamma \end{array}$$

The $\hat{\rho}$ is a predicted hybrid meson with quantum numbers $I^G J^{PC} = 1^-1^{-+}$. Parity requires that the $\hat{\rho}$ and π be in an L=1 state, so the angular momentum coupling constants for the $\overline{p}p$ reaction, $T_{\lambda_{\hat{\rho}}\lambda_\pi}$, are created by the following statements:

```
Clear[ tpbp, tpbpf ]
tpbp = Table[ ClebschGordan[ {1,0},   {1,lrh}, {0,lrh} ] *
              ClebschGordan[ {1,lrh}, {0,0},   {1,lrh} ],
              {lrh,1,-1,-1} ]
tpbpf[ lrh_ ] := tpbp[[ 2-lrh ]]
```

The $\hat{\rho}$ can produce a $b_1\pi$ system with L=0 or L=2, so the function to return $T_{\lambda_{b_1}\lambda_\pi}$ is formed in such a way as to allow either possibility:

```
  Clear[ trhL0, trhL2, trhf, c, d ]
  trhL0 = Table[ ClebschGordan[ {0,0}, {1,lb1}, {1,lb1} ] *
                 ClebschGordan[ {1,lb1}, {0,0}, {1,lb1} ],
                 {lb1,1,-1,-1} ]
  trhL2 = Table[ ClebschGordan[ {2,0}, {1,lb1}, {1,lb1} ] *
                 ClebschGordan[ {1,lb1}, {0,0}, {1,lb1} ],
                 {lb1,1,-1,-1} ]
  trhf[ lb1_ ] := c /; (lb1 == 1) || (lb1 == -1)
  trhf[ lb1_ ] := d /;  lb1 == 0
```

The first three statements are not really necessary; they merely show the user that for L=0 all three constants are 1, and that for L=2 they are $\sqrt{1/10}, -\sqrt{2/5}, \sqrt{1/10}$ for $\lambda_{b_1} = 1, 0, -1$. This gives guidance in defining the function to allow for either state in the decay.

Similarly, the b_1 decay produces an $\omega\pi$ system with L=0 or L=2:

```
  Clear[ tb1L0, tb1L2, tb1f, a, b ]
  tb1L0 = Table[ ClebschGordan[ {0,0}, {1,lw}, {1,lw} ] *
                 ClebschGordan[ {1,lw}, {0,0}, {1,lw} ],
                 {lw,1,-1,-1} ]
  tb1L2 = Table[ ClebschGordan[ {2,0}, {1,lw}, {1,lw} ] *
                 ClebschGordan[ {1,lw}, {0,0}, {1,lw} ],
                 {lw,1,-1,-1} ]
  tb1f[ lw_ ] := a /; (lw == 1) || (lw == -1)
  tb1f[ lw_ ] := b /;  lw == 0
```

The π° and γ from the ω decay are in a P-wave, so the $T_{\lambda_\gamma\lambda_\pi}$'s are defined by the following:

```
  Clear[ tw, twf ]
  tw = Table[ ClebschGordan[ {1,0}, {1,lg}, {1,lg} ] *
              ClebschGordan[ {1,lg}, {0,0}, {1,lg} ],
                 {lg,1,-1,-1} ]
  twf[ lg_ ] := tw[[ 2-lg ]]
```

Now the full amplitude can be formed:

```
 Clear[ amplitude, Ds, q,r,s,t,u,v,y,z ]

 amplitude = Table[ Sum[ Sqrt[1/(4Pi)] * tpbpf[ lrh ] *
                         Sqrt[3/(4Pi)] * trhf[ lb1 ]  *
Ds[1,lrh,lb1,s,t] *
                         Sqrt[3/(4Pi)] * tb1f[ lw ]   * Ds[1,lb1,lw,
u,v] *
                         Sqrt[3/(4Pi)] * twf[ lg ]    * Ds[1,lw,lg,
```

```
y,z],
                        {lrh,1,-1,-1}, {lb1,1,-1,-1}, {lw,1,-1,-1} ],
                  {lg,1,-1,-1} ]
```

Since the initial state has $J = 0$, no $\mathcal{D}$ function is needed ($\mathcal{D}^0_{m,\mu}(\theta,\phi) = 1$). The angles are described in Table 1. Clearing the definition for `Ds[]` ensures that the sums and multiplications will be done in

Mathematica variables	Description
`s,t`	θ_{b_1}, ϕ_{b_1} in the rest frame of the $\hat{\rho}$
`u,v`	$\theta_\omega, \phi_\omega$ in the rest frame of the b_1
`y,z`	$\theta_\gamma, \phi_\gamma$ in the rest frame of the ω

Table 1: Definition of the angles used in the example.

terms of $\mathcal{D}^*$ functions instead of trigonometric functions, allowing this part of the calculation to be done quickly, with as few terms as possible. In order to get a simplified formula containing trigonometric functions, one must read in files containing definitions of the $\mathcal{D}^*$ functions and rules for simplifying such expressions:

```
<<Trigonometry.m
<<Dfunc.m
```

Now *Mathematica* can be instructed to simplify the expression for the amplitudes:

```
Clear[ a1, a2, a3, a4 ]
a1 = TrigReduce[ ComplexToTrig[ amplitude ]]
a2 = ExpandAll[ a1 ]
a3 = Simplify[ a2 ]
a4 = Factor[ a3 ]
```

This leaves a formula which is easy to implement in FORTRAN:

```
  { ( 3 ( a c Cos[u] Cos[v] Cos[y] Cos[z] Sin[s] +
          a d Cos[s] Cos[y] Cos[z] Sin[u] + I a c Cos[z] Sin[s] Sin[v]
+
          b d Cos[s] Cos[u] Sin[y] - b c Cos[v] Sin[s] Sin[u] Sin[y] +
          I a c Cos[u] Cos[v] Sin[s] Sin[z] + I a d Cos[s] Sin[u]
Sin[z] -
          a c Cos[y] Sin[s] Sin[v] Sin[z])) / (32 Pi^2),
    0,
    ( 3 ( a c Cos[u] Cos[v] Cos[y] Cos[z] Sin[s] +
          a d Cos[s] Cos[y] Cos[z] Sin[u] - I a c Cos[z] Sin[s] Sin[v]
+
          b d Cos[s] cos[u] Sin[y] - b c Cos[v] Sin[s] Sin[u] Sin[y] -
          I a c Cos[u] Cos[v] Sin[s] Sin[z] - I a d Cos[s] Sin[u]
Sin[z] -
          a c Cos[y] Sin[s] Sin[v] Sin[z])) / (32 Pi^2) \}
```

This is a 1×3 table, one element for each γ helicity. Of course, the amplitude is zero for $\lambda_\gamma = 0$. This just happens to work out in this case due to the Clebsch-Gordan coefficients for the ω decay, but one can set it to zero by hand if necessary, since it must be so from a physics perspective.

To check the normalization, one must read in rules for taking complex conjugates, then form the conjugate of the amplitude:

```
  Clear[ astar, asa, wt ] ;
  <<conj_rules.ma ;
  astar = (( Conjugate[ a4 ] //. {conjsum,conjprod} ) /.
{conjsin,conjcos} ) ;
```

The initial density matrix is just 1, since the initial state has $J = 0$, so $f\rho f^\dagger$ will be a 3×3 matrix. The angular distribution will be the trace of that matrix, which is just $\sum_{\lambda_\gamma} |f_{\lambda_\gamma}|^2$.

```
  asa = a1 astar ;
  wt = Sum[ asa[[2-lg]], {lg,1,-1,-1}] ;
  Integrate[ wt Sin[y] Sin[u] Sin[s],
             {Cos[q],1,-1}, {r,0,2Pi},
             {s,0,Pi},      {t,0,2Pi},
             {u,0,Pi},      {v,0,2Pi},
             {y,0,Pi},      {z,0,2Pi} ]
```

The answer is, of course, one. Note the not so subtle difference between table multiplication *with* a dot (matrix multiplication) and *without* a dot (element by element multiplication). Since the formula for the amplitude has no dependence on the angle `q`, the integration is more efficient if done over $d(\cos q)$.

In this example, all of the decays occur sequentially, that is, only one meson from each decay undergoes a subsequent decay. When *both* decay products of a resonance decay, one must be careful to use the correct angles and helicities in the $\mathcal{D}$ functions. This is illustrated by the following example.

Consider the decay chain:

$$\begin{aligned} \overline{p}p(^1S_0) &\longrightarrow a_2\pi \\ a_2 &\longrightarrow \rho\omega \\ \rho &\longrightarrow \pi\pi \\ \omega &\longrightarrow \pi^\circ\gamma \end{aligned}$$

Both decay products of the a_2 decay. The orbital angular momentum between the a_2 and π (from the $\overline{p}p$ reaction) is L=2, so the $T_{\lambda_{a_2}\lambda_\pi}$'s are defined by the following:

```
  Clear[ tpbp, tpbpf ]
  tpbp = Table[ ClebschGordan[ {2,0},  {2,la2}, {0,la2} ] *
                ClebschGordan[ {2,la2},{0,0},   {2,la2} ],
                {la2,2,-2,-1} ]
  tpbpf[ la2_ ] := tpbp[[ 3 - la2 ]]
```

For the a_2 decay products, L=0 and the total spin $s = 2$:

```
 Clear[ ta2, ta2f ]
```

```
 ta2 = Table[ ClebschGordan[ {0,0},   {2,lrho-lw},{2,lrho-lw} ] *
              ClebschGordan[ {1,lrho},{1,-lw},    {2,lrho-lw} ],
              {lrho,1,-1,-1}, {lw,1,-1,-1} ]
 ta2f[ lrho_, lw_ ] := ta2[[ 2-lrho, 2-lw ]]
```

The following defines the $T_{\lambda_\gamma \lambda_\pi}$'s:

```
  Clear[ tw, twf ]
  tw = Table[ ClebschGordan[ {1,0}, {1,lg}, {1,lg} ] *
              ClebschGordan[ {1,lg}, {0,0}, {1,lg} ],
                 {lg,1,-1,-1} ]
  twf[ lg_ ] := tw[[ 2-lg ]]
```

For the sake of illustration, the $T_{\lambda_\pi \lambda_\pi}$'s can be defined for the ρ decay:

```
  Clear[ trho, trhof ]
  trho = ClebschGordan[ {1,0}, {0,0}, {1,0} ] *
         ClebschGordan[ {0,0}, {0,0}, {0,0} ]
  trhof := trho
```

Now the amplitude is computed in terms of $\mathcal{D}$ functions:

```
  Clear[ amplitude, Ds, a1, a2, s,t,u,v,y,z ]
  amplitude =
   Table[ Sum[ Sqrt[1/(4Pi)] * tpbpf[ la2 ]
               Sqrt[5/(4Pi)] * ta2f[ lrho, lw ] * Ds[2, la2, lrho-lw,
s,t] *
               Sqrt[3/(4Pi)] * twf[ lg ]        * Ds[1,lw,   lg,
y,z] *
               Sqrt[3/(4Pi)] * trhof            * Ds[1,lrho, 0,
u,v],
               {la2,2,-2,-1}, {lrho,1,-1,-1}, {lw,1,-1,-1} ],
          {lg,1,-1,-1} ]
```

Note that λ for the a_2 decay is $\lambda_\rho - \lambda_\omega$, so the angles `s,t` are θ_ρ, ϕ_ρ, the angles of the ρ in the frame of the a_2. This definition λ was also used in the construction of $T_{\lambda_\rho \lambda_\omega}$. *When $\lambda = \lambda_A - \lambda_B$, use the angles of particle A in the $\mathcal{D}$ function.* The following commands simplify the result:

```
  <<Trigonometry.m
  <<Dfunc.m
  a1 = ExpandAll[ TrigReduce[ ComplexToTrig[ amplitude ]]]
  a2 = Simplify[ a1 ]
  a3 = Expand[ TrigReduce[ a2 ]]
  a4 = Factor[ a3 ]
```

For $\lambda_\gamma = 1$, the result is:

```
Sqrt[3/2] (3 Cos[s] Cos[u] Cos[y] Cos[z] Sin[s] -
           2 Cos[v] Cos[y] Cos[z] Sin[u] +
           3 Cos[s]^2 Cos[v] Cos[y] Cos[z] Sin[u] +
           I Cos[z] Sin[u] Sin[v] +
             Cos[u] Sin[y] -
           3 Cos[s]^2 Cos[u] Sin[y] +
           3 Cos[s] Cos[v] Sin[s] Sin[u] Sin[y] +
           3 I Cos[s] Cos[u] Sin[s] Sin[z] -
           2 I Cos[v] Sin[u] Sin[z] +
           3 I Cos[s]^2 Cos[v] Sin[u] Sin[z] -
             Cos[y] Sin[u] Sin[v] Sin[z]) / (32 Pi^2)
```

For $\lambda_\gamma = 0$, the amplitude is zero, and for $\lambda_\gamma = -1$, the amplitude is the complex conjugate of that for $\lambda_\gamma = 1$.

The following example has a $J = 1$ initial state. Consider the decay chain:

$$\begin{aligned} \overline{p}p(^3P_1) &\longrightarrow \hat{\rho}\pi \\ \hat{\rho} &\longrightarrow b_1\pi \\ b_1 &\longrightarrow \omega\pi \\ \omega &\longrightarrow \pi^\circ\gamma \end{aligned}$$

The $\hat{\rho}$ and π are produced in an S-wave, so $T_{\lambda_{\hat{\rho}}\lambda_\pi} = 1$ for all λ's. The $\hat{\rho}$ decay products can have $L = 0$ or 2, so $T_{\lambda_{b_1},0}$ will be formed as in the previous example involving this decay:

```
Clear[ trhf ]
trhf[ lb1_ ] := c /; (lb1 == 1) || (lb1 == -1)
trhf[ lb1_ ] := d /;  lb1 == 0
```

Similarly, the b_1 decay produces an $\omega\pi$ system with L=0 or L=2:

```
Clear[ tb1f ]
tb1f[ lw_ ] := a /; (lw == 1) || {lw == -1)
tb1f[ lw_ ] := b /;  lw == 0
```

The π° and γ from the ω decay are in a P-wave, so the $T_{\lambda_\gamma\lambda_\pi}$'s are defined by the following:

```
Clear[ tw, twf ]
tw = Table[ ClebschGordan[ {1,0}, {1,lg}, {1,lg} ] *
            ClebschGordan[ {1,lg}, {0,0}, {1,lg} ],
               {lg,1,-1,-1} ]
twf[ lw_ ] := tw[[ 2-lg ]]
```

Now the full amplitude can be formed:

```
  Clear[ amplitude, Ds, a1 ]
  amplitude = Table[ Sum[ Sqrt[3/(4Pi)] Ds[1, m,  lrh, q,r] *
                          Sqrt[3/(4Pi)] Ds[1, lrh,lb1, s,t] trhf[ lb1
] *
                          Sqrt[3/(4Pi)] Ds[1, lb1,lw,  u,v] tb1f[ lw ]
 *
                          Sqrt[3/(4Pi)] Ds[1, lw, lg,  y,z] twf[ lg ],
                          [lrh,1,-1,-1}, {lb1,1,-1,-1}, {lw,1,-1,-1}
],
                     {lg,1,-1,-1}, {m,1,-1,-1} ]
  <<Trigonometry.m
  <<Dfunc.m
  a1 = Factor[ Simplify[ ExpandAll[ TrigReduce[ ComplexToTrig[
amplitude ]]]]]
```

Note that the table of amplitudes has three rows (one for each helicity of the final state γ), and three columns (one for each $^3P_1$ magnetic substate). Seeing the result of this calculation (which is too long to show here) will heighten your appreciation for why one would use *Mathematica* instead of performing these calculations by hand.

4 Using SPIN to Test Helicity Code

After one has implemented the formulas for helicity amplitudes in a FORTRAN program, a convenient way to test them is to send identical events through that program and through SPIN (see Reference [4]). The following comments may be helpful in making such comparisons.

SPIN stores the angles which it uses in the common block JACKAN. These are the angles of the *second* daughter particle of the decaying resonance, so be sure to order the particles appropriately in the SPIN input file.

Specifying the width of a resonance as -1 will make SPIN calculate the helicity amplitudes only; the Breit-Wigner and centrifugal barrier functions will be set to 1. This allows one to compare computations of the helicity amplitudes without complications from kinematics. Remember that if the initial state decays with $L \neq 0$, a centrifugal barrier will be computed. It is important to set the width of the initial state to -1 when comparing helicity amplitudes.

SPIN does not include the constants $\sqrt{(2J+1)/4\pi}$ in its calculation, so the results from *Mathematica* will differ from those from SPIN by these multiplicative constants.

A Some *Mathematica* Hints

Reference [3] is the definitive source for information on *Mathematica*. The following hints may be useful in following the examples in the text, though.

To run *Mathematica* on Unix, just type `mathematica`. This uses an X interface, so issue the command `setenv DISPLAY ...` first.

Hit Shift-Return to execute a statement.

Putting a semicolon after a *Mathematica* statement suppresses the output.

Below are descriptions of some statements used in the text:

Table[*expr*,**{***i*,i_{first},i_{last},i_{step}**}**] Returns a table of *expr* evaluated at each value of the index i between i_{first} and i_{last} at intervals of i_{step}. Additional indexes can be used to generate tables of higher dimensions.

ClebschGordan[**{**j_1,m_1**},{**j_2,m_2**},{**J,M**}**] Returns Clebsch-Gordan coefficients with the same phase convention as is used in the Particle Data Book.

Sum[*expr*,**{***i*,i_{first},i_{last},i_{step}**}**] Returns the sum of *expr* with the index i running from i_{first} to i_{last} at intervals of i_{step}. Additional indexes can be used to evaluate a multiple sum.

Integrate[*expr*,**{***x*,x_{first},x_{last}**}**] Returns the integral of *expr* with respect to x over the interval from x_{first} to x_{last}. Additional integration variables can be used to evaluate a multiple integral.

tf[l_] := **t**[[**2** - l]] Defines a function **tf** which takes one argument, l, and returns the (2-l)th element of table **t**.

tf[l_] := **a** /; l == **1** || l == **-1** Defines a function **tf** which takes one argument, l, and returns **a** IF l is equal to 1 OR l is equal to -1.

<<*filename* Reads in a *Mathematica* file and evaluates each expression in it.

expr /. **{***rules***}** Applies the set of transformation rules *rules* to *expr*.

expr //. **{***rules***}** *Recursively* applies the set of transformation rules *rules* to *expr*.

B Some *Mathematica* Files

One can create a file containing rules for manipulation of expressions. The package **Trigonometry.m** is such a file which is provided with *Mathematica*. Two other files have been useful in simplifying expressions for helicity amplitudes. Below is a listing of the file **Dfunc.m**:

```
dfunc[J_,m1_,m2_,t_] :=
 ( Sum[
       Power[-1,n] Sqrt[Times[(J+m2)!,(J-m2)!,(J+m1)!,(J-m1)!]]
       /Times[(J-m1-n)!,(J+m2-n)!,(n+m1-m2)!,n!] *
       Power[Cos[t/2],(2 J+m2-m1-2 n)] *
       Power[-Sin[t/2],(m1-m2+2 n)],
       {n,0,2 J}] );

Dfunc[J_,m1_,m2_,t_,p_] := Exp[ -I m1 p ] dfunc[J,m1,m2,t];

Ds[J_,m1_,m2_,t_,p_]    := Exp[ I m1 p ] dfunc[J,m1,m2,t];
```

Notice that this definition of `dfunc[]` is that of Richman [1], and that it differs from that of Amsler and Bizot [2] in the order of the indices. Careful inspection of these two publications, though, will assure you that the two treatments produce identical results.

Below is a listing of the file **conj_rules.m**:

```
  conjprod = Conjugate[ x_ y_] -> Conjugate[x] Conjugate[y] ;

  conjsum  = Conjugate[ x_ + y_] -> Conjugate[x] + Conjugate[y] ;

  conjsin  = Conjugate[ Sin[ x_ ] ] -> Sin[ x ] ;

  conjcos  = Conjugate[ Cos[ x_ ] ] -> Cos[ x ] ;

  Clear[ Ds, Dfunc ] ;

  conjDs    = Conjugate[ Ds[j_,m1_,m2_,t_,p_] ] -> Dfunc[j,m1,m2,t,p]
;

  conjDfunc = Conjugate[ Dfunc[j_,m1_,m2_,t_,p_] ] -> Ds[j,m1,m2,t,p]
;
```

The rules for $\sin(x)$ and $\cos(x)$ are, of course, valid only for real arguments. A more complete set of macros is available via WWW from `http://www.phys.cmu.edu/cb/cb.html`.

References

[1] J. D. Richman. *An Experimenter's Guide to the Helicity Formalism.* Caltech report CALT-68-1148. 1984.

[2] C. Amsler and J. C. Bizot, *Computer Physics Communications*, 30 (1983) 21-30

[3] Stephen Wolfram. *Mathematica.* Addison Wesley, New York. 1991.

[4] C. Amsler, *Spin Simulation Program* SPIN. CB Note 153.