# Partial Wave Amplitude Calculator

*Paul M Eugenio*
*University of Massachusetts*
*31-Oct-97*

**This Mathematica package calculates amplitudes in the same spirit of the E852 code "gamp". The formulas follow that of S.U. Chungs's *Formulas for Partial Wave Analysis Ver. II* but differ in formula # 3.2, where both this package and the PWA code use 2 BIG D functions and not 1 D function and 1 d function.**

As of this date, calculation are limited to one-isobar decays such as
X-> rho pi, X-> omega pi , X->a2 eta , ...

## Blatt-Weisskopf centrifugal-barrier factors F[L,p]

**(See my note on generating barrier factors.)**

```
h1[n_,z_] := (-I)^(n+1)*Exp[I*z]/z*
    Sum[I^s/(s!(2*z)^s) (n +s)!/(n-s)! ,{s,0,n}]

NFactor[l_] := Switch[l, 0,1, 1,2, 2,13, 3,277, 4,12746,
    5,99881, 6,118394977]
F[l_,q_] :=
 Sqrt[NFactor[l]/(q^2 h1[l,q] ComplexExpand[Conjugate[h1[l,q]]])]
 Pr = 0.1973
```

*Out[445]=*

```
0.1973
```

*In[447]:=*

```
N[Simplify[F[2,x]]]
```

*Out[447]=*

```
                    4
                   x
3.60555 Sqrt[-----------------]
                       2    4
             9. + 3. x  + x
```

# D-functions

In[535]:=

```
d[l_,mp_,m_,theta_] := (-1)^(mp-m) *
    Sqrt[(l+mp)! (l-mp)! (l+m)! (l-m)!] *
    Sum[( (-1)^k Cos[theta/2]^(2l-(mp-m+2k)) *
        Sin[theta/2]^(mp-m+2k) )/( (l-mp-k)! (l+m-k)! *
        (mp-m+k)! k! ) ,{k,Max[0,m-mp],Min[l-mp,l+m]}]

Dfun[l_,mp_,m_,alpha_,beta_,gamma_] :=
    Exp[-I mp alpha] d[l,mp,m,beta] Exp[-I mp gamma]
Dfunc[l_,mp_,m_,alpha_,beta_,gamma_] :=
    Conjugate[Exp[-I mp alpha] d[l,mp,m,beta] Exp[-I mp gamma]]
Dfun2[l_,mp_,m_,phi_,theta_] :=
    Exp[-I mp phi] d[l,mp,m,theta]
Dfun2c[l_,mp_,m_,phi_,theta_] :=
    Conjugate[Exp[-I mp phi] d[l,mp,m,theta]]
```

Dfun2 is a 2 angle modified D function. The function exploits our choice of phase convention.

In[328]:=

```
N[Dfun2c[1,-1,0,0.650474*Pi,0.708039]]
```

Out[328]=

```
-0.209384 - 0.40943 I
```

In[4]:=

```
CG[l1_,m1_,l2_,m2_,l_,m_] :=ClebschGordan[{l1,m1},{l2,m2},{l,m}]
```

In[262]:=

```
N[CG[2,0,1,-1,1,-1]]
```

Out[262]=

```
0.316228
```

# Reflectivity Basis

*In[537]:=*

```
TH[m_] := Switch[m,1,1/Sqrt[2],0,1/2,-1,0]
```

*In[536]:=*

```
reDfunc[J_,m_,lam_,phi_,theta_,P_,eps_] :=
    TH[m]*( Dfun2c[J,m,lam,phi,theta] - eps P (-1)^(J-m) *
    Dfun2c[J,-m,lam,phi,theta])
```

# Decay Amplitude in the Reflectivity Basis

*In[538]:=*

```
rf_Amp[J_,m_,l_,s_,phi_,theta_,phih_,thetah_,P_,eps_] :=
    Sqrt[2*l+1]*Sqrt[2*s+1]*F[l,p/Pr]*F[s,q/Pr]*BW*
    Sum[reDfunc[J,m,lam,phi,theta,P,eps] *
    Dfun2c[s,lam,0,phih,thetah] CG[l,0,s,lam,J,lam],
    {lam,-s,s}]
```

For Omega decays remove the Barrier Factor F[s,q/Pr] and use the normalized lambda factor as the BW factor. For helicity angles use the normal to the omega decay plane as the analyzer.

**All inputs for the calculation are available in the gamp debug print out**

# Amplitude Calulations

**Add you inputs below and Shift-Enter to calculate.**

*In[508]:=*

```
J=1
P=+1
M=1
eps=-1
l=2
s=1
theta=2.77
phi=0.69*Pi
thetah=0.708
phih=0.65*Pi
BW=0.1861
p=0.0641353
q= 0.002
Print[Amplitude is]
N[reAmp[J,M,l,s,phi,theta,phih,thetah,P,eps]]
J=.
P=.
M=.
eps=.
l=.
s=.
theta=.
phi=.
thetah=.
phih=.
BW=.
p=.
```

*Out[508]=*

```
1
```

*Out[509]=*

```
1
```

*Out[510]=*

```
1
```

*Out[511]=*

```
-1
```

*Out[512]=*

```
2
```

*Out[513]=*

```
1
```

*Out[514]=*

```
2.77
```

*Out[515]=*

```
0.69 Pi
```

*Out[516]=*

```
0.708
```

*Out[517]=*

```
0.65 Pi
```

*Out[518]=*

```
0.1861
```

*Out[519]=*

```
0.0641353
```

*Out[520]=*

```
0.002
```

```
Amplitude is
```

*Out[522]=*

$1.78863\ 10^{-20} + 0.000225975\ I$