Dalitz analysis in Bch→D(*)^{0}K decays via K-matrix approach

Ben Lau

Princeton University

BaBar Collaboration Meeting, 22 May 2005

Dalitz plot via K-matrix

History of the Analysis:

1. March 2004 : A first attempt to $D^0 \rightarrow$Ks pi+ pi- in breco group.
2. August 2004: announced the dalitz plot fit using Isobar Model (sum of BW's)

   (See: BL talk: http://www.slac.stanford.edu/BFROOT/www/Organization/CollabMtgs/2004/detJul04/Fri1/yanpan.pdf

3. October 2004: Start the K-matrix fit attempt to solve σ(500) and σ(1000)
4. Feb 2005: Preliminary fit using K-matrix is announced.

   (See: LL talk: http://www.slac.stanford.edu/BFROOT/www/Organization/CollabMtgs/2005/detFeb05/Wed4a/Ligioi.pdf

5. May 2005: Updated fit results(this talk!)

K-matrix for dummies

We start from first principle: S-matrix respect Unitary:
(unitary means probality conservations)

$$SS^{\dagger} = I$$ (equation 1: unitary of S-matrix)

$$S = I + 2iT$$ (equation2: rewrite S-matrix as T-matrix)

$$(T^{-1} + iI)^{\dagger} = (T^{-1} + iI)$$ (equation3: subsitute 2 in 1)

Now I define K:

Equation4: Most important equation

$$K^{-1} = T^{-1} + iI$$

Notice: (K and T is now related)

→ Unitary condition will be preserved.

From (4): re-write T as a function of K

$$T = K \bullet (I - iK)^{-1}$$

(the T-matrix: The one you observed from the experiment!)

Proof of the Breit-Wigner in K-matrix

One of the easiest way to write down the K-matrix is just sum of the poles…

Start from K:

$$K_{ij}(s) = \sum_a \frac{g_i g_j}{m_a^2 - s}$$

(sum of the K-matrix pole)

K-matrix diverge when $m_a = s$

If we have one single resonance →1 pole is needed.

$$K_{11} = \frac{m\Gamma}{m_a^2 - s}$$

where $g^2 = m\Gamma$

$$1 - iK = \frac{(m_a^2 - s) - im\Gamma}{(m_a^2 - s)}$$

$$\rightarrow K \bullet (I - iK)^{-1} = \frac{m\Gamma}{(m_a^2 - s)} \cdot \frac{(m_a^2 - s)}{(m_a^2 - s) - im\Gamma}$$

$$\rightarrow T = \frac{m\Gamma}{(m_a^2 - s) - im\Gamma}$$

Since:

$$T = K \bullet (I - iK)^{-1}$$

Breit-Wigner formula

Flatte's Formula

K matrix can be generalized to coupled channel decay, eg:

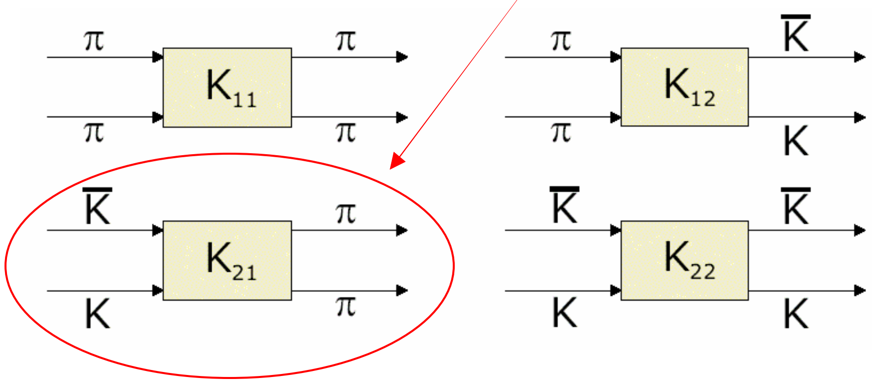
$$f_0(980) \rightarrow pp\,, KK$$

Rescattering is NOT
included in sum of BW's model

Consider 2 by 2 matrices:

$$K = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}$$

In picture:



Follow the pervious exercise, one
can derive the famous Flatte's Formula:

$$T = \frac{\begin{pmatrix} g_1^2 & g_1 g_2 \\ g_1 g_2 & g_2^2 \end{pmatrix}}{m_a^2 - s - im\Gamma(r_1 g_1^2 + r_2 g_2^2)}$$

Proof: LASS parameterization is equivalent as K-matrix

- Now consider K-matrix with background, I can write:

$$K = Tan(d_r + d_b)$$

where the delta_b is the background phase

Since: $$T = K \bullet (I - iK)^{-1}$$

We got: $$T = \sin(d_b)e^{id_b} + e^{2id_b}\sin(d_r)e^{id_r}$$

Exactly the LASS parameterization

Don't believe me? Let's check it in Mathematica:

$T = K \bullet (I - iK)^{-1}$

In[2]:= $\frac{\texttt{Tan[b + r]}}{\texttt{1 - I * Tan[b + r]}} - \left(e^{i\,b}\,\texttt{Sin[b]} + e^{2\,i\,b + i\,r}\,\texttt{Sin[r]}\right)$ `// FullSimplify`

LASS parameterization

Out[2]= 0

They are equivalent parameterization!

hep-ph/0312040

ππ S-wave parameterization

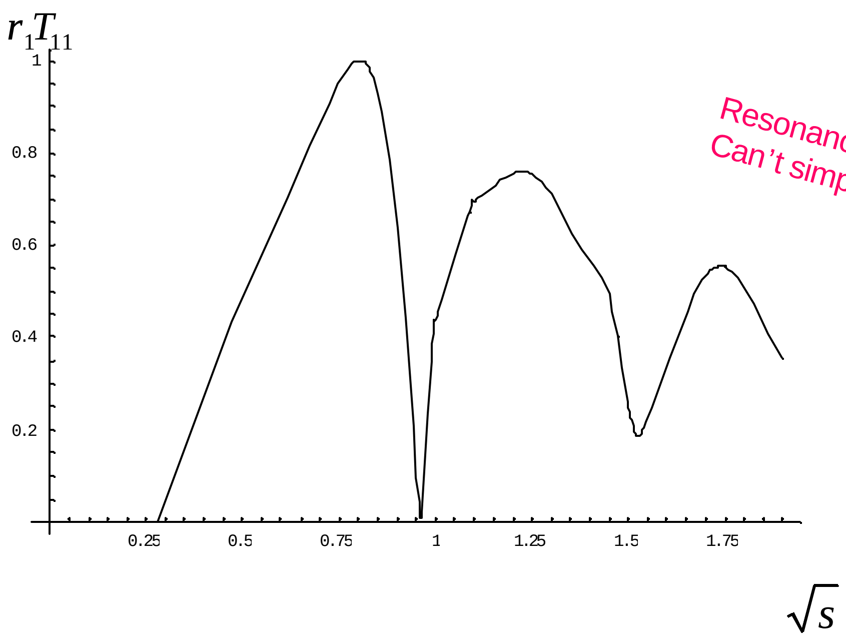
$$K_{ij}^{00}(s)=\left(\sum_a \frac{g_i^{(a)}g_j^{(a)}}{m^2{}_a - s} + f_{ij}^{scatt}\frac{1-s_0^{scatt}}{s-s_0^{scatt}}\right)\frac{s-s_A m_p^2/2}{(s-s_{A0})(1-s_{A0})}$$

pole term

slow varying function (background term)

Adler zero term. (suppressed the false kinematics singularity)

Intensity plot

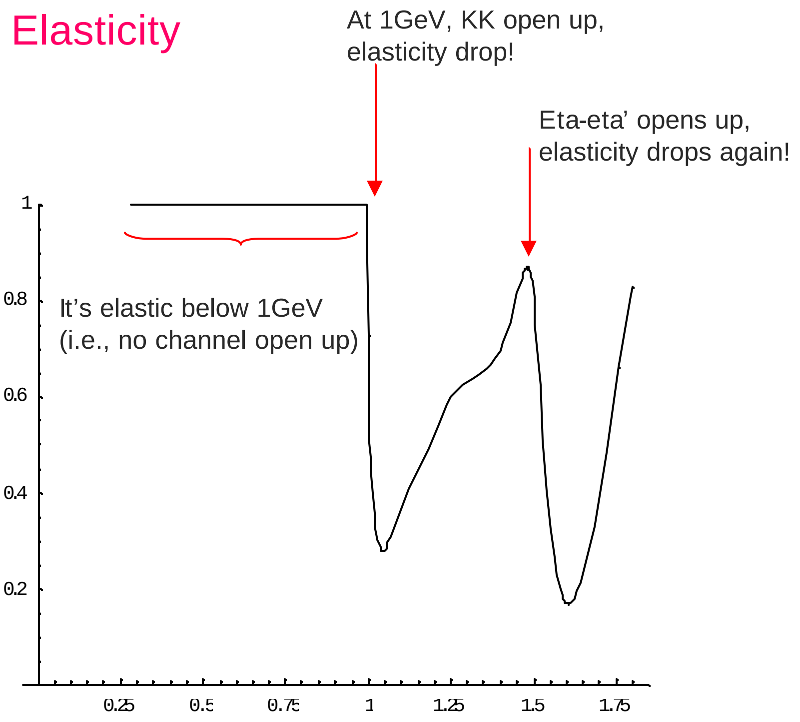


Resonances are heavily overlapped.
Can't simply sum BW 's

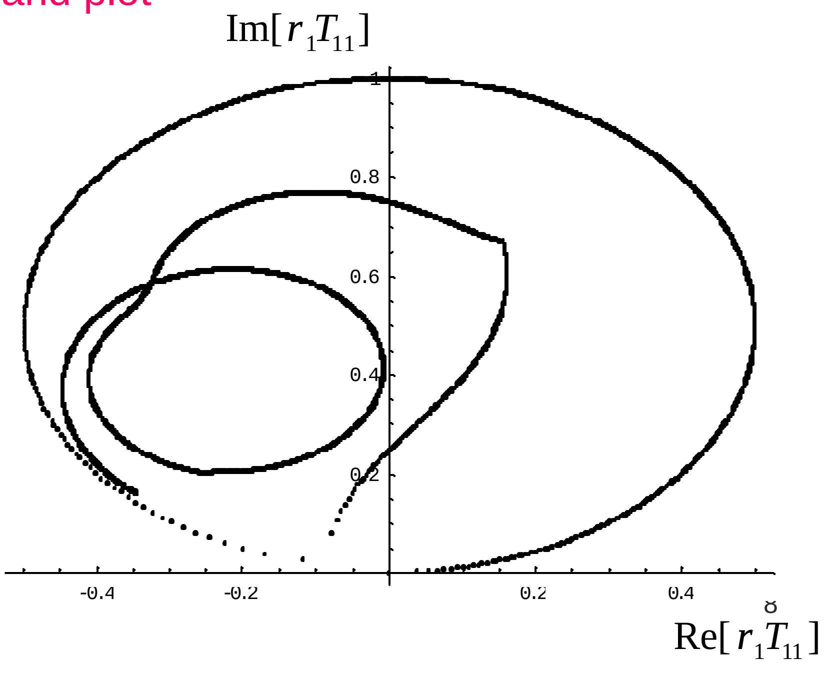
ππ S-wave parameterization

$$T = (I - iK \cdot r)^{-1} K$$

Elasticity



Argand plot



K-matrix parametization

- The S-wave are dominated by the board, overlapped resonance.

SUM OF THE BW's MODEL in ICHEP 04: (BAD 899)

Isobar Model: $$\sum_j a_j e^{iq_j} \times^J \mathrm{M}_r \times BW^r + a_0 e^{iq_0}$$

--Angular Dependence (spin 0,1,2)

--Relativistic Breit-Wigner Form Factor

Non-resonance term

K-matrix Model: $$A(D) = a_0 e^{id_0} + \sum_{i=1}^{m} a_i e^{id_i} F_i^{BW} + \sum_{i=m+1}^{n} a_i e^{id_i} F_i^{K}$$

Sum of BW's for Spin1 and 2

Sum of K-matrix Pole for Spin 0

K matrix is highly non-trivial, here I show part of the code to compute the Inverse of K-matrix: (>400 line of code for K-matrix only)

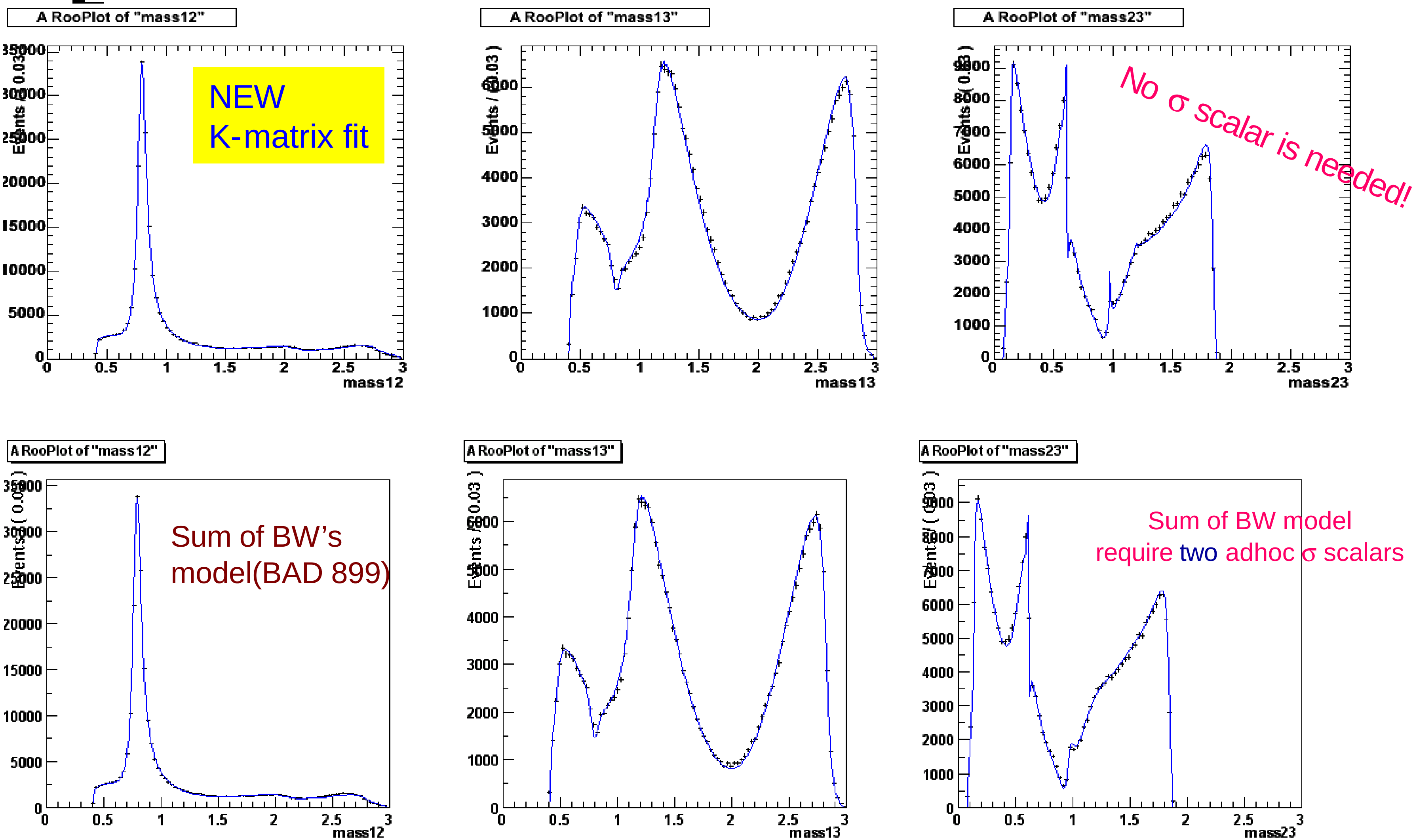
→Computation very challenging. (A single fit can take >24 hours without optimization)

```
//Compute this crazy determinent
 det = (n15*n24*n33*n42*n51 - n14*n25*n33*n42*n51 - n15*n23*n34*n42*n51 +
  n13*n25*n34*n42*n51 + n14*n23*n35*n42*n51 - n13*n24*n35*n42*n51 -
  n15*n24*n32*n43*n51 + n14*n25*n32*n43*n51 + n15*n22*n34*n43*n51 -
  n12*n25*n34*n43*n51 - n14*n22*n35*n43*n51 + n12*n24*n35*n43*n51 +
  n15*n23*n32*n44*n51 - n13*n25*n32*n44*n51 - n15*n22*n33*n44*n51 +
  n12*n25*n33*n44*n51 + n13*n22*n35*n44*n51 - n12*n23*n35*n44*n51 -
  n14*n23*n32*n45*n51 + n13*n24*n32*n45*n51 + n14*n22*n33*n45*n51 -
  n12*n24*n33*n45*n51 - n13*n22*n34*n45*n51 + n12*n23*n34*n45*n51 -
  n15*n24*n33*n41*n52 + n14*n25*n33*n41*n52 + n15*n23*n34*n41*n52 -
  n13*n25*n34*n41*n52 - n14*n23*n35*n41*n52 + n13*n24*n35*n41*n52 +
  n15*n24*n31*n43*n52 - n14*n25*n31*n43*n52 - n15*n21*n34*n43*n52 +
  n11*n25*n34*n43*n52 + n14*n21*n35*n43*n52 - n11*n24*n35*n43*n52 -
  n15*n23*n31*n44*n52 + n13*n25*n31*n44*n52 + n15*n21*n33*n44*n52 -
  n11*n25*n33*n44*n52 - n13*n21*n35*n44*n52 + n11*n23*n35*n44*n52 +
  n14*n23*n31*n45*n52 - n13*n24*n31*n45*n52 - n14*n21*n33*n45*n52 +
  n11*n24*n33*n45*n52 + n13*n21*n34*n45*n52 - n11*n23*n34*n45*n52 +
  n15*n24*n32*n41*n53 - n14*n25*n32*n41*n53 - n15*n22*n34*n41*n53 +
  n12*n25*n34*n41*n53 + n14*n22*n35*n41*n53 - n12*n24*n35*n41*n53 -
  n15*n24*n31*n42*n53 + n14*n25*n31*n42*n53 + n15*n21*n34*n42*n53 -
  n11*n25*n34*n42*n53 - n14*n21*n35*n42*n53 + n11*n24*n35*n42*n53 +
  n15*n22*n31*n44*n53 - n12*n25*n31*n44*n53 - n15*n21*n32*n44*n53 +
  n11*n25*n32*n44*n53 + n12*n21*n35*n44*n53 - n11*n22*n35*n44*n53 -
  n14*n22*n31*n45*n53 + n12*n24*n31*n45*n53 + n14*n21*n32*n45*n53 -
  n11*n24*n32*n45*n53 - n12*n21*n34*n45*n53 + n11*n22*n34*n45*n53 -
  n15*n23*n32*n41*n54 + n13*n25*n32*n41*n54 + n15*n22*n33*n41*n54 -
  n12*n25*n33*n41*n54 - n13*n22*n35*n41*n54 + n12*n23*n35*n41*n54 +
  n15*n23*n31*n42*n54 - n13*n25*n31*n42*n54 - n15*n21*n33*n42*n54 +
  n11*n25*n33*n42*n54 + n13*n21*n35*n42*n54 - n11*n23*n35*n42*n54 -
  n15*n22*n31*n43*n54 + n12*n25*n31*n43*n54 + n15*n21*n32*n43*n54 -
  n11*n25*n32*n43*n54 - n12*n21*n35*n43*n54 + n11*n22*n35*n43*n54 +
  n13*n22*n31*n45*n54 - n12*n23*n31*n45*n54 - n13*n21*n32*n45*n54 +
  n11*n23*n32*n45*n54 + n12*n21*n33*n45*n54 - n11*n22*n33*n45*n54 +
  n14*n23*n32*n41*n55 - n13*n24*n32*n41*n55 - n14*n22*n33*n41*n55 +
  n12*n24*n33*n41*n55 + n13*n22*n34*n41*n55 - n12*n23*n34*n41*n55 -
  n14*n23*n31*n42*n55 + n13*n24*n31*n42*n55 + n14*n21*n33*n42*n55 -
  n11*n24*n33*n42*n55 - n13*n21*n34*n42*n55 + n11*n23*n34*n42*n55 +
  n14*n22*n31*n43*n55 - n12*n24*n31*n43*n55 - n14*n21*n32*n43*n55 +
  n11*n24*n32*n43*n55 + n12*n21*n34*n43*n55 - n11*n22*n34*n43*n55 -
  n13*n22*n31*n44*n55 + n12*n23*n31*n44*n55 + n13*n21*n32*n44*n55 -
  n11*n23*n32*n44*n55 - n12*n21*n33*n44*n55 + n11*n22*n33*n44*n55);
```

```
//The 1st row of the inverse natrix:
 i[1][1] = (n25*n34*n43*n52 -
     n24*n35*n43*n52 - n25*n33*n44*n52 + n23*n35*n44*n52 +
     n24*n33*n45*n52 - n23*n34*n45*n52 - n25*n34*n42*n53 +
     n24*n35*n42*n53 + n25*n32*n44*n53 - n22*n35*n44*n53 -
     n24*n32*n45*n53 + n22*n34*n45*n53 + n25*n33*n42*n54 -
     n23*n35*n42*n54 - n25*n32*n43*n54 + n22*n35*n43*n54 +
     n23*n32*n45*n54 - n22*n33*n45*n54 - n24*n33*n42*n55 +
     n23*n34*n42*n55 + n24*n32*n43*n55 - n22*n34*n43*n55 -
     n23*n32*n44*n55 + n22*n33*n44*n55)/det;

 i[1][2] = ( -n15*n34*n43*n52 +
     n14*n35*n43*n52 + n15*n33*n44*n52 - n13*n35*n44*n52 -
     n14*n33*n45*n52 + n13*n34*n45*n52 + n15*n34*n42*n53 -
     n14*n35*n42*n53 - n15*n32*n44*n53 + n12*n35*n44*n53 +
     n14*n32*n45*n53 - n12*n34*n45*n53 - n15*n33*n42*n54 +
     n13*n35*n42*n54 + n15*n32*n43*n54 - n12*n35*n43*n54 -
     n13*n32*n45*n54 + n12*n33*n45*n54 + n14*n33*n42*n55 -
     n13*n34*n42*n55 - n14*n32*n43*n55 + n12*n34*n43*n55 +
     n13*n32*n44*n55 - n12*n33*n44*n55)/det;

 i[1][3] = (n15*n24*n43*n52 -
     n14*n25*n43*n52 - n15*n23*n44*n52 + n13*n25*n44*n52 +
     n14*n23*n45*n52 - n13*n24*n45*n52 - n15*n24*n42*n53 +
     n14*n25*n42*n53 + n15*n22*n44*n53 - n12*n25*n44*n53 -
     n14*n22*n45*n53 + n12*n24*n45*n53 + n15*n23*n42*n54 -
     n13*n25*n42*n54 - n15*n22*n43*n54 + n12*n25*n43*n54 +
     n13*n22*n45*n54 - n12*n23*n45*n54 - n14*n23*n42*n55 +
     n13*n24*n42*n55 + n14*n22*n43*n55 - n12*n24*n43*n55 -
     n13*n22*n44*n55 + n12*n23*n44*n55)/det;

 i[1][4] = ( -n15*n24*n33*n52 +
     n14*n25*n33*n52 + n15*n23*n34*n52 - n13*n25*n34*n52 -
     n14*n23*n35*n52 + n13*n24*n35*n52 + n15*n24*n32*n53 -
     n14*n25*n32*n53 - n15*n22*n34*n53 + n12*n25*n34*n53 +
     n14*n22*n35*n53 - n12*n24*n35*n53 - n15*n23*n32*n54 +
     n13*n25*n32*n54 + n15*n22*n33*n54 - n12*n25*n33*n54 -
     n13*n22*n35*n54 + n12*n23*n35*n54 + n14*n23*n32*n55 -
     n13*n24*n32*n55 - n14*n22*n33*n55 + n12*n24*n33*n55 +
     n13*n22*n34*n55 - n12*n23*n34*n55)/det;

 i[1][5] = (n15*n24*n33*n42 -
     n14*n25*n33*n42 - n15*n23*n34*n42 + n13*n25*n34*n42 +
     n14*n23*n35*n42 - n13*n24*n35*n42 - n15*n24*n32*n43 +
     n14*n25*n32*n43 + n15*n22*n34*n43 - n12*n25*n34*n43 -
     n14*n22*n35*n43 + n12*n24*n35*n43 + n15*n23*n32*n44 -
     n13*n25*n32*n44 - n15*n22*n33*n44 + n12*n25*n33*n44 +
     n13*n22*n35*n44 - n12*n23*n35*n44 - n14*n23*n32*n45 +
     n13*n24*n32*n45 + n14*n22*n33*n45 - n12*n24*n33*n45 -
     n13*n22*n34*n45 + n12*n23*n34*n45)/det;
```

Fit Results!

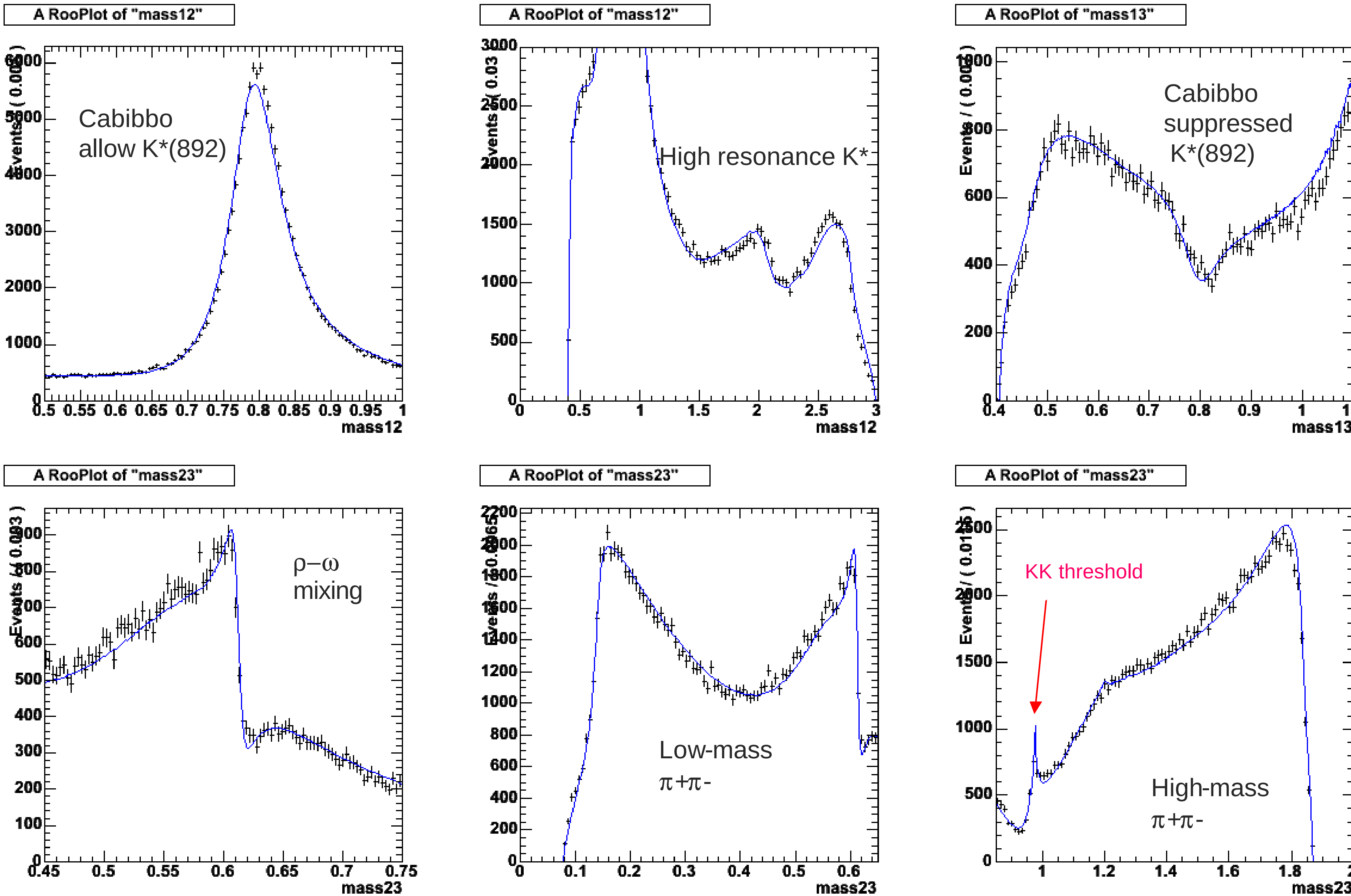
253991 events!



253991 events

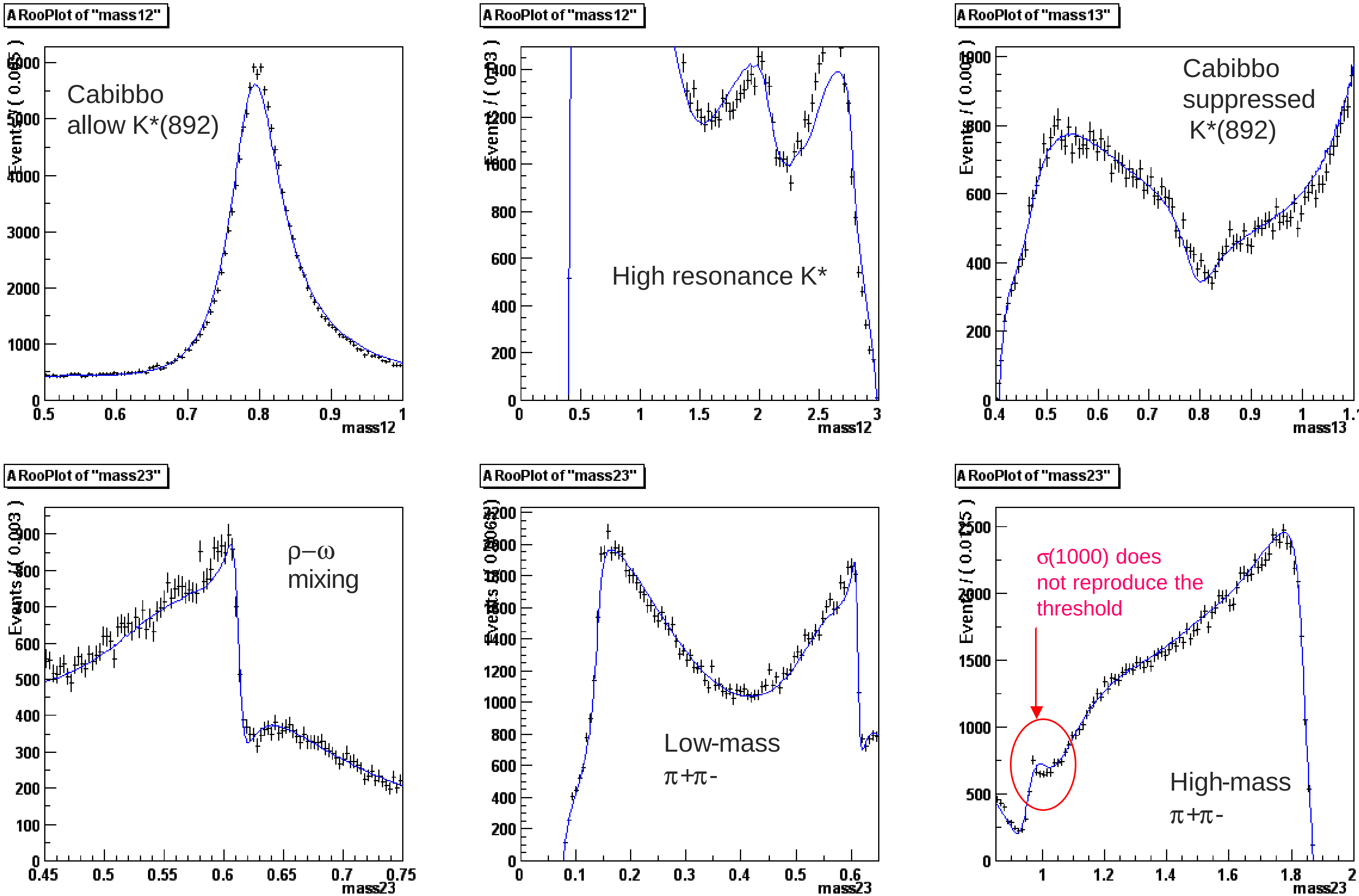
BaBar NEW K-matrix model

still preliminary!



250000 events

Traditional IsoBar model



Fit result:

Floating Parameter	InitialValue	FinalValue +/- Error	GblCorr.
K2*1430_DCS_amp	1.2022e-01	1.6209e-01 +/- 1.83e-02	0.786938
K2*1430_DCS_phase	6.4537e+02	6.4892e+02 +/- 6.78e+00	0.895225
K2*1430_amp	1.1370e+00	1.1446e+00 +/- 1.96e-02	0.891549
K2*1430_phase	-4.4972e+01	-4.6090e+01 +/- 1.20e+00	0.891217
Kst1430_DCS_amp	5.6307e-01	4.3214e-01 +/- 2.61e-02	0.895199
Kst1430_DCS_phase	-3.8426e+02	-3.7339e+02 +/- 5.02e+00	0.945632
Kst1430_amp	2.8573e+00	2.8636e+00 +/- 3.03e-02	0.869751
Kst1430_phase	-3.6636e+02	-3.6864e+02 +/- 7.51e-01	0.936726
Kst1680_amp	1.0790e+00	1.1924e+00 +/- 5.51e-02	0.937881
Kst1680_phase	-2.1716e+02	-2.2152e+02 +/- 3.32e+00	0.925722
Kstminus_amp	1.7883e+00	1.7678e+00 +/- 7.19e-03	0.842941
Kstminus_phase	1.3105e+02	1.3102e+02 +/- 3.94e-01	0.796268
Kstplus_amp	1.6603e-01	1.7034e-01 +/- 4.18e-03	0.463510
Kstplus_phase	-4.2842e+01	-4.4859e+01 +/- 1.32e+00	0.576553
beta1_amp	4.3949e+00	4.1525e+00 +/- 5.63e-02	0.952798
beta1_phase	-9.3074e+02	-9.3187e+02 +/- 7.81e-01	0.945931
beta2_amp	9.7971e+00	9.5410e+00 +/- 7.48e-02	0.926354
beta2_phase	1.4491e+01	1.3717e+01 +/- 1.18e+00	0.981180
beta4_amp	1.1707e+01	1.0897e+01 +/- 3.21e-01	0.956714
beta4_phase	-5.2635e+00	-5.2790e+00 +/- 1.45e+00	0.929301
f2_1270_amp	9.0673e-01	8.9453e-01 +/- 1.45e-02	0.883170
f2_1270_phase	-7.3457e+02	-7.3663e+02 +/- 1.46e+00	0.938541
fprod1_amp	1.1824e+01	1.1167e+01 +/- 1.42e-01	0.964835
fprod1_phase	-1.4818e+02	-1.4867e+02 +/- 9.31e-01	0.984848
omega782_amp	4.1584e-02	4.3944e-02 +/- 9.14e-04	0.385270
omega782_phase	8.3092e+02	8.3509e+02 +/- 1.25e+00	0.568867

We now print out the fit fraction

The fit fraction for Rho is 21.55010989
The fit fraction for K*(892) is 58.80443747
The fit fraction for K*(892)_DCS is 0.5727592137
The fit fraction for omega782 is 0.6598806402
The fit fraction for f2_1270 is 2.701877159
The fit fraction for Kst1430 is 10.32969833
The fit fraction for K2*1430 is 3.005658518
The fit fraction for Kst1430_DCS is 0.2368047567
The fit fraction for K2*1430_DCS is 0.05997408577
The fit fraction for Kst1680 is 0.6039532434
The fit fraction for S-wave is 16.0437926

The total fit fraction is 1.145689459

The total fit fraction is 115%, which is even smaller compared with the sum of BW's model (124%)

→GOOD!

Notice: only 26 parameters floating in K-matrix model , while 38 parameters in the Sum of BW's model(BAD 899)

→A more confident fit!

New method: Looking at the moments!

We can also perform Partial Wave expansion on the matrix elements

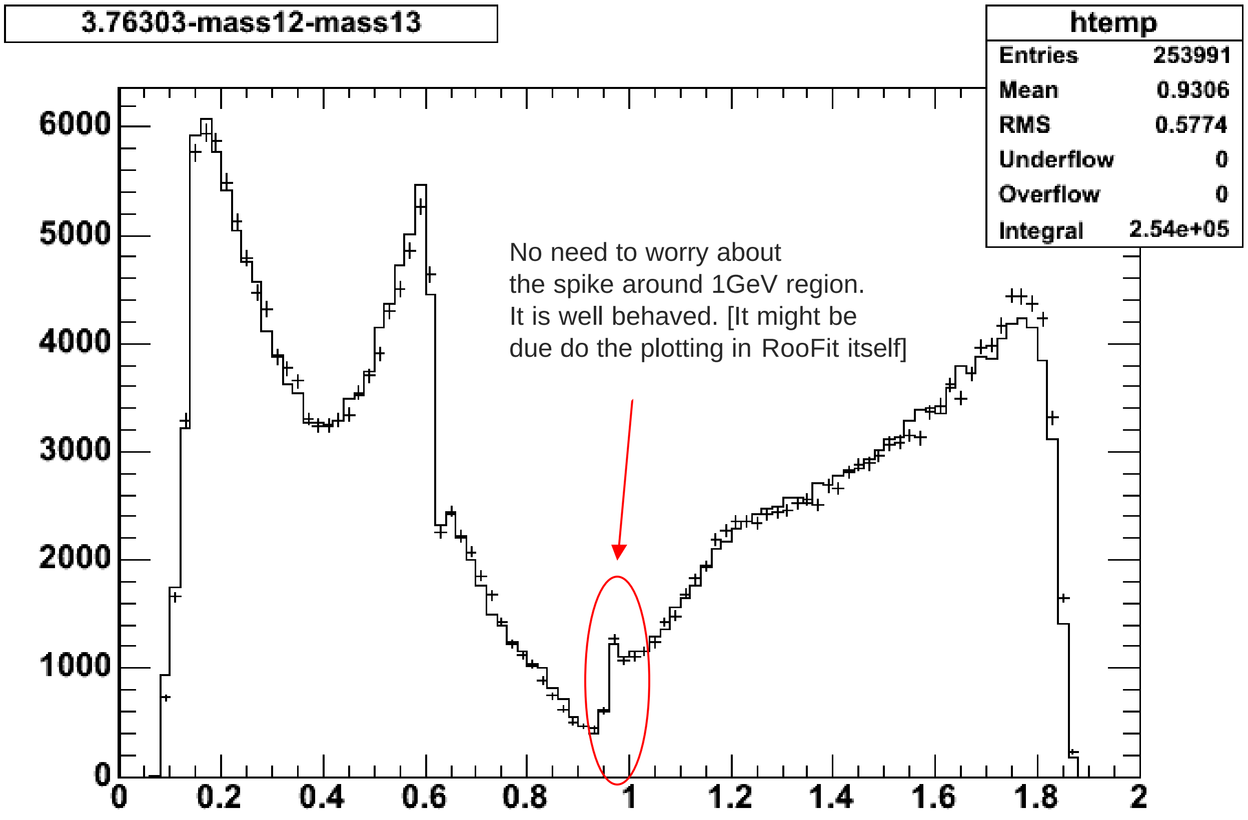
$$M(m_{12}, m_{13}) = \sum_l R(m_{23}) P_l(\cos q_{23})$$

Since the Legendre Polynormals are orthogonal,
One can plot the moments and Examine the quality of the fit.

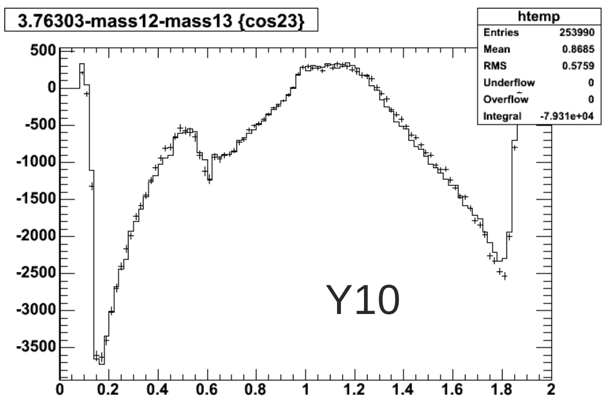
The model are generated from the ToyMC using 250K MC events and the data are plot on top for direct comparison.

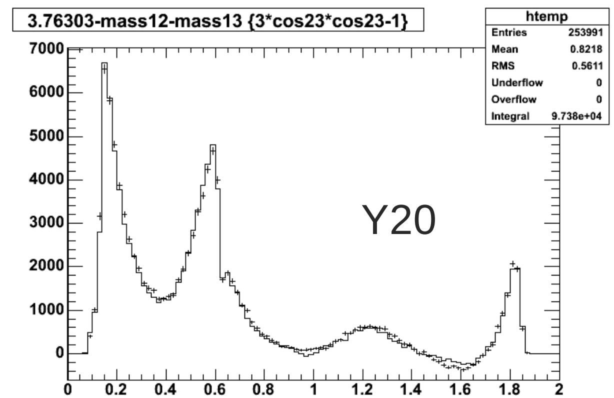
Y00 moments

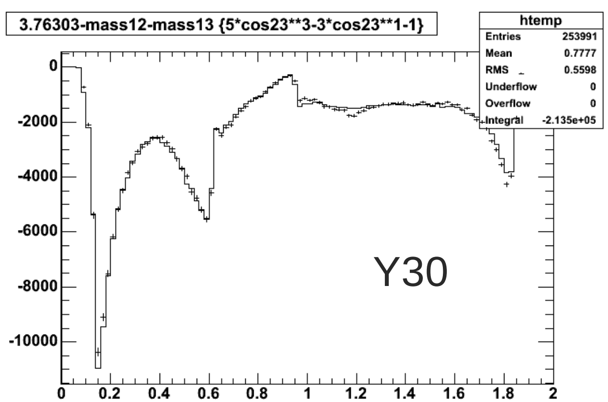
Recall, the lowest order of the spherical harmonics is just a constant,
Thus this is the usual m23 mass projection plot. Clearly seen ~1GeV region
is well behaved.

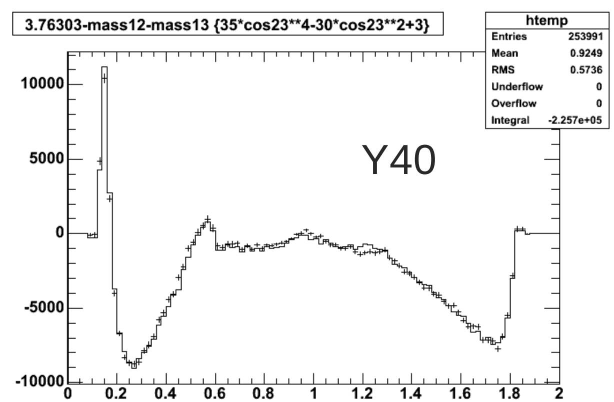


Higher order moments









The model basically can reproduce the correct angular distributions!

Conclusion

1. In the traditional Isobar model we need to introduce adhoc $\sigma(1000)$ and $\sigma(500)$ scalar, now the problem is solved using K-matrix.

2. We still don't get good fit for the K* resonance, no matter in traditional Isobar model, or new K-matrix model.

3. Since we have ultra high statistics, a small imperfection of the parameterization of the lineshape can show the problems.

Example: The value of the Blatt-Weisskopf factor, the choice of Zemach/Helicity model

5. More importantly, we are using the mass and value of the K* from small statistics sample [Mostly before 1984], they subject to large error.

6. Since our analysis are dominated by P-wave(80% in total), thus, it is very hard to improve the chi2/dof if we don't have precise measurement of the K*

End of the Presentation!