BaBar Note # **000**
November 4, 1999

A Tool for Reconstruction of $B^0 \rightarrow J/\psi\ K_L^0$

A. B. Breon
Lawrence Berkeley Laboratory

Abstract

Reconstruction of B^0 decays involving a K_L^0, in particular the mode $B^0 \rightarrow J/\psi\ K_L^0$, will be important in fulfilling BaBar's physics mission. Because the signal of a K_L^0 in the BaBar detector will only give a rough measurement of its direction, it is necessary to perform a series of vector calculations to describe the decay correctly. While the technique is conceptually simple, its implementation is sufficiently complex that it was decided to encapsulate the algorithm as part of the BetaTools package.

FindOmittedParticle computes the most likely K_L^0 momentum given a fully reconstructed J/ψ, known particle masses, and the CM boost. While developed for K_L^0 reconstruction, this tool is potentially useful in other analyses, hence its generic name. Originally written and tested on ASLUND simulated data in release 4.2.1, it has been confirmed to work on SP1 data in release 8.2.5h.

In the original tests, the method achieved energy resolution (for the K_L^0) of approximately 100 MeV, and angular resolution on the order of 100 mrad in the absence of real backgrounds. Testing with current code and more realistic simulations has not yet been conducted.

1 Introduction

The decay $B \rightarrow J/\psi\ K^0$ will be extremely useful in the study of CP violation. There are two resulting modes with approximately equal probability: $B^0 \rightarrow J/\psi\ K^0_S$ and $B^0 \rightarrow J/\psi\ K^0_L$. Both B^0 and $\overline{B^0}$ decay to these final states. For this paper, B^0 will be the particle whose decay we study, while the $\overline{B^0}$ will be ignored. The analysis is identical if they are interchanged.

The first decay is straightforward to reconstruct; the J/ψ decays 12% of the time to two leptons, while the K^0_S decays immediately to two pions; 69% of the time they are charged pions. Leptons and charged pions are easy to detect and measure with high precision in the BaBar detector.

K^0_L, however, is difficult to detect. Neutral, it leaves no trace in the SVT or drift chamber. Being long-lived, it tends not to decay inside the detector. It deposits only some of its energy in the IFR (and less or none in the EMC), with mediocre angular resolution. Studying the mode $B^0 \rightarrow J/\psi\ K^0_L$ will therefore require more effort than for the K^0_S mode. We must be able to reconstruct both decays to make the best possible measurement of CP violation.

In the summer and fall of 1997, I developed code to determine the four-momentum of K^0_L in the decay $B^0 \rightarrow J/\psi\ K^0_L$. This code has been tested on simulated data, largely using ASLUND. BBSIM data was not extensively tested due to its slowness and apparent memory leaks. At the end of 1997 I extracted the basic algorithm, encapsulating it in the BetaTools class FindOmittedParticle.

2 Methods

In developing this analysis method, we consider the following sequence of decays: $\Upsilon(4S) \rightarrow B^0 \rightarrow J/\psi\ K^0_L$, where we have neglected the $\overline{B^0}$. We want to know the four-momentum of four particles. The $\Upsilon(4S)$ momentum is considered as a known experimental parameter. The J/ψ is easily reconstructed, so its four-momentum is considered known. We assume that we have determined a hit in the IFR to be due to the K^0_L, and therefore know its direction well. We know conservation of momentum holds in the second decay. We know the masses of the K^0_L and the B^0 (the $\Upsilon(4S)$ and J/ψ masses are included in their four-momenta). Because we are not considering the $\overline{B^0}$, the first reaction does not conserve momentum as such, but because the $\Upsilon(4S)$ mass is known, the magnitude of the B^0 decay momentum in the $\Upsilon(4S)$ frame is known. These considerations are summarized in Table 1.

Considering all this, it appears that we have an over-determined system, and it should be feasible to determine all the momenta in the decay, as well as apply a stringent cut on whether the decay is consistent with $B^0 \rightarrow J/\psi\ K^0_L$. However, due to variation in the two beam momenta, the 4-momentum of the $\Upsilon(4S)$ is only approximately known. Considering only longitudinal variation, the $\Upsilon(4S)$ boost and invariant mass have some uncertainty. At this time I have not done a detailed analysis of the effect of these variations; qualitatively, they appear to be non-zero but small. For the remainder of this note, this effect will be neglected.

2.1 Solid Geometry

The most convenient frame in which to work with the decay momenta is the $\Upsilon(4S)$ rest frame. The J/ψ 4-momentum is known, and the B^0 mass and energy are known. Because the decay momentum of the J/ψ in the B^0 frame is known (it is a two-body decay), we can boost the J/ψ momentum into the B^0 frame, then solve for the angle θ_B in Figure 1.

Unknown Quantity	Degrees of Freedom
p^μ of 4 particles	16

Known Quantity	Degrees of Freedom
p^μ of J/ψ	4
direction of K_L^0	2
conservation of momentum	4
mass of K_L^0, B^0	2
p^μ of $\Upsilon(4S)$	4
decay $\lvert\mathbf{p}\rvert$ of B^0	1
Total Known	17

Table 1: Known and unknown quantities considered when designing the analysis code.

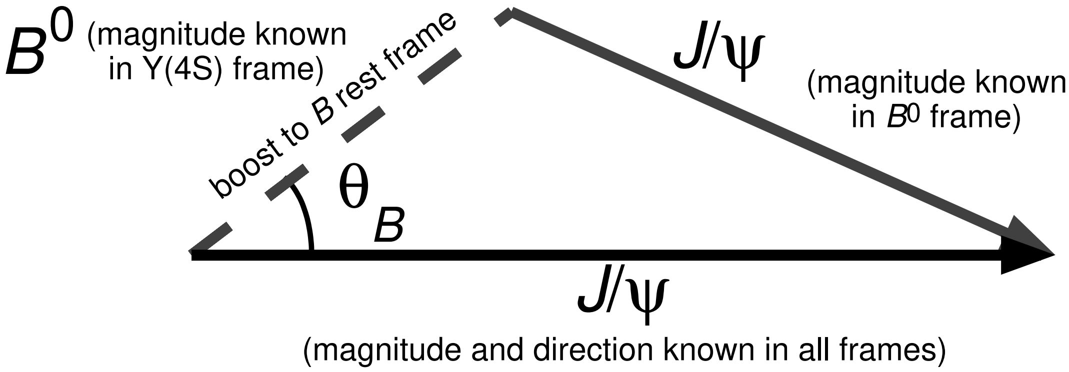

Figure 1: Determination of B^0 angle in $\Upsilon(4S)$ frame using Lorentz transform

Alternately, we can use conservation of 4-momentum. $p_B^\mu = p_{J/\psi}^\mu + p_K^\mu$, and we know $p_{J/\psi}^\mu$ and p_B^0 in the $\Upsilon(4S)$ frame. Thus we can solve for p_K^0. Since the mass of the B^0 and K_L^0 are known, we can find the magnitudes of the 3-momenta of all three particles, and then solve for the relative angles of the three momenta in a triangle (Figure 2).

While the second method is simpler, the first was implemented first and worked well. However, when rewriting the algorithms for general-purpose use, the simpler algorithm was implemented.

These triangles provide an opportunity to cut on the data; impossible angles $|\cos\theta| > 1$ (momenta not satisfying the triangle inequality, $a + b \geq c$) are ruled out as background.

Having determined the triangle of B^0, J/ψ, and K_L^0 momenta by one or both of the above methods, there is still one degree of freedom. Specifically, the K_L^0 momentum must lie in a cone whose axis is the J/ψ momentum (Figure 3), where the azimuthal angle (around the J/ψ) is as yet undetermined.

In order to fix this degree of freedom, we finally make use of the detected neutral cluster. We know the direction (2 degrees of freedom) in the lab frame. While in general the direction in the $\Upsilon(4S)$ frame will change, because the boost is in the z-direction, only the polar angle will change. We can therefore look at the plane defined by ϕ_K as it intersects with the cone of possible K_L^0 decays (Figure 4). There will in general be two intersections, only one of which is correct. Cases

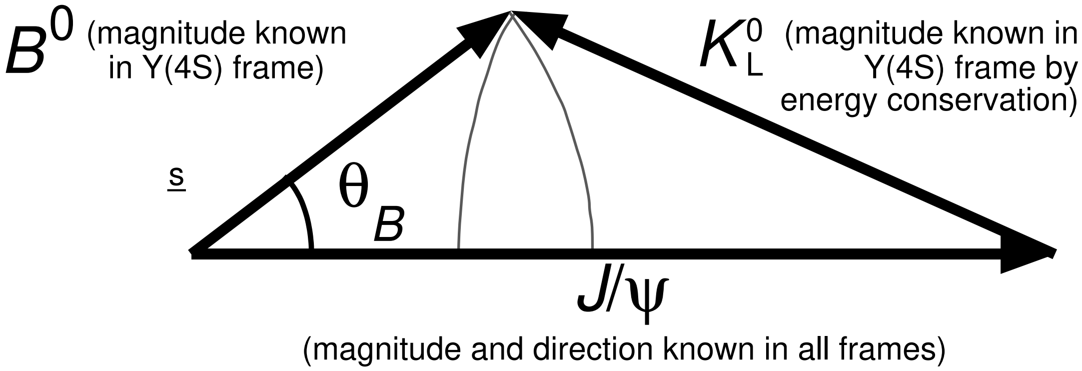

Figure 2: Determination of B^0 angle in $\Upsilon(4S)$ frame using vector addition

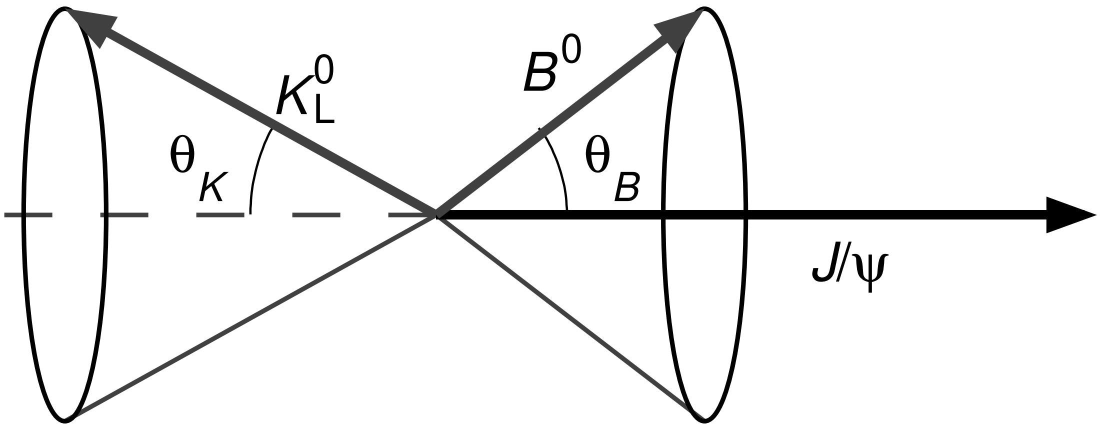

Figure 3: Cones of possible decay directions in $\Upsilon(4S)$ frame

with zero intersections are considered background and discarded.

The two K_L^0 candidates are boosted back to the lab frame. The lab frame angle between each and the measured neutral cluster is compared; the closer candidate is taken as the signal.

3 Implementation

Note: the code for this implementation is available at http://online04.lbl.gov/˜abbreon/mycode/. General code has been encapsulated in the class BetaTools/FindOmittedParticle (Appendix A).

The algorithm described in Section 2 was implemented as part of the BaBar analysis framework, as a C++ BetaUser module in release 4.2.1. The algorithm was encapsulated for release 4.2.1, and continues to work (after fixing a simple bug) in release 8.2.5h.

In testing with version 4.2.1, the resulting data were saved as ntuples for purposes of testing possible cuts using the data analysis package PAW. An additional utility function was implemented outside of the class for easier access to genealogy information of candidates.

In addition, the code included a treatment where the B^0 decay momentum is considered negligible. This was developed first, and the numerous problems encountered with this method motivated

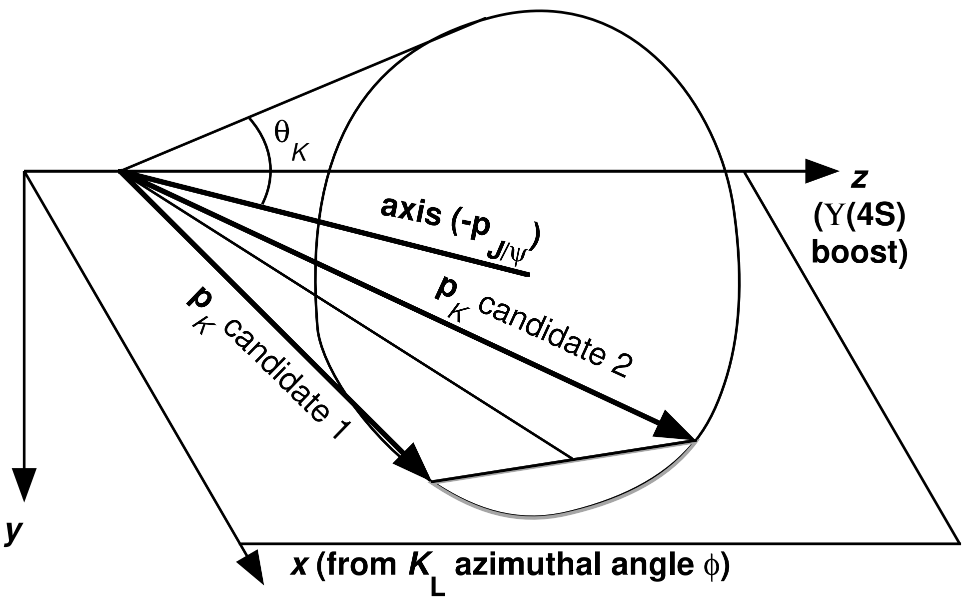

Figure 4: Intersection of azimuth plane (from measured cluster) with K_L^0 cone in $\Upsilon(4S)$ frame

the more complicated approach described in this Note. The results from this code provide a convenient baseline for the quality of the reconstruction.

3.1 Testing data recording

For testing, MC truth was extracted from the simulation and compared to the results of the reconstruction code.

Data was stored in bulky ntuple format to permit interactive testing of possible cut methods. The 4-momentum of the K_L^0 was parametrized by energy, 3-momentum magnitude, and polar and azimuthal angle. Data was stored for both reconstructed and MC particles. In addition, the cosine of the lab frame angle between the MC and the reconstructed K_L^0 was recorded. As well, an entry in the ntuple stored whether the reconstructed J/ψ and detected K_L^0 in fact correspond to the MC decay. Cuts on the angles of the B^0 and K_L^0 cones were applied in PAW, allowing experimentation with the parameters at the expense of larger data files.

4 Results

The code was run on data from ASLUND simulation with B^0 decaying to J/ψ K_L^0 and $\overline{B^0}$ decays turned off. Using this setup, I ran 5000 events. Due to an undiagnosed bug, with $\overline{B^0}$ decays turned on (to examine backgrounds) the code can only run 2400 events.

Using a more complete simulation would have provided a better picture of likely backgrounds, but a variety of problems with simulations available in late 1997 made them inferior to the ASLUND runs. ASLUND was used to obtain the results in this note. Tests with current (SP1) simulations are ongoing.

4.1 Cuts

The standard cut consisted of the following requirements for acceptance:

- Deposited energy in neutral cluster > 100 MeV
- B^0 cone angle cosine between -1 and 1 (Built into FindOmittedParticle)
- K_L^0 cone angle cosine between -1 and 1 (Built into FindOmittedParticle)
- K_L^0 cone *does* intersect with K_L^0 azimuth plane (Built into FindOmittedParticle)
- Lab frame angle between reconstructed K_L^0 and neutral cluster: cosine > 0.97

The first cut is efficient (for low-background ASLUND) at removing clusters not caused by MC truth K_L^0. About half of the events are reconstructed after this cut. The restriction to physical cosines cuts counts by less than 10%. The harshest cut is the requirement that the cone and plane intersect, improving the resolution but cutting the counts by a third. The final angle cut is used to fine-tune the analysis; It reduces counts significantly, but improves resolution.

4.2 Problems with naive approach

Before presenting the results of the analysis, I quickly present the motivation for such a seemingly baroque analysis. I originally assumed the B^0 to have zero momentum in the $\Upsilon(4S)$ frame. The analysis is much simpler, but results in a wide variation in calculated mass, including $m_K^2 < 0$! It was determined to be due to nonzero B^0 momentum, prompting the more complicated method. A striking demonstration is the correlation between the computed K_L^0 mass and the cosine of the angle between the monte-carlo K_L^0 and B^0 momenta in the $\Upsilon(4S)$ rest frame (Figure 5).

4.3 Results from geometry with clean ASLUND

When the more sophisticated technique is applied to the same data, the improvement is dramatic (Figure 6). Because we use the K_L^0 mass in the calculation, the reconstructed mass should be forced to be equal to the known mass to within a fraction of an MeV. In the original implementation, a few candidates ended up with dramatically lower masses. However, after encapsulating the algorithm in BetaTools/FindOmittedParticle this effect went away. This is not surprising, given that the new code is far less baroque.

The figures of merit used in fine-tuning the cuts were the overall rate, the energy resolution (variation in the difference between reconstructed and MC K_L^0 energy), and the angular resolution (variation in the cosine of the angle between reconstructed and MC K_L^0 momentum). The selection has not been optimized, however; cut paramters have been adjusted iteratively while qualitatively evaluating the trade-offs.

As described in Section 4.1, the final cut, on the angle between the reconstructed and measured momentum, is easily tuned. A tighter cut reduces the RMS of the energy difference distribution, but makes the average energy difference become more negative, although still consistent with zero. It also improves the angular resolution. While there appears to be room for improvement using extremely high thresholds, I used 0.97 as a conservative number.

The result of these cuts, for an ASLUND simulation of 5000 decays, is plotted in Figure 6. The summarized results:

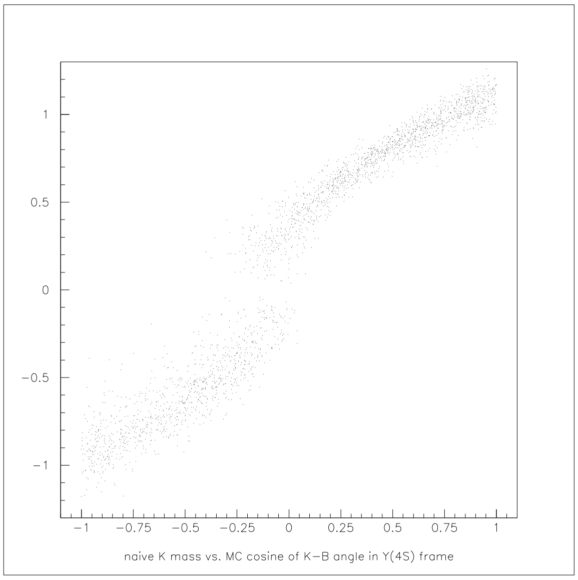

Figure 5: Correlation of naively reconstructed K_L^0 mass and cosine of angle between monte-carlo K_L^0 and B^0 (in $\Upsilon(4S)$ frame)

- RMS energy variation is 83 MeV around a mean 32 MeV too low.
- reconstructed K_L^0 angle from MC ranges up to 450 mrad; average angle $\approx$ 130 mrad.
- of 5000 events run, roughly half satisfy the first cut, and half of these get past further cuts.

5 Conclusion

In the case that we can determine that a neutral hit corresponds to a K_L^0, the algorithm described above provides fairly good reconstruction of the decay $B^0 \rightarrow J/\psi\ K_L^0$.

While the analysis is still only partially complete, these initial results under simulation are sufficiently encouraging to warrant further development. Much could be done:

- Compare results from two cone-finding algorithms.
- Consider variations in $\Upsilon(4S)$ momentum.
- Calculate error in results.

Already, however, this seems a powerful tool for reconstructing K_L^0. Furthermore, the techniques for manipulating the geometry of the decay in the $\Upsilon(4S)$ frame may be useful in other decay modes where we have only partial knowledge of the final state. Accordingly, the algorithm has been implemented as the class BetaTools/FindOmittedParticle, to assist other users in reconstructing such decays.

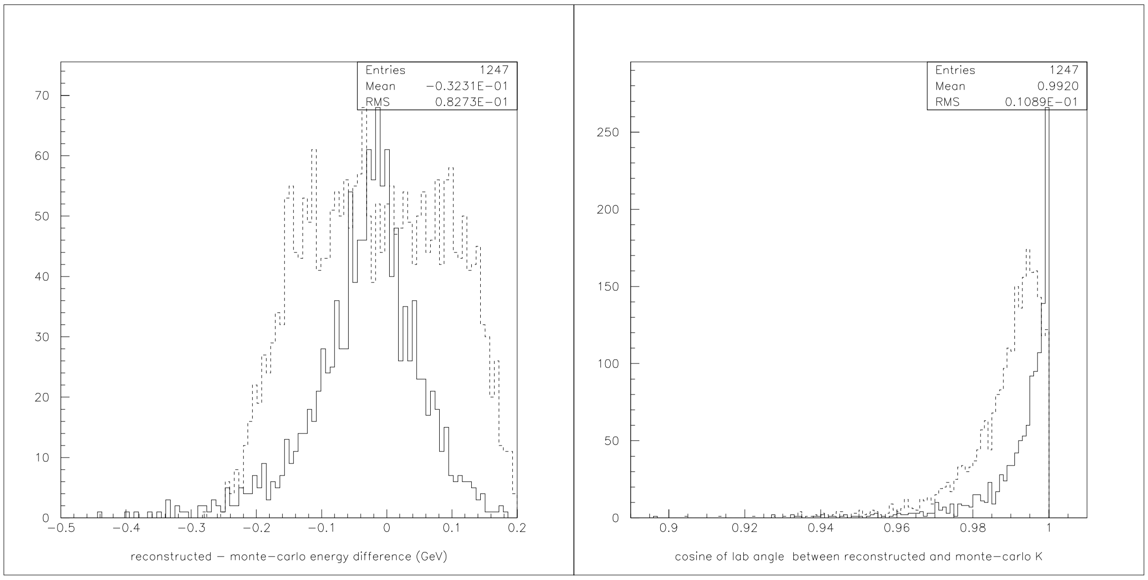

Figure 6: Energy (GeV) and angular ($\cos\theta$) resolution (ASLUND with $\overline{B^0}$ decays turned off) (Dotted lines are results from naive approach for comparison, with only the energy cut.)

A Header File for BetaTools/FindOmittedParticle

```
//--------------------------------------------------------------------------
// File and Version Information:
//  $Id: FindOmittedParticle.hh,v 1.2 1998/12/20 02:11:17 jake Exp $
//
// Description:
// Class FindOmittedParticle.
// Given resonance (Y(4S)) momentum, B0 momentum, and
// K0L mass at creation, this object then finds the best
// candidate for the K0L 4-momentum given the 4-momentum
// of the J/psi and the direction of the IFR cluster.
//
// Because this is a general utility class, B0 is labeled Child,
// Y(4S) is labeled Reson (for Resonance), J/psi is labeled Obs
// (for Observed), and K0L is labeled Sought (for the calculated
// values) or Seen (for the actual IFR cluster).
//
// Constructor takes CM 4-momentum, Child mass, and Sought mass.
//
// fitToSeen() calls makeCone(), then closestFit().
//
// makeCone() finds the possible Sought 4-momenta given the Observed
// momentum.
```

```
//
// closestFit() finds the best fit between the cone of possibilities
// and the Seen neutral cluster.
//
// If there is demand, a function may be added to extract the
// parameters of the cone describing possible Sought momenta;
// makeCone() and closestFit() would then be made public;
// or perhaps a subclass could implement this functionality.
//
// Note: this class currently only works in coordinate systems with
// the boost along the z-axis.
//
// Environment:
// Software developed for the BaBar Detector at the SLAC B-Factory.
//
// Author List:
//      Adam Breon                          Original Author
//
// Copyright Information:
// Copyright (C) 1997    Lawrence Berkeley Laboratory
//
//------------------------------------------------------------------------
/**/
#ifndef FINDOMITTEDPARTICLE_HH
#define FINDOMITTEDPARTICLE_HH

// Access defaults for B0 mass?
// #include "PDT/PdtLund.hh"

#include "CLHEP/Vector/ThreeVector.h"
#include "CLHEP/Vector/LorentzVector.h"

class FindOmittedParticle {

//--------------------
// Instance Members --
//--------------------

public:

  // Constructors
  FindOmittedParticle(
     const HepLorentzVector & ip4Reson, // set Y(4S) p4, usually.
     const double imChild, // B0 mass.
     const double imSought ); // K mass (p4 not known).
```

```
  virtual ~FindOmittedParticle( );

  HepLorentzVector fitToSeen( const HepLorentzVector & p4Obs,
      const Hep3Vector & p3Seen );

protected:

  void makeCone( const HepLorentzVector & p4Obs );

  HepLorentzVector closestFit( const Hep3Vector & p3Seen );


  HepLorentzVector p4Reson; // 4-momentum of resonance
  float mChild, mSought; // mass of Child and Sought

  float cmEChild, cmpChild; // energy, 3-momentum of Child.

  float cmBetaChild, cmGammaChild; // beta, gamma of Child.

  // Boost vectors.
  Hep3Vector beta; // From lab to CM
  Hep3Vector debeta; // From CM to lab

  // The cone
  Hep3Vector cmAxis; // Axis of CM cone
  double        openCos; // cos of opening angle.
  double cmESought; // CM energy of sought particle.
  double cmpSought; // CM 3-momentum of sought.

};
#endif
```