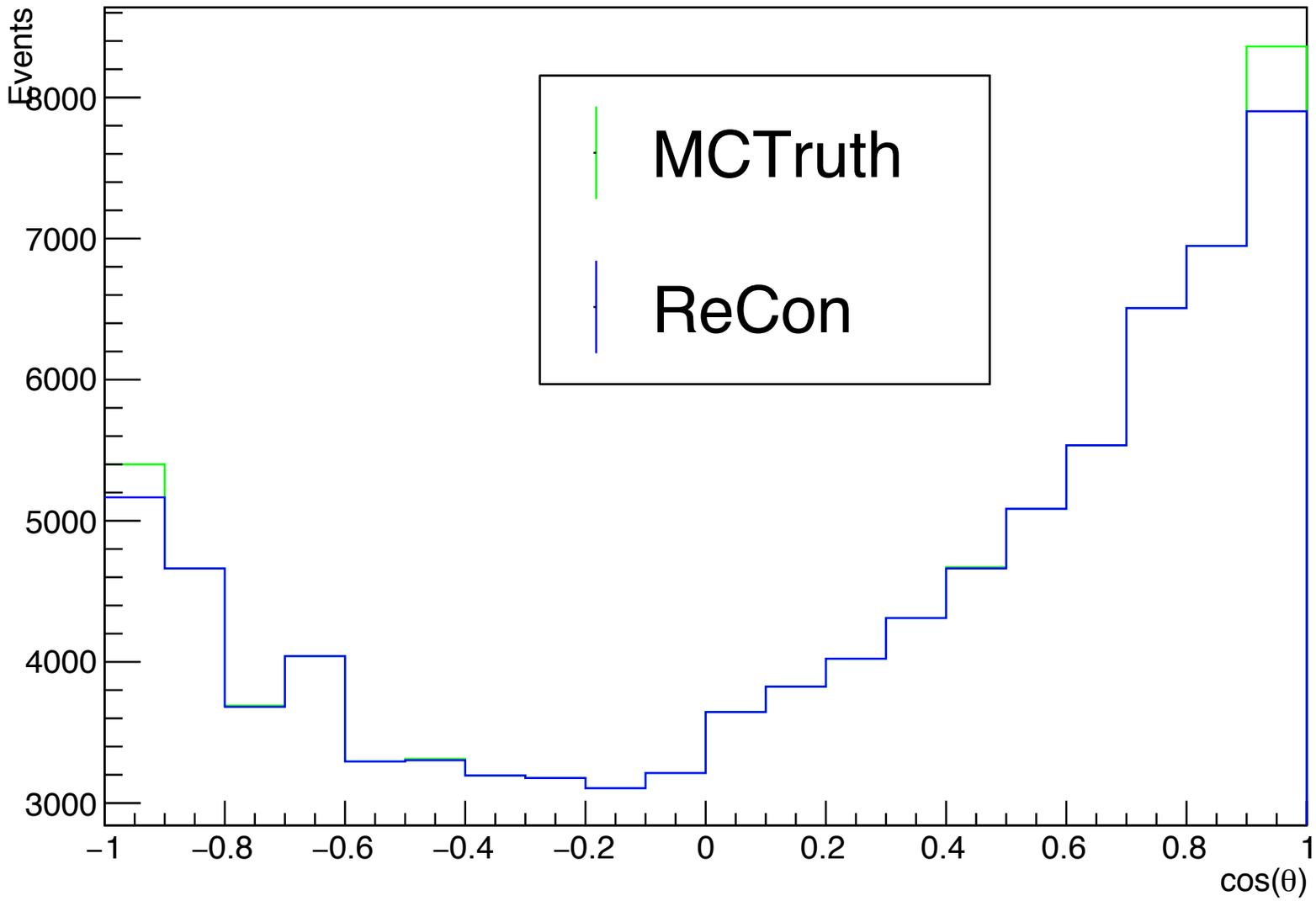
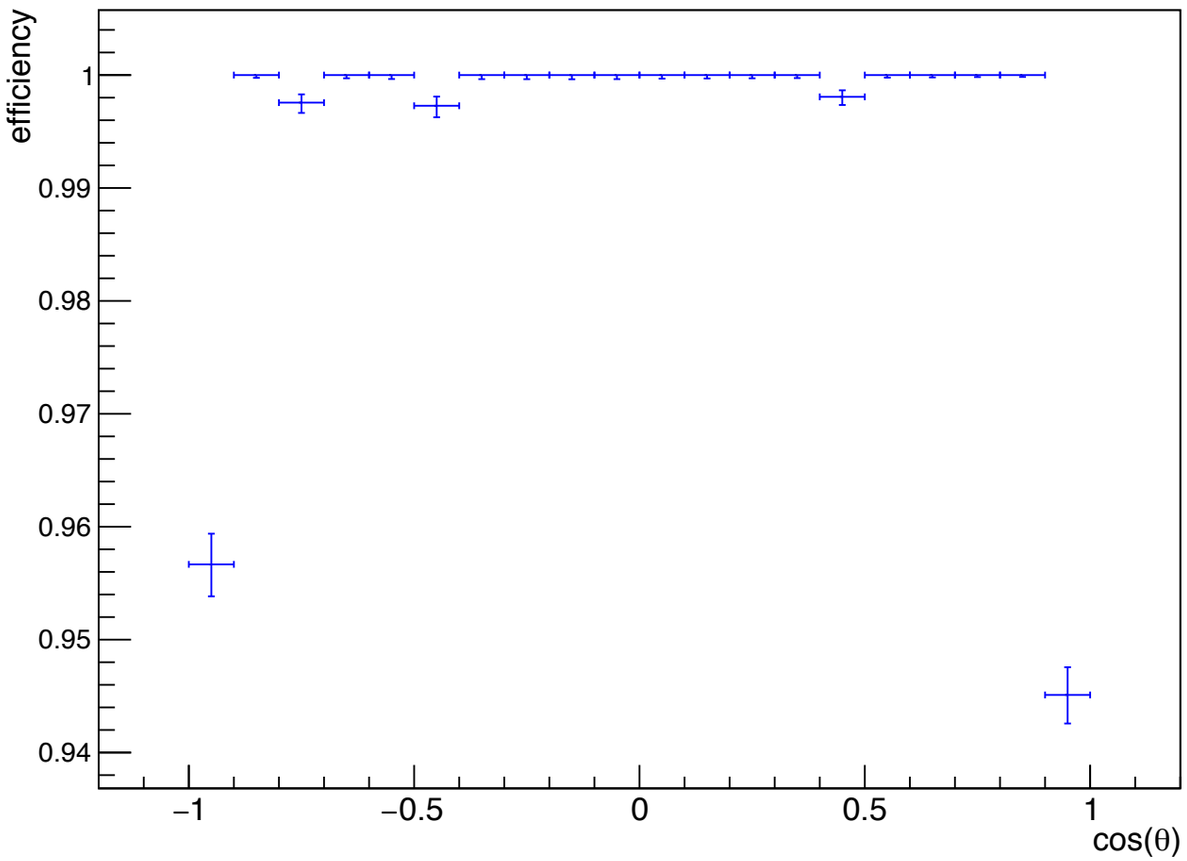
Forward-Backward Asymmetry at 91.1876GeV

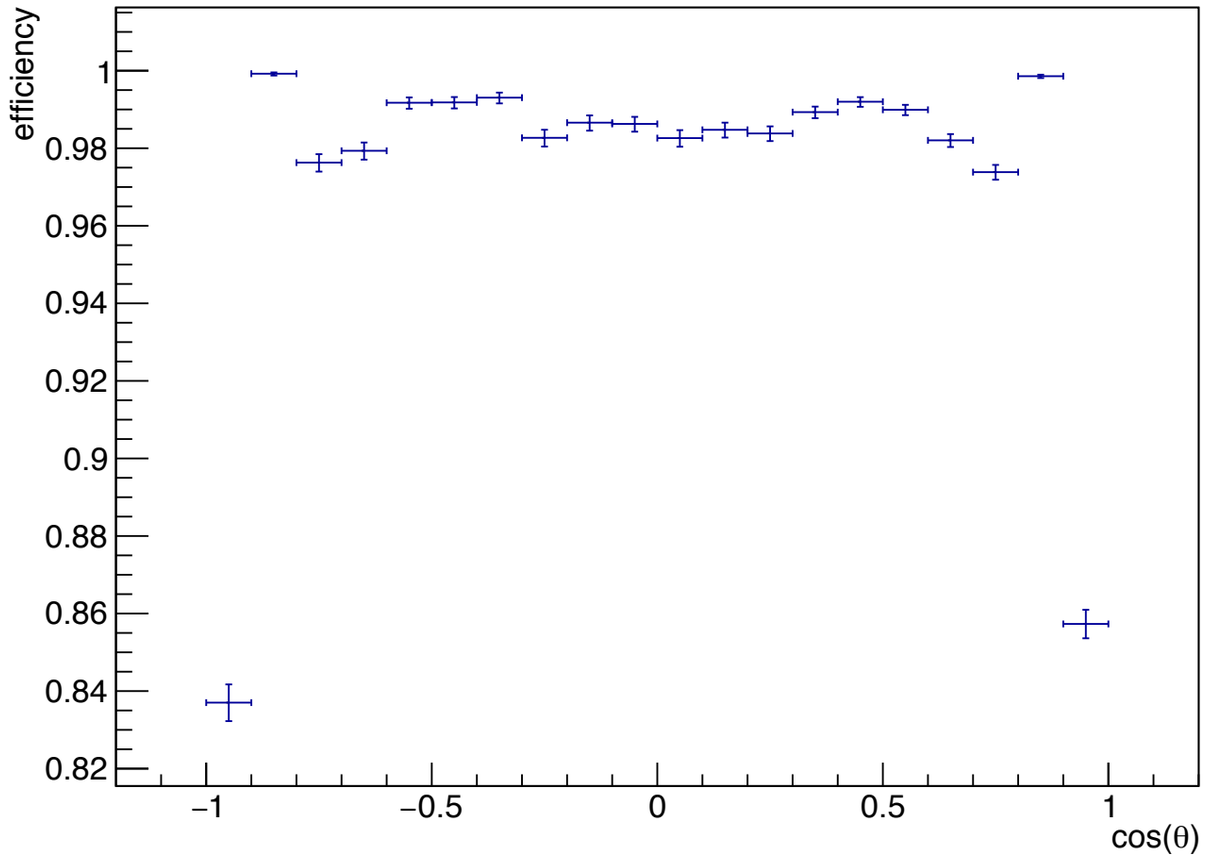
Menrgan
July 3rd, 2017



Make the efficiency considering connection between MCTruth events and Reconstructed events.
It means Recon histogram is filling with events in MC Truth level which have corresponding Reconstructed events



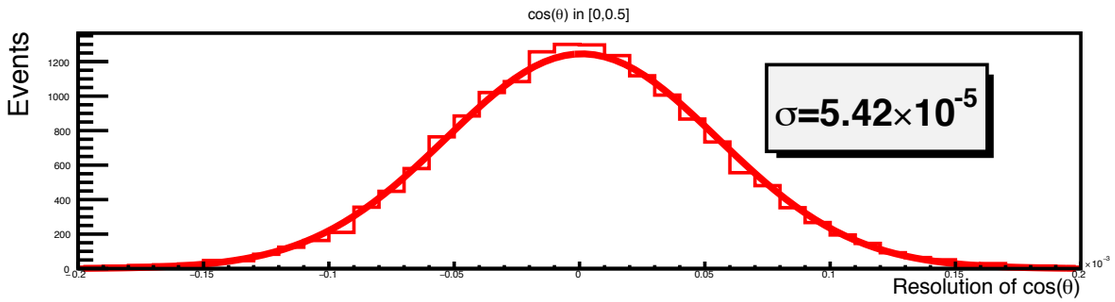
Reconstructed with only cut of two muons

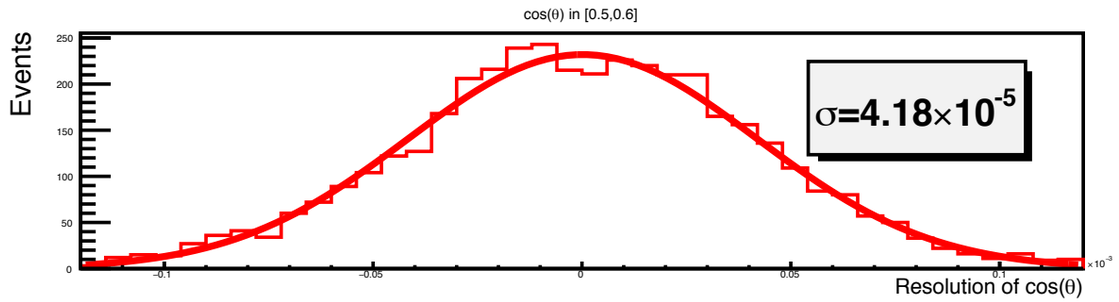


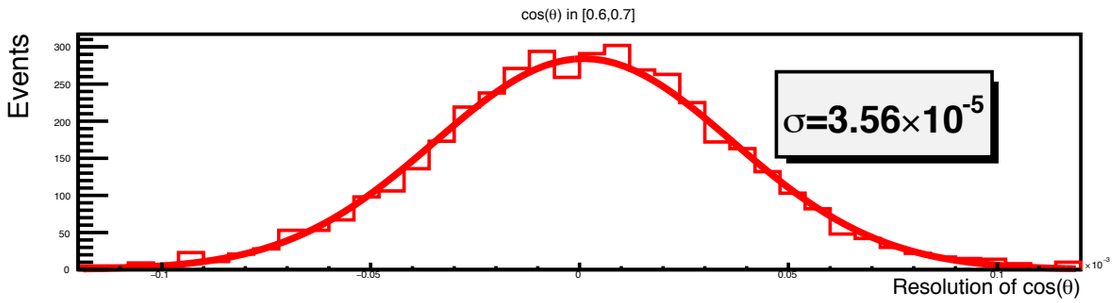
Old one with a cut: has at least one possible combination of particles for this final state

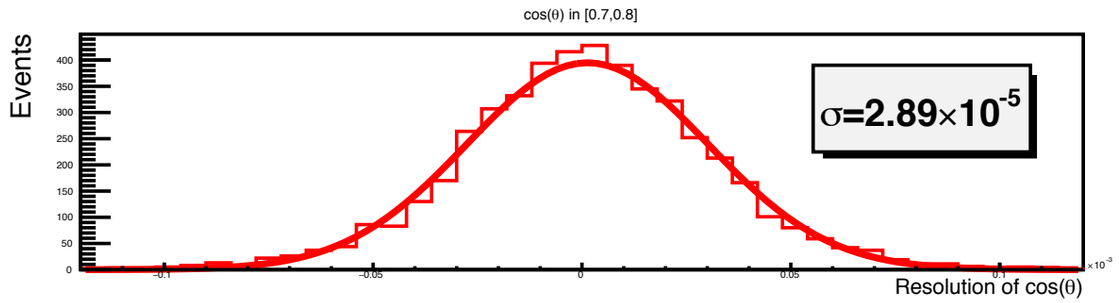
This time, efficiency is high than I made before, when I used reconstruction code from Gang. In that code, there is a cut of back-to-back.

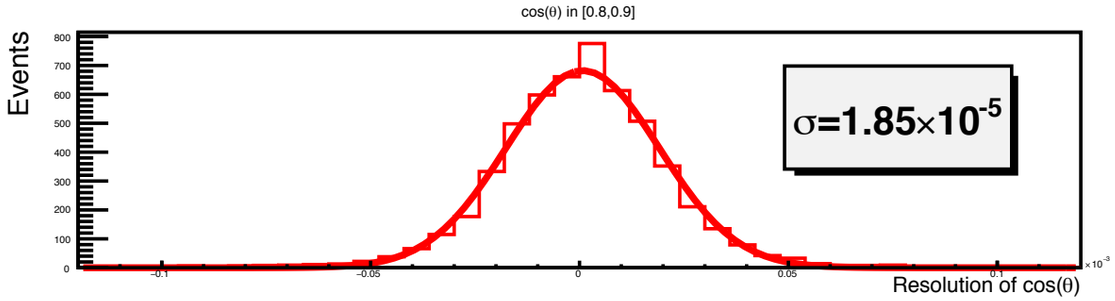
Since the error of cosθ is very small, it should not have a visible difference

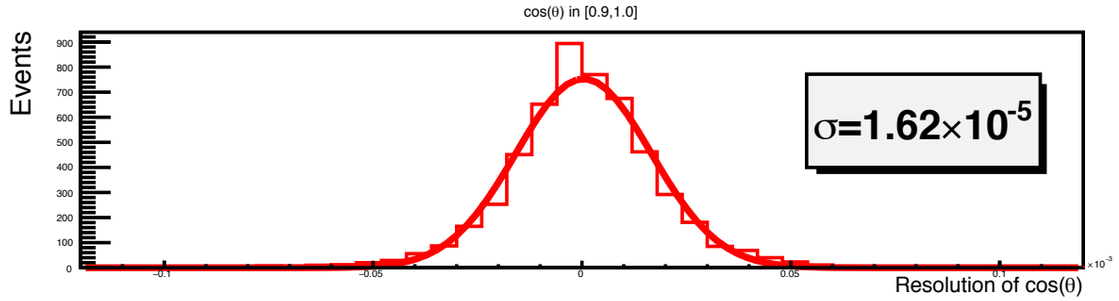












```
//**********************************************************************
//
//   PUT TOGETHER ALL COMBINATIONS OF PARTICLES FOR THIS FINAL STATE
//
//**********************************************************************
vector< vector< FSParticle* > > pCombos; pCombos.clear();
for (unsigned int ifsp = 0; ifsp < particleNames.size(); ifsp++){

  int   status = particleStatus[ifsp];
  vector< FSParticle* >  pList ; pList.clear();
  if( status == 1 ) pList  = DictPList[particleNames[ifsp]];
  else{
    pList.push_back(new FSParticle(particleNames[ifsp], true));
    ParticleTrash.push_back(pList[0]); // save address for memory freeing
  }
  vector< vector< FSParticle* > > pCombosTemp = pCombos; pCombos.clear();
  if  ( pList.size() == 0 ) break;

  for ( unsigned int ipl = 0; ipl < pList.size(); ipl++ ){

    if ( ifsp == 0 ) {

      vector< FSParticle* > combo1; combo1.clear();
      combo1.push_back(pList[ipl]);
      if ( checkCombination( combo1, (combo1.size()==particleNames.size()), fsinfo->inclusive()||fsinfo->missingMassFit() ) )
        pCombos.push_back(combo1);

    } else {

      for (unsigned int itc = 0; itc < pCombosTemp.size(); itc++){
        vector< FSParticle* > combo2 = pCombosTemp[itc];
        bool duplicate = false;
        if  ( !pList[ipl]->missed() ) {
          for (unsigned int ic = 0; ic < combo2.size(); ic++){
            if ( combo2[ic]->missed() ) continue;
            bool shared = pList[ipl]->duplicate(combo2[ic], m_full);
            if ( shared ) { duplicate = true; break; }
          }
          if(m_debug>4){
            printf("duplicated one ? %2d\n", duplicate );
          }
          if ( duplicate ) continue;
        }
        combo2.push_back(pList[ipl]);
        if ( checkCombination( combo2, (combo2.size()==particleNames.size()), fsinfo->inclusive()||fsinfo->missingMassFit() ) )
          pCombos.push_back(combo2);

        if ( pCombos.size()>100000 ) break;
      }


    }
    if (pCombos.size() > 100000) break;

  }

  if (pCombos.size() > 100000) break;

 }
m_cutpass[ifs][2]++;
```

The Old Cut. Efficiency of this step is 96%