MCMC and correlated Pairs

Claire Prouve

DalitzMCMC

MCMC: Markov Chain Monte Carlo

implemented in MINT by Jeremy (Veronika)

tests by Jeremy look good!

super fast but **biased for small number of events**
-> make huge sample and randomly select events

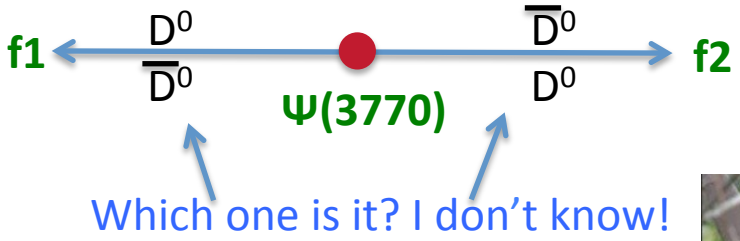
New: *DalitzMCMC* now automatically generates a huge sample and randomly selects the amount you need:
in DalitzMCMC.h:

```
void FillEventList( DalitzEventList& eventList,
                    const unsigned int& NEvents, double rejectionFactor = 100 );
```

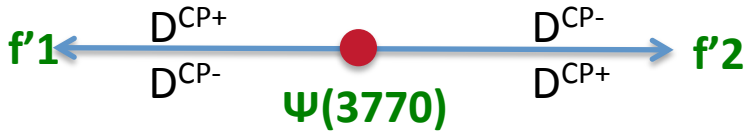
-> only every 100th generated event gets put into the eventlist

Correlated Pairs

Generation of correlated pairs: this is going on at CLEOc



f1/f2: final states accessible both by D^0 and $\overline{D}^0$ decays



f'1/f'2: final states accessible both by D^{CP+} and D^{CP-} decays

But we can use the interference in the decay paths to extract some important parameters for our analysis!

$$A(\Psi(3770) \to f1,f2) = A(D^0 \to f1) \bullet A(\overline{D}^0 \to f2) - A(D^0 \to f2) \bullet A(\overline{D}^0 \to f1)$$

Implement a class in MINT that can generate correlated pairs:

use the MCMC method to generate an event with **two final states** that follows the PDF corresponding to $A(\Psi(3770) \to f1,f2)$.

DalitzMCMC_corrPairs

- Completely analogue to DalitzMCMC:

  DalitzMCMC_corrPairs(**const DalitzEventPattern& pat1,**
  const DalitzEventPattern& pat2, const unsigned int& seed=0);

  void FillEventList(**DalitzEventPairList& myPairList,**
  const unsigned int& NEvents, double rejectionFactor = 100);

- New structure to store (correlated) pairs analogue to DalitzEventList:
  class DalitzEventPairList:
  public MINT::EventList< std::pair<DalitzEvent, DalitzEvent> >

  Same methods, for example:
  bool saveAsNtuple(const std::string& fname="DalitzEvents.root",
  const bool addSij = true)const;

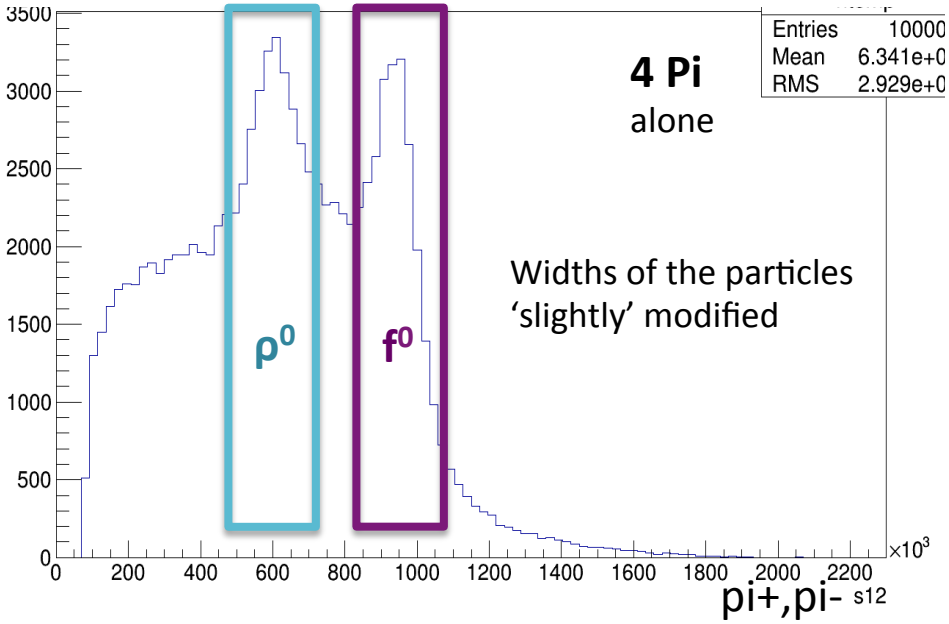
  Note: the PDG information about the D mesons in the tuple is obviously nonsense.

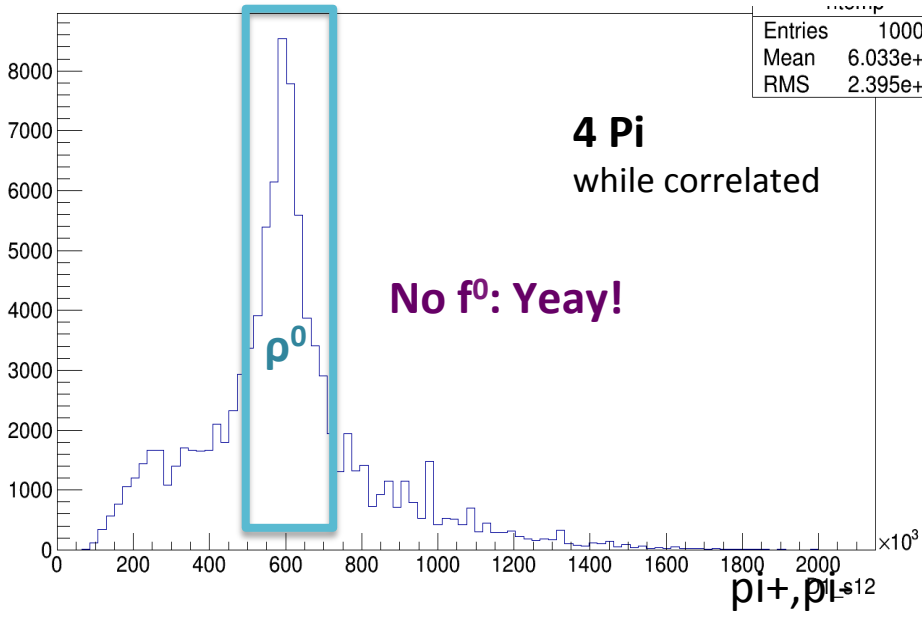
If you want more info about the new classes come and ask me☺

Testing DalitzMCMC_corrPairs

1. Amplitude for the K_sPiPi: D0->rho(770)0(->pi+,pi-),K(S) ⟶ **CP even**
2. Amplitude for the 4Pi: D0[P]->rho(770)0(->pi+,pi-),rho(770)0(->pi+,pi-) ⟶ **CP even**
   D0->f(0)(980)0(->pi+,pi-),f(0)(980)0(->pi+,pi-) ⟶ **CP even**

-> If the correlation thing works, the f(0)(980)0(->pi+,pi-) should be suppressed!





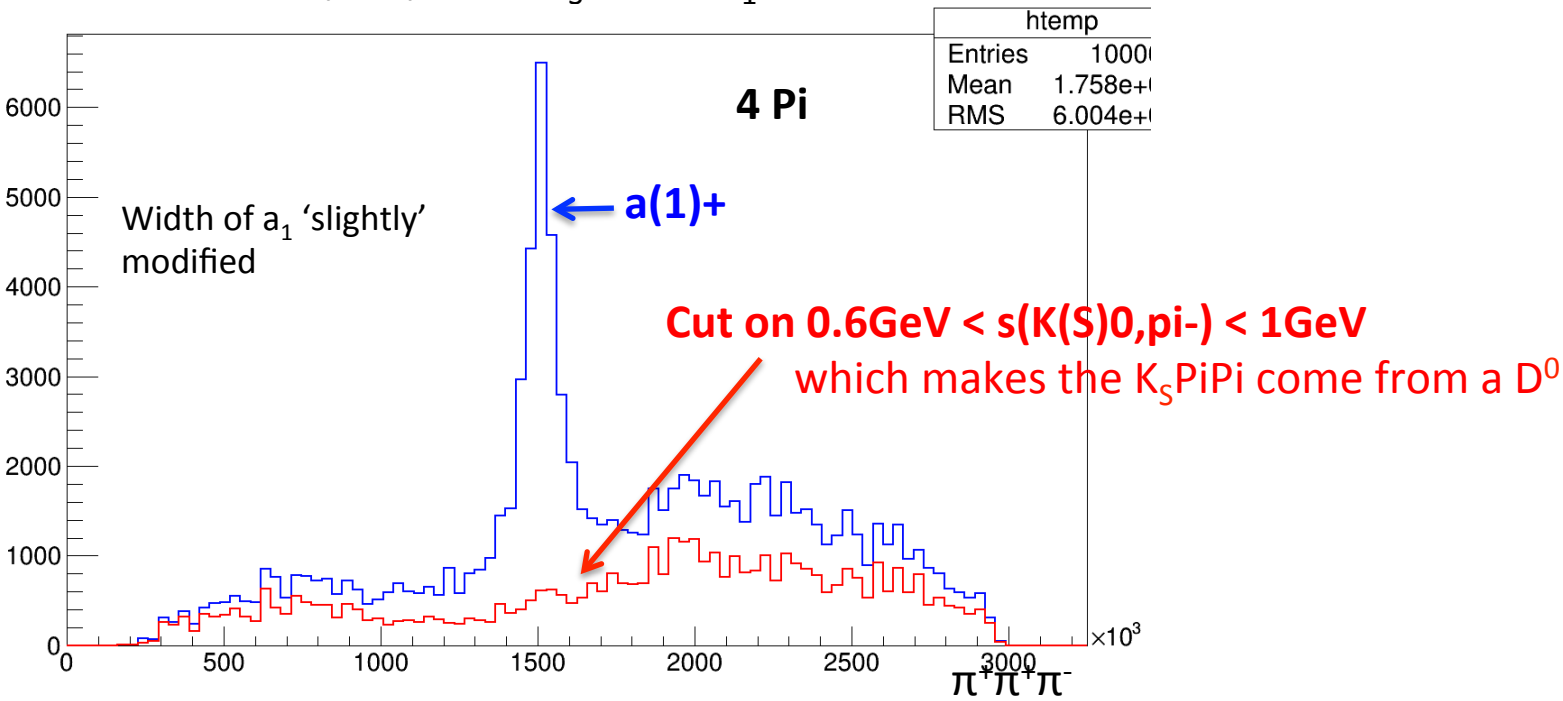
Testing DalitzMCMC_corrPairs

1. Amplitude for the $K_s\pi^+\pi^-$: D0->K*(892)bar-(->K(S)0,pi-),pi+
2. Amplitude for the 4π: D0->a(1)(1260)+(->sigma10(->pi+,pi-),pi+),pi-

   $\overline{D0}$->**a(1)(1260)-**(->sigma10(->pi+,pi-),pi-),pi+

Flavour specific states

If we see the K*(892)bar in $K_S\pi^-$, the a_1^+not be seen in $\pi^+\pi^+\pi^-$



Future MCMC

- Pull study to see when MCMC converges/ how reliable it is
- Maybe see if we can reproduce the results of the K_SPiPi analysis with our DalitzMCMC_corrPairs
- Analysiiiiiiiis!!!