MintDalitz/Mojito Tutorial

MINT

- Before you do this tutorial, go to doc and do the tutorial there, MintTutorial.
- Once you're done, continue here.

Directory Structure, compilation, etc

- Different groups of classes are put into some directory, say NamedParameter
- All source code is stored under the src/ directory. All header files are under Mint/ and all test files are under test/ in a mirror directory structure to src/
- Each directory contains a subdirectory called src with the source code. Testing code, is stored under test/. Only the code in the test directories contains a main() function and can be compiled.
- To compile, got the the test directory of interest and type make.

Make your own

- The code that you have now will certainly change in the future.
- To make this less painful it would be wise not to put any of your own code into any existing directory (at least not code that you want to keep).
- Make your own directory:

```
cd MindDalitz/Users
mkdir myOwn
mkdir myOwn/src
mkdir myOwn/test
cd myOwn/test
cp ../../Jonas/test/Makefile* .
```

DecayTree

- A lot of the fitting code is based on DecayTree. They are simply trees that identify the amplitude that's being considered.
- Go to test/Mojito/DecayTrees/test
- Run ./testTrees (no input file needed)
- Have a look through the code. Create and print out your own decay tree. Use DecayTree::oneLiner() to print the decay tree in a single line.

Amplitude

- Amplitudes take decay trees as an argument in their constructor. They also take the eventList as an argument to have access to the Dalitz Events.

- All other aspects of the amplitude are created automatically from the DecayTree, and the information provided by ParticleProperties. This includes spin factors, masses, widths. Even which lineshape to use (so far the options are BW or GS). Have a look at SpinFactorMaker.cpp and LineShapeMaker.cpp to get an idea.

- Amplitudes/test has some code, although not as well tidied up as the other examples.

Amplitudes and ParticleProperties

- ParticleProperties is similar to the ParticleProperties service in DaVinci. It gives you access to PDG masses, widths, spin etc.

- There is a short example in ParticleProperties/test, run by typing ./testPP < testPP.txt

- By default, ParticleProperties gets its information from ParticleProperties/src/mass_width.csv

- If you need to change any of that (e.g. to add a sigma2 or so), make a copy of that file in your local directory and modify that. The local file ./mass_width.cvs always has priority over the default file.

Amplitudes in Practice

- NamedDecayTrees is a class that contains a list of all decay trees that the program knows about. Have a look at the file Mint/NamedDecayTrees.h to see how it's done - simply one decay tree after the other is created and added to the list. If you want to add decay tree that doesn't exist, this is the place to do it.
- These DecayTrees are accessed by their name. This, by default, what comes out if you execute DecayTree::oneLiner()
- You won't access the DecayTrees directly in most cases, but...

FitAmplitudes

- FitAmplitude is simply an Amplitude with a phase and an magnitude that can be fitted.
- The phase and magnitude are FitParameters which can be initialised in a text file.
- Their names are those of the respective DecayTree.
- FitAmpSum creates all possible amplitudes compatible with the relevant final state. It also provides a file called protoFitAmplitudeFile.txt with the names for the FitParameters of all Amplitudes the program knows about.
- Have a look at GammaFits/threeBody/protoFitAmplitudeFile.txt

Event Pattern

- This is imply an array of integers, containing the PDG code for the mother an the final state.
- Used in MC generation
- Akso used for example by FitSum to decide if an amplitude is compatible with the events or not.
- Check the code in GammFit/threeBody (next slide) for examples.

Gamma Fit

- The current state of the gamma fit is in GammaFits/threeBody.
- The file with the fit amplitudes and other parameters is called "threeJim.txt" (sorry Jim), this is because it contains exactly the values Jim uses in his analysis, so I can compare the output of the routines.
- There is also a file called mass_width.csv. This is a file with the pdg masses and widths and other information used by ParticleProperties (and hence determines the Amplitudes). There is a default one in ParticleProperties/src. However, the local one has priority, if it exists.

Gamma Fit Hands-On

- So far the gamma fit only has Bplus events and fits gamma while keeping delta and rb constant.
- I'll modify this shortly to do a full fit to all three parameters. But this could also be a good excercise, so why don't you try that, too. It will require that you generate some B minus decays and do a simultaneous fit.
- For the simultaneous fit, you will need to create a Neg2LLSum, to which you will add the Neg2LL for Bplus and Bminus.
- Neg2LLSum is untested and it might not work. So if you're not feeling very adventurous today, maybe better wait until I've tested it.

More stuff

- Most test directories have code that works and is sometimes more, sometimes less instructive.
- I didn't really talk about event generation - there are examples in the code we looked at though.
- And finally: This is new code, a lot will change. I'll try to minimise the impact on you, but I'll have to ask for a bit of patience if every now and again I make some non-backward compatible changes - will happen more frequently in the beginning.
- Let me know what changes you'd like to see.