

LesHouches Package ~overview/work plan

Soushi Tsuno, A.S. Thompson, and Henry J. Frisch

- 1 Introduction
- 1 Review
- 1 Outline of LesHouches Module
- 1 Plan

Simulation Meeting on Oct.21.2004

Introduction

LesHouchesModule is the common interface of ME-generators for CDF offline code.

This interface is based on the common agreement at Les Houches 2000, which proposed the common block structure for ME-generators, so that any ME-generators are easily embedded into showering-MCs (**PYTHIA** or **HERWIG**) according to this common manner.

 **LHA format** : Common data format for ME-generators

Nowadays, recent development of HERWIG are ready to use as well as PYTHIA. And most of ME-generators takes this format.

We should go ...

Review

Stan and I have started this development since 2001 which was the time we used PYTHIA.6.138 before upgrading PYTHIA.6.203.

[pythia_i/pythia_service/](#)

However HERWIG (v6.4) still not worked on LHA format in those time.

Under upgrading the coming PYTHIA.6.203, we had lots of problems.

- 1) **conflict problem** : same name of subroutine and common block variables.
- 2) **executable size** : if we include the whole ME-calculation routines the executable becomes to be > 500 Mbytes.
- 3) **compile option** : depending on this options is unpreferable feature of the CDF coding policy since Run I.

Then, we realized that the ME-generators should be outside CDF offline.

 prepare the (unweighted) event file from the ME generator,
then, fed it into our CDF-offline interface.

LesHouches Package

Now, HERWIG.6.5 is ready to use LHA format.
we extend “pythia_i/pythia_service” to HERWIG.

LesHouches Package

We do not need “pythia_i/pythia_service” anymore.

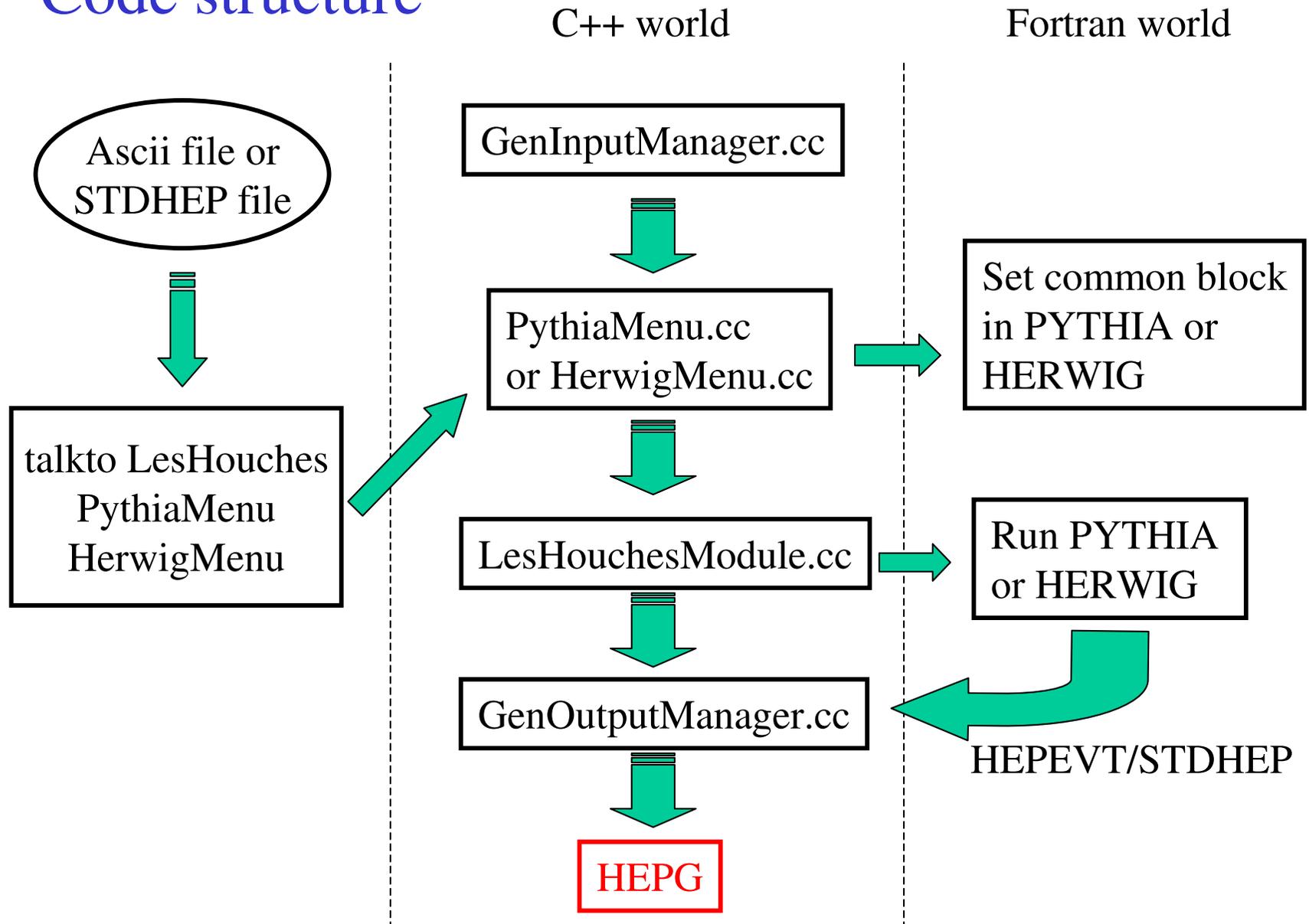
Same as before, LesHouchesModule only works for the interface.
LesHouchesModule requests :

1) (unweighted) event file from the ME-generator. **That's it!!**

We can select PYTHIA or HERWIG via tcl file of LesHoucesMoule.

If it does not work, check the two core subroutine, **UPINIT/UPEVNT**
between CDF code and your code. (if else, I will describe later.)

Code structure



What we expect

- 1) **Coding will be simple.** Because LesHouches does not need to run the ME-generators inside CDF-offline. Users just need to prepare the event file which was produced outside CDF-offline code.
 - ➡ easy to debug , guarantee the right operation of ME-generators.

- 2) **MC production will be much smarter.** We do not need HEPG files (and STDHEP files) before making Production data. LesHouches just allows us to start the (unweighted) event file from ME-generators.
 - ➡ the storing file size will be minimum.

- 3) **Easy for a systematic study of MC.** At least, ME has no systematics. So that all parameters are controlled by the tcl files.
 - ➡ In our current scheme (to start from HEPG/STDHEP), we can not trace what parameters are used in showering MCs.
 - ➡ On the other hand, in LesHouchesModule, just look at tcfs, and if we want to change the parameters for the systematic study, just need to change the tcfs.

Parameters

- LesGenType** : Select ME-generators of your input event file.
Alpgen/CompHep/MadGraph/Grappa/MCFM/USER
- LesDatFile** : Name of your input event file.
- LesPSModel** : Set showering-MC of PYTHIA or HERWIG.
- firstEventToRead** : The starting event number to store in HEPG.
- putResonance** : Set resonance particles (W or Z) in the event history.
- scaleOfKtClus** : Set energy scale for the parton shower based on kt-clus algo..
- useTAUOLA** : Use or not. The helicity info. from ME is connected correctly.
- useStdHep** : Read from STDHEP files or write the STDHEP file.
- StdhepFileName** : The name of STDHEP file to read or write.
- NumOfStdEvent** : The event number to read or write in STDHEP file.
- PythiaMenu** : Menu of PYTHIA. This is exactly same as CDF-PYTHIA.
- HerwigMenu** : Menu of HERWIG. This is exactly same as CDF-HERWIG.
- (useJIMMY)** : (Use JIMMY underlying event model for HERWIG.)

How to include your generator (case A)

In case if your event file does not work in LesHouchesModule,

1) Check the event file format. Type,

```
~% diff [your generator]/UPINIT LesHouches/src/[your generator]_UPINIT
~% diff [your generator]/UPEVNT LesHouches/src/[your generator]_UPEVNT
```

If you see a lot of lines, your event format is not suitable in LesHouchesModule.

But!

To use it, replace your **UPINIT/UPEVNT** (and the other related subroutines) into LesHouches/src/[your generator]_les.F as the subroutine name of [your generator]_UPINIT/UPEVNT.

It should work.

How to include your generator (case B)

In case if your generator does not exist in LesHouchesModule,

- 1) Copy **UPINIT** and **UPEVNT** in your generator into LesHouchesModule/src/**user_les.F** with the name of **USER_UPINIT** and **USER_UPEVNT**.
- 2) Run it with the “**LesGenType set USER**” in tcl file.

More specified details is in <http://nkek15.fnal.gov/~tsuno/leshouches.html>

Comment

The LesHouchesModule provides a simple framework to embed your generator into CDF offline code if you have the event files from ME-generators.

- ➡ I don't want any ME-generators to run inside CDF offline code.
Watch out the fortran code. The common block is the global variable.
We can not guarantee their behaviors if we include them into CDF offline.
- ➡ Ariadne / Grappa / MCFM / Vecbos / WGRAD / Wbbgen in CDF offline will be dangerous if we include more ME-generators inside CDF offline.

Plan

- 1) We want to include it in the next version.
Actually, we have the procedure to run it in the current version.
- 2) For JIMMY underlying event model for HERWIG, will also be available in LesHouchesModule, soon.
- 3) Revise CDF note.