Herwig++ Tutorial

MC Tools for the LHC

Stefan Gieseke

Philip Stephens

Herwig++ Installation

- Download the source from http://www.hep.phy.cam.ac.uk/~gieseke/Herwig++
- Make sure you have a working version of CLHEP (v 1.8) and ThePEG.
- Unpack the Herwig++ distribution.
- Set the environment variables (on lxplus.cern.ch)
  - THEPEGPATH e.g. ~lonnblad/public/local
  - CLHEPPATH e.g. ~lonnblad/public/local
- ./configure; make; make check

A Simple Herwig++ Program

- Below is the simplest Herwig++ program

```
#include "Herwig++/Utils/HerwigRun.h"
using namespace std;
using namespace ThePEG;
using namespace Herwig;
int main(int argc, char **argv) {
   try {
        HerwigRun hw(argc,argv);
        if(hw.isRunMode() && hw.isPreparedToRun()) {
                for(int i=0; i<hw.getN(); i++) {
                        EventPtr e = hw.generateEvent();
                        cout << *e << endl;
                }
        } else if(hw.isRunMode()) { } // flag as an error somehow
   } catch(exception &e) { do something }
   } catch(...) { do something else }
   return 0;
}
```

Makefile for making an executable

```
CC = g++
FLAGS = -ansi -pedantic -Wall -g
SOFLAGS = -fpic -rdynamic

# Change these paths to point to your installations
HERWIGPATH = /usr/local
THEPEGPATH = /usr/local
CLHEPPATH = /usr/local
INCLUDE = -I$(THEPEGPATH) -I$(HERWIGPATH) -I$(CLHEPPATH)/include
SOURCES := $(wildcard *.cc)
OBJECTS := $(SOURCES:.cc=.o)
PROGS := *** Your program here ***

HWLIBS = -lHwHadronization -lHwUtils -lHwDecay -lHwShower -lHwME
THEPEGLIBS = -lThePEGRepo -lThePEGHandlers -lThePEGPDF -lThePEGME -lThePEGEvent -lThePEGPDT -lThePEGSM –lThePEGInter \
        -lThePEG -lThePEGMEQCD
DLLIB = -ldl
CLHEPLIB = -lCLHEP
LIBS = -L$(THEPEGPATH)/ThePEG/lib -L$(HERWIGPATH)/Herwig++/lib $(THEPEGLIBS) $(HWLIBS) $(DLLIB) $(CLHEPLIB)
RPATH = $(THEPEGPATH)/ThePEG/lib:$(HERWIGPATH)/Herwig++/lib

all: $(PROGS)

%.o: %.cc
        $(CC) -c $(FLAGS) $(INCLUDE) $<

***Your program here ***: $(OBJECTS)
        export LD_RUN_PATH=$(RPATH); $(CC) $(SOFLAGS) -o $@ $(LIBS) $(OBJECTS)
```

Output of Simple Program

- This program will give an event record output. One event looks like

```
*********************************************************************
Event number 1 (id: LEP) performed by SimpleLEPHandler
=============================================

--- Colliding particles:
 1       e-      11  (3,8)
                        0.000     0.000    45.600    45.600     0.001
 2       e+     -11  (4,9)
                        0.000     0.000   -45.600    45.600     0.001
---------------------------------------------------------------------------
Primary sub-process performed by MEee2gZ2qq
--- incoming:
 3       e-      11 [1] (5)
                        0.000     0.000    45.600    45.600     0.000
 4       e+     -11 [2] (5)
                       -0.000     0.000   -45.600    45.600     0.000
--- intermediates:
 5       Z0      23 [3,4] (6,7)
                       -0.000     0.000    -0.000    91.200    91.200
--- outgoing:
 6       c       4 [5]   >7 {+1}
                       39.152     6.146   -22.483    45.585     1.350
 7      cbar     -4 [5]   6> {-1}
                      -39.152    -6.146    22.503    45.595     1.350
---------------------------------------------------------------------------
```

```
Step 1
--- intermediates:
 3       e-      11 [1] (5)
                        0.000     0.000    45.600    45.600     0.000
 4       e+     -11 [2] (5)
                       -0.000     0.000   -45.580    45.580     0.000
 5       Z0      23 [3,4] (6,7)
                       -0.000     0.000     0.020    91.180    91.180
--- final:
 6       c       4 [5]   >7 {+1}
                       39.152     6.146   -22.483    45.585     1.350
 7      cbar     -4 [5]   6> {-1}
                      -39.152    -6.146    22.503    45.595     1.350
 8     gamma     22 [1]
                        0.000     0.000     0.000     0.000     0.000
 9     gamma     22 [2]
                        0.000     0.000    -0.020     0.020     0.000
---------------------------------------------------------------------------
    Sum of momenta:             0         0         0      91.2      91.2
---------------------------------------------------------------------------
Etc…
```

Some Analysis Programs

- Three simply analysis programs
  1. Count average π^+
     - Need to understand ThePEG::Particle
  2. Find average $\mathbf{p_T}$
     - Need to understand CLHEP::LorentzVector
     - Need to understand ThePEG::Lorentz5Vector
  3. Analysis of rapidity
     - Use our simple histogram booking

Count average π^+

```
try {
    HerwigRun hw(argc,argv);
    if(hw.isRunMode() && hw.isPreparedToRun()) {
          long pip = 0;
          for(int i=0; i<hw.getN(); i++) {
                    EventPtr e = hw.generateEvent();
                    tPVector fs = hw.getFinalState();
                    for(tPVector::iterator it = fs.begin();
                        it != fs.end(); it++) {
                             if((*it)->id() == ParticleID::piplus) pip++;
                    }
          }
          cout << "<Npi0> = " << pi0/((double)hw.getN()) << endl;
    } else if(hw.isRunMode()) { } // flag as an error somehow
}
```

Mean p_T

```
try {
    HerwigRun hw(argc,argv);
    if(hw.isRunMode() && hw.isPreparedToRun()) {
          Energy p_evt, p_all = 0;
          for(int i=0; i<hw.getN(); i++) {
                    EventPtr e = hw.generateEvent();
                    p_evt = 0;
                    tPVector fs = hw.getFinalState();
                    for(tPVector::iterator it = fs.begin();
                        it != fs.end(); it++) {
                             p_evt += (*it)->momentum().perp();
                    }
                    p_evt /= fs.size();
                    p_all += p_evt;
          }
          cout << "<pT> = " << p_all/((double)hw.getN())/MeV << " MeV\n";
    } else if(hw.isRunMode()) { } // flag as an error somehow
}
```

Rapidity Analysis

```
try {
    HerwigRun hw(argc,argv);
    if(hw.isRunMode() && hw.isPreparedToRun()) {
          SampleHistogram y(0.0,5.0,1./10.0);
          for(int i=0; i<hw.getN(); i++) {
                    EventPtr e = hw.generateEvent();
                    tPVector fs = hw.getFinalState();
                    for(tPVector::iterator it = fs.begin();
                        it != fs.end(); it++) {
                             if(abs((*it)->rapidity() < 5.0))
                                      y += abs((*it)->rapidity());
                             }
                             if((i+1)%10==0) y.printGnuplot("ydist.dat");
                    }
          }
    } else if(hw.isRunMode()) { } // flag as an error somehow
}
```