1st Terascale Monte Carlo School
21 – 24 April 2008

PHYSICS AT THE TERA SCALE
Helmholtz Alliance

CASCADE worksheet

Hannes Jung
DESY

- **Introduction**
  In the MC school we will use CASCADE as installed via GENSER. However, you can also run CASCADE in a very fresh and standalone mode, which is described in the appendix. The CASCADE manual can be found on:
  http://projects.hepforge.org/cascade/cascade.pdf

- **Login at DESY**
  login as user `schoolxx` to `mcschool.desy.de`
  either via the login panel of the desktop or via
  `ssh -X schoolxx@mcschool.desy.de`

1) Running CASCADE for MC school (via GENSER)

The example code for CASCADE is available under:
`/afs/desy.de/user/m/mccascad/public/cascade`

The description below is also contained in the file `README` in the same directory.

Copy all the files needed to your directory:

```
cd
cd public
mkdir cascade
cd cascade
cp -rp /afs/desy.de/group/alliance/mcg/public/mcschool2008/examples/Cascade/* .
```

to set all the environment variables, type:

```
source /afs/desy.de/group/alliance/mcg/public/mcschool2008/libpath.sh
```

and then type:

```
make example_CascadeToHepMC.exe
```

Now you can run CASCADE with:

```
example_CascadeToHepMC.exe < steer_pp-top
```

The code of `example_CascadeToHepMC.cc` is very simple:

```
#include <iostream>
#include "HepMC/PythiaWrapper.h"
#include "CascadeWrapper.h"
#include "HepMC/IO_HEPEVT.h"
#include "HepMC/IO_GenEvent.h"
#include "HepMC/GenEvent.h"
#include "PythiaHelper.h"
extern "C" {
    extern struct {
       int Nevent;
    } steer1_;
}
#define steer1 steer1_
int main() {
    //........................................HEPEVT
    // CASCADE uses HEPEVT with 4000 entries and 8-byte floating point
    //  numbers. We need to explicitly pass this information to the
    //  HEPEVT_Wrapper.
    //
    HepMC::HEPEVT_Wrapper::set_max_number_entries(4000);
    HepMC::HEPEVT_Wrapper::set_sizeof_real(8);
    //
    //    initPythia();
    //........................................CASCADE INITIALIZATIONS
    //--initialise CASCADE parameters
    call_casini();
    //--read steering file
    call_steer();
    //-- change standard parameters of CASCADE
    call_cascha();
    //-- change standard parameters of JETSET/PYTHIA
    call_pytcha();
    //-- set up for running CASCADE
    call_cascade();
    //-- print result from integration
    call_caend(1);
    //
    //........................................EVENT LOOP
    int Nevent = steer1.Nevent;
    for ( int i = 1; i <= Nevent; i++ ) {
       call_event();      // generate one event with CASCADE
    }
    //........................................TERMINATION
    //  Print out of generated event summary
    call_caend(2);
    // write out some information from Pythia to the screen
    //    call_pystat( 1 );

    return 0;
}
```

- **Explanation of steering file:**
  The steering file should be self explaining.... However, some of the relevant parameters are given here:

```
* ++++++++++++++++ Kinematic parameters +++++++++++++++
*
     'PBE1'     1       0       -7000.   ! Beam energy
     'KBE1'     1       0        2212    ! 2212: proton
     'IRE1'     1       0        1       ! 0: beam 1 has no structure
*                                         ! 1: beam 1 has structure
*
     'PBE2'     1       0        7000.   ! Beam energy
     'KBE2'     1       0        2212    ! 2212: proton
     'IRE2'     1       0        1       ! 0: beam 2 has no structure
*                                         ! 1: beam 2 has structure
* +++++++++++++++ Hard subprocess selection ++++++++++++++++++
*
     'IPRO'     1       0        11      ! (D=1)
*                                        ! 10: Light quarks
*                                        ! 11: Heavy quarks
*                                        ! 102: g g -> Higgs
*
     'IHFL'     1       0         6      ! (D=4) produced flavour for IPRO=11
*                                        ! 4: charm
*                                        ! 5: bottom
*                                        ! 6: top
*
     'PTCU'     1       0        1000.   ! (D=0) p_t **2 cut for process
*
* ++++++++++++ Parton shower and fragmentation ++++++++++++
*
     'NFRA'     1       0        1       ! (D=1) Fragmentation on=1 off=0
*
     'IFPS'     1       0        3       ! (D=3) Parton shower
*                                        ! 0: off
*                                          ! 1: initial state PS
*                                        ! 2: final state PS
*                                        ! 3: initial and final state PS
*
*     'IGLU'     1       0        1001   ! (D=1010)Unintegrated gluon density
*                                        !  1: CCFM old set JS2001
*                                        !  1001: CCFM J2003 set 1
*                                        !  1002: CCFM J2003 set 2
*                                        !  1003: CCFM J2003 set 3
*                                        !  1010: CCFM set A0
*                                        !  1011: CCFM set A0+
*                                        !  1012: CCFM set A0-
*                                        !  1013: CCFM set A1
*                                        !  1020: CCFM set B0
*                                        !  1021: CCFM set B0+
*                                        !  1022: CCFM set B0-
*                                        !  1023: CCFM set B1
```

- **Using the predefined example to generate HEPMC files**

  After you have copied everything from
  `/afs/desy.de/group/alliance/mcg/public/mcschool2008/examples/Cascade`
  you should find a example in your directory:
  `example_CascadeToHepMC.cc`

  Please read and check what is in:
  `example_CascadeToHepMC.cc`

  As an example, insert in the event loop the following to produce first the HEPEVT event record which then is exported to HEPMC (... a bit complicated, but the easiest way using existing code from PYTHIA):

```
for ( int i = 1; i <= Nevent; i++ ) {
     if ( i%50==1 ) std::cout << "Processing Event Number "
                         << i << std::endl;
     call_event();      // generate one event with CASCADE
     // pythia pyhepc routine convert common PYJETS in common HEPEVT
     call_pyhepc( 1 );
     HepMC::GenEvent* evt = hepevtio.read_next_event();
     //
     // add some information to the event
     evt->set_event_number(i);
     evt->set_signal_process_id(11); // IPRO from CASCADE
     // write the event out to the ascii file
     ascii_io << evt;
     //
     // we also need to delete the created event from memory
     delete evt;
  }
```

  Now you can analyse the HEPMC event file, which you have generated with CASCADE. An example how tro read in the HEPMC file and to fill to simple histograms is given under:
  `/afs/desy.de/group/alliance/mcg/public/mcschool2008/examples/HepMC`

  Just create a new directory on your account:

```
cd
cd public
mkdir hepmc
cd hepmc
cp -rp /afs/desy.de/group/alliance/mcg/public/mcschool2008/examples/HepMC/* .
```

  Now `infile` specifies the HEPMC file which you want to analyse. In order to avoid duplicating files and wasting lot of diskspace, just create a soft link:

```
rm infile
ln -s ../cascade/example_MyCASCADE.dat infile
```

  Now run the program `hepmc_analysis.exe` to read the HEPMC file `example_MyCASCADE.dat` and to fill 2 simple ROOT histograms.

  Start ROOT according to the procedure given in the ROOT primer, by selecting then the appropriate `hepmc_histos.root` file. Plot and print the 2 Histograms.

Physics studies:

For all studies following, run CASCADE to produce a HEPMC output file (or use the already generated one, to have enough statistics) and analyse the output with the help of the program `hepmc_analysis.exe` in
`/afs/desy.de/group/alliance/mcg/public/mcschool2008/examples/HepMC`

Please copy the files to your cascade directory, and edit the file `hepmc_analysis.cc` according to your needs.

- **physics to be investigated: $t\bar{t}$ production at the LHC**
  - Process Nr for heavy quark production is: `IPRO=11`. You also need to select, which of the heavy quarks you want to produce, this is done via `IHFL=6` (`top=6, bottom=5 and charm = 4`).
  - find out, where the top quark sits in the event record
    - what are the different entries ?
    - in the program `hepmc_analysis.cc` print a listing of the event record for 5 events (which is already in the code)
    - try to draw the color stings which combine the top quarks with the proton remnants
    - understand how the event record is build, and how to extract information
    - consult the HEPMC primer how to extract infos from the event record

- plot:
  - $p_\perp$and η of top quark
  - calculate $p_\perp$ of $t\bar{t}$ pair
  - charged particle multiplicity in central region for top events ($|\eta| < 2.5$)
  - charged particle multiplicity in central region ($|\eta| < 2.5$) also as function of energy deposit in fwd region. Require summed energy in $6 < \eta < 7$ to be larger than $E_{fwd} > 100, 500, 1000$ GeV

- **studies:**
  - effect of initial & final state PS on $p_\perp$ and η of top quark
  - effect of initial & final state PS on $p_\perp$ of $t\bar{t}$ pair
  - effect of initial & final state PS on charged particle multiplicity ($|\eta| < 2.5$)
  - to switch on/off initial and final state parton shower, user switch `IFPS` in steering file
  - use DGLAP instead of CCFM evolution (via switch ICCFM =0/1 in steering file) and check the effect of the different initial state parton shower evolution on $p_\perp$ and η of top quark and the $p_\perp$ of $t\bar{t}$ pair
  - understand why there is a difference at large $p_\perp$ of $t\bar{t}$ pair
  - use high statistics sample with already generated files available on `/afs/desy.de/group/alliance/mcg/public/mcschool2008/examples/Cascade` to study effect at large transverse momenta.
  - compare your result using DGLAP and CCFM uPDFs with the distribution obtained from MC@NLO (hep-ph/0305252)

- **physics to be investigated: Higgs production at the LHC (if time left, otherwise leave it as a homework exercise)**
  - run Higgs production at LHC `(IPRO=102)`
  - plot pt of Higgs
  - effect of initial and final state PS
  - "jet" (high pt parton) multiplicity

- **investigation of random number generators**
  - CASCADE uses RLUXGO: find a description on the web... cernlib
  - check effect on $p_\perp$ and charged particle multiplicity by changing random number seed
  - check effect on $p_\perp$ and charged particle multiplicity by changing to different luxory levels of the random number generator (what do they mean ?)

Appendix:

- **Installation of CASCADE (in a standalone mode)**
  Skip the following section, if you want to use CASCADE for the Monte Carlo school.
  The source code of CASCADE can be found under
  http://projects.hepforge.org/cascade/
  The installation is described in the README file and consists of 2 steps:

  get PYTHIA 6 in form of a library

  install CASCADE:

  1) Get the source

```
tar xvfz cascade-XXXX.tar.gz
cd cascade-XXXX
```

  2) set environment variables for PYTHIA, CERNLIB
  example (Please change to the proper path of the libraries):
  in csh:

```
setenv CERN_LIBS "/cern/99/lib -lmathlib -lkernlib -lpacklib"
setenv PYTHIA "/home/jung/cvs/pythia6410"
```

  in zsh:

```
export CERN_LIBS="/cern/99/lib -lmathlib -lkernlib -lpacklib"
export
PYTHIA="/afs/desy.de/group/alliance/mcg/public/MCGenerators/pythia6/
416/i586_rhel40/lib"
```

  2) Generate the Makefiles (by default creating shared libraries)

```
./configure --prefix=install-path
```

  if you do not want shared libraries:

  ./configure --disable-shared --prefix=install-path

  3) Compile the binary

```
make
```

  4) Install the execuatable and PDF files

```
make install
```

  4) the execuatble is in bin
  set the path for the updf data files, if different from the default

```
export
PDFPATH=/afs/desy.de/group/alliance/mcg/public/MCGenerators/cascade/
2.0.1/data
```

  run it with:

```
cascade < steer_pp-top
```