PYTHIA 8 — The First Release

Torbjörn Sjöstrand
CERN/PH and
Department of Theoretical Physics, Lund University

PYTHIA 8.040 released on 20 July 2005
Available on Pythia webpage
http://www.thep.lu.se/~torbjorn/Pythia.html
clicking on the “Future” link in the index

What is in it?
On To C++

Currently HERWIG and PYTHIA are successfully being used, also in new LHC environments, using C++ wrappers

A1: Need to clean up!

Q: Why rewrite?

A2: Fortran 77 is limiting Fortran 90

A3: Young experimentalists will expect C++

PYTHIA7 project $\Rightarrow$ ThePEG

Toolkit for High Energy Physics Event Generation

(L. Lönnblad; S. Gieseke, A. Ribon, P. Richardson)

HERWIG++: complete reimplementation

(B.R. Webber; S. Gieseke, A. Ribon, P. Richardson, M. Seymour, P. Stephens, 3 new)

ARIADNE/LDC: to do ISR/FSR showers, multiple interactions

(L. Lönnblad; N. Lavesson)

SHERPA: in C++ from start, partly wrappers to PYTHIA Fortran

(F. Krauss; T. Gleisberg, S. Hoeche, A. Schaelicke, S. Schumann, J. Winter)
PYTHIA8: A fresh start

Problem: PYTHIA7 stalled, no other manpower
Solution?: take a sabbatical and work “full-time”!
(⇒ baseline model, S. Mrenna & P. Skands join later ?)

Tentative schedule:

<table>
<thead>
<tr>
<th>time</th>
<th>date</th>
<th>processes</th>
<th>final states</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1 Sept. 2004</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>1</td>
<td>1 Sept. 2005</td>
<td>LHA-style input</td>
<td>incomplete draft</td>
</tr>
<tr>
<td>2</td>
<td>1 Sept. 2006</td>
<td>a few processes</td>
<td>complete, buggy(?)</td>
</tr>
<tr>
<td>3</td>
<td>1 Sept. 2007</td>
<td>more processes</td>
<td>stable, debugged</td>
</tr>
</tbody>
</table>

…but don’t forget Murphy’s law

Objectives:

- clean up, keep the most recent models
- Les Houches Accord style input central
- independent of ThePEG (or anything else), but
- interface to ThePEG later written by L. Lönnblad (?)
Current PYTHIA8 structure

The User (≈ Main Program)

Pythia

Event process

Event event

ProcessLevel

LHAinit
LHAevent
(PYTHIA 6.3)
(...??)

PartonLevel

TimeShower
SpaceShower
MultipleInteractions
BeamRemnants

HadronLevel

StringFragmentation
MiniStringFrag...
ParticleDecays
(...??)

BeamParticle

Vec4, Random, Settings, ParticleData, StandardModel, ...
### Current PYTHIA8 status

#### Existing classes

<table>
<thead>
<tr>
<th>Process Level</th>
<th>Parton Level</th>
<th>Hadron Level</th>
<th>—</th>
</tr>
</thead>
<tbody>
<tr>
<td>LHAInit</td>
<td>TimeShower</td>
<td>StringFragmentation</td>
<td>Event</td>
</tr>
<tr>
<td>LHAevtnt</td>
<td>SpaceShower</td>
<td>MiniStringFrag...</td>
<td>BeamParticle</td>
</tr>
<tr>
<td>(PYTHIA 6.3)</td>
<td>MultipleInteractions</td>
<td>ParticleDecays</td>
<td>Vec4, Random</td>
</tr>
</tbody>
</table>

#### Missing classes/topics

- Cross section administration
- Phase space selection
- Process matrix elements
- Parton density libraries
- Resonance decays
- ThePEG input (?)
- MI/ISR/FSR interleaving
- colour flow models
- ME/PS matching
- Popcorn baryons
- updated decay tables
- Bose-Einstein
- event analysis routines
- ...and much, much more
## Distribution

Contents of Pythia 8.040 distribution:

<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction (.pdf)</td>
<td>20 pp</td>
</tr>
<tr>
<td>24</td>
<td>Header files (.h)</td>
<td>3,850 lines</td>
</tr>
<tr>
<td>22</td>
<td>Code files (.cc)</td>
<td>14,750 lines</td>
</tr>
<tr>
<td>1</td>
<td>PYTHIA 6.3 file (.f)</td>
<td>71,500 lines</td>
</tr>
<tr>
<td>25</td>
<td>Documentation files (.man)</td>
<td>4,700 lines</td>
</tr>
<tr>
<td>5</td>
<td>Sample main programs (.cc)</td>
<td>870 lines</td>
</tr>
<tr>
<td>3</td>
<td>Input to above</td>
<td>1,380 lines</td>
</tr>
<tr>
<td>1</td>
<td>Makefile</td>
<td>150 lines</td>
</tr>
<tr>
<td>1</td>
<td>pythia8040.tar.gz (all)</td>
<td>1 MB</td>
</tr>
</tbody>
</table>

To get going: download ⇒ gunzip ⇒ tar xvf  
⇒ make ⇒ run test programs(s)

Self-contained, but hooks to external programs for
- hard processes, Les Houches Accord style
- parton distribution functions
- decays (of some particles, e.g. $\tau$, $B^0$, $B^+$)
- random number generators (shared with other programs)
Event generation structure

1) Initialization step
   - select process(es) to study
   - modify physics parameters
   - set kinematics constraints
   - modify generator settings
   - initialize generator
   - book histograms

   ```
   #include "Pythia.h"
   using namespace Pythia8;
   Pythia pythia;
   pythia.readString("command");
   pythia.readFile("command.file");
   pythia.init(idBeamA,idBeamB,eCM);
   ```

2) Generation loop
   - generate one event at a time
   - analyze it (or store for later)
   - add results to histograms
   - print a few events

   ```
   pythia.next();
   pythia.process.list();
   pythia.event.list();
   int id = pythia.event[i].id();
   ```

3) Finishing step
   - print deduced cross-sections
   - print/save histograms etc.

   ```
   pythia.statistics();
   pythia.settings.listChanged();
   ```
Example of a main program

// Test program main06: study pTZ spectrum at the Tevatron.
#include "Pythia.h"
using namespace Pythia8;
int main() {
    Pythia pythia;
    pythia.readString("Pythia6:msel = 11");
    pythia.readString("Pythia6:ckin(1) = 80.");
    pythia.readString("PartonLevel:MI = off");
    pythia.readString("Beams:primordialKTwidth = 2.");
    pythia.init(2212, -2212, 1960.);
    Hist pTZ("dN/dpTZ",100,0.,100.);
    // Begin event loop. Generate event. Skip if error. List first few.
    for (int iEvent = 0; iEvent < 10000; ++iEvent) {
        if (!pythia.next()) continue;
        if (iEvent < 2) pythia.event.list();
        // Loop over particles in event. Find last Z0 copy. Fill its pT.
        int iZ = 0;
        for (int i = 0; i < pythia.event.size(); ++i)
            if (pythia.event[i].id() == 23) iZ = i;
        pTZ.fill( pythia.event[iZ].pT() );
    }
    pythia.statistics();
    cout << pTZ;
    return 0;
}
Outlook

• C++ PYTHIA 8 is coming along •
  ★ Roughly according to three-year plan so far! ★
  ★ ~ 1 sub-subversion per working week (backup) ★
  ★ Slowdown during autumn, then pick up again early 2006 ★
  ★ Release latest sub-subversion every 2–3 months ★
  ★ First useful — but incomplete — version in a year’s time (?) ★
    ★ First production-quality release, 8.100, early 2007 (?) ★
    ★ Debugged and tuned by LHC startup (?) ★
    ★ Overtaking Fortran version usage by 2008–2009 (???) ★

• Early feedback is most welcome •
  ★ Now is the time for any major course changes ★
  ★ In a year’s time the structure will be frozen (?) ★