PYTHIA Event Generator

XI SERC School on Experimental High-Energy Physics
NISER Bhubaneswar
November 07 - 27, 2017

Event Generators Session

Authors:

Torbjorn Sjostrand ,
Stefan Askb
Jesper R. Christiansena
Richard Corkea
Nishita Desaic
Philip Iltend
Stephen Mrennae
Stefan Prestelf,g
Christine O. Rasmussena
Peter Z. Skands

What is PYTHIA?

- ✓ PYTHIA is a program for the generation of high energy physics events in elementary particle collisions
- ✓ Can simulate the collision between particles such as e^-, e^+, proton and anti-proton in various combination
- ✓ Energy range: 10 GeV to 100 TeV
- ✓ Uses perturbative QCD for both low-p_T and high-p_T regions.
- ✓ p_T > 2 GeV the parton scatterings is correctly described by perturbative QCD
- ✓ The parton-parton scattering in the low-pt region is described by multiparton interaction.

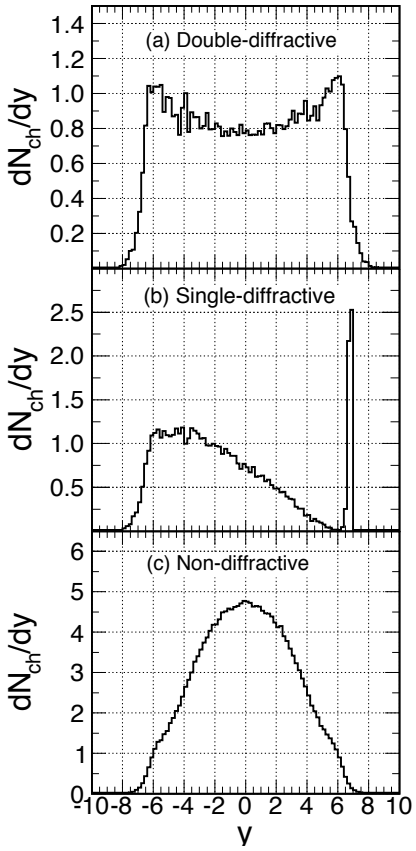
Physics processes in PYTHIA

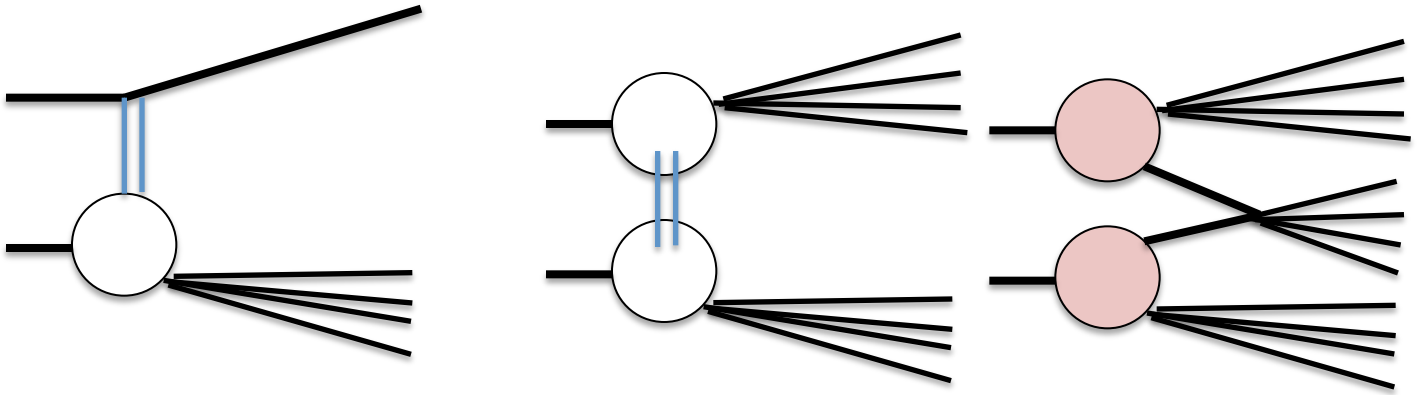
QCD processes

Soft QCD subprocesses

Hard QCD subprocesses

1. Elastic (AB-> AB)
2. Single diffractive (SD) (AB->XB or AB->AX)
3. Double diffractive (DD) (AB->X)
4. Non diffractive (ND)



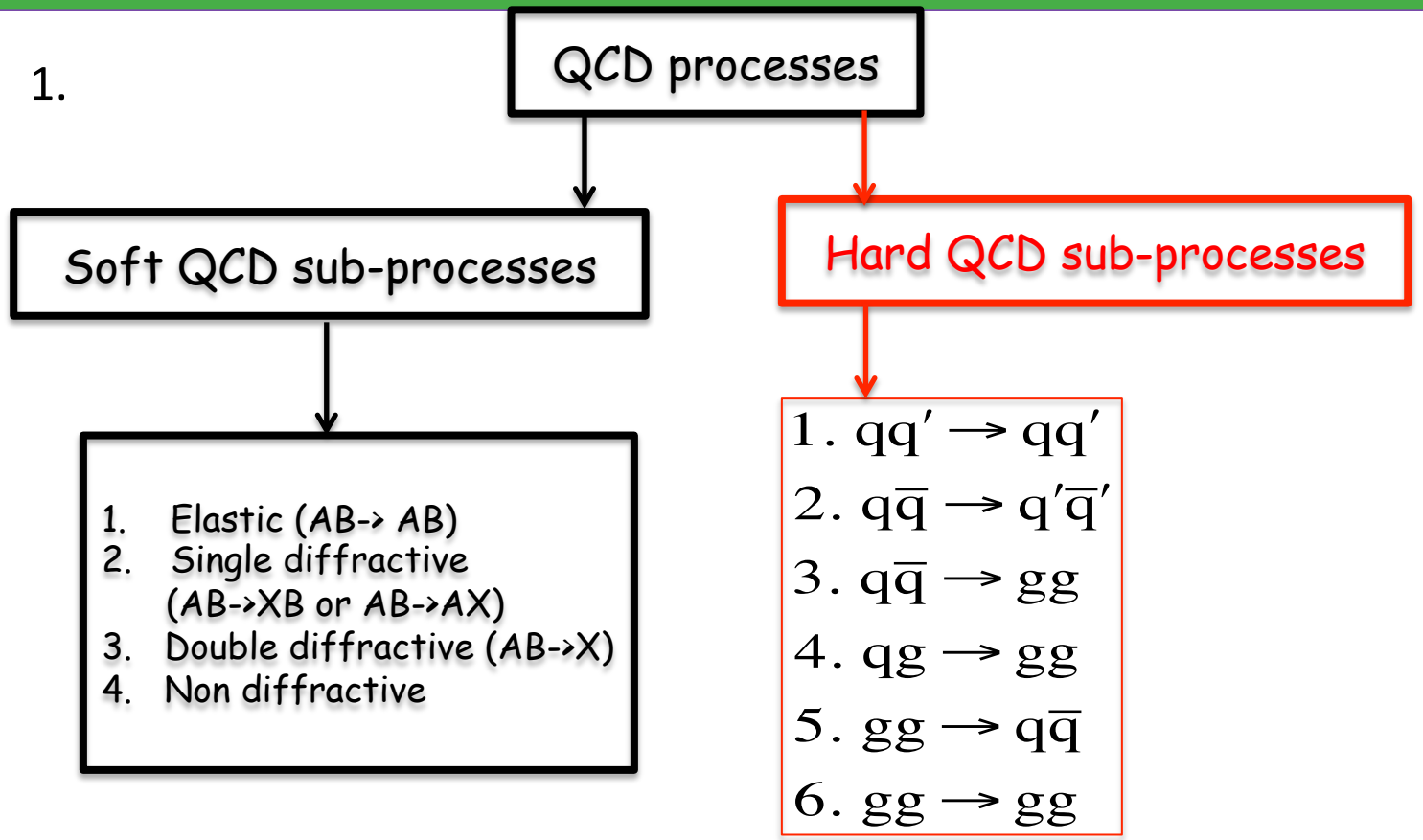


SD DD ND

$$\sigma_{total} = \sigma_{elastic} + (\sigma_{SD} + \sigma_{DD} + \sigma_{ND})$$

Physics processes in PYTHIA

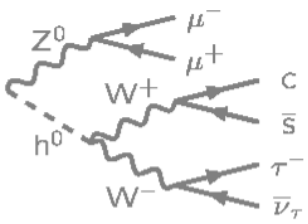
1.



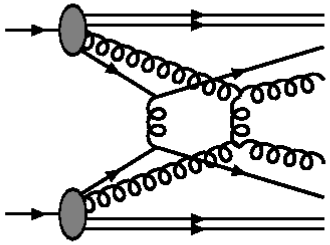
So on

Physics processes in PYTHIA

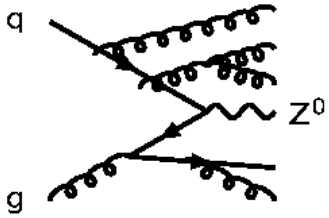
2. Resonance decays(Heavy mass)



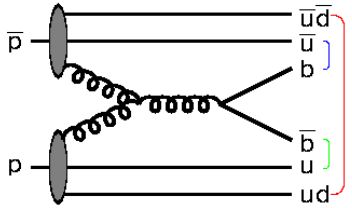
5. Multiple parton-parton interactions(MPI)



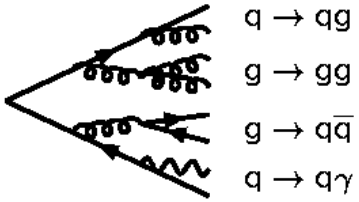
3. Initial-state parton showers(ISR)



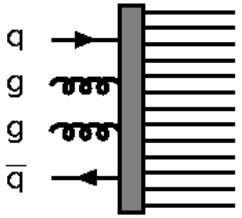
6. Beam remnants



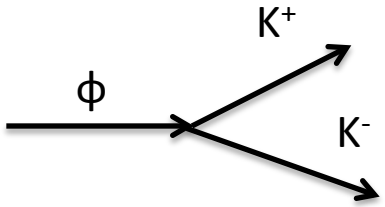
4. Final-state parton showers (FSR)



7. Hadronization



8. Ordinary decays



Installation

-> Go to
http://home.thep.lu.se/Pythia

-> Download the package
pythia8230.tgz to a directory of your choice.

-> In a terminal window,
cd path to the directory containing pythia8230.tgz

tar -xvfz pythia8230.tgz
cd pythia8230
Directories: bin, include, share, src & examples
Files: AUTHORS, CODINGSTYLE, COPYING GUIDELINES, Makefile, README, Makefile.inc
& configure
-> compile
make

Now you are ready to run PYTHIA

Program structure

-> Proper header file must be included:

#include "Pythia8/Pythia.h"

using namespace Pythia8;

-> create a generator object

Pythia pythia;

-> Pythia's settings and particle data

pythia.readString(string);// for changing a single variable

pythia.readFile(fileName); //for changing a set of variables, one per line in the input file.

-> initialize all aspects of the subsequent generation

pythia.init();

-> to generate the next event

pythia.next();

-> run statistics

pythia.stat();

How to run example

//go to the example directory
cd example
Open a file mymain01.cc using an editor

```
#include "Pythia8/Pythia.h"
using namespace Pythia8;
int main() {
// Generator. Process selection. LHC initialization. Histogram.
Pythia pythia;
pythia.readString("Beams:eCM = 7000.");
//pythia.readString("HardQCD:all = on");
//pythia.readString("SoftQCD:all = on");
pythia.readString("SoftQCD:inelastic = on");
// Pick new random number seed for each run, based on clock.
pythia.readString("Random:setSeed = on");
pythia.readString("Random:seed = 0");
pythia.init();
// Begin event loop. Generate event. Skip if error. List first one.
for (int iEvent = 0; iEvent < 100; ++iEvent) {
if (!pythia.next()) continue;
// Extract event classification.
//int code = pythia.info.code();
// Find number of all final charged particles and fill histogram.
int nCharged = 0;
for (int i = 0; i < pythia.event.size(); ++i)
if (pythia.event[i].isFinal() && pythia.event[i].isCharged()) ++nCharged;
} // End of event loop.
  pythia.stat();
  return 0;
}
```

To compile

make mymain01

To run
./mymain01

Main event and particle informations

Event information
Number of produced particles (pythia.event.size())
Process code (pythia.info.code())
No. of Multi Partonic Interaction (pythia.info.nMPI())
and more

The event.list() listing provides the main properties of each particles, by column:

- the index number of the particle
- PDG particle identity code (method id());
- particle name (method name())
- status (method status());
- mothers and daughters (methods mother1(), mother2(), daughter1() and daughter2());
- the colour flow of the process (methods col() and acol());
- the components of the momentum four-vector (px, py, pz, E), in units of GeV with c = 1 (methods px(), py(), pz() and e());
- the mass (method m()).

-------- PYTHIA Event Listing (complete event) ----------

no	id	name	status	mothers	daughters	colours	p_x	p_y	p_z	e	m
0	90	(system)	-11	0 0	0 0	0 0	0.000	0.000	0.000	8000.000	8000.000
1	2212	(p+)	-12	0 0	21 0	0 0	0.000	0.000	4000.000	4000.000	0.938
2	2212	(p+)	-12	0 0	22 0	0 0	0.000	0.000	-4000.000	4000.000	0.938
3	21	(g)	-21	7 0	5 6	102 103	0.000	0.000	0.114	0.114	0.000
4	2	(u)	-21	8 8	5 6	101 0	0.000	0.000	-95.398	95.398	0.000
5	21	(g)	-23	3 4	9 9	101 103	1.192	-0.892	-90.000	90.012	0.000
6	2	(u)	-23	3 4	10 10	102 0	-1.192	0.892	-5.284	5.500	0.330
7	2	(u)	-41	21 21	11 3	102 0	0.000	0.000	0.389	0.389	0.000
8	2	(u)	-42	22 22	4 4	101 0	-0.000	-0.000	-95.398	95.398	0.000
9	21	(g)	-44	5 5	12 13	101 103	1.139	-0.892	-82.242	82.255	0.000
10	2	(u)	-44	6 6	24 24	102 0	-2.124	0.890	-12.042	12.264	0.330

Main event and particle information
(Only final state particles)

Event number
Number of produced particles
Process code (PC)

The event.list() listing provides the main properties of each particles, by column:

- the index number of the particle (Sl. No)
- status (SC);
- PDG particle identity code (PID);
- charge
- the components of the momentum four-vector (px, py, pz, E), in units of GeV with c = 1

Evt No.	No of Final state particle	PC
0	37	101

Sl. No.	SC	PID	Charge	Px	Py	Pz	Energy
0	83	-211	1	0.792496	-0.124813	1.214790	1.462472
1	83	211	1	0.256525	-0.240649	3.067628	3.090880
2	83	-211	1	0.663035	-0.294862	1.500487	1.672573
3	83	323	1	-0.229209	-1.240522	21.880644	21.933297
4	83	-313	0	-0.052649	0.011149	35.977780	35.989572
5	84	111	0	0.014681	-0.079127	13.459230	13.460147
6	84	2112	0	-0.007491	-0.074880	31.640963	31.655000
7	83	-211	1	-0.373604	0.580952	4.230446	4.288734
8	83	111	0	0.483675	-0.278793	5.516121	5.545943
9	83	211	1	0.025505	0.350238	0.471848	0.604514
10	83	-211	1	-0.513858	-0.587210	0.451658	0.912327

compile with root

cd pythia8230

-> compile
./configure --enable-shared --with-root=path to root-installation-directory
make

Now you are ready to run PYTHIA with root library

For problem in compiling (main92.cc):
http://home.thep.lu.se/Pythia/pythia82html/ROOTusage.html

Exercises

1. Download and install Pythia

2. Generate 100000 events

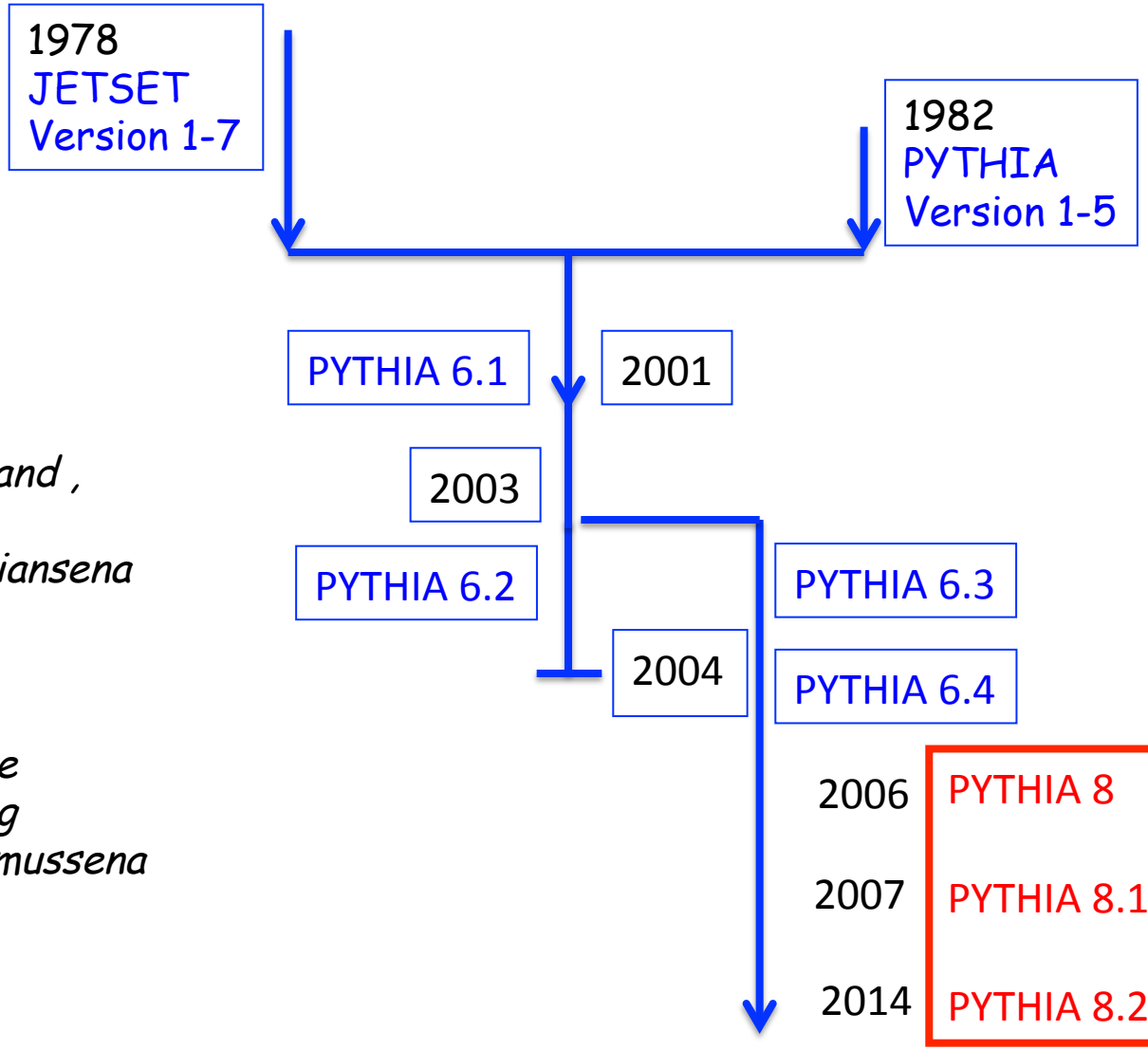
3. Read the output and draw the following distributions like Px, Py, Pz, Pt, eta, phi for charged particles and neutral particle multiplicity distribution.

Backup

PID codes

1	d	11	e^-	21	g	211	π^+	111	π^0	213	ρ^+	2112	n
2	u	12	ν_e	22	γ	311	K^0	221	η	313	K^{*0}	2212	p
3	s	13	μ^-	23	Z^0	321	K^+	331	η'	323	K^{*+}	3122	Λ^0
4	c	14	ν_μ	24	W^+	411	D^+	130	K^0_L	113	ρ^0	3112	Σ^-
5	b	15	τ^-	25	H^0	421	D^0	310	K^0_S	223	ω	3212	Σ^0
6	t	16	ν_τ			431	D^+_s			333	ϕ	3222	Σ^+

Brief history of PYTHIA?



Authors:

Torbjorn Sjostrand ,
Stefan Askb
Jesper R. Christiansena
Richard Corkea
Nishita Desaic
Philip Iltend
Stephen Mrennae
Stefan Prestelf,g
Christine O. Rasmussena
Peter Z. Skands

Where and why event generators are used?