Υπλογστική Φυσική Στοιχεωδών Σωματιδίων

Κ.Κορδάς

Εγκατάσταση και εξοικοίωση με την ΡΥΤΗΙΑ

Κώστας Κορδάς Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Υπολογιστική Φυσική Στοιχειωδών, Μ.Sc Υπολογιστικής Φυσικής, Α.Π.Θ, 8 Μαΐου 2017

Γεννήτορες Monte Carlo (Monte Carlo generators)

 Λογισμικά πακέτα που μας επιτρέπουν να προσομοιώσουμε γεγονότα σαν κι αυτά που έχουμε στα πειράματά μας

- Π.χ., pp → Z , pp → Higgs , etc.

- Χρησιμοποιούν γεννήτορες τυχαίων αριθμών για να να καταλήξουν γεννήσουν ένα γεγονός με συγκεκριμένα χαρακτηριστικά (κάθε σύγκρουση έχει μια πιθανότητα να δώσει κάποιο συγκεκριμένο αποτέλεσμα) → γεννήτορες Monte Carlo
- Πολύ σημαντικά εργαλεία για να σχεδιάζουμε το πείραμά μας και να υπολογίζουμε τι περιμένουμε να δούμε και να εξηγούμε τι είδαμε!
 - Π.χ., Τι είδος, τι ορμή, και τι κατεύθυνση έχουν τα σωματίδια που γεννιούνται; Πρέπει να σχεδιάσω τον ανιχνευτη μου ώστε να είναι ικανός να μετράει καλά τέτοια σωματίδια
 - Π.χ., Με δεδομένο τον ανιχνευτή μου, πόσα γεγονότα περιμένω να παρατηρήσω από την τάδε διαδικασία;

Γεννήτορες Monte Carlo (Monte Carlo generators)

Υπάρχουν γεννήτορες (π.χ., που γεννάνε το διάγραμμα Feynman, πχ., u ubar → Z → e+ e-

χωρίς έξτρα δραστηρίοτητα.

- π.χ., MadGraph

- Και κάποιοι που γεννάνε όχι μόνο αυτά, αλλά και όλα τα σωματίδια που θα βλέπαμε στο πείραμα.
 - Π.χ. PYTHIA η οποία παίρνει τα κουαρκ και γκουλόνια που γεννιούνται και φτιάχνει τα αδρόνια στα οποία αυτά καταλήγουν.
- Για ρεαλιστικές προσομοιώσεις, η πρώτη κατηγορία γεννητόρων χρησιμοποιεί και κάποιο μοντέλλο για την αδρονοποίηση των παραγόμενων κουακ και γκλουονίων, συνήθως την PYTHIA ή τη HERWIG
- Θα εγκαταστήσουμε και θα χρησιμοποιήσουμε την ΡΥΤΗΙΑ

1. Εγκατάσταση ΡΥΤΗΙΑ

 Πηγαίνετε στο http://home.thep.lu.se/~torbjorn/Pythia.html και αριστερά στο Recent

(i home.thep.lu.se/	-torbjorn/Pythia.html						
🥮 Come iniziare 🔊 Ultime notizie 🔹 👼 Red Hat, Inc. 🔜 Red Hat Network 📟 IKEA Suisse. Différent 🐲 ΕΙΔΗΣΕ							
	Important News The webpage of the MCnet School 2017 in Lund, Sweden, 2 - 7 July 2017, is not and techniques used in modern Monte Carlo event generators via a series of lec application is 8 May. See the webpage for further details.						
PYTHIA	Present program version: PYTHIA 8.2						
CPD	While the release of PYTHIA 8.100 marked a clear break with the Fortran-based I evolution. That is, the new version does involve several new features, and some I User-written programs that worked with 8.186 are also likely to work with 8.200, o						
	Given the low threshold, we expect the advantages with the new version to be sut should the need arise, a bug update could still be issued for the 8.1 series.						
Index	The structure of this webpage closely follows the 8.1 one, but does not copy all of presentations and special tutorials.						
Introduction	Look here for information on how to obtain new version alerts, how to contact us ϵ						
Past Recent	See the Introduction for remarks on the VINCIA and DIRE shower plugins.						
Present Future	Installation						
Contact	To get going with the program, do the following (on a Linux or Mac OS X system): • Download the file pythia8226.tgz to a suitable location.						

Κ. Κορδάς - Εγκατάσταση και χρήση ΡΥΤΗΙΑ

- Πηγαίνετε στο Installation. Κατεβάστε το αρχείο pythia8186.tgz
 και ακολουθείστε τις οδηγίες
- Εγώ έφτιαξα έναν φάκελο: % mkdir ~/Programs
- % cd ~/Programs ; mv ~/Downloads/pythia8186.tgz .



Recent program version: PYTHIA 8.1

With the release of PYTHIA 8.1, this new C++ version series took over from the older Fortran 77-based PYTHIA 6.4 one as the new standard. It was strongly recommended to move to PYTHIA 8.1 for LHC studies, although for practical reasons the transition was a slow one in the experimental community.

This initial PYTHIA 8.1 release is focussed towards LHC and Tevatron applications, i.e. high-energy p p and pbar p collisions. Also e+ e- and mu+ muannihilation processes may be simulated, but not e.g. e p, gamma p or gamma gamma collisions. This is the major exception where PYTHIA 6.4 still has more to offer. For the rest PYTHIA 8.1 offers many features not found in PYTHIA 6.4, which should make a transition worthwhile.

The release of PYTHIA 8.2 in October 2014 means that PYTHIA 8.1 will not be developed any further, and that bug fixes would result in a new 8.1 subversion only if there are special reasons for it. The main reason for this is that the step from the later 8.1 subversions to 8.2 is so small that we expect most users to take it rather quickly.

Installation

LUND

UNIVERSITY

ndex

Introduction

Past

Recent

Present Future

Contact

Θ/víkn 5-May-2017

PYTHIA

To get going with the program, do the following (on a Linux or Mac OS X system):

- Download the file pythia8186.tgz to a suitable location
- Unzip and expand it with tar xvfz pythia8186.tgz.
- Move to the thus created pythia8186 directory.
- Read the README file in it for installation instructions, and apply them. (If you are not going to link any external libraries, or have any other special demands, you only need to type make.)
- Move to the examples subdirectory and read the README file there for instructions how to do some test runs.
 (Again, if you do not link to external libraries, you only need to type make mainNN followed by ./mainNN.exe > out, where NN is a two-digit number in the range 01 28.)

- Πηγαίνετε στο Installation. Κατεβάστε το αρχείο pythia8186.tgz
 και ακολουθείστε τις οδηγίες
- Φτιάξτε φάκελο: % mkdir ~/Programs (ή: mkdir \$HOME/Programs)
- % cd ~/Programs ; mv ~/Downloads/pythia8186.tgz
- % cd pythia8186
- Στο αρχείο: ~/.bashrc (που εκτελείται αυτόματα όταν ανοίγετε ένα command-line terminal), βάλτε στο τέλος τις εξής γραμμές:
 # PYTHIA:

```
export PYTHIA8=$HOME/Programs/pythia8186
```

```
export PYTHIA8DATA=$PYTHIA8/xmldoc
```

```
export LD_LIBRARY_PATH=$PYTHIA8/lib:$LD_LIBRARY_PATH
```

• Αν δεν έχετε ήδη το setup για τη ROOT στο .bashrc, βάλτε και κάτι τέτοιο:

```
# ROOT precompiled:
export ROOTSYS=$HOME/Programs/root
$ROOTSYS/bin/thisroot.sh
```

- Τώρα ανοίξτε ένα καινούργιο terminal για να ξεκινήσετε παίρνοντας υπ' όψιν τις αλλαγές στο \$HOME/.bashrc
- % cd ~/Programs/pythia8186
 % more README # ~όλα τα πακέτα λογισμικού έχουν τέτοιο αρχείο
 % ./configure --enable-shared
 - % make (ή gmake : ό,τι δουλεύει στο σύστημά σας)

2. Παραδείγματα ΡΥΤΗΙΑ

Παραδείγματα και οδηγίες ΡΥΤΗΙΑ

- % cd ~/Programs/pythia8186/examples
- % less README # άλλο README σε αυτό το φάκελο
- Στο φάκελο αυτό υπάρχουν διάφορα προγράματα-παραδείγματα.
 Οδηγός: ~/Programs/pythia8186/worksheet.pdf
- Επίσης έχετε τις ιστοσελίδες-οδηγό της PYTHIA και τοπικά!

<mark>kordas@kordas-toshiba:~/Programs/pythia8186/examples</mark>\$ ls ~/Programs/pythia8186/htmldoc/Welcome.html /home/kordas/Programs/pythia8186/htmldoc/Welcome.html _

Ανοίξ'τε έναν browser και βάλτε να πάει σ' αυτό το αρχείο:





PYTHIA 8 Index

Program Overview

Frontpage Program Flow Settings Scheme

PYTHIA 8

Welcome to PYTHIA - The Lund Monte Carlo!

PYTHIA 8 is the successor to PYTHIA 6, rewritten from scratch in C++. With the release of PYTHIA 8.1 PYTHIA version, although PYTHIA 6.4 will be supported in parallel with it for some time to come. Speci been enough tested and tuned for it to have reached the same level of reliability as the older one. This t begin to work with the program, however, which is why we encourage a gradual transition to the new ve new physics features in PYTHIA 8.1, that would make use of it more attractive, but also some topics still be used. Further, many obsolete features will not be carried over, so for some backwards compatibility s choice.

Παράδειγμα main01.cc

% cd ~/Programs/pythia8186/examples ; more main01.cc

// main01.cc is a part of the PYTHIA event generator.
// Copyright (C) 2014 Torbjorn Sjostrand.
// PYTHIA is licenced under the GNU GPL version 2, see COPYING for details.
// Please respect the MCnet Guidelines, see GUIDELINES for details.

// This is a simple test program. It fits on one slide in a talk.
// It studies the charged multiplicity distribution at the LHC.

```
#include "Pythia8/Pythia.h"
using namespace Pythia8;
int main() {
  // Generator. Process selection. LHC initialization. Histogram.
  Pvthia pvthia:
  pythia.readString("Beams:eCM = 8000.");
  pythia.readString("HardOCD:all = on");
  pythia.readString("PhaseSpace:pTHatMin = 20.");
  pythia.init();
 Hist mult("charged multiplicity", 100, -0.5, 799.5);
  // Begin event loop. Generate event. Skip if error. List first one.
  for (int iEvent = 0; iEvent < 100; ++iEvent) {</pre>
   if (!pythia.next()) continue;
    // Find number of all final charged particles and fill histogram.
    int nCharged = 0;
    for (int i = 0; i < pythia.event.size(); ++i)</pre>
      if (pythia.event[i].isFinal() && pythia.event[i].isCharged())
        ++nCharged;
   mult.fill( nCharged );
  // End of event loop. Statistics. Histogram. Done.
  pythia.stat();
  cout << mult;
  return 0;
```

Τρέχουμε το παράδειγμα main01.cc

- % cd ~/Programs/pythia8186/examples/
- % make main01
- %./main01.exe # τυπώνει τα πάντα στην οθόνη
- % ./main01.exe > keep_out_main01.out # τυπώνει σε αρχείο
- % more keep_out_mani01.out η % less keep_out_mani01.out
- Δείχνει τις επι μέρους διαδικασίες με τις οποίες φτιάχνει γεγονότα: gg → gg , gg → q qbar , κλπ.
- Την ενεργό διατομή για κάθε επι μέρους διαδικασία και την ολική ενεργό διατομή
- Τη λίστα των σωματιδίων αν ήταν να ζωγραφίσεις ένα διάγραμμα Feynman με παρτόνια (gluons, quarks), λεπτόνια και τα διανυσματικά μποζόνια
- Και τη λίστα όλων των σωματιδίων: τελικών (πχ., π+ , μ- , κλπ), αλλά και τους γονεις και τα παιδιά του κάθε ενός → μπορείς να δείς πώς η PYTHIA έφτιαξε όλο το γεγονός ("event").

Παράδειγμα main01.cc – event setup

* PYTHIA Process Initialization		** I
We collide p+ with p+ at a CM energy of	of 8.000e+03 Ge\	/
	1	
Subprocess	Code	Estimated
		max (mb)
	l I	
	111	2.223e+00
g g -> q qbar (uds)	112	2.283e-02
q g -> q g	113	1.242e+00
q q(bar)' -> q q(bar)'	114	1.294e-01
qqbar->gg	115	9.369e-04
q qbar -> q' qbar' (uds)	116	4.182e-04
gg->ccbar	121	7.525e-03
q qbar -> c cbar	122	1.386e-04
g g -> b bbar	123	6.766e-03
q qbar -> b bbar	124	1.311e-04
	**	
* End PYTHIA Process Inttlatizat	tion	
* PYTHIA Multiparton Interaction	ns Initializatio	on*
		 I
sigmaNonDiffractive	= 52.12 mb	i
<pre>pT0 = 2.77 gives sigmaInteraction</pre>	= 217.58 mb: a	accepted
Θ/νίκη 5-May-2017 Επά ΡΥΤΗΙΑ Multiparton Interac	<u>ctions Initializ</u> Ισταση και χρήση	ation* PYTHIA

Παράδειγμα main01.cc – hard process

 Monte Carlo partilce ID codes : ψαξτε online (θα τα βρείτε στο PDG)

	PYTHIA	Info Listing												
Beam A: Beam B:	id = 22 id = 22	212, pz = 4.00 212, pz = -4.00	00e+03, e = 00e+03, e =	4.000	0e+03, 0e+03,	m = 9. m = 9.	.383e- .383e-	01. 01.						
In 1: id In 2: id	= 21, = 21,	x = 9.968e-02 x = 3.236e-03	2, pdf = 9. 3, pdf = 1.	518e-(599e+(01 at (01 at s	Q2 = 5. same Q2.	.910e⊣ ∙	+02.						
Subproces It has sh pTh thetah alpha	ssgg-> Hat = 2. Hat = 2. Hat = 2. aEM = 7.	> g g with code .065e+04, t⊦ .431e+01, m3ł .796e+00, phił .702e-03, alpł	e 111 is 2 -: Hat = -2.004 Hat = 0.000 Hat = 5.524 haS = 1.726	> 2. e+04, e+00, e+00. e-01	uHa m4Ha at (at = -6, at = 0, Q2 = 5,	.089e+ .000e+ .910e+	+02, +00, +02.						
Impact pa Max pT so Number of	arameter cale for f MPI =	<pre>b = 6.146e-01 MPI = 2.431e+ 10, ISR =</pre>	1 gives enha +01, ISR = 2 14, FSRpro	ncemer 2.4310 c =	nt fact e+01, I 120, I	tor = 1 FSR = 2 FSRresor	1.884€ 2.431€ n =	2+00. 2+01. 0.						
		THIA Info Listi	(based proce											
	PTIMIA			ess)										
no	id	name	status	MO	thers	daught	ters	CO	Lours	p_x	Р_У	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	3	0	0	0	0.000	0.000	4000.000	4000.000	0.938
2	2212	(p+)	-12	0	0	4	0	0	0	0.000	0.000	-4000.000	4000.000	0.938
3	21	(g)	-21	1	0	5	6	102	101	0.000	0.000	398.735	398.735	0.000
4	21	(g)	-21	2	0	5	6	104	103	0.000	0.000	-12.944	12.944	0.000
5	21	g	23	3	4	0	0	104	101	17.639	-16.728	-0.802	24.323	0.000
6	21	g	23	3	4	0	0	102	103	-17.639	16.728	386.592	387.356	0.000
			Charge s	um: (0.000		Мог	nentum	sum:	0.000	0.000	385.790	411.679	143.686

Παράδειγμα main01.cc – πλήρης λίστα

- Για κάθε σωματίδιο, τελικό ή ενδιάμεσο, δίνονται αν είνι τελικό ή ενδιάμεσο, η μάζα του, η ορμή του , κλπ,
 - και οι γονείς του και τα παιδιά του,
 - κι έτσι μπορούμε να δούμε πώς το κάθε σωματίδιο έχει συμμετάσχει στο να γίνει το πλήρες γεογνός (event)

	PYTHIA	Event Listing	(complete	event)										
no	id	name	status	mot	hers	daugh	ters	co	lours	p_x	P_Y	p_z	е	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	Θ	495	0	0	0	0.000	0.000	4000.000	4000.000	0.938
2	2212	(p+)	-12	0	Θ	496	0	0	0	0.000	0.000	-4000.000	4000.000	0.938
3	21	(g)	-21	7	Θ	5	б	102	101	0.000	0.000	398.735	398.735	0.000
4	21	(g)	-21	8	8	5	б	104	103	0.000	0.000	-12.944	12.944	0.000
5	21	(g)	- 23	3	4	9	9	104	101	17.639	-16.728	-0.802	24.323	0.000
6	21	(g)	-23	3	4	10	10	102	103	-17.639	16.728	386.592	387.356	0.000
7	2	(u)	-41	21	21	11	3	102	0	0.000	-0.000	1590.914	1590.914	0.000
8	21	(g)	- 42	17	17	4	4	104	103	0.000	0.000	-12.944	12.944	0.000
9	21	(g)	- 44	5	5	14	14	104	101	17.537	-16.245	0.043	23.905	0.000
10	21	(g)	- 44	б	б	15	16	102	103	-20.995	32.611	393.062	394.971	0.000
11	2	(u)	-43	7	0	12	13	101	0	3.458	-16.366	1184.864	1184.982	0.330
12	2	(u)	- 51	11	0	25	25	105	0	17.360	-6.655	951.677	951.858	0.330
13	21	(g)	- 51	11	0	20	20	101	105	-13.409	-10.167	233.188	233.795	0.000
14	21	(g)	- 52	9	9	18	19	104	101	17.045	-15.789	0.042	23.234	0.000
15	21	(g)	- 51	10	0	24	24	102	106	-6.638	3.994	215.632	215.771	0.000
16	21	(g)	- 51	10	0	27	27	106	103	-14.356	28.618	176.878	179.752	0.000
17	21	(g)	- 53	22	0	8	8	104	103	-0.000	0.000	-13.497	13.497	0.000

Παράδειγμα main01.cc – απολογισμός γέννησης γεγονότων

Subprocess	Code	Nu	umber of eve	ents	sigma +	- delta
		Tried	Selected	Accepted	(estimat	ed) (mb)
				 65	2 5960 81	1 0040 0
	111	401	05	1 1	2.5800-01	1.9040-02
g g -> q qbar (uus)	112	220	1	1 27	1 4700 01	1 4550 0
q q - 2 q q	115	230	21	21	1.4790-01	1.455e-0
	114	20	/ 0	(0	1.0750-02	0 0000+0
a abar -> a' abar' (uds)	115	0	0	0	0.000000000	0.0000000
q q d q d	121	0	0	0	0.00000+00	0.0000+0
n obar -> c cbar	122	e e	õ	0	0.000e+00	0.000c+0
a -> b bbar	123	2	õ	0	0.000e+00	0.000e+0
q qbar -> b bbar	124	0	õ	õ	0.000e+00	0.000e+0
sum		733	100	100	4.309e-01	2.527e-0
End PYTHIA Event and Cross	Section Statistics · ssages Statistics ·					
times message						
0 no errors or warnings to r	eport					

Παράδειγμα main01.cc – ιστόγραμμα ΡΥΤΗΙΑ

Πόσα τελικά φορτισμένα σωματίδια έχουμε σε κάθε γεγονός;

017-05-07 23:04	charged multiplicity	Hist mul	t("charged	multiplicity", 100, -0.5, 799.5);					
0.404404.0		<pre>// Begin event loop. Generate event. Skip if error. List first one.</pre>							
8.10*10^ 0		for (int	<pre>for (int iEvent = 0: iEvent < 100: ++iEvent) {</pre>						
7.80*10^ 0	XX	if (In	if (Invthia.next()) continue:						
7.50*10^ 0	X X		() Sind surbas of all final shared continions and fill bistory						
7.20*10^ 0	X X	// F UI	u number o	n att ithat charged particles and itte itstogram.					
6.90*10^ 0	X X	int nC	harged = C						
6.60*10^ 0	X X	for (i	nt i = 0;	i < pythia.event.size(); ++i)					
6.30*10^ 0	X X	if (pvthia.eve	<pre>nt[i].isFinal() && pythia.event[i].isCharged())</pre>					
6.00*10^ 0	X XX	++	nCharged:						
5.70*10^ 0	X XX	mult f	ill(ochor						
5.40*10^ 0	X XX	MULC. F	iii nchar	ged);					
5.10*10^ 0	7X7 XX	// End o	f event lo	op. Statistics. Histogram. Done.					
4.80*10^ 0	XXX XX	3							
4.50*10^ 0	XXX XX								
4.20*10^ 0	333 3 33XXX3 XX 3	3X							
3.90*10^ 0	XXX X XXXXXX XX X	XX							
3.60*10^ 0	XXX X XXXXXX XX X	XX							
3.30*10^ 0	XXX X XXXXXX XX X	XX		ό το ιστόχοσμικα είναι δύσκολο					
3.00*10^ 0	XXX X XXXXXX XX X	XXX		στο το τογραμμα είναι ουοκολό.					
2.70*10^ 0	XXX X XXXXXX XX X	XXX							
2.40*10^ 0	XXX X XXXXXX XX X	XXX		Αν εχουμε κουτ μπορουμε					
2.10*10^ 0	XXX X7XXXXXX7XX7X 7 7	XXX777							
1.80*10^ 0	XXX XXXXXXXXXXXXXXXXX X X	XXXXXX		λα φτιαχνουμε κατ. ευθειαν					
1.50*10^ 0	XXX XXXXXXXXXXXXXXXXX X X	XXXXXX							
1.20*10^ 0	3 3XXX3XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	333XXXXXX3		ιστονραμματα ΚΟΟΙ					
0.90*10^ 0	X XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX							
0.60*10^ 0	X XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXX							
0.30*10^ 0	* *********	xxxxxxxxx							
Contents									
*10^ 0	000001014441424458542682412021	1113452221000000	000000000000000000000000000000000000000	000000000000000000000000000000000000000					
*10^-1	000000000000000000000000000000000000000	000000000000000000000000000000000000000	000000000000000000000000000000000000000	000000000000000000000000000000000000000					
*10^-2	000000000000000000000000000000000000000	000000000000000000000000000000000000000	000000000000000000000000000000000000000	000000000000000000000000000000000000000					
*10^-3	000000000000000000000000000000000000000	000000000000000000000000000000000000000	000000000000000000000000000000000000000	000000000000000000000000000000000000000					
Low edge									
*10^ 2	0000000000001111111111122222	22222222233333333	333334444444	44444455555555555555666666666666666677777777					
*10^ 1	001234456788901223456678900123	3445678890122345	66789001234	45678890122345667890012344567889012234566789					
*10^ 0	186420864208642086420864208642	2086420864208642	08642086420	8642086420864208642086420864208642086420					
Entries =	100 Mean = 1.6846e+02	Underflow = 0	.0000e+00	Low edge = -5.0000e-01					
All chan = 1.00	00e+02 Rms = 6.8209e+01	Overflow = 0	.0000e+00	High edge = 7.9950e+02					

Παράδειγμα με ROOT – hist.cc

% cd ~/Programs/pythia8186/rootexamples ; more hist.cc

/ ROOT, for histogramming tinclude "TH1.h"

// ROOT, for interactive graphics
#include "TVirtualPad.h"
#include "TApplication.h"

/ ROOT, for saving file. include "TFile.h" // Create file on which histogram(s) can be saved.
TFile* outFile = new TFile("hist.root", "RECREATE");

```
// Book histogram
```

```
TH1F *mult = new TH1F("mult", "charged multiplicity", 100, -0.5, 799.5);
```

```
// Begin event loop. Generate event; skip if generation aborted.
for (int iEvent = 0; iEvent < 100; ++iEvent) {
    if (!pythia.next()) continue;</pre>
```

// Find number of all final charged particles

```
int nCharged = 0;
for (int i = 0; i < pythia.event.size(); ++i)
if (pythia.event[i].isFinal() && pythia.event[i].isCharged())
++nCharged;
```

```
// Fill charged multiplicity in histogram. End event loop.
mult->Fill( nCharged );
```

```
// Statistics on event generation.
pythia.stat();
```

```
// Show histogram. Possibility to close it.
mult->Draw();
std::cout << "\nDouble click on the histogram window to quit.\n";
gPad->WaitPrimitive();
```

```
// Save histogram on file and close file.
mult->Write();
delete outFile;
```

```
// Done.
return 0:
```

Παράδειγμα με ROOT – hist.cc : output % make hist ; ./hist.exe # και έχουμε το ιστογραμμα



Double click on the histogram window to quit.

Παράδειγμα με ROOT – hist.cc → hist.root

% root hist root # ανοίξτε το αρχείο hist root με τη ROOT

root [1] .ls # δείτε όλα τα αντικέιμενα ROOT στο αρχείο αυτό Ένα αντικέιμενο τύπου TH1F λέγεται mult, το φτιάξαμε στο hist.cc και το αποθηκεύσαμε εδώ στο hist.root

root [2] mult->Draw(); # δείτε το



root [3] .q

Θ/νίκη 5-May-2017

Κ. Κορδάς - Εγκατάσταση και χρήση ΡΥΤΗΙΑ

Παράδειγμα με ROOT TTree – tree.cc

pythia.readString("Beams:eCM = 14000.");
pythia.init();

// Set up the ROOT TFile and TTree.
TFile *file = TFile::Open("pytree.root","recreate");
Event *event = &pythia.event;
TTree *T = new TTree("T","ev1 Tree");
T->Branch("event",&event);

// Begin event loop. Generate event; skip if generation aborted.
for (int iEvent = 0; iEvent < 100; ++iEvent) {
 if (!pythia.next()) continue;</pre>

// Fill the pythia event into the TTree.

// Warning: the files will rapidly become large if all events

// are saved. In some cases it may be convenient to do some

// processing of events and only save those that appear

// interesting for future analyses.

T->Fill();

// End event loop.

// Statistics on event generation.
pythia.stat();

// Write tree
T->Print();
T->Write();
delete file;

// Done.
return 0;

Φυσικά, τα σωστά include files Στην αρχή είναι απαραίτητα:

// File: tree.cc

// This is a simple test program.

// Modified by Rene Brun and Axcel Naumann to put the Pythia::event
// into a TTree.

// Copyright (C) 2014 Torbjorn Sjostrand

// Header file to access Pythia 8 program elements.
#include "Pythia8/Pythia.h"

// ROOT, for saving Pythia events as trees in a file.
#include "TTree.h"
#include "TFile.h"

using namespace Pythia8;

Παράδειγμα με ROOT – tree.cc : TTree T

% Is -larth # I=long, a=all, t=time ordered, r=reverse order, h=human readable info (MBytes)

% root -l pytree.root # -l = μην δείχνεις το εικονίδιο της ROOT

			,
kordas@kordas-toshiba:~/Programs	/pythia8186/rootexamples\$ root -l py	tree.root	
root [0]			
Attaching file pytree.root as _f	lle0		
Warning in <fclass::fclass>: no</fclass::fclass>	dictionary for class Pythia8::Event	ls avallable	
warning in <iclass::iclass>: no</iclass::iclass>	dictionary for class Pythia8::Partic	le is available	
Warning in <iclass::iclass>: no</iclass::iclass>	dictionary for class Pythia8::vec4 i	s available	
warning in <relass::relass>: no</relass::relass>	dictionary for class Pythia8::Juncti		
TEile**			
TFile* pytree.coot		📕 A TIree object,	called I
KEY: TTree T:1 ev1 Tree			
<pre>root [2] T->Print():</pre>			
*****	*****	**************************************	woor k
*Tree :T : ev1 Tree			JWSELK,
*Entries :	22060576 bytes File Size =	7706625	
* : : Tree compr	ession factor = 2.86		
***************************************	******	********* Ολη η πληρα	σφορια
*Branch :event		για κάθε σω	ματίδιο
*Entries : 100 : BranchEler	nent (see below)		ματιστο
	•••••••••••••••••••••••••••••••••••••••	είναι στο er	itry
*Entring : 100 : Total Siz	077 bytes File Size -	101	
*Backets · 1 · Backet Siz	e= 977 Dytes File Size =	1 70	
*	e= 32000 bytes compression=	1.70	
*Br 1:entrv : Inttentr	·v		
*Entries : 100 : Total Siz	e= 27859 bytes File Size =	528 Παρα πολύ τ	τληροφρια
*Baskets : 1 : Basket Siz	e= 32000 bytes Compression=		νίνεται
*	· · · · · · · · · · · · · · · · · · ·		
*Br 2 :entry.idSave : Int_t i	.dSave[entry_]	γρήγορα τε	Οάστιο
*Entries : 100 : Total Siz	e= 683314 bytes File Size =	107235 KOLUD TOOK	τικό
*Baskets : 25 : Basket Siz	e= 32000 bytes Compression=	5.36 και μη πρακ	IINU
* Α/νίκη 5-Μay-2017	Κ Κορδάς - Ενκατάστασ	και χρήση ΡΥΤΗΙΔ	22
O_{1} with J_{1} indy 2011	$\mathbf{R}_{\mathbf{r}}$		22

3. Δικά μας events με ΡΥΤΗΙΑ και ROOT

Δικό μας Ttree με τη ROOT: pp → Z → 2l (1)

- % cd ~/Programs/pythia8186/rootexamples
- Πάρ' τε όλο το tar ball Z_pythia.tar από: http://kordas.web.cern.ch/kordas/Teaching/Diplom/

Δείτε στη σελιδα την κόκκινη γραμμή "Gia to M.Sc Ypologistikns" http://kordas.web.cern.ch/kordas/Teaching/Diplom/Z_pythia.tar

- Κατεβάστο το Z_pythia.tar μέσα στο ~/Programs/pythia8186/rootexamples/
- Κάντε:

% tar xvf Z_pythia.tar # φτιαχνει το φάκελο Z_pythia/ % cd Z_pythia

Ακολουθείστε τις οδηγίες στο README_Ζ

Δικό μας Ttree με τη ROOT: pp → Z → 2l (2)

 Δίνουμε τις αρχικές οδηγίες στην PYTHIA σε ανεξάρτητο αρχείο (κι έτσι ΔΕΝ τις γράφουμε μέσα στον κώδικα που γεννάει τα γεγονότα)

Δείτε τη λίστα των σωματιδίων που παίρνουν μέρος στο hard-scatttering : πολύ μικρή λίστα σε σχέση με το full Event

% make zpythia1 ; ./zpythia1.exe zpythia.cmd

 Δείτε πώς μπορώ να φτιάξω δέντρα με συγκεκριμένη πληροφορία κατά τη διάρκεια γέννησης των events, και να σώσω αυτά τα δέντρα για περαιτέρω ανάλυση σε ένα αρχείο .root

% make zpythia2 ; ./zpythia2.exe zpythia.cmd

Δημιουργείται το αρχείο zpythia.root, το οποίο έχει τα δέντρα

% root -l zpythia.root

root [] .ls

root [] mcInfo->Print()

root [] mcInfo->Draw("mz")

root [] mcInfo->Draw("pidI1","pidI2==1")

Δικό μας Ttree με τη ROOT: pp → Z → 2l (3)

 Μπορούμε πρίν σώσουμε στο Ttree τα γεγονότα που γεννάει η PYTHIA, να τα αλλάξουμε λίγο με χρήση random-number για να τα κάνουμε πιό ρεαλιστικά (π.χ., να βάζουμε ότι τάχα το σημείο που βλέπουμε οτι έγινε η σύγκρουση είναι αποτέλεσμα μέτρησης, και άρα μπορεί να μην έχει μετρηθεί ακριβώς εκεί που έγινε στ' αλήθεια).

% make zpythia ; ./zpythia.exe zpythia.cmd

Δημιουργείται το αρχείο zpythia.root, το οποίο έχει τα δέντρα

• Ανάλυσηση του αρχείου zpythia.root σε δέυτερο στάδιο

% root -l zpythia.root

root [] .ls

root [] leptons->MakeClass("myZcode")

Και γράφετε κώδικα στη μέθοδο Loop() να παίρνετε ό,τι θέλετε από τα events. Π.χ μπορείτε να φτιάχνετε invariant mass of two leptons κλπ...

• Παράδειγμα: το έτοιμο αρχείο zleptons.C και zleptons.h

 \rightarrow Δείτε το αρχείο README_Z για να δείτε πώς το τρέχετε

Πέρα από την ΡΥΤΗΙΑ ή βάζοντας νέα μοντέλα στην ΡΥΤΗΙΑ

🗲 🛈 file:///home/kordas/Progra	ms/pythia8186/htmldoc/Welcome.html 👔 C 🔍 pythia 8 installation 🦻 🏠 🖨 🖶 🎓 🐵 🗸 💟
🥹 Come iniziare 📓 Ultime notizie 🔻	🤜 Red Hat, Inc. 🤜 Red Hat Network 🚥 IKEA Suisse. Différent 🍻 ΕΙΔΗΣΕΙΣ - ΕΛΛΑΔΑ – 🔤 CERN Document Serve 🚞 Support 🔻 🚞 Shop 🔻
Event Information Event Statistics Event Analysis Histograms	MadGraph 5 Processes By far the easiest way to implement new processes into PYTHIA 8 is by using the matrix-element generator MadGraph 5. This program has
Link to Other Programs	an option to output the results of a matrix-element calculation as a set of PYTHIA 8 C++ classes (plus further auxiliary code), that can then be linked and used as semi-internal processes, meaning they are handled identically with normal internal ones. This way, MadGraph 5 can be used to implement processes from any model that can be written in terms of a Lagrangian. Any $2 \rightarrow 1$, $2 \rightarrow 2$ and $2 \rightarrow 3$ processes can be implemented, the limit being set by the absence of efficient phase space generator algorithms for higher multiplicities in PYTHIA. Features such as <i>s</i> -channel resonances are automatically implemented in the process classes. Besides the process library and necessary
Les Houches Accord SUSY Les Houches Accord HepMC Interface	model files, also an example main program is generated for each set of processes, which can be easily modified to perform the desired analyses.
Semi-Internal Processes Semi-Internal Resonances MadGraph 5 Processes	In order to create a PYTHIA 8 process library with MadGraph 5, first download the MadGraph 5 package from https://launchpad.net /madgraph5, and untar the package. You can then specify the location of your pythia81xx directory in the file input/mg5_configuration.txt:
Alpgen Event Interface Matching and Merging	pythia8_path = ./pythia81xx The location can be either relative (to the directory MadGraph5_v_x_x_x/.) or absolute.
CKKW-L Merging Jet Matching UMEPS Merging	For any model that is already implemented in the MadGraph 5 package, you can directly use the model. Start the MadGraph 5 interface bin/mg5, and do:
NLO Merging User Hooks	<pre>import model model_name generate your_process_in_mg5_syntax</pre>