HepSim Monte Carlo repository and integration of its software with key4hep

S.Chekanov (HEP/ANL)

FCC-ee software meeting, CERN

May 30, 2023

What is HepSim?

https://atlaswww.hep.anl.gov/hepsim/

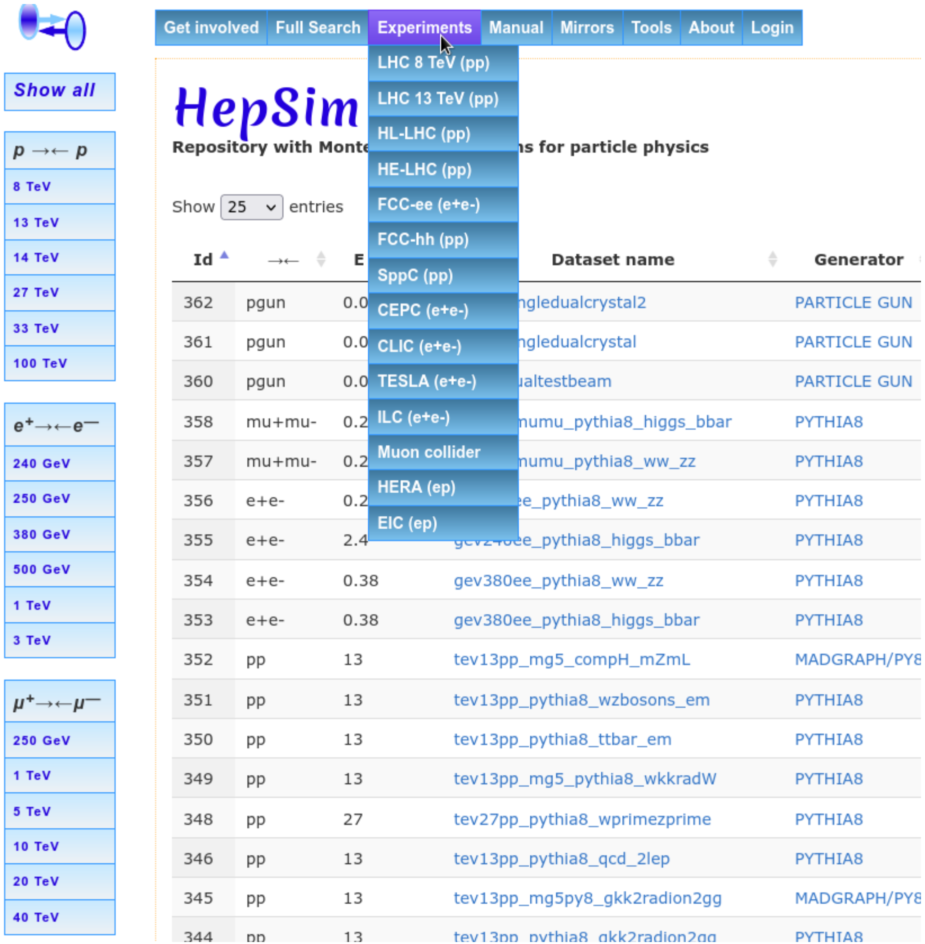
Repository with MC files & software for

- Physics (discovery potential, future precision measurements, etc.)
- Exploration of general aspects of detectors & interactions with material using fast and full Geant4 simulations

Experiment neutral: Can be used for any current & future experiment & phenomenological paper

Organize MC samples for download using collision energy, collision type & physics topics

Event samples assigned to Digital Object Identifiers (DOI) in the form xx.yyyy/zzzzz (see osti.gov link)

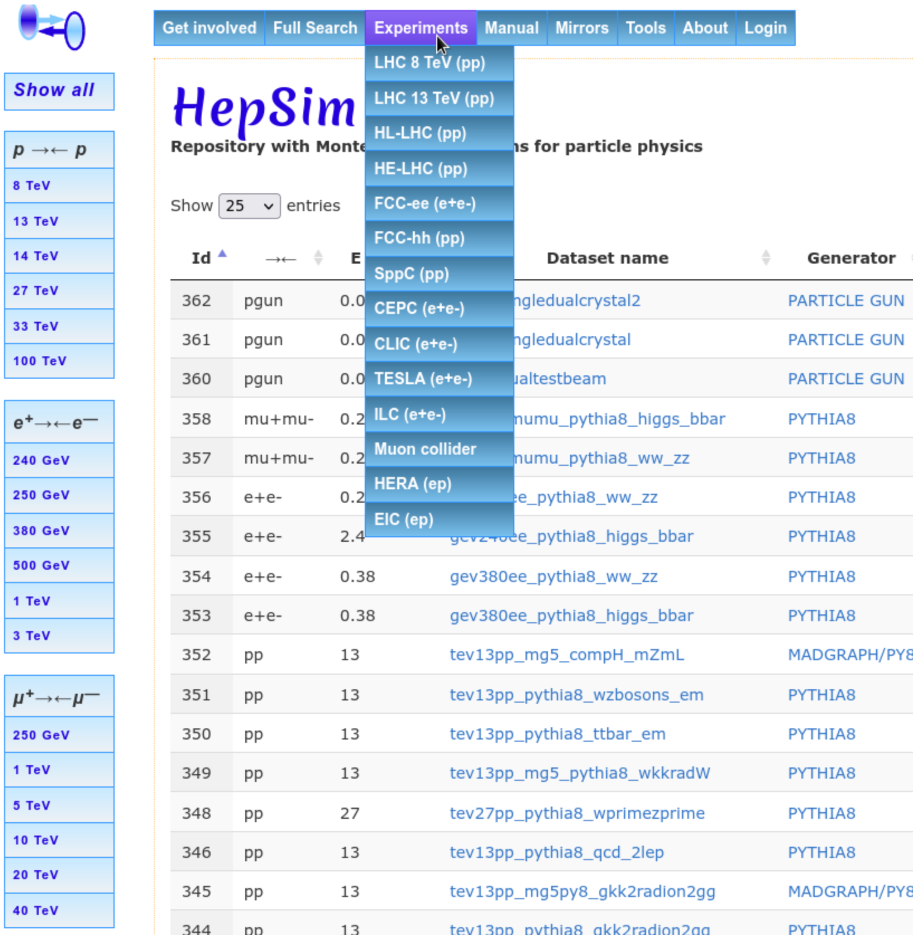


Id	→←	E	Dataset name	Generator
362	pgun	0.0	ngledualcrystal2	PARTICLE GUN
361	pgun	0.0	ngledualcrystal	PARTICLE GUN
360	pgun	0.0	ualtestbeam	PARTICLE GUN
358	mu+mu-	0.2	numu_pythia8_higgs_bbar	PYTHIA8
357	mu+mu-	0.2	numu_pythia8_ww_zz	PYTHIA8
356	e+e-	0.2	e_pythia8_ww_zz	PYTHIA8
355	e+e-	2.4	gev240ee_pythia8_higgs_bbar	PYTHIA8
354	e+e-	0.38	gev380ee_pythia8_ww_zz	PYTHIA8
353	e+e-	0.38	gev380ee_pythia8_higgs_bbar	PYTHIA8
352	pp	13	tev13pp_mg5_compH_mZmL	MADGRAPH/PY8
351	pp	13	tev13pp_pythia8_wzbosons_em	PYTHIA8
350	pp	13	tev13pp_pythia8_ttbar_em	PYTHIA8
349	pp	13	tev13pp_mg5_pythia8_wkkradW	PYTHIA8
348	pp	27	tev27pp_pythia8_wprimezprime	PYTHIA8
346	pp	13	tev13pp_pythia8_qcd_2lep	PYTHIA8
345	pp	13	tev13pp_mg5py8_gkk2radion2gg	MADGRAPH/PY8
344	pp	13	tev13pp_pythia8_gkk2radion2gg	PYTHIA8

What is HepSim?

https://atlaswww.hep.anl.gov/hepsim/

- Consists of a web interface, distributed web storage, command-line tools, event browser, containerized software (docker/singularity image)
- Began at Snowmass DPF 2013 (Top/Higgs, see URL) and evolved to →
- Since 2015 used for physics and detector studies for future experiments (HL-LHC, HE-LHC, FCC, CLIC, CEPC, EIC, etc.) and several ATLAS/LHC papers
- 360 truth-level samples:
  - ~20% converted to Delphes ROOT files
  - ~5% with Geant4 simulation/reconstruction

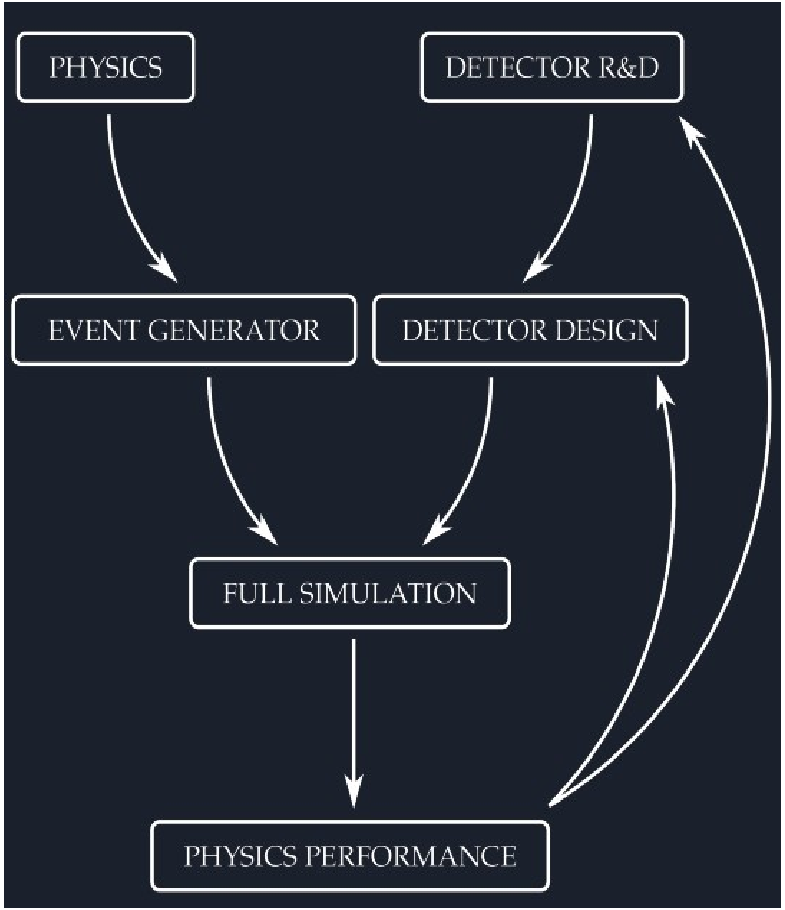


Id	→←	E	Dataset name	Generator
362	pgun	0.0	ngledualcrystal2	PARTICLE GUN
361	pgun	0.0	ngledualcrystal	PARTICLE GUN
360	pgun	0.0	ualtestbeam	PARTICLE GUN
358	mu+mu-	0.2	numu_pythia8_higgs_bbar	PYTHIA8
357	mu+mu-	0.2	numu_pythia8_ww_zz	PYTHIA8
356	e+e-	0.2	ee_pythia8_ww_zz	PYTHIA8
355	e+e-	2.4	gev240ee_pythia8_higgs_bbar	PYTHIA8
354	e+e-	0.38	gev380ee_pythia8_ww_zz	PYTHIA8
353	e+e-	0.38	gev380ee_pythia8_higgs_bbar	PYTHIA8
352	pp	13	tev13pp_mg5_compH_mZmL	MADGRAPH/PY8
351	pp	13	tev13pp_pythia8_wzbosons_em	PYTHIA8
350	pp	13	tev13pp_pythia8_ttbar_em	PYTHIA8
349	pp	13	tev13pp_mg5_pythia8_wkkradW	PYTHIA8
348	pp	27	tev27pp_pythia8_wprimezprime	PYTHIA8
346	pp	13	tev13pp_pythia8_qcd_2lep	PYTHIA8
345	pp	13	tev13pp_mg5py8_gkk2radion2gg	MADGRAPH/PY8
344	pp	13	tev13pp_pythia8_gkk2radion2gg	PYTHIA8

Why HepSim?

https://atlaswww.hep.anl.gov/hepsim/

- **Open access**
  - No authentication for use of event files
- **Preservation of MC data, settings and detectors**
- **Mitigate reproducibility problem in publications**
  - Cite Monte Carlo data using DOI identifies
- **Cache for iterative experiment design process**
- **Analysis using platform-independent software on Linux/Mac/Windows (+ URL data streaming)**



PHYSICS

DETECTOR R&D

EVENT GENERATOR

DETECTOR DESIGN

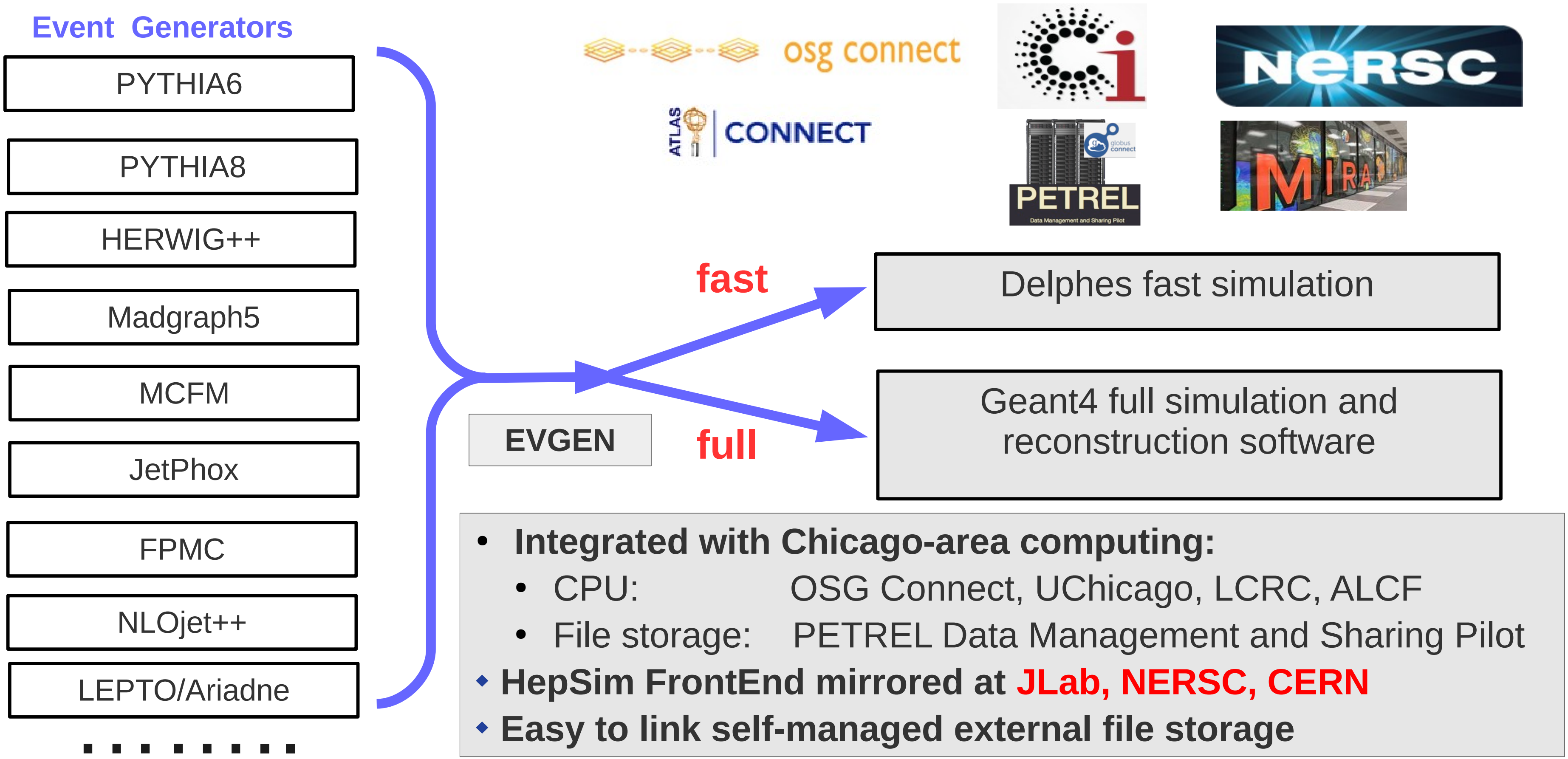
FULL SIMULATION

PHYSICS PERFORMANCE

Credits to W.Armstrong (Physics/ANL)

How it works

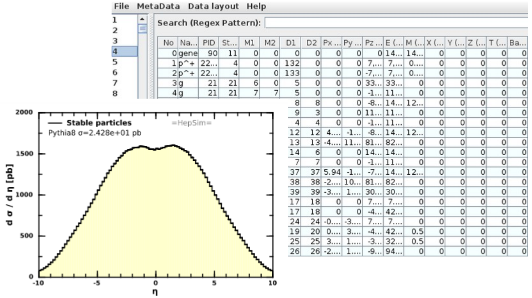
Leveraging large-scale computing



- **Integrated with Chicago-area computing:**
  - CPU: OSG Connect, UChicago, LCRC, ALCF
  - File storage: PETREL Data Management and Sharing Pilot
- **HepSim FrontEnd mirrored at JLab, NERSC, CERN**
- **Easy to link self-managed external file storage**

Software for end user

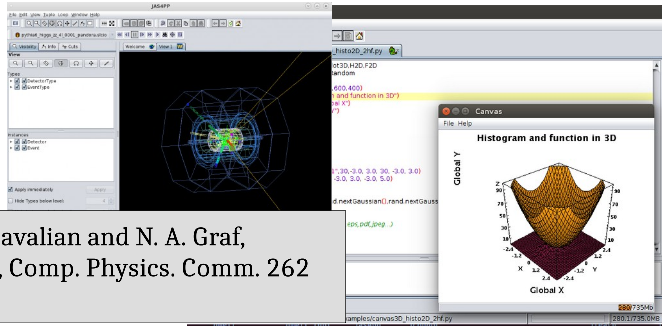
https://atlaswww.hep.anl.gov/hepsim/



	No	Na...	PID	St...	M1	M2	D1	D2	Px ...	Py ...	Pz ...	E (...	M (...	X (...	Y (...	Z (...	T (...	Ba...
1	0	gene	90	11	0	0	0	0	0	0	0	14...	14...	0	0	0	0	0
2	1	p^+	22...	4	0	0	132	0	0	0	7,...	7,...	0....	0	0	0	0	0
3	2	p^+	22...	4	0	0	133	0	0	0	-7,...	7,...	0....	0	0	0	0	0
4	3	g	21	21	6	0	5	0	0	0	33...	33...	0	0	0	0	0	0
5	4	g	21	21	7	7	5	0	0	0	-1...	11...	0	0	0	0	0	0
6							8	8	0	0	-8...	14...	12...	0	0	0	0	0
7							9	3	0	0	11...	11...	0	0	0	0	0	0
8							4	4	0	0	-1...	11...	0	0	0	0	0	0
							12	12	4....	-1...	-8...	14...	12...	0	0	0	0	0
							13	13	-4....	11...	81...	82...	0	0	0	0	0	0
							14	6	0	0	14...	14...	0	0	0	0	0	0
							7	7	0	0	-1...	11...	0	0	0	0	0	0
							37	37	5.94	-1...	-7...	14...	12...	0	0	0	0	0
							38	38	-2....	10...	81...	82...	0	0	0	0	0	0
							39	39	-3....	1....	30...	30...	0	0	0	0	0	0
							17	18	0	0	7....	7....	0	0	0	0	0	0
							17	18	0	0	-4...	42...	0	0	0	0	0	0
							24	24	-0....	-3....	7....	7....	0	0	0	0	0	0
							19	20	0....	3....	-4...	42...	0.5	0	0	0	0	0
							25	25	3....	1....	-3...	32...	0.5	0	0	0	0	0
							26	26	-2....	1....	-9...	94...	0	0	0	0	0	0

Two OS-independent packages

- **(1) hs-toolkit (30 MB)**
  - Discovery and download Monte Carlo files from remote sites
  - Event browser for truth-level events
  - Processing truth-level files and full data-analysis with Lorentz vectors, jet algorithms, etc.
  - Histograms in 2D, 3D, X-Y plotting etc., → vector-graphics images
  - Includes full Python 2.7 API via Jython
- **(2) JAS4PP- *Java Analysis Studio for Particle Physics*** (https://atlaswww.hep.anl.gov/asc/jas4pp/) **(110 MB)**
  - hs-toolkit included
  - User friendly IDE
  - Analysis of detector-level files in LCIO file format
  - ROOT I/O + many physics libraries
  - Full experiment-independent event display



```
lot3D,H2D,F2D
Random

,600,400)
n and function in 3D")
bal X")
Y")

1",30,-3.0, 3.0, 30, -3.0, 3.0)
 -3.0, 3.0, -3.0, 5.0)

d.nextGaussian(),rand.nextGauss

eps,pdf,jpeg...)
```

S.V. Chekanov, G. Gavalian and N. A. Graf, arXiv:2011.05329, Comp. Physics. Comm. 262 (2021) 107857

Event samples for e+e- studies

https://atlaswww.hep.anl.gov/hepsim/

- 250 GeV: 25 events samples (truth level), ~ 30% after full & Delphes simulation
- 380 GeV: 24 event samples (truth level), ~ 20% after full & Delphes studies
- Also some 240 GeV, 500 GeV, 1 TeV samples are available
- See full list:
  - https://atlaswww.hep.anl.gov/hepsim/index.php?c=epem&e=0&t=all

Note on iteration:

- Any external event samples can easily be integrated in HepSim
- Put an event sample on Apache web server and index it as explained in https://atlaswww.hep.anl.gov/hepsim/doc/doku.php?id=hepsim:dev_hepsim
- Low maintenance!

Tutorial using e+e- samples

- How to search and download a file with 10,000 e+e-: H→ bbar process
- What is inside the download file?
- How to fill a few histogram from the truth-level file

Require Java

- How to create Pythia8 events in the HepSim format
- Conversions to ROOT, STDHEP, LCIO, HEPMC formats
- How to create fast Delphes simulations

Require Linux / ROOT / GCC compilers /
/cvmfs/sw.hsf.org/key4hep/

URL link with this tutorial:
https://atlaswww.hep.anl.gov/hepsim/doc/doku.php?id=fcs:fccee:tutorial

FCC-ee HepSim tutorial - 1

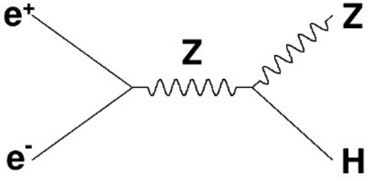
This part of the tutorial does not use any C++ specific libraries and can be done on any computers with Java installed. Check java:

```
java -version
```

Typically, it tells “openjdk version “1.8.0_352” or higher Java version.

```
wget https://atlaswww.hep.anl.gov/hepsim/soft/hs-toolkit.tgz -O - | tar -xz;
source hs-toolkit/setup.sh
```

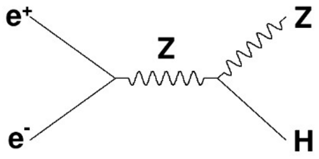
Look at few events: Z→ Z H, where Z→ nunu, and H decays to bbar.



The CM energy is 250 GeV.

The sample is described in https://atlaswww.hep.anl.gov/hepsim/info.php?item=353

FCC-ee HepSim tutorial – 2 part



First, print all files with Higgs processes:

```
# hs-find higgs
```

Then grab the file with H to bbar at e+e-:

```
# hs-ls gev250ee_pythia8_zhiggs_nunubbar
```

Download 10 files (in 2 threads):

```
# hs-get gev250ee_pythia8_zhiggs_nunubbar data 2 10
```

We should have 10 files in the directory “data”. Take a look at a single file. We what to check how many events in the file etc.

```
# hs-info  data/gev250ee_zh_nunubb_001.promc
```

Do you want to print 1st event? Do this:

```
# hs-info data/gev250ee_zh_nunubb_001.promc 1
```

Want to examine the log file? Do this:

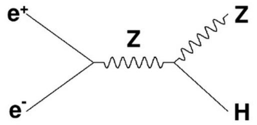
```
# hs-log  data/gev250ee_zh_nunubb_001.promc
```

Let's study each event in the GUI mode (needs X-session!). Start this GUI and click each event number using the left panel:

```
# hs-view  data/gev250ee_zh_nunubb_001.promc
```

Run over this file using Python syntax and make a few simple distributions:
https://atlaswww.hep.anl.gov/hepsim/doc/doku.php?id=fcs:fccee:tutorial#validation

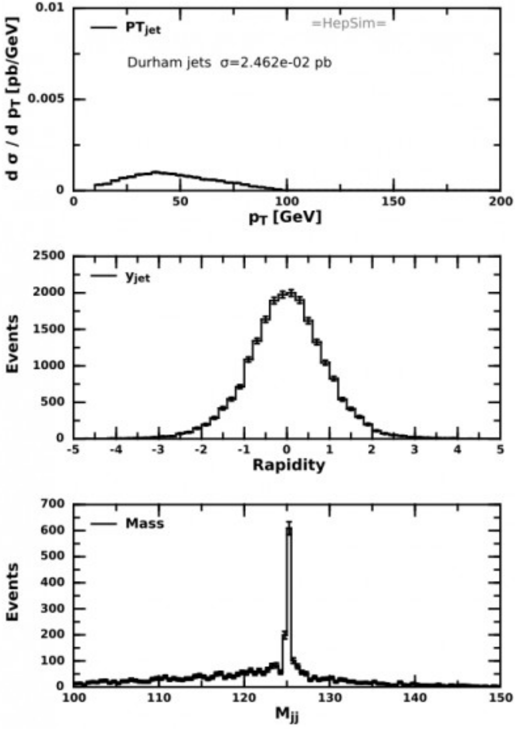
Making a first plot (truth-level events)



https://atlaswww.hep.anl.gov/hepsim/doc/doku.php?id=fcs:fccee:tutorial#validation

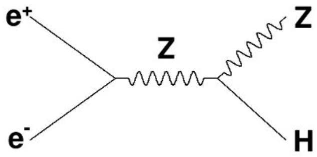
The example is based on ~30 MB *hs-toolkit* package and a script written in in Python (Jython/Java backend)

- Run over local PROMC files or streaming events over the network by specifying dataset name
- Fill histograms using physics libraries with jet algorithms, event shapes, Lorenz vectors etc..
- Export images to vector format



HepSim includes about ~100 scripts for data validation for truth-level samples: https://atlaswww.hep.anl.gov/hepsim/macrolist.php

Delphes fast simulations



Fast simulations: Use the key4hep setup with gcc11 + ROOT:

```
# source /cvmfs/sw.hsf.org/key4hep/setup.sh

# wget http://atlaswww.hep.anl.gov/asc/promc/download/current.php -O ProMC.tgz
# tar -zvxf ProMC.tgz
# cd ProMC
# ./build.sh                      # build all source files
# ./install.sh lib                # install into the "lib" directory
# source lib/promc/setup.sh       # make it available
```

Install Delphes:

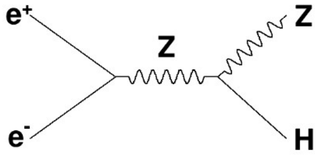
```
# wget http://cp3.irmp.ucl.ac.be/downloads/Delphes-3.5.0.tar.gz
# tar -zxf Delphes-3.5.0.tar.gz
# cd Delphes-3.5.0;  make
```

Run:

```
# ./DelphesProMC  ./cards/delphes_card_CircularEE.tcl \
  ../data/gev250ee_zh_nunubb_001.root  ../data/gev250ee_zh_nunubb_001.promc
```

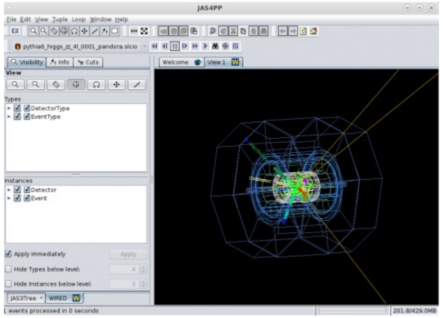
Using Jas4pp IDE

S.V. Chekanov, G. Gavalian and N. A. Graf, arXiv:2011.05329, Comp. Physics. Comm. 262 (2021) 107857



User-oriented desktop application. Download installer: https://atlaswww.hep.anl.gov/asc/jas4pp/ and install as usual desktop application. Click the icon ● and launch it

- Can search & download HepSim truth-level / reco files
- Supports Python, Groovy (faster!) syntax and Java
- Analyze data using the **ProMC (truth-level)** or **LCIO (detector level)** formats (or any native Java/Python IO supported too)
- ROOT-like graphics and large number of physics libraries



pythia6_higgs_zz_4l_0001_pandora.slcio

☑ Apply immediately

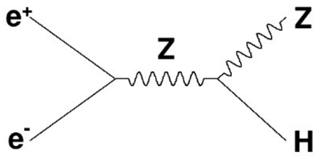
□ Hide Types below level: 4

□ Hide Instances below level: 3

1 events processed in 0 seconds 201.8/429.0MB

+ detector-independent event display (Wired3/Jas3)

Streaming e+e- data over the network



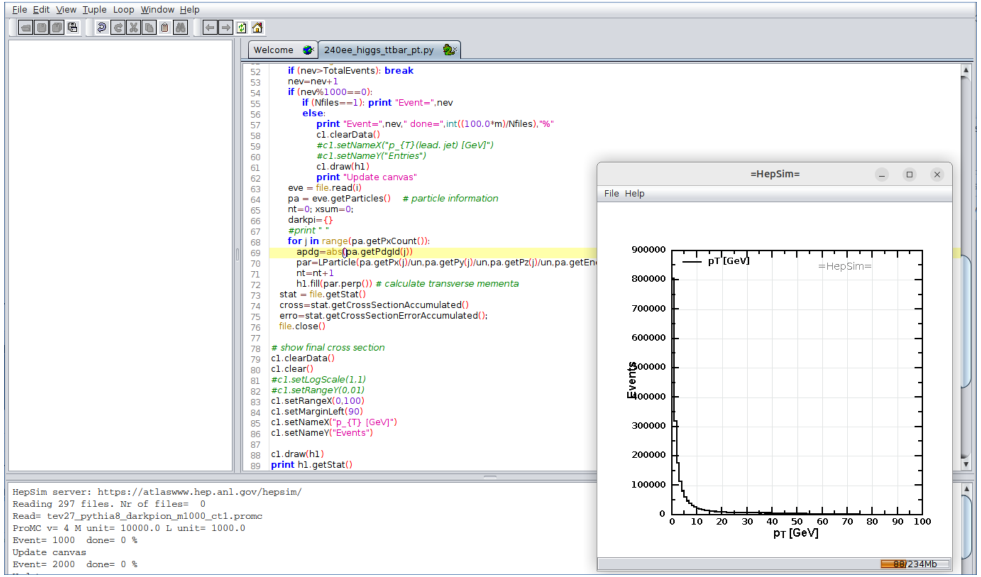
One can also specify HepSim data sample inside a script (without downloading files)

Example for H→ bbar at 240 GeV e+e-. Copy Python code and save as “example.py”
https://atlaswww.hep.anl.gov/hepsim/doc/doku.php?id=fcs:fccee:tutorial#data_streaming

Open analysis macro inside Jas4pp and click “Run”

The e+e- data will be streamed, pT-distribution plot will be updated in real time.
→ Used for validation tasks

Note: Data streaming is also supported using minimalistic hs-toolkit package



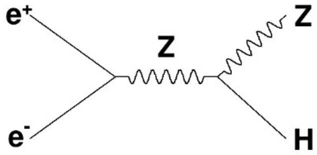
```
    if (nev>TotalEvents): break
    nev=nev+1
    if (nev%1000==0):
        if (Nfiles==1): print "Event=",nev
        else:
            print "Event=",nev," done=",int((100.0*m)/Nfiles),"%"
            c1.clearData()
            #c1.setNameX("p_{T}(lead. jet) [GeV]")
            #c1.setNameY("Entries")
            c1.draw(h1)
            print "Update canvas"
    eve = file.read(i)
    pa = eve.getParticles()   # particle information
    nt=0; xsum=0;
    darkpi={}
    #print " "
    for j in range(pa.getPxCount()):
      apdg=abs(pa.getPdgId(j))
      par=LParticle(pa.getPx(j)/un,pa.getPy(j)/un,pa.getPz(j)/un,pa.getEn
      nt=nt+1
      h1.fill(par.perp()) # calculate transverse mementa
  stat = file.getStat()
  cross=stat.getCrossSectionAccumulated()
  erro=stat.getCrossSectionErrorAccumulated();
  file.close()

# show final cross section
c1.clearData()
c1.clear()
#c1.setLogScale(1,1)
#c1.setRangeY(0,01)
c1.setRangeX(0,100)
c1.setMarginLeft(90)
c1.setNameX("p_{T} [GeV]")
c1.setNameY("Events")

c1.draw(h1)
print h1.getStat()
```

```
HepSim server: https://atlaswww.hep.anl.gov/hepsim/
Reading 297 files. Nr of files=  0
Read= tev27_pythia8_darkpion_m1000_ct1.promc
ProMC v= 4 M unit= 10000.0 L unit= 1000.0
Event= 1000  done= 0 %
Update canvas
Event= 2000  done= 0 %
```

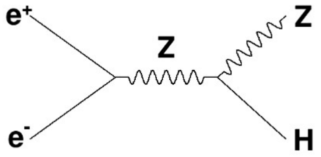
Analyzing LCIO files (detector-level)

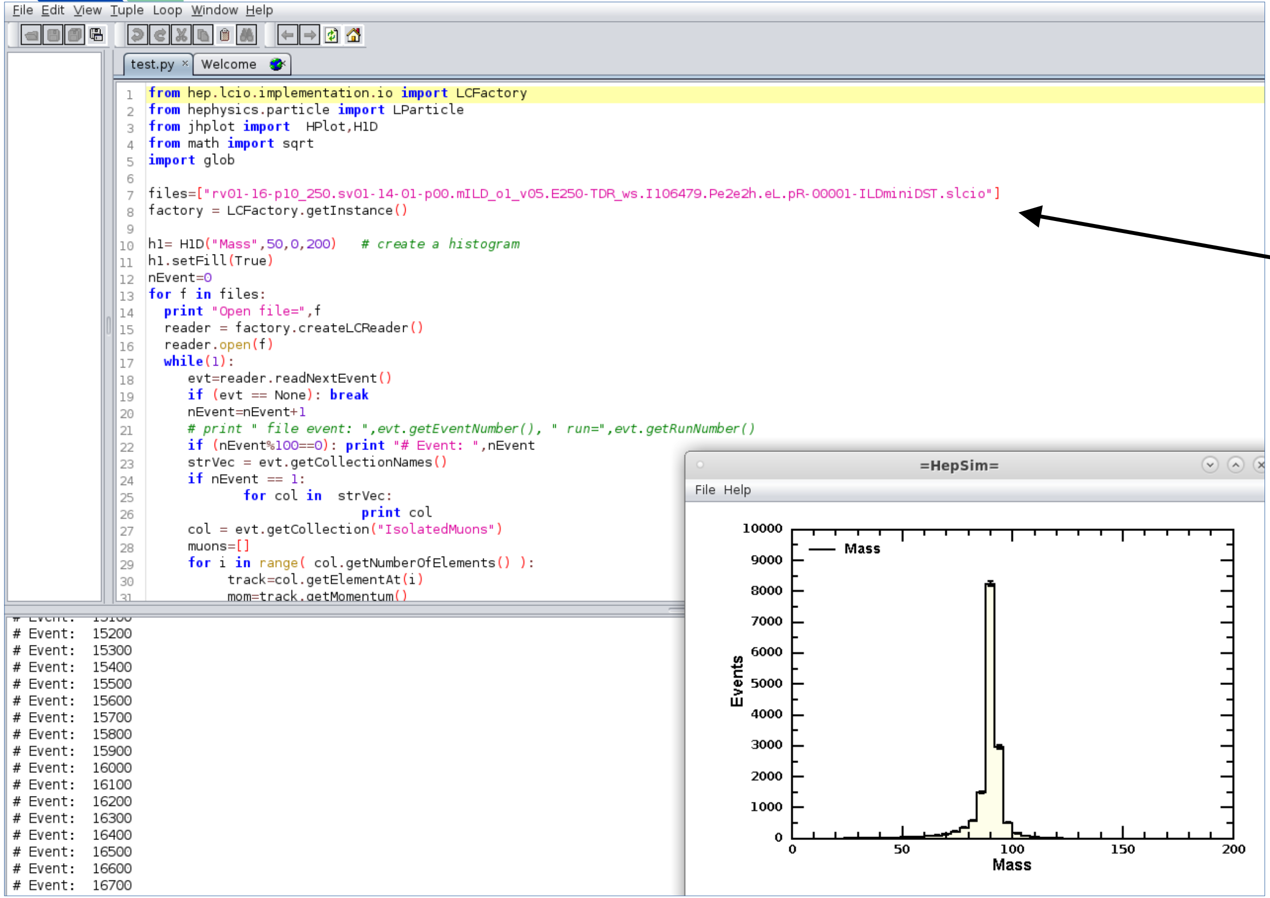


LCIO truth and detector-level files (See the talk LCIO turns 20: N.Graf CHEP2023 [PDF talk]) can also be analyzed in Python scripts using Java backend (or Java)

- LCIO files can be download from HepSim using “detector tags”.
- Collection of Python scripts for LCIO input files listed in:
  - https://atlaswww.hep.anl.gov/hepsim/macrolist_lcio.php
- Many *.py and *.java scripts located in the directory “examples” of *Jas4pp* installation
  - Shows how to analyze tracks, hits, calorimeter clusters using LCIO files
- miniDST files from ILD detector (also SID and CLIC) can be used too:
  - Full analysis e+e- example running in jas4pp: https://atlaswww.hep.anl.gov/asc/wikidoc/doku.php?id=asc:jas4pp#reading_minidst_files

Z peak created using miniDST LCIO input





```python
from hep.lcio.implementation.io import LCFactory
from hephysics.particle import LParticle
from jhplot import  HPlot,H1D
from math import sqrt
import glob

files=["rv01-16-p10_250.sv01-14-01-p00.mILD_o1_v05.E250-TDR_ws.I106479.Pe2e2h.eL.pR-00001-ILDminiDST.slcio"]
factory = LCFactory.getInstance()

h1= H1D("Mass",50,0,200)   # create a histogram
h1.setFill(True)
nEvent=0
for f in files:
  print "Open file=",f
  reader = factory.createLCReader()
  reader.open(f)
  while(1):
     evt=reader.readNextEvent()
     if (evt == None): break
     nEvent=nEvent+1
     # print " file event: ",evt.getEventNumber(), " run=",evt.getRunNumber()
     if (nEvent%100==0): print "# Event: ",nEvent
     strVec = evt.getCollectionNames()
     if nEvent == 1:
            for col in  strVec:
                          print col
     col = evt.getCollection("IsolatedMuons")
     muons=[]
     for i in range( col.getNumberOfElements() ):
          track=col.getElementAt(i)
          mom=track.getMomentum()
```

```
# Event:  15100
# Event:  15200
# Event:  15300
# Event:  15400
# Event:  15500
# Event:  15600
# Event:  15700
# Event:  15800
# Event:  15900
# Event:  16000
# Event:  16100
# Event:  16200
# Event:  16300
# Event:  16400
# Event:  16500
# Event:  16600
# Event:  16700
```

List with SLCIO miniDST files from the ILD project

Link to manual with Python code

Ideas related to integration with FCC-ee (key4hep) software. Part I

- ProMC C++ library (https://atlaswww.hep.anl.gov/asc/promc/) can be put under key4hep. It allows reading HEPSIM files in C++ and develop the needed converters ProMC→ EDM (truth) and back, i.e. promc2edm and edm2promc
- Delphes should be recompiled after setting up $PROMC environmental variable. This variable is set automatically after running ProMC "setup.sh" scrip.
  - Compiling Delphes with $PROMC env variable creates "**DelphesProMC"** binary file
- Few other converters should be created:
  - LCIO→ EDM and EDM→LCIO
  - STDHEP→EDM, HEPMC2→EDM (to support legacy files from ILC)
- If EDM files are publicly stored (under Apache web server), HepSim Web interface can link such files and make them searchable & downloadable

Ideas related to integration with FCC-ee (key4hep) software. Part II

- The rest of HepSim software is Java-centric. Does not require compilation.
  To make it fully available in ke4hep, we need:
  - JDK18 (or above)
  - hs-toolkit (https://atlaswww.hep.anl.gov/hepsim/doc/doku.php?id=hepsim:hs-tools) used to discover files, validate files and process
- Jas4pp is optional (it is a desktop application). Requires X-window and thus it is easier to run it on a desktop

Thanks!

For more information, see the HepSim web manual and **hs-help** on the command line.

HepSim manual: https://atlaswww.hep.anl.gov/hepsim/doc/

HepSim contributors:
https://atlaswww.hep.anl.gov/hepsim/doc/doku.php?id=hepsim:contributions

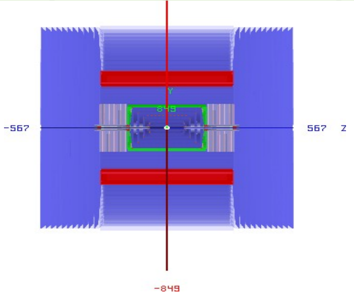
HepSim public results:
https://atlaswww.hep.anl.gov/hepsim/doc/doku.php?id=hepsim:public

~30 articles. Contributions to CEPC, CLIC, FCC-hh etc. conceptual design reports

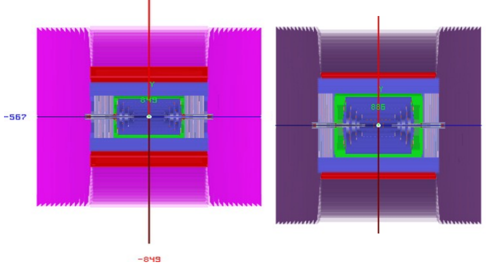
Backup

‘All-silicon’ design concepts supported in HepSim

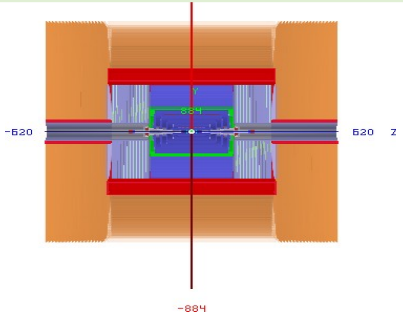
SiD (SiD LO3)
($e^+ e^-$ up to 1 TeV)



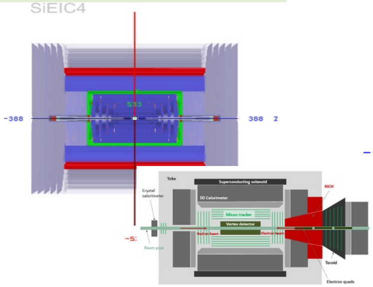
SiCPEC, SiDB
($e^+ e^-$ 250 GeV)



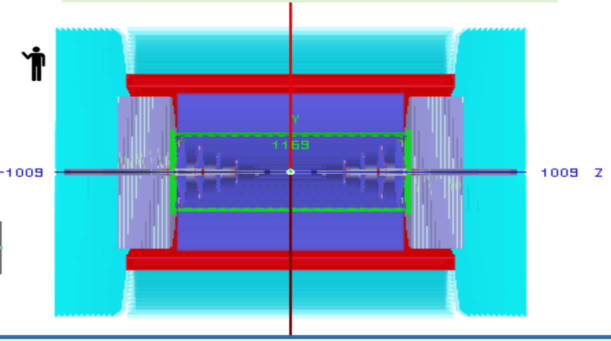
CLIC-SiD (CDR)
($e^+ e^-$ up to 3 TeV)



SiEIC, TopSide
(ep, 35-141 GeV)



SiFCC + 7 variations
(FCC-hh, pp 100 TeV)

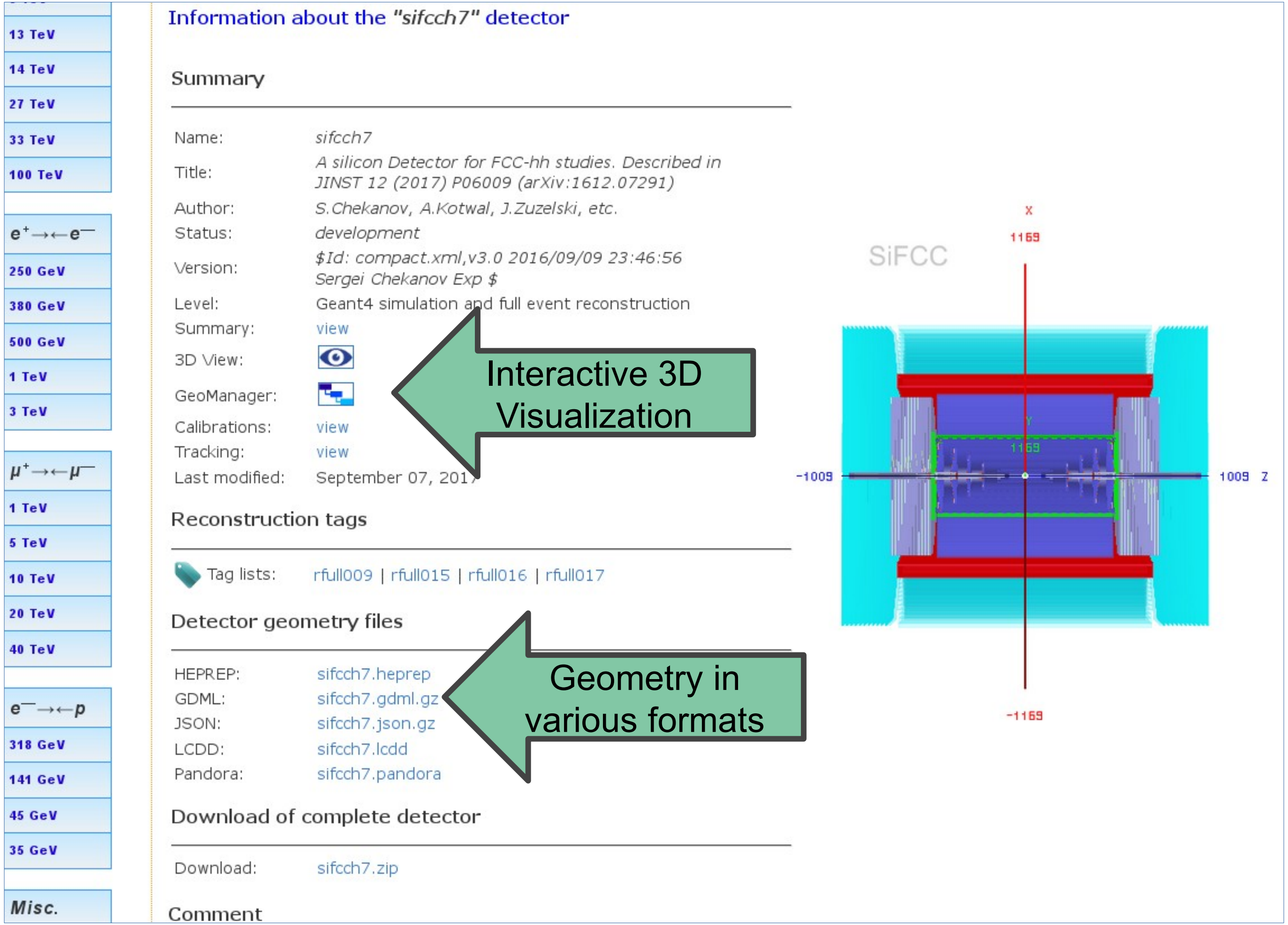


Performance detectors:

- Physics reach studies using Geant4 simulations & full reconstruction
- Playground for various technologies and detector optimizations
- Fast turnover to modify detector & create events samples

Share similar design, but differ in sizes, calorimeter readouts etc

Interfaced with common Monte Carlo samples



13 TeV
14 TeV
27 TeV
33 TeV
100 TeV

$e^+ \rightarrow\leftarrow e^-$
250 GeV
380 GeV
500 GeV
1 TeV
3 TeV

$\mu^+ \rightarrow\leftarrow \mu^-$
1 TeV
5 TeV
10 TeV
20 TeV
40 TeV

$e^- \rightarrow\leftarrow p$
318 GeV
141 GeV
45 GeV
35 GeV

Misc.

Information about the "*sifcch7*" detector

Summary

Name:	*sifcch7*
Title:	*A silicon Detector for FCC-hh studies. Described in JINST 12 (2017) P06009 (arXiv:1612.07291)*
Author:	*S.Chekanov, A.Kotwal, J.Zuzelski, etc.*
Status:	*development*
Version:	*$Id: compact.xml,v3.0 2016/09/09 23:46:56 Sergei Chekanov Exp $*
Level:	Geant4 simulation and full event reconstruction
Summary:	view
3D View:	👁
GeoManager:	
Calibrations:	view
Tracking:	view
Last modified:	September 07, 2017

Interactive 3D Visualization

Reconstruction tags

Tag lists: rfull009 | rfull015 | rfull016 | rfull017

Detector geometry files

HEPREP:	sifcch7.heprep
GDML:	sifcch7.gdml.gz
JSON:	sifcch7.json.gz
LCDD:	sifcch7.lcdd
Pandora:	sifcch7.pandora

Geometry in various formats

Download of complete detector

Download: sifcch7.zip

Comment

Web Interface - Detector

Main event file format ProMC and ProIO

http://atlaswww.hep.anl.gov/asc/promc/ and https://github.com/proio-org

- Archive-style self-described format to keep MC events - **ProMC**:
  - Event records, NLO,original logfiles, PDG tables etc.
- 30% smaller files than any existing formats after compression

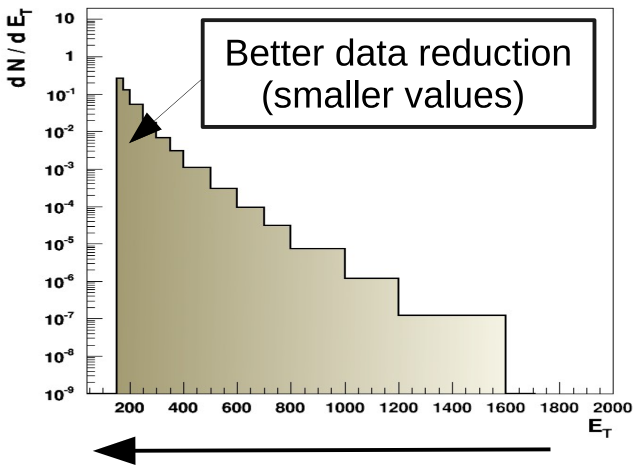
protobuf

Protocol Buffers - Google's data interchange format

Google's Protocol buffers

Number of used bytes depends on values.
Small values use small number of bytes

8-bytes → varint



Better data reduction (smaller values)

compression strength keeping precision of representation constant

- Effective file size reduction for pile-up events where particles with small momenta use less bytes
- Separate events can be streamed over the Internet
- Other HepSim formats: **ROOT, LHE** and **LCIO** (full simulation)

ProMC: S.C., E.May, K. Strand, P. Van Gemmeren, Comp. Physics Comm. 185 (2014), 2629

NLO QCD calculations as “ntuples”

Theorists can use it too!

S.C. Adv. High Energy Physics, vol. 2015, 13609

- Stored several NLO QCD calculations (MCFM, JETPHOX, NLOjet++)
- Data structure is different compared to full parton-shower MC
  - “Particle record”: Usually 4-momenta of 3-4 particles per events
  - “Event record”: includes “weights” and deviations from central weights for different PDF eigenvector sets for calculations of PDF uncertainties

$$\boxed{w_n = \left[1000 \times (1 - \frac{PDF(n)}{PDF(0)})\right]}$$

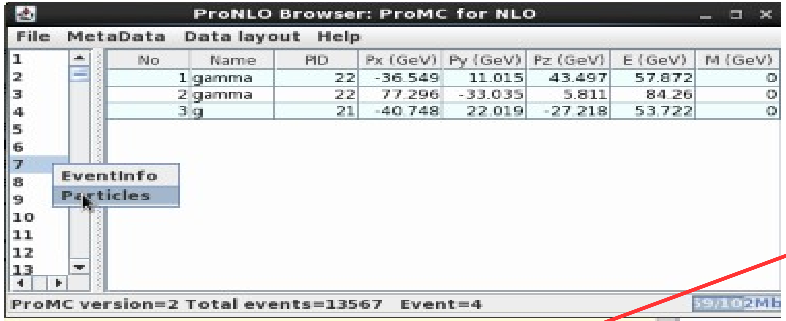
N=1...51, for CT10 PDF

Weighted NLO events can be compactly stored using Google ProtocolBuffers:
→ *double precision “weights” → int64 varint (deviations) → 2 bytes per weight*
→ *Large deviations are stored using 4 or 8 bytes (rarely)*

NLO QCD calculations as “ntuples” for HEP experiments

MCFM prediction for H(→ γγ)+jet (pp 100 TeV)
“higgsjet_gamgam_mcfm” sample

Some NLO samples using MCFM
have been created on Mira supercomputer
(BlueGene/Q)



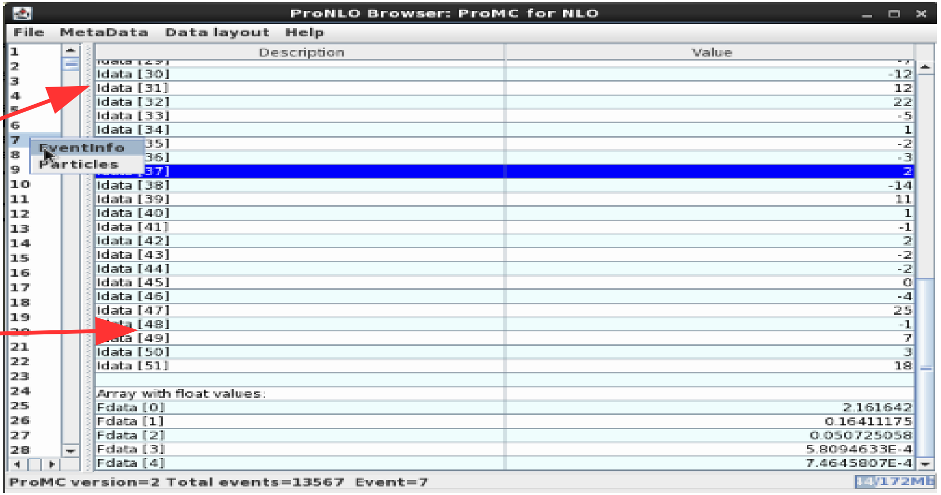
No	Name	PID	Px (GeV)	Py (GeV)	Pz (GeV)	E (GeV)	M (GeV)
1	gamma	22	-36.549	11.015	43.497	57.872	0
2	gamma	22	77.296	-33.035	5.811	84.26	0
3	g	21	-40.748	22.019	-27.218	53.722	0

ProMC version=2 Total events=13567 Event=4

4-momenta of particles

Event weights

PDF variations for CT10 (51)



Description	Value
Idata [30]	-12
Idata [31]	12
Idata [32]	22
Idata [33]	-5
Idata [34]	1
35]	-2
36]	-3
[37]	2
Idata [38]	-14
Idata [39]	11
Idata [40]	1
Idata [41]	-1
Idata [42]	2
Idata [43]	-2
Idata [44]	-2
Idata [45]	0
Idata [46]	-4
Idata [47]	25
ta [48]	-1
ata [49]	7
Idata [50]	3
Idata [51]	18
Array with float values:	
Fdata [0]	2.161642
Fdata [1]	0.16411175
Fdata [2]	0.050725058
Fdata [3]	5.8094633E-4
Fdata [4]	7.4645807E-4

ProMC version=2 Total events=13567 Event=7