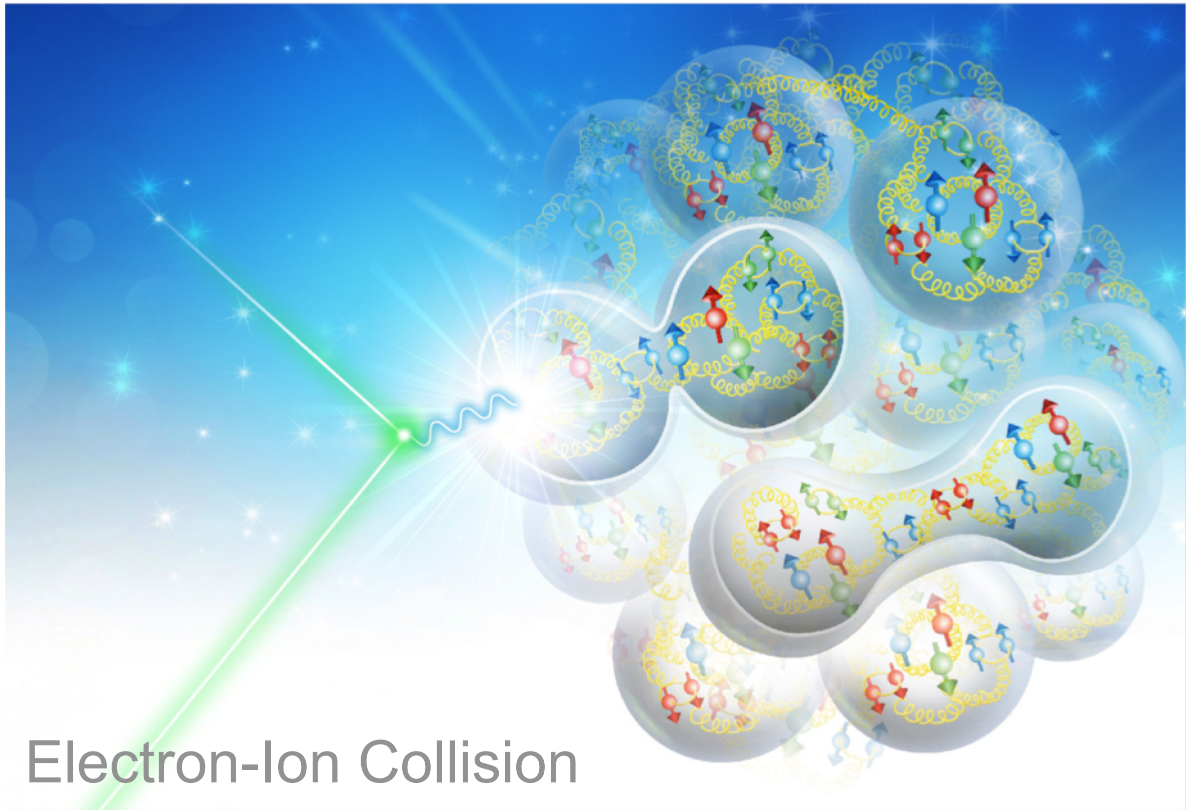
EICUG SWG – Introduction



Electron-Ion Collision

Andrea Bressan (INFN, University of Trieste)
Markus Diefenthaler (Jefferson Lab)
Torre Wenaus (Brookhaven Lab)

BROOKHAVEN NATIONAL LABORATORY

Jefferson Lab

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Role of Software Working Group

Develop

Support

Workflow environment for EICUG

- **to use** (tools, documentation, support) **and**
- **to grow with user input** (direction, documentation, tools)

Involvement from EICUG

e.g. benchmark processes, detector designs, reconstructions algorithms

Role of Software Working Group in the Yellow Reports organization

Physics Working Group

Conveners + **ex-offico**

Detector Working Group

Conveners + **ex-offico**

Simulation team: Software Working Group

Software Sub-group

Introduction

Getting started

Point of entry

HOME | JOIN EICUG | SCIENCE | ORGANIZATION | PHONEBOOK | CALENDAR | SOFTWARE | DOCUMENTS | MEDIA | LOGIN

Home » EIC Software

EIC Software

Software Working Group

The EICUG has formed a Software Working Group that collaborates with EIC Software initiatives and other experts in NP and HEP on detector and physics simulations for the EIC. The short-term goal of the working group is to meet in FY20 the requirements for common tools and documentation in the EICUG. The current work focusses on a common Geant4 infrastructure for the EIC that allows geometry exchange between the eRHIC and JLEIC concepts.

JupyterLab

The Software Working Group has adapted JupyterLab as a collaborative workspace to further develop EIC Science, to examine detector requirements, and to work on detector designs and concepts. JupyterLab is a web-based interactive analysis environment to create and share documents that contain the analysis code, the narrative of the analysis including graphics and equations, and visualizations of the analysis results. This will allow the EICUG not only to pursue simulations in a manner that is accessible, consistent, and reproducible to the EICUG as a whole, but also to build a collection of analyses and analysis tools in the fully extensible and modular JupyterLab environment. A quick start tutorial for fast simulations is available on the website for EIC Software.

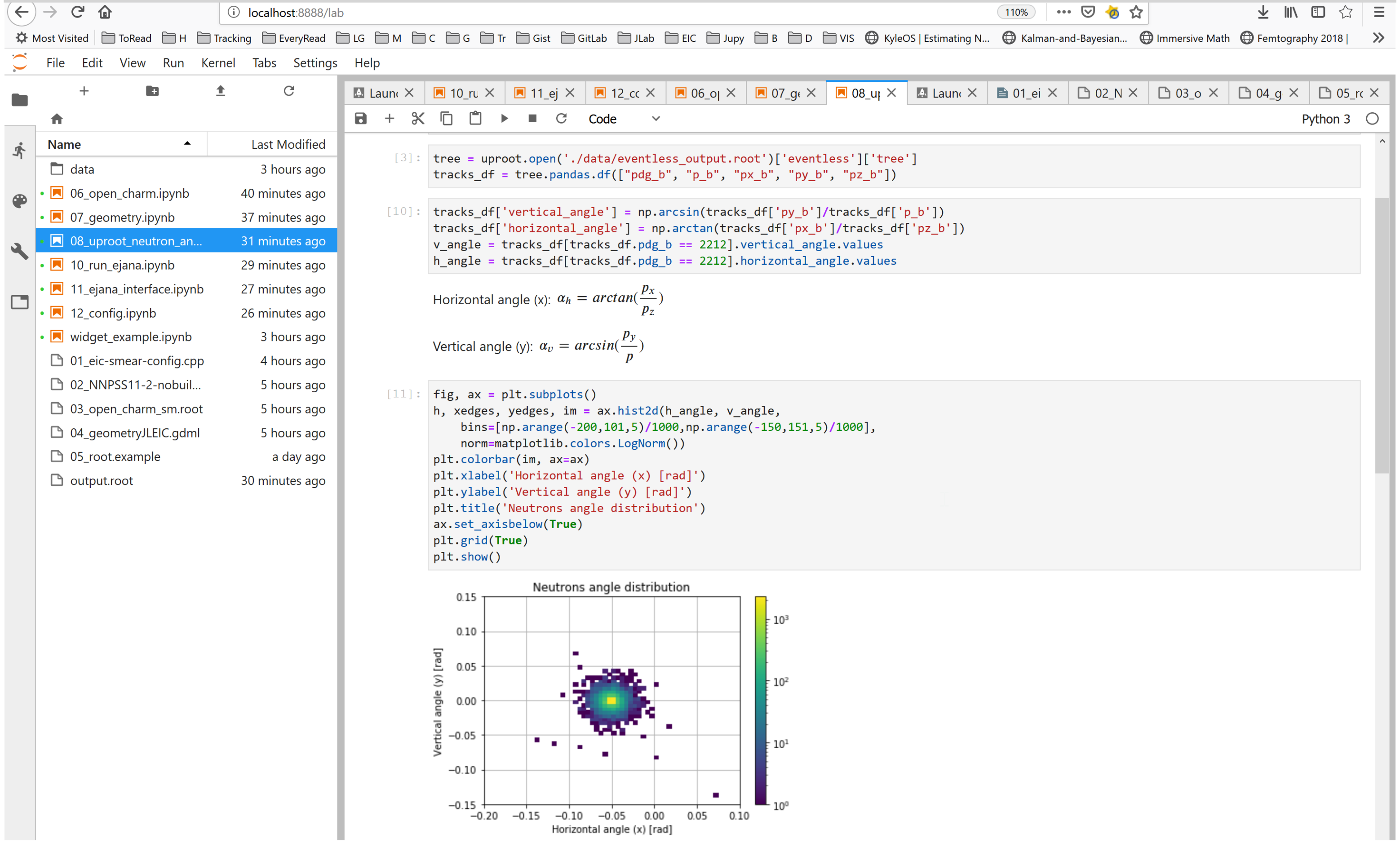
Important links

Mailing list	eicug-software@eicug.org (subscribe via Google Group)
Repository	http://gitlab.com/eic
Website	https://software.eicug.org

Collaborative workspace for EICUG

JupyterLab

- web-based interactive analysis environment



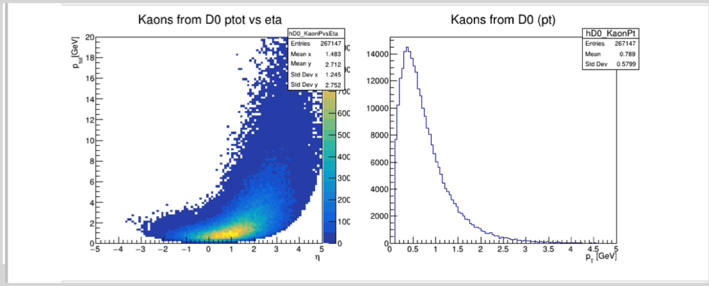
Jupyter Notebooks

- writing analysis code

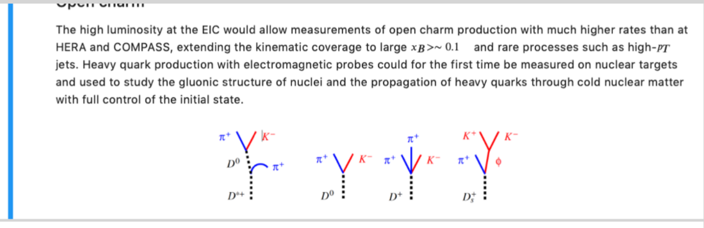
Python

Root/C++

- visualization of results

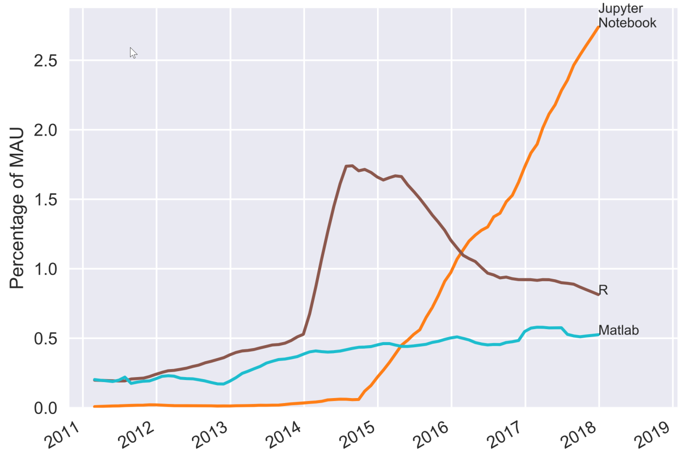


- narrative of the analysis



JupyterLab environment

- **bridge to modern data science**, e.g.,



  - *Nature* **563**, 145-146 (2018): “Why Jupyter is data scientists’ computational notebook of choice”
  - more than three million Jupyter Notebooks publicly available on GitHub

- **collaborative workspace** to create and share Jupyter Notebooks

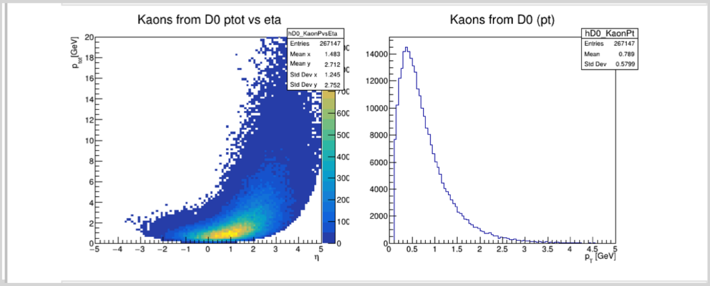
- **web-based interactive analysis environment** accessible, consistent, reproducible analyses

- **fully extensible and modular** build a collection of analyses and analysis tools

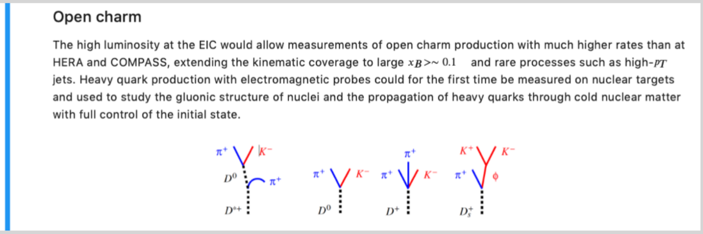
Jupyter Notebooks

- **writing analysis code**

- **visualization of results**



- **narrative of the analysis**



Modular design

Escaping complexity scaling trap

- provide interfaces to internal layers
- interaction between layers must be clear

Modularity each layer must be replaceable

simple	JupyterLab web interface
moderate	analysis scripts, python
complex	eJANA, plugins, C++
expert	JANA, eic-smear, *fun4all*, ROOT, Geant4

../data/beagle_eD.txt

```
[3]: jana.run()
```

Total events processed: 10001 (~ 10.0 kevt)

▸ **Full log**

▾ **Run command**

```
ejana
-Pplugins=beagle_reader,vmeson,event_writer
-Pnthreads=1
-Pnevents=10000
-Poutput=beagle.root
../data/beagle_eD.txt
-Pjana:debug_plugin_loading=1
```

EIC Software

Simulation of physics processes
Physics Working Group

Monte Carlo Event Generators
Tutorials in preparation

Simulation of detector responses
Detector Working Group

Fast simulations
Tutorials ✓

Full simulations
Tutorials **today**

Physics analysis
Physics and Detector Working Groups

Reconstruction of physics processes
Tutorials in preparation

Remote tutorials

Jan. 9 — **Introduction** fast simulations, JupyterLab analysis **recording**
9:00 a.m. and 6:00 p.m. (EDT) with **79 participants**

Today — **Detector Full Simulations** implement and integrate subdetector in existing detector concepts, modify detector concept **here at BNL**

Feb. 6 — **Detector Full Simulations** repetition of **Today**

Continuing tutorials according to survey and other requests

Feb. — **MCEG, reconstruction** stay tuned

January 2020

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	**9**	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	**29**	**30***	**31***	

* EIC Generic Detector R&D

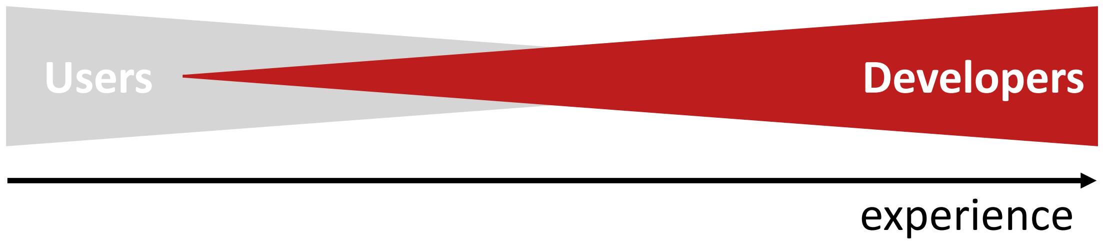
February 2020

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	**6**	7	8

Support

support team

being built up
weekly shifts



software-support@eicug.org

Mailing list (anyone can contact)
Google forum (for archive of support requests and start of knowledge base)

http://eicug.slack.com/

EICUG Slack workspace with software-support channel

Section

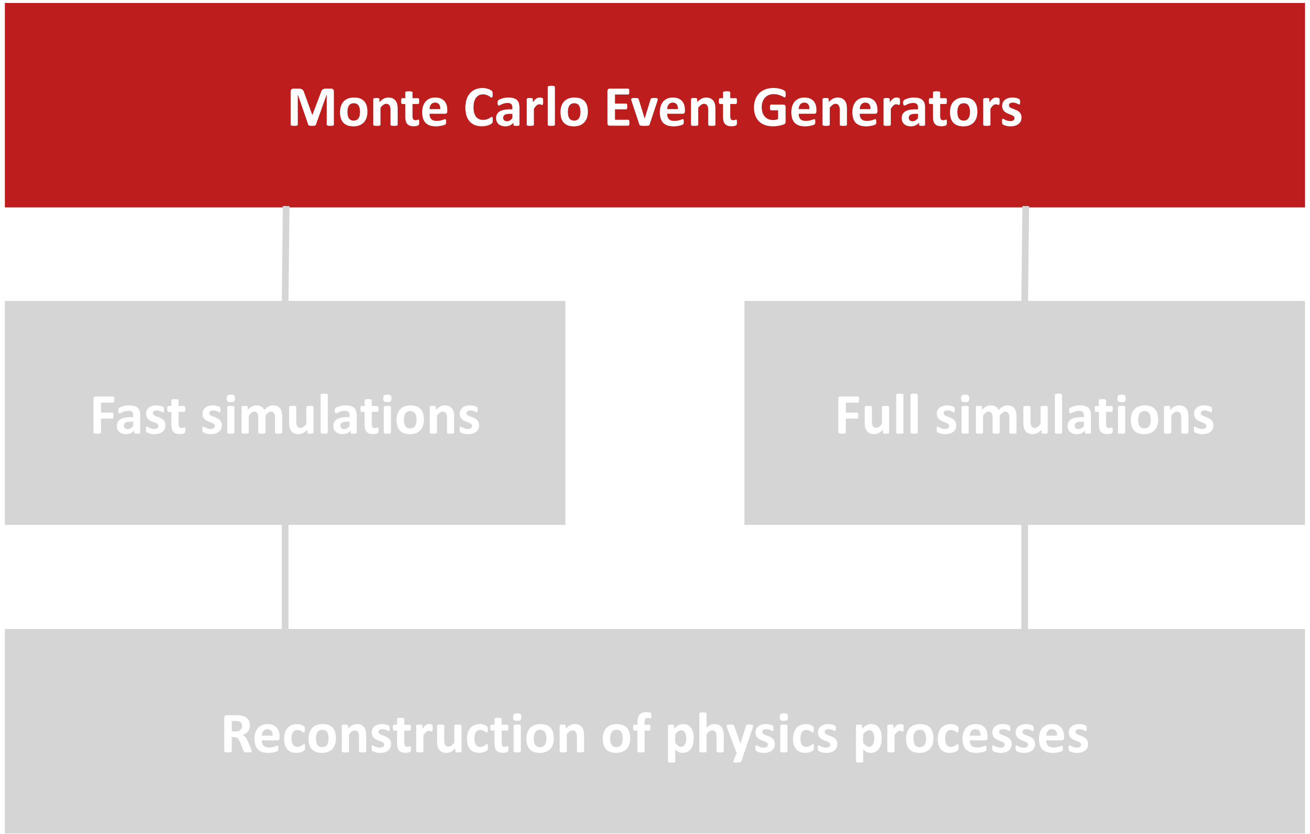
Status of EICUG simulations

Simulations of physics processes and detector responses

Simulation of physics processes

Simulation of detector responses

Physics analysis



Broad collection of event generators used for EIC

Monte Carlo Event Generators (MCEG)

The following event generators are available:

- ep
  - DJANGOH: (un)polarised DIS generator with QED and QCD radiative effects for NC and CC events.
  - gmc_trans: A generator for semi-inclusive DIS with transverse-spin- and transverse-momentum-dependent distributions.
  - LEPTO: A leptoproduction generator - used as a basis for PEPSI and DJANGOH
  - LEPTO-PHI: A version of LEPTO with "Cahn effect" (azimuthal asymmetry) implemented
  - MILOU: A generator for deeply virtual Compton scattering (DVCS), the Bethe-Heitler process and their interference.
  - PYTHIA: A general-purpose high energy physics event generator.
  - PEPSI: A generator for polarised leptoproduction.
  - RAPGAP: A generator for deeply inelastic scattering (DIS) and diffractive $e + p$ events.
- eA
  - BeAGLE: Benchmark eA Generator for LEptoproduction - UNDER CONSTRUCTION - a generator to simulate ep/eA DIS events including nuclear shadowing effects (based on DPMJetHybrid)
  - DPMJet: a generator for very low Q2/real photon physics in eA
  - DPMJetHybrid: a generator to simulate ep/eA DIS events by employing PYTHIA in DPMJet
  - Sartre is an event generator for exclusive diffractive vector meson production and DVCS in ep and eA collisions based on the dipole model.

From https://wiki.bnl.gov/eic/index.php/Simulations and available in https://gitlab.com/eic/mceg

MCEG R&D for EIC

Unique MCEG requirements for EIC Science

- MCEG for polarized ep, ed, and eHe^3
  - including novel QCD phenomena: GPDs, TMDs
- MCEG for eA

MCEG community

- focus of last two decades: **LHC**
  - **lesson learned** high-precision QCD measurements require high-precision MCEGs
  - MCEG not about tuning but about physics
- ready to work on ep/eA

Online catalogue for MCEGs (in preparation)

- **Categories** ep, eA, radiative effects
- Name
- Contact information
- **Brief Description** What processes are described? What is unique about the MCEG? Include version number as reference.
- **References (links)** website, repository, documentation, container, validation plots

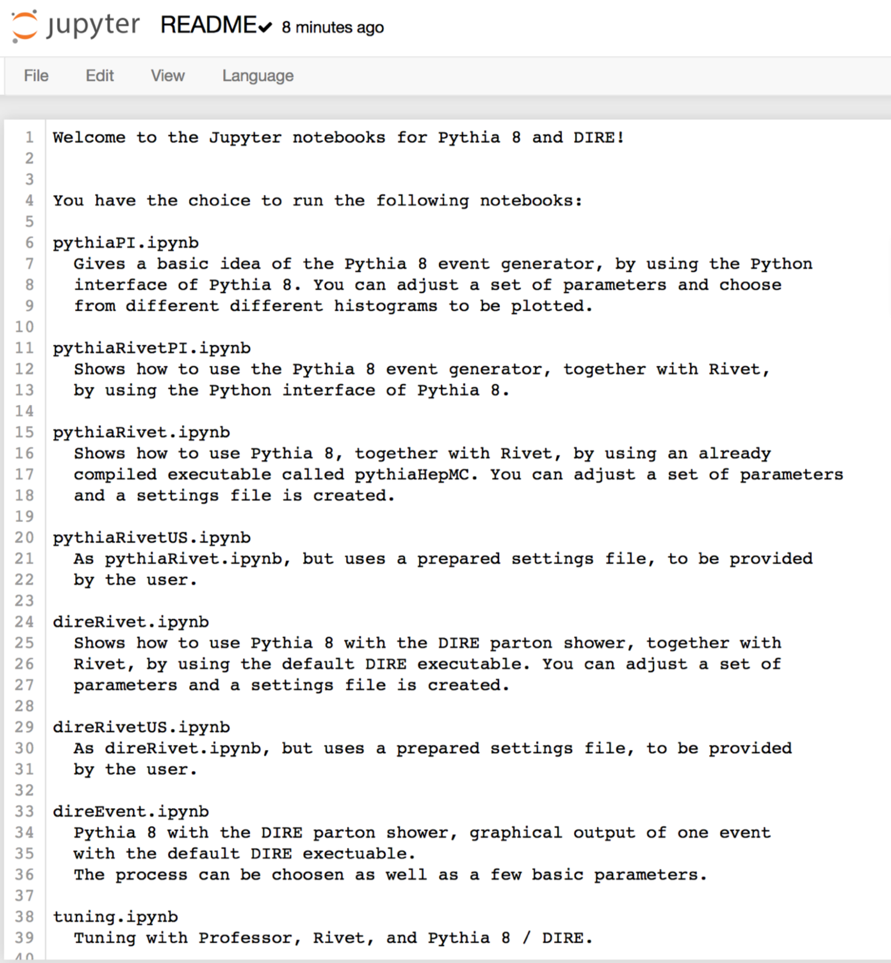
Example

- **Category** ep, eA, exclusive vector meson production, general photoproduction
- **Name** eSTARlight
- **Contact Information** Spencer Klein, srklein@lbl.gov
- **Brief description** eSTARlight simulates coherent photoproduction and electroproduction of vector mesons in ep and eA collisions. It can simulate a variety of different vector mesons, and it also includes an interface to DPMJET, which allows for general simulation of photonuclear interactions. It internally simulates most simple (2-body) vector meson decays with a correct accounting for the initial photon polarization (transverse for Q^2 ~ 0, with an increasing longitudinal component with increasing Q^2) in the angular distributions of the final state. It can also interface to PYTHIA8 to simulate more complicated decays.
- **References** The code is freely available from https://estarlight.hepforge.org/ The Readme file includes a fairly comprehensive users manual. The physics behind the code is documented in M. Lomnitz and S. Klein, Phys. Rev. C99, 015203 (2019).

Example MC simulations will be available on **HepSim** for benchmarks and validation (more and more examples added).

JupyterLab integration of MCEG (ongoing)

Example: Container for Pythia8+DIRE

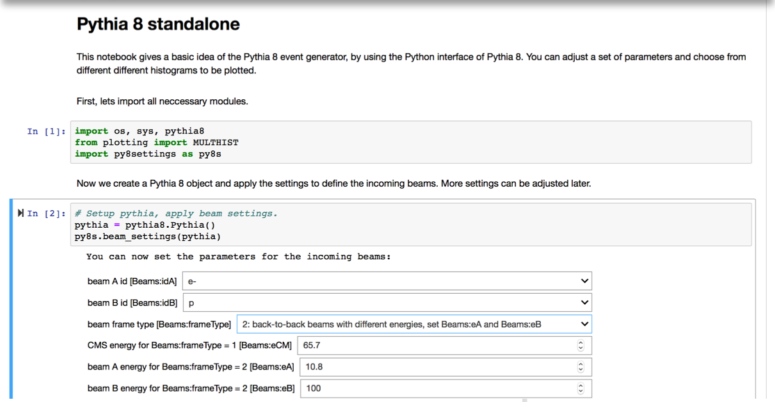


jupyter README✔ 8 minutes ago Logout

File Edit View Language Plain Text

```
Welcome to the Jupyter notebooks for Pythia 8 and DIRE!


You have the choice to run the following notebooks:

pythiaPI.ipynb
  Gives a basic idea of the Pythia 8 event generator, by using the Python
  interface of Pythia 8. You can adjust a set of parameters and choose
  from different different histograms to be plotted.

pythiaRivetPI.ipynb
  Shows how to use the Pythia 8 event generator, together with Rivet,
  by using the Python interface of Pythia 8.

pythiaRivet.ipynb
  Shows how to use Pythia 8, together with Rivet, by using an already
  compiled executable called pythiaHepMC. You can adjust a set of parameters
  and a settings file is created.

pythiaRivetUS.ipynb
  As pythiaRivet.ipynb, but uses a prepared settings file, to be provided
  by the user.

direRivet.ipynb
  Shows how to use Pythia 8 with the DIRE parton shower, together with
  Rivet, by using the default DIRE executable. You can adjust a set of
  parameters and a settings file is created.

direRivetUS.ipynb
  As direRivet.ipynb, but uses a prepared settings file, to be provided
  by the user.

direEvent.ipynb
  Pythia 8 with the DIRE parton shower, graphical output of one event
  with the default DIRE exectuable.
  The process can be choosen as well as a few basic parameters.

tuning.ipynb
  Tuning with Professor, Rivet, and Pythia 8 / DIRE.
```

Jupyter notebook interface



Pythia 8 standalone

This notebook gives a basic idea of the Pythia 8 event generator, by using the Python interface of Pythia 8. You can adjust a set of parameters and choose from different different histograms to be plotted.

First, lets import all neccessary modules.

```
In [1]: import os, sys, pythia8
        from plotting import MULTIHIST
        import py8settings as py8s
```

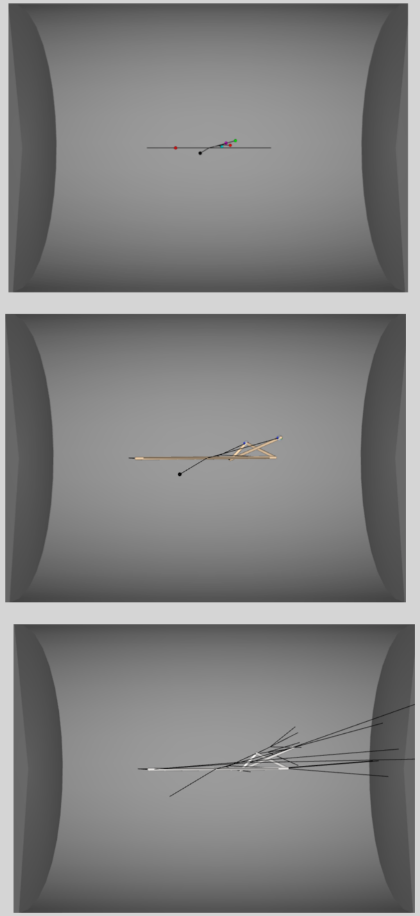
Now we create a Pythia 8 object and apply the settings to define the incoming beams. More settings can be adjusted later.

```
In [2]: # Setup pythia, apply beam settings.
        pythia = pythia8.Pythia()
        py8s.beam_settings(pythia)
```

You can now set the parameters for the incoming beams:

beam A id [Beams:idA]	e-
beam B id [Beams:idB]	p
beam frame type [Beams:frameType]	2: back-to-back beams with different energies, set Beams:eA and Beams:eB
CMS energy for Beams:frameType = 1 [Beams:eCM]	65.7
beam A energy for Beams:frameType = 2 [Beams:eA]	10.8
beam B energy for Beams:frameType = 2 [Beams:eB]	100

Visualization of ep collision

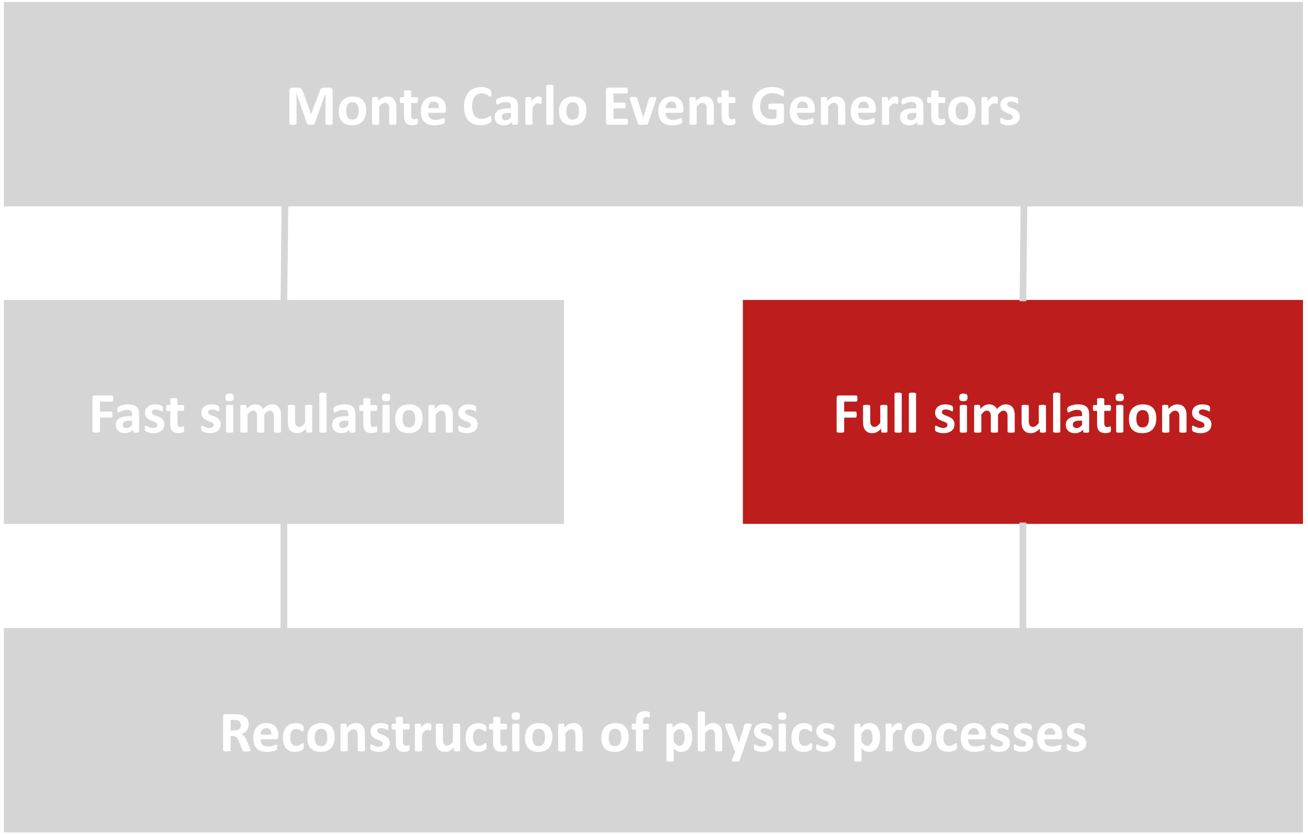


Simulations of physics processes and detector responses

Simulation of physics processes

Simulation of detector responses

Physics analysis



Detector Simulation

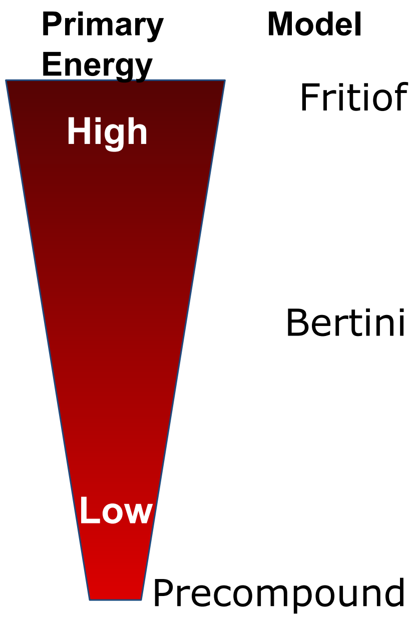
- **collaboration with Geant4 International Collaboration**
    - **liaison** Makoto Asai (SLAC)
- **Geant4 for EIC**
    - coordinate input for Geant4 validation based on EIC physics list maintained by (former) SLAC Geant4 group
    - Geant4 10.6 recommended (released Dec. 6)

09/24 Geant4 Technical Forum on EIC

- EIC detector and physics simulations rely on Geant4
- knowledge transfer (e.g., sub-event parallelism or tessellated solids)
- maintain EIC physics lists
- **request** improved photo-nuclear and electro-nuclear reactions

EIC

- energy range is different from LHC
- validation, tuning and extension including test beam studies



Geant4 infrastructure for EICUG

Requirements

- **EIC Generic Detector R&D program (T. Ullrich)** *"a simple lite setup with a well defined geometry description standard that is easy to use"*
- **EICUG** Flexible accelerator and detector interface with full support of existing IR designs and detector concepts

Approach

- common repository for detector R&D for EIC
- common detector description in Geant4 (C++) and not yet DD4hep (sub-detectors developed in Geant4 (C++))
- common detector naming convention for EIC
- possible common hits output structure
- concise document and template on how to implement and integrate subdetector in EIC detector concepts

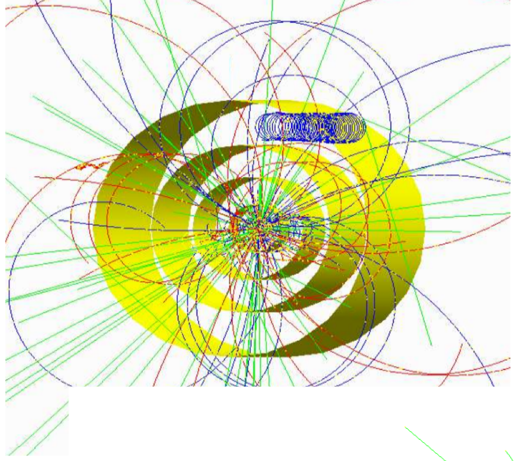
Discussion

- **two in-person meetings**
  - 07/10 EIC Software Meeting at BNL (minutes)
  - 09/24 EIC Software Meeting at JLAB (minutes)
- **evaluation** 09/30, 10/21, 10/28, 11/18, 11/25

Two solutions proposed

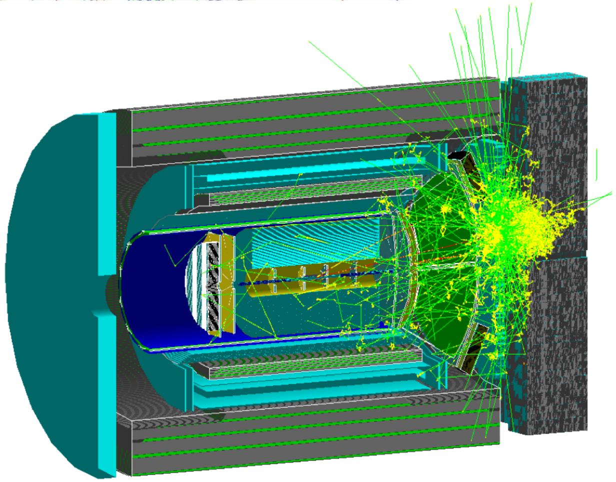
1. detector simulations in **fun4all**, major update for common EIC simulations
2. Geant4 application **g4e**, integrated in JupyterLab

Fun4All + GEANT4

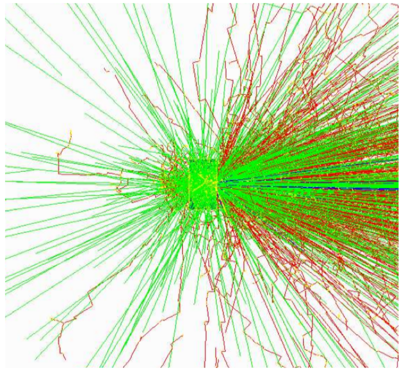


Pythia8 in a six layer silicon detector mockup and 2T field

- Mature Framework based on ROOT, steering with ROOT macros
- Modular – each detector is its own entity
- No central code needs to be modified when adding new detectors
- Detectors are combined using ROOT macros
- Distribution as singularity container + libraries in cvmfs*
- Daily builds + Continuous Integration
- No geometry model enforced
- Interface to eic-smear: most EIC specific Event generators accessible
- Pre-canned configurations for EIC-sPHENIX and partial JLEIC
- Used to provide input for our EIC detector LOI**
- Generic Volumes (box, cylinder, cone) can be implemented no macro level



Sarte as seen by an EIC detector



10 GeV Au on water phantom (NASA Space Radiation Lab)

*Installation: https://github.com/EIC-Detector/Singularity

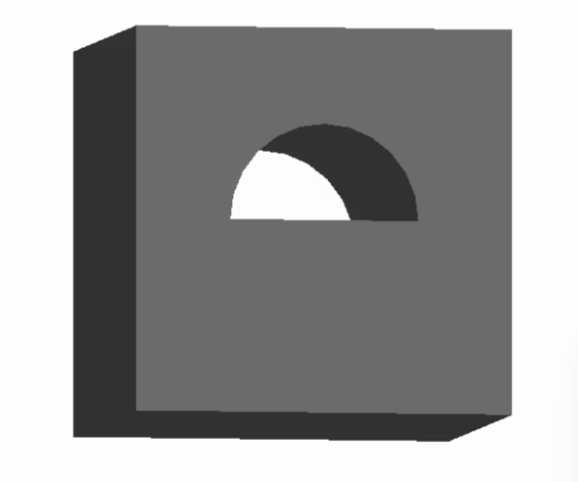
**https://arxiv.org/pdf/1402.1209.pdf
https://indico.bnl.gov/event/5283/attachments/20546/27556/eic-sphenix-dds-final-2018-10-30.pdf

For details: see selected Fun4All presentations https://www.phenix.bnl.gov/WWW/publish/pinkenbu/EIC/

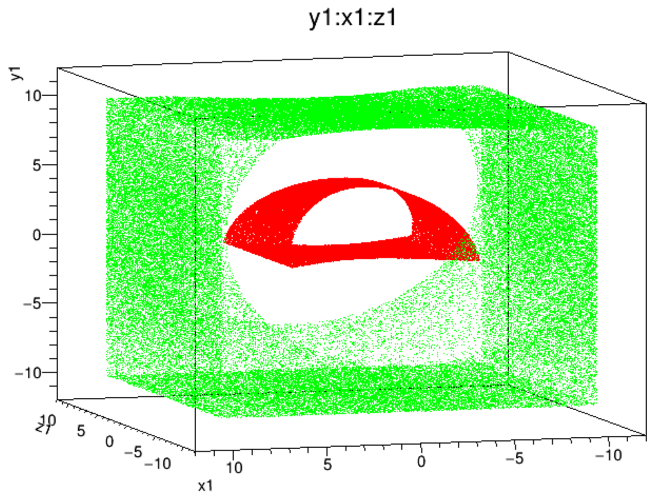
Implementing a Detector in Fun4All

Simplest Example, more sophisticated to come:
https://github.com/EIC-Detector/g4exampledetector:
simple/source: Simplest case - everything hardcoded, only active volumes
simple/macro: Fun4All_G4_Example01.C to run the show (and save Hits in ntuple)

Let’s call your detector PDirc*, 3 classes need to be implemented :
G4PDircSubsystem → interface between Fun4All and Detector
G4PDircDetector → GEANT4 Construct method
G4PDircSteppingAction → select which quantities to store for each hit
*Detector names can be set on the command line but you do not want identically named sources



Example01: block with
½ cylindrical hole



Geantino Scan to verify geometry using entry/exit coordinates of geantino tracks

You will find that help will always be given at Hogwarts to those who ask for it.

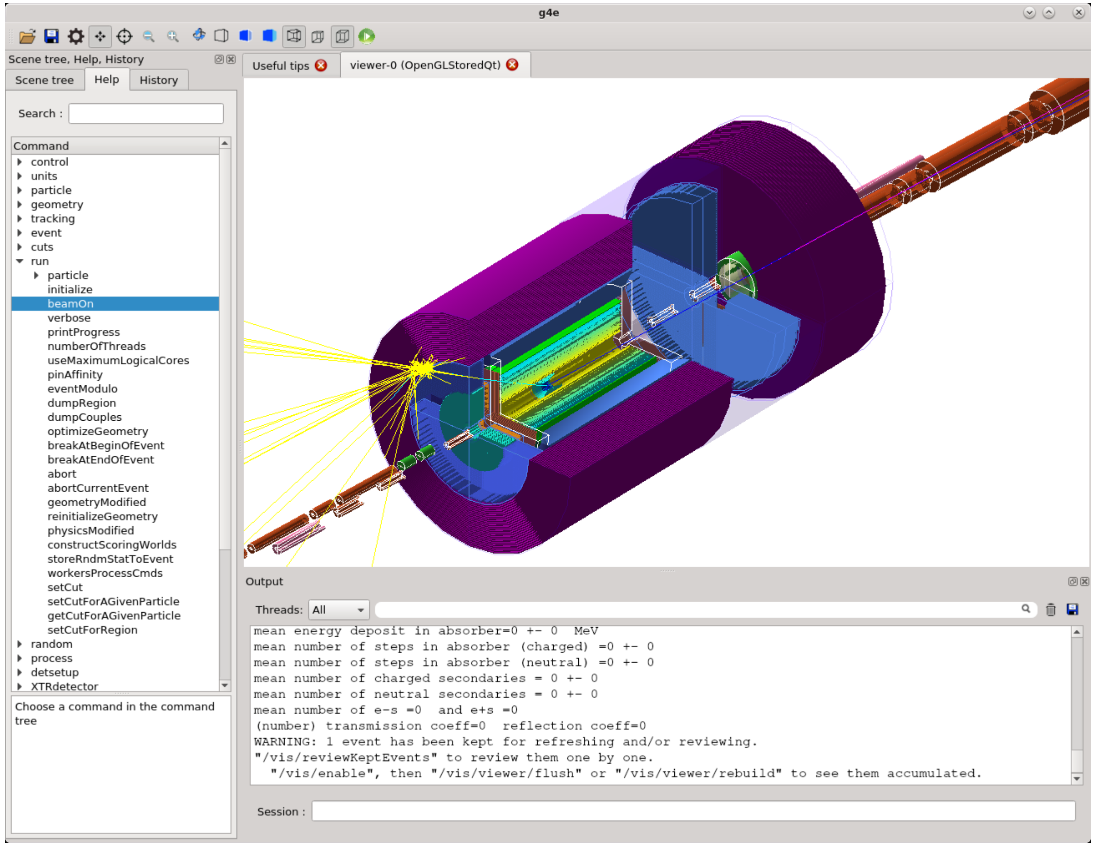
– Dumbledore

Tutorials:
https://github.com/EIC-Detector/tutorials

Join slack channel for support:
https://join.slack.com/t/eic-design-study/signup

Email:
Chris Pinkenburg pinkenburg@bnl.gov
Jin Huang jhuang@bnl.gov

G4E (Geant 4 EIC)



Standalone C++ Geant4 application

Various EIC MC file formats
Beagle, Pythia6, HEPMC - Pythia8, Herwig and others

Integration with accelerator elements

Infrastructure to import Geant4 detector geometry and simulation code

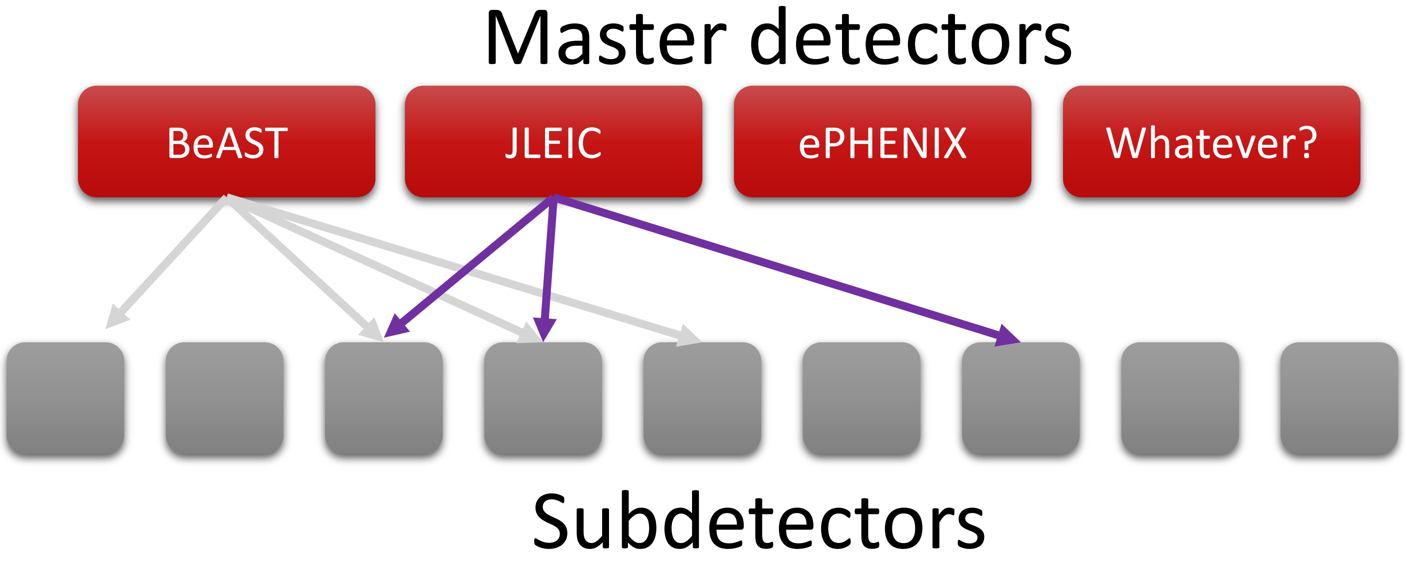
Subdetector integration in G4E

G4E goals

- Ability to have several “master” detectors (and IRs) such as BeAST, JLEIC etc.
- Ability to easily import Geant4 standalone detector implementations
- Ways to send the “right” configuration to a subdetector depending on selected “master” detector, so that one detector implementation could serve for different IRs
- **Stay as close to Geant4 as possible, to stay convenient for Geant4 experts**

To import a subdetector:

- *SubDetectorInterface* class implementation
- Subscribe to various standard Geant 4 actions *(SteppingAction, StackingAction, etc)*
- Define subdetector’s place in one or many “master” detectors



Accelerator interface

Accelerator design (beam elements)

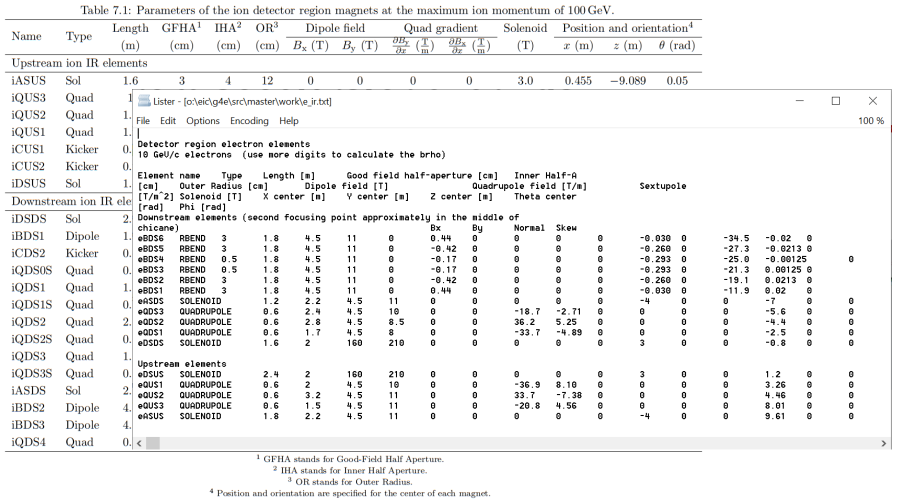


Table 7.1: Parameters of the ion detector region magnets at the maximum ion momentum of 100 GeV.

Name	Type	Length (m)	GFHA[1] (cm)	IHA[2] (cm)	OR[3] (cm)	Dipole field B_x (T)	Dipole field B_y (T)	Quad gradient $\frac{\partial B_y}{\partial x}$ ($\frac{T}{m}$)	Quad gradient $\frac{\partial B_x}{\partial x}$ ($\frac{T}{m}$)	Solenoid (T)	x (m)	z (m)	θ (rad)
Upstream ion IR elements													
iASUS	Sol	1.6	3	4	12	0	0	0	0	3.0	0.455	−9.089	0.05
iQUS3	Quad	1											
iQUS2	Quad	1.											
iQUS1	Quad	1.											
iCUS1	Kicker	0.											
iCUS2	Kicker	0.											
iDSUS	Sol	1.											
Downstream ion IR ele													
iDSDS	Sol	2.											
iBDS1	Dipole	1.											
iCDS2	Kicker	0.											
iQDS0S	Quad	0.											
iQDS1	Quad	1.											
iQDS1S	Quad	0.											
iQDS2	Quad	2.											
iQDS2S	Quad	0.											
iQDS3	Quad	1.											
iQDS3S	Quad	0.											
iASDS	Sol	2.											
iBDS2	Dipole	4.											
iBDS3	Dipole	4.											
iQDS4	Quad	0.											

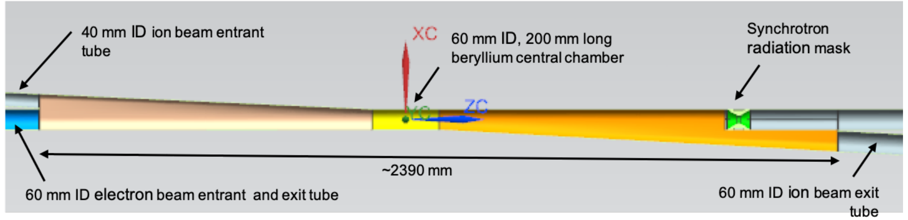
[1] GFHA stands for Good-Field Half Aperture.
[2] IHA stands for Inner Half Aperture.
[3] OR stands for Outer Radius.
[4] Position and orientation are specified for the center of each magnet.

Lister - [o:\eic\g4e\src\master\work\e_ir.txt]

File Edit Options Encoding Help 100 %

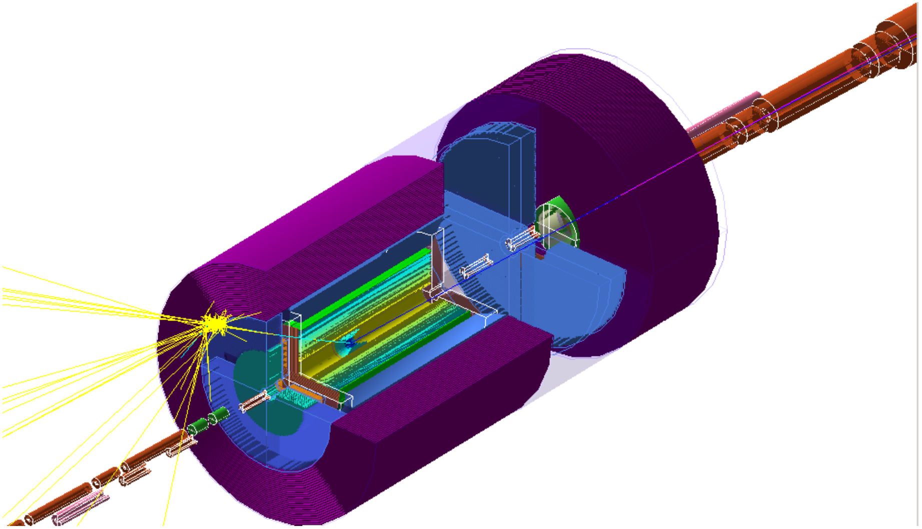
```
Detector region electron elements
10 GeV/c electrons  (use more digits to calculate the brho)

Element name    Type    Length [m]      Good field half-aperture [cm]   Inner Half-A
[cm]    Outer Radius [cm]       Dipole field [T]                Quadrupole field [T/m]          Sextupole
[T/m^2] Solenoid [T]    X center [m]    Y center [m]    Z center [m]    Theta center
[rad]   Phi [rad]
Downstream elements (second focusing point approximately in the middle of
chicane)                                        Bx      By      Normal  Skew
eBDS6   RBEND   3       1.8     4.5     11      0       0.44    0       0       0       -0.030  0       -34.5   -0.02   0
eBDS5   RBEND   3       1.8     4.5     11      0       -0.42   0       0       0       -0.260  0       -27.3   -0.0213 0
eBDS4   RBEND   0.5     1.8     4.5     11      0       -0.17   0       0       0       -0.293  0       -25.0   -0.00125        0
eBDS3   RBEND   0.5     1.8     4.5     11      0       -0.17   0       0       0       -0.293  0       -21.3   0.00125 0
eBDS2   RBEND   3       1.8     4.5     11      0       -0.42   0       0       0       -0.260  0       -19.1   0.0213  0
eBDS1   RBEND   3       1.8     4.5     11      0       0.44    0       0       0       -0.030  0       -11.9   0.02    0
eASDS   SOLENOID        1.2     2.2     4.5     11      0       0       0       0       0       -4      0       0       -7      0       0
eQDS3   QUADRUPOLE      0.6     2.4     4.5     10      0       0       -18.7   -2.71   0       0       0       0       -5.6    0       0
eQDS2   QUADRUPOLE      0.6     2.8     4.5     8.5     0       0       36.2    5.25    0       0       0       0       -4.4    0       0
eQDS1   QUADRUPOLE      0.6     1.7     4.5     8       0       0       -33.7   -4.89   0       0       0       0       -2.5    0       0
eDSDS   SOLENOID        1.6     2       160     210     0       0       0       0       0       3       0       0       -0.8    0       0

Upstream elements
eDSUS   SOLENOID        2.4     2       160     210     0       0       0       0       0       3       0       0       1.2     0       0
eQUS1   QUADRUPOLE      0.6     2       4.5     10      0       0       -36.9   8.10    0       0       0       0       3.26    0       0
eQUS2   QUADRUPOLE      0.6     3.2     4.5     11      0       0       33.7    -7.38   0       0       0       0       4.46    0       0
eQUS3   QUADRUPOLE      0.6     1.5     4.5     11      0       0       -20.8   4.56    0       0       0       0       8.01    0       0
eASUS   SOLENOID        1.8     2.2     4.5     11      0       0       0       0       0       -4      0       0       9.61    0       0
```

Engineering Design (CAD)



Detector Simulations (Geant4)

Tuning



Status

eRHIC and JLEIC information available
Common interface under active development

Why two options?

- At The Software Working Group was caught by the start of the "Yellow Report" effort with two ongoing developments for full simulations:

  - **fun4all**, originated from within (s)PHENIX, mature and centered around the use of ROOT macros
  - **g4e**, build up for the EIC (and therefore in a "younger" stage of development) constructed as a pure GEANT4 application (and integrated into JupyterLab environment)

- Each of the two is supported by a core team of developers.

- We put forward both options, leaving the "users" the freedom to choose base on their coding preferences.

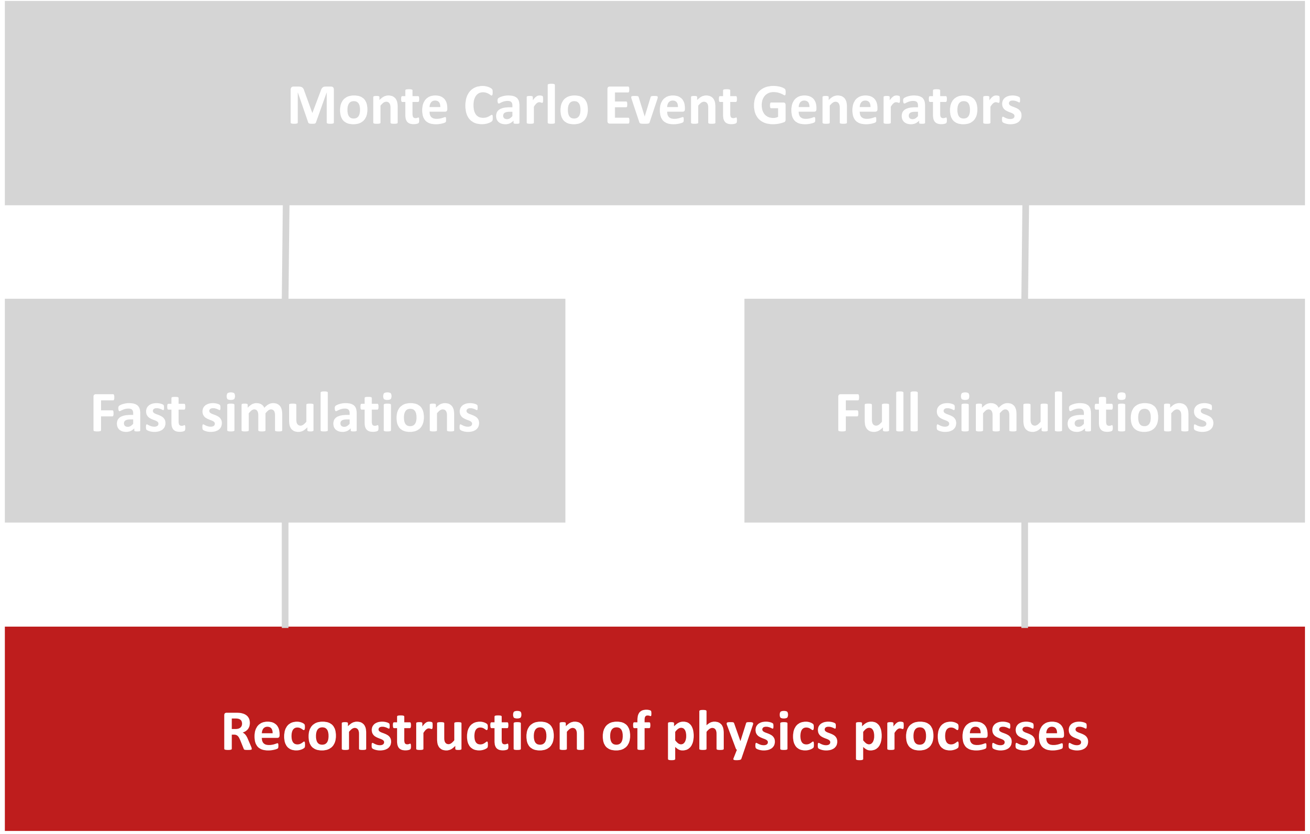
- We will take advantage of the two codes to cross-check few selected and critical results in order to improve our confidence in the outcome of the simulations.

Simulations of physics processes and detector responses

Simulation of physics processes

Simulation of detector responses

Physics analysis



Reconstruction options (in alphabetical order)

A Common Tracking Software (ACTS)

ATLAS software → generic, framework- and experiment-independent track reconstruction software

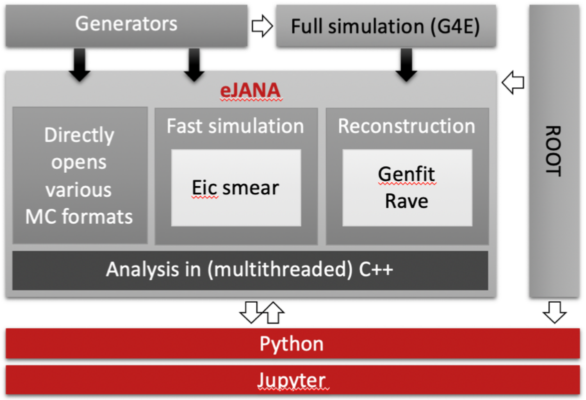
Collaboration of LBNL NP and HEP (Y. S. Lai et al.) for ACTS for EIC

EiCRoot (slide 29)

e$^{\text{JANA}}$

JANA2 (slide 30) **+ plugins**

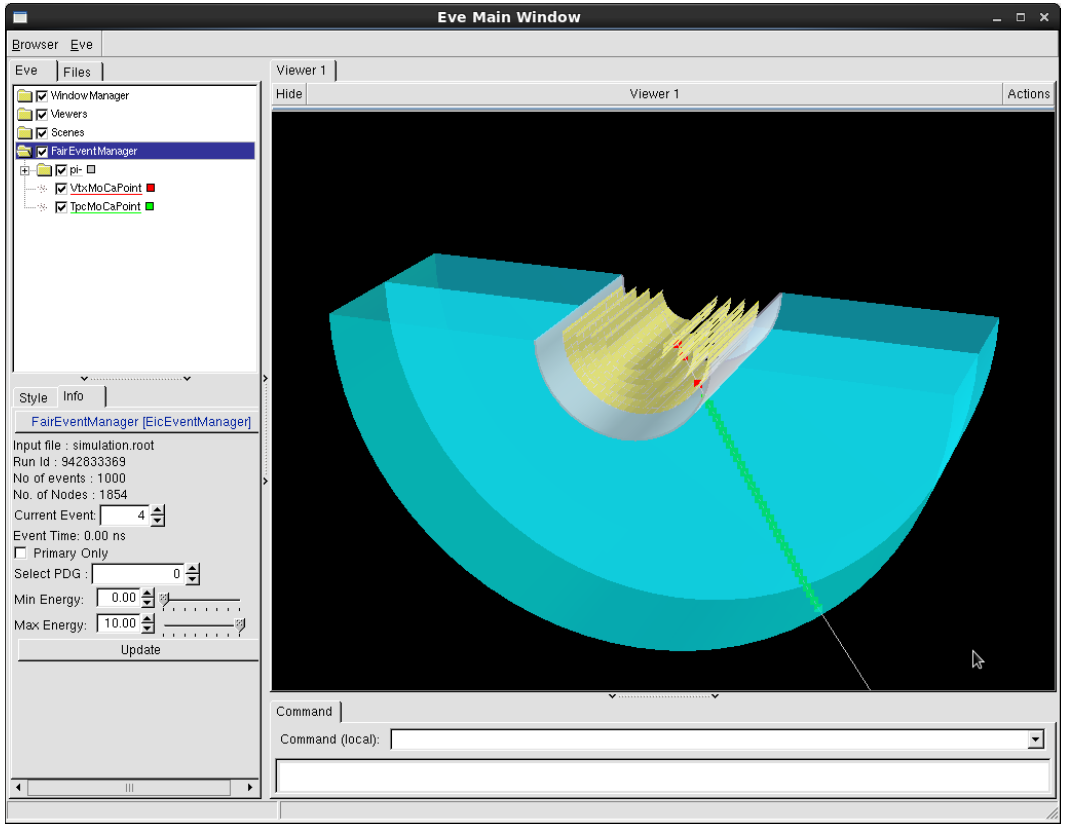
- EIC data reconstruction
- EIC data analysis

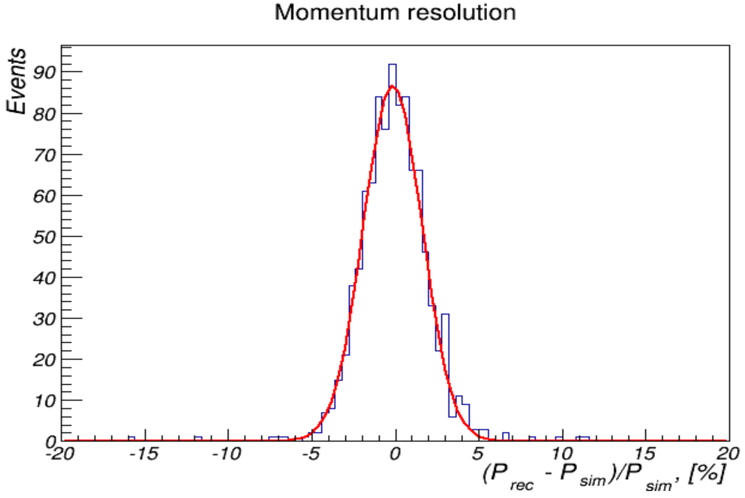


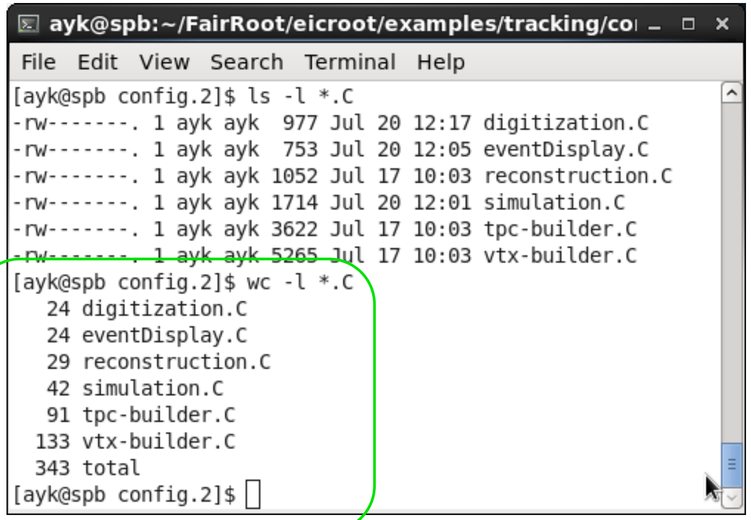
Fun4all (more in Chris's tutorial)

EicRoot: Example tracking study

Consider vertex tracker + TPC in 3T field; shoot 10 GeV/c pions at θ=75°



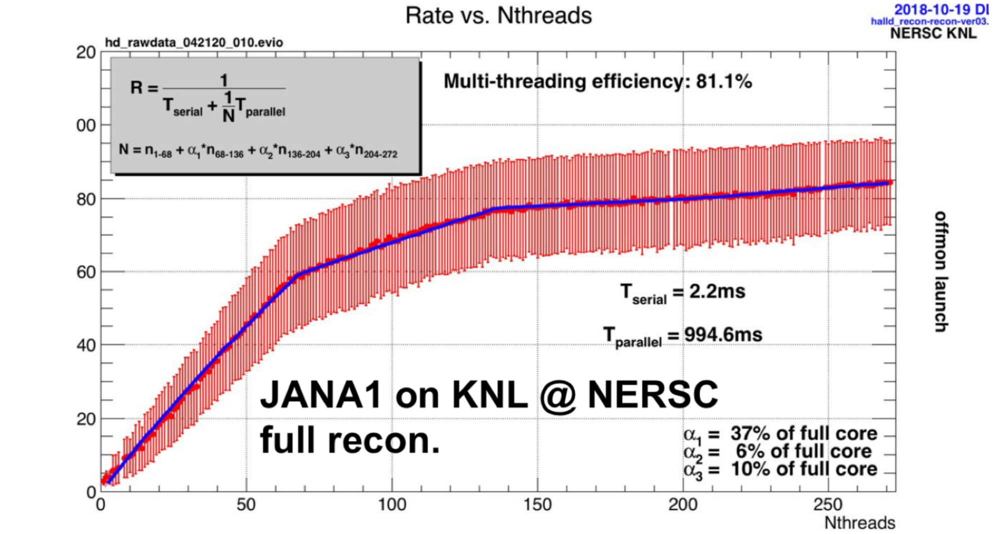


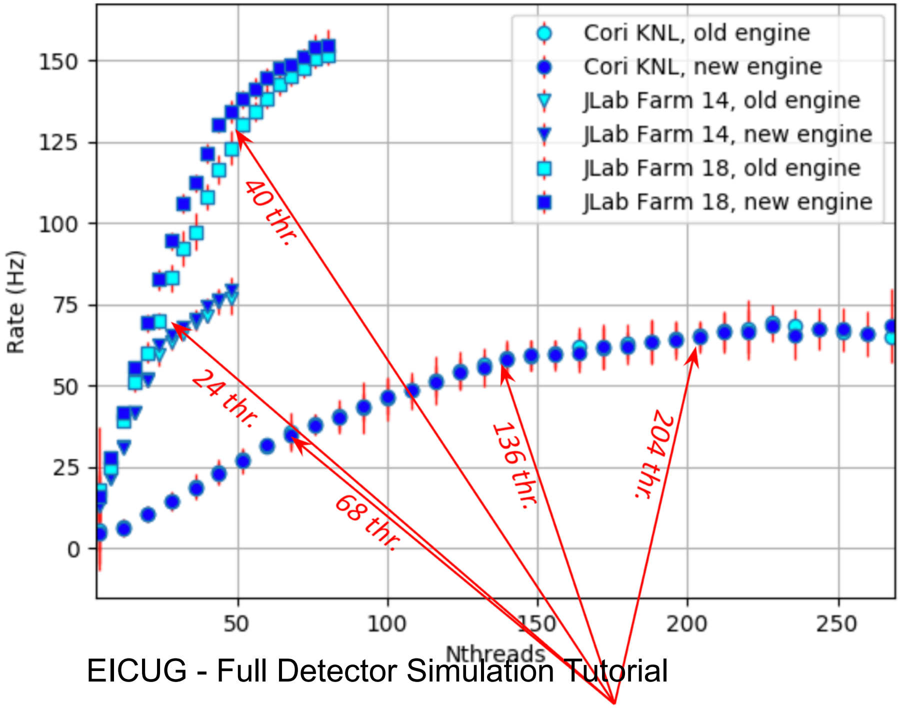


-> see examples/tracking/config.2 directory for details

- Once Docker image is downloaded it takes <5 minutes to generate this plot

JANA2





JANA C++ event processing framework

- **factory model** on demand interface, user-centered design
- multi-threaded with > 10 years experience
- **plugin support** provide mechanism for many physicists to contribute, multi-threading external to contributed code (parallelizer)

JANA2 active development (JLAB LDRD)

- take advantage of new C++ standards
- Python interface
- part of Streaming Readout Grand Challenge at Jefferson Lab (C++ streamed events processing framework)

Agenda's of today's EIC Software Meeting

Detector Full Simulation Tutorial

- implement and integrate subdetector in existing detector concepts
- modify detector concept

9:30 a.m. (EDT)	**Luminosity Monitor as example subdectector (Jaroslav)**
10:00 a.m. (EDT)	**fun4all (Chris)** **Quickstart** https://github.com/EIC-Detector/Singularity **Download** https://www.phenix.bnl.gov/WWW/publish/phnxbld/sPHENIX/Singularity/index.html
2:00 p.m. (EDT)	**g4e example (Dmitry)** **Quickstart** https://eic.gitlab.io/documents/quickstart/ **Preparations** docker pull electronioncollider/epic-gui

Greenfield Discussion

classify EICUG Simulation efforts and feedback into NP and HEP Software & Computing landscape