

ELECTRON ION COLLIDER USER GROUP
SOFTWARE NEWS

M. Diefenthaler on behalf of the
EICUG Software Working Group

EICUG Software Working Group

66 members

Convener

A. Bressan (Trieste)
M. Diefenthaler (JLAB)
T. Wenaus (BNL)

Mailing list eicug-software@eicug.org

subscribe via Google Group

Repository <https://github.com/eic>

Website <http://www.eicug.org/web/content/eic-software>

Core Group



J. Adam (BNL)



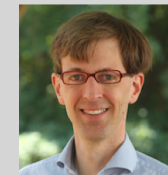
M. Asai (SLAC)



N. Brei (JLAB)



A. Bressan (Trieste)



W. Deconinck (Manitoba)



M. Diefenthaler (JLAB)



J. Furletova (JLAB)



S. Furletov (JLAB)



S. Joosten (ANL)



K. Kauder (BNL)



A. Kiselev (BNL)



J. Lauret (BNL)



D. Lawrence (JLAB)



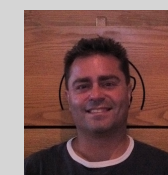
C. Pinkenburg (BNL)



M. Potekhin (BNL)



D. Romanov (JLAB)



M. Ungaro (JLAB)



T. Wenaus (BNL)

SWG activity since 1st EIC Yellow Report Workshop



Documentation

- GitHub and GitHub pages (s. 24)
- tutorials (s. 7)

Fast simulations

- eic_smear updates and validation (s. 9, 10)
- fiducial volumes
- **options** Delphes, **greenfield** fast simulations in Geant4

Integration

- accelerator interface
- services

MC HepMC3 (supports HEPEVT / LUND formats)

OSG (with Computing groups at BNL and JLAB, s. 21-22)

Storage (with Computing groups at BNL and JLAB, (s. 22)

Versioning

- integration with DetectorMatrix (s. 41)
- software deployment with smaller downloads and native HTC / HPC support (cvmfs, Spack, s. 21)

Role of **Software Working Group** in Yellow Reports organization

Physics Working Group

Conveners

Detector Working Group

Conveners + **ex-officio**

Simulation team Software Working Group

(new) Coordination Biweekly meetings of DWG+PWG+SWG conveners

Role of Software Working Group in Yellow Report initiative

Develop

Support

Workflow environment for EIC simulations


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



Involvement from EICUG

2nd EIC Yellow Report Workshop
Discussion on
PWG / DWG / SWG workflow

Online tutorials




EIC User Group
3 subscribers

<https://www.youtube.com/channel/UCXc9WfDKdILXoZMGrotkf7w>

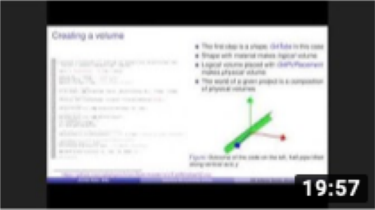
SUBSCRIBE

HOMEVIDEOSPLAYLISTSCHANNELSDISCUSSIONABOUT

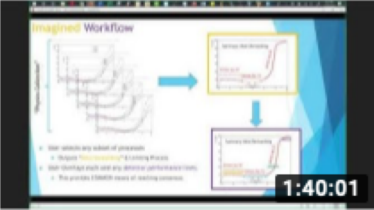
Uploads ▶ PLAY ALL




EIC Software Group: An Introduction (01/09/2020)
5 views • 1 week ago



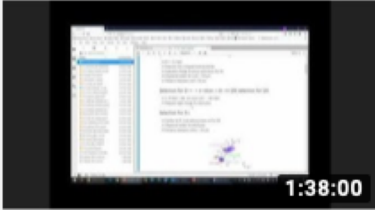
EIC Software Tutorial: Example Detector...
3 views • 1 week ago




EICUG Software Working Group: Greenfield
31 views • 1 month ago



EIC Software Tutorial: Detector Full Simulations in...
35 views • 1 month ago



EIC Software Tutorial: Detector Full Simulations in...
32 views • 1 month ago

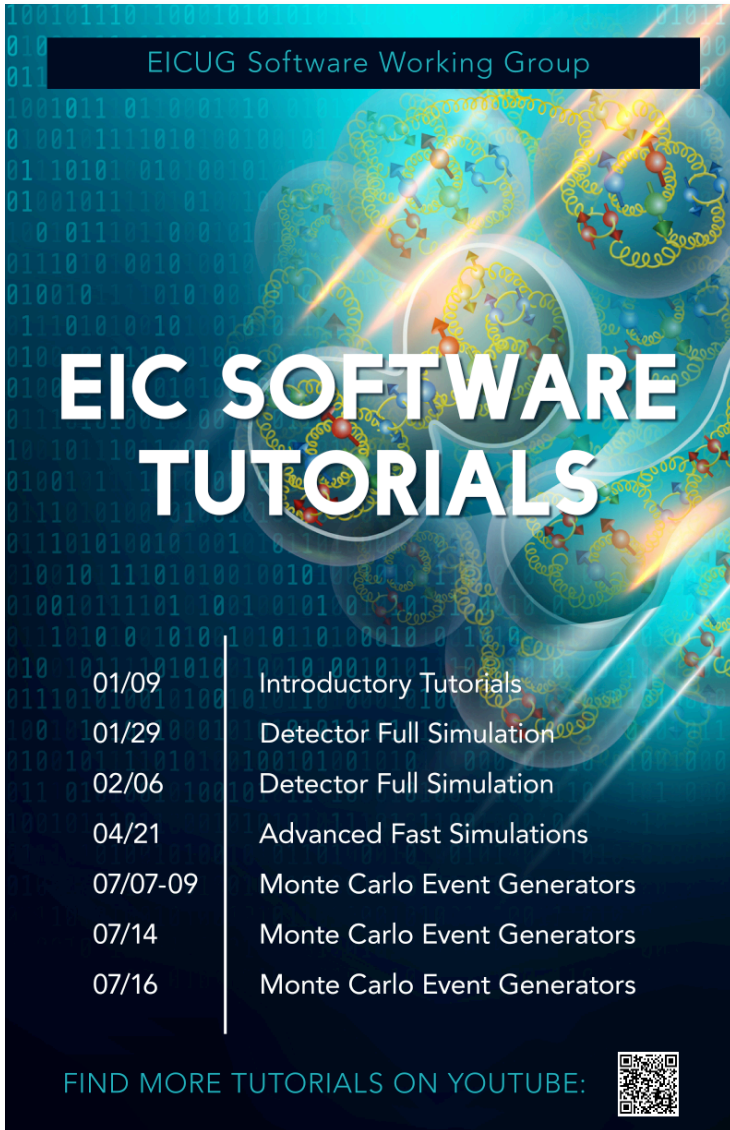


EIC Software Tutorial: Fast Simulations (01/09/2020)
30 views • 1 month ago

Recordings from tutorials

- **Fast Simulation Tutorial** Introduction to JupyterLab workspace, using fast simulations as example
- **Detector Full Simulation Tutorials** Geant4 for EIC, how to modify existing detector concepts, and how to integrate a new detector into one of the existing detector concepts.
- **(new) Jim Pivarski Tutorial: uproot and Awkward Array** process and analyze Root files with pure Python libraries
- **(new) Advanced Fast Simulation Tutorial** Fast simulations on the command line and in JupyterLab, singularity


Upcoming meetings and tutorials

A poster for EIC Software Tutorials. The background is dark blue with a grid of binary code (0s and 1s) and several glowing, translucent spheres containing molecular or particle-like structures. The text 'EIC SOFTWARE TUTORIALS' is prominently displayed in the center in large, white, bold letters. At the top, a black banner reads 'EICUG Software Working Group'. On the left side, a vertical list of dates and topics is provided. At the bottom, there is a QR code and the text 'FIND MORE TUTORIALS ON YOUTUBE:'.

EICUG Software Working Group

EIC SOFTWARE TUTORIALS

01/09	Introductory Tutorials
01/29	Detector Full Simulation
02/06	Detector Full Simulation
04/21	Advanced Fast Simulations
07/07-09	Monte Carlo Event Generators
07/14	Monte Carlo Event Generators
07/16	Monte Carlo Event Generators

FIND MORE TUTORIALS ON YOUTUBE: 

May 27 RICH simulations

Topic description of the RICH detectors in Geant4

By M. Asai (Geant4), S. Easo (CERN)

June 10 ACTS

Topic track and vertexing reconstruction software

By X. Ai (Berkeley)

July 7-9, 14, 16 Monte Carlo Event Generators

Topic Pythia8+DIRE, Herwig 7, Sherpa 2, Rivet, discussion

By C. Bierlich (Copenhagen, LUND), S. Hoeche (FNAL), S. Plätzer (Vienna), and S. Prestel (LUND)

Update on fast simulations

Simulation of physics processes

Monte Carlo Event Generators

Simulation of detector responses

Fast simulations

Full simulations

Physics analysis

Reconstruction of physics processes

Most important use case:

“I want to smear with Detector A v 1.0 and Detector B v2.3”

- Requires decoupling of detectors from the eic-smear repo ← **Done**

Versioning Status

- **Users can now easily download and use existing parameterizations**
- Overview table can be linked to, and link back to, detector matrix, wiki, any other resource

Coming soon (~this week)

- Convenience tools for running
- Compiled detector library support (has to interact nicely with a global installation)

- Obtain latest detector collection from <https://github.com/eic/eicsmeardetectors>
- Peruse available detectors:

Official parameterizations

These are recommended for Yellow Report work. This collection will grow as the detector matrix gets upgraded and concrete designs are parameterized.

Name	min. version	Details and Comments
HandBook 1.2	1.0.4	Based on HANDBOOK v1.2 (Feb 20, 2020) with adaptations where information was incomplete or missing
PerfectDetector	1.0.4	Perfect detection and PID in $ \eta < 15$

- Check version

```
$ eic-smear -v # to be added
```

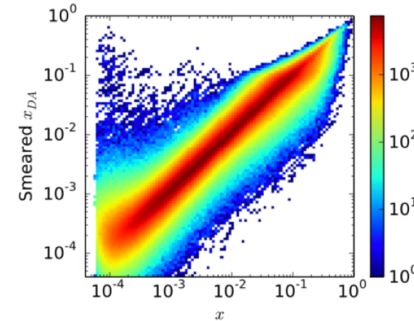
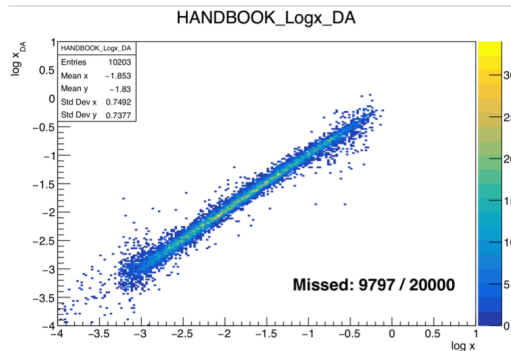
- Load and run! Supports tab completion

```
root [0] .L SmearHandBook_1_2.cxx
root [1] SmearTree(BuildHandBook_1_2(), "in.root", "out.root")
```


`tests/qaplots.cxx` serves a dual purpose:

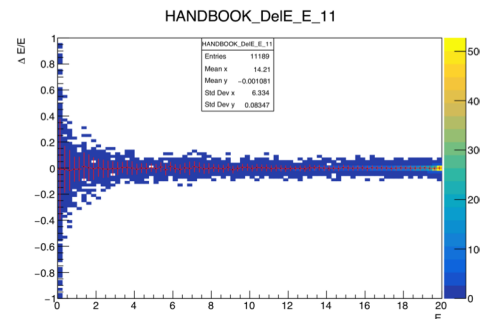
- Demonstrate and document use of smeared trees
- **Generate a suite of QA and performance plots**
- Currently:
 - x , y , Q^2 response using lepton method, Jacquet-Blondel, or Double Angle → compare to arXiv:1409.1633

```
$ ./tests/qaplots -h
usage: ./tests/qaplots
[-i txtfilename] (Lund-style file)
[-o OutFileBase] (extension will be added)
[-N Nevents] (<0 for all)
[-addpid pid] (can be called multiple times)
[-det detstring] handbook, perfect, ...
```



Dependent on input files! Here: small example included in the package

- Resolutions etc. for final particles:
- also have particle gun



Validation

- eic-smear is a **consumer** of parameterizations → can only check that implementation is **faithful to the provided input**
- Matrix-style: simple – fits to QA plots should confirm supplied σ
- "real" detector: no one-size-fits-all option – have to overlay fastsim with provided slowsim

Update on full simulations

Simulation of physics processes

Monte Carlo Event Generators

Simulation of detector responses

Fast simulations

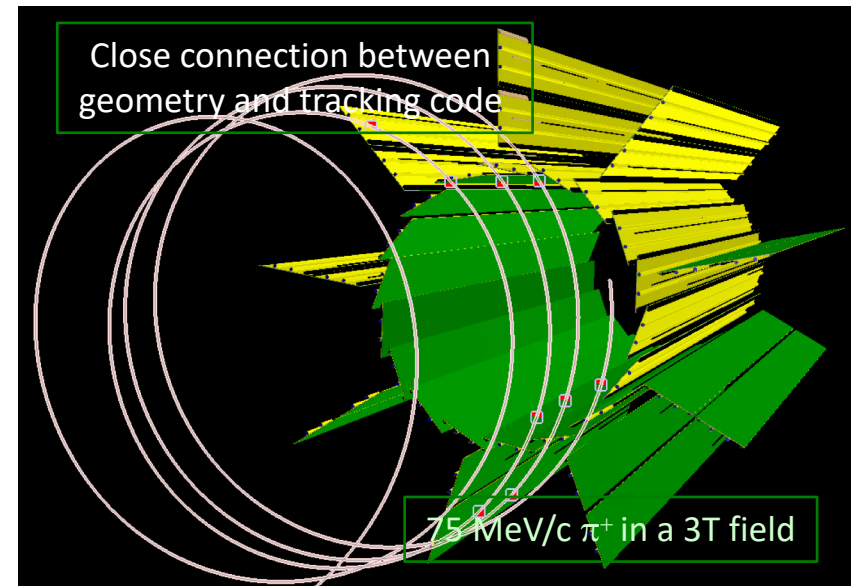
Full simulations

Physics analysis

Reconstruction of physics processes

- Re-packaged completely (no FairSoft bundle dependency)
- Switched to ROOT6; ROOT5 support discontinued
- Primary application(s):
 - ongoing tracking studies by various groups
 - universal detector geometry export to escalate & fun4all (?)
 - forward acceptance studies
- Software dependencies:
 - Latest versions of G4 (and G3 v3.21) VMC
 - GEANT4 v10.05.p01 (*more or less recent*)
 - ROOT 6.14.0 (*works with 6.20.4 as well*)
 - *eic-smear, Jana*
- Codes uploaded to <https://github.com/eic/EicRoot>
- Docker container available (presently based on vanilla CERN CC7)
- *Expert level* documentation, examples, cosmetics: this week
- Support for experts for (at least) the Yellow Report studies period

Yes, that's it.



- Simple command line interface for fast simulations
- g4e
 - Jleic magnetic fields for ERHIC beamline
 - Many changes to far forward region
 - Alice like ZDC
 - Fast simulation with limited decay depth
- ejana
 - Standalone plugins – better workflow with “analysis in its own repo” case
 - Fast simulations validation and DIS plots
 - HepMC3 support
 - Work in progress with ACTS team on full reconstruction - vertexing+tracking (now tracking is implemented with genfit)
- Pythia8 + Dire for DIS simulations, other MCEG support
- All updated tutorials
- Many smaller updates and bug fixes

- Added simple command line API for smearing:

```
smear any_eic_mc_file.txt
```

- Easy to change the detector and version (all detectors are versioned):

```
smear -d beast any_eic_mc_file.txt
```

```
smear -l      # list available detectors and versions
```

- Supports many smear engines. Now eic-smear but also “simple smear” (by Yulia Furletova). Delphes may be in future. Engine is selected automatically.
- Available on docker, singularity, on farms and in JupyterHub
- **Simple instruction**

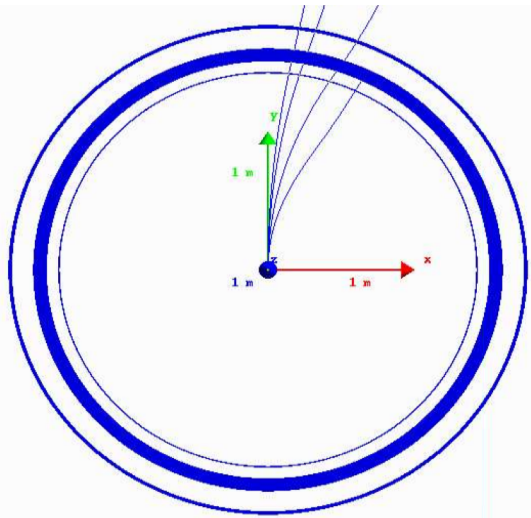
- Available both for docker, singularity, jupyterhub (experimental)
- *electroncollider/escalate:1.0.1* versions table:

Dev tools		HENP		MCEG		EIC	
Packet	Version	Packet	Version	Packet	Version	Packet	Version
gcc	9.2.1	root	6-20-04	LHAPDF6	6.2.3	ejpm	0.3.12
CMake	3.17.0	geant	10.6.1	pythia8	8.244	eic-smear	1.0.4f1
python	3.7.5	eigen3	3.3.7	DIRE	2.004	jana	2.0.2
True	True	clhep	2.3.2.2	Cernlib	2006-12-20	ejana	1.2.2
		hepmc	2.6.9	lhpdf5	5.9.1-6	g4e	1.3.4
		hepmc3	3.2.1	PYTHIA6	RAD-CORR		
		vgm	4.5				
		genfit	2020.1				
		acts	0.22.00				
		delphes	3.4.2				
		fastjet	3.3.3				

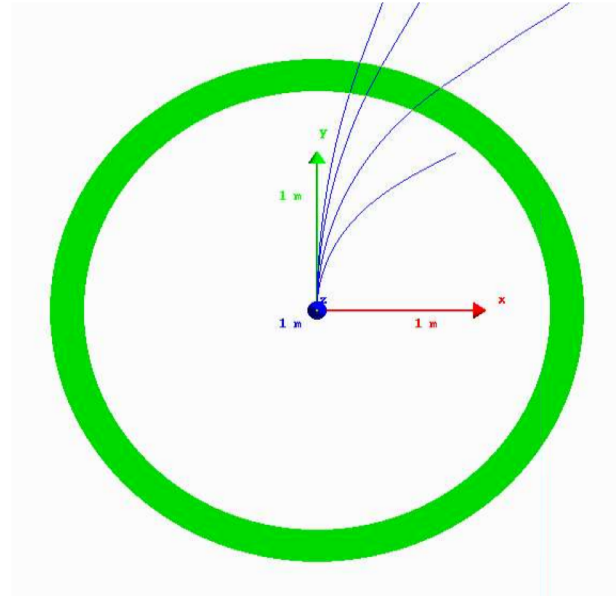
Fun4All Magnets:

Fieldmaps for Babar, Beast and JLeic, magnets in IP12 from data file

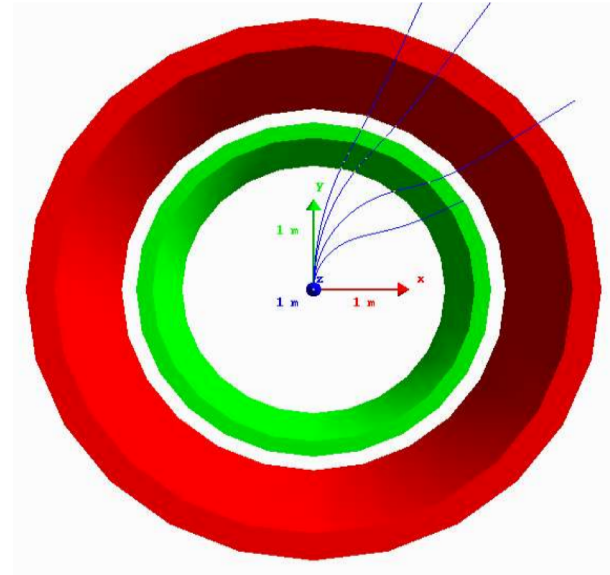
Chris Pinkenburg (BNL)



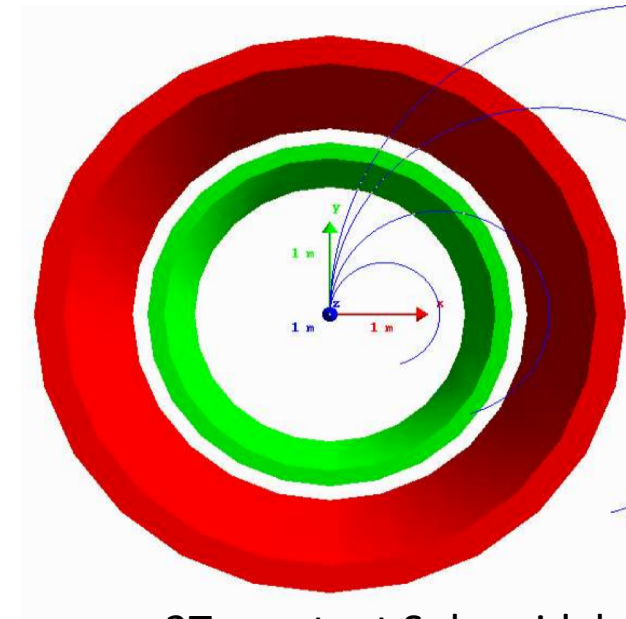
Babar 1.4T



JLeic 2T



Beast 3T



3T constant Solenoidal field for verification

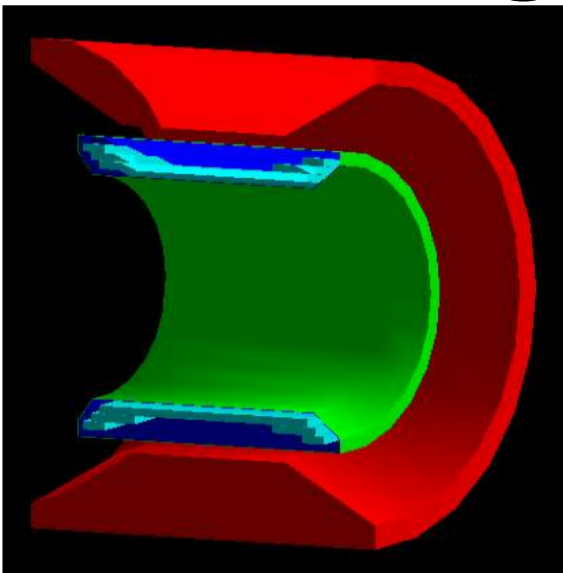
Tracks from charged geantinos:
0.5, 1, 2, 3 GeV/c
Verified fieldmaps by comparing with constant solenoidal field

IP 12 Magnets

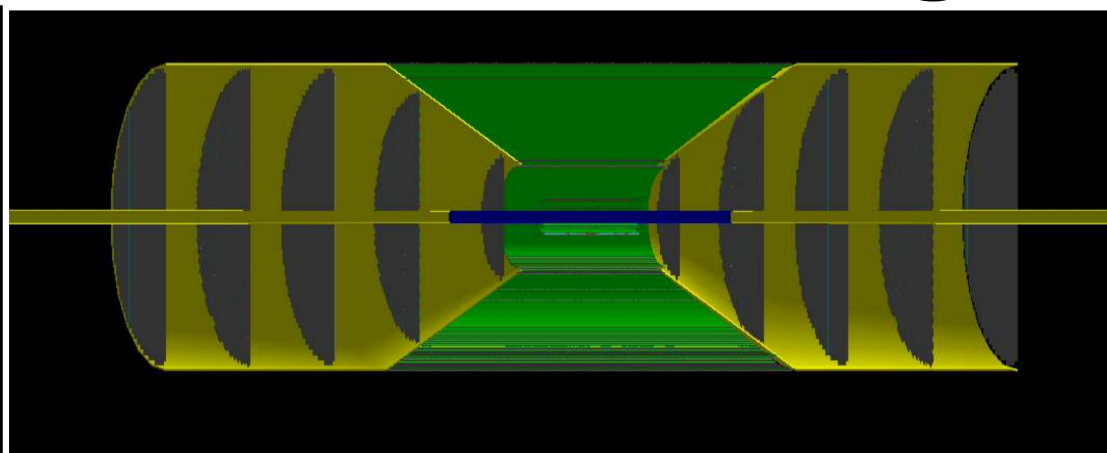


Fun4All Geometry: Importing Detectors from gdml files

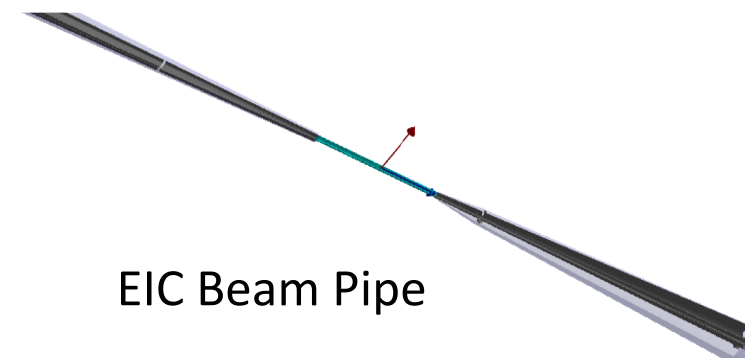
Chris Pinkenburg (BNL)



Detailed Beast Magnet



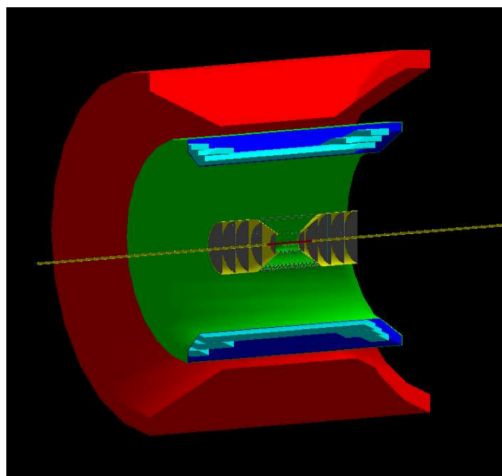
All Silicon Tracker



EIC Beam Pipe

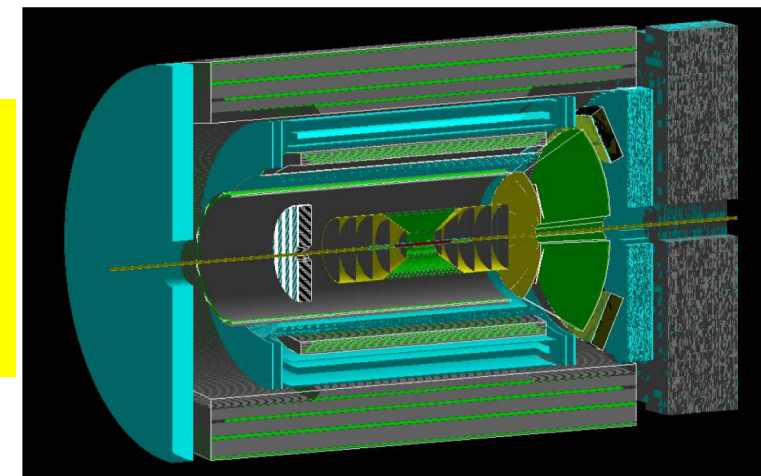
Exported EICRoot TGeo based
models to gdml

- Direct comparison
EICRoot/Fun4all
- Easy transition



Beast + All Silicon Tracker

adding/combining of
detectors from gdml
descriptions on the
macro level

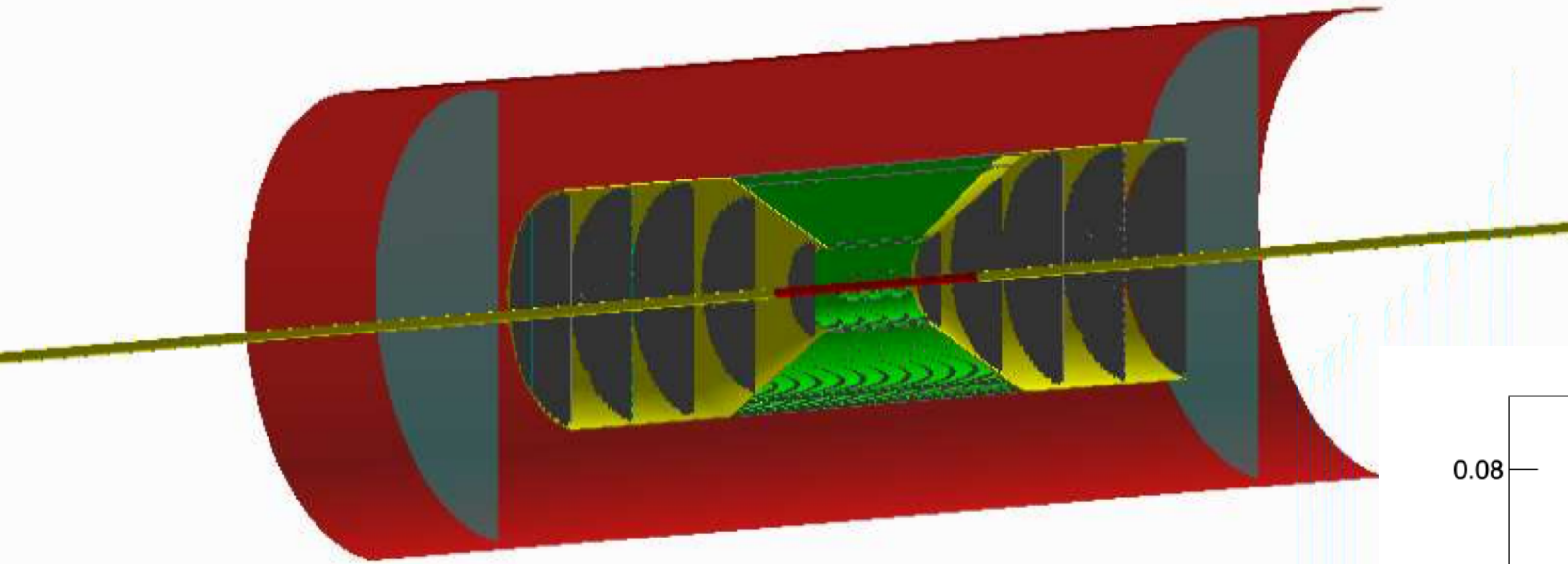


Babar + All Silicon Tracker

Fun4All Reconstruction: Fast Momentum Resolution

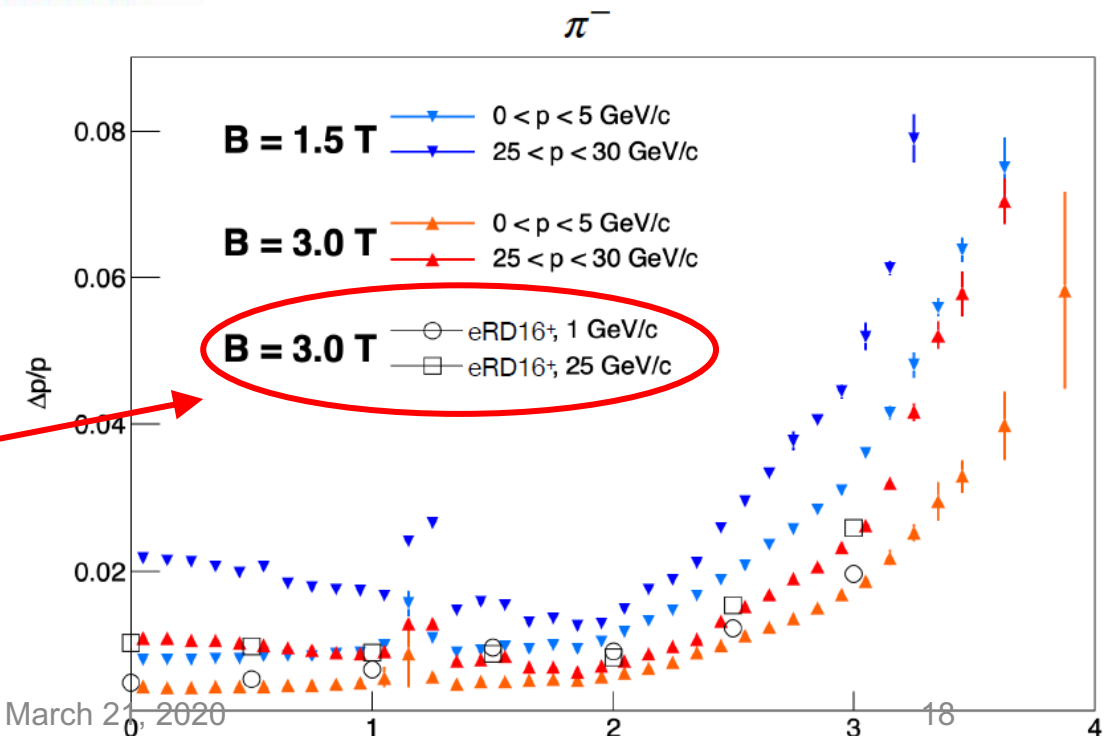
Chris Pinkenburg (BNL)

New capability: track
projections to arbitrary
cylinders and vertical planes
for PID detectors (and grey)

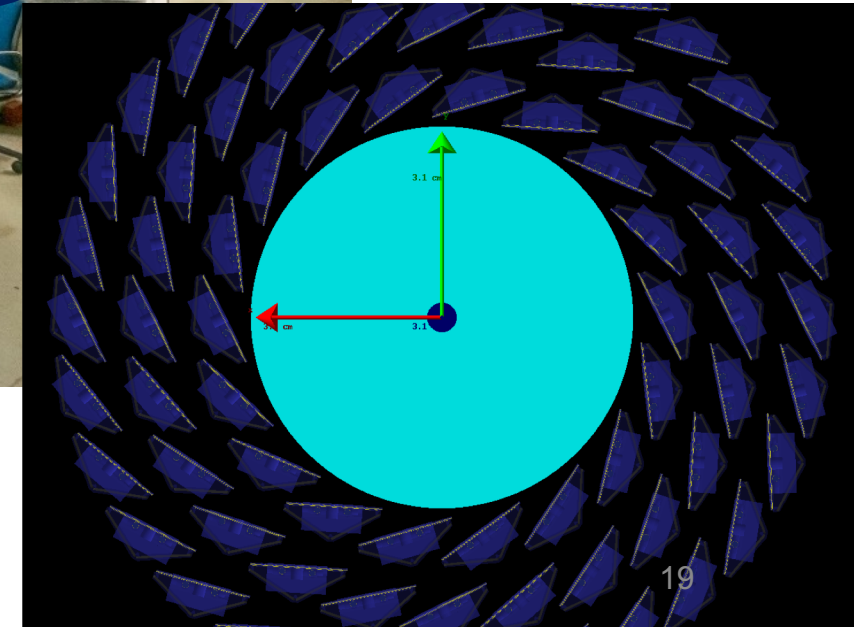


Only the silicon strips are active
Hits can be fed into fast momentum reconstruction on macro level
Results compatible with EICRoot:

- Identical geometry
- Similar track fitting
- Similar hit resolution
- Similar em physics



Fun4All Reconstruction: Test beam validation for Maps detector



The test beam results for the sPHENIX MVTX are being finalized leading to much more realistic maps detector descriptions

2nd EIC Yellow Report Workshop, March 21, 2020

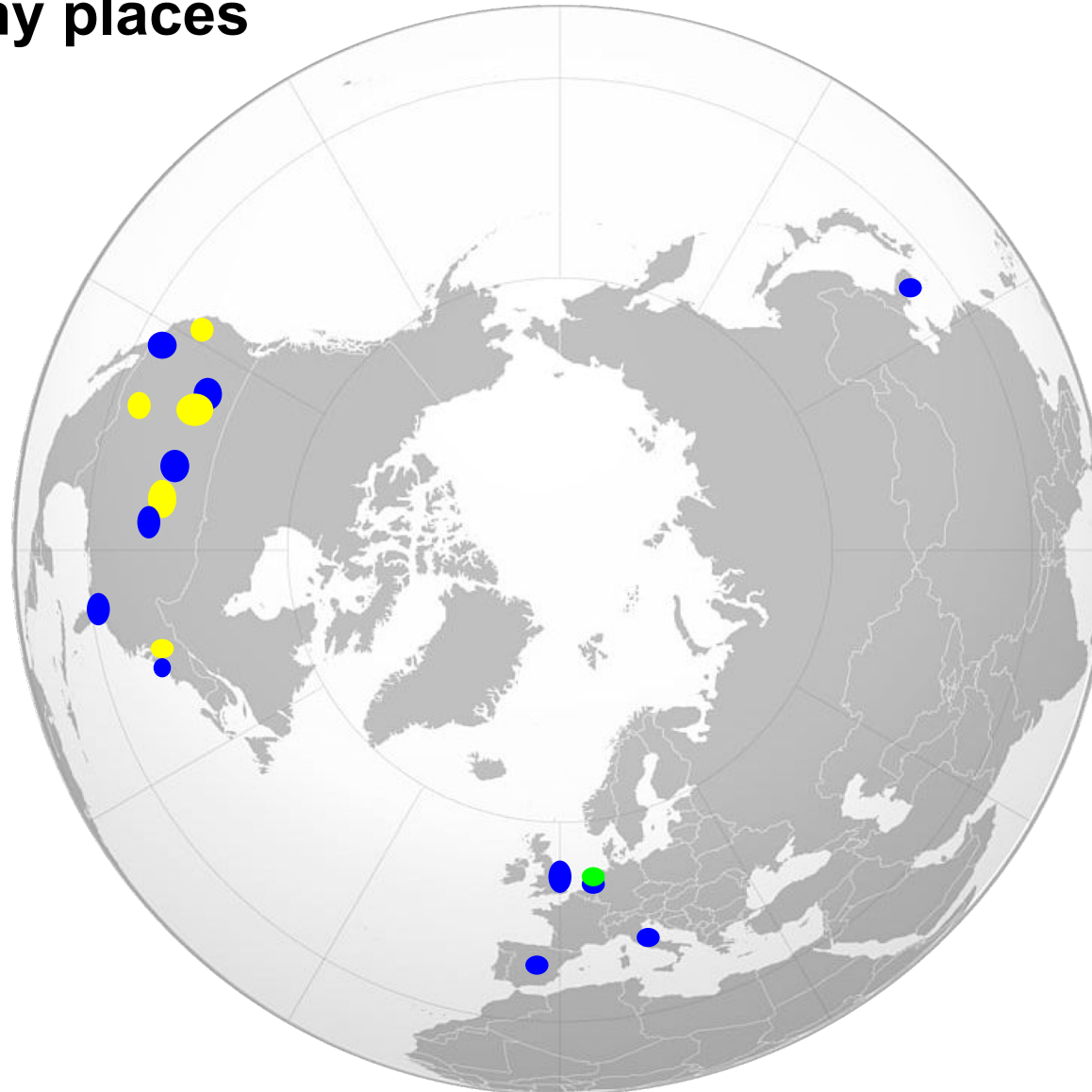
Approach is to use Federate resources as much as possible

- **cvmfs/oasis** as a production service - discussion on a common structure in CVMFS ongoing
- **Scientific document DB for experiments** Zenodo instance (@BNL) deployment exists for testing. InCommon Federated Identity supported.
- Excellent progress on the **OSG/EIC** front. Submit hosts ready on both sides - resources are near
 - Users at both JLAB and BNL have the option to run in both proprietary EIC resources and OSG opportunistic resources in a transparent way.
 - The model scales and is easy to add other submit points (i.e CNAF - under discussion). Example job can be found on the [OSG site](#)
 - +ProjectName = "EIC"
 - Next steps - xrootd as a production service (next week) - XCache

(X)Caches managed by OSG Graham Heyes (JLAB) and Jerome Lauret (BNL)

caches exists at many places

- Cache at institution
- Cache in the backbone
- Future Deployments



Storage

BNLBox / NextCloud (Kolja Kauder (BNL))

- **Available** storage on BNL RACF / SDCC, pre-generated events from Pythia, Milou, Djangoh and other MCEGs (13 TB and growing)
- **Read data** only requires guest account, web interface on <https://bnlbox.sdcc.bnl.gov>
- **Write data** requires BNL account
- **Documentation** <https://eic.github.io/computing/storage.html>

Petrel / Globus tentatively available starting next week (Sylvester Joosten (ANL))

- **Available** 100 TB of storage accessible from local system, computer farm, or HTC/HPC systems, upcoming mirror of MC data from BNL and JLAB
- **Read data** login to <https://globus.org> using your lab or university credentials
- **Write data** requires to be added to EIC Globus group
- **Documentation** <https://eic.github.io/computing/petrel.html>

Communication

eicug-software@eicug.org



The world's most powerful microscope for studying the "glue" that binds the building blocks of visible matter.

The EICUG Software Working Group is working on physics and detector simulations that enable for the Yellow Report Initiative a quantitative assessment of the measurement capabilities of the EIC detector(s) and their physics impact. The common simulation tools and workflow environment being setup by the working group allow the EICUG to pursue Yellow Report studies in a manner that is accessible, consistent, and reproducible to the EICUG as a whole.

Communication

The EICUG Software Working Group will start to announce software updates, known bugs, and other software related news on the eicug-software@eicug.org mailing list. While summaries of the announcements will be shared on eicug-users@eicug.org and the monthly newsletter, we encourage all working groups to subscribe to the eicug-software@eicug.org.

JupyterHub

We are preparing JupyterHub resources at BNL and JLAB that will allow you to run the EIC Software in the cloud and to easily share your analyses with other users. [A prototype is being tested at JLAB.](#)

Support us to support you better

The common simulation tools and workflow environment can only grow with user input. The EICUG Software Working Group asks interested users to join our support team to ensure fast response to user requests. This is an ideal opportunity to learn more about the software and its development.

Tutorials

The tutorials for fast and detector full simulations are available on the [YouTube channel of the EICUG](#). The next tutorials will be on Wednesday, April 8, and Thursday, April 9.

Uproot and Awkward Array Tutorial

- Jim Pivarski (Princeton), known for innovative software in [DIANA/HEP](#), [IRIS-HEP](#) and other projects, will give tutorials on how to process and analyze Root files with pure Python libraries: [uproot](#) and [awkward.array](#). The tutorials are aimed for anyone who

1

Mailing list please subscribe via Google Group

Have to start using it consistently and reliably for Software News

would like to learn more about using data science tools in Python for Nuclear Physics analysis.

- Wednesday, April 8, at 10:00 a.m. (EDT)
- Remote access will be provided via BlueJeans: <https://bluejeans.com/920347364>
- The tutorial will be recorded.
- More details are available on: <https://indico.bnl.gov/event/8242/>

Advanced Fast Simulations Tutorial

- Dmitry Romanov (JLAB) and Kolja Kauder (BNL) will give an advanced tutorial on fast simulations based on the feedback and requests of the last months.
- Thursday, April 9, at 10:00 a.m. (EDT)
- Remote access will be provided via BlueJeans: <https://bluejeans.com/920347364>
- The tutorial will be recorded.
- More details are available on: <https://indico.bnl.gov/event/8243/>

Software Update

In our first Software News, updates on `eic_smear` and `ESCalate` are included. Further updates on these and other packages will be in future newsletters.

`eic_smear`

Legacy version (1.0.3-final) The version used for the EIC White Paper (with a minor fix and some new tests) was fixed to 1.0.3-final for preservation.

Yellow Report versions (1.0.4+) The current version, tentatively named 1.0.4-RC, has a few impactful updates:

- The code was made compliant to C++17, the latest stable C++ standard.
- A bug was found and fixed that translated negative smeared values into NaN. This affects only a small phase space, mostly for very small phi values, but it is nevertheless recommended to update at your earliest opportunity and check for differences in your studies.
- Calculations for x and y according to the Jacquet Blondel method can produce a slew of warnings because values slightly smaller than 0 or larger than 1 are generated through roundoff errors. This is fixed.
- Testing and Q&A macros have been added in the `tests/` directory.

In the works

- Implementations according to the [Electron-Ion Collider Detector Requirements and R&D Handbook](#): Once out, this should be the default used by everybody until vetted reference detectors are compiled. Note that none of the existing smearers were ever meant as truly representative of their namesakes, mostly as starting points.
- pT and pz smearing is currently not functional. While no existing smearers rely on them, this is nevertheless a bug that will be fixed in the next update. More examples and QA are to follow as well.

2



ESCalate framework

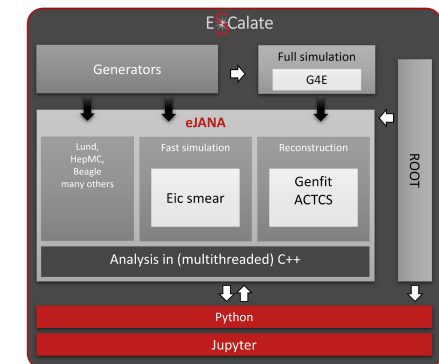
A lightweight modular fast and full simulation framework for EIC.
<https://gitlab.com/eic/escalate>

Previously separate packages, G4E and eJana, are now distributed as the ESCalate framework. ESCalate brings together [Monte Carlo event generators](#), [eic-smear](#) for fast simulations, [G4E](#) for full detector simulations, as well as [JANA2](#) and [eJana](#) for reconstruction. The structure of the ESCalate framework is shown in [Fig. 1](#). Part of ESCalate are [eipm](#) for distributing the software and a variety of smaller tools. In addition to [tutorials](#), examples for [Geant4 detectors](#), [eJANA plugins for reconstruction and physics analyses](#), and [analyses in JupyterLab](#) are provided.

ESCalate provides modularity in two ways:

- Subpackage modularity: Each package inside the framework can be used solely and separately from the rest of the framework.
- Subprocess modularity: JANA2 (and eJana) modularize code in small libraries called plugins allowing to run tasks in a seamless multithreaded way.

The framework itself ensures that data between packages is consistent and output of one package can be correctly read from another. ESCalate provides Python orchestration to facilitate the configuration of each package and allow for a workflow between the packages.



3

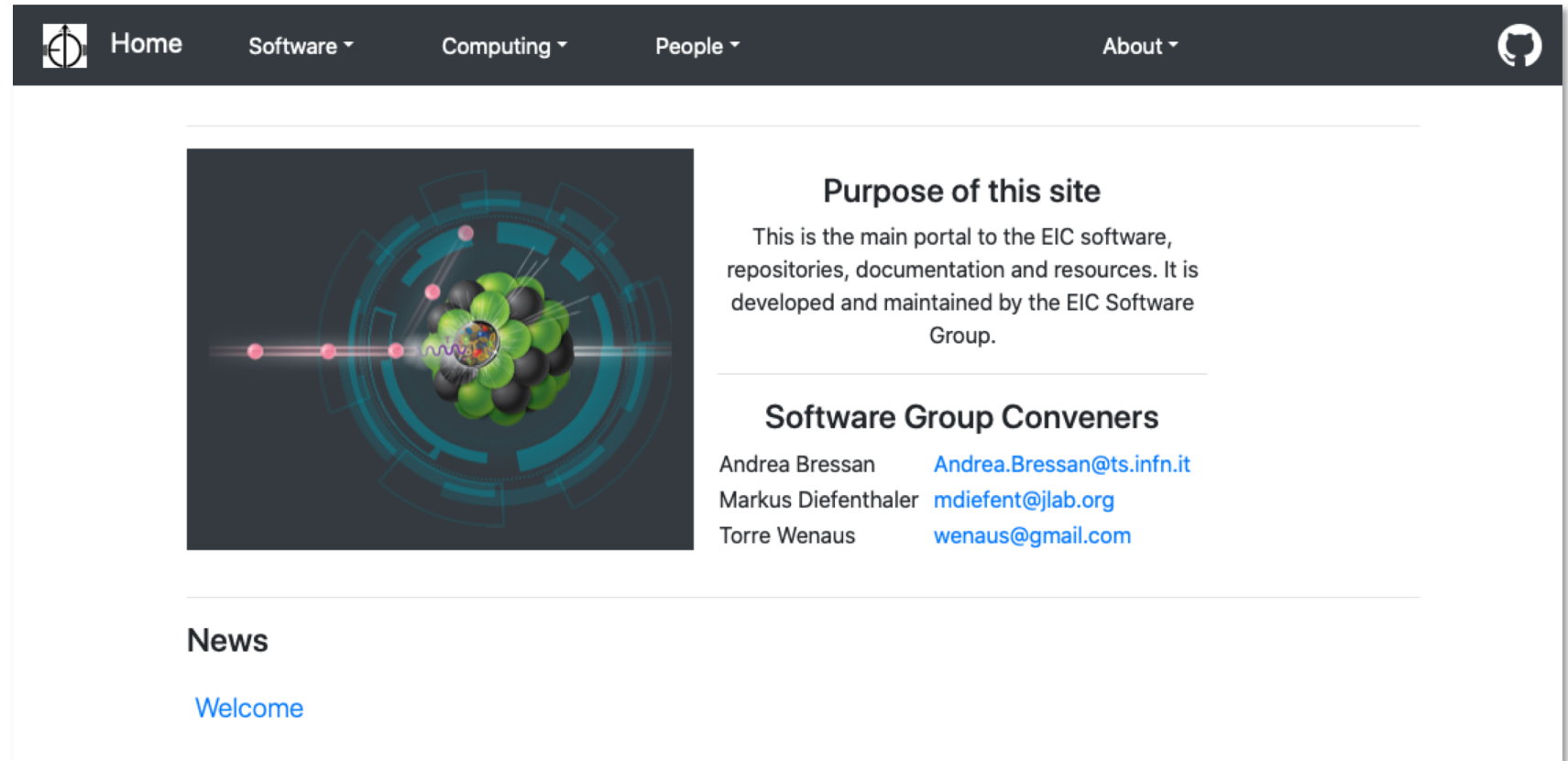
Collecting, organizing, and documenting EIC Software

new GitHub organization for the EIC community <https://github.com/eic>

- Please help us to make your software available on the GitHub organization.

<https://eic.github.io>

- **new** SWG website
- based on HEP Software Foundation website that Torre et al. created
- set up by Maxim
- regular updates on content and structure by SWG
- available for your group and your documentation



Support each other

Develop

Workflow environment for EIC simulations

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

Support

software-support@eicug.org
mailing list anyone can contact

<http://eicug.slack.com/>
with software-support channel

in-person support

Involvement from EICUG

- **Coordinate simulations** Please continue to reach out to us or join the SWG.
- **Analysis preservation** Please make your software available and integrate it.
- We rely on your expertise:
 - **Design detectors**
 - **Developing reconstruction algorithms**
 - **Develop physics analysis**

EICUG Software Working Group

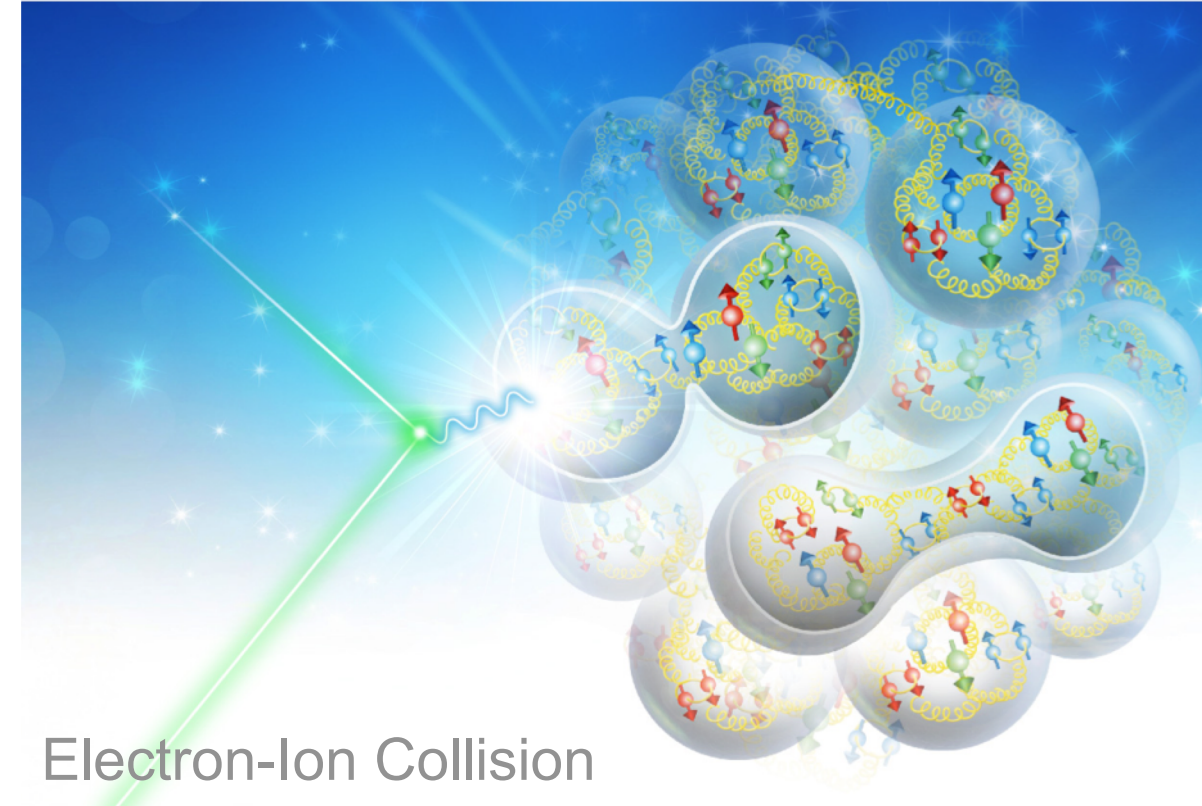
Workflow environment for EICUG

- **fast and full simulation** available and being extended with community input
- **documentation** started and being improved with community input
- **support** available

Grow with user input

- excited to be core part EIC Physics and Detector Conceptual Development / Yellow Report

Discussion on DWG+PWG+SWG workflow



Electron-Ion Collision

Discussion on DWG+PWG+SWG workflow



Role of PWG, DWG, and SWG



- **PWG+SWG:** “Are there, at least for some event-generators, QA-ed baseline settings and inputs e.g. for PDFs?” To what extent can and will the PWG help here?
- **DWG+SWG:** “What’s needed are “true” parameterizations. What is the timeline and procedure to get those?” How close are we to Yellow Report detector(s)?
- There is a lot of activity by the DWG+PWG+SWG but many opportunities for working more together, e.g., “**Could the roottrees from the various kinematic maps across the studies be made available in a uniform format and with macros for different needs?**” Project by PWG. Could be added to GitHub, integrated in EIC Software collection, and developed together.

Collecting, organizing, and documenting EIC Software

new GitHub organization for the EIC community <https://github.com/eic>

- Please help us to make your software available on the GitHub organization.



Involvement from EICUG

Let's store **now** all macros, scripts, and software tools on <https://github.com/eic> This will help us to understand and reproduce our studies.

How can we compare results?

- “EICsmear and DELPHES have both been used to perform initial event-smearing studies. (...) What cross-validation has been or needs to be done vis-a-vis EICsmear?”
- “GEANT-based simulations have been performed with EICroot, G4E, and Fun4All. What will be the approach going forward to ensure that the results will be archived and reproducible? What cross-validation of results exists and what needs to be done to ensure that studies from these tool-sets can be compared on equal, or at least understood, footings?”

Any study you share with the SWG can be used to **benchmark & validate** the EIC Software tools. If you send us analysis scripts and macros, the SWG can reproduce your studies and build up a **validation scheme and tools** on top of it. But this can only work with strong support by the DWG and PWG. The validation suite would be run by DWG+PWG+SWG.

Benchmarking

- There is the need of a defined set of plots for benchmarking different versions, fast and full simulations
- It will be easier if there is agreement on a unified format for the output ROOT tree resulting as “end of processing” simulation chains.
- This can be organized in well structured branches
 - True/reconstructed kinematic variables ($x, y, Q^2, \nu, W \dots$)
 - Relevant charged, neutral particles and jet information in Lab, GNS...
 - Some switches to turn on/off branches if not needed by the physics channel
- And analyzed with common scripts

Would the DWG+PWG+SWG support that?
Requires input from DWG+PWG+SWG.

How to grow the EIC Software collection?

- *"EICsmear and DELPHES have both been used to perform initial event-smearing studies. Will or should this effort adopt DELPHES? If so, what will be done to ensure it gets out of the realm of private versions?"*
- "Delphes seem to have some significant advantages in particular for jet and HF physics that will be difficult to replicate in other smearing codes. Do you plan to support it in the existing framework?"

Status

- PWG would like to avoid any duplication of the effort and have asked SWG earlier for one and only one tool for fast simulations, eic_smear. However, DELPHES is being used in the PWG.

Path forward

- This is an example for the discussion about how to grow the EIC Software Collection.
 - **In general** Even if we would have more manpower, we will not be able to support any tool. Here, we rely on guidance from the community.
 - **Particular for Delphes** Someone would have to update Delphes parametrizations (DetectorMatrix -> Delphes) and maintain the Delphes validation scripts

How to build the EIC Software collection?

Question on Pythia 6 vs. Pythia 8

- There seem to be different MC codes for SIDIS events propagated by the SW group. In particular Pythia 6 and Pythia 8 with DIRE. Our studies showed substantial difference in the simulated events. Are you planning to select on default MC, tune, so simulations become comparable? We realize that we could do this within our WG, but it seems a more general question.
- Why is Pythia 8 promoted as a generator even so it does *not* reproduce the HERA data and gives factors of different c.s. to Pythia 6 which reproduces *all* the data?

Example for discussion when to adapt modern MCEG for the EIC

Next slides A discussion by C. Bierlich (Copenhagen, LUND), M. Diefenthaler (JLAB), I. Helenius (Jyväskylä), S. Joosten (ANL), S. Prestel (LUND)

Pythia (1978 – now)

General-purpose MCEG

- extensively used for e^+e^- , ep and pp physics, e.g. at LEP, HERA, Tevatron, and LHC
- as a building block used in heavy-ion and cosmic-ray physics
- recent pA effort in Pythia 8 with Angantyr model

Pythia 6 product of over thirty years of progress

Pythia 8 successor to Pythia 6, standalone generator, but several optional hooks for links to other programs are provided

December 2012

Dear Pythia Users and Supporters,

We would like to communicate to you some important information regarding the development of the Pythia event generator.

Over the years, we have enjoyed a fruitful relationship with the major collider experiments. You have provided important feedback on the comparison of data and predictions, noted errors in the code, and made suggestions for improvements. We have responded to requests, helped debug code when necessary, and been a partner in the data analysis process. We look forward to continuing this relationship in the future.

A key request of the LHC community has been for us to transition from Fortran to C++. We have been manpower-limited, so that project has taken much longer than it ought to have. However, since some time now, the new Pythia 8 code should be able to do just about everything the old Pythia 6 code could, and then some more.

We have retained a commitment to Pythia 6 because of its continued use in Run 2 and the early stages of the LHC. However, we find it is no longer tenable to split our resources between the development of Pythia 8 and the support of Pythia 6.

Therefore we here present our timeline for transitioning to a frozen version of Pythia 6, that will then remain a legacy code.

Development of Pythia 6 now stops. We will still provide support and urgent fixes to the code, if necessary, until 1 March 2013. At this point, the Pythia 6 code will be frozen, and a final legacy version will be released later in 2013. We will then continue to answer questions regarding the behavior of Pythia 6 until 1 July 2013, after which only Pythia 8 will be actively developed and supported.

We believe this decision will allow us to have the most positive impact on the field. We look forward to continuing our current relationship based on Pythia 8.

Sincerely,
The Pythia6 Collaboration
(Torbjörn Sjöstrand, Stephen Mrenna, Peter Skands)

Why has DIS been first missing in Pythia 8

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

Problems with default parton shower for DIS (used in Pythia6)

- The parton shower has been developed for positron-electron annihilation and Drell-Yan.
- The parton shower is using a \hat{s} approach where $\hat{s} = x_1 * x_2 * s$ at all scales. This works well for hadron-hadron collisions, e.g., for preserving the W/Z mass in the parton shower.
- When expanding the parton shower for electron-hadron scattering, one has to replace one incoming parton with an electron at $x=1$. The Bjorken- x value of the event will not be preserved during the reconstruction of the initial state shower, as the introduction of the transverse momentum will change the value of $P * q$. This also implies that the cross-section is changed.
- This was solved (for a single splitting) by a very specific handling of the initial and final state cascades and limiting the maximum allowed virtuality to W^2 with additional rejection techniques.

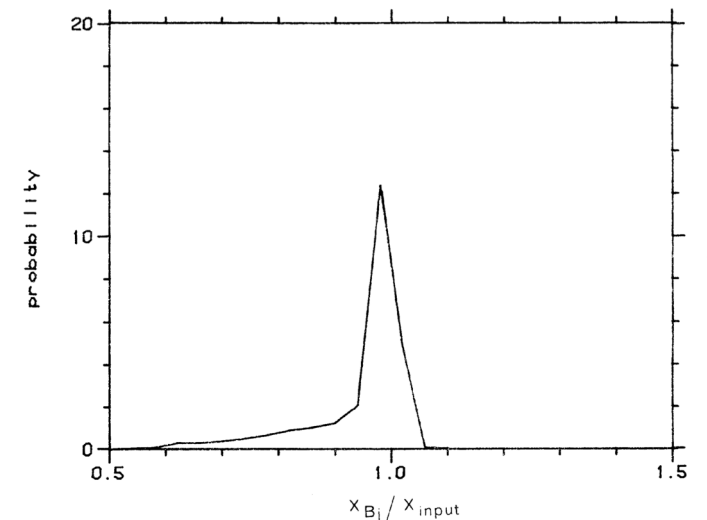


Fig 4

Status of ep simulations in Pythia 8

DIS

New option for dipole-recoil shower introduced by Torbjörn and B. Cabouat (EPJC 78 2018 no.3, 226) which addresses the limitations of the default shower for DIS. Reasonable agreement for some HERA DIS data as shown in the paper but not extensively tested nor tuned. DIRE should provide a more complete setup but more studies would be welcome. The phase-space generation still needs some refining.

Photoproduction

Photoproduction (for quasi-real photons so $Q^2 = 0$) have been implemented into Pythia 8 and compared to Pythia 6 and HERA data for dijets and charged hadron production. Using the same PDFs etc. the partonic cross sections are the same in Pythia 6 and 8 but there are of course some minor differences due to the updated shower implementation and small changes in the hadronization.

Transition region ($1 \text{ GeV}^2 < Q^2 < 10 \text{ GeV}^2$)

So far no implementation is present for this region. This is something we have made plans to consider in detail later on but so far left as an open question. Note that in Pythia 6, a description of the transition region is available, heavily relying on tweaking parameters. Thus, Pythia 6 cannot provide a predictive model, and is thus dangerous to use.

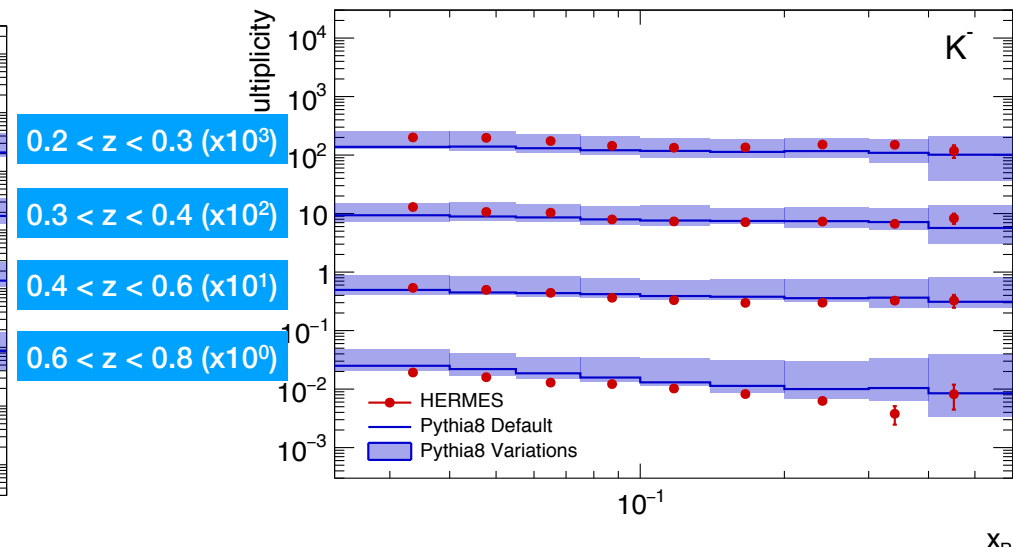
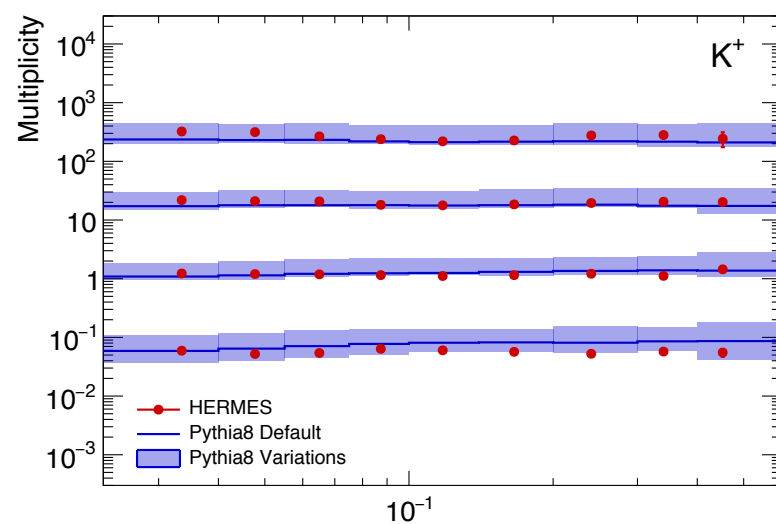
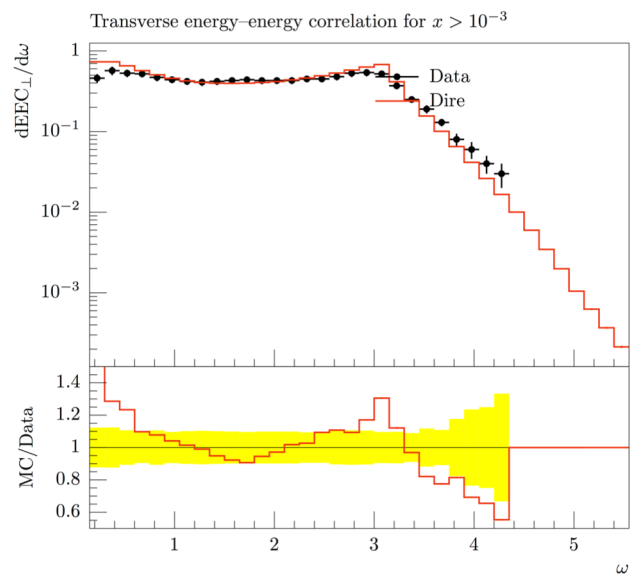
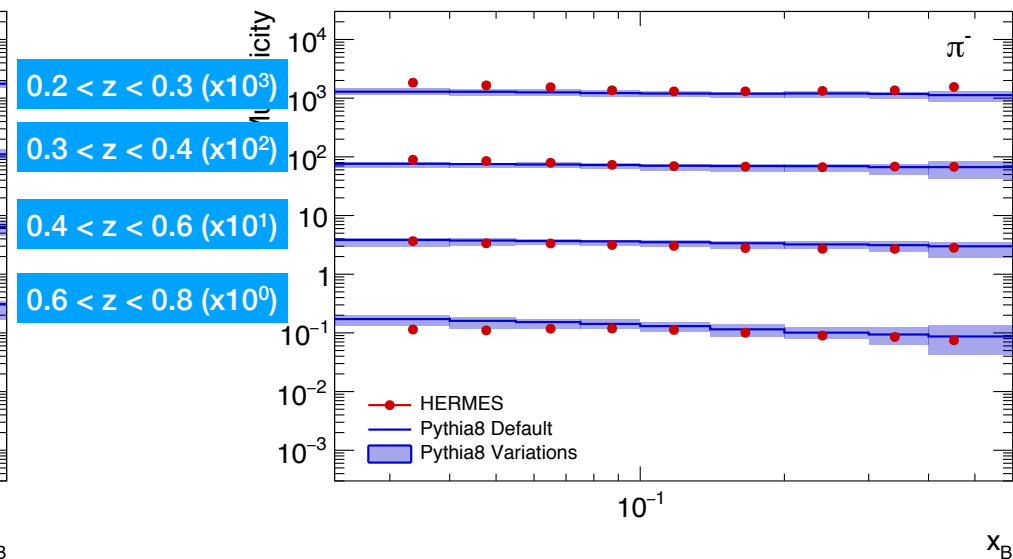
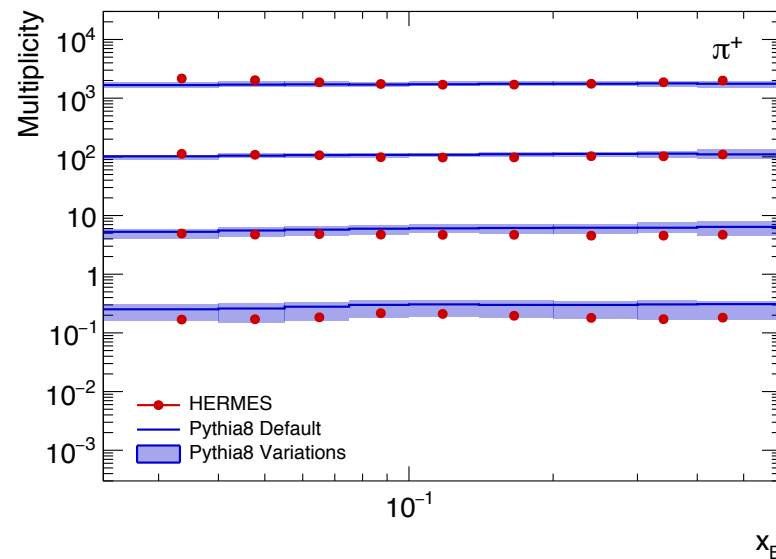
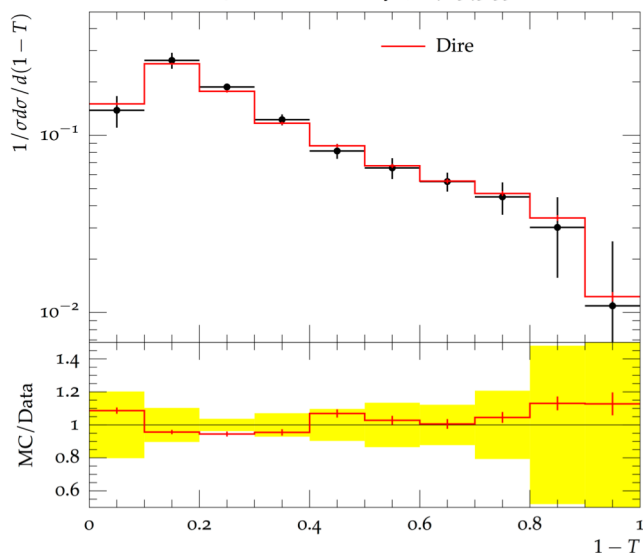
Diffraction

Soft diffraction in photoproduction includes the same parametrizations as Pythia 6. Exclusive vector-meson production cross sections is in a fair agreement with HERA data for low-mass states (ρ , ω) but not for high-mass states (ϕ , J/ψ).

Hard diffraction (like dijets) is not implemented in Pythia 6 but the new Pythia 8 implementation provide a fair description of the ZEUS and H1 data for photoproduction. Diffractive DIS have not been studied since the main feature of the model, MPI rejection, would not do anything there.

DIS in Pythia 8 (early studies by MD, SJ, SP)

H1 data, $14 < Q < 16$ GeV, Eur.Phys.J.C46:343-356,2006



DIS with Pythia

New shower option: dipoleRecoil

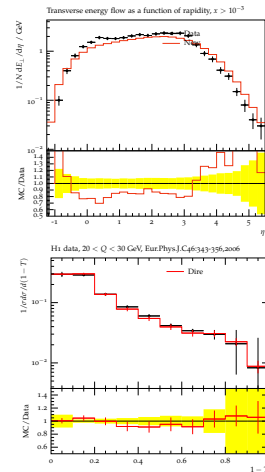
[B. Cabouat and T. Sjöstrand, EPJC 78 (2018 no.3, 226)]

- No PS recoil for the scattered lepton
- Reasonable description of single-particle properties, such as transverse energy flow
- Results based on tune with the default global-recoil shower

DIRE plugin (to be included in PYTHIA 8.3)

[S. Höche, S. Prestel, EPJC 75 (2015) no.9, 461]

- Correct soft-gluon interference at lowest order
- Inclusive NLO corrections to collinear splittings
- Good agreement with HERA data for thrust T



10

Hard diffraction in photoproduction with PYTHIA

Implemented from PYTHIA 8.235

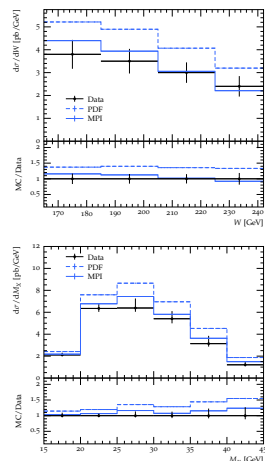
[I.H., C. O. Rasmussen, Eur.Phys.J. C79 (2019) no.5, 413]

- Based on dynamical rapidity gap survival [C. O. Rasmussen, T. Sjöstrand, JHEP 1602 (2016) 142]
- Begin with factorized approach with diffractive PDFs (Ingelman-Schlein picture)
- Reject events where MPIs between resolved γ and p would destroy the rapidity gap

Comparison to HERA diffractive dijet data

[H1: EPJC 51 (2007) 549, ZEUS: EPJC 55 (2008) 177]

- More MPI suppression towards higher W , M_X
- Natural explanation for observed factorization breaking in pp and γp



12

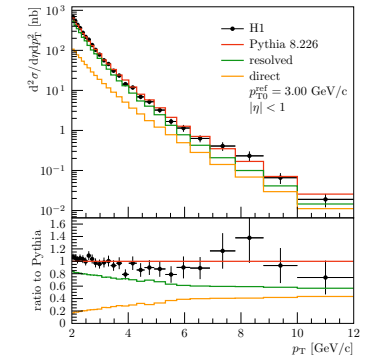
Photoproduction with PYTHIA

Photoproduction in PYTHIA 8

- Hard and soft QCD processes
- Mix of resolved and direct processes
- Photon PDFs from CJKL fit
- MPIs for the resolved processes \Rightarrow Regulated with p_{T0} as in pp
- Applicable also for UPCs

Inclusive hadron spectra from H1

- Resolved contribution dominates
- Good agreement with the data using $p_{T0}^{\text{ref}} = 3.00$ GeV (pp: $p_{T0}^{\text{ref}} = 2.28$ GeV) \Rightarrow MPI probability reduced from pp



[H1: Eur.Phys.J. C10 (1999) 363-372]

11

Exclusive vector meson production with PYTHIA

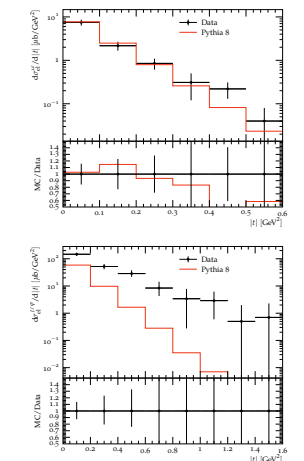
Implemented from PYTHIA 8.240

[I.H., C. O. Rasmussen, in progress]

- Based on (pre-HERA) SAS parametrizations [G.A. Schuler, T. Sjöstrand, Phys.Rev. D49 (1994) 2257-2267]
- Includes ρ , ω , ϕ and J/ψ production via elastic scattering

Comparison to HERA data

- Good agreement with low-mass mesons (ω)
- Underestimate heavy-meson (J/ψ) production \Rightarrow Require improved parametrizations using HERA data



13

Pythia 6 vs. 8

Pythia 6 is nice on the user end, because with the click of a button, you can produce something that looks like data, without thinking too much about the limitations of the models. This is, however, also the drawback. The modelling of the intermediate region in Pythia 6 is not really on solid ground and limits its predictive power.

The same thing goes for the elaborate tuning efforts carried out by experiments. Once an MC has been tuned to give a full description of a particular data set and little else, all predictive power is lost, and the MC at that point basically functions just as a parametrization of that data set. While parametrizations can be good, there is no guarantee that a specific tune will have anything reasonable to say about EIC. Basing R&D decisions on such a specific parametrization of data, can be outright dangerous.

Ideal path forward

- comparison of Pythia 8 + DIRE to HERA data (**ongoing**)
- comparison of Pythia 6 with current tune to HERA data
- discussion of Pythia 6 intermediate region model and why it is not continued
- discussion on difference between global and local tunes

Path forward for Yellow Report initiative?

What is required / expected?

“Do we have or do we anticipate to have stable /common /recommended versions for studies going into the Summer EICUG collaboration meeting and/or the third workshop?”

Putting the focus on full simulations

- **Frameworks** EicRoot (Virtual Monte Carlo: GEANT-3/Geant4), EScalate (Geant4), Fun4All (Geant4)
- **EIC Generic Detector R&D** Handbook detector documented, only sub-detectors implemented in EicRoot, EScalate, or Fun4All
- **Yellow Report Initiative**
 - Detector design ongoing, only sub-detectors implemented in EicRoot, EScalate, or Fun4All
 - Full detector simulations required, e.g., for benchmark studies

Priorities of Yellow Report Initiative

- Detector simulations keeping the path open to alternative technologies
- Foundation for full detector simulations including all materials, support structures welcome where possible

Priorities for full (and fast) detector simulation software

← We need your guidance!

Versioning

PWG deploy the simulation software in a way that each version number refers not only to a given version of our simulation tools but also to a given set of accelerator and detector parameters

SWG

- Eic-smear Detector versioning and decoupling of detectors from eic-smear repository has been implemented .
- detector parameters are documented in the interactive DetectorMatrix
- **link the DetectorMatrix with our software repositories**
 - add version numbers to the DetectorMatrix
 - add a field to the DetectorMatrix that would allow to specify a direct link to the software repository where the parameterizations are stored
 - Walt Akers (JLAB) will update the DetectorMatrix accordingly.
 - In the discussion, we also wondered about highlighting the fields in the DetectorMatrix that have been changed since the latest version so that it would be easier to follow up on changes to the DetectorMatrix. (tbd)

DWG+SWG

- DWG update parameters and add references
- At a certain moment, DWG freezes a given set of parameters and assigns a version number.
- SWG translates the information in the DetectorMatrix into parameterizations for simulation tools.

Further discussions

