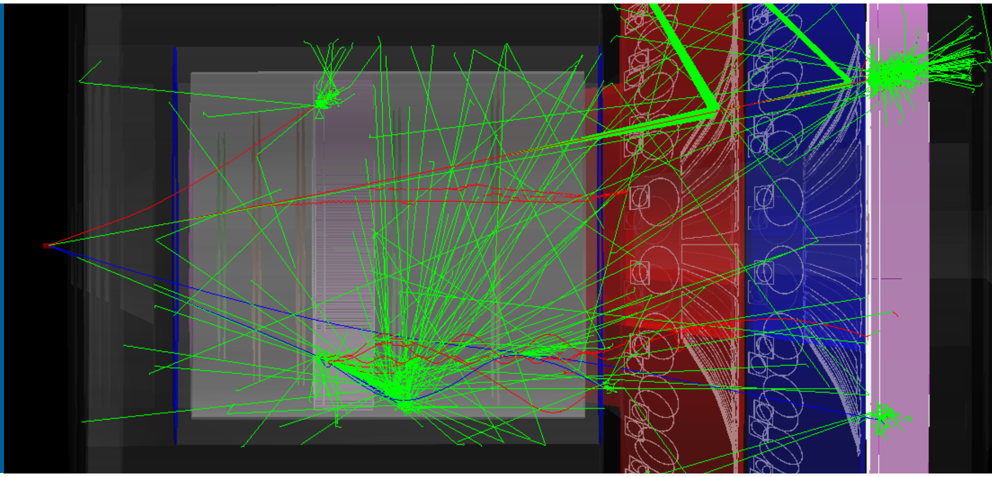
TOWARDS AN END-TO-END SIMULATION

AN EXAMPLE: SOLID J/Ψ CASE



SYLVESTER JOOSTEN
sjoosten@anl.gov

WHITNEY ARMSTRONG
JIHEE KIM
ZEIN-EDDINE MEZIANI

CHAO PENG
TOM POLAKOVIC
JUNQI XIE

U.S. DEPARTMENT OF ENERGY — Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC.

This work is supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics, under contract DE-AC02-06CH11357.

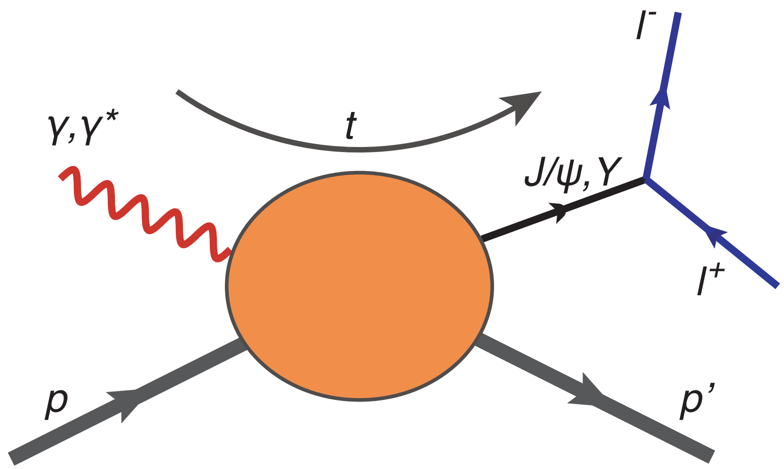
Remote Solid Collaboration Meeting
June 8, 2020

SIMULATING J/PSI PRODUCTION

Use *J/ψ* electro-production (“3-fold coincidence”) as example

- Electro-production
  - Measure **scattered electron and decay leptons**
  - t-channel J/ψ rate: ~90/day
  - Clean signal (less background)
  - Closer to threshold

- Photo-production
  - Measure **decay leptons and recoil proton**
  - t-channel J/ψ rate: >1600 per day
  - Ultra-high rate



MONTE-CARLO GENERATOR

lAger l/A event generator

- Meant to be a general purpose generator
- Currently implements various models for *J/ψ* and *Y* production
- Available to the public
- Standard HepMC3 output works perfectly with DD4hep

l/A-event Generator

This is the Argonne generic l/A-event generator (`lAger`), a flexible MC generator system to simulate electro- and photo-production off nucleons and nuclei.

Below you can find an overview of the release versions, as well as a short tutorial and copyright notice. If you use lAger to generate data used in a presentation or an article in a scientific publication, please cite:

S. Joosten, Argonne l/A-event Generator (2020), GitLab repository, https://eicweb.phy.anl.gov/monte_carlo/lager

Versions

- v3.1.0 First stable release version of `lAger` .

Tutorial

Setup of the lager singularity container on your system:

The default mode to run the generator is through singularity. To setup the generator on your system, first ensure singularity is installed. Then follow these instructions:

1. Clone this repository and checkout the desired stable release (e.g. v3.1.0)

```
git clone https://eicweb.phy.anl.gov/monte_carlo/lager.git
cd lager && git checkout v3.1.0
```

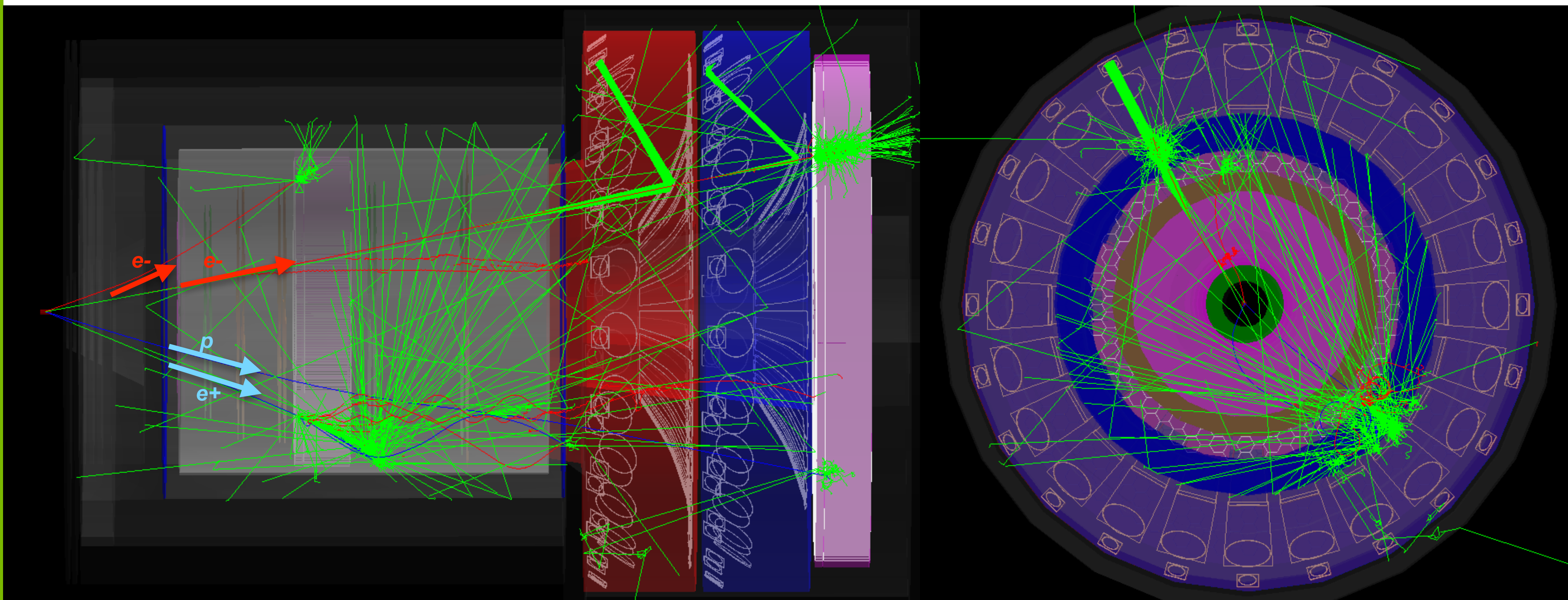
2. Run the `deploy.py` script to install the container to a prefix of your choice, e.g. `$HOME/local/opt/lager` .

```
./deploy.py $HOME/local/opt/lager
```

https://eicweb.phy.anl.gov/monte_carlo/lager

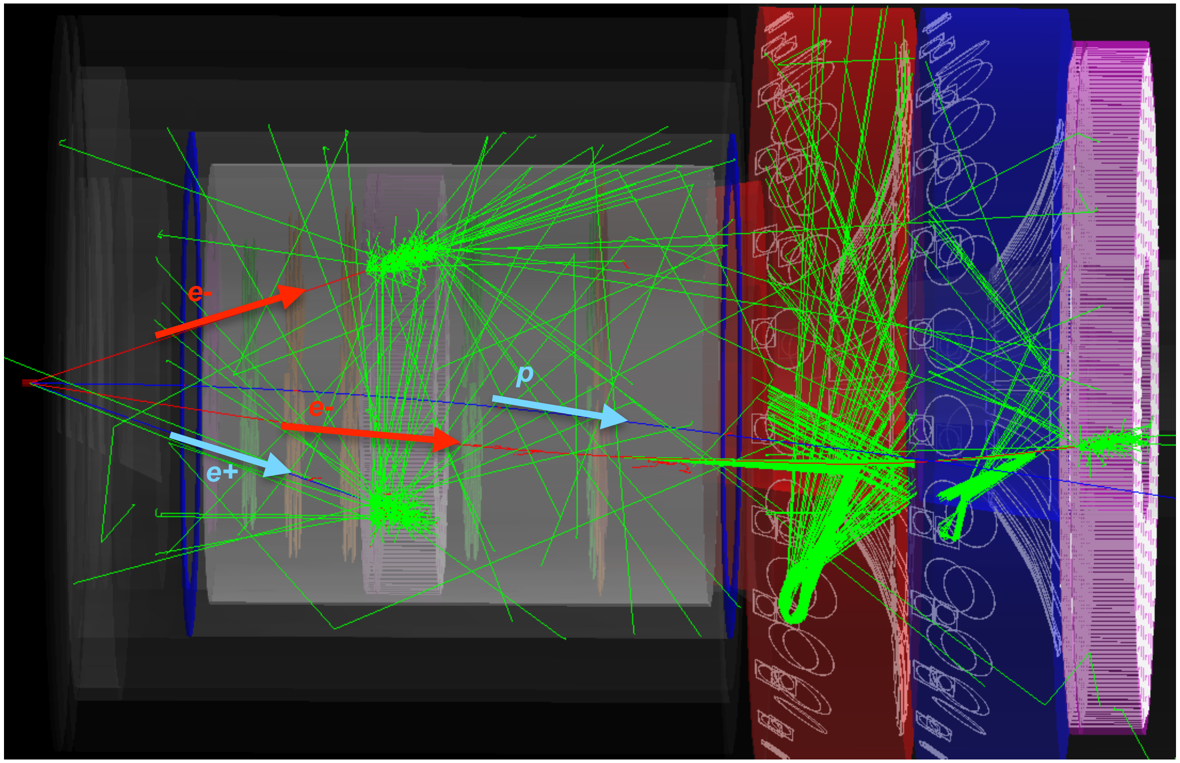
4-FOLD COINCIDENCE J/Ψ EVENT

From new DD4hep software



3-FOLD COINCIDENCE J/Ψ EVENT

From new DD4hep software

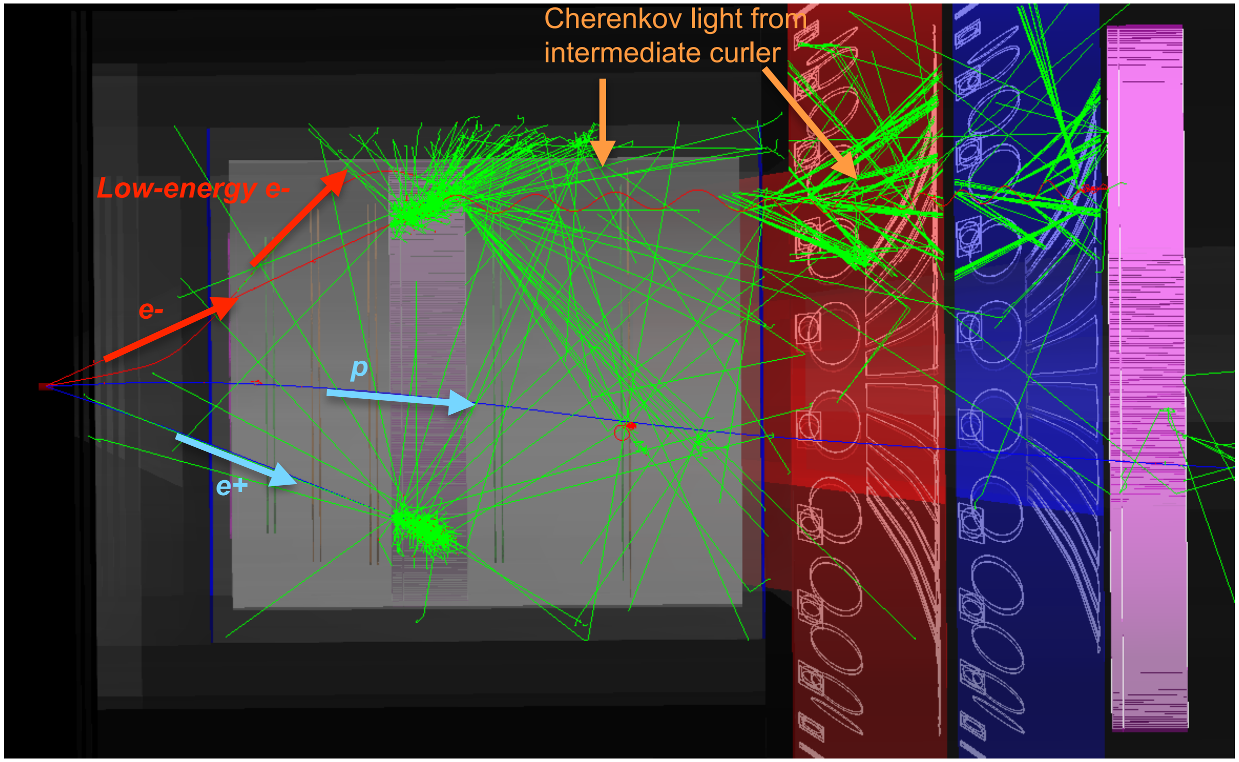


- Recoil proton barely misses detection
- J/psi decay products in at large angle
- Scattered electron in forward detector.

UNDETECTED J/Ψ EVENT

From new DD4hep software

- J/psi decay products in at large angle
- Low-energy scattered electron not detected
- Recoil proton again barely misses detection



NEXT STEPS

A full showcase of the new software

1. **Cross-validate** old gemc implementation with new DD4hep implementation:
   - Start from EC background rate using Wiser generator
   - Will be simple example of better workflow through DD4hep
2. Use closed-loop **simulation + reconstruction** to ensure **optimization** for *J/ψ*
3. Update ***J/ψ* trigger** for science review (high-priority!)
4. Leverage our experience to be **comprehensive example** of new software
   - Will naturally highlight pro's and con's to collaboration
   - Full implementation for SoLID-Jpsi will be a good guide for other configurations!

MORE TO COME SOON!

This work is supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics, under contract DE-AC02-06CH11357.

Comparing status quo with new software ecosystem

ARGUMENTS FOR NEW SOFTWARE TOOLCHAIN

	Old toolchain	New toolchain
Geometry	No single geometry source means we will have to solve this ourselves	DD4hep provides geometries
Tracking	Custom tracking software requires significant workforce, especially as we need very good performance for SoLID	Leverage state-of-the-art with ACTS, multithreading/HPC comes for free
Simulation	Existing gemc implementation of SoLID, collaboration already familiar.	Comes for free with DD4hep, but requires some re-learning
Maintainability	More software maintenance to be done by collaboration, depends on smaller software efforts, will require significant resources.	Depend on large-scale software efforts by LHC community for high-luminosity upgrade, synergies with EIC
Framework	Does not enforce a framework choice	Does not enforce a framework choice, designed to be easy to integrate.
Documentation	Mostly institutional knowledge, explicit support by JLab for some components	Comprehensive documentation, no strong need for external support
Performance	Older programing paradigm, harder to optimize	Off-the-shelf ready for HPC, design for high-luminosity LHC operation perfect for SoLID
Learning curve	Older and more heterogeneous software has steeper learning curve	More integrated toolchain with modern tools easier to learn for new students

STATUS OF NEW SOFTWARE

After 2 months of time

- Software stack in good shape
  - [x] Detector simulation implemented through DD4hep
  - [x] SoLID Geometry and magnetic field integrated through NPDet detector library
  - [x] Digitization implemented
  - (*ongoing*) Tracking with GENFIT, migration to ACTS in progress
  - [ ] Use J/ψ physics goals to benchmark software
  - [ ] Investigate use of ML techniques to accelerate closed-loop optimizations