ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-THPAB217

LIGHTSOURCE UNIFIED MODELING ENVIRONMENT (LUME), A START-TO-END SIMULATION ECOSYSTEM

C. E. Mayes*, P. H. Fuoss, J. R. Garrahan, H. Slepicka, A. Halavanau, J. Krzywinski, A. L. Edelen, F. Ji, W. Lou, N. R. Neveu, SLAC National Accelerator Laboratory, Menlo Park, CA, USA
A. Huebl, R. Lehe, Lawrence Berkeley National Laboratory, Berkeley, CA, USA
L. Gupta, University of Chicago Department of Physics, Chicago, IL, USA
C. M. Gulliford, D. C. Sagan, Cornell University, Ithaca, NY, USA
J. C. E, C. Fortmann-Grote, European XFEL GmbH, Schenefeld, Germany

Abstract

SLAC is developing the Lightsource Unified Modeling Environment (LUME) for efficient modeling of X-ray free electron laser (XFEL) performance. This project takes a holistic approach starting with the simulation of the electron beams, to the production of the photon pulses, to their transport through the optical components of the beamline, to their interaction with the samples and the simulation of the detectors, and finally followed by the analysis of simulated data.

LUME leverages existing, well-established simulation codes, and provides standard interfaces to these codes via open-source Python packages. Data are exchanged in standard formats based on openPMD and its extensions. The platform is built with an open, well-documented architecture so that science groups around the world can contribute specific experimental designs and software modules, advancing both their scientific interests and a broader knowledge of the opportunities provided by the exceptional capabilities of X-ray FELs.

INTRODUCTION

Simulation software for particle accelerators often have highly diverse domains of applicability, levels of development, user bases, and philosophies. Data input and output formats are often non-standard and require experts to interpret. Realistically, no single simulation code is foreseen that can cover the full domain of a facility like the Linac Coherent Lightsource (LCLS) at SLAC, and composite (start-to-end) simulations are rarely performed due to the expertise required.

LUME is an ecosystem of open-source software tools and standards to enable large-scale start-to-end simulations from particle generation through experimental analysis [1]. A generalization of the concepts first introduced in the SimEx platform [2], LUME aims to wrap standard, well-developed electron/photon simulation codes with a common interface with minimal installation complexity, in ways that are ultimately platform-independent. It utilizes the recently developed openPMD standards for data exchange, and is developing standards for describing accelerator and X-ray beamline components in a database. Figure 1 illustrates the concept of a pipeline of simulations we envisage.

* cmayes@stanford.edu

DATA STANDARDS

LUME supports the recently developed openPMD standard [3] for particle-mesh data. The base standard is quite general, and includes a system for consistently describing units in hierarchical data formats. For the specific applications of particle accelerators and FELs, we have developed the extensions to the base standard:

openPMD-beamphysics standard for describing particles and fields commonly encountered in accelerator physics simulations. It includes definitions for spin, electromagnetic fields at particles, photon tracking, external fields (fieldmaps), and relations to lattice elements.

openPMD-wavefront standard for describing complex electric field meshes used in FEL physics applications.

Similar to the base standard, these are written standards that do not specify any particular file format.

SOFTWARE TOOLS

The LUME team supports and develops a number of software tools towards realizing the pipeline show in Fig. 1. For all of these tools, we have chosen Python [4] as the primary interface, and the conda-forge [5] infrastructure as the primary installation mechanism. Individually each is referred to as a *package*, and all provide high level objects (Python classes) that encapsulate the relevant data or simulation setup. Because this development is dynamic, links to all packages are indexed at the LUME website [1].

Particle & Wavefront Analysis

openPMD-beamphysics provides the `ParticleGroup` and `FieldMesh` objects for loading particle and fieldmap data from HDF5 files that obey the openPMD-beamphysics standard, respectively. `ParticleGroup` provides convenient methods for calculating derived quantities, such as angular momentum or normalized coordinates, and also methods for statistical quantities such as 4D emittance and higher order energy spread. It includes methods for converting to and from many commonly used accelerator physics code formats. In addition to conversion and analysis, `FieldMesh` provides simple methods for tracking particles.

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-THPAB217

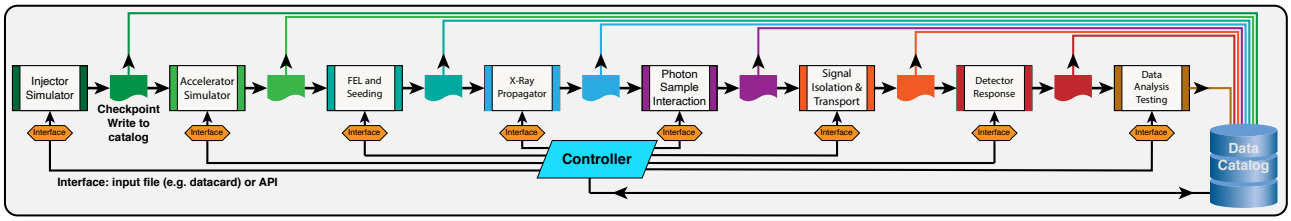


Figure 1: Overview of a general LUME simulation pipeline. The output data (file) of each simulation step is archived in a standard format in a data catalogue, which can be read by the next step. A controller (program or person) interacts with each simulation with a well-defined interface (Python).

openPMD-wavefront provides functions for manipulating HDF5 files that obey the openPMD-wavefront standard. Currently these focus on manipulating the wavefront data from the Genesis 1.3 [6] and SRW [7] simulation codes.

The openPMD-beamphysics package serves as a dependency for distgen, LUME-Astra, LUME-GPT, and LUME-Impact described below.

Particle Generation

distgen provides the `Generator` object for creating particles in 6D phase space according to combinations of probability density functions (PDFs). It includes various 1D and 2D PDFs, including the ability to read them files. Units are strictly handled at the array level. The output is given as a `ParticleGroup` object.

Figure 2 shows the transverse projection of particles created by **distgen**, and also illustrates the built-in plotting from the `ParticleGroup` object.

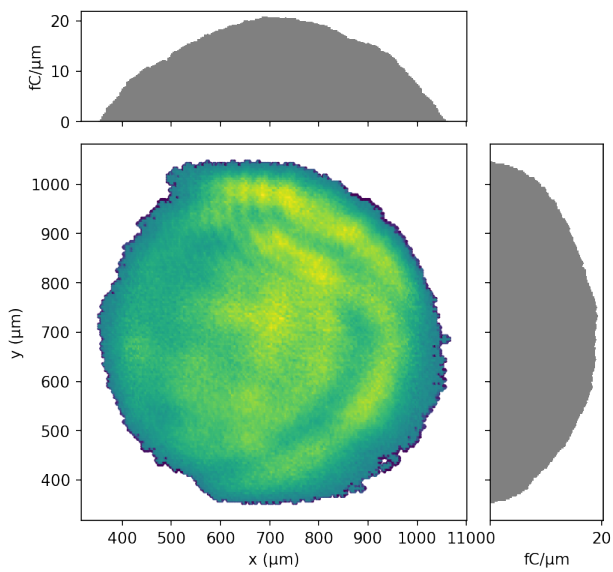


Figure 2: Particles created by **distgen** according to a measured laser profile, plotted using built-in plotting from the **openPMD-beamphysics** package. Labels and units with convenient SI prefixes are automatically chosen.

Injector Simulation

LUME-Astra provides the `Astra` object for encapsulating an Astra [8] simulation, instantiated with a standard Astra input file. The user must provide the Astra executable.

LUME-GPT provides the `GPT` object for encapsulating a GPT [9] simulation, instantiated either from a standard GPT input file, or formed by convenient element objects. The user must provide the appropriate GPT executables and license.

LUME-Impact provides the `Impact` object for encapsulating an Impact-T [10] simulation, instantiated with a standard Impact-T input file. We have worked with the author to provide the Impact-T executables (serial and MPI) via conda-forge.

All of these packages have a very similar interface. The primary objects have methods to initialize them using a standard input file, configure a temporary working space in memory, write input to this space, run the actual simulation executable, and load all output. All input and output are automatically parsed as Python objects, with all particles parsed and stored as `ParticleGroup` objects. Figure 3 illustrates the convenient plotting from each package.

These simulations can be computationally expensive, so all packages have methods to archive the complete input and output of the simulation to HDF5 files. The archive files can be used to instantiate new simulation objects.

Accelerator Simulation

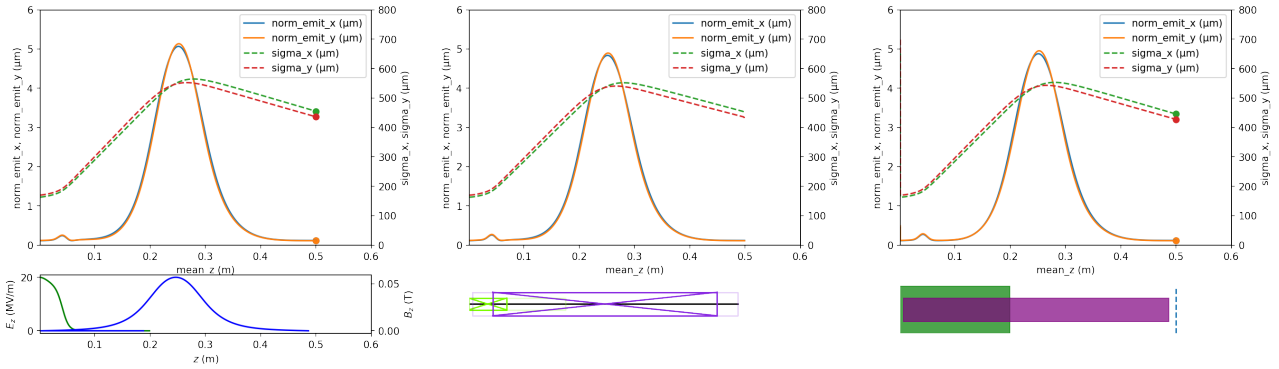
PyTao provides the `Tao` object that drives the Tao program. Tao, based on the Bmad subroutine library [11], has a special syntax to exchange data via strings and direct memory access via a C interface. This depends on a compiled Bmad distribution.

The openPMD-beamphysics standard was developed closely with Bmad, and Bmad is able to read and write HDF5 files that conform to the openPMD-beamphysics standard.

FEL Simulation

LUME-Genesis provides the `Genesis` object for encapsulating a Genesis 1.3 version 2 simulation [6]. The

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-THPAB217



(a) LUME-Astra (b) LUME-GPT (c) LUME-Impact

Figure 3: Simulations of the LCLS-II gun using various LUME Python packages. The plots are created automatically from the `Astra.plot`, `GPT.plot`, and `Impact.plot` methods. All simulations use the same initial particles shown in Fig. 2.

output field data is written directly to an openPMD-wavefront conforming HDF5 file. Partial support for version 4 is included. We received permission of the author to provide maintained version 2 executables (serial and MPI) via conda-forge.

Optimization

Xopt provides an `Xopt` object for performing constrained multi-objective optimization. It can be run in parallel using multiple methods, including MPI and Dask [12]. It standardizes the definition of the variables, objectives, and constraints, and has a compact YAML-based input format that can be used on the command line.

Xopt is a general tool that is compatible with all of the LUME simulation tools via their functional interfaces.

Production Deployment

LUME-Model provides data structures to encapsulate physics simulation and machine learning (ML) models.

LUME-EPICS is a dedicated API for serving and manipulating LUME-Model variables with EPICS [13].

APPLICATIONS

LUME tools are currently used in many projects at SLAC, including LCLS-II-HE SRF injector design, LCLS-II beam shaping [14], hollow beam shaping [15, 16], XLEAP chirp-taper, enhanced self-seeding [17], cavity based XFELs [18], X-ray laser oscillators [19], double-bunch FEL [20], and dual color soft X-ray self-seeding [21]. They have also been used during the new SXR and HXR undulator commissioning.

Figure 4 illustrates how LUME tools are used to perform online (live) modeling of the LCLS injector. The tools are also used for creating large simulation datasets, which are then used to train fast-executing ML-based surrogate models [22, 23].

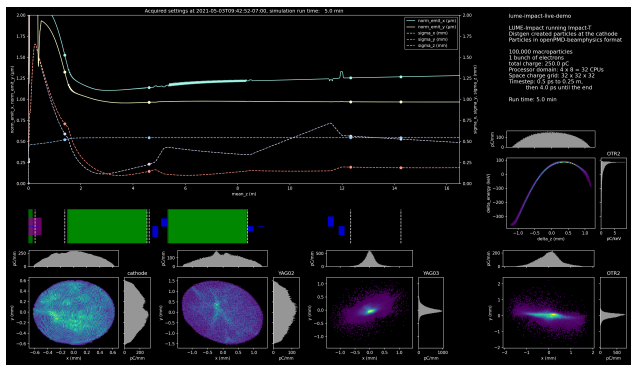


Figure 4: Live model of the LCLS injector using **LUME-Impact**. A neural network-based surrogate model of this simulation using **LUME-Model** also runs live, and serves data over the network using **LUME-EPICS**.

CONCLUSION

The LUME team has developed many software tools and data standards that are currently being used in particle accelerator research and operations. The team collaborates with authors of the underlying software to expand compatibility, as well as with the developers of SimEx and VINYL [24] to extend the breadth of the software covered.

ACKNOWLEDGEMENTS

This work was supported in part by the U.S. Department of Energy Office of Science, Office of Basic Energy Sciences under Contract No. DE-AC02-76SF00515, as well as the PaNOSC project from the European Union's Horizon 2020 research and innovation program (under grant agreement No. 654220). This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy.

REFERENCES

[1] Lightsource Unified Modeling Environment (LUME), `https://www.lume.science`

ISBN: 978-3-95450-214-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2021-THPAB217

[2] C. Fortmann-Grote *et al.*, "SIMEX: Simulation of Experiments at Advanced Light Sources", 2016. arXiv:1610.05980

[3] A. Huebl *et al.*, "openPMD: A meta data standard for particle and mesh based data". doi:10.5281/zenodo.591699

[4] Python, https://www.python.org

[5] Conda-forge, https://conda-forge.org

[6] S. Reiche, Genesis 1.3, https://github.com/svenreiche/.

[7] O. Chubar, Synchrotron Radiation Workshop (SRW), https://github.com/ochubar/SRW

[8] ASTRA - A Space-charge TRacking Algorithm, http://www.desy.de/~mpyflo/.

[9] GPT - General Particle Tracer, Pulsar Physics, http://www.pulsar.nl

[10] J. Qiang, S. Lidia, R. D. Ryne, and C. Limborg-Deprey, "Three-dimensional quasi-static model for high brightness beam Dynamics simulation", *Phys. Rev. ST Accel. Beams*, vol. 9, p. 044204, 2006. doi:10.1103/PhysRevSTAB.9.044204

[11] D. Sagan, "Bmad: A relativistic charged particle simulation library", *Nucl. Instrum. Meth. A*, vol. 558, pp.356–359, 2006. doi:10.1016/j.nima.2005.11.001

[12] Dask: Scalable analytics in Python, https://dask.org

[13] L. R. Dalesio *et al.*, "The experimental physics and industrial control system architecture: past, present, and future", *Nucl. Instrum. Meth. A*, vol. 352, pp. 179-184, 1994. doi:10.1016/0168-9002(94)91493-1

[14] R. Lemons *et al.*, "Dispersion-controlled Temporal Shaping of Picosecond Pulses via Non-colinear Sum Frequency Generation", 2020. arXiv:2012.00957

[15] A. Halavanau, Y. Ding, C. E. Mayes, P. Piot, and S. Baturin, "Hollow Electron Beams in a Photoinjector", in *Proc. of the 11th Int. Particle Accelerator Conf. (IPAC'20)*, CAEN, France, May 2020, paper WEVIR06, pp. 49-52.

[16] A. Halavanau, S. J. Gessner, C. E. Mayes, and J. B. Rosenzweig, "Hollow and flat electron beam generation at FACET-II", presented at the 12th Int. Particle Accelerator Conf. (IPAC'21), Campinas, Brazil, May 2021, paper MOPAB101, this conference.

[17] E. Hemsing, A. Halavanau, and Z. Zhang, "Enhanced Self-Seeding with Ultrashort Electron Beams", *Phys. Rev. Lett.*, vol. 125, p. 044801, 2020. doi:10.1103/PhysRevLett.125.044801

[18] G. Marcus *et al.*, "Refractive Guide Switching a Regenerative Amplifier Free-Electron Laser for High Peak and Average Power Hard X Rays", *Phys. Rev. Lett.*, vol. 125, p. 254801, 2020. doi:10.1103/PhysRevLett.125.254801

[19] A. Halavanau *et al.*, "Population inversion X-ray laser oscillator", *PNAS*, vol. 117, pp. 15511-15516, Jul. 2020. doi:10.1073/pnas.2005360117

[20] A. Halavanau, F-J. Decker, C. Emma, J. Sheppard, and C. Pellegrini, "Very high brightness and power LCLS-II hard X-ray pulses", *J. Synchrotron Rad.*, vol. 26, pp. 635–646, Apr. 2019. doi:10.1107/S1600577519002492

[21] A. Halavanau, D. Cocco, E. Hemsing, G. Marcus, D. S. Morton, and G. R. Wilcox, "Dual Color Self-Seeding at LCLS-II Soft X-Ray Undulator Line", presented at the 12th Int. Particle Accelerator Conf. (IPAC'21), Campinas, Brazil, May 2021, paper MOPAB097.

[22] A. Edelen, N. Neveu, C. Emma, D. Ratner, and C. Mayes, "Machine learning models for optimization and control of x-ray free electron lasers", in *Proc. NeurIPS'19*, Vancouver, Canada, pp. 1-5.

[23] A. Edelen, N. Neveu, M. Frey, Y. Huber, C. Mayes, and A. Adelmann, "Machine learning for orders of magnitude speedup in multiobjective optimization of particle accelerator systems", *Phys. Rev. Accel. Beams*, vol. 23, p. 044601, 2020. doi:10.1103/PhysRevAccelBeams.23.044601

[24] J. C. E *et al.*, "VINYL: The VIrtual Neutron and x-raY Laboratory and its applications", in *Proc. of Advances in Computational Methods for X-Ray Optics V*, California, USA, Aug.-Sep. 2020, pp. 114930Z-1-114930Z-11. doi:10.1117/12.2570378