ISBN: 978-3-95450-220-2 ISSN: 2673-5474 doi: 10.18429/JACoW-FEL2022-MOP06

BRINGING GENESIS TO THE CLOUD WITH SIREPO

C. C. Hall, P. Moeller, J. P. Edelen, E. Carlin, M. Keilman,
G. Khalsa, R. Naglar, R. O'Rouke, RadiaSoft LLC, CO, USA

Abstract

Genesis is widely used in the free electron laser community as a simulation tool for studying both simple and complex FEL systems. Until now, this has necessitated learning the command line interface, which can be challenging for new users. Sirepo Genesis provides an intuitive graphic interface for building Genesis simulations in the browser that can then be run using our cloud computing services. Our interface also provides the ability to export simulations for command line use, simulation post-processing with publication quality graphics and the power to share their results with the click of a button to anyone, anywhere, in the world. This poster describes our new GUI and highlights the notable features that have been developed.

INTRODUCTION

RadiaSoft has built a computer aided engineering (CAE) gateway called Sirepo [1], using our open source cloud computing framework of the same name [2–4]. Sirepo supports over 1,500 (and growing) free users, of whom approximately 200 are active in a given month. Sirepo is also available through a series of subscription programs. Sirepo Premium provides users enhanced support, greater access to compute cores, and scientific consulting from RadiaSoft scientists and engineers. The US Particle Accelerator School[1] is our first Sirepo Education customer and the NSLS-II is our first Sirepo Private customer. RadiaSoft supports a few university programs and workshops each year as a service to the community. Recent beneficiaries include Stanford University and the SAGE-S 2021 Summer Camp[2]

Sirepo is designed to bring scientific, engineering and educational softwares to the cloud, with a rich browser-based interface that works in any modern browser on any computing device. The number of supported codes is continuing to grow, with more features for code coupling and benchmarking. For particle accelerator modeling, we support: MAD-X, elegant, OPAL, Synergia, Zgoubi, Warp PBA (plasma-based accelerators with a cylindrical mesh and azimuthal modes), and JSPEC (electron cooling and intra-beam scattering in rings). For X-ray beamlines, synchrotron radiation and X-ray optics, we support: Shadow for ray tracing and the Synchrotron Radiation Workshop (SRW) for full physical optics. Other interesting apps include: Warp VND (using the Warp code as a 2D and 3D electrostatic PIC engine) to simulate thermionic converters and other vacuum nanoelectronic devices, and the Radia code for magnet design.

More recently, Sirepo is supporting our own app, called Activait,for the use of machine learning techniques to analyze data. The newest app, called Controls, enables users to explore control system algorithms, using MAD-X as a virtual particle accelerator.

Some codes Sirepo supports are proprietary, e.g. FLASH for hydrodynamic and MHD plasma simulations. Proprietary codes are enabled through Sirepo's role-based authorization system. Authentication of users is secured through password-less logins via email using short-lived one-time tokens. For some codes, Sirepo also gives users the option to launch simulations on the Cori supercomputer at NERSC, if they have access to a NERSC repository. Sirepo supports the two-factor authentication mandated by NERSC.

Sirepo provides single sign-on access to an integrated Jupyter[3] server, enabling interactive cloud computing with Python and other languages. All of the Sirepo-supported codes and associated dependencies are pre-installed, together with standard machine learning tools and libraries. Our Jupyter configuration supports a variety of graphics and numeric libraries including Matplotlib, Pandas, Plotly, Seaborn, and yt.

GENESIS IN SIREPO

We have recently added Genesis to the family of codes supported in Sirepo. Due to the uniqueness of Genesis the code the interface does not follow the same convention as many of our other particle accelerator design tools. When users enter the Sirepo Genesis app they have access to a sandbox with an array of examples from notable free electron lasers. Figure 1 shows a screenshot of the examples folder in the Genesis application. Note that users can start a simulation from scratch, import an existing simulation, or build off of one of the provided examples.

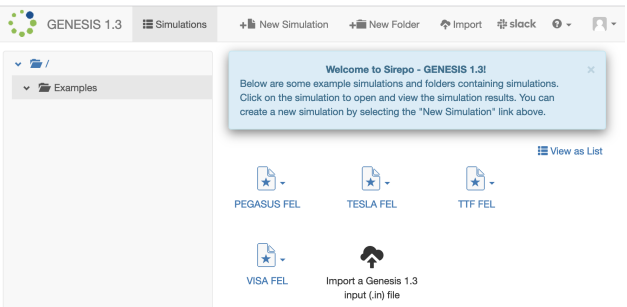

Figure 1: Screenshot of the examples folder for Sirepo Genesis.

The app allows users to configure and execute time-independent simulations. Users can adjust all parameters specified in the Genesis input file related to time-independent simulations. The full set of Genesis inputs are

[1] https://uspas.fnal.gov/
[2] https://conf.slac.stanford.edu/sage/
[3] https://jupyter.org

ISBN: 978-3-95450-220-2 ISSN: 2673-5474 doi: 10.18429/JACoW-FEL2022-MOP06

available to the user. They are broken down based on which aspect of the simulation is being modified, (e.g. the electron beam, the undulator, the simulation parameters such as the resolution, the diagnostic output, etc.). Figure 2 shows a screenshot of the simulation configuration tab. Note that when a user enters the source tab they are only presented with parameters that are directly manipulated as opposed to computed parameters. Computed parameters can be viewed on subsequent pages.

Once the simulation is set up, users are able to execute the simulation and visualize the results in the Visualization tab. The outputs are broken into three reports: Parameter, Particle Distribution, and Field Distribution, Fig. 3. The Parameter report provides information about, for example, the laser power, electron beam bunching, rms parameters, average beam energy, and other aggregate parameters produced by `Genesis`. The Particle Distribution and Field Distribution reports show more fine grain details about the electron beam phase space, as well as the radiation field intensity and phase. Users are able to toggle the output parameters and the appearance of the plots, as well as download PNGs of the reports or the raw data used to produce them if further post-processing is required. We show the beam distribution, the photon distribution, and the beam optics functions as they evolve through the undulator. Figure 3 shows a screenshot of this interface.

FUTURE WORK

We have plans to extend our interface capabilities which will allow users to build undulaors using the familiar drag-and-drop model. We will develop the ability to specify tapering period-by-period, or by using a set of mathematical functions that will automatically compute the tapering from a user-specified function. Users will be able to specify the initial distribution in a Source tab using either the output of a simulation, an external file, or by specifying the relevant bunch parameters such as an initial bunching factor and energy modulation. Finally, we will implement plots of relevant information such as the radiation power, bunching factor, and electron beam phase space, among the other relevant FEL diagnostics, through the `Sirepo` Visualization tab.

We will allow users to specify initial radiation by uploading a supported file format, or by exporting radiation field information from the `SHADOW` and `SRW` simulations. We will also make the exported radiation field from `Genesis` readable by the synchrotron radiation codes for full inter-operability. This will require some extensions to the `Switchyard` class we developed above for exchanging particle data between tracking codes. In the end, we will be able to read in initial particle distributions from any of the accelerator tracking codes, export that particle distribution to a tracking code for modeling the longitudinal phase space diagnostic, and read in radiation fields from a synchrotron radiation code and read out the fields from `Genesis` into those codes.

We also have plans to extend the `Sirepo` framework to handle building multi-code end-to-end simulations of a free electron laser. This includes the photoinjector, LINAC sections, beamline sections, undulators, and diagnostics. The basic back-end code for handing off simulations and controlling the flow in Python exists and is utilized within the RSOpt library. We plan to integrate these features fully into `Sirepo` providing a more integrated development environment for end-to-end modeling.

REFERENCES

[1] "The Sirepo scientific gateway." (2021), `https://sirepo.com`

[2] D. L. Bruhwiler *et al.*, "Knowledge Exchange Within the Particle Accelerator Community via Cloud Computing," in *Proc. IPAC'19*, Melbourne, Australia, May 2019, pp. 3548–3551, `doi:10.18429/JACoW-IPAC2019-THPMP046`

[3] M. S. Rakitin *et al.*, "Sirepo: An open-source cloud-based software interface for x-ray source and optics simulations," *Journal of Synchrotron Radiation*, vol. 25, no. 6, pp. 1877–1892, 2018, `doi:10.1107/S1600577518010986`

[4] "The Sirepo framework for scientific cloud computing." (2021), `https://github.com/radiasoft/sirepo`

ISBN: 978-3-95450-220-2 ISSN: 2673-5474 doi: 10.18429/JACoW-FEL2022-MOP06

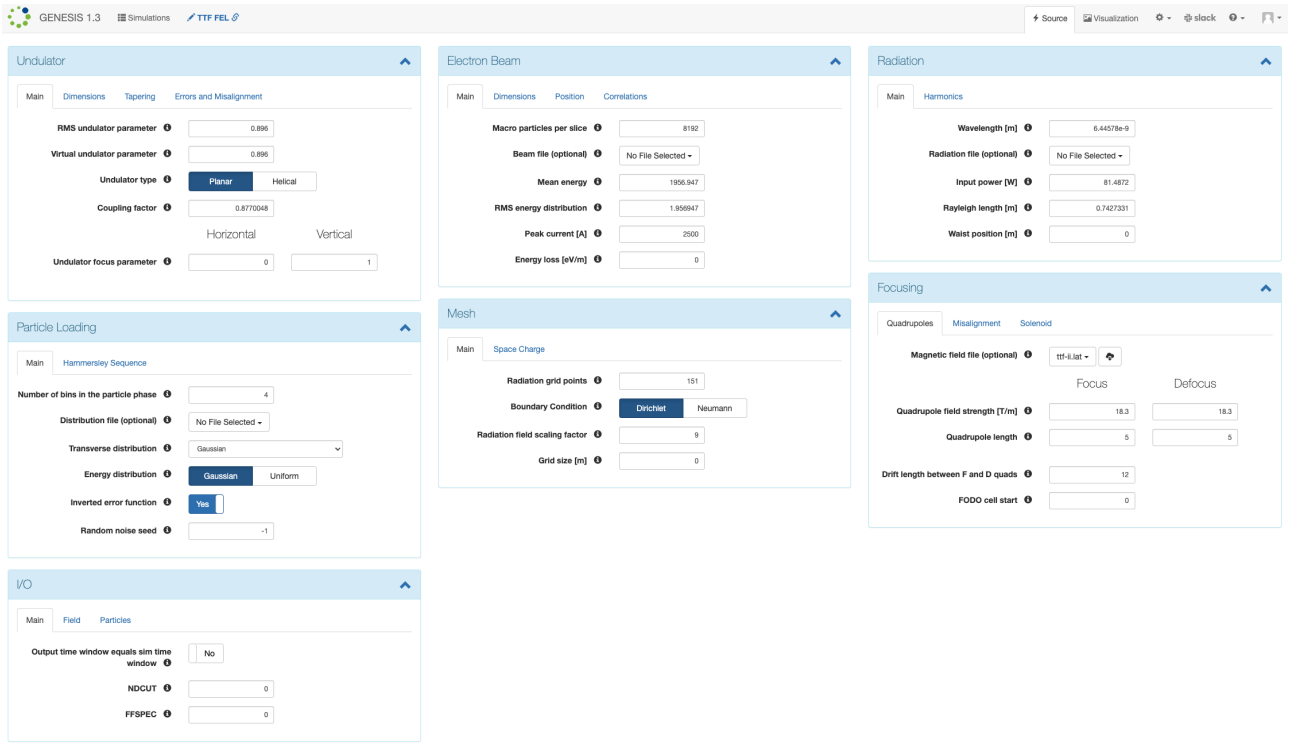

Figure 2: Screenshot of the input tab for building a `Genesis` simulation in `Sirepo`. Note, the user configuration tab is more reminiscent of our JSPEIC tools.

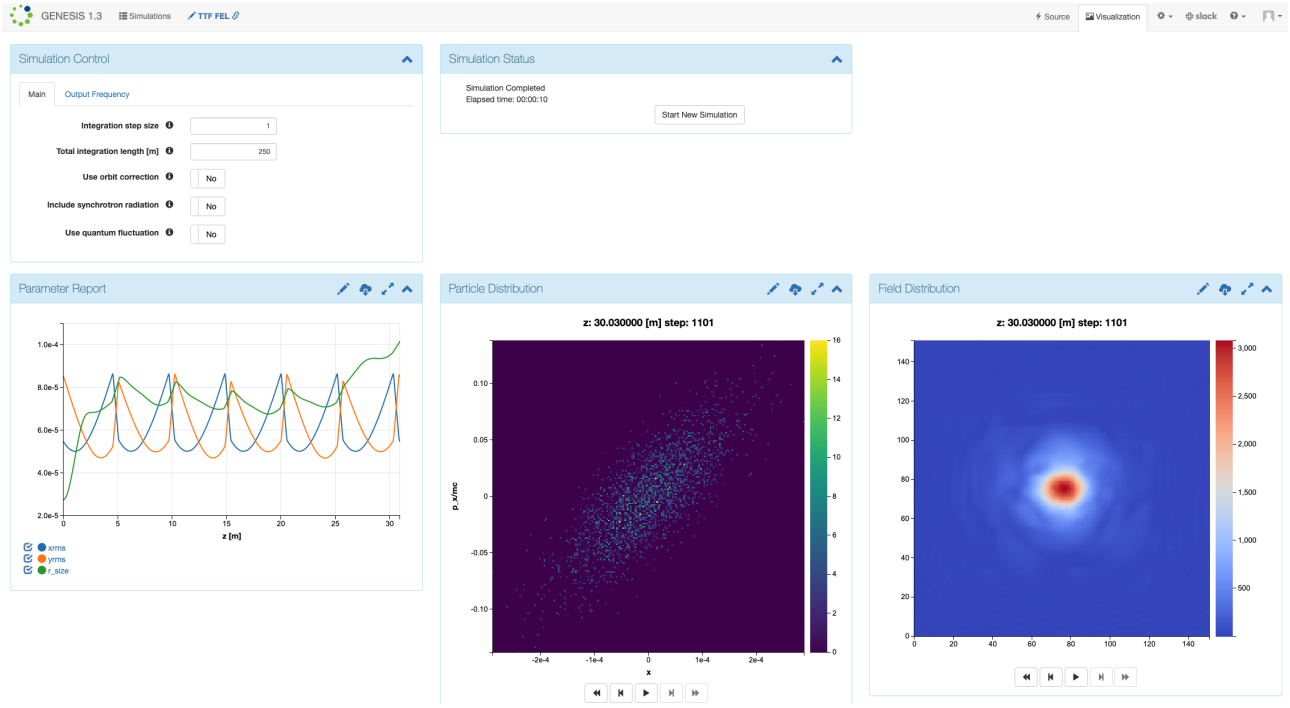

Figure 3: Visualizations of time-independent `Genesis` output.