ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-WEPR58

STATUS OF THE TIME-DEPENDENT FEL CODE GENESIS 1.3

S. Reiche*, Paul Scherrer Institute, Villigen, Switzerland
C. Lechner, European XFEL, Schenefeld, Germany

Abstract

Version 4 of the widely used time-dependent FEL code Genesis 1.3 has been released. The C++ code keeps the entire bunch in memory and thus allows for self-consistent effects such as wakefields or long-range space charge fields. With sufficiently allocated distributed memory, Genesis 1.3 can represent each individual electron. This solves the problem of the shot noise statistics at any arbitrary frequency in the simplest way and allows for sorting and redistribution of particles among the computer cores for advanced FEL applications such as the Echo-Enabled Harmonic Generation schemes. This presentation reports on the new physics added to the code as well as features which simplify the setup of the simulations as well and the ability to link user-made libraries to adapt to the specific needs of each user.

INTRODUCTION

The Free-electron Laser (FEL) is coupled system between relativistic electrons and a co-propagating radiation field, yielding an exponential amplification of the radiation field. Due to the large number of the degrees of freedom, numerical simulations are a useful approach to estimate the performances of the FELs. Genesis 1.3 is a code [1, 2], which can resolve the temporal dependence of the electron bunch and the radiation field as they co-propagate along the periodic magnetic field of an undulator. At the time of the first version computer resources were very limited, which was reflected by the core algorithm of Genesis 1.3. However, with the rapid progress in high performance computing, new possibilities arise. This has been addressed with the latest version of the code, presented in this proceeding.

CHANGES TO THE CODE STRUCTURE

For the current version 4 the code Genesis 1.3 uses C++ and three libraries: the message passing interface (MPI) [3], HDF5 [4] as the native file format for binary data, and the FFTW3 [5, 6] library for efficient calculation of the Fourier transformation. The last library is currently optional but might become mandatory in the future. The compilation is facilitated with CMake, which configures the Makefile for compilation and adds the information on the version, compilation date and the used source code version (git commit ID).

The structure of the code differs significantly from that of the previous version. Only some of the core algorithm (Runge-Kutta solver, ADI solver for the radiation field) has been translated directly to C++. In general, the particle and field distribution – the core objects for the FEL simulation – are represented by classes, while the beam and field solvers are member functions of the corresponding classes and as such adapted to various instances. Examples are different grid sizes for the fundamental and third harmonic of the radiation field.

Sequentially Driven Actions by Namelists

Execution of Genesis is no longer controlled by a single, large namelist. Instead the input deck contains a series of smaller, more specialized namelists. Genesis executes these namelists when they occur. Therefore it depends on the order and the beam or field must be defined with the `beam` and `field` namelist before they are both propagated through the undulator with the `track` namelist. Thus output such as the particle dump can be issued before and after the tracking. In fact, the tracking command can be evoked several times with some processing of the beam or field distribution. Examples would be HGHG [7] simulations with the harmonic conversion in a single run.

Lattice Syntax

The new version of Genesis supports variable integration step sizes but aims to provide a step size close to the value specified in the input deck. Therefore the lattice is no longer restricted to have the integration stepsize as a common divider to all elements lengths. This allows for a more flexible lattice design and we follow a convention similar to Elegant [8, 9]. In general it has two types of inputs. The first is a basic beamline element definition, such as an undulator or a quadrupole magnet. All these definitions must have a label, with which they can be referred to. The second type are definitions of beamline sections, which are a sequence of basic elements or other beamlines. At least one beamline has to be specified in the main input deck through which the electrons are tracked with the `track` namelist. Several variations of a beamline (e.g. the full beamline, a subsection of it, or the beamline without focusing quadrupole) can be kept in the same lattice file and only the main input file needs to be edited to switch among them. This applies also for any staged simulation such as HGHG or EEHG. The setup of a beamline supports the multiplication of elements (e.g. repeating a basic short layout) and an absolute placement of elements (e.g. a quadrupole lattice superimposed onto a continuous undulator). The `marker` is a beamline element, new in the current verson of Genesis, allowing certain action at specific location. Currently the dump of the radiation field or particle distribution, as well as sorting and an early termination of the tracking are supported.

All Particles in Memory

The largest innovation of Genesis 1.3, Version 4 is the ability to keep all particles in memory at the same time. In

* sven.reiche@psi.ch

ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-WEPR58

version 2.0 or before, Genesis would progress sequentially through the bunch from the back to the front. Since the memory demand of a full beam, resolving all electrons, is in the order of 100 GByte, a distributed memory among multiple nodes is needed. This change allows for two majors improvements in the calculation: First, collective effects such as long range space charge or wakefields can be calculated during the run, based on the current profile. They do not need to be provided externally by importing the effective potential. Second, the migration of particles, resolving all individual electrons, between adjacent longitudinal slices is supported by sorting the particles longitudinally. Since in the majority of simulations, the particles only migrate to adjacent slices, bubble sort provides a very efficient algorithm with a simple MPI implementation. Sorting and redistribution of the particles allows schemes such as EEHG to be realized in a single run. Since all electrons are resolved the redistribution among the longitudinal slices still provides the correct shot noise, a problem the older version had problems with for large harmonic conversions.

Diagnostic Plugins

Genesis 1.3 provides a complete set of electron beam and light field diagnostics that are sufficient for most applications. However, the study of advanced FEL schemes calls for special diagnostics capturing the evolution of the relevant parameters. One solution to address these special diagnostic requirements would be to regularly dump the electron beam and light fields to HDF5 files and then obtain the desired information using postprocessing software, a procedure that can be quite time-consuming.

Recently, Genesis 1.3 was extended with diagnostics capability based on plugin modules in shared libraries that can be configured and loaded at runtime (using the new namelists `add_plugin_beamdiag` and `add_plugin_fielddiag`). While this capability was initially devised for analysis of light fields in simulations for the planned upgrade of European XFEL with a superconducting undulator afterburner aiming at extending the photon energy range beyond the nominal 25 keV [10, 11], there exist numerous applications such as FELs generating light with orbital angular momentum (OAM) [12, 13].

Every time diagnostic information is collected, the currently loaded plugin modules (for light fields and electron beam) are called, with one call for every harmonic (light fields) and with the data in every slice. This corresponds to the information in a complete field/beam dump, with the key advantage being that the data is analyzed locally on the CPU cores having the data in RAM as they are also performing the FEL simulation. Consequently, the additional time required for performing the diagnostic task is typically moderate. The results computed in the plugin modules are at the end of the tracking process stored at a configurable location in the output file.

The user-provided, custom diagnostics needs to be implemented in C++. The result of the build process is a shared library (in an `.so` file), that is loaded at run time into Genesis 1.3. Several demonstration examples are available in the source code repository [2] that can be used as starting point for implementations by users. In addition, a stand-alone test environment is now available, reducing the complexity of many software development and debugging tasks since neither a complete Genesis 1.3 simulation run nor debugging in the MPI environment are needed.

Performance Benchmarking

The performance of Genesis 1.3 for different clusters and varying number of cores has been tested for the benchmark case of SwissFEL beamline Aramis, lasing at 1 Angstrom. The studies uses three clusters: Euler [14], hosted by the ETHZ, and Merlin5 and 6 [15], both hosted at the Paul Scherrer Institute. The figure of merit is the total amount of CPU time. The ideal behaviour corresponds to no penalty for intra-process communication of the MPI protocol. The results are shown in Fig. 1.

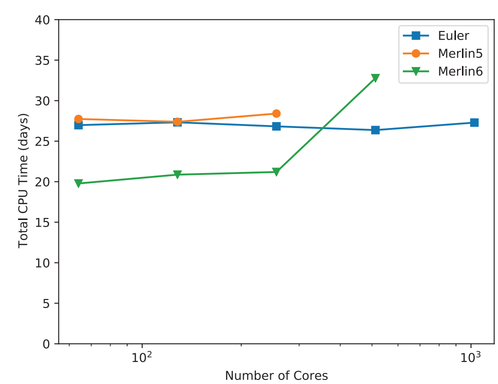

Figure 1: Total CPU time at different clusters for various numbers of cores.

The performance shows in general an excellent scaling of Genesis, with nearly a constant total CPU time. Thus the wall clock time for these simulations scales almost inversely with the number of requested cores. Only the last case for the Merlin 6 cluster shows an increase. This is also the only case where the hyperthread function of the Merlin6 nodes was used to increase the number of available cores.

The presented times are only measured once the batch system has started the Genesis simulation. It excludes the waiting time in the queue of the batch system since this depends strongly on the allocated priority of the simulation job and the scheduling policy of the batch system. In the case of the Merlin cluster, the batch waiting time scales disproportionally with the number of requested cores, so that smaller jobs are more favorable. Submitting two jobs with 32 cores in parallel became more efficient that submitting them sequentially with 64 cores requested.

ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-WEPR58

EXAMPLES

Echo-Enabled Harmonic Generation

The Echo-enabled Harmonic Generation EEHG [16] transfers the coherence properties of an external laser to that of the FEL output, involving a very high harmonic conversion as well as a significant redistribution of electrons over multiple wavelengths of the seed laser. Previous approaches of harmonic conversions, by simply scaling the longitudinal phase, have the drawback that it reduces the sample rate by the harmonic number (e.g. for $h = 100$ the beam is sampled every hth wavelength). This restricts the resolved bandwidth of the simulation so that even slight various in the electron beam energies would push the resonant FEL wavelength outside of this band. Second, the reduced sample rate becomes comparable to the FEL cooperation length and thus violating the requirement of a sufficiently high sample rate, causing systematic errors in the numerical calculation.

For reproducing the results of the EEHG experiment [17], the new feature of Genesis has been used to sort and redistribute the electrons after the two stages of the EEHG. Since all electrons have been resolved, any redistribution of the macro particles will still provide the correct shot noise statistic [18] on a first principle basis.

Enhanced SASE

Enhanced SASE [19] is a process where an electron beam is modulated by the interaction of an external laser and then compressed to a series of current spikes with a small chicane in front of the main undulator beam line. However these current spikes generate a strong space charge field [20], which can alter the performance of the FEL, since many electrons are shifted in energy during the FEL process.

Genesis allows the self-consistent calculation of the space charge field over the entire extension of the modelled electron beam. Since the space charge field generates an effect on the tail of the electron bunch, arising from the distribution in the front, this feature was not possible in the previous version due to the sequential progression of the calculation through the bunch. With all particles in memory this limitation is no longer given and the space charge field can be calculated even after some beamline components which alter the current profile (namely the chicanes of the ESASE process).

Harmonic Lasing

Harmonic lasing [21] have been proposed to extend the wavelength range of an FEL towards higher photon energies. The challenge is to suppress the fundamental wavelength while keeping the amplification on the higher harmonics preserved. Besides the approach of tuning phase shifters to non-integer phase shifts of the fundamental wavelength [22, 23] another practical approach is to alternate between various fundamental wavelengths and harmonics, e.g. undulator tuned to either 5 and 3 nm with a continuous amplification at 1 nm (5th and 3rd harmonic, respectively). This switching between the two fundamentals while using only one electron distribution is challenging. The latest version of Genesis allows for harmonic and subharmonic conversion during the run. To model the case, given above, the generation would be setup at 5 nm and the harmonic radiation field at 1 nm. After the first stage the electron distribution is converted to the 5th harmonic and then to the subharmonic of 3 nm. It requires a reslicing and sorting of the particle distribution, which can be done if all electrons are resolved in the simulation. The radiation field at 1 nm is kept during all the stages while the fundamental swaps between 3 and 5 nm.

CONCLUSION

Version 4 of Genesis 1.3 has been released which supports more complex problems to be modelled, in particular multi-staged FEL such as HGHG or EEHG. The code keeps the entire particle distribution in memory for a self-consistent calculation of space charge and wakefields. In particular, each electron can be resolved which preserves the correct shotnoise statistic at all wavelengths. This eliminates the constraints of quiet loading algorithm, allowing for advanced manipulation of the longitudinal phase such as Echo-enabled Harmonic Generation and a succeeding sorting of the particle distribution into the corresponding radiation grid.

ACKNOWLEDGEMENTS

The numerical simulations were supported by the "MAXWELL" computational resources operated at Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany.

REFERENCES

[1] S. Reiche, "GENESIS 1.3: a fully 3D time-dependent FEL simulation code,"*Nucl. Instrum. Methods Phys. Res., Sect. A*, vol. 429, no. 1–3, pp. 243–248, Jun. 1999. doi:10.1016/s0168-9002(99)00114-x

[2] Source Code of "Genesis 1.3", version 4: https://github.com/svenreiche/Genesis-1.3-Version4 (last access: May 14, 2024)

[3] Website of MPI Forum, https://www.mpi-forum.org/ (last access: May 3, 2024)

[4] HDF5 website, https://www.hdfgroup.org/solutions/hdf5/ (last access: May 3, 2024)

[5] M. Frigo and S. G. Johnson, "The Design and Implementation of FFTW3,"Proceedings of the IEEE, vol. 93, no. 2, pp. 216–231, Feb. 2005. doi:10.1109/jproc.2004.840301

[6] FFTW website, https://www.fftw.org/ (last access: May 3, 2024)

[7] L. H. Yu, "Generation of intense uv radiation by subharmonically seeded single-pass free-electron lasers,"*Phys. Rev. A*, vol. 44, no. 8, pp. 5178–5193, Oct. 1991. doi:10.1103/physreva.44.5178

[8] M. Borland, "elegant: A Flexible SDDS-Compliant Code for Accelerator Simulation,"ANL, IL, USA, Rep. Advanced Photon Source LS-287, Aug. 2000.

[9] Elegant website, https://www.aps.anl.gov/Accelerator-Operations-Physics/Software (last access: May 3, 2024)

[10] S. Casalbuoni *et al.*, "Superconducting undulator activities at the European X-ray Free-Electron Laser Facility,"*Front. Phys.*, vol. 11, Jun. 2023. doi:10.3389/fphy.2023.1204073

[11] C. Lechner, S. Casalbuoni, G. Geloni, E. Schneidmiller, S. Serkez, and H. Sinn, "Numerical simulation studies of superconducting afterburner operation at SASE2 beamline of European XFEL,"*X-Ray Free-Electron Lasers: Advances in Source Development and Instrumentation VI*, Jun. 2023. doi:10.1117/12.2669177.

[12] E. Hemsing, A. Marinelli, and J. B. Rosenzweig, "Generating Optical Orbital Angular Momentum in a High-Gain Free-Electron Laser at the First Harmonic,"*Phys. Rev. Lett.*, vol. 106, no. 16, Apr. 2011. doi:10.1103/physrevlett.106.164803

[13] J. Yan and G. Geloni, "Self-seeded free-electron lasers with orbital angular momentum,"*Advanced Photonics Nexus*, vol. 2, no. 3, p. 036001, Mar. 2023. doi:10.1117/1.apn.2.3.036001

[14] Euler-Cluster website, https://scicomp.ethz.ch/wiki/Euler

[15] Merlin-Cluster website, https://www.psi.ch/en/awi/the-merlin-hpc-cluster

[16] D. Xiang and G. Stupakov, "Echo-enabled harmonic generation free electron laser,"*Phys. Rev. ST Accel. Beams*, vol. 12, no. 3, p.030702, Mar. 2009. doi:10.1103/physrevstab.12.030702

[17] P. Rebernik Ribič *et al.*, "Coherent soft X-ray pulses from an echo-enabled harmonic generation free-electron laser,"*Nat. Photon.*, vol. 13, pp. 555–561, 2019. doi:10.1038/s41566-019-0427-1

[18] W. M. Fawley, "Algorithm for loading shot noise microbunching in multidimensional, free-electron laser simulation codes,"*Phys. Rev. ST Accel. Beams*, vol. 5, p. 070701, 2002. doi:10.1103/PhysRevSTAB.5.070701

[19] A. A. Zholents, "Method of an enhanced self-amplified spontaneous emission for x-ray free electron lasers,"*Phys. Rev. ST Accel. Beams*, vol. 8, p. 040701, 2005. doi:10.1103/PhysRevSTAB.8.040701

[20] G. Geloni, E. Saldin, E. Schneidmiller, and M. Yurkov, "Longitudinal Impedance and Wake from XFEL Undulators. Impact on Current-Enhanced SASE Schemes,"*Nucl. Instr. Meth.* A, vol. 583, pp. 228–247, 2007. doi:10.1016/j.nima.2007.09.019

[21] B. W. J. McNeil, G. R. M. Robb, M. W. Poole, and N. R. Thompson, "Harmonic Lasing in a Free-Electron-Laser Amplifier,"*Phys. Rev. Lett.*, vol. 96, p. 084801, 2006. doi:10.1103/PhysRevLett.96.084801

[22] E.A. Schneidmiller and M.V. Yurkov, "Harmonic lasing in x-ray free electron lasers,"*Phys. Rev. ST Accel. Beams*, vol. 15, p. 080702, 2012. doi:10.1103/PhysRevSTAB.15.080702

[23] G. Penn, "Simple method to suppress the fundamental in a harmonic free electron laser,"*Phys. Rev. ST Accel. Beams*, vol. 18, p. 060703, 2015. doi:10.1103/PhysRevSTAB.18.060703