UPDATE ON THE FEL CODE GENESIS 1.3

S. Reiche, Paul Scherrer Institut, Villigen, Switzerland

Abstract

The widely used time-dependent code Genesis 1.3 has been modified to address new needs of users worldwide. The existing limitation of tracking isolated slices of the FEL beam has been overcome by keeping the entire electron beam in memory, which is tracked as a whole through the undulator. This modification allows for additional features such as allowing particles to migrate into other slices or applying self-consistent wakefield and space charge models.

INTRODUCTION

Since the first high gain Free-Electron Lasers [1–4] numerical codes have aided the users to understand experimental results and to design future facilities. Several codes are available [5–9] and have been benchmarked against each other and experiments [10]. With the ongoing development of the general computer technology and infrastructures the FEL codes can expand the complexity of the underlying models or operate with more particles for finer resolutions. Currently single-pass FELs at the Angstrom level can be run in a few hours or less.

However new FEL schemes, e.g. self-seeding [11], EEHG [12], high-brightness SASE FELs [13], are rather complex and very difficult to model for users of the FEL codes. The primary obstacle is that most codes have evolved from a single processor platform to a large scale parallel computer platform while preserving the ability to run on a single computer. Nevertheless a new algorithm, where a parallel computer network is deeply embedded, can offer new features beyond the capabilities of existing codes.

In this paper I present the current status of the code Genesis 1.3 [6], which has been modified under the assumption that the computer cluster is large enough to hold the entire electron beam and radiation field in memory. The beam and field is propagated through the undulator as a whole with a resolution down to each individual electron. The core algorithm is still based on the slowly varying envelope approximation (SVEA) [14], where the equations of motion are averaged over one undulator period. It allows one to choose integration step sizes larger than the undulator period to keep the number of integration steps within a reasonable limit even for very long hard-X-ray FELs such as LCLS or SwissFEL. A non-averaged approach is not pursued, but which has been successfully implemented by new codes such as PUFFIN [8].

CURRENT LIMITATION

During the development of Genesis in the late nineties one important factor was the available memory for the calculation. Keeping the entire radiation field and particle distribution was way beyond the practical limit of those days and an extensive bookkeeping has to be done to reduce the required footprint in memory space. This was done by tracking a single electron slice through the undulator interacting with many radiation field slices which are slipping in from behind and then slipping out after a few integration steps, depending on the length of the electron slice. The bookkeeping is storing temporarily the field which slips out to feed it to the next electron slice once the tracking of the current slice has been done. Using this approach the memory needs to store only the data of a single electron slice and the radiation field over one slippage length compared to the entire time window which can be many times longer than a slippage length. However this restricts the algorithm to work sequentially through the electron bunch from the tail to the head. No information can propagate in the backward direction.

Recent ideas to improve longitudinal coherence in SASE FELs [13, 15, 16] are based on an enhancement of slippage to cover the entire bunch. That way the spectral brightness is improved. The consequence is that the record for storing the slippage field needs to be increased by a large factor. In fact, it would use the same memory size if the entire radiation field were kept in memory at all time. The latter approach has the advantage that one could calculate the spectrum during runtime and not, as it is now, as a post-processing step.

A second limitation arises from proposed schemes which are utilizing a large harmonic conversion, either by a multi-staged approach in HGHG cascades or a direct conversion with high efficiency in EEHG schemes [17]. Genesis particle distribution is based on a quiet loading where macro particle are mirrored and evenly distributed in longitudinal position to cancel out completely any Fourier component for a given wavelength. In an explicit step in the beam loading algorithm a controlled random offset is applied to the particle to give the correct statistics in the bunching factor [18]. To include more harmonics more mirror particles are needed, preferably at least twice the number than the highest harmonic considered. For the 100th harmonic this would be at least 200 mirror particles. If one used 1000 particle to generate the remaining 5 D distribution and then apply the mirroring process one slice would be filled with 200k macro particles or if sliced to the final harmonics 2000 particles. For an FEL operating at 1 nm with a 1 kA beam current that is almost of the same order as the real number of electrons to be modeled. With a moderate increase in the particle number then a real one-one simulation could be carried out with the advantage that no mirroring needs to be done and therefore the transverse distribution is much smoother (effectively filled with 200k particles rather than only 1k).

ISBN 978-3-95450-133-5

Last, Genesis is limited by restricting particles into a single slice and keep them over the entire undulator length. For SASE FELs and an electron beam with almost no energy spread it is a valid assumption but breaks down for over-compression schemes such as self-seeding or EEHG, where particle are moved into different slices even when they are initially grouped together with a distance shorter than the radiation wavelength. To move particles into a different slice would mean that a particle is taken out of the beamlet which is formed by the mirroring process to model the correct shot noise. To illustrate this a minimal beamlet has two particles, separated by half a radiation wavelength, and is split apart and moved into different slices. Prior to this step the phases of both particles are almost canceling and the overall emission level is low. After the splitting each particle is in a different slice and has no corresponding particle to cancel the bunching factor when evaluating the source term for the Maxwell equation. The emission level would be effectively enhanced and determined by the number of macro particles in the simulation and not by the number of electrons to be modeled.

CORE ALGORITHM

The fundamental change in the upcoming version of Genesis is the exchange of the loop order in the code. In the new version the inner loop cycles through all slices to advance them by one integration step along the undulator. This is repeated till the end of the undulator is reached. No record of a slippage field is needed because the entire radiation field is kept in memory at any time.

The code supports the distribution of the calculation over many computer nodes in a computer cluster, following the MPI standard of inter-node communication. Unlike in previous versions, where each node only holds a single slice (all adjacent to each other), the time window covers the entire bunch and radiation field and each node is assigned to a subdomain of the time window. Each node holds an array of many slices filling up the subdomain. Figure 1 illustrates the different approaches of the old and new version to assign the time domain to the nodes.

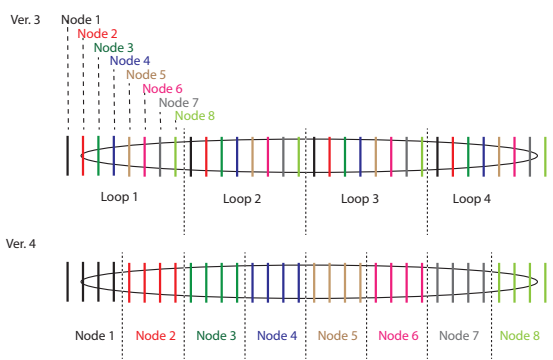

Figure 1: Assignment of radiation slices to the computer nodes in version 3 and 4 of Genesis.

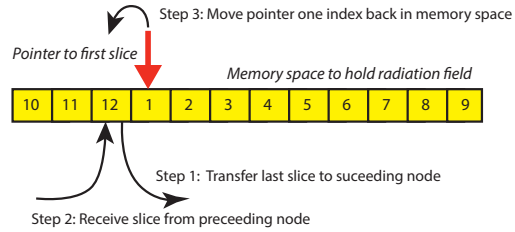

Figure 2: Indexing the radiation field when slippage is applied.

Management of the slices is done in the C++ part of the code, using the vector container of the C++ Standard Template Library (STL). One entry in the vector corresponds to a given slice. These containers are wrapped by the `Beam` and `Field` classes. Therefore harmonics and/or polarization can be modeled as an array of the field class avoiding the complicated and error-prone bookkeeping of a single array of the previous versions.

The most atomic algorithm advances a single slice of the electron distribution and the radiation field by one integration step. The particle tracker is based on a 4th order Runge-Kutta integrator [19] while the radiation field is advanced by an Alternating Direction Implicit (ADI) field solver [20]. This part is preserved from the Fortran code of the older version. Therefore the new version is still a hybrid version between C++ and Fortran albeit the Fortran part is significantly reduced. In a future release Genesis will move to C++ completely. Between the integration of the the beam and the field, Genesis now allows additional steps such as sorting the particle distribution and transition of particles between slices (see last section).

The code also differentiates between interaction within an undulator and the propagation through break sections. Also the integration step size can vary along the undulator. This avoids the restriction in older version that the undulator lattice has to fit to the granularity of a fixed integration step size. Slippage is now explicitly calculated and applied only once the field has advanced by more than the slice spacing of the electron slices. This corrects the bug in Genesis that slippage is enforced after each integration step even in undulator break section where the condition of one wavelength per undulator period no longer holds.

When slippage is applied then all wavefronts are advanced by one slice. However each node holds multiple wavefronts and shifting all data in memory is highly inefficient. Instead an index pointer, which indicates the first slice, is decreased. So when looping over all slices the loop starts at this index and ends at the final index, which is the start index plus the number of slices minus one. When the actual slice in the record is accessed the modulo of the loop index with the number of slices is taken. With this extra little bookkeeping very little data transfer is needed. Only one wavefront is sent to the next node in a 1D topology. Figure 2 shows the memory management of the radiation field for a node.

ISBN 978-3-95450-133-5

While calculating the slippage Genesis can also apply an autophasing for the drift section of two adjacent undulator modules, which simplifies the set up of new undulator lattices. The new element of a phase shifter allows adjusting the phase to the user's will.

CHANGES TO INPUT AND OUTPUT FORMAT

The latest version of Genesis breaks with the input format from the previous versions. Instead of a very long Fortran namelist with over 100 parameters the input is broken up into several smaller, thematically grouped namelists. The description of the undulator is completely removed from the main input deck and is now fully described by a lattice file. The overall structure is similar to the code Elegant [21], which needs also at least one input file and a lattice file. In the following the two files are briefly described.

Lattice File

The lattice files follow the syntax: *Label : Type = {Argument List};*. Elements are referred to by the label and can also be used to predefine new lattice elements by using the reference argument. They are basic elements such as undulators, drifts, quadrupoles and, as a new type, a line container, which can hold both types, elements and lines. The code unrolls references and nested lines up to 10 iterations. If at that level the element is not fully defined, the code will generate an error message and stop execution.

The line container allows also tagging each element with two features. One is the repetition of the element, which is indicated by a preceding integer number and the multiplication sign. Second, the absolute position of an element within the line can be specified by a succeeding @-sign and the position in meters. This feature is useful when quadrupoles are super-imposed onto an undulator.

An example of an input file is given below, which defines an undulator lattice with a FODO cell structure and 12 undulator modules.

```
QF1: Quadrupole = { l=0.15, k1= 1.1 };
QF2: Quadrupole = { ref=QF1, k1=-1.1 };
D1: Drift = { l=0.15 };
D2: Drift = { l=0.3 };
Un: Undulator = {lambdau=0.015,nwig=265,
                              aw=0.8485281};
FODO: LINE = {Un,D1,QF1,D2,Un,D1,QF2,D2};
SWISSFEL: LINE = {6*FODO};
```

The user must define at least on line, which is referred by the main input deck. Other lines, which are defined but not referred to, are ignored. Also this lattice can be regarded as a base lattice, from which modified lattices can be derived, such as taper profiles, quadruple misalignment etc. An explicit list of elements and the allowed syntax is given in the manual of the new version of Genesis.

Main Input File

The biggest change in the Genesis input is that the parsing of the input deck is event driven. So whenever a name list is closed by the *&end* statement, the content is parsed and, if needed, action is applied. The *setup* name list is mandatory and must come first. The *track* name list invokes the actual tracking of the particles and radiation field through the undulator. Before that the beam and field needs to be defined.

An example input file is listed below. Elements in namelist, which are not defined, are falling back to their default value. The complete list of all name lists and their elements is given in the manual.

```
&setup
 rootname=output-test
 lattice=SwissFEL.lat
 beamline=SwissFEL
 lambda0=1e-10
 gamma0=11357.8165
 delz=0.075
&end

&time
 slen=40e-6
 sample=10
&end

&lattice
 zmatch=9.15
&end

&sddsbeam
 file=inputdist.sdds
 match=true
 center=true
 output=true
&end

&field
 power=1000
 dgrid=0.3e-3
 ngrid=151
 waist_size=50e-6
&end

&track
 output_step=1
&end
```

The example shows some new features, such as automatic matching in the *lattice* name list. The matching point defines the end point in the cell of a periodic lattice (e.g. a FODO cell). With the given lattice definition and the reference energy *gamma0* Genesis calculates the matched β- and α-function values. These value can be used in the beam-

ISBN 978-3-95450-133-5

loading routine, but only if the match command/name list occurs before.

Genesis supports the direct import of Elegant output distributions by calling an external shell script which itself converts the SDDS format into an HDF5 format by using functions of the SDDS-Toolbox kit. The converted file is then read by Genesis and used to load the particle distribution within. If Elegant and the Toolbox Kit are not installed, Genesis will stop execution.

A field name list can occur multiple times to generate a superposition of field distributions. This can either be in the transverse direction (different Gauss-Hermite modes) or in longitudinal direction (different wavelengths). The latter is only possible in time-dependent simulations. In a similar way higher harmonics can be defined.

Output files

Only a few changes have been made to the output files. The format is still HDF5 but now the output for electron beam, undulator lattice and radiation field are grouped in the root level of the output file. Different harmonics will have different group names such as */field3* for the third harmonic. There are only particle and field dumps but the information can be dumped at various positions within the undulator lattice, reproducing the functionality of `IPPART` and `IPRADI` of the older versions.

POSSIBLE EXTENSION TO THE CODE

Based on the change of the core algorithm, keeping the entire radiation field and particle distribution in memory offers many possibilities which will expand the capability of Genesis 1.3. The following describes briefly the most significant features which will be added soon.

Sorting

Simulations, which resolve each individual electrons (also available in the older versions of Genesis), benefit from simpler algorithms avoiding the limitation of the quiet loading. The statistics are automatically correct at any harmonic which allows a very simple harmonic conversion in Genesis by simply slicing the particle distribution at the selected harmonics. Also particles can migrate into other slices without the restriction of a beamlet.

The sorting has to be done on all nodes simultaneously and tests have shown that a rather simple bubble sort algorithm is superior because the majority of electron motions is still limited and will end up in adjacent slices. Thus a single iteration of the bubble sort algorithm will be sufficient most of the time. The exceptions are strong mixing of particles such as in the chicanes of dispersive sections of laser-based or self-seeding schemes. Even here the motion is limited and the Bubble sort will be efficient, except in a very high harmonic conversion in EEHG schemes with particles shifted around on the scale of the total bunch length.

Self-Consistent Wakes and Space Charge Fields

In the older version a space charge field or wakefield has to be calculated externally and then added to the simulation as an effective potential. For that the current profile needs to be extracted from the input distribution. This can be done self-consistently in the new version because the current profile is known at any given time and can be broadcast to all nodes. Then each node calculates the resulting wakes over the given length of its assigned time-domain window based by the current profile ahead of the node. This makes the use of wake fields and space charge field less error prone (such as a mismatch in the wake potential and current profile in older versions because they are fed to Genesis by different files).

Self-Seeding Schemes

Similar to calculating wake fields from the entire bunch distribution dispersive effects in self-seeding configuration can be applied because the field content is known. As a prerequisite step the field is transferred into frequency domain by a 3D Fast Fourier Transformation. The field is now represented by plane waves with the total field vector k and the transverse wave numbers k_x and k_y. In frequency space the beam can easily be transported or the Green's function of a dispersive element can be applied. At the beginning of the next undulator section the field is converted back into the time domain.

Chicanes

In intra-undulator sections the electron transport is now completely decoupled from the field propagation and the electrons are no longer bound to the radiation field grid. This allows for more flexibility, in particular the implementation of chicanes. Here the particles are tracked by transport matrices and the total path lengths of the electrons are compared to the path lengths of the radiation field. The difference is compensated by shifting the radiation field in the forward direction.

CONCLUSION

Genesis 1.3 is currently adapted to large size computer clusters, utilizing the benefits of keeping the entire particle distribution and radiation field in memory, distributed over the nodes of the cluster. With the reverse in the looping order, collective effects such as wake fields, change in current profile due to magnetic chicanes and dispersive elements for the radiation field (e.g. self-seeding) can be modeled within a single run of Genesis, reducing the chances of error by manipulating input and output files for the electron distribution and radiation field.

The current status of Genesis allows reproducing the core features of the older version. Initial tests have shown an increase in the execution time by 10 to 20%. New features (magnetic chicanes, sorting, automatic harmonic conversion) will be added in the upcoming months.

ISBN 978-3-95450-133-5

REFERENCES

[1] J. Andruszkow *et al*, Phys. Rev. Lett. **85** (2000) pp. 3825-3829

[2] P. Emma *et al*, Nature Photonics **4**, (2010) pp. 641-647

[3] T. Ishikawa *et al*, Nature Photonics **6** (2012), pp. 540-544

[4] E. Allaria, *et al*, New Journal of Physics **14** 113009 (2012)

[5] W.M. Fawley, LBNL-49625-Rev. I (2004)

[6] S. Reiche, Nucl. Instr. & Meth. **A 429** (1999) 243.

[7] E.L. Saldin, E.A. Schneidmillser, and M.V. Yurkov, Nucl. Instr. &Meth. **A 429** (1999) 233

[8] L.T. Campbell and B.W.J. McNeil, Physics of Plasmas, **19** (2012) pp. 093119-093119-13

[9] H.P. Freund and T.M. Antonsen, Phys. Rev. **E 52** (1995) 5401

[10] G. Marcus *et al*, "FEL Code Comparison for the Production of Harmonics via Harmonic Lasing", presented at this conference: *36th Int. Free-Electron Laser Conf.*, Basel, 2014, TUP031.

[11] G. Geloni, V. Kocharyan and E. Saldin, Journal of Modern Optics **58:16** 1391 (2011)

[12] D. Xiang and G. Stupakov, Phys. Rev. STAB **12** (2009) 030702

[13] N. R. Thompson and B. W. J. McNeil, Phys. Rev. Lett. **100** 203901 (2008)

[14] A.E. Siegman, *Lasers*, (University Science Books, 1986, Sausalito)

[15] J. Wu, A. Marinelli and C. Pellegrini, "Generation of Longitudinally Coherent Ultra High Power X-ray FEL Pulses by Phase and Amplitude Mixing" in *Proc. 34th Int. Free-Electron Laser Conf.*, Nara, 2012, pp. 237-240

[16] D. Xiang *et al*, Phys. Rev. ST Accel. Beams **16**, 010703 (2013)

[17] S. Reiche and M. Carla, "Detailed Modeling of Seeded Free-electron Lasers" in *Proc. 34th Int. Free-Electron Laser Conf.*, Nara, 2012, pp. 101-104

[18] W.M. Fawley, Phys. Rev. STAB **5** (2002) 070701

[19] P. Kaps and P. Rentrop, Numerische Mathematik, **33** (1979) 55

[20] W.F. Ames, Numerical Methods for Partial Differential Equations (Academic Press, New York, 1977)

[21] M. Borland, *elegant: a Flexible SDDS-Compliant Code for Accelerator Simulation* APS LS-287 (2000)

ISBN 978-3-95450-133-5