ISBN: 978-3-95450-210-3 doi:10.18429/JACoW-FEL2019-TUP049

SIMULATING SHOT-NOISE OF 'REAL' ELECTRON BUNCHES

P. Traczykowski[1,2,3], L.T. Campbell[1,2,3], B.W.J. M[c]Neil[1,2]
[1]SUPA, Department of Physics, University of Strathclyde, Glasgow, G4 0NG
[2]Cockcroft Institute, Warrington, WA4 4AD, UK
[3]ASTeC, STFC Daresbury Laboratory, Warrington, WA4 4AD, UK

Abstract

An algorithm and numerical code for the up-sampling of a system of particles, from a smaller to a larger number, is described. The method introduces a Poissonian 'shot-noise' to the up-sampled distribution, typical of the noise statistics arising in a bunch of particles generated by a particle accelerator. The algorithm is applied on a phase-space distribution of relatively few simulation particles representing an electron beam generated by particle accelerator modelling software, for subsequent injection into an Free Electron Laser (FEL) amplifier which is used here to describe the model. A much larger number of particles is usually required to model the FEL lasing process than is required in the simulation models of the electron beam accelerators that drive it. A numerical code developed from the algorithm was then used to generate electron bunches for injection into to the unaveraged 3D FEL simulation code, Puffin. Results show good qualitative and quantitative agreement with analytical theory. The program and user manual are available at [1].

INTRODUCTION

When modelling a Free Electron Laser (FEL) [2], the number of simulation particles used to model the acceleration stages often needs to be increased before simulation of the FEL itself. This is mainly due to the fine longitudinal electron bunching structures at the FEL radiation wavelength that need to be modelled. Such fine detail is unlikely to be required when modelling the accelerator stages before the FEL. The data from the accelerator stages usually have a relatively sparse '*macroparticle*' distribution in phase space, and do not model the Poissonian noise statistics of a real electron distribution. Here, a method is described which breaks up this sparse phase–space distribution of macroparticles into a greater number of '*microparticles*', each representing fewer electrons, to give a more dense phase–space distribution that is suitable for injection into an FEL simulation code such as Puffin [3], used here as the target code to demonstrate the methods used. The method ensures that the microparticle distribution has the correct Poissonian shot–noise statistics of a real electron beam [4]. This is necessary to simulate the spontaneous light generation which arises from the shot-noise and acts as the seed field from which shorter wavelength amplifier FELs start up in the Self Amplified Spontaneous Emission mode of operation [2].

DENSITY DISTRIBUTION FUNCTIONS

The method first uses the macroparticles of the accelerator modelling stage to create a discrete charge histogram of bin

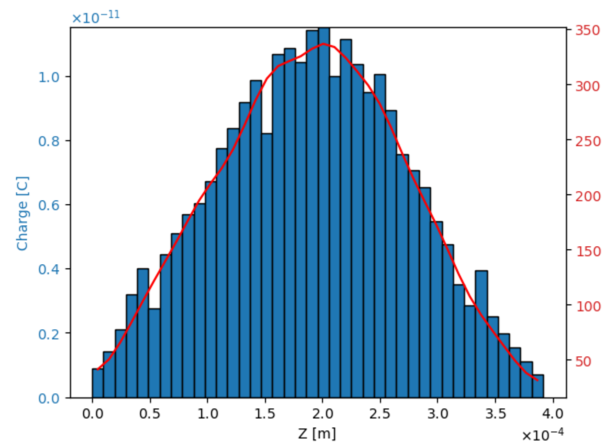

Figure 1: Sample electron density profile along the z–axis. The plot shows the original binned histogram of the macroparticle charge and the interpolated data. The original beam is sampled over 50 bins and then smoothed using the *Python SciPy scipy.ndimage.gaussian* filter [5].

width Δz_h along the longitudinal z-axis of beam propagation at a given time (e.g. on entering the FEL). A continuous, optionally smoothed, longitudinal charge distribution function $f_z(z)$, so the beam current $I(z) \approx c f_z(z)$ for the relativistic beams assumed here, is then created by interpolating from this histogram. This charge density function is then used to apply the correct charge $f_z(z)\Delta z_s$, to each longitudinal slice of microparticles of width Δz_s, where usually $\Delta z_h > \Delta z_s$. A transverse, optionally smoothed, charge density function $f_\perp(x, y, z)$ is then created for each longitudinal slice using a similar 2-D histogram-interpolation method. Here the z-dependence of $f_\perp(x, y, z)$ will be in integer units of the slice width Δ_s. The longitudinal density function and the transverse density function can then be interpolated to create a continuous 3-D charge density function, $f(x, y, z) = f_z(z) f_\perp(x, y, z)$. This function is significantly smoother than a discrete charge distribution of the initial, often relatively sparse, macroparticle distribution and allows for a less noisy, more realistic, distribution of microparticles for input into the FEL modelling software. The new set of microparticles is created in each longitudinal slice via a 2-D Cumulative Distribution Function (CDF) approach in the transverse plane – in this case we will call it Joint Distribution Function (JDF) [6]. The effects of Poissonian shot-noise are then added to each microparticle using the method of [4]. The microparticle distributions so generated can then model complex electron beam phase spaces. The longitudinal charge histogram and the corresponding current

ISBN: 978-3-95450-210-3 doi:10.18429/JACoW-FEL2019-TUP049

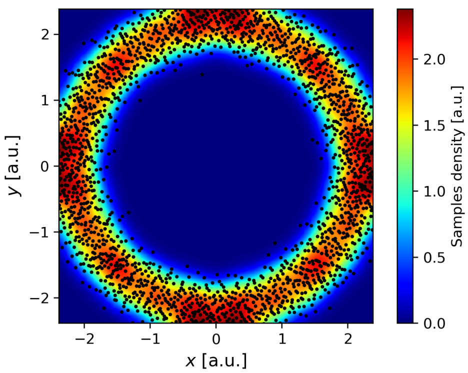

Figure 2: An example of a electron density map of a hollow beam in the transverse (x, y) plane. The initial 150 macroparticles (not shown) were first binned over 50x50 histogram grid. A Gaussian smoothing of factor of 1.5 times this grid spacing was used to obtain the transverse density function, here plotted over 500 x 500 points in a colour density map. The black dots are 2000 microparticles created using the JDF method.

evaluated from the charge distribution function, is shown in the example of Fig. 1. The electron beam parameters are similar to a beam generated in designs for the CLARA FEL test facility [7] and are: I_{pk}=395 A, ϵ_n=0.3 mm mrad, $\bar{\gamma}$=475, σ_γ=0.04, σ_t=273 fs and Q=250 pC.

MICROPARTICLE GENERATION

The joint probability distribution function approach described above for the transverse plane maintains all of the transverse electron beam structure which can occur across a beam. Projection of charge onto the x and y axes, so that $f_\perp(x, y, z) = f(x, z) \times f(y, z)$, would remove more complex transverse beam structures that may occur in real electron beams. An example of a more complex transverse structure that requires the use of a JDF is a hollow electron beam in the transverse plane. Fig. 2 demonstrates the use of the JDF in the transverse cross–section of such a beam, maintaining the hollow beam structure.

MOMENTUM DISTRIBUTION

Following the generation of the microparticles, the momentum of the macroparticles is mapped onto new microparticle set. The Python *SciPy* library [5] allows this via its *griddata* module. *griddata* interpolates the values of the macroparticle momenta onto the new microparticles using either nearest neighbour or a linear interpolation. This method of using JDF for creating new microparticle positions, and *griddata* for momentum, has proven to be efficient and sufficiently accurate.

NOISE APPLICATION

Shot-noise is applied to the microparticles only along the longitudinal z-axis as described in [4]. The shot-noise is applied via each microparticle's charge weight and its mean position in the z-axis. For example, if the distribution had N_λ microparticles per resonant wavelength λ_r, then the mean distance between each would be λ_r / N_λ. The noise is applied to position of the microparticle by altering its position so that $z \rightarrow z + \delta z$, where:

$$\delta z = \frac{\lambda_r}{N_\lambda \sqrt{N_e}} Rnd, \tag{1}$$

where N_e is the microparticle charge weight (i.e. the number of electrons it represents) and Rnd is random number in the range ±0.5. This value is calculated and applied individually to each microparticle. A random Poisson noise is then applied to the microparticle charge weight.

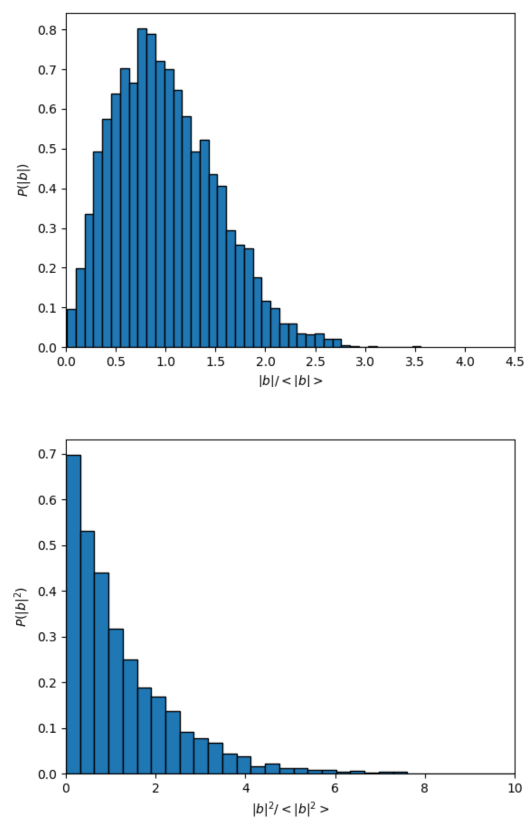

Figure 3: Microparticle probability distribution after JDF is applied showing top $P(|b|)$ (Rayleigh distribution function) and bottom $P(|b|^2)$ (negative exponential distribution function) in good agreement with the resuts of [4].

The results were tested by analysing the bunching statistics shown on Fig.3 and other beam statistics presented on Fig. 4. Additional test was performed to verify the bunching behaviour in absence of FEL interactions in simulation software Puffin where a flat top current was first generated with a sparse resolution and next the described up-sampling procedure was used to generate the new set of particles. The newly generated particles were then used as input file while

ISBN: 978-3-95450-210-3 doi:10.18429/JACoW-FEL2019-TUP049

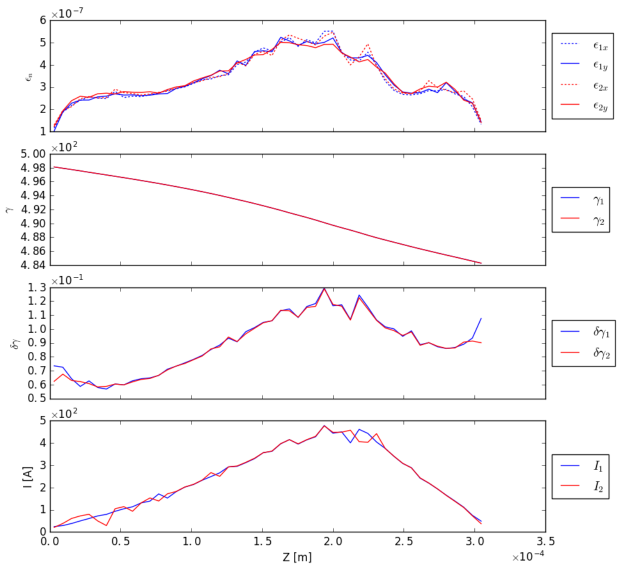

Figure 4: The plots show from the top: the normalised emittance ϵ_n; local Lorentz factor γ, local spread in the Lorentz factor δ_γ and current I for both the macroparticle beam (blue) and, after JDF is applied, the microparticle beam (red).

FEL interactions were turned off to allow the electron bunch propagate through undulator. The analytic model described in [4] was used to set the outer boundaries for the bunching parameters and slice of one wave length from the middle of the beam was selected for comparison. The expected result was that in absence of FEL interactions the beam will propagate through undulator and the bunching value in the selected slice will not exceed the boundaries set by analytic model. The results of the numerical experiment are shown on Fig. 5. The bunching statistics show good agreement with the model of [4].

COMPARISON OF DATA

In changing from the macroparticle to microparticle distribution, the macroscopic beam parameters, such as current, energy spread, emittance etc, should remain essentially unchanged. In this example the source was electron bunch generated by the ASTRA [8] software for a CLARA FEL [7] simulation. Fig. 4 plots beam parameters as calculated for both the macroparticle distribution at the output from ASTRA and the microparticle distribution following application of the JDF method as described above. Note that, as expected, the plots obtained for the microparticle distribution are a little smoother than the original macroparticle distribution. This is due to much more dense microparticle density along the z-axis. Further smoothing can be applied as described above.

CONCLUSION

The JDF method of up-sampling a system of macroparticles into a greater number of microparticles that have the

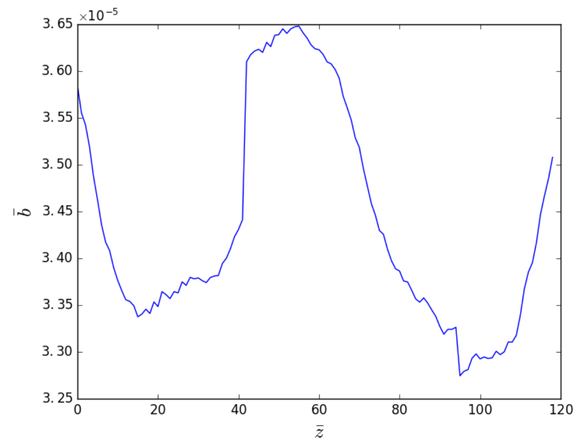

Figure 5: Bunching in electron beam propagating through the undulator with FEL interactions turned off. The $\overline{b}$ predicted by theory [4] for this configuration is 3.97×10^{-5} and $\sigma_{|b|}$ is 2.077×10^{-5}. As one can observe the results from numerical model of noise presented in this paper keeps within the limit predicted be the theoretical model.

equivalent shot-noise statistics as a system of electrons, is shown to give good agreement with the original macroparticle of energy spread, energy, emittance and current.The new microparticle beam also agrees with theoretical values for the bunching statistics. The JDF processed microparticle beam is more realistic that the initial macroparticle beam. The JDF method is computationally efficient and easy applicable in most computer programming languages - here Python was used. It has been successfully applied in a start–to–end type simulation of CLARA FEL using Puffin software. The output data was compared in [9] with the same simulation using Genesis 1.3 [10] and gave very good agreement.

ACKNOWLEDGMENTS

We gratefully acknowledge support of: The Science and Technology Facilities Council Agreement Number 4163192 Release #3; STFC's ASTeC department for HPC access, using the STFC Hartree Centre; and John von Neumann Institute for Computing on JUROPA at Jülich Supercomputing Centre under project HHH20.

REFERENCES

[1] https://github.com/UKFELs/JDF

[2] Brian W.J. McNeil and Neil R. Thompson, "X-ray free-electron lasers, *Nature Photon.*, vol. 4, 814 (2010). doi:10.1038/nphoton.2010.239

[3] L.T. Campbell and B.W.J. McNeil, "Puffin: A three dimensional, unaveraged free electron laser simulation code", *Phys. Plasmas* **19**, 093119 (2012). doi:10.1063/1.4752743

[4] B.W.J. McNeil, M.W. Poole and G.R.M. Robb, *Physical Review Special Topics - Accelerators and Beams* Vol 6, 070701 (2003), doi:10.1103/PhysRevSTAB.6.070701

ISBN: 978-3-95450-210-3 doi:10.18429/JACoW-FEL2019-TUP049

[5] E. Jones, E. Oliphant, P. Peterson, *et al.*, SciPy: Open Source Scientific Tools for Python, 2001 - , http://www.scipy.org/

[6] E. W. Weisstein, "Joint Distribution Function," http://mathworld.wolfram.com/JointDistributionFunction.html

[7] J.A. Clarke *et al.*, "CLARA conceptual design report"Journal Of Instrumentation, vol. 9, T05001 (2014). doi:10.1088/1748-0221/9/05/t05001

[8] K. Floettmann,"ASTRA - A space charge tracking algorithm," http://www.desy.de/mpyflo

[9] D. J. Dunning *et al.*, "Start-to-End Simulations of the CLARA FEL Test Facility", in *Proc. 9th Int. Particle Accelerator Conf. (IPAC'18)*, Vancouver, Canada, Apr.-May 2018, pp. 4430–4433. doi:10.18429/JACoW-IPAC2018-THPMK060

[10] S. Reiche, "GENESIS 1.3: a fully 3D time-dependent FEL simulation code", *Nucl. Instr. Meth. Phys. Res.* Sect. A, vol. 429, p. 243, 1999. doi:10.1016/S0168-9002(99)00114-X