ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-THPMK112

AN UPDATED DESCRIPTION OF THE FEL SIMULATION CODE PUFFIN

L.T. Campbell[1,2], J.D.A. Smith[3], P. Traczykowski[1] and B.W.J. M^{c}Neil[1]
[1] SUPA, Department of Physics, University of Strathclyde, Glasgow, G4 0NG and Cockcroft Institute, Warrington, WA4 4AD, UK
[2] ASTeC, STFC Daresbury Laboratory, Warrington, WA4 4AD, UK
[3] Tech-X UK Ltd, Sci-Tech Daresbury, Warrington, UK

Abstract

Puffin [1] is an unaveraged 3D FEL simulation tool with no Slowly Varying Envelope Approximation (SVEA), no undulator period averaging of the electron motion, and no periodic slicing of the electron beam, enabling simulation of broadband and high resolution FEL phenomena. It is a massively parallel code, written in modern Fortran and MPI, which scales from single core machines to HPC facilities. Its use in a number of projects since its initial description in 2012 has necessitated a number of additions to expand or improve its capability, including new numerical techniques, and the addition of a wide and flexible array of undulator tunings and polarizations along with electron beam optics elements for the undulator line. In the following paper, we provide an updated description of Puffin including an overview of these updates.

INTRODUCTION

The Free Electron Laser (FEL), being a tool for the investigation into ultra-small, ultra-fast phenomena [2], has multiple codes devoted to its simulation and design (see *e.g.* [3] - [6]). Many of these codes employ some standard approximations which reduce the domain of validity to a usually acceptable regime of slowly varying, slowly amplified mechanisms.

Puffin, originally reported in [1], was the first dedicated 3D FEL code without the SVEA, undulator period averaging of the electron motion, and periodic slicing of the electron beam. The intention was to provide the opportunity to research and design FELs in regimes where these nominally valid approximations break down, and to try to identify whether newly proposed schemes to produce improved temporal coherence, multiple colours or other more esoteric output required a more detailed description. It neglects the backwards propagating wave, and makes the paraxial approximation.

Since the initial description, Puffin has been used in several projects [7] - [14], some of which have necessitated changes to the numerical algorithms, and others to the feature set available in the code. Compared to the original publication, the code is now capable of more realistic scenarios, with more flexible input. The code has been published with an open source licence, and is developed on Github [15]. The motivation has been to continue to provide a flexible resource capable of modelling these esoteric situations, whilst enabling modelling of a typical FEL undulator line.

ANALYTIC MODEL

The up-to-date system of equations are written in terms of the normalised magnetic fields $b_\perp = b_x - ib_y$ (transverse) and b_z (longitudinal), where the magnetic field is $B_w(x,y,z) = B_0(b_x\hat{x} + b_y\hat{y} + b_z\hat{z})$, with B_0 being the peak magnetic field. The equations are:

$$\left[\frac{1}{2}\left(\frac{\partial^2}{\partial\bar{x}^2}+\frac{\partial^2}{\partial\bar{y}^2}\right)-\frac{\partial^2}{\partial\bar{z}\partial\bar{z}_2}\right]A_\perp = -\frac{1}{\bar{n}_p}\frac{\partial}{\partial\bar{z}_2}\sum_{j=1}^{N}\frac{\bar{p}_{\perp j}}{\Gamma_j}(1+\eta p_{2j})\delta^3(\bar{x}_j,\bar{y}_j,\bar{z}_{2j}) \tag{1}$$

$$\frac{d\bar{p}_{\perp j}}{d\bar{z}} = \frac{1}{2\rho}\left[i\alpha b_\perp - \frac{\eta p_{2j}}{\kappa^2}A_\perp\right] - i\kappa\frac{\bar{p}_{\perp j}}{\Gamma_j}(1+\eta p_{2j})\alpha b_z \tag{2}$$

$$\frac{d\Gamma_j}{d\bar{z}} = -\rho\frac{(1+\eta p_{2j})}{\Gamma_j}(\bar{p}_{\perp j}A_\perp^* + c.c.) \tag{3}$$

$$\frac{d\bar{z}_{2j}}{d\bar{z}} = p_{2j} \tag{4}$$

$$\frac{d\bar{x}_j}{d\bar{z}} = \frac{2\rho\kappa}{\sqrt{\eta}\Gamma_j}(1+\eta p_{2j})\Re(\bar{p}_{\perp j}) \tag{5}$$

$$\frac{d\bar{y}_j}{d\bar{z}} = -\frac{2\rho\kappa}{\sqrt{\eta}\Gamma_j}(1+\eta p_{2j})\Im(\bar{p}_{\perp j}), \tag{6}$$

where

$$\bar{z}_{2j} = \frac{ct_j - z}{l_c}, \qquad \bar{z} = \frac{z}{l_g},$$
$$\bar{p}_\perp = \frac{p_\perp}{mca_{u0}}, \qquad A_\perp = \frac{e\kappa l_g}{\gamma_0 mc^2}E_\perp,$$
$$(\bar{x},\bar{y}) = \frac{(x,y)}{\sqrt{l_g l_c}}, \qquad l_g = \frac{\lambda_w}{4\pi\rho},$$
$$l_c = \frac{\lambda_r}{4\pi\rho}, \qquad \Gamma_j = \frac{\gamma_j}{\gamma_0},$$
$$\rho = \frac{1}{\gamma_0}\left(\frac{a_{u0}\omega_p}{4ck_u}\right)^{2/3}, \qquad a_{u0} = \frac{eB_0}{mck_u},$$
$$\kappa = \frac{a_{u0}}{2\rho\gamma_0}, \qquad b_\perp = b_x - ib_y,$$
$$\eta = \frac{1-\beta_{z0}}{\beta_{z0}}, \qquad \eta p_{2j} = \frac{1-\beta_{zj}}{\beta_{zj}}$$

Note that in the above, the undulator parameter a_{u0} is the *peak* undulator parameter, $\beta_{z0} = v_{z0}/c$ indicates the velocity of reference particle with energy γ_0 averaged over an

ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-THPMK112

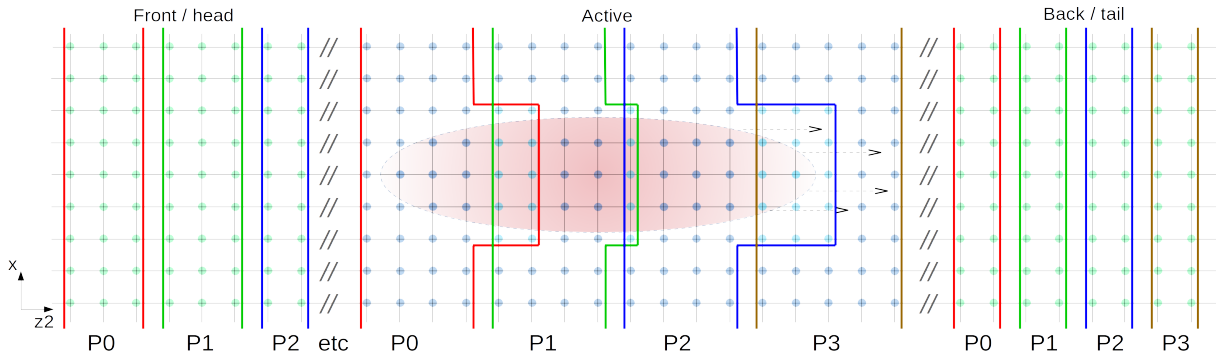

Figure 1: 2D representative illustration of the field mesh distribution over 4 MPI processes, labelled $P0 - P3$ in the above. The mesh is split into a front, back and active section as shown, where the active section is defined around where the electron beam is located (indicated by the red ellipse), and where it predicted to go. In this frame, electrons only move from left to right, as they slip backwards through the field. The field nodes from each of the sections are then distributed using a slab decomposition, and the electron macroparticles are then distributed to be with their relevant section of the active mesh. In the active section, buffer sections are setup to allow the macroparticles to stay on the same process as they slip into the buffers. The buffers must be continuously synchronised, and the dimensions of the buffers required by each process are calculated by predicting the motion of that process's macroparticles on a short time-scale.

undulator period, and the undulator tuning is varied using $\alpha(\bar{z}) = a_u(\bar{z})/a_{u0}$ [9].

3D magnetic fields for planar, curved pole, and helical undulators are as described in [16]. In addition, a generally polarised elliptical undulator is available with polarization in each undulator module controlled through relative field strengths u_x and u_y, where $0 \leq (u_x, u_y) \leq 1$, defined as

$$b_x = u_x \cos(\bar{z}/2\rho) \tag{7}$$

$$b_y = u_y \sin(\bar{z}/2\rho) \tag{8}$$

$$b_z = \frac{\sqrt{\eta}}{2\rho}(u_x \bar{x} \sin(\bar{z}/(2\rho)) + u_y \bar{y} \cos(\bar{z}/(2\rho))), \tag{9}$$

which possesses no off-axis field variation in the x and y magnetic fields. Nevertheless, it provides a natural focusing channel through the interaction with the transversly varying b_z field. The undulator ends are optionally smoothly tapered up and down at the ends of all undulator types to avoid CSE effects [17].

The inputs may be in the form of these scaled variables, or in SI notation (which is then converted internally by Puffin). The scaled reference frame is defined in the main input file by specifying ρ, γ_0, λ_u and a_{u0}. These are only scaling parameters, and S.I. input is scaled internally by Puffin using these input values.

The electron beam may now be input by: 1) a homogeneous beam distribution, specifying the emittance, charge, Twiss parameters and/or standard deviations in each of the 6 beam dimensions, from which a macroparticle distribution is generated internally; 2) a series of sub-wavelength slices in $\bar{z}_2$, each specified by a charge, Gaussian mean and standard deviation in each of the remaining 5 dimensions, from which macroparticles are generated internally; 3) importing an externally generated macroparticle distribution, or output from a previous Puffin simulation.

In addition, Puffin now models physical drifts between undulator modules, chicanes, quads (thin lens approximation), and 'modulation' sections as simple point transforms, to allow more representative modelling of a realistic undulator line design.

NUMERICAL SOLUTION

The basic split-step Fourier-RK4 method is still as described in [1], whereby each step is decomposed into a free-space field propagation (radiation diffraction) step, and a step solving the radiation generation from the electron beam concurrently with the electron equations (2) - (6) using the standard RK4 method. However, the Finite Element Method is no longer used, as the electron macroparticles are interpolated directly onto the field mesh, removing the need for an external linear solver.

The solution of the radiation diffraction is still performed in Fourier space, providing an analytic solution which is valid for all frequencies, and FFTW v3.3+ is now used for the Fourier Transforms rather than FFTW v2.1.5 for the purposes of portability.

Another significant change is that the field mesh is now fully parallelized, rather than being stored on each MPI process in its entirety. Rather than using a straightforward slab decomposition, for more efficient computational load balancing an active volume of the mesh is defined, corresponding to the volume of the mesh which contains the electron macroparticles. This active mesh is then distributed amongst MPI processes in a slab decomposition in $\bar{z}_2$, and the macroparticles are distributed to their relevant active local slabs. The electron beam is guaranteed to move to increasing $\bar{z}_2$, and so overlapping, parallel buffers are added to the end of each process's mesh, which must be synchronized every step. See Fig. 1. Once a macroparticle travels beyond the edge of this buffer, the parallel distribution is no longer valid, and the position of the active mesh is redefined according to the new beam layout, and redistributed.

ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-THPMK112

The length of each local mesh buffer is calculated from the process's local macroparticles, so that,

$$L_b = \max(p_2) L_{\bar{z}} \approx \frac{L_{\bar{z}}}{\min(\Gamma)} \frac{(1+\alpha^2 a_{u0}^2)}{(1+a_{u0}^2)}, \qquad (10)$$

where the approximation is valid in the limit $\gamma_0 \gg 1$, the *max* and *min* indicate the *local* maximum and minimum values, and $L_{\bar{z}}$ is a parallel tuning parameter which specifies how long the parallel layout is wished to be valid for. $L_{\bar{z}}$ therefore controls the amount of communication every step, is by default 2 undulator periods, and can be modified by the user.

To further restrict the amount of communication in each step, the number of transverse nodes needed to be continually synchronized for the present parallel layout is also calculated, and only these (usually inner) nodes are syncronised, leading to an N_x^2 reduction in the communication load.

The leftover 'front' and 'back' sections of the field mesh are also distributed evenly over MPI processes to prevent uneven *memory* load. They are not involved in the field source computation, since by definition the electron macroparticles are only interacting with the active mesh, but they must be included with the radiation diffraction steps, since the entire mesh must be Fourier transformed. Unfortunately, FFTW does not allow parallel transforms with anything but a straightforward slab decomposition, and so for every diffraction step, the mesh is redistributed to the layout required for FFTW. The communication to redistribute into the FFTW layout and back again is costly, as is the Fourier transform itself, being an essentially all-to-all communication. To this end, a 'localised' Fourier space solution as in [18] may be implemented in the future. For now, the problem is mitigated by performing the diffraction step at an interval which is by default every undulator period rather than every RK4 step. This interval may be altered by the user in the Puffin input file. In most practical cases there is negligible difference when solving radiation diffraction every undulator period, as most often the Rayleigh range is much larger than the undulator period. This reduces the number of diffraction calculations required by ≈ 30 times.

TIME-SAVING MODES

The full 3D model may be reduced to a 1D model, as much of the extra physics emerging through the use of Puffin (fast temporal variations in the system) are essentially 1D phenomena. Enhancing the 1D energy spread to give the same M. Xie gain [19] as the full 3D model usually results in almost identical results.

An additional mode has been added which utilizes periodic boundaries over the field mesh in $\bar{z}_2$. Recalling that $\bar{z}_2$ is the radiation field frame, utilizing periodic boundaries here results in an effectively infinitely long beam/radiation field system being modelled, but retains the sub-cycle behaviour enabled by the non-averaged, non-SVEA model. This can be used to simulate, for example, 1 full cycle of an EEHG seed (sometimes $> 50\lambda_r$ long), or a system only one λ_r long,

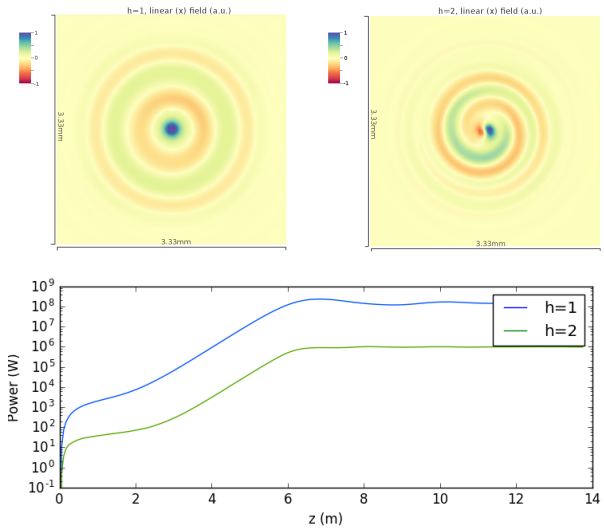

Figure 2: Top - phase profiles of the fundamental (left) and second harmonic (right) from the UK test FEL CLARA [20] using helical undulators, lasing at $100nm$, radiated after $10m$. Bottom - Power vs z for the 2 modes. The radiated power of the 2^{nd} harmonic ($h = 2$) is $\gtrsim 0.1\%$ of the fundamental power.

but containing the full harmonic spectrum (limited only by the Nyquist frequency), with all the relative phase information included. 3D simulations in this mode can be used to optimize the system without tracking a fully temporal (and computationally expensive) run. Using the periodic mode one can feasibly perform a 3D Puffin simulation, solving for quads and drifts as well as undulators, on a laptop in $\approx$ 1 minute, modelling the sub-cycle FEL behaviour and harmonics.

Filtering the harmonics out from the field in post, one can show their interesting transverse phase relationships, and their evolution in the high gain regime. For example, one can observe the evolution of the harmonics arising from helical undulators, see figure 2.

The output files in Puffin are now by default in HDF5 format [21], and include vizSchema [22, 23] metadata for remote visualization from an HPC cluster in Visit [24]. Data files include the full dump of the radiation field vectors ($A_\perp$) at the nodes of the field mesh, the full electron macroparticle 6D phase space coordinates + charge weights, and an integrated data file, containing reduced data like current, power, beam radius, *etc*. The frequency of 'integrated' and 'full' data dumps are controlled independently, with the integrated data being written every period by default.

Puffin includes a number of exemplar Python post-processing and visualization packages utilizing numpy and matplotlib, which can *e.g.* match an external beam to the lattice, and include routines to plot the power, energy, and other such 'standard' FEL plots. It is expected, however, that different projects will have their own requirements, due to the nature of the code and the different scenarios it will be employed to model.

ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-THPMK112

ACKNOWLEDGMENTS

We gratefully acknowledge support of the Science and Technology Facilities Council Agreement Number 4163192 Release #3; ARCHIE-WeSt HPC, EPSRC grant EP/K000586/1; EPSRC Grant EP/M011607/1; and John von Neumann Institute for Computing (NIC) on JUROPA at Jülich Supercomputing Centre (JSC), under project HHH20.

REFERENCES

[1] L.T. Campbell and B.W.J. M^{c}Neil, Phys. Plasmas p. 19, 093119 2012

[2] B.W.J. M^{c}Neil and N. R. Thompson, Nat. Photonics p. 4, 814 2010

[3] S. Reiche, Nucl. Instrum. Meth. A p. 429 243-248 1999

[4] H.P. Freund, P.J.M. van der Slot, D.L.A.G. Grimminck, I.D. Setya, and P. Falgari, New J. Phys. p. 19, 023020 (2017)

[5] E.L. Saldin et al., Nucl. Instrum. Methods Phys. Res. A p. 429, 233-237 1999

[6] W.M. Fawley, in Proceedings of FEL 2006, BESSY, Berlin, Germany, JACoW, pp. 218-221 2006

[7] L.T. Campbell and A.R. Maier, New J. Phys. p. 19 033037 2017

[8] S.V. Benson *et al*, J. Phys.: Conf. Ser. p. 493 012025 2014

[9] L.T. Campbell *et al*, New J. Phys. p. 16 103019 2014

[10] E. Hemsing *et al*, Phys. Rev. ST Accel. Beams p. 20 060702 2017

[11] B. Garcia *et al*, Proceedings of FEL2017, Santa Fe, NM, USA 2017

[12] B. Garcia *et al*, Phys. Rev. ST Accel. Beams p. 19, 090701 2016

[13] J.R. Henderson *et al*, New J. Phys. p. 17 083017 2015

[14] J.R. Henderson *et al*, New J. Phys. p. 18 062003 2016

[15] https://github.com/UKFELs/Puffin

[16] J.R. Henderson *et al*, Proceedings of FEL2014, Basel, Switzerland 2014

[17] B.W.J. McNeil and G.R.M. Robb, Phys. Rev. E, p. 65, 046503 2002

[18] J.L. Vay *et al*, J. Comp. Phys. p. 243 260-268 2013

[19] M. Xie, in Proceedings of the Particle Accelerator Conference, Dallas, TX, 1995 (IEEE, New York, 1995)

[20] J.A. Clarke *et al*, JINST p. 9 T05001 2014

[21] http://hdf.ncsa.uiuc.edu/HDF5/

[22] S. Shasharina *et al*, Proceedings of IPAC10, Kyoto, Japan

[23] https://ice.txcorp.com/trac/vizschema/wiki/WikiStart

[24] H. Childs *et al*, A Contract-Based System for Large Data Visualization, Proceedings of IEEE Visualization 2005, pp 190-198, 2005