A Time-Discrete Vlasov Approach to LSC Driven Microbunching in FEL-like Beam Lines

Philipp Amstutz*

Hamburg University, Dep. of Physics, Hamburg, Germany, EU

Mathias Vogt

Deutsches Elektronensynchrotron DESY, Hamburg, Germany, EU

Multi-stage bunch compression setups which are commonplace in Free-Electron Laser (FEL) beam lines have shown to be the source of a collective beam instability, the so-called microbunching instability. Collective forces can, in conjunction with the dispersive effect of the magnetic chicane present in such a compressor stage, lead to an amplification of inhomogeneities in the longitudinal charge density of an electron bunch. On the basis of a well established model this process can be investigated via the evolution of a time-discrete Vlasov system. Due to the exotic shape of the longitudinal phase space covered by typical FEL-enabled electron bunches, naive grid-based approaches to the computer simulation of such systems are prone to be prohibitively wasteful in terms of memory and computation time. In this contribution we present an efficient simulation approach based on the Perron-Frobenius operator method and quadtree domain decomposition.

1. Motivation

Free-electron lasers require high brightness electron bunches to drive the quintessential FEL instability in the undulators. Producing beams with the required peak current directly at the gun is problematic, as space-charge effect will quickly deteriorate the quality of the – at this stage – low energetic bunches. FEL facilities therefore typically employ the concept of bunch compression. A moderately long bunch is produced at the photo cathode, which is successively compressed to the desired peak current by means of a series of bunch compressor stages at increasing energies. The RF-cavities in such a compressor are operated off-crest which causes a longitudinal energy chirp to be imprinted on the bunch. Compression is then achieved due to the dispersion in a subsequent magnetic chicane. It has become clear that such a setup can give rise to the so-called micro-bunching instability that can potentially disturb the bunch to a degree making it unusable for FEL operation. This is because the described setup allows the interaction between energy- and density-inhomogeneities present in the bunch in a way that can lead to the amplification of both. Along drift sections, Coulomb interaction with self-fields generated by any density inhomogeneities results in a growth of local energy deviations. In dispersive sections these energy inhomogeneities then affect the longitudinal charge density distribution, potentially enhancing the initial inhomogeneities therein.

*philipp.amstutz@desy.de

2. Bunch-Compressor Model

In our investigations we will consider a one-dimensional model[1–3] of the system in the longitudinal phase space with the coordinates $\vec{z} = (q, p)^T = (\xi - \xi_0, E - E_0)^T$, where $\xi_{(0)}$ denotes the longitudinal (reference) position in the bunch and $E_{(0)}$ is the (reference) energy. We will consider ultra-relativistic bunches and hence ignore velocity effects. As this model shall focus on space-charge driven effects, (coherent) synchrotron radiation will be neglected. In this model the effect of a magnetic chicane on the longitudinal phase-space vector of an electron is given by the drift-type map

$$D^{\text{chic}} : \mathbb{R}^2 \to \mathbb{R}^2, \quad \begin{pmatrix} q \\ p \end{pmatrix} \mapsto \begin{pmatrix} q + R(p) \\ p \end{pmatrix}, \tag{1}$$

with a dispersion function $R \in C^1(\mathbb{R}, \mathbb{R})$. This map does not include any collective terms, which is justified by the assumption that the chicane is short compared to the linac part of the compressor stage, in which the significant portion of the Coulomb interaction will take place. Such a dispersion-free linac section is described by the kick-type map

$$K[\Psi]^{\text{cav,LSC}} : \mathbb{R}^2 \to \mathbb{R}^2, \quad \begin{pmatrix} q \\ p \end{pmatrix} \mapsto \begin{pmatrix} q \\ p + F[\Psi](q) \end{pmatrix}, \tag{2}$$

where the force term $F \in C^1(\mathbb{R}, \mathbb{R})$ describes the position dependent change in energy due to off-crest acceleration and space-charge effects. Collective effects are accounted for by the explicit dependence of F on the phase-space density $\Psi(\cdot\,; s) \in \mathcal{L}_1(\mathbb{R}^2, \mathbb{R})$ of the bunch at position s along the beam line. Note that for our numerical approach we will additionally require the phase-space density to be sufficiently smooth with respect to the chosen interpolation scheme, i.e. $\Psi(\cdot\,; s) \in C\mathcal{L}_1^n \equiv \mathcal{L}_1(\mathbb{R}^2, \mathbb{R}) \cap C^n(\mathbb{R}^2, \mathbb{R})$ in case of $(n-1)$th order bipolynomial interpolation. Without further knowledge, it appears that determining a self-consistent kick-map from Equation (2) requires a time-continuous solution for the evolution of Ψ along the linac. This is not the case, as will be illustrated in the following section. For a more detailed discussion of the presented maps refer to the contribution[4] of M. Vogt which uses the same base-model.

3. Perron-Frobenius Operator

We present a Perron-Frobenius type simulation code, which tracks a grid-based numerical representation of the phase-space density Ψ itself, rather than single particles. The evolution of a phase-space density can be elegantly formalized by the introduction of the Perron-Frobenius operator. Let $M(\cdot; s) : \mathbb{R}^2 \to \mathbb{R}^2$ be the solution family of the single-particle equations of motion along a beam-line section parameterized by the coordinate s. For measure-preserving maps M it can be deduced that the phase-space density of a system is conserved along single particle

trajectories; a result well known as Liouvilles theorem

$$\Psi(\vec{z};0) = \Psi\big(M(\vec{z};s);s\big) \quad \forall s. \tag{3}$$

If M additionally is invertible equation (3) directly yields an expression for the time-forward phase-space density

$$\Psi(\vec{z};s) = \Psi\big(M^{-1}(\vec{z};s);0\big), \tag{4}$$

so that the solution can be identified as

$$\Psi(\cdot\,;s) = \Psi(\cdot;0) \circ M^{-1}(\cdot\,;s) =: \mathcal{M}(s)\Psi(\cdot\,;0), \tag{5}$$

where $\mathcal{M}(s) \in \mathrm{lin}(\mathcal{L}_1, \mathcal{L}_1)$ is called the Perron-Frobenius operator associated to the map $M(\cdot\,;s)$. Again, for the numerics we require $M(\cdot\,;s)$ to be suitably smooth, so that $\mathcal{M}(s) \in \mathrm{lin}(C\mathcal{L}_1^n, C\mathcal{L}_1^n)$. In the context of the application in a simulation code, the central result is that a time-forward phase-space density can be evaluated at a given point by tracking this point back to a position in the beamline where the phase-space density is known and evaluate this density at the resulting point. The maps (1) and (2) are symplectic, thus measure preserving and invertible, so that Liouville's theorem can be applied using the Perron-Frobenius operators $\mathcal{D}^{\mathrm{chic}}$ and $\mathcal{K}[\Psi]^{\mathrm{cav,LSC}}$. Hence, the phase-space density after the nth bunch compressor stage is given by

$$\Psi_n = \mathcal{D}^{\mathrm{chic}}\, \mathcal{K}[\Psi_{n-1}]^{\mathrm{cav,LSC}}\, \Psi_{n-1}. \tag{6}$$

Furthermore, we can now motivate, why our model allows for a self-consistent, yet time-discrete treatment of collective effects in the linac section. Considering only space-charge effects, the collective part of $F[\Psi]$ in the linac-map (2) describes the interaction with the electric fields generated by inhomogeneities in the longitudinal charge density $\lambda(q;s) = \int_{\mathbb{R}} \Psi(q,p;s)\mathrm{d}p$. The fields are determined by Poisson's equation which generally yields solutions in the form of a convolution (here denoted by the $*$ operator)

$$E^{\mathrm{LSC}}(\cdot\,;s) \propto \tilde{Q}[S,\hat{S}](\cdot\,;s) * \lambda(\cdot\,;s), \tag{7}$$

where $\tilde{Q}[S,\hat{S}]$ is the Fourier transform of an LSC-impedance function that depends on geometrical assumptions which will be discussed in the next section. It is important to note that the dependence of the kick-map (2) on $\Psi(\cdot;s)$ therefore is restricted to a dependence on $\lambda(\cdot;s)$ – that is the projection of $\Psi(\cdot;s)$ – which, in fact, is invariant under general kick-maps $K(s) : (q,p) \mapsto (q, p + \kappa(q;s))$

$$\lambda(q;s) = \int_{\mathbb{R}} \Psi(q,p;s)\mathrm{d}p = \int_{\mathbb{R}} \mathcal{K}(s)\Psi(q,p;0)\mathrm{d}p \tag{8}$$

$$= \int_{\mathbb{R}} \Psi(q,p-\kappa(q;s);0)\mathrm{d}p = \int_{\mathbb{R}} \Psi(q,p;0)\mathrm{d}p \tag{9}$$

$$= \lambda(q;0). \tag{10}$$

The spatial charge density is frozen along the linac! Therefore equation (7) can be integrated over the length d of the linac section

$$F^{\mathrm{LSC}}[\Psi] \propto \int_0^d \tilde{Q}[S,\hat{S}](\cdot\,;s) * \lambda(\cdot\,;s)\,\mathrm{d}s \tag{11}$$

$$= \lambda(\cdot\,;0) * \int_0^d \tilde{Q}[S,\hat{S}](\cdot\,;s)\,\mathrm{d}s. \tag{12}$$

As a result, our model allows the treatment of the magnetic chicane as well as the linac part via a single time-discrete maps, which is visualized in Figure 1.

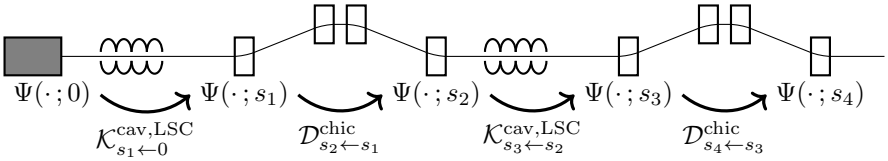

Fig. 1. Time-discrete evolution of the phase-space density through two bunch-compressor stages.

4. LSC impedance model

For a one-dimensional LSC force model, assumptions about the bunch geometry in the transverse direction have to be made. In our approach we model this longitudinal space charge force as the force acting on a test-charge distribution with the transverse shape $\hat{S}(r,\phi)$ at the position q within a bunch with transverse shape $S(r,\phi)$. Following standard literature[5], it can be seen that the dependence of this force on the shape functions can be absorbed in single impedance term $Q[S,\hat{S}]$

$$F^{\mathrm{LSC}} = \frac{\mathrm{i}\,q_0\,\mathrm{e}}{\epsilon_0}\mathcal{F}^{-1}\left(Q[S,\hat{S}]\cdot\mathcal{F}\lambda\right), \tag{13}$$

where q_0 is the bunch charge, $\mathcal{F}$ denotes Fourier transformation and

$$Q[S,\hat{S}](k) := \sum_{m=-\infty}^{\infty} \int_0^{2\pi}\!\!\int_{-\infty}^{\infty} k\,r\,\bar{r}\,S(r,\phi)\,\hat{S}(\bar{r},\bar{\phi})\,e^{\mathrm{i}\,m(\bar{\phi}-\phi)} \times I_m(|k|\,\min(r,\bar{r}))\,K_m(|k|\,\max(r,\bar{r}))\mathrm{d}\phi\,\mathrm{d}\bar{\phi}\,\mathrm{d}r\,\mathrm{d}\bar{r}. \tag{14}$$

We have calculated this term for various useful combinations of the shape functions

$$S^D_a(r,\phi) \equiv \begin{cases} \frac{1}{\pi\,a^2} & r \le a \\ 0 & r > a \end{cases}, \quad S^\delta(r,\phi) \equiv \frac{\delta(r)}{2\pi\,r}, \quad \text{and} \quad S^G_\sigma(r,\phi) \equiv \frac{\mathrm{e}^{-\frac{1}{2}(\frac{r}{\sigma})^2}}{2\pi\sigma^2}, \tag{15}$$

representing a uniformly charged a-Disk, a single on-axis electron, and a radially Gaussian distribution respectively. The resulting functions are shown in Figure 2 in terms of their dimensionless equivalent $\mathfrak{Q}[\cdot,\cdot](\xi) \equiv a\,Q[\cdot,\cdot](\xi/a)$,

$$\mathfrak{Q}[S_a^D, S^\delta](\xi) = \frac{2}{\xi}\big[1 - |\xi|\,K_1(|\xi|)\big] \tag{16}$$

$$\mathfrak{Q}[S_a^D, S_a^D](\xi) = \frac{2}{\xi}\big[1 - 2\,K_1(|\xi|)\,I_1(|\xi|)\big] \tag{17}$$

$$\mathfrak{Q}[S_{a/2}^G, S^\delta](\xi) = \frac{\xi}{4}\,\mathrm{e}^{\frac{\xi^2}{8}}\,\Gamma\left(0, \frac{\xi^2}{8}\right). \tag{18}$$

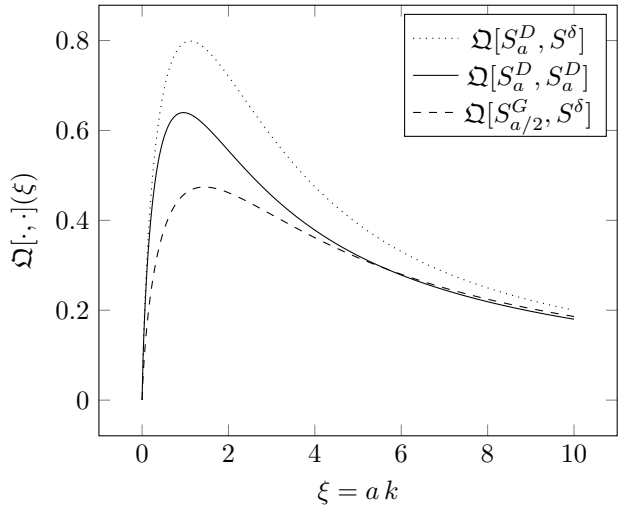

Fig. 2. Plot of the impedance terms resulting from the Disk-δ, Disk-Disk and the Gauss-δ model. Note that all functions are odd and only the positive domain is shown.

It can be seen that while the maxima and bandwidths of the impedance terms depend on the choice for the bunch geometry, all terms converge in the high frequency limit. Note that $\sigma = a/2$ has been chosen for the Gaussian term in this comparison, because the transversal distributions S_a^D and $S_{a/2}^G$ feature the same variance.

5. Quadtree Domain Decomposition

Due to the collective dependence an analytical solution to Equation (6) can in general not be obtained, so that one has to resort to numerical solution methods. In general, any such approach involves a numerical representation of the phase-space density inside the limited memory of a classical computer. Typically the PSD drops off sufficiently fast, so that it can be truncated at a threshold ϵ without loosing significant contributions to the dynamics of the system. One of the most

straight-forward ways to represent the truncated PSD is to store its function values on a homogeneously sized grid and use an interpolation scheme for the evaluation between the grid points. Trying to apply this to the case at hand one quickly encounters a problem that is depicted in Figure 3. Typical FEL bunches form a thin, hair-like structure that exhibits a strong curvature in the longitudinal phase space. We call these PSDs *exotic* (of degree η) because the area of their minimal enclosing rectangle is much larger than the area they actually cover, or formally

$$(\sup Q_{{}_\epsilon\Psi} - \inf Q_{{}_\epsilon\Psi})(\sup P_{{}_\epsilon\Psi} - \inf P_{{}_\epsilon\Psi}) = (\eta + 1)\,||\mathrm{supp}({}_\epsilon\Psi)||, \tag{19}$$

where $Q_{{}_\epsilon\Psi} \equiv \{q \,|\, \exists p : (q,p) \in \mathrm{supp}({}_\epsilon\Psi)\}$, $P_{{}_\epsilon\Psi}$ analogously, and ${}_\epsilon\Psi$ is the truncated PSD defined by ${}_\epsilon\Psi(\vec{z}) \equiv \Psi(\vec{z})$ if $\Psi(\vec{z}) > \epsilon$ and 0 otherwise. sup, inf and supp denote the supremum, infimum and support respectively. In the best case, sampling an η-exotic truncated PSD on a homogeneous $n \times n$ grid results in $\approx \eta\, n^2$ zeros to be stored in memory. Hence, $O(n^2)$ memory (and computation time) is wasted for handling values that do not contribute to the simulation at all, which makes this approach infeasible at high resolutions.

Our code minimizes this waste of computational resources by employing the method of quadtree[6] domain decomposition, which is illustrated in Figure 3. The basic idea is to divide the phase space into a hierarchy of rectangles decreasing in size. Any rectangle that intersects the support of ${}_\epsilon\Psi$ is divided in four similar, disjoint *child*-rectangles unless a maximum recursion depth r is reached. In a graph, where the nodes represent rectangles and edges represent parent-child relations this results in a tree, where each non-leaf node is of degree 4. Only on the smallest rectangles – the bottom-level leafs – the function values of the PSD are stored on a uniform grid.

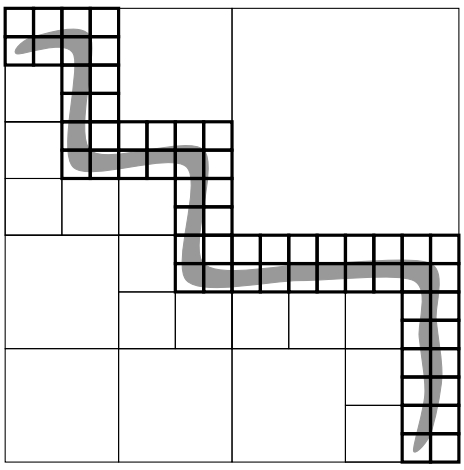

Fig. 3. Schematic of quadtree domain decomposition with depth 3. The support of ${}_\epsilon\Psi$ is shown in grey. Smallest *leafs* are marked with thick lines; only here function values are stored.

Such a tree data structure can be easily implemented in any programming language that supports pointers. To evaluate the function at $\vec{z}$, the bottom-level leaf containing $\vec{z}$ is found and an interpolated function value based on the samples stored in this leaf is returned. If $\vec{z}$ does not lie inside one of the smallest rectangles, 0 is returned. Using pointers, the algorithm to find the containing leaf has a runtime $O(r)$ and hence is negligible in performance considerations. Such a tree will contain bottom-level leaf that actually do not contain any density themselves, but were generated because one of their siblings does. The amount of these artifacts scales with the number of leafs on the boundary of $\mathrm{supp}({}_\epsilon\Psi)$. Hence, the waste of memory is reduced from $O(n^2) = O(2^{2r})$ to $O(2^r) = O(n)$. In addition to the parent-child pointers, in our implementation we also include pointers connecting neighbouring rectangles of the same size.

The resulting data-structure excels naive data-grids in many more respects besides the memory efficiency. The pointer structure allows for the direct application of well-known graph algorithms that can be used to efficiently implement many physically important operations. Traversal algorithms, for instance, can be used for a fast evaluation of integrals of Ψ such as projections, statistical momenta and general expectation values. Additionally boundary-finding algorithms can be employed to gain geometrical information about the phase-space density, e.g. its shape and location in phase-space. As described in the next section this can be used to overcome the possible problem of densities *escaping* from the simulation window.

6. Perron-Frobenius Simulation Step

In this section an efficient numerical implementation of a Perron-Frobenius simulation step will be presented. Given an initial phase-space density Ψ_i and a solution M of the single particle equations of motion along a beam-line element, the final phase-space density Ψ_f can be determined from Equation (5), which states that the value of Ψ_f at $\vec{z}$ is equal to the value of Ψ_i at the *backward-tracked* phase-space position $M^{-1}(\vec{z})$.

Firstly Ψ_i is brought from its input format (analytic mathematical expression, particle ensembles, etc.) into the quadtree form described in the previous section. In the process a sensible threshold ϵ is chosen[†]. In a next step the minimum bounding rectangle (MBR) of $\mathrm{supp}(\Psi_f)$ is calculated. For this, the boundary $\partial\mathrm{supp}(\Psi_i)$ is determined, which can be achieved efficiently using the aforementioned neighbour-pointers for PSDs with connected support. It can be seen that

$$\mathrm{MBR}[\mathrm{supp}(\Psi_f)] = \mathrm{MBR}[\partial\mathrm{supp}(\Psi_f)] = \mathrm{MBR}[M(\partial\mathrm{supp}(\Psi_i))].$$

Subsequently a new quadtree PSD is initialized, with its root rectangle covering

[†]To avoid notational clutter, we will no longer explicitly denote this truncation in the following: ${}_\epsilon\Psi \to \Psi$. Furthermore we will use standard mathematical notation, even when referring to numerical equivalents, e.g. "a set of points that fully characterizes the boundary of the support of Ψ_f" $\to \partial\mathrm{supp}(\Psi_f)$.

the calculated phase-space region. This tree is then refined at a *seed* position $M(\vec{z}), \vec{z} \in \text{supp}(\Psi_i)$ to the desired depth. Starting from this seed leaf the tree is then grown by utilizing a flood-fill algorithm: The value of $\Psi_i \circ M^{-1}$ is calculated for a representative number of points, lying on the edges of the leaf. If a function value exceeds the threshold the same-size neighbour-leaf in the respective direction is generated, in case it did not exist yet. This process is repeated for all newly generated bottom-level leafs. If This algorithm terminates when all bottom-level leafs necessary to cover the connected component of $\text{supp}(\Psi_f)$ containing the seed are generated. In case of multiple connected components the process is repeated with a seed from the next connected component. Note that *generating a leaf* here refers to inserting the respective node into the tree and storing the values of $\Psi_i \circ M^{-1}$ on a grid covering the leaf for later interpolation. Figure 4 summarizes the described simulation steps visually.

This flood-fill approach is efficient in the sense that Ψ_i is rarely evaluated unnecessarily, i.e. evaluated outside of its support. This only happens when the flood-fill algorithm works on the border of $\text{supp}(\Psi_f)$ so that the waste in computation time is $O(||\partial\text{supp}(\Psi_f)||) = O(2^r)$.

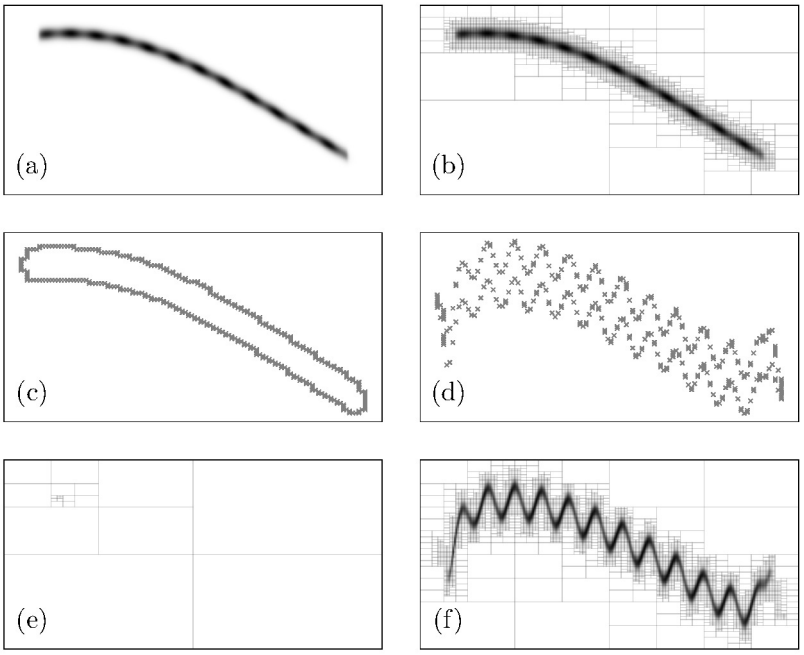

Fig. 4. (a) Initial phase-space density Ψ_i, (b) Ψ_i in tree form, (c) $\partial\text{supp}(\Psi_i)$, (d) $M(\partial\text{supp}(\Psi_i))$, (e) New tree structure, initialized around MBR$[M(\partial\text{supp}(\Psi_i))]$ and refined at a seed position, (f) Ψ_f in tree form as the result of the flood-fill algorithm

7. Benchmarks

We implemented the presented numerical methods in a C++ code with the working name `PFTC2D`; a *P*erron-*F*robenius *T*ree *C*ode in two *D*imensions. The code is under develepement but first benchmark results shown in Figure 5 and 6 already prove the viability of the approach. The test case is the repeated application of a rotation map (one revolution per 100 steps) to an off-center Gaussian distribution. It can be seen that the integrated phase-space density is very well preserved even for low recursion depths. The loss rate is lower than 10^{-5} per 100 steps. Execution times for all 300 steps on an i5-6500T CPU range from $\approx 3\,s$ at $r = 5$ to ≈ 11 min at $r = 9$ [‡]. Note that with the time-discrete model introduced in section 2 a complete bunch-compressor stage can be simulated in 2 steps!

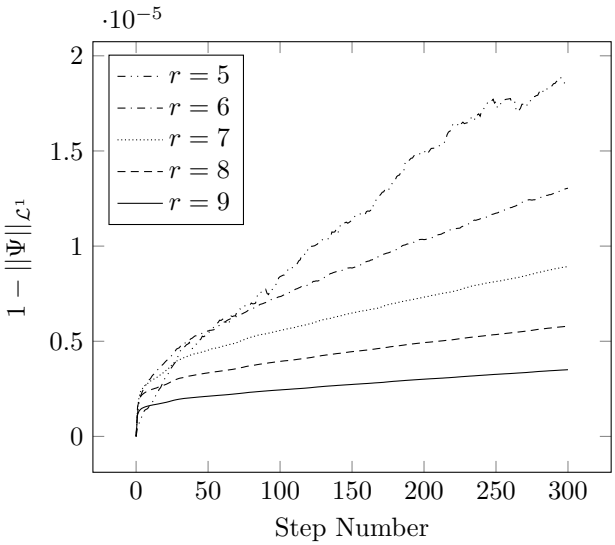

Fig. 5. Evolution of the absolute error of the integrated phase-space density for different recursion depths r.

8. Conclusion & Outlook

Quadtree-domain decomposition is a viable tool to approach the numerical challenges arising from the exotic phase-space structure of FEL bunches. We implemented this method in `PFTC2D`, which is currently under development. This code utilizes a time-discrete Vlasov model which is very well suited for investigating LSC driven microbunching in cascaded bunch-compressor setups.

[‡]Gaussians are hardly exotic so that these timings do not reflect the full potential of the code. For more exotic phase-space densities faster execution times at even higher recursion depths are to be expected.

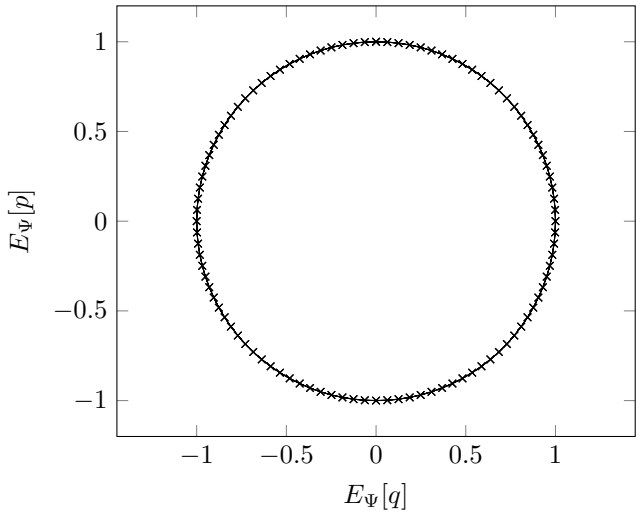

Fig. 6. Evolution of the centroid position at a recursion depth of 5. Note that this plot shows 3 revolutions.

It is noteworthy that the presented tree-based Perron-Frobenius simulation method is in no way restricted to time-discrete systems with one degree of freedom like the one we presented. In fact, the underlying data structures of `PFTC2D` were developed to support phase-spaces of arbitrary dimension. At higher dimensions, a tree of hypercubes is generated instead of a quadtree but all arguments made in sections 5 and 6 still hold or can be trivially extended to this case. Systems that require time-continouos treatment can be simulated by employing a suitable operator-splitting scheme. This, for instance, would be neccessary to include CSR effects in the model.

References

1. E.Saldin, E.Schneidmiller, M.Yurkov "Longitudinal Space Charge Driven Microbunching Instability in TTF2 Linac", *Proceedings of the ICFA Future Light Sources Sub-Panel Mini Workshop on Start-to-End Simulations of X-RAY FELs, Zeuthen* (2003), `http://www.desy.de/s2e/Talks/Monday/Talk_schneidmiller.pdf`.
2. M.Dohlus "Space Charge Instability in XFEL", *talk given at the DESY XFEL Beam Dynamics Seminar* (21.11.2005), `http://www.desy.de/xfel-beam/data /talks/talks/dohlus_-_sc_gain_500mev_20x5_20051121.pdf`.
3. M.Vogt, T.Limberg, D.Kuk "Simulation of Micro Bunching Instability Regimes", THPC111, *EPAC2008, Genoa, Italy* (2008).
4. M.Vogt, Ph.Amstutz "Arbitrary Order Perturbation Theory for Time-Discrete Vlasov Systems with Drift Maps and Poisson Type Collective Kick Maps ", *in this proceedings* (2017).
5. J.D.Jackson "Classical Electrodynamics; Third Edition", *Wiley* (1998)
6. M.de Berg et al. "Computational geometry: Algorithms and applications; 3rd ed.", *Springer* (2008)