Parallelized Vlasov-Fokker-Planck solver for desktop personal computers

Patrik Schönfeldt,* Miriam Brosi, Markus Schwarz,
Johannes L. Steinmann, and Anke-Susanne Müller

Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany

(Received 17 November 2016; published 14 March 2017)

The numerical solution of the Vlasov-Fokker-Planck equation is a well established method to simulate the dynamics, including the self-interaction with its own wake field, of an electron bunch in a storage ring. In this paper we present *Inovesa*, a modularly extensible program that uses OpenCL to massively parallelize the computation. It allows a standard desktop PC to work with appropriate accuracy and yield reliable results within minutes. We provide numerical stability-studies over a wide parameter range and compare our numerical findings to known results. Simulation results for the case of coherent synchrotron radiation will be compared to measurements that probe the effects of the microbunching instability occurring in the short bunch operation at ANKA. It will be shown that the impedance model based on the shielding effect of two parallel plates can not only describe the instability threshold, but also the presence of multiple regimes that show differences in the emission of coherent synchrotron radiation.

DOI: 10.1103/PhysRevAccelBeams.20.030704

I. INTRODUCTION

At synchrotron light sources electron bunches of a length of a few millimeters are used to produce coherent synchrotron radiation (CSR) in the terahertz (THz) frequency range. Due to the coherent emission, the intensity scales with the number of emitting particles squared, instead of linearly as for incoherent emission. In storage rings the spatial compression is achieved by using magnet optics with a small momentum compaction factor α_c. The compression leads to the microbunching instability. On the one hand, this instability limits the electron bunch charge that can be used in stable operation; on the other hand the emerging substructures emit coherent radiation also in a wavelength smaller than the electron bunch length.

First observations of microbunching in a storage ring as well as of the increase of coherent emission in the spectral range of interest were made at the NSLS VUV ring [1,2]. It then has been studied both experimentally, to map out the parameters governing the bursting behavior [3–10] and theoretically to predict thresholds [11–14] and to simulate the dynamics. To simulate the dynamics, it is possible to solve the Vlasov-Fokker-Planck equation for the longitudinal phase space density [15,16], or, using supercomputers, to do particle tracking with one million macroparticles or more [9,17].

Recent advances in detector development and readout electronics facilitate fast mapping of the microbunching instability [18] over a wide range of physical parameters. To cover these settings in simulation as well, it calls for an ultrafast simulation technique. Also, the simulation tool should be designed such that the influences of both the simulated physics and of numerical effects can be studied and separated. As we are interested in simulating an instability, in particular sources of numerical instabilities should be ruled out. In this paper we present *Inovesa* (*Inovesa Numerical Optimized Vlasov-Equation Solver Application*, available at [19]), a Vlasov-Fokker-Planck solver that runs on standard desktop PCs and yields robust results within minutes.

Section II summarifzes the theoretical description of the problem. The actual implementation is described in Sec. III, while Sec. IV presents numerical studies that show the robustness of the implementation and compares results of simulation and measurements.

II. LONGITUDINAL PHASE-SPACE DYNAMICS

A. Vlasov-Fokker-Planck equation

The phenomenon of microbunching happens in the longitudinal phase space, which is spanned by the position z relative to the synchronous particle and the energy E. Taking the particle density $\psi(z, E, t)$ of electrons in a storage ring to be a smooth function, its evolution with time t can be described by the Vlasov-Fokker-Planck equation (VFPE). Following the notation of [13] it reads

$$\frac{\partial\psi}{\partial\theta}+\frac{\partial H}{\partial p}\frac{\partial\psi}{\partial q}-\frac{\partial H}{\partial q}\frac{\partial\psi}{\partial p}=\beta\frac{\partial}{\partial p}\left(p\psi+\frac{\partial\psi}{\partial p}\right), \quad (1)$$

with the time given in multiples of synchrotron periods $\theta = f_s t$, the normalized coordinates $q = z/\sigma_{z,0}$, and $p = (E - E_0)/\sigma_{E,0}$, the Hamiltonian H, the reference

*patrik.schoenfeldt@kit.edu

particle's energy E_0, and $\beta = 1/(f_s\tau_d)$, where τ_d is the longitudinal damping time. The quantities $\sigma_{E,0}$ and $\sigma_{z,0}$ describe, respectively, energy spread and bunch length in the equilibrium state that exists for small bunch charges. It is convenient to also define a normalized current. It can be expressed as

$$\xi = \frac{Q_b f_{\mathrm{rev}}}{\alpha_c \gamma (\sigma_{E,0}/E_0)^2 I_A}\left(\frac{R}{\sigma_{z,0}}\right)^{1/3}, \tag{2}$$

where Q_b is the bunch charge, f_{rev} the revolution frequency, α_c the momentum compaction factor, γ the Lorentz energy factor, $I_A = 17045$ A the Alfvén current, and R the beam path's radius.

To solve this partial differential equation there exists a formalism that was developed in plasma physics [20] and first applied to particle accelerators by Warnock and Ellison [21]. It uses a grid to discretize $\psi(q,p)$ and assumes that the collective force due to self-interaction with the bunch's own coherent synchrotron radiation is constant for small time steps. The perturbation due to the collective effects is described as a perturbation to the Hamiltonian

$$\begin{aligned} H(q,p,t) &= \underbrace{H_e(q,p,t)}_{\text{external fields}} + \underbrace{H_c(q,t)}_{\text{collective effects}} \\ &= \frac{1}{2}(q^2+p^2) + \int_q^\infty Q_c \times V_c(Z_c, q', t)\mathrm{d}q', \end{aligned} \tag{3}$$

where Q_c is the charge involved in the perturbation, and V_c is the potential due to the collective effect, which can be expressed in terms of an impedance Z_c.

It is then possible to use the homogeneous solution, which in the unperturbed case is represented by a rotation in phase space, and add the influence of diffusion and damping as a particular solution. To model the perturbation, the influence of V_c is implemented as a "kick" along the energy axis.

B. Microbunching instability

To calculate the effect of the perturbation term introduced in Eq. (3), one needs the electric field $E(q,s)$ at the longitudinal position s. It is convenient to express it via a wake potential

$$V(q) = \int E(q,s)\mathrm{d}s, \tag{4}$$

which directly gives the energy difference for the electrons (in eV). The wake potential can be obtained from the wake function $W(q)$, which describes the field produced by one single particle. The wake potential $V(q)$ then is obtained by convolving it with the charge density ϱ [21]

$$V(q) = \int_{-\infty}^{\infty} W(q-q')\varrho(q')\mathrm{d}q'. \tag{5}$$

As in frequency space closed and smooth expressions exist for many commonly used impedances, we decided to work in frequency space. Then the wake potential can be deduced directly from the impedance $Z(k)$ in every time step using

$$V(q) = \int_{-\infty}^{\infty} Z(k)\tilde{\varrho}(k)e^{ikq}\mathrm{d}k, \tag{6}$$

where $\tilde{\varrho}(k)$ is the Fourier transform of the bunch profile.

This method allows us to implement different impedance models for *Inovesa* in just a few lines of code. As we are mostly interested in CSR-driven dynamics, we currently implemented two cases. Both will be given for $n \geq 0$, where $n = f/f_{\mathrm{rev}}$ is frequency expressed in multiples of the revolution frequency f_{rev}. For $n < 0$ they are continued $Z(n) = Z^*(-n)$, where the star marks the complex conjugate.

The first, the free space CSR impedance, describes the effect of coherent synchrotron radiation of particles traveling on a curved path in vacuum. A good approximation for the CSR impedance in a perfect circle is [22]

$$Z(n) \approx Z_0 \frac{\Gamma(2/3)}{3^{1/3}}\left(\frac{\sqrt{3}}{2} + \frac{i}{2}\right)n^{1/3}, \tag{7}$$

where Z_0 is the vacuum impedance.

The second implemented impedance approximates the shielding effect of the beam pipe by two parallel plates with distance g [22]. It can be approximated with the Airy functions Ai and Bi [14]:

$$\begin{aligned} Z(n, R/g) \approx{}& \frac{4\pi^2 2^{1/3}}{\epsilon_0 c}\left(\frac{R}{g}\right)n^{-1/3} \\ &\times \sum_p \mathrm{Ai}'(u_p)\mathrm{Ci}'(u_p) + u_p \mathrm{Ai}(u_p)\mathrm{Ci}(u_p), \end{aligned} \tag{8}$$

where the prime marks the first derivative with respect to the argument, R is the beam path's radius, $\mathrm{Ci} := \mathrm{Ai} - j\mathrm{Bi}$, and

$$u_p := \frac{\pi^2(2p+1)^2}{2^{2/3}}\left(\frac{R}{g}\right)^2 n^{-4/3}. \tag{9}$$

This more complex impedance has already proven to describe the microbunching instability threshold within the uncertainty of the measurements [18]. For the limit $g \to \infty$ it converges to the free space impedance. Figure 1 shows these two impedances, for the case of an accelerator with $f_{\mathrm{rev}} = 8.582$ GHz, and (for the shielded case) $g = 32$ mm.

III. IMPLEMENTATION

A. Discretization

Following the approach of Warnock and Ellison [21], the VFPE is discretized on a grid to be solved numerically. For

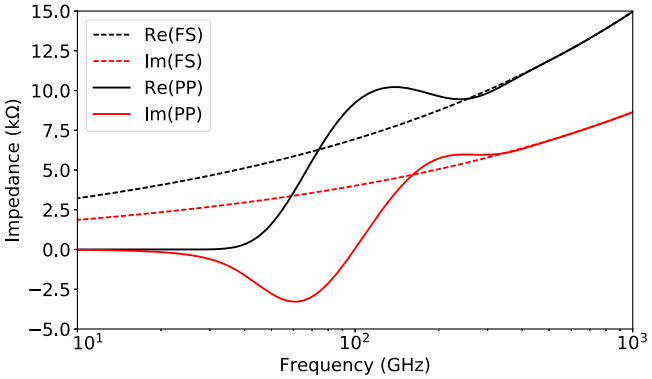

FIG. 1. Unshielded (free space) CSR impedance (FS) and CSR impedance shielded by parallel plates (PP), calculated for the case of an accelerator with $f_{\mathrm{rev}} = 8.582$ GHz, and (for the shielded case) $g = 32$ mm. For high frequencies both impedances converge, as short wavelength are not effected by the shielding. For $f \to 0$ Hz both impedances approach $Z = 0\ \Omega$.

Inovesa, we define a grid starting from a minimum value that can be expressed for each dimension. For $q > q_{\mathrm{min}}$ and $p > p_{\mathrm{min}}$, the generalized coordinates $q, p \in \mathbb{R}$ become the grid coordinates $x_r, y_r \in \mathbb{R}_{\geq 0}$. With Δp, the granularity of the grid in energy direction, the transformation between the coordinate systems can be expressed as

$$y_r(p) = (p - p_{\mathrm{min}})/\Delta p, \tag{10}$$

and accordingly for $x_r(q)$. Function values are only stored at integer coordinates m, where m refers to either x or y. When a function value at an arbitrary noninteger coordinate m_r is needed, *Inovesa* approximates it by interpolation using

$$f(m_r) \approx \vec{f}_N(m) \cdot \vec{p}_N(\{m\}) =: P_N(m_r) \tag{11}$$

where $\{m\} = m_r - m$ denotes the fractional part of m_r. The interpolation multiplies the vector of the function values at the N surrounding mesh points

$$\vec{f}_N(m) = (f(m - \lfloor (N-1)/2 \rfloor), \ldots, f(m), \ldots, \times f(m + \lceil (N-1)/2 \rceil))^T \tag{12}$$

with a vector containing N interpolation coefficients

$$\vec{p}_N(\{m\}) = (l_{0,N}(\{m\}), \ldots, l_{N-1,N}(\{m\}))^T, \tag{13}$$

where $l_{\nu,N}(\{m\})$ are the Lagrange basis polynomials [23]

$$l_{\nu,N}(\{m\}) := \prod_{k=0 \neq \nu}^{(N-1)} \frac{\{m\} - m_k}{m_\nu - m_k}. \tag{14}$$

We have implemented this for up to $N = 4$, which results in a cubic interpolation scheme. Interpolation using these polynomials sometimes overshoots the values of the neighboring grid cells. Section IV B will show that these numerical artifacts can even self-amplify and become a very dominant feature in the simulation. To avoid overshooting, one has to use clamped or saturated interpolation functions. A simple, clamped version of Eq. (11) reads

$$\mathrm{clamp}(f(m_r)) = \max\{\min[\min(f_N(m), f_N(m+1)), \times \vec{f}_N(m) \cdot \vec{p}_N(\{m\})], \times \max(f_N(m), f_N(m+1))\}. \tag{15}$$

Keeping in mind that not only $q, p \in \mathbb{R}$ is discretized to $x, y \in \mathbb{N}$ but also $\psi(x, y)$ is discretized in the computer's memory, we analyzed the effect of this second discretization. To be able to change the accuracy in small steps, we used a fixed point representation [24] where the number of bits for the fractional part could be chosen freely. The result of this test will be shown in the convergence studies (Sec. IV A).

B. Simulation steps

Each time step $f\colon \psi_t(q, p) \to \psi_{t+\Delta t}(q, p)$ is composed of a number of simulation steps that model rotation, kick, damping and diffusion $f = f_{\mathrm{rot}} \circ f_{\mathrm{kick}} \circ f_{\mathrm{damp,diff}}$. *Inovesa* splits the direct implementation of each of these simulation steps into two sub-steps. The information on the actual coordinates (q, p) is used by the first half simulation step. We call its result a "source map" (SM). Then, in the second half step only the grid coordinates x, y are used. Further we define $z = N_y \times x + y$ with the number of grid cells in energy direction N_y. Doing so, the map $f\colon \psi_t(\mathbb{N}^2) \to \psi_{t+\Delta t}(\mathbb{N}^2)$ can be rewritten in a one-dimensional form $f_{\mathrm{SM}}\colon \psi_t(\mathbb{N}^1) \to \psi_{t+\Delta t}(\mathbb{N}^1)$, which depends only on z.

This method—by construction—produces the same results as the single-step implementation. Practically speaking, the source map expresses the information which data of the current simulation step contributes to a grid point for the next simulation step directly in terms of position in the computer's memory. For many functions—such as rotation—the SM will look the same for the whole runtime of the program. For that reason, it only has to be computed once during a simulation run and can be kept for multiple usages.

The source map formalism does not only allow us to keep intermediate results, it also gives a handy interface to implement arbitrary functions on the phase space. Furthermore, the reduction of the problem's dimension also leads to a speedup of the calculations. Results of a benchmark of the computational performance for the particular case of rotation are shown in Fig. 2. In this case source mapping halves the runtime. Parallelization using OpenCL [25] allows further speedup. One advantage of OpenCL is that it can utilize not only multi-core CPUs but also graphic processors. OpenCL advocates a data parallel approach that processes multiple data with one single instruction. So in

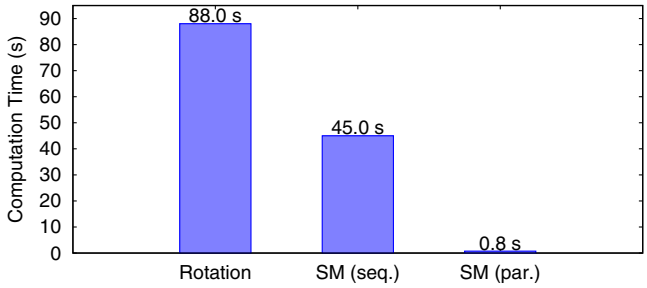

FIG. 2. Computational time needed by an Intel Core i5 4258U for 1000 cubically interpolated rotation steps using different implementations. In this test case, the grid has 512 points per axis, no optimizations (besides SM) were used. The first bar represents the computational time a direct implementation of the rotation takes. The second and third bar show the time the implementation using the source map formalism take—running sequentially on the CPU or parallel on the integrated graphics processor.

contrast to other forms of parallelization, there is no need for parallel tasks. In total, a nonoptimized program takes days for a typical simulation run. Using the method described above, *Inovesa* can reduce this to 15 minutes when running on a customer grade graphics card.

Using the SM formalism, we implemented different versions of the simulation steps necessary to solve Eq. (1). For the rotation, we provide a direct implementation (as in [21], "standard rotation"). Additionally, we use a symplectic integrator [26] to implement the rotation. This method offers two advantages: First, since the method is symplectic, it is automatically area preserving—also for truncated power series. Using the map from an infinitesimal rotation given by a direct implementation, symplecticity is easily lost in the numeric treatment. Second, the symplectic method provides additional numerical stability also for the interpolation. The standard rotation algorithm requires a two-dimensional interpolation, which involves $N \times N$ data points, leading to interpolation coefficients $l^2_{\nu,N} \propto \{x\}^{N-1} \times \{y\}^{N-1} \ll 1$. The symplectic map, in contrast, splits the rotation into an energy-dependent drift followed by a location dependent rf kick. Each requires a one-dimensional interpolation, involving N data points. The resulting coefficients are $l^1_{\nu,N} \propto \{m\}^{N-1} \gg l^2_{\nu,N}$, minimizing the vulnerability to numerical absorption. Since a point is rotated by moving on straight lines in perpendicular directions, in the following the symplectic approach will be referred to as "Manhattan rotation."

The damping and diffusion terms [right-hand side of Eq. (1)] need numerical differentiation. We found that the same type of artifacts that we found for the interpolation (see Sec. IV B) can also occur because of the differentiation. This can be explained by the fact that the algorithm usually used for numerical differentiation [23] is equivalent to differentiating the quadratic interpolation polynomial P_3 [see Eq. (11)]

$$\left.\frac{\partial f(x)}{\partial x}\right|_{x_0} \approx \frac{f(x_0+\Delta x)-f(x_0-1)}{2\Delta x} = \left.\frac{\partial P_3(x)}{\partial x}\right|_{x_0},$$

where the distance between the sampling points in our case is $\Delta x = 1$. As a consequence, we target this by using the cubic interpolation polynomial P_4, and obtain

$$\left.\frac{\partial f(x)}{\partial x}\right|_{x_0} \approx \frac{-2f(x_0-\Delta x)-3f(x_0)}{6\Delta x} + \frac{6f(x_0+\Delta x)-f(x_0+2\Delta x)}{6\Delta x} = \left.\frac{\partial P_4(x)}{\partial x}\right|_{x_0}. \tag{16}$$

Aside from this improvement, we proceed analogously to [21].

To implement the perturbation via a kick, we just had to translate the wake potential [see Eq. (6)] to the grid coordinate system using Eq. (10). Furthermore, our implementation uses the fact that both $Z(k)$ and $\tilde{\varrho}(k)$ are Hermitian. This means that optimized algorithms like the ones from Refs. [27,28] only need explicit function values for $k \geq 0$ to perform the inverse Fourier transform. Using this symmetry also brings an improvement in both speed and memory usage by roughly a factor of two [27].

IV. RESULTS

A. Convergence studies

In this section we compare the effect of different numerical settings, which ideally should not affect the physical result. We also compare different implementations of the rotation as described in Sec. III B. For the sake of simplicity, we go to the unperturbed case [meaning $H_c = 0$ in Eq. (3)]. So any starting distribution should exponentially converge to a Gaussian distribution with $\sigma_q = \sigma_p = 1$.

At first, we investigate the effect on the results of using different data types. To do so, we observe the evolution of a Gaussian distributed charge density with $\sigma_q = \sigma_p = 1.8$, sampled up to $q = p = \pm 6$, when the damping time is set to five synchrotron periods ($\beta = 0.2$). We start by reading the distribution from a 16-bit grayscale PNG. This brings initial quantization noise, but also provides well defined starting conditions for every data type: Initial rounding errors will be the same in the different runs. Figure 3 shows the different simulation results. We find that the results for the natural energy spread obtained using fixed point representations with a fractional part of at least 44 bits, and the tested floating point representations (`binary32` and `binary64` [29], often referred to as "single precision" and "double precision") show a common difference from the analytic result of about 10^{-4}. The relative differences between the data types are less than 10^{-6} and therefore can be neglected. As most libraries are developed focusing on floating point data types, we implemented the calculation of the wake potential only

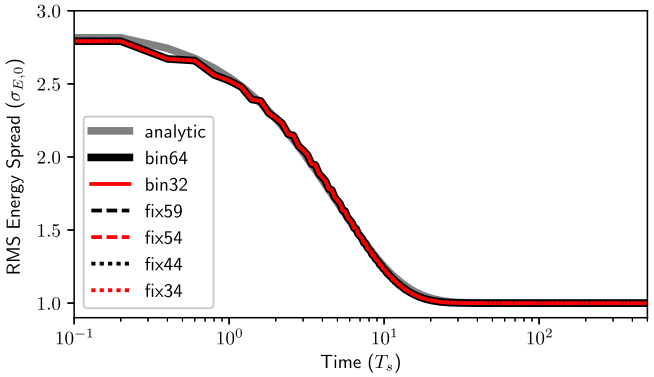

FIG. 3. Exponential damping of the energy spread σ_E as a function of time for different data types. There is an initial jitter due to quantization noise in the test pattern that has been read in from a 16 bit *PNG* file and due to the fact that the initial distribution is not fully covered by the grid. Even with this problematic starting conditions, after $T = 60\mathrm{T_s}$, all data types have converged to $\sigma_p \approx 1$. Note that all simulation curves are almost perfectly overlapping, there is no noticeable difference coming from the used data type.

for those types. In the following, we default to `binary32` as it is much faster than `binary64`.

To investigate the influence of the grid and the time steps, we observe the evolution of a Gaussian distributed charge density with $\sigma_q{=}\sigma_p{=}1.44$, again sampled up to $q = p = \pm 6$. For this we set the damping to $\beta = 0.02$. In a second set of simulation runs it is changed to $\beta = 0.002$. The number of simulation steps (ΔT) per T_s is varied between 500 and 4000, the number of grid cells between 64 and 512. For every possible combination standard and Manhattan rotation are used. The reconstructed values for damping time $\tau_{d,s}$ and energy spread $\sigma_{E,0,s}$ are obtained by a fit to the resulting curves $\sigma_{E,s}(t) = (\sigma_{E,\max,s} - \sigma_{E,0,s}) \times \exp(-2t/\tau_{d,s}) + \sigma_{E,0,s}$. The results of the different scans are shown in Fig. 4. All settings reproduce the exponential damping, nevertheless slower damping is more challenging to simulate as can be seen in the broader fluctuations in the reconstructed damping time and energy spread for the runs with $\beta = 0.002$. One contribution is that errors by the rotation algorithm have more influence for slower damping, but also the damping itself becomes numerically inaccurate at some point. It can also be seen that different combinations of the relative time step Δt, the damping factor β, the rotation algorithm, and the grid size may converge to different values. For example with the slower damping ($\tau_d = 500\mathrm{T_s}$), a grid of 256×256, and 1000 time steps per synchrotron period, Manhattan rotation yields $\tau_{d,s} = 498\mathrm{T_s}$ and $\sigma_{E,0,s} = 1.001\sigma_{\mathrm{E},0}$, while it is $\tau_{d,s} = 494\mathrm{T_s}$ and $\sigma_{E,0,s} = 0.984\sigma_{\mathrm{E},0}$ for standard rotation. The tendency that fewer but bigger time steps are less prone to numerical problems is due to the fact that the errors accumulate.

In summary, it can be concluded that grid size and time step should not be too small for good convergence. As the second requirement is contrary to the one for the approximation of the wake force [21], $\varrho(t) \approx \varrho(t + \Delta t)$, it is very valuable that the numerical error can be significantly reduced using Manhattan rotation—especially for slow damping. Independently from the used starting conditions, we found that the relative error can be reliably kept below 1%—provided that β does not become too small.

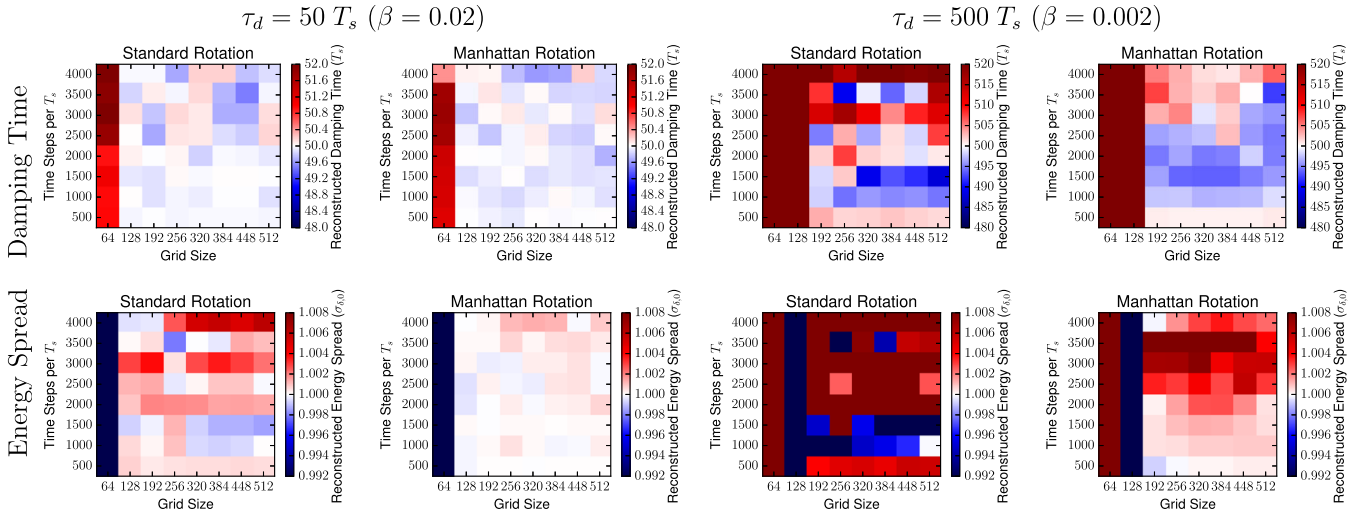

FIG. 4. Results of simulation runs with different grid sizes and numbers of time steps to simulate one synchrotron period. Every run is represented by two tiles: The reconstructed damping time ($\tau_{d,s}$) is shown in the upper row, the natural energy spread ($\sigma_{E,0,s}$) can be found in the lower row. For the left two columns, the set damping time is $\tau_d = 50T_s$ ($\beta = 0.02$), for the others, it is $\tau_d = 500T_s$ ($\beta = 0.002$). The value corresponding to the analytic result is white, higher values are shown in red, lower ones in blue. It can be seen that slow damping is numerically more challenging—it requires bigger time steps (less steps per T_s) for good convergence. Independently from the numerical settings, Manhattan rotation shows to be more robust in reconstructing the set values.

TABLE I. Parameters for an example run to check for numerical artifacts. For the given set no artifacts were observed. Changing to quadratic interpolation however, triggered the occurrence of (nonphysical) structures with a period length of two grid cells (see Fig. 5).

Parameter	Value
Grid points per axis	256
Steps per T_s	4000
Rotation method	Manhattan
Interpolation method	Cubic
Impedance model	Free space
Normalized current (ξ)	0.516
Damping time	$200T_s$

B. Numerical artifacts

Besides the numerical inaccuracies discussed above there are also numerical artifacts. For those we identify two main sources: interpolation and numerical differentiation. As an example, we do two simulation runs with the same current distribution, one run using cubic interpolation, the second using quadratic interpolation. The complete set of simulation parameters is listed in Table I. For the simulation run that uses cubic interpolation, the distribution stays in a relatively calm state with just little oscillation. Note that this is not equilibrium: As expected for the unshielded CSR case and a normalized current $\xi > 0.5$, we have $\sigma_E > \sigma_{E,0}$ [12] and an oscillation with $f \approx 2f_s$. If there was no influence of the different interpolation schemes, one would expect no difference between the two runs. However, as Fig. 5 depicts, this is not guaranteed.

The energy spread simulated using quadratic interpolation rapidly increases to a higher value after one synchrotron period ($T = 1T_s$). On a longer time scale, it will damp down again, and after that a new numerical instability might rise. If the aim is to find out whether the simulated conditions are above the microbunching threshold these artifacts might not matter—below the threshold the initial, numerical modulation should not be amplified. However, we want to track the evolution of the charge distribution, and this numerical artifact might be interpreted as an unphysical slow bursting frequency. So we have to avoid numerical artifacts as they occur in the run using quadratic interpolation (cf. rhs. of Fig. 5): The period length of the ripples is exactly two grid cells, and they continue to exist even to a position where they create negative charge densities. In this particular case, the instability is clearly triggered by numerical artifacts (overshoots) of the interpolation.

Note that also higher order polynomials show overshoots and that also numerical differentiation can be expressed in terms of interpolation polynomials. In our tests, however, we did not find any case where a more complex differentiation method than the one described in Eq. (16) was needed to avoid artifacts. In contrast to the differentiation, there were rare cases where we observed interpolation artifacts even when using cubic interpolation polynomials. In principle those artifacts can be circumvented by the choice of a different combination of numerical parameters, e.g. a different grid size or a different time step. However, this would require manual checks for artifacts to make sure a combination of parameters remains stable for the specific physical conditions, e.g. an increased grid size would in principle be prone to the same artifact with a (in physical units) different period length. As an alternative those numerical artifacts can be completely suppressed by clamping [Eq. (15)].

C. Comparison with measurements

For our comparison with measurements results, we scale the quantities that are a function of the revolution frequency by a factor of $2\pi R/C$ where R is the bending radius and C is the circumference of ANKA. This way measurements

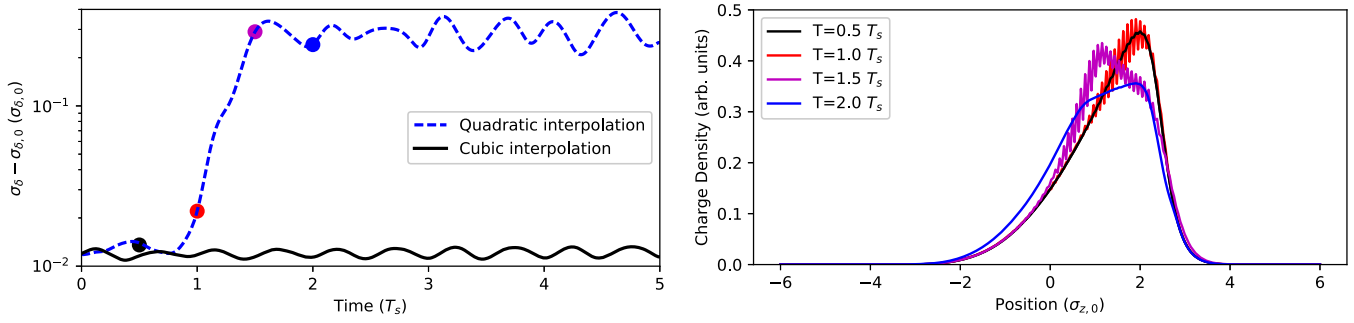

FIG. 5. Evolution of the energy spread over time (left) and bunch profiles of the simulation run using quadratic interpolation (right) at the points in time marked by the disks. For the case where cubic interpolation is used (left, solid black line), the energy spread stays at $\sigma_E \approx 1.01\sigma_{E,0}$. When using quadratic interpolation (left, dashed blue line), an increase in energy spread can be observed at $T = 1T_s$. The right-hand side reveals that this increase is a numerical artifact. The bunch profiles computed during the initial increase of the energy spread show large ripples with a period length of two grid cells. The earlier and later profiles do not show such structures. This implies that the increase of energy spread is driven by a numerical instability.

TABLE II. Parameters of an isomagnetic accelerator comparable to ANKA (t_d, h, f_s, and f_{rev} are scaled by $2\pi R/C = 0.316$).

Parameter	Symbol	Value
Beam energy	E_0	1.285 GeV
Energy spread	$\sigma_{E,0}$	0.47×10^{-3}
Damping time	t_d	3.353 ms
Harmonic number	h	58.21
Rf voltage	V_{rf}	1048 kV
Revolution frequency	f_{rev}	8.582 MHz
Synchrotron frequency	f_s	28.13 kHz
Vacuum chamber height	g	32 mm

are comparable to the simulations—which assume an isomagnetic ring with the same bending radius. The parameters used here are listed in Table II. Note that this is just one set of possible parameters because the magnet optics (and thus f_s) as well as the rf voltage V_{rf} can be gradually changed at ANKA.

There are some effects we neglect for the simulation such as the frequency response of the used detector, and the coherent tune shift observed in the measurement. Also, contributions from other impedances than shielded CSR (e.g. geometric impedance) are not studied.

We do separate simulation runs for about 150 different currents between $I = 1.3$ mA and $I = 0.5$ mA. For the first one, we start with the highest current and a Gaussian charge distribution that is significantly broader than the expected distribution. To compensate for this, we allow some extra time for convergence. For the following simulation runs, we take the final charge distribution of the run before that has the (slightly) higher current as starting parameters. (Different approaches to create starting distributions are discussed in the Appendix A.) For each of these runs, the simulation time on a AMD Radeon R9 290 graphics card is a bit more then ten minutes, which makes a total simulation time of about 19 hours.

The spectrum of the emitted CSR is calculated using

$$P(t,k) \propto \Re(Z_k) \times |\tilde{\rho}(k,t)|^2, \tag{17}$$

where $\Re(Z_k)$ is the real part of the impedance and $|\tilde{\rho}(k,t)|^2$ the form factor of the bunch profile. It is then integrated to obtain the power a detector would measure

$$P(t) \propto \int P(t,k)\mathrm{d}k. \tag{18}$$

In analogy to what is done for the measured data, the resulting signal over time is Fourier transformed to obtain a spectrogram of the "bursting" frequencies. Figure 6 shows this spectrogram of $P(t)$. The general structure of the simulation and the measurement results agree very well: The instability threshold is marked by the occurrence of an isolated finger pointing down to $I \approx 0.21$ mA. For slightly higher currents, the finger broadens and fluctuations in the lower frequency range ($f < 10$ kHz) start. A third regime is observed at the highest currents. It shows parallel frequency lines that stay approximately constant with changing current. There are slight mismatches, e.g. in the threshold currents and in the frequencies, but most features are well reproduced by the simulation.

This means that for ANKA not only the thresholds (see also [18]) but also the dynamics of the microbunching instability are governed by an impedance that can be approximated by the parallel plates CSR impedance. To explain the details, a more complex model will be needed. Two possibilities are to take into account higher orders of the momentum compaction factor α_c or additional impedance contributions.

V. SUMMARY

We introduced *Inovesa*, a Vlasov-Fokker-Planck-solver that uses a runtime-optimized implementation of the computation steps. Utilizing OpenCL for parallelized computation, it can simulate the dynamics of the longitudinal phase

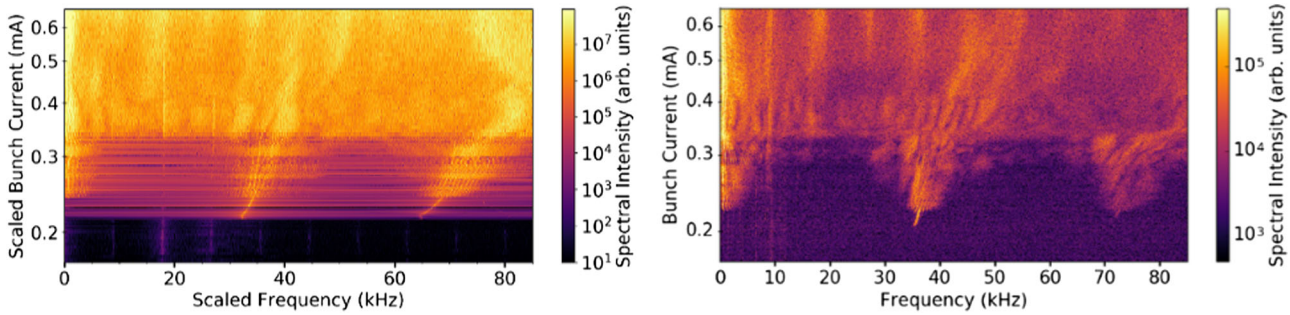

FIG. 6. Example for a simulated (left) and a measured (right) bursting spectrogram. For the simulated spectrogram the axis is scaled with a factor of $2\pi R/C$ to correct the mismatch due to the isomagnetic approximation. There are small differences, e.g. in the threshold current and in the frequencies. However, keeping in mind the very simple model, the general structure of the spectrograms matches quite well: There is an isolated finger pointing down (at $f \approx 3$ kHz); the fingertip ($I \approx 0.21$ mA) marks the instability threshold. For slightly higher currents, fluctuations in the lower frequency range ($f < 10$ kHz) start and the finger broadens. For the highest currents displayed here, there is a regime showing parallel frequency lines that stay approximately constant with changing current.

space more than a 150 times faster than a non-optimized implementation—using a dedicated (consumer-grade) graphics card. Furthermore, we eliminated sources of numerical artifacts and have done numerical stability studies to show that relative errors can usually be kept clearly below 1%.

Using *Inovesa* we were able to simulate the dynamics of the longitudinal phase space of ANKA in the regime of the microbunching instability. To do so, we used an impedance model that assumes CSR of electrons moving on a circular path shielded by parallel plates. Considering the simplicity of the model, the numerical results show an excellent agreement to the measurements.

ACKNOWLEDGMENTS

C. Evain is acknowledged for useful discussions, G. Stupakov for his suggestion to investigate numerical effects of different rotation schemes. We also thank M. Klein for giving us her code as a reference, and T. Boltz for beta testing. M. Brosi, P. Schönfeldt, and J. L. Steinmann want to acknowledge the support by the Helmholtz International Research School for Teratronics (HIRST). This work has been supported by the German Federal Ministry of Education and Research (Grants No. 05K13VKA, No. 05K16VKA), and the Helmholtz Association (Contract No. VH-NG-320).

APPENDIX A: STARTING DISTRIBUTION

In equilibrium, the energy is distributed according to a Gaussian function and the bunch profile is described by the Haïssinski distribution [30]. However, here we are interested in the dynamics of the microbunching instability above the threshold current, which means in nonequilibrium. In this physical state, there is no simple one-dimensional function for any possible charge distribution.

One possibility is to start the simulation with a Gaussian distribution that is broader than the expected charge distribution. It will damp down until the physical state of the instability is reached. Although for currents well above the instability threshold any possible starting distribution will reach the same state, we chose these initial conditions because it shows good convergence. When alternatively using narrower distributions unphysical structures form and might persist for a long time.

In Fig. 7 the evolution of the RMS energy spread over 500 synchrotron periods is shown for three different starting distributions at $I = 1.5$ mA. When using the final distribution of a previous run with slightly higher current (here $I = 1.54$ mA), the simulation converges quasi-instantaneously. As shown, the Haïssinski distribution is immediately blown up and becomes larger than the Gaussian distribution that has been set to be larger than the expected distribution. Also in the beginning ($t < 300\mathrm{T_s}$) the oscillation is systematically enlarged.

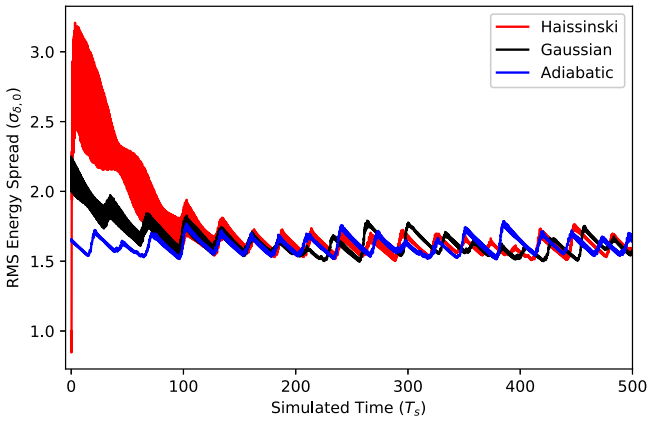

FIG. 7. Evolution of the RMS energy spread over a time of 500 synchrotron periods for $I = 1.5$ mA. The width of the line is caused by high frequency oscillations. For the starting distributions the Haïssinski distribution (equilibrium at $I = 675$ μA), a Gaussian distribution with $\sigma_p = \sigma_q = 2.25$, and the final distribution of the previous run ("adiabatic," $I = 1.54$ mA) were used. Note that the Haïssinski distribution is immediately blown up and becomes larger than the Gaussian distribution, and that the high frequency oscillation is systematically higher until $T \approx 300\mathrm{T_s}$.

APPENDIX B: RUNNING INOVESA

Inovesa is implemented as a non-interactive command-line tool. To automatically simulate all data for a spectrogram as shown in Fig. 6, a script may be used. Using the "adiabatic" method discussed in Appendix A, an example bash script reads:

```
1 #! /bin/bash
2 config="inovesa-run123.cfg"
3 lasti="500"
4 ./inovesa -I ${lasti}e-6 --config
  $config -T 1500 -o $lasti.h5
5 for curri in {500..100..5}
6 do
7 ./inovesa -I ${curri}e-6 -i
  $lasti.h5 -c $config -T 500 -o
  $curri.h5
8 lasti=$curri
9 done
```

The parameters used here are:

`--BunchCurrent` (or `-I`) for the ring current due to a single bunch given (in Ampere)

`--config` (or `-c`) the file name of a configuration file

`--rotations` (or `-T`) the total simulation time (in synchrotron period lengths T_s)

`--InitialDistFile` (or `-i`) the file name of an *Inovesa* result file to use for the initial particle distribution

`--output` (or `-o`) the file name to save results to

The configuration file used for the script contains all relevant parameters in a key-value-representation.

Comments may be added using "#". Here is an example configuration file:

```
BeamEnergy=1.3e+09 # in eV
BeamEnergySpread=0.00047 # relative
BendingRadius=5.559 # in m
BunchCurrent=1.2e-05 # in A
DampingTime=0.01 # in s
GridSize=256 # grid points per axis
HarmonicNumber=184 # f_RF/f_rev
PhaseSpaceSize=12 # sigma_z/E per axis
AcceleratingVoltage=1.4e+06 # in V
RevolutionFrequency=2.7e+06 # in Hz
alpha0=2e-4 # momentum compaction
VacuumGap=0.032 # full distance in m
gui=true # show live preview of results
outstep=50 # write output every N steps
padding=8 # factor for FFT zero-pading
```

All parameters are optional: If a parameter is not set *Inovesa* will fall back to default values. You might also overwrite settings from a configuration file by passing the same parameter as a command line argument. For short tests it is a good idea to enable the live preview (`-g true`) and not to save the results (by seting `-o /dev/null`).

[1] G. Carr, S. Kramer, J. Murphy, J. LaVeign, R. Lobo, D. Reitze, and D. Tanner, Investigation of coherent emission from the NSLS VUV ring, in *Proceedings of the 18th Particle Accelerator Conference, New York, 1999* (IEEE, New York, 1999), p. 134.

[2] G. Carr, S. Kramer, J. Murphy, R. Lobo, and D. Tanner, Observation of coherent synchrotron radiation from the NSLS VUV ring, Nucl. Instrum. Methods Phys. Res., Sect. A **463**, 387 (2001).

[3] J. M. Byrd, W. P. Leemans, A. Loftsdottir, B. Marcelis, M. C. Martin, W. R. McKinney, F. Sannibale, T. Scarvie, and C. Steier, Observation of Broadband Self-Amplified Spontaneous Coherent Terahertz Synchrotron Radiation in a Storage Ring, Phys. Rev. Lett. **89**, 224801 (2002).

[4] A.-S. Müller, Accelerator-based sources of infrared and terahertz radiation, Rev. Accel. Sci. Techol. **03**, 165 (2010).

[5] M. Abo-Bakr, J. Feikes, K. Holldack, P. Kuske, and G. Wustefeld, Coherent emission of synchrotron radiation and longitudinal instabilities, in *Proceedings of the 2003 Particle Accelerator Conference, Portland, OR* (IEEE, New York, 2003), p. 3023.

[6] W. Shields, R. Bartolini, G. Boorman, P. Karataev, A. Lyapin, J. Puntree, and G. Rehm, Microbunch instability observations from a THz detector at diamond light source, J. Phys. Conf. Ser. **357**, 012037 (2012).

[7] E. Karantzoulis, G. Penco, A. Perucchi, and S. Lupi, Characterization of coherent THz radiation bursting regime at Elettra, Infrared Phys. Technol. **53**, 300 (2010).

[8] G. Wüstefeld, J. Feikes, M. Hartrott, M. Ries, A. Hoehl, R. Klein, R. Müller, A. Serdyukov, and G. Ulm, Coherent THz measurements at the metrology light source, in *Proceedings of the International Particle Accelerator Conference, Kyoto, Japan* (ICR, Kyoto, 2010), p. 2508.

[9] C. Evain, J. Barros, A. Loulergue, M. A. Tordeux, R. Nagaoka, M. Labat, L. Cassinari, G. Creff, L. Manceron, J. B. Brubach, P. Roy, and M. E. Couprie, Spatio-temporal dynamics of relativistic electron bunches during the micro-bunching instability in storage rings, Europhys. Lett. **98**, 40006 (2012).

[10] Y. Takashima, M. Katoh, M. Hosaka, A. Mochihashi, S. ichi Kimura, and T. Takahashi, Observation of intense bursts of terahertz synchrotron radiation at UVSOR-II, Jpn. J. Appl. Phys. **44**, L1131 (2005).

[11] M. Venturini and R. Warnock, Bursts of Coherent Synchrotron Radiation in Electron Storage Rings: A Dynamical Model, Phys. Rev. Lett. **89**, 224802 (2002).

[12] G. Stupakov and S. Heifets, Beam instability and micro-bunching due to coherent synchrotron radiation, Phys. Rev. ST Accel. Beams **5**, 054402 (2002).

[13] K. L. F. Bane, Y. Cai, and G. Stupakov, Threshold studies of the microwave instability in electron storage rings, Phys. Rev. ST Accel. Beams **13**, 104402 (2010).

[14] Y. Cai, Theory of microwave instability and coherent synchrotron radiation in electron storage rings, in *Proceedings of the 2nd International Particle Accelerator Conference, San Sebastiáán, Spain* (EPS-AG, Spain, 2011), p. 3774.

[15] M. Venturini, R. Warnock, R. Ruth, and J. A. Ellison, Coherent synchrotron radiation and bunch stability in a compact storage ring, Phys. Rev. ST Accel. Beams **8**, 014202 (2005).

[16] E. Roussel, C. Evain, C. Szwaj, and S. Bielawski, Microbunching instability in storage rings: Link between phase-space structure and terahertz coherent synchrotron radiation radio-frequency spectra, Phys. Rev. ST Accel. Beams **17**, 010701 (2014).

[17] E. Roussel, C. Evain, C. Szwaj, S. Bielawski, J. Raasch, P. Thoma, A. Scheuring, M. Hofherr, K. Ilin, S. Wünsch, M. Siegel, M. Hosaka, N. Yamamoto, Y. Takashima, H. Zen, T. Konomi, M. Adachi, S. Kimura, and M. Katoh, Microbunching Instability in Relativistic Electron Bunches: Direct Observations of the Microstructures Using Ultrafast YBCO Detectors, Phys. Rev. Lett. **113**, 094801 (2014).

[18] M. Brosi, J. L. Steinmann, E. Blomley, E. Bründermann, M. Caselle, N. Hiller, B. Kehrer, Y.-L. Mathis, M. J. Nasse, L. Rota, M. Schedler, P. Schönfeldt, M. Schuh, M. Schwarz, M. Weber, and A.-S. Müller, Fast mapping of terahertz bursting thresholds and characteristics at synchrotron light sources, Phys. Rev. Accel. Beams **19**, 110701 (2016).

[19] Inovesa source code, https://github.com/Inovesa/Inovesa (2016).

[20] C. Z. Cheng and G. Knorr, The integration of the vlasov equation in configuration space, J. Comput. Phys. **22**, 330 (1976).

[21] R. L. Warnock and J. A. Ellison, SLAC Technical Report No. SLAC-PUB-8404, 2000.
[22] J. Murphy, S. Krinsky, and R. Gluckstern, Longitudinal wakefield for an electron moving on a circular orbit, Part. Accel. **57**, 9 (1997).
[23] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd ed. (Cambridge University Press, Cambridge, England, 2007).
[24] Fixed point class documentation, http://www.codeproject.com/Articles/37636/Fixed-Point-Class (2009).
[25] OpenCL website, https://www.khronos.org/opencl/ (2016).
[26] A. Wolski, *Beam Dynamics in High Energy Particle Accelerators* (Imperial College Press, London, United Kingdom, 2014).
[27] FFTW manual, http://www.fftw.org/fftw2_doc/fftw_2.html (2016).
[28] clFTW manual, http://clmathlibraries.github.io/clFFT/ (2016).
[29] IEEE Std. 754-2008 (2008).
[30] J. Haissinski, Exact longitudinal equilibrium distribution of stored electrons in the presence of self-fields, Il Nuovo Cimento **18B**, 72 (1973).