FERMILAB-PUB-08-236-APC

Space Charge Templates for High-Current Beam Modeling

Leonid G. Vorobiev

APC, Fermilab, P.O.Box 500, Batavia, IL 60510

Abstract

A computational method to evaluate space charge potential and gradients of charged particle beam in the presence of conducting boundaries, has been introduced. The three-dimensional (3D) field of the beam can be derived as a convolution of macro Green's functions (template fields), satisfying the same boundary conditions, as the original beam. Numerical experiments gave a confidence that space charge effects can be modeled by templates with enough accuracy and generality within dramatically faster computational times than standard combination: a grid density + Poisson solvers, realized in the most of Particle in Cell codes. The achieved rapidity may significantly broaden the high-current beam design space, making the optimization in automatic mode possible, which so far was only feasible for simplest self-field formulations such as rms envelope equations. The template technique may be used as a standalone program, or as an optional field solver in existing beam dynamics codes both in one-passage structures and in rings.

I. SPACE CHARGE HIERARCHY: KV, PICS AND IN BETWEEN

The domain of computational approaches for high-current beam dynamics simulation has two very different formulations in terms of generality and performance. The rms envelope equations (aka KV-equations), derived by Kapchinsky and Vladimirsky [1], generalized further by Sacherer, Lapostolle and other authors [2, 3] may be considered as the least general space charge model. In it's turn particle-in-cell codes may be named as the most general space charge formulation. Both envelope and PIC methods are self-consistent and can be either two-dimensional (2D) or three-dimensional (3D) [4–6]. There is a large gap between these two and the rest of computational methods for intense beams dynamics fill that gap.

Elegant envelope equations deal with beam envelopes, which are governed by ordinary differential equations. The results of the integration are fast and accurate. However envelope model is valid only for linear beam field and linear external focusing in the absence of conducting boundaries. Moreover the rms emittances must be always stationary. Nevertheless, even this limited linear approximation may be very helpful for preliminary designs, when a rapidity of envelope model allows to incorporate a space charge into optimization blocks. It works well for matching, final focusing design, searching an optimal focusing regimes automatically (see Section V below).

The methods using macro particles are for the most detailed beam simulation, including as much generality as possible. These codes deal with N_p particles, rather than with two or three envelopes, and integrate N_p differential equations simultaneously, evaluating the non-linear space-charge and external fields on each step of integration.

Several approaches based on macro particles are known. One of them finds space charge fields via the Green's function, as pair-wise forces between N_p particles in the ensemble.

$$\mathbf{X}_{old} \rightarrow \begin{bmatrix} \text{Green} \\ \text{Function} \end{bmatrix} \rightarrow \begin{matrix} \text{Trajectories} \\ \text{Integrator} \end{matrix} \rightarrow \mathbf{X}_{new} \qquad (1)$$

where $\mathbf{X} = \{x_1, x'_1, y_1, y'_1, z_1, z'_1, \quad \ldots \quad, x_{N_p}, x'_{N_p}, y_{N_p}, y'_{N_p}, z_{N_p}, z'_{N_p}\}$. All N_p trajectories (N_p ranges in millions) may be integrated by explicit 2nd order (as leap-frog) or higher-order integrator. For the cases of explicit integration (satisfying the energy/momentum conservation laws), a numerical solution of $6 \times N_p$ differential equations becomes even slower (see [7, 8]).

The advantage of this straightforward method (1) is that it is exact and easy to program.

Unfortunately the Green's function has an analytical form only for simple beam and boundary configurations [9] and even for those cases, the calculations require $O(N_p^2)$ operations on the each step of trajectories integration, that is too time-consuming even for modest N_p (typically $N_p \geq 10^5$). Other difficulties such as a numerical noise due to discretization and the singularities of short-range forces for particles with close coordinates [14] make this approach numerically limited.

A substantial development of the scheme (1) was done in [10, 11], when a bi-linear approximation of the Green's function in free space allowed manage singularities via analytical representation and ultimately to speed up the calculations.

Another type of particle codes, the general particle-in-cell (PIC) algorithm determines the 3D space charge forces, solving the finite-difference Poisson equation with specified boundary conditions on the spatial (e.g., the Cartesian) grid. Firstly a grid charge density ρ_{3D} is derived from the particle coordinates, e.g., by the standard cloud-in-cell (CIC) or PIC technique, which requires $O(N_p)$ operations and efficiently controls numerical noise [12, 14]. The beam potential $\mathbf{u}_{3D}$ and gradients on the grid may be found then by rapid Poisson Solvers [12, 13]. The self fields $\mathbf{E} = (E_x, E_y, E_z)$ between the grid nodes are interpolated from the grid fields and the trajectory integrator, taking into account both space charge and external forces, advances the particles.

The one step advance of 3D particle ensemble in PIC may be described schematically as:

$$\mathbf{X}_{old} \rightarrow \begin{bmatrix} \rho_{3D} \\ \downarrow \\ \mathbf{u}_{3D} \end{bmatrix} \rightarrow [\mathbf{E}] \rightarrow \begin{array}{c} \text{Trajectories} \\ \text{Integrator} \end{array} \rightarrow \mathbf{X}_{new} \tag{2}$$

The KV and PIC methods have their own design space, which is primarily aimed on one-passage structures, such as linacs, final focusing systems or matching sections, but they are hardly useful for beam simulation in accelerator and storage rings. The envelope equations are just too simple to study dynamics in rings, omitting resonances, non-linearity, etc. As for general PIC codes, they are too time-consuming.

In practice, for multiple-particle simulation in rings, one uses the 2-nd or higher order matrix/map formalism (1st order matrices are used only for simplest applications), implemented in COSY [15], DIMAD [16], MAD, MAD-X [17], ORBIT [18] programs, etc. (see [20]). The space charge models in these codes are either linear (as in KV model), or assume a pre-assigned charge density distributions, e.g., the Gaussian with ellipsoidal symmetry (COSY, MAD). There is no accurate feedback between the particles and the assigned density. These simplifications due to a necessity to simulate dynamics over thousands of turns, strictly speaking are not fully self-consistent.

More generally the space charge force is approximated by separate horizontal and vertical kicks from grid Poisson solvers, as in ORBIT code. Numerical experiments showed that for very long bunches it works. (The ORBIT was the primary computational tool for beam dynamics numerical studies in PSR/LANL and SNS/ORNL). For shorter bunches the 3D Laplace operator can not be split that way, therefore the separate kicks are invalid, resulting in error, as shown in [21] and misinterpretation of the longitudinal space-charge waves and instabilities [22, 23]. For those situations the PIC remains the only remedy, but again, thousand/million turns simulations are inappropriately slow for that.

Most of particle dynamics codes [20] use a massive parallelization, such as Synergia framework [19] with a suite of space charge solvers, and there are many other efforts to speed up the PIC scheme, improving their components, combining a PIC integrator in (2) with matrix formalism [24], using Hermite polynomials for charge density fast evaluation [25], etc., so that the above review may be incomplete.

The goal of this paper is to create a fast and accurate block for space charge calculations, which works in frame of PIC formalism and could be applicable for rings as well. Instead of the scheme (2) we obtain

$$\mathbf{X}_{old} \rightarrow \begin{bmatrix} \text{Template} \\ \text{Method} \end{bmatrix} \rightarrow \begin{matrix} \text{Trajectories} \\ \text{Integrator} \end{matrix} \rightarrow \mathbf{X}_{new} \tag{3}$$

The difference between (3) and (2) is only in the square brackets: there is no grid density and grid field solvers anymore [8, 27, 28]. The main source of computational economy of template algorithm is just in the absence of 3D charge density block and in using of a library of pre-calculated template potentials/gradients. Ultimately, the number of particles N_p involved in the model is orders of magnitude smaller, than that in standard PIC. The fields are derived from the particle coordinates by the so-called template method, which, in fact, stems from the Green's function formulation, with template potentials and field, as new macro Green's functions.

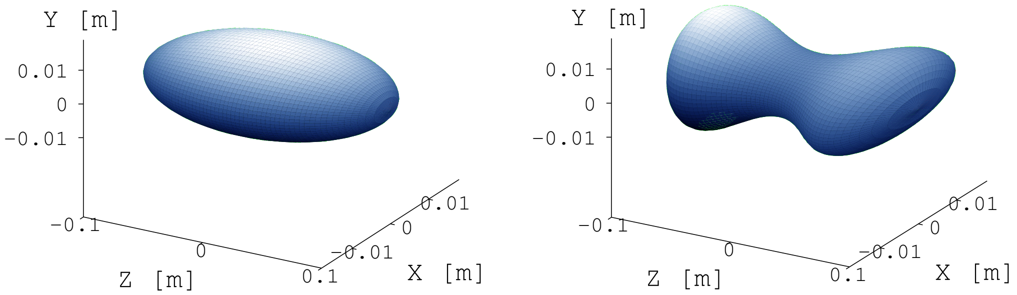

FIG. 1: Ellipsoid-like beam with semi-axes: $0.01 \times 0.01 \times 0.1[\mathrm{m}^3]$ and a beam with more arbitrary symmetry, both carrying a total charge of $Q_{total} = 10^{-11}$ C.

The template-based 3D space charge solver is more general than envelope formalism and on the other hand is dramatically faster than general solvers based on 3D grid density and Poisson solvers. The templates fields, being the macro Green's function, satisfy the same boundary conditions, as the whole beam and their convolution produces the resulting field of the original beam. The feedback between particle coordinates and density is more flexible than any other pre-assigned particle distribution.

Shown in Fig. 1 are possible 3D beams, varying from 3D ellipsoid to a rather arbitrary 3D beam configurations with the only requirement of elliptical transverse cross-sections. The conducting cylindrical boundaries (not shown) are also accommodated by the template algorithm.

Being less general than (2), the scheme (3) is still quite appropriate for a large class of particle distributions and boundary constraints.

II. TEMPLATE CONCEPT

Throughout this paper we use the word "template" and "a library of templates", which have nothing to do with the standard template library (STL), which is a feature of C++ language. Although a technical realization of our algorithm may use the STL, again, in this paper we mean that templates are macro geometrical objects, which approximate the original charged particle beam. They are elliptical disks or rings of different sizes and aspect ratios, as depicted in Fig. 2. Their thickness may vary from zero (infinitesimally thin slices) up to any positive value, when the templates look rather like lumps. The fewer templates

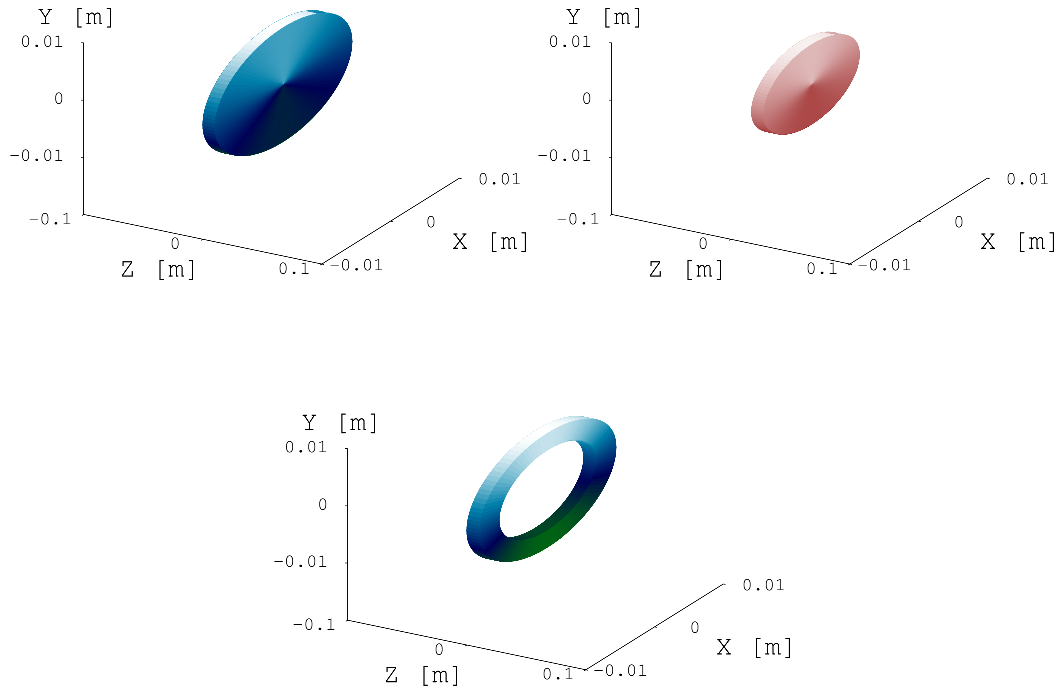

FIG. 2: Templates: discs and rings. A ring is a superposition of positively (blue) and negatively (red) charged discs.

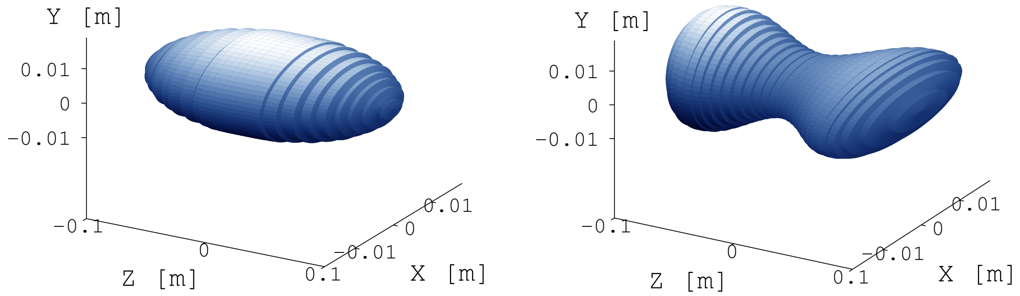

FIG. 3: Ellipsoid-like and an arbitrary beams from Fig. 1, represented as a sequence of templates.

are employed, the higher is the rapidity. However, to meet the accuracy requirements, both thick and thin slices may be used. Fig. 3 demonstrates a few different ways to represent the original beams via templates, depending on actual beam profiles.

The potential $u_{tmp}(\mathbf{x})$, where $\mathbf{x} = (x, y, z)$ of each template is called the template potential. Within a region $\mathcal{R}$ with a boundary $\partial\mathcal{R}$, u_{tmp} satisfies the Poisson equation:

$$\begin{cases} \Delta u_{tmp}(\mathbf{x}) = -4\pi\sigma_{tmp}(\mathbf{x}), \ \ \mathbf{x} \in \mathcal{R} \\ \\ u_{tmp}(\mathbf{x})|_{\partial\mathcal{R}} = \overline{U}_{tmp}(\mathbf{x}) \end{cases} \tag{4}$$

If the beam bunch allows a discrete representation via templates, then according to the principle of superposition, the total 3D beam potential u_{beam} can be found via a superposition of individual template potentials u_{tmp} as a convolution integral:

$$u_{beam}(\mathbf{x}) = \int u_{tmp}(x, y, z - \tilde{z}, S_{x,y}(\tilde{z}), \Lambda(\tilde{z}))d\tilde{z} \tag{5}$$

where the template potentials, correspond to the beam shape functions $S_{x,y}$ and are scaled by the longitudinal charge density Λ (see Section III C).

The computed $u_{beam}(\mathbf{x})$ will satisfy the Poisson equation (for an ideal conducting boundary, we have $\overline{U} = 0$):

$$\begin{cases} \Delta u_{beam}(\mathbf{x}) = -4\pi\rho_{beam}(\mathbf{x}), \ \ \mathbf{x} \in \mathcal{R} \\ \\ u_{beam}(\mathbf{x})|_{\partial\mathcal{R}} = \overline{U}_{beam}(\mathbf{x}) \end{cases} \tag{6}$$

Let us assume that the template potentials are known a priory as a library of pre-calculated data. Then we do not need to solve a series of the boundary problems (4) for individual templates. We just substitute ready-to-use u_{tmp} into (5). The resulting whole beam potential u_{beam} will automatically satisfy the boundary constraints. This is the essence of the template potential idea.

The following questions should be answered: what are the memory capacities required to store the template library, and how fast and general is this technique at all?

III. TEMPLATE CALCULUS

A. Template potential and fields in free space

The 3D potential in free space is

$$u_{tmp}(\mathbf{x}) = \iiint_{\mathcal{R}} \rho(\tilde{\mathbf{x}})d\tilde{\mathbf{x}}/|\mathbf{x} - \tilde{\mathbf{x}}| \tag{7}$$

for a particular case of a round ring of the outer radius R_{tmp}, and the inner radius $R_{tmp} - dR$ (dR is the thickness) with constant urface density σ_{tmp}, (7) can be rewritten for $\mathbf{x} = (0, 0, z)$ as $u^{free}_{tmp}(\mathbf{x}) = 2\pi\sigma_{tmp}(\sqrt{R^2_{tmp} + z^2} - \sqrt{(R_{tmp} - dR)^2 + z^2})$ (see [29, 31] for details).

Introducing notations $\tilde{R} = \sqrt{R^2_{tmp} + z^2}$, and $\hat{R} = \sqrt{(R_{tmp} - dR)^2 + z^2}$, the following expressions for the potential, the field $E^{free}_{tmp,z} = -\partial u/\partial z$ and it's derivative $E^{free}_{tmp,zz} = -\partial^2 u/\partial z^2$ are the following:

$$\begin{cases} u^{free}_{tmp}(\mathbf{x}) = 2\pi\sigma_{tmp}(\tilde{R} - \hat{R}), \quad E^{free}_{tmp,z}(\mathbf{x})|_{z=0} = 0 \\ E^{free}_{tmp,z}(\mathbf{x}) = 2\pi\sigma_{tmp}(z/\tilde{R} - z/\hat{R}) \\ E^{free}_{tmp,zz}(\mathbf{x}) = 2\pi\sigma_{tmp}(1/\tilde{R} - 1/\hat{R} - z^2/\tilde{R} + z^2/\hat{R}) \end{cases} \tag{8}$$

For a solid disc ($dR = R_{tmp}$) the expressions are:

$$\begin{cases} u^{free}_{tmp}(\mathbf{x}) = 2\pi\sigma_{tmp}(\tilde{R} - |z|), \quad E^{free}_{tmp,z}(\mathbf{x})|_{z=0} = 0 \\ E^{free}_{tmp,z}(\mathbf{x})|_{z<0} = -2\pi\sigma_{tmp}(z\tilde{R}^{-1} + 1) \\ E^{free}_{tmp,z}(\mathbf{x})|_{z>0} = -2\pi\sigma_{tmp}(z\tilde{R}^{-1} - 1) \\ E^{free}_{tmp,zz}(\mathbf{x})|_{z\neq 0} = -2\pi\sigma_{tmp}(\tilde{R}^{-1} - z^2\tilde{R}^{-3}) \\ E^{free}_{tmp,zz}(\mathbf{x})|_{z=0} = +\infty \end{cases} \tag{9}$$

(the importance of $E_{tmp,zz}(z)$ will be shown in Section IV B)).

Figs. 4, 5 show the functions (8), scaled by σ_{tmp} for the infinitesimally thin template. For a thin solid disc both $E_{tmp,z}$ and $E_{tmp,zz}$ have singularities at $z = 0$, and become smoother for rings. In fact the algorithm deals with templates of finite thicknesses $\Delta \neq 0$, as will be shown later in Figs. 13,14. For the $E_{tmp,zz}$, e.g., it results in $E^{free}_{tmp,zz}(\mathbf{x})|_{z=0} = (E^{free}_{tmp,z}|_{z=+\Delta/2} - E^{free}_{tmp,z}|_{z=-\Delta/2})/\Delta$. The derived formulae (8) are the first step in building the template library.

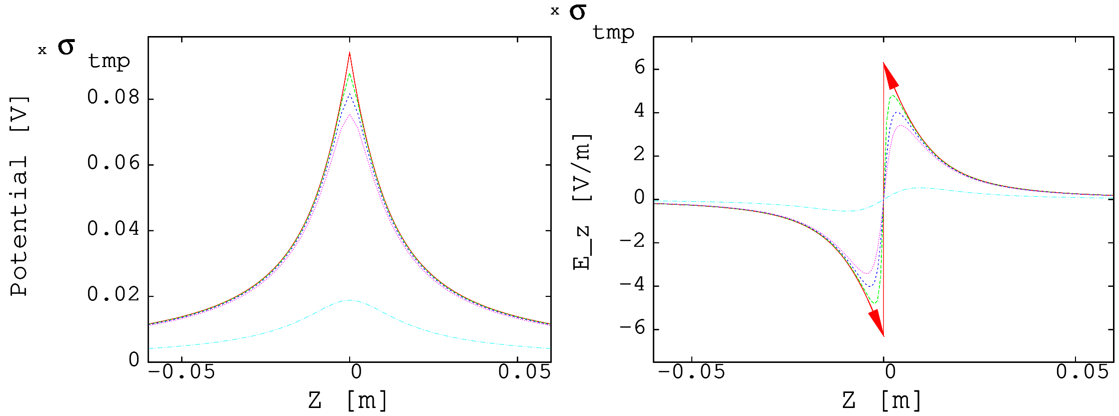

FIG. 4: On-axis potential and fields (8) from a test template ring of $2R_{tmp}$ =3 cm in diameter and $dR = \{1.5,\ 1.4,\ 1.3,\ 1.2,\ 0.3\}$ mm (the solid red line corresponds to dR =1.5 mm and so on).

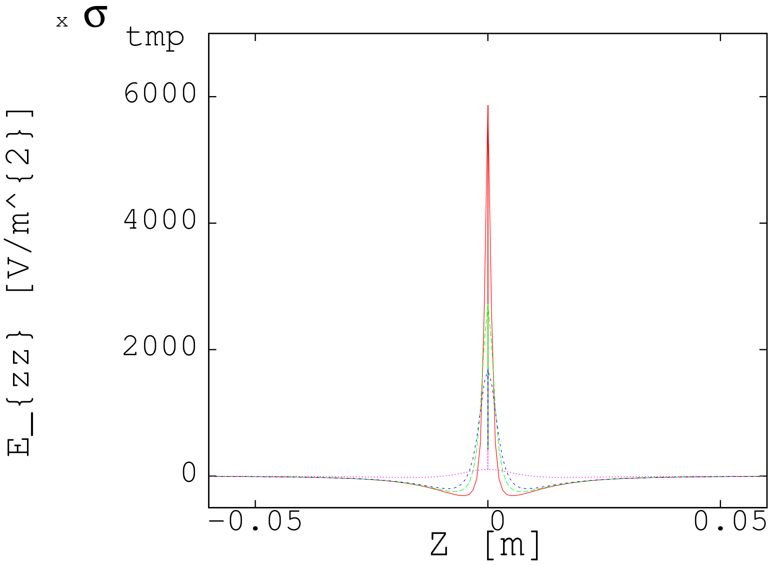

FIG. 5: Function $E_{zz}(z)|_{x=y=0}$ in (8) from a test template ring of $2R_{tmp}$ =3 cm in diameter and $dR = \{1.4,\ 1.3,\ 1.2,\ \text{and}\ 0.3\}$ mm. For a solid disk (dR=1.5 mm), $E_{tmp,zz}(\mathbf{x})|_{z=0} = +\infty$.

B. Template within a conducting boundary

In the presence of conducting boundaries if the boundary and/or the template have no axial symmetry and the charge density is not uniform, a total potential of template u^{total}_{tmp} can be evaluated only numerically. For example, the moment method [30] finds a total potential as a sum of a free space and image charge potentials,

$$u^{total}_{tmp}(\mathbf{x}) = u^{free}_{tmp}(\mathbf{x}) + u^{image}_{tmp}(\mathbf{x}) \tag{10}$$

The formalism, known as charge density method [30], is commonly used in electron and ion optics, when the boundary has a complicated configuration.

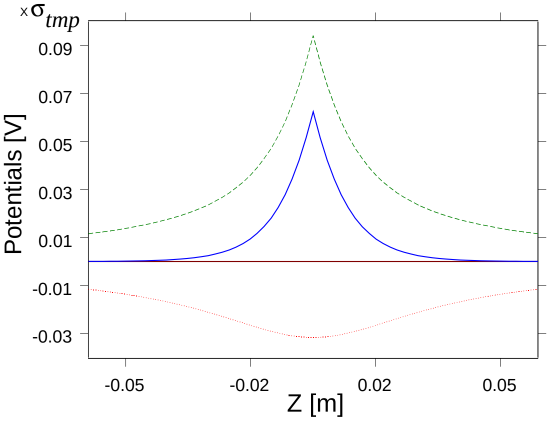

FIG. 6: Total $u_{tmp}^{total}(0,0,z)$ (solid line), free-space $u_{tmp}^{free}(0,0,z)$ (dashed) and image $u_{tmp}^{image}(0,0,z)$ (dotted) potentials.

Fig. 6 demonstrates the corresponding potentials from a round template 3 cm in diameter, within a conducting cylinder 4 cm in diameter, found numerically with an error of 10^{-5}, compared to (8) for u_{tmp}^{free}.

As shown in [29, 31, 32], this approach of integral equations works very slow, but it is easy to program and to apply to arbitrary boundaries; it derives a "split" solution, consisting of image and free space potentials. It may be useful for to some applications (see Section III E below). Alternatively, the grid Poisson solvers, which derive a total potential, can be incomparably faster, but the boundary $\partial\mathcal{R}$ must be simple.

We used both approaches and emphasize, that at this point, we need just the accuracy not the rapidity. The calculations of u_{tmp} are carried out to fill the template library only once and the stored data does not change from design to design.

C. Shape functions for rms-matched beams

In our method a beam is built by templates, filling a 3D beam shell, as shown in Fig. 3 and the shell is determined by the so-called shape-functions (or shell-functions) $S_{x,y}$. In fact the shape functions replace the density block in the conventional PIC algorithm (2). We will find $S_{x,y}(z,p)$ via second momenta $< x^2 >, < y^2 >$ of the distribution function $f(x,y,z,x',y',p)$ in frames of the concept of equivalent beams [4].

Let us consider 2D charge density distribution, which depends on transverse coordinates

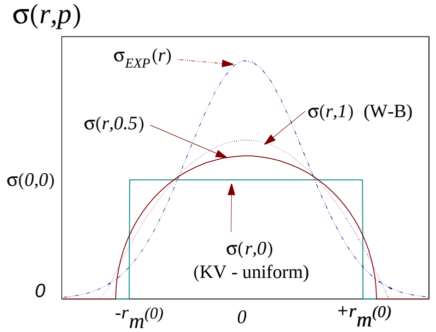

FIG. 7: Charge densities $\sigma(x, y, z, p)$, as functions of "r" for different "p".

(x, y) and a parameter $p \geq 0$, as:

$$\sigma(x, y, z, p) = \sigma_m(p) \left[1 - \frac{x^2}{S_x^2(z,p)} - \frac{y^2}{S_y^2(z,p)}\right]^p \qquad (11)$$

For the case of round beam ($S_x = S_y = r_m$) a substitution of (11) into the transverse rms-size $< x^2 >= \int x^2 \sigma(x, y, z, p) dxdy / \int \sigma(x, y, z, p) dxdy$ yields to $< x^2 > (z, p) = r_m^2(z, p)/2(p + 1)$ and $\sigma_m(p) = 2(p + 1)\sigma_m(0)/(p + 2)$. The same for the vertical rms-size. See [26] for more details.

A generalization for the aspect ratio $\chi \neq 1$ and the semi-axes $a_x = \chi r_m$, $a_y = \chi^{-1} r_m$, results in the expressions for the second momenta as

$$\begin{aligned} < x^2 > (z, p) &= \tfrac{\chi^2 r_m^2(z,p)}{2(p+2)} \\ < y^2 > (z, p) &= \tfrac{\chi^{-2} r_m^2(z,p)}{2(p+2)} \end{aligned} \qquad (12)$$

A variety of charge density distributions may be represented by $\sigma_{tmp}(x, y, z, p)$ in (11). For cases with $p > 0$ the density is maximal at the beam center and going to zero at the edge, for $p = 0$ the density is constant, for $\lim_{p\to\infty} \sigma(r, p) = \sigma_{\exp}(r, p) = (2\pi {<} x^2 >)^{-1} \exp(-r^2/2 < x^2 >)$. All the distributions are rms-matched, having the same rms-sizes in (12).

Finally, for $p < 0$ in (11), the template transverse density σ_p is becoming hollow, appropriate for halo approximation. However, it is more efficient to use ring templates instead of discs from Fig. 8 (see below in Section IV C).

The relations (12) suggest the algorithm about how to find the shape function from the the ensemble of N_p particles with the coordinates $\{\mathbf{x}_k\}$. The longitudinal interval is divided by N_T parts (the subscript means "thick"). The horizontal rms size of a beam for

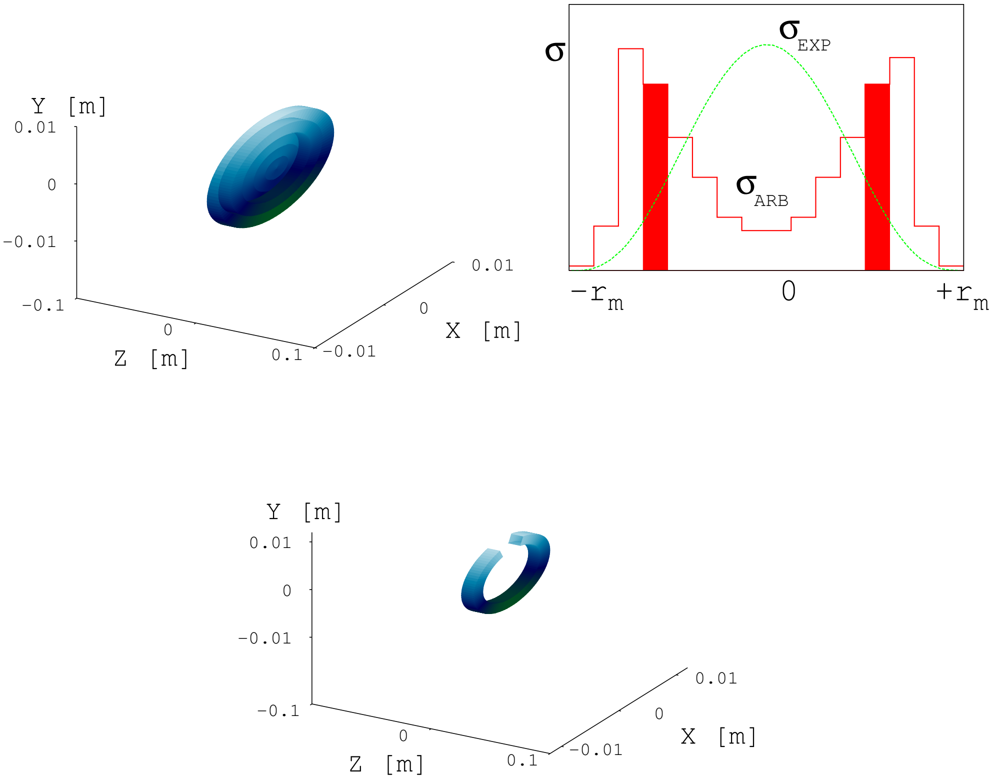

FIG. 8: Ring templates with different densities (left). Arbitrary charge density $\sigma_{ARB}(x, y, z, p)$ (right), formed by ring templates (an exponential charge density σ_{EXP} is shown for comparison). Red bars correspond to the density contribution from a specific ring (bottom). The cut in that ring indicates the geometrical limits, including corresponding particles.

the coordinate z^* is: $< x^2 > (z^*) = \frac{1}{n^*} \sum x_k^2$, with n^* is the number of particles within $|z_k - z^*| \leq (\max z_k - \min z_k)/2N_T = H_T/2$, where $\sum n^* = N_p$. The same procedure is for $< y^2 > (z^*)$. After choosing the parameter "p", the shape functions are:

$$S_x(z^*, p) = \sqrt{2(p+2) < x^2 > (z^*)}$$

$$S_y(z^*, p) = \sqrt{2(p+2) < y^2 > (z^*)} \tag{13}$$

Fig. 9 illustrates the algorithm. From the uniformly populated macro-particle (dots), the dashed line (left) shows the rms-profiles, which define the shape functions (piecewise lines, right). Further smoothing gives the continuous profiles (solid smooth lines, right). The longitudinal line density, used for the scaling in (5) is $\Lambda(z, p) = \lambda_0 \lambda(z, p)$, with $\lambda(z, p) = n^*/\pi S_x(z, p) S_y(z, p) H_T$ and $\lambda_0 = N_p^{-1} \int dz \int \sigma(x, y, z, p) dxdy$, for each "thick" slice $[z^* - H_T/2, z^* + H_T/2]$, with a width H_T.

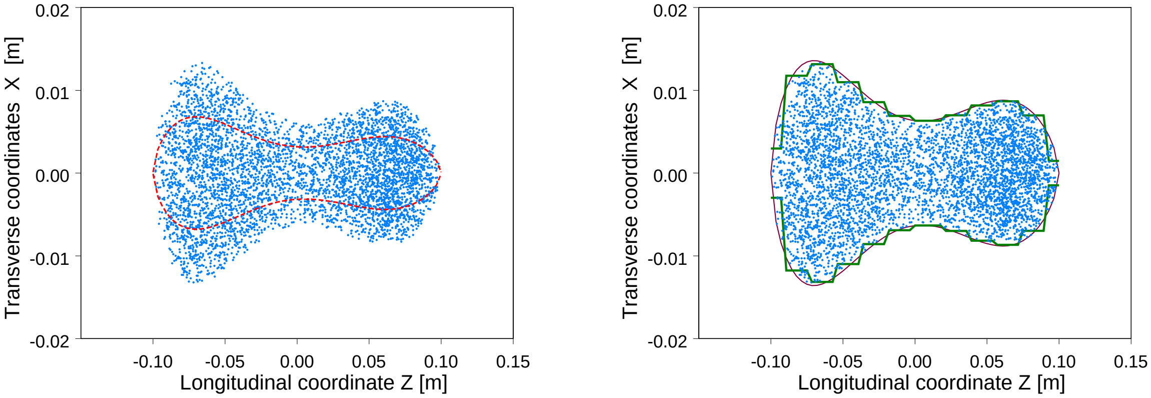

FIG. 9: Illustration of how the shape functions are derived from particle coordinates.

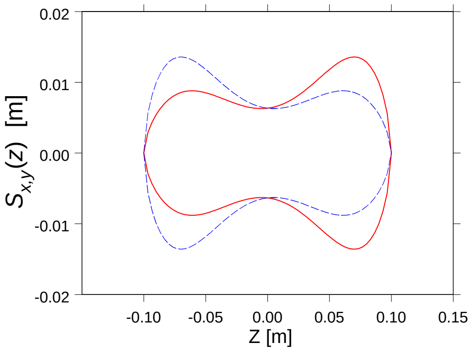

FIG. 10: The shape functions $S_{x,y}(z,p)$.

The shape functions $S_{x,y}(z,p)$ are shown in Fig. 10.

A definition of the shape functions via second momenta may result in some strayed particles, which will be outside of the limits as outlined by $S_{x,y}(z,p)$. This is quite physical, than a derivation of the envelopes, as maximal particle coordinates. Beam representation by the templates is shown in Fig. 3 and a set of corresponding templates (for the arbitrary bunch) is plotted in Fig. 11.

D. Reconstruction of beam potential via templates

According to the shape functions $S_{x,y}$, the templates fill the 3D shell and corresponding template potentials and gradients contribute to the total potential and fields in the

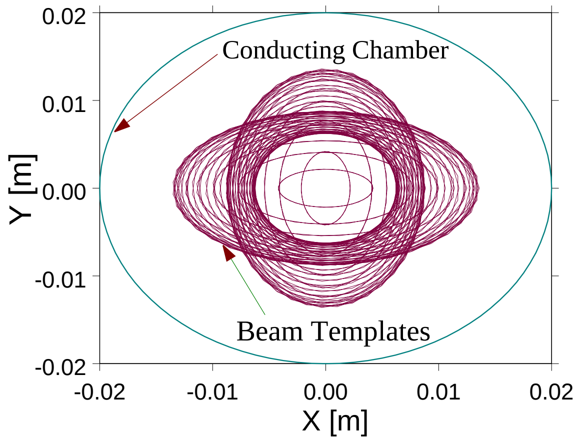

FIG. 11: The templates of different sizes and aspect ratios, used to represent the arbitrary beam from Fig. 3.

convolution integral (5), which is approximated numerically as a finite sum:

$$u_{beam}(x_i, y_j, z_k) = h\sum_{n=1}^{N} u_{tmp}(x_i, y_j, z_{k-n}, S_{x,y}) \tag{14}$$

A number of templates N, depends on beam profiles $S_{x,y}(z_k, p)$ and for bunches from Fig. 1, $N \sim 30 - 40$ was found to be sufficient. A more rapid reconstruction of the total potential may be obtained from the fact that the template potentials drop fast (due to the image forces), as demonstrated in Fig. 6, that allows neglect the contributions from distant templates, narrowing the limits of summation. The situation will be even better for template fields, which drop to zero faster than potentials.

For a real 3D beam, with dimensions shown in Fig. 1 and carrying a total charge of $Q_{total} = 10^{-11}$C the template potentials u_{tmp} and the total beam potential u_{beam} are plotted in Fig. 12 as functions of z, for arbitrary off-axis $x_0 = 0.0025$, $y_0 = 0.004$ [m]. We can see how the boundary problem (6) is substituted by the convolution of template solutions (4). The method delivers potentials for any off-axis shifts with a good accuracy, which can be verified by the successive over-relaxation, three-dimensional (SOR-3D) method, similar to cross-checking procedure from [21].

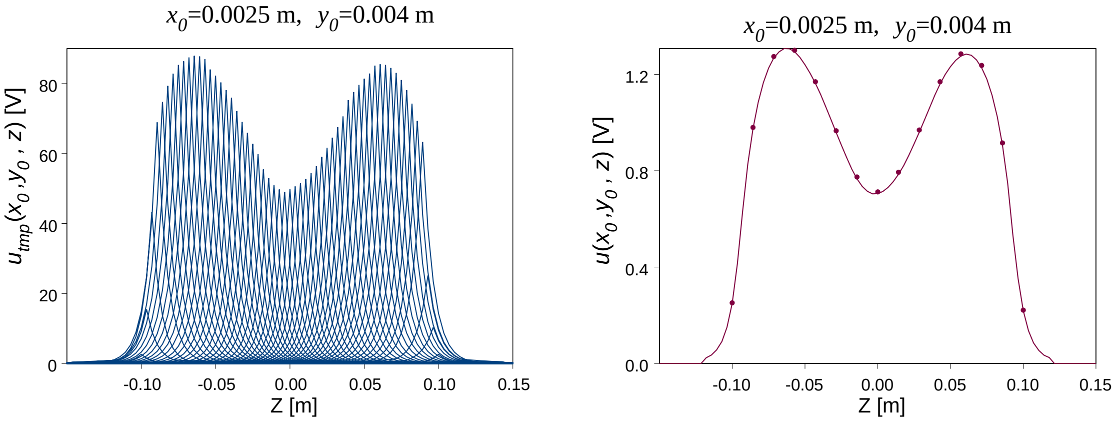

FIG. 12: Template potentials $u_{tmp}(\mathbf{x}_0)$ and the total potential $u_{beam}(\mathbf{x}_0)$.

E. Parametrization. Templates as special functions

As we have mentioned, the time-consuming procedure to find u_{tmp} runs only once and then the computed template potentials are used to reproduce the total bunch potential as a finite sum (14). To provide a large range of possible beam configurations, we have to calculate and store u_{tmp} for all possible template sizes and aspect ratios, as shown in Fig. 11) and for off-axis coordinates for all "z". This requires a lot of memory.

To significantly reduce template data base is possible by using parametrization, when the template potentials are approximated as analytical functions of variable "z":

$$u_{tmp}(\mathbf{x}) = \exp(P_n(\mathbf{x})) = \exp(\sum_n a_n(x, y)z^n) \tag{15}$$

with $u_{tmp}(x, y, z) = u_{tmp}(x, y, -z)$. The coefficients a_n are obtained by a least-square minimization of the array $u_{tmp}(\mathbf{x})$ as function of "z" for fixed (x, y). The 2nd order polynomials P_2 are found to give nearly an ideal approximation of the original potentials. Thus, instead of keeping $u_{tmp}(x_i, y_j, z)$ as functions of "z" for each potential, we need to store only 3 coefficients $a_{0,1,2}$ for all necessary (x_i, y_j). This radically reduces the stored data and note, the analytical representations (15) can be found exactly (within machine accuracy).

IV. HIERARCHY OF TEMPLATE-BASED FIELD SOLVERS

The template formalism generates a family of space charge solvers with different degrees of generality. For each particular case of beam simulation, a specific solver may be chosen.

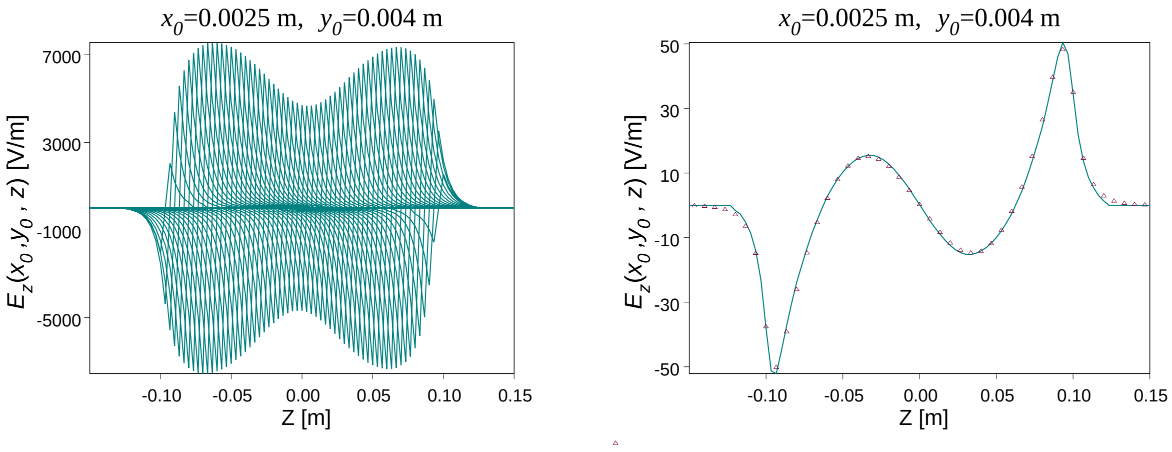

FIG. 13: Template fields $E_{tmp,z}(x_0, y_0, z)$ and total $E_z(x_0, y_0, z)$ field.

A. A fully-parametrized template field solver

An analytical representation (15) allows the longitudinal self-fields to be obtained as the derivatives [27]

$$E_{tmp,z}(\mathbf{x}) = -\frac{\partial u}{\partial z} = -(2a_2 z + a_1) \exp(P_2(\mathbf{x})) \tag{16}$$

with $E_{tmp,z}(x, y, z) = -E_{tmp,z}(x, y, -z)$ (odd functions). The same coefficients a_i, derived for potentials (15), are used for longitudinal fields (16). No additional stored data is required, the derivations are fast, and there is no concomitant loss of accuracy due to numerical differentiation!

The template fields $E_{tmp,z}$ and the total beam field $E_{beam,z}$ are plotted in Fig. 13 as functions of z for arbitrary off-axis coordinates $x_0 = 0.0025$, $y_0 = 0.004$ [m].

For the parametrization of transverse gradients, $E_{tmp,x,y}$, we use the same approach, representing the fields through the exponent and finding the polynomial coefficients by least-square approximation. $E_{tmp,x,y}$ are even functions, similar to the potentials u_{tmp} and behave likewise but much steeper. See [28] for more details. We obtain E_x, as a finite difference, namely, for $\delta \neq 0$:

$$E_{tmp,x}(\mathbf{x}) = -\partial u/\partial x \approx -\left[u_{tmp}(x + \delta, y, z) - u_{tmp}(x, y, z)\right]/\delta \tag{17}$$

The similar procedure is for obtaining $E_{tmp,y}(\mathbf{x})$.

A parametrization reduces the pre-calculated potential and gradients data into a compact library of polynomial coefficients. This library has an analogy to a library of special functions and does not depend on specific beam characteristics. For various currents, sizes of the beam and boundary, the template data can be easily scaled and interpolated. Using these tabulated functions, the solution of the boundary value problem (6) can be found accurately as a sum (14) approximating the convolution integral (5). With the library of $E_{tmp,x,y,z}$, we obtain a fully-parametrized field solver.

B. Sub-3D PIC solvers

The algorithm described above has the best performance, but also the least degree of generality among the others. An idea of more universal sub-3D solvers was introduced in [21]. The original 3D problem was reduced to one or a series of 2D boundary problems. The 3D Laplace operator in (6) may be split, as $\Delta_2 u_{beam} = -4\pi\rho_{beam} - \partial^2 u_{beam}/\partial z^2$, with the notations $\Delta_2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$ for Cartesian (or cylindrical $\Delta_2 = r^{-1}\partial/\partial r(r\partial/\partial r) + r^{-2}\partial^2/\partial\phi^2$) coordinates

Let us combine the second z-derivative with the 3D density and re-write the original 3D Poisson equation (6) as a 2D Poisson equation:

$$\begin{cases} \Delta_2 u_{beam}(\mathbf{x}) = -4\pi\rho_{corr}(\mathbf{x}) \\ \\ u_{beam}(\mathbf{x})|_{\partial\mathcal{R}} = 0 \end{cases} \tag{18}$$

with the appropriately “corrected” 2D density for a “thick” slice with the thickness H_z^T: $\rho_{corr}(x,y,z) = \rho_{beam} + (\partial^2 u_{beam}/\partial z^2)/4\pi = \rho_{2D}/H_z^T - E_{zz}/4\pi$.

The scheme (18) finds 2D potential and, correspondingly, transverse gradients $E_{beam,x,y}$ more accurately than they would be derived from the library of template fields $E_{tmp,x,y}$ by the fully-parametrized solver. Since we have an analytical representation (16), then

$$\begin{aligned} E_{tmp,zz}(\mathbf{x}) &= \partial E_{tmp,z}/\partial z = -({P_2}'' + {P_2}'^2)\exp(P_2) \\ &= -(4a_2^2z^2 + 4a_1a_2z + a_1^2 + 2a_2)\exp(P_2(\mathbf{x})) \end{aligned} \tag{19}$$

that yields after the convolution the second derivative term E_{zz} in the corrected density, which is found analytically without loss of accuracy.

The sub-3D solver substitutes a 3D Poisson equation by a 2D Poisson problem(s) with all the advantages following thereafter [21].

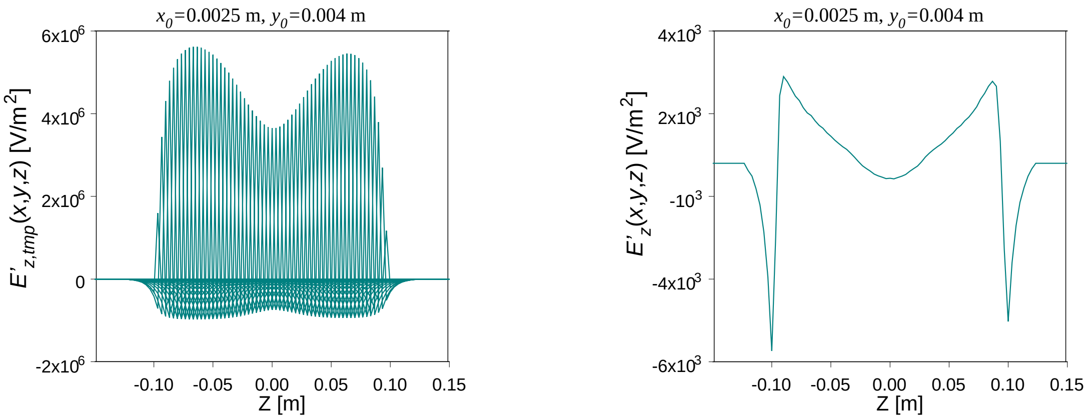

FIG. 14: The function $E_{tmp,zz}(x_0, y_0, z)$ and $E_{beam,zz}(x_0, y_0, z)$ of the arbitrary bunch from Fig. 1.

As we see now, $E_{tmp,zz}$ also plays an important role in the sub-3D computational scheme and the parametrization allows use of the same coefficients $a_{0,1,2}$, that are found for the template potentials (15). Fig. 14 demonstrates $E_{tmp,zz}$ and $E_{beam,zz}$ as functions of z found for off-axis coordinates $x_0 = 0.0025$, $y_0 = 0.004$ [m].

There are two ways to use $E_{tmp,z}$ and E_{zz}. The simple one employs only the axial longitudinal field for the whole beam. There are arguments in support of this model due to the weak sensitivity of E_z from the radial deviations. Fig. 15 shows the longitudinal gradients $E_z(r, z)$, $r = \sqrt{x^2 + y^2}$ for an uniformly charged ellipsoidal-like beam from Fig. 1 (left), carrying the charge $Q_{total} = 10^{-11}$ C, with semi-axes $r_m \times r_m \times Z_m = 0.01 \times 0.01 \times 0.1$ [m] within a conducting cylinder of 4 cm in diameter for several radial positions (due to axial symmetry there is no azimuthal dependence). The fields are non-linear due to image fields (in free space they would be linear). For all internal particle coordinates there is a very good agreement of $E_z(r, z)$, for $0 \leq r \leq r_m$. Only for outside coordinates $r > r_m$, where actually there are no particles, the difference becomes noticeable.

Fig. 16 shows the longitudinal gradients for several off-axis positions for a general beam from Fig. 1 (right), propagating in the same conducting cylinder and carrying the same charge.

Now the longitudinal field has even stronger non-linear behavior than that in Fig. 15. Nevertheless, the on-axis and off-axis fields coincide for a wide range of coordinates within the beam. Errors occur again only at the beam edges.

Therefore, for the beams with nearly uniform densities ($p \approx 0$ in (11)), there is a good

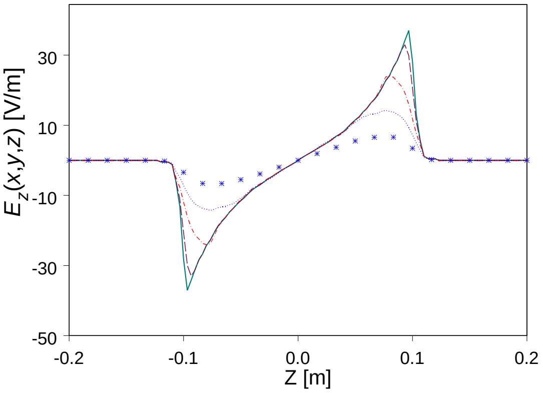

FIG. 15: Longitudinal field $E_z(r,z)$ of ellipsoidal beam bunch from Fig. 1, as function of "z" for different radii of $r = (0, 1/3, 2/3, 1, 4/3) \times r_m$ plotted as solid, dashed, dotted, dashed-dotted lines and symbols correspondingly. Just for the cross-checking purposes note, $E_z(0,z)$ here is identical to the field E_z from Fig. 5.14(c) in the reference [4], p. 407., which are found independently for the same beam, the space charge and the boundary.

reason to deal only with the axial gradient $E_z(0,0,z)$. And, remember that in free space the exact formulae (8) give us a way to obtain on-axis $E_{beam,z}$ and $E_{beam,zz}$ fields on the fly.

A more general approach using off-axis E_{zz} and E_{zz} is more consistent, especially for the situations when the transverse charge density drops fast, e.g., for the case of $\sigma_{\exp}(z, p = \infty)$ in (11).

The computational schemes of the field solvers, are demonstrated in Fig. 17. This is what is used in the square brackets of the algorithm (3). The calculations make use of the precalculated data from the library, which are obtained after solving a series of the boundary problems (4) and a parametrization. Note that the fully-parametrized solver derives all three components of $\mathbf{E}_{beam}$ field from the template library, whereas the sub-3D also derives $E_{beam,z}$ from the library and $E_{beam,x,y}$ are found from (18). Both solvers may use a simplified axial gradients for higher performance.

C. Ring template to generalize transverse densities: halo, hollow beams, etc.

Beam losses due to halo formation may jeopardize the performance of modern high-current accelerators, either linacs or circular machines. To understand halo mechanism and

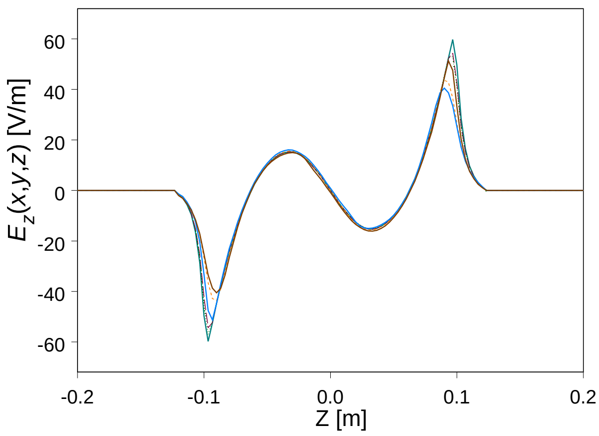

FIG. 16: Longitudinal fields $E_z(x_0, y_0, z)$ of the arbitrary bunch from Fig. 1, as function of "z", for different radii $r_0 = \sqrt{x_0^2 + y_0^2} = (0, 1/3, 2/3) \times 0.01$ m, and angles $(0°, 45°, 90°)$,plotted as solid, dashed, dashed-dotted lines correspondingly.

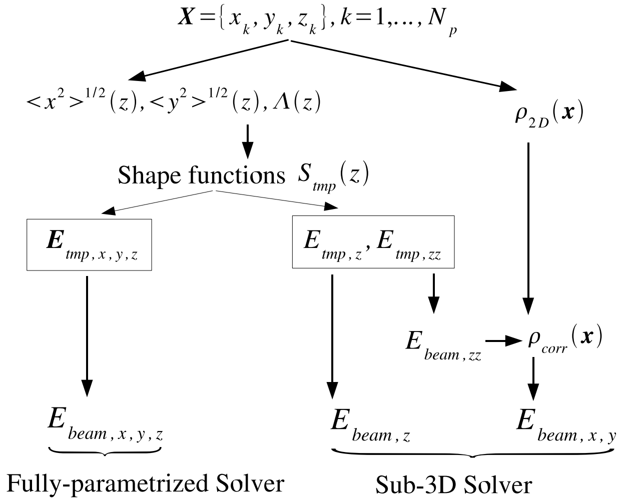
$\boldsymbol{X}=\{x_k, y_k, z_k\}, k=1,\ldots,N_p$
$<x^2>^{1/2}(z), <y^2>^{1/2}(z), \Lambda(z)$
$\rho_{2D}(\boldsymbol{x})$
Shape functions $S_{tmp}(z)$
$\boldsymbol{E}_{tmp,x,y,z}$
$E_{tmp,z}, E_{tmp,zz}$
$E_{beam,zz}$
$\rho_{corr}(\boldsymbol{x})$
$E_{beam,x,y,z}$
$E_{beam,z}$
$E_{beam,x,y}$
Fully-parametrized Solver
Sub-3D Solver

FIG. 17: Template-based space charge calculations by the fully-parametrized and the sub-3D solvers. The rectangles denote the convolution of the template fields $E_{tmp,x,y,z}$ and $E_{tmp,zz}$, borrowed from the pre-calculated template library.

find possible remedies to control it is one of the serious topics of computational accelerator physics. A few approximate numerical models were developed, such as the particle-in-core (PCM) model (see [38] and the references therein) for halo studies in high-intensity rings. The PCM splits the beam by the core and "the rest" and works for essentially non-round beams, including beam in bends, momenta dispersion, etc.

Hollow 2D template charge densities may be described by the formulae (11) with $p < 0$, but the ring templates are more appropriate for beam halo numerical studies. A superposition of rings reconstructs accurately complicated transverse distributions, as in Fig. 8. In free space we already found on-axis ring potentials and fields analytically. In the presence of conducting boundary one builds charged rings, using a superposition of two templates: a bigger one and a smaller one with an opposite sign. By subtraction one gets a general ring-template with pre-calculated potential and fields.

Once we have rings, the problem is solved. Allocating them and scaling the ring's potentials to the number of associated particles (see Fig.8) gives us a total potential of a halo beam with arbitrary aspect ratios as a superposition of template-ring potentials. All machinery of template technique and the template library remain the same.

Schematically, for "M" transverse layers it looks like the following

$$u_{beam}^{halo}(x_i, y_j, z_k) = h \sum_{n=1}^{N} \sum_{m=1}^{M} u_{tmp}^{ring}(x_i^m, y_j^m, z_{k-n}, S_{x,y}) \tag{20}$$

where $x_i^m = x_i \in [m \times S_x/M - \frac{1}{2}S_x/M, m \times S_x/M + \frac{1}{2}S_x/M]$. Similarly for "$y_j^m$".

This technique still needs the shape functions $S_{x,y}(z, p)$, but does not need different distributions in (11). For each template-ring, we assume an uniform charge density. The template library need to be created only for $p = 0$ in (11). The described approach deals with exact potentials of round/non-round rings, taking into account image forces. The performance of solid templates Fig. 7 is faster, whereas the ring templates in Fig. 8 allow more generalization and are still extremely fast. Both ring and "solid" templates could be combined during the simulation. Anyway, they will use the same template data base.

D. Applications to 2D beams

The topic of this paper is the 3D space charge calculation. In the meantime the template idea was also applied to 2D beam models, by a generalization of standard rms envelope equations including the image forces. In [31] we used the templates, which had a form of long cylinders and the total potentials were found from a 2D version of the template method.

The library of 2D templates is smaller than that for 3D ones and the extraction, interpolation and scaling of space charge forces are being done with the speed of calculations of a regular rms envelope model. The cross checking with the 2D PIC code confirmed the validity of the generalized envelope model. As a result, the expanded rms envelope formalism accommodates general boundary as conventional 2D PIC codes do and is dramatically faster. Simulation of beam dynamics can be done in seconds instead of hours.

V. TEMPLATES AND OPTIMIZATION

An optimization of high-current beams is the application, in which the templates could be quite useful, if not indispensable, because the optimization requires repeated beam dynamics calculations to find extreme values of optimization criteria, involving beam and focusing lattice constraints. An optimal design of heavy-ion final focusing system (FFS), when an intense ion beam should be focused onto a tiny pellet providing a thermonuclear confinement fusion, is such an example.

Despite of dealing with strong space charge repelling forces, the optimal focusing should reduce distortions, which are driven by non-linear self-field, and provide fine focus [33], to optimize ear fields [34, 35] for z-compression, reduce the power consumption, and, ultimately, reduce the total cost. All in all, the FFS was the main bottleneck of the thermonuclear facility based on heavy ion confinement fusion [36].

A standard 3D PICs with a typical run time of many hours are inappropriately slow for this task, if we need many iterations (in the optimization loop) to perform. Only the rms envelop formalism with run times in seconds gives an acceptable rapidity and, in fact, is used in many matching/optimization codes for high-current beam design [20, 33].

As discussed in Section I, the KV/envelope equations are useful for many high-current beam designs and for 2D beam models. They may give a good solution, especially when

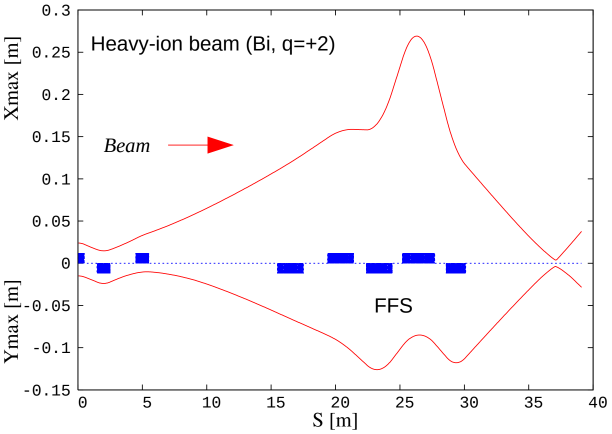

FIG. 18: FFS for heavy ion fusion.

beams envelopes change adiabatically. However, the optimum found by the envelopes for the FFS can be far away from the reality.

The limitations of KV model are: the linear space charge (image forces are non-linear) and external focusing (no multipole correctors, such as sextupole and octupole lenses), quasi-stationary rms emittances (the emittances growth in the FFS is more than 50% for the beams with $p > 0$ in (11)) and the beam maintaining the elliptical symmetry (see the self-explanatory Fig. 1 of a general 3D beam). Moreover, while for adiabatically changing beams the 2D envelopes are more or less appropriate for 2D designs, and serve as a starting point, the 3D envelopes are valid only for sphere-like beams. For more general particle distributions their definition is ill-posed. The transverse envelop sizes are uniquely defined only for the ellipsoid, as maximal values equal to the ellipsoid semi-axes. How then should we assign the transverse envelopes for the configuration as that from Fig. 1 (right)?

The conclusion is that the appropriate accuracy for real design and optimization are achievable only by multiple particle simulation. Since the performance of general PICs is too slow, the templates are intermediate in term of computation times and a generality, giving much more opportunities for the real design and optimization.

For the illustration, consider a result of template-based simulation for the FFS. To be focused on a tiny pellet, the upcoming beam must be expanded horizontally and vertically, so that the magnifying factor is about $\max(x, y)/target(x, y) \approx 100$. The beam parameters were taken: Bi ions (q=+2), E=20 GeV, I =2188 kA ($\beta \approx 0.4$) and a typical bunch length of 4 m. Last quadrupole lenses in the FFS should be superconducting with pole tip fields of 3-4 T. The results of PIC simulation are demonstrated in Figs. 18-20. When the simulation was

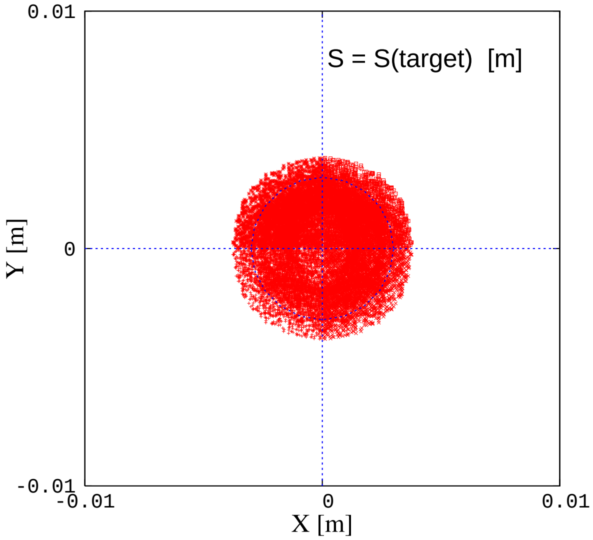

FIG. 19: Particles (x, y) on the target of FFS from Fig. 18. A round target is shown (blue color).

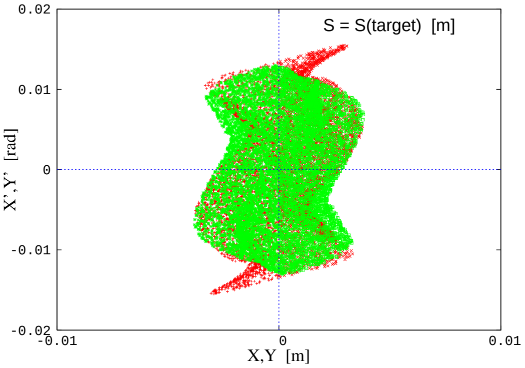

FIG. 20: Particle phase coordinate (x, x') and (y, y') on the target of FFS from Fig. 18.

performed for a beam with KV initial phase distribution, it agreed well with rms-envelope equations with minor imperfections: focusing spot dilution, phase ellipses distortions, etc. due to approximation of such a singular phase distribution as KV. When using the water-bag or exponential beam distributions, the deviation grows further and a correction is needed to suppress chromatic and geometrical aberrations. See refs. [33], [7] for more details.

For a driver with an advanced charge-symmetrical scheme [7], beam envelopes and phase pictures are plotted in Figs. 21-23. The beam parameters were taken: Pt ions (q=±1), E=20 GeV. The use of positive and negative ions allows to boost the beam current up to 10 kA [37] and practically suppresses space charge effects in the driver's transport line (see details in [7]). At the target position one can see strong non-linear distortions of a beam, which may be corrected by use of octupole lenses. This procedure can not be accomplished by the envelope formalism. The only way is to use a general PIC scheme, with drastically

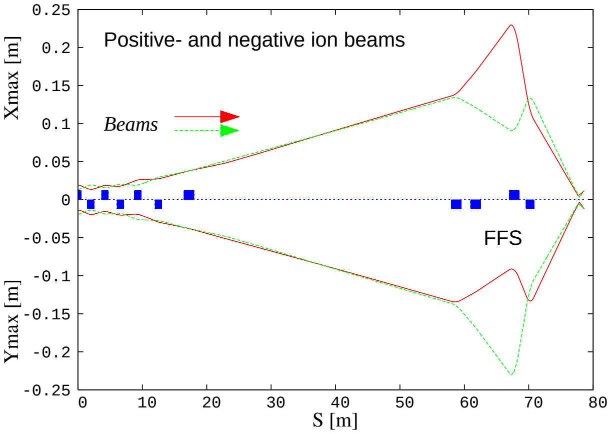

FIG. 21: Charge-symmetrical FFS with positive and negative beams (red and green colors).

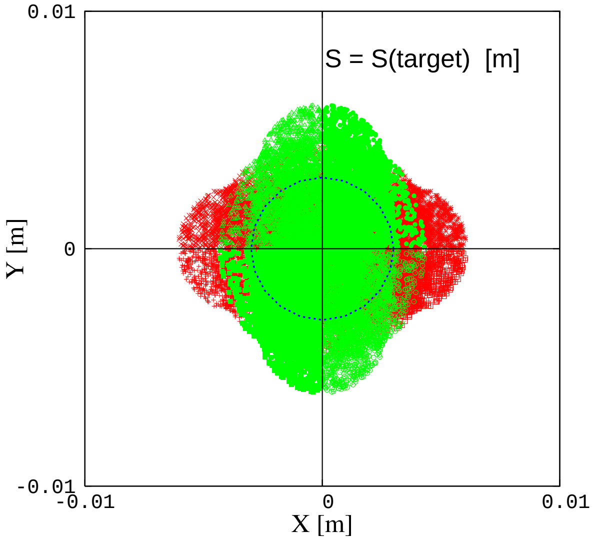

FIG. 22: Particle (x, y) on the target for both beams (red and green colors) of FFS from Fig. 21. A round target is shown (blue color).

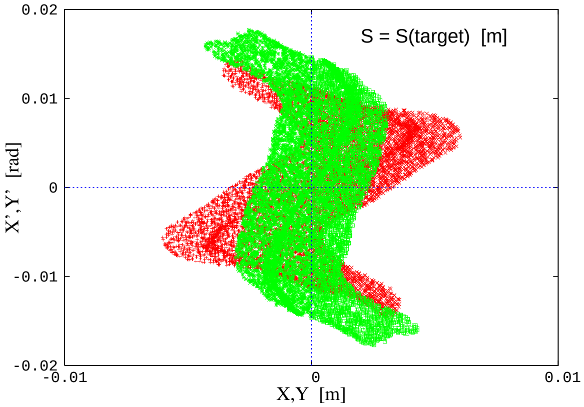

FIG. 23: Particle phase coordinate (x, x') and (y, y') on the target of FFS from Fig. 21 are shown for one beam. For another beam the pictures are identical in opposite order.

improved space charge calculations.

A preliminary version of 3D beam dynamics simulations, using templates, was applied to the above-described FFS design and demonstrated a performance improvement by two order of magnitude with appropriate accuracy. Thus, one beam passage through the FFS (one iteration of the optimization procedure) using templates required 5 min ($N_p = 2 \cdot 10^4$), whereas it was 30 hours with $N_p = 10^7$ for 3D PIC. More testing and detailed implementations, including the correction multipole lenses and arranging the optimization loop aiming to the elimination of spherical aberrations and other non-linear distortions, is the next step of our studies.

VI. DISCUSSION

A. Generality, limitations, applications

A natural question may arise: whether the template technique is somewhat equivalent to the regular grid-based solvers, just applied to coarser grids and running with fewer macro particles. The short answer is no. The template methods provide *exact* space charge calculations for the beams from Fig. 1 having distributions (11). The coarse (and any denser) grid models can hardly achieve the same accuracy.

When real particles evolve, they may not preserve the initially assigned phase space distribution. The grid-based 3D PIC schemes respond to the beam changes through the grid density ρ_{3D}, whereas the template formalism adapts to the changes via the shape function $S_{x,y}$ and a choice of "p" in (11) for solid disc templates, or scaling the ring templates as in Fig. 8. The stored template data is universal and, when properly scaled, is valid for a number of beams and chambers configurations.

The following extensions may increase the performance:

- "Thin" and "thick" templates. See Fig. 3.

- Other non-elliptical templates that can be tabulated [39].

The advanced versions may include a derivation and update of the parameter "p" from the particle distribution (11); a generalization for bent boundaries and non-zero boundary constraints $\overline{U} \neq 0$ in (4),(6) to allow longitudinal wall impedance calculation. In this case the continuity equation for the image charge and the Ohm's law for the boundary surface current should be incorporated to $\overline{U} \neq 0$ in (4) and we have to deal with Maxwell's equations more generally.

We can switch from one level of generality to another, extending the template library, but something will be beyond the reach of the template paradigm. For example, it looks not realistic that the method will accommodate off-set beam with filaments, clusters in complex boundaries. And perhaps it should not. The templates must not cover all variety of particle distributions and boundary constraints, where only the universal 3D grid-based methods are adequate.

For the FFS, the collective instabilities would have not enough time to develop, whereas in the rings the instabilities may be easily covered by templates. The sub-3D PIC (as the most general in the hierarchy of template codes) is the excellent tool for that. It models the 3D effect, using the modified 2D PIC, via corrected density, and properly combines the longitudinal field.

B. Performance and memory requirements

The sub-3D field solver calculates the 3D space charge with the speed of regular 2D grid methods, deriving the transverse gradients from (18). This is very efficient.

As for the fully-parametrized solvers, they are much faster. This is because fewer particles are required to find the shape functions. For example, one can use $N_p \sim 10^4$ particles to build the smooth $S_{x,y}(z,p)$ from Fig. 10. A grid-based method can not afford such a small amount of particles; typically ρ_{3D} needs at least 2 order of magnitude more particles ($N_p \approx 10^6 - 10^7$). The search for the $S_{x,y}$ uses a fast 1D redistribution scheme, giving an additional gain, with the number of operations is just proportional to N_p, unlike the extremely slow 3D redistribution scheme for the spatial grid density. The convolution of template also takes less time than the Poisson grid solvers. The overall performance is faster than the standard grid schemes by at least two orders of magnitude,

The memory required can be estimated straightforwardly. For axial-symmetric beams, 100 pre-calculated templates with different radii (10×) and different off-axis positions (10×) are used to calculate $E_{beam,z}$. For cases without symmetry, e.g. for arbitrary beam from Fig. 1, the number of template potentials increases to 3000 to include different azimuthal positions (3×) and aspect ratios (10×). Thus, the total array of coefficients a_i will consist of 9000 numbers. The parametrization for transverse fields $E_{beam,x,y}$ requires additional data. So, the total volume of the table of coefficients for all three components of the template gradients is tripled: 2.7×10^4. Finally, if we use three values (for ring templates we need only one power $p = 0$) of the parameter "p" from (11), the overall amount will be less than 10^5 numbers, and further extensions are possible. This is a very modest volume to be stored!

VII. CONCLUSION

Multi-particle simulation is an indispensable tool for high-current beams design. To meet the required accuracy, a reduction of execution time will always remain to be a challenging problem, which has been tackled from several directions by different approaches. One of them, the template formalism, incorporates analytics into a numerical procedure.

The idea of templates is to build a basis (library) of macro Green functions, similar to libraries of special functions in mathematical software. Indeed, when using the Bessel or Hankel functions, nobody cares how much time has been spent to tabulate and to store them. The same is with templates. Once we have created their database, further implementations in particle codes would be done on the fly. The stored data will be valid for many applications and does not need to be recomputed.

The template potential data is compact due to analytical parametrization and analytics also help to derive fields not via finite differentiation, but exactly and very fast. The macro Green's functions are deprived of many disadvantages of the point-size Green's function, such as singularities of close-range forces, noise, etc. It turned out to be rewarding to step back from the grid-based solvers and use a seemingly outmoded Green's function formulation.

Since the template method uses particles, it is not limited by artificial geometrical approximations, such as ellipsoids, and manages a number of realistic density distributions, presenting flexible and computationally efficient space charge solvers both in terms of computational intensity (speed, memory) and beam model details.

Using templates, one needs few orders of magnitude less number of particles N_p to reconstruct the shape functions than what would be required by the conventional 3D PIC scheme to reconstruct 3D density. This is the base of rapidity. If 3D PIC uses symmetries (r-z beam), or other simplifications, the template may also use them (the round templates require smaller library). The same for parallelization, it will speed up the technique even more.

The template block may perfectly lay into a standard multi-particle code. It can be naturally included as optional solvers into existing 3D PIC codes, for example, the Synergia package [19]. It will expand the design space and allow to provide an optimization regime, which otherwise would be too time consuming in frames of conventional PIC schemes. So far the electrostatic 3D PIC model (e.g., the code WARP3D [24]) was considered, which is used, for instance, for heavy ion inertial fusion simulations with low β, typically $\beta \approx$0.4. For relativistic beams the template solvers are still appropriate in the moving coordinates, including a self magnetic force into global coordinates. The inclusion of wake fields is another challenging application for space charge templates, important for beam dynamics in long linacs and in rings with $\beta \approx 1$.

Acknowledgements

The author wishes to thank Dr. W. Chou for many useful comments. The discussions with Dr. V.I. Balbekov and Dr. P. Spentzouris are greatly appreciated. The author is especially grateful to Prof. V.Ya. Ivanov for the criticism.

[1] Kapchinsky, I.M., Vladimirsky, V.V.: Proc. Int. Conf. on High-Energy Accelerators and Instrumentation, CERN 1959, p.274-288.

[2] Lapostolle, P.M., IEEE Trans Nucl. Sci. NS-18, 1101 (1971).

[3] Sacherer, F., IEEE Trans Nucl. Sci. NS-18, 1105 (1971).

[4] Reiser, M.: Theory and Design of Charged Particle Beams. Wiley, New York (1994).

[5] Lapostolle, P.M., CERN Report AR/Int SG/65-15, 1965.

[6] Wangler, T.: RF Linear accelerators, Wiley, New York (1998).

[7] Vorobiev L.G.: Institute of Theoretical and Experimental Physics, Preprint ITEP-22 (Moscow, 1994).

[8] Vorobiev L.G., York, R.C., in Proceedings of the 2001 Particle Accelerator Conference, Chicago (IEEE, Piscataway, NJ, 2001), p. 3072. http://accelconf.web.cern.ch/

AccelConf/p01/PAPERS/RPAH097.PDF

[9] Jackson, J.D.: Classical Electrodynamics. Wiley, New York (1975).

[10] Ivanov V., Trudy Instituta Matematiki, **15**, Nauka, Sibirskoe Otdelenie Akademii Nauk, Novosibirsk, (1989), pp. 172-187.

[11] Ivanov V.: Green's Function Technique in Forming of Intensive Beams, XV Int. Workshop "Beam Dynamics and Optimization", 10-13 July 2008, St. Petersburg, Florida, USA. Submitted to "Modern Physics, A".

[12] Hockney, R.W., Eastwood, J.W.: Computer Simulation Using Particles. McGraw-Hill, (1981).

[13] Press, W.H., Teukolsky, S.A., Vetterling, W.,T., Flannery B.P.: Numerical Recipes. Second Edition. Cambridge (1992).

[14] Birdsall, C.K., Langdon, A.B.: Plasma Physics via Computer Similation. McGraw-Hill (1985).

[15] Berz, M.: COSY Infinity, http://cosyinfinity.org.

[16] Servranckx, R.V.: DIMAD, TRIUMF, Vancouver, BC, (1993).

[17] The MAD Program. http://mad.web.cern.ch/mad/.

[18] ORNL Beam Physics Group, ORBIT User Manual. http://www.sns.gov/APGroup/Codes/

[19] Synergia. http://cd-amr.fnal.gov/aas/nSynergia.html

[20] Los Alamos Beam Dynamics Software. http://laacg1.lanl.gov/laacg/c-on-line/bd.html

[21] Vorobiev, L.G., York, R.C.: Phys. Rev. ST Accel. Beams **3**, 114201 (2000). http://prst-ab.aps.org/abstract/PRSTAB/v3/i11/e114201

[22] Callahan D.A., Langdon A.B., Friedman A., Haber I., LLNL preprint, UCRL-JC-112247, 1993.

[23] Wang D.X., Wang J.G., Reiser M., Phys. Rev. Letters, **73**, No. 1, 1994, p.66.

[24] Virtual National Laboratory for Heavy Ion Fusion, WARP. http://hif.lbl.gov/theory/WARP_summary.html

[25] Lapostolle P.M., et al., Proc. 18th Int. Linac Conf., CERN, Geneva, 1996, p.375.

[26] Vorobiev L.G., York, R.C., in Proceedings of the 2001 Particle Accelerator Conference, Chicago (IEEE, Piscataway, NJ, 2001), p. 3075. http://accelconf.web.cern.ch/AccelConf/p01/PAPERS/RPAH098.PDF

[27] Vorobiev L.G., York, R.C.: in Proceedings of the 2002 European Particle Accelerator Conference, (Paris, 2002), p. 1679. http://accelconf.web.cern.ch/AccelConf/e02/PAPERS/

WEPLE092.PDF

[28] Vorobiev, L.G., York, R.C.: in Proceedings of the 2003 Particle Accelerator Conference, Portland, OR (IEEE, Piscataway, NJ, 2003), p. 3533. http://accelconf.web.cern.ch/AccelConf/p03/PAPERS/FPAG035.PDF

[29] Vorobiev L.G., York, R.C.: Michigan State University Report No. MSUCL-1117, (1998).

[30] Harrington, R.F.: Field Computation by Moment Methods. Macmillan Company, New York (1968).

[31] Vorobiev, L.G., York, R.C., in: Proceedings of the 1999 Particle Accelerator Conference, New York, (IEEE, Piscataway, NJ, 1999) 2781-2783. http://accelconf.web.cern.ch/AccelConf/p99/PAPERS/THA79.PDF

[32] Vorobiev L.G., York, R.C.: in Computational Science, edited by P.M.A. Sloot et al. (Springer, 2002), Vol. LNCS 2331, p. 315. http://link.springer.de/link/service/series/0558/bibs/2331/23310315.htm

[33] Vorobiev, L.G., Zenkevich, P.R., Koshkarev D.G.: Atomnaya Energiya **63**, 3, p.202, 1987.

[34] Friedman A.: in AIP Conference Proceedings 377, Bloomington, IN, 1995, p. 401.

[35] Sharp, W.M., Callahan, D.A., Grote, D.P., ibid., p. 434.

[36] HIBALL- II: An improved conceptual heavy ion beam driven fusion reactor study, Kernforschungszentrum Karlsruhe, KfK 3840 (Jul. 1985).

[37] Koshkarev D.G., Preprint ITEP-7, (Moscow, 1994).

[38] Holmes, J.A. et al.: Phys. Rev. ST Accel. Beams **2**, 114202 (1999). http://prst-ab.aps.org/abstract/PRSTAB/v2/i11/e114202

[39] For space charge dominated beams the elliptical shape of the beam transforms to a box-like bunch with rectangular profiles. I. Hofmann and J. Struckmeier, private communication, GSI, Darmstadt 2002. See also the reference [4], p. 355.