EDITED BY M BERZ AND K MAKINO

COMPUTATIONAL ACCELERATOR PHYSICS 2002

Proceedings of the 7th International Conference on Computational Accelerator Physics, Michigan, USA, 15–18 October 2002

175

IoP

Computational Accelerator Physics 2002

**Proceedings of the Seventh International Conference on
Computational Accelerator Physics
Michigan, USA, 15–18 October 2002**

Edited by
Martin Berz and Kyoko Makino

Preface

This volume contains selected papers from the Seventh International Computational Accelerator Physics Conference that took place 15–18 October, 2002 at Michigan State University. Following a now well established tradition, the meeting succeeded the conference in Darmstadt in 2000, and in turn will be followed by the conference in St. Petersburg, Russia in 2004.

Kyoko Makino and Martin Berz

Other titles in the series

The Institute of Physics Conference Series regularly features papers presented at important conferences and symposia highlighting new developments in physics and related fields. Previous publications include:

Contents

Persistent Currents in Superconducting Filaments Due to Arbitrary Field Changes in the Transverse Plane

Martin Aleksa, Bernhard Auchmann,
Stephan Russenschuck, Christine Völlinger

Abstract. Magnetic field changes in the coils of superconducting magnets are shielded from the filaments' core by so-called persistent currents which can be modeled by means of the critical state model. This paper presents an semi-analytical 2-dimensional model of the filament magnetization due to persistent currents for changes of the magnitude of the magnetic induction *and* its direction while taking the field dependence of the critical current density into account. The model is combined with numerical field computation methods (coupling method between boundary and finite elements) for the calculation of field errors in superconducting magnets. The filament magnetization and the field errors in a nested multipole corrector magnet have been calculated as an example.

1. Introduction

The Large Hadron Collider (LHC) [6], a proton-proton superconducting accelerator, will consist of about 8400 superconducting magnet units of different types, operating in super-fluid helium at a temperature of 1.9 K. The applied magnetic field changes induce currents in the filaments that screen the external field changes (so-called persistent currents). The filaments are made of type II hard superconducting material with the property that the magnetic field penetrates into the filaments with a gradient that is proportional to the magnitude of the persistent currents. Macroscopically, these currents (that persist due to the lack of resistivity if flux creep effects are neglected) are the source of a magnetization $\vec{M}(\vec{B})$ of the superconducting strands. One way to calculate this magnetization would be to mesh the coil with finite elements and solve the resulting non-linear field problem numerically by making use of a measured $\vec{M}(\vec{B})$-curve. This approach has two main drawbacks: The numerical field computation has to be combined with a hysteresis model for hard superconductors, and the coil has to be discretized with highest accuracy also accounting for the existing gradient of the current density due to the trapezoidal shape of the cables, the conductor alignment on the winding mandrel, and the insulation layers. Hence, we aimed for computational methods that avoid the meshing of the coil by combining a semi-analytical magnetization model with the BEM-FEM coupling method [5].

In the straight section of accelerator magnets the magnetic induction is almost perpendicular to the filament axis. The effect of a magnetic induction parallel to a superconducting filament is small (see [11]) and has therefore been neglected here. A model to calculate the magnetization of the superconducting strands is presented, considering external fields that change their magnitude and direction. For this purpose, the model introduced in [1] has been extended to account for filament magnetizations non-parallel to the outside field. As in [1], the model does not attempt to describe the microscopic flux pinning, but applies the critical state model [2] which states that any external field change is shielded from the filament's core by layers of screening currents at critical density $j_c(B, T)$. The model differs from other attempts to describe a superconducting filament's response to arbitrary field changes in the transverse

plane as, e.g., in [8]. It takes into account the dependence of the critical current density on the applied external fiel and the resulting field distribution in the filament cross-section. As a consequence, also low field effects such as the peak-shifting (asymmetry in the magnetization curve for vanishing external field) are reproduced by the model.

The described model is combined with the coupled boundary element / finite element method (BEM-FEM) [5] which avoids the representation of the coil in the finite element mesh since the coil is located in the iron-free BEM domain. The fields arising from current sources in the coil are calculated by means of the Biot-Savart law, while the surrounding ferromagnetic iron yoke has to be meshed with finite elements. Hence, the discretization errors due to the finite-element part in the BEM-FEM formulation are limited to the iron-magnetization arising from the yoke structure. In order to account for the feed-back of the filament magnetization on the magnetic field, an $\vec{M}(\vec{B})$-iteration is performed.

The magnetization model is based on the input function $j_c(B, T)$ of the critical current density, which represents the material properties of the superconductor, but is independent of geometrical parameters such as the filament diameter or shape and the ratio λ of the superconductor to total strand volume. The method reproduces the hysteretic behavior for arbitrary turning points in the magnet's excitational cycle including minor loops and rotating external fields.

2. The 1-dimensional magnetization model

For a better understanding of the magnetization model, let us first consider a field change of the form $\Delta B = \xi B \vec{e}_B$ where $\xi \in [-1, 1]$ and B is the nominal field strength in some direction $\vec{e}_B$ perpendicular to the axis of a circular superconducting filament, which we shall call a 1-dimensional field change. This field change induces a shielding-current layer of a relative thickness q^* called the relative penetration depth, see Fig. 1. It is measured on the scale of the relative penetration parameter q that is zero on the outside and one in the center of the filament. The currents are directed as to create a magnetic induction that opposes the applied field change on the conductor surface, thus shielding the field change from the filament's core. The thickness of the layer depends on the amplitude of the applied field sweep, on the filament radius, and on the critical current density in the superconducting material.

The generation of a shielding field can be modeled by the perfectly uniform dipole field produced by two intersecting circles with opposite current densities shifted by the relative distance $\Delta q = q_2 - q_1$,

$$|\Delta \vec{t}| = \frac{\mu_0 r}{2} \int_{q_1}^{q_2} j_c(B(q)) \, \mathrm{d}q, \tag{1}$$

where $\Delta \vec{t}$ is the shielding field, r is the filament radius and q_1, q_2 are the relative penetration parameters that limit the shielding current layer, [3]. Such pairs of circles are nested inside concentric circles. This equation will later be used to find a differential equation for the differential shielding $\mathrm{d}\vec{t}(\mathrm{d}q)$. In Fig. 1 these nested pairs of circles are represented with finite thickness, notwithstanding the continuous nature of the mathematical model, which will be introduced in Sec. 3.1. Figure 1 (left) shows the cross-section of a filament after a 1-dimensional change of the external field from $\vec{0}$ to $\vec{B}_{\mathrm{new}}$. The nested circles each shield a fraction of the outside field from the inside, thus increasing $j(B)$ in the inner circles, as represented in Fig. 1 (left bottom diagram). The figure also yields a vector representation of the 1-dimensional field change and the corresponding shielding effect. The vector $\vec{t}(q)$ indicates the shielding magnetic induction as a function of the penetration from the outside $(q = 0)$, where $\vec{t}(0) = \vec{0}$ and $\vec{B}(0) = \vec{B}_{\mathrm{out}} + \vec{0}$, to the inner boundary of the shielding layer
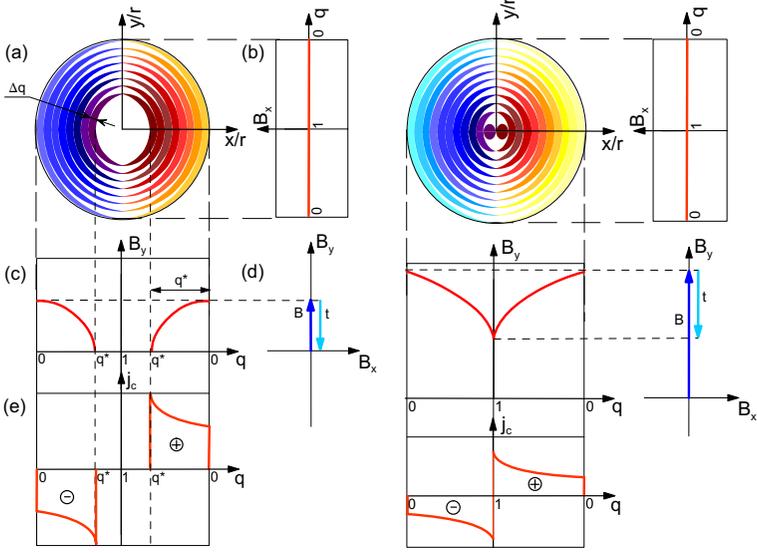
Figure 1. Circular superconducting filament in a magnetic induction $\vec{B}$ of fixed direction, for different penetration states. The individual graphs contain: (a) A schematic view of the circular filament with inscribed pairs of circles. Orange colors indicate positive currents in z-direction, blue colors indicate negative currents. The color intensity represents the absolute value of the currents at critical density $j_c(B(q), T)$; (b) The B_x component of the magnetic induction over the relative penetration parameter q; (c) The B_y component of $\vec{B}(q)$ over q; (d) The vector representation of the shielding problem in the B_x/B_y-plane; B and T denote the external field vector $\vec{B}$ and the shielding field vector at penetration q^*, $\vec{t}(q^*)$, respectively; (e) The shielding currents at critical density $j_c(B(q), T)$ in the cross-section. Left: Penetration to a relative penetration depth of q^*. Right: Full penetration ($q^* = 1$).

$(q = q^*)$, where $\vec{B}(q^*) = \vec{B}_{\text{out}} + \vec{t}(q^*) = \vec{0}$. The absolute value of $\vec{t}(q)$ depends in a non-linear way on the penetration parameter q (see $j_c(B(q))$-relation in Eq. (6) and on the applied field change.

The right hand side of Fig. 1 shows the situation where a larger field change is applied to the filament surface. The entire cross-section contains shielding currents of critical density which are, however, unable to completely shield the field from the inside. The field has thus fully penetrated the filament ($q^* = 1$). In the vector representation, $\vec{t}$ points from the induction at the filament surface $\vec{B}(q = 0)$ to the value of the induction at the center of the filament $\vec{B}(q = q^*)$.

Fig. 2 (left) shows the case where the magnetic induction outside the filament is ramped up to $\vec{B}_{\text{old}}$ (previously denoted $\vec{B}_{\text{out}}$) and subsequently reduced to $\vec{B}_{\text{new}}$. A new layer of shielding currents is generated, leaving the remaining inner layers untouched. The new field change is again shielded from the filament's core. The shielding vector $\vec{t}_{\text{new}}(q)$ is now to oppose the new field change. It further has to fulfill continuity requirements on the outer ($q = 0$) and inner boundary of the new current layer ($q = q^*$):

$$\vec{B}(0) = \vec{B}_{\text{new}} + \underbrace{\vec{t}_{\text{new}}(0)}_{0} = \vec{B}_{\text{new}}, \qquad (2)$$

$$\vec{B}(q^*) = \vec{B}_{\text{new}} + \vec{t}_{\text{new}}(q^*) = \vec{B}_{\text{old}} + \vec{t}_{\text{old}}(q^*). \qquad (3)$$

4

With given $\vec{B}_{\text{old}}$, $\vec{t}_{\text{old}}$ and $\vec{B}_{\text{new}}$, the mathematical problem consists in the determination of a penetration parameter q^* and the corresponding shielding vector $\vec{t}_{\text{new}}$ that satisfies the Eqs. (2) and (3). For the 1-dimensional field change this task has been discussed in [1].
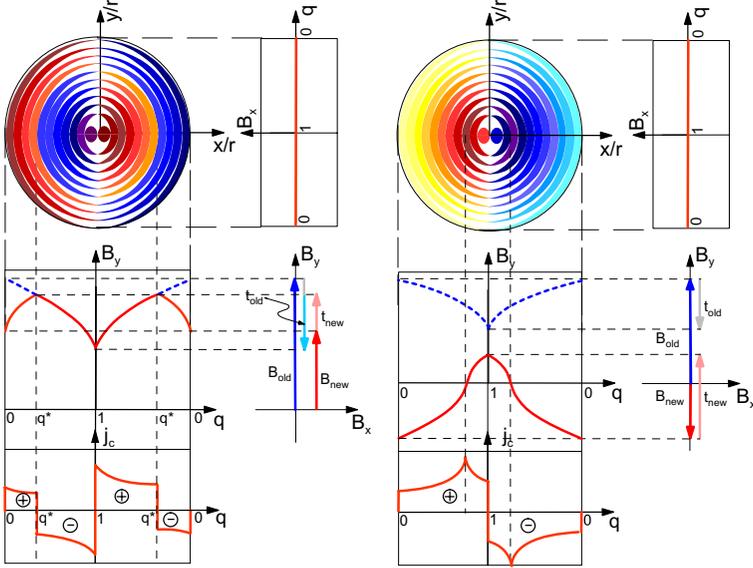
Figure 2. Sequel to Fig. 1 (right). Different 1-dimensional field changes applied to a fully penetrated superconducting filament. Left: The diminution of the external field causes the creation of a new current layer of relative thickness q^* with opposite current densities. Right: The outer field changes sign, causing the new current layer to completely erase the previous layer ($q^* = 1$). The gray part of the $\vec{t}_{\text{old}}$ vector represents $\vec{t}_{\text{old}}(q^*)$, the part of the shielding magnetic induction that has been replaced by the new shielding layer $\vec{t}_{\text{new}}(q^*)$.

Figure 2 (right) shows a case where a field changes wipes out the previous current layer(s) entirely. This happens, if the field change is too large (or the field is turned into a direction that does not allow for an intersection of shielding vectors, see Sec. 3). It further shows that the critical current density reaches its maximum at $B(q) = |\vec{B}(q)| = 0$ where the shielding effect is biggest and the $B(q)$ curve has the steepest inclination.

3. Persistent Currents Model for 2-dimensional field changes

The more general case of a magnetic induction that changes its absolute value *and* direction is discussed now. We shall denote the underlying model therefore as "2-dimensional" or the "vector magnetization model". Starting again from the situation described in Fig. 1 (right), a clockwise rotation and diminution of the magnetic induction is now studied. It can be expected from the results presented in Section 2 that a new shielding-current layer of relative thickness q^* is created to oppose the change of the induction. Of course, the continuity equations (2) and (3) also have to hold for rotational field changes.

Figure 3 (left) illustrates the case. The shielding vector $\vec{t}_{\text{new}}(q^*)$ points to $\vec{B}_{\text{old}} + \vec{t}_{\text{old}}(q^*)$. It is directed as to shield field changes from the filament's core in accordance with the continuity
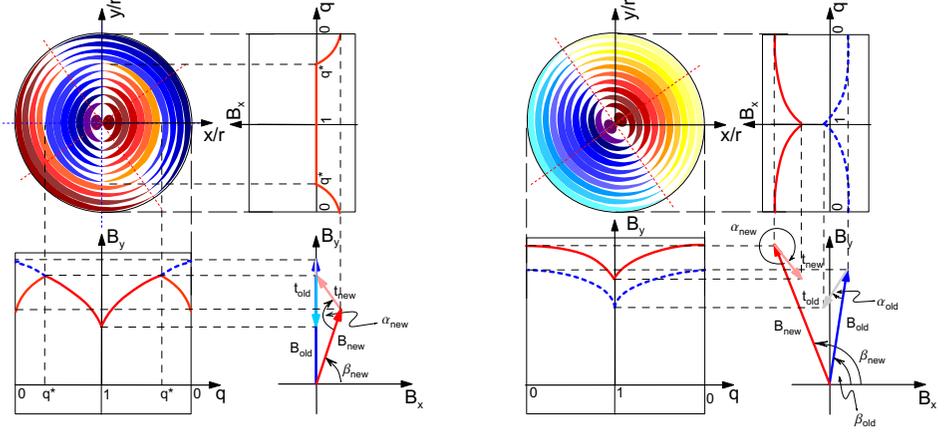
Figure 3. Left: Sequel to Fig. 1 (right). The outer field $\vec{B}_{new}$ is decreased and turned by $\beta_{new} - \beta_{old}$ with respect to the previous excitation step $\vec{B}_{old}$. A new current layer is created with relative thickness q^*, that shields the field change from the filament's core. Right: $\vec{B}_{new}$ is increased and rotated with respect to $\vec{B}_{old}$. The field change fully penetrates the cross-section. In the vector representation, the shielding vector $\vec{t}_{new}$ points to the arrowhead of the old shielding vector $\vec{t}_{old}$.

equations (2) and (3). The magnetic induction in the filament cross-section is, thus, given by

$$\vec{B}(q) = \begin{cases} \vec{B}_{new} + \vec{t}_{new}(q), & 0 < q < q^* \\ \vec{B}_{old} + \vec{t}_{old}(q), & q^* < q < 1. \end{cases} \tag{4}$$

Similar to the case presented in Fig. 2 (right), the current layer resulting from the outer field change shown in Fig. 3 (right) penetrates the entire cross-section. Again, the previous shielding-current layer is completely removed ($q^* = 1$). As the effects are computed for successive excitational conditions and the computational results ought to be independent of the step sizes of these excitations, the $\vec{t}_{new}$ vector for $q^* = 1$ points to the arrowhead of $\vec{t}_{old}$. This behavior is illustrated in Fig. 4.

3.1. Mathematical description of the vector magnetization model

Most magnetization models for superconducting filaments (e.g. [11]) neglect the field-dependence of the critical current density j_c. This is reasonable only if the excitational field is large compared to the field generated by the filament magnetization (i.e., all the filaments are fully penetrated). There are, however, regions in superconducting coils that yield a magnetic induction in the transverse plane that is close to zero even at nominal level. Strands that are positioned in these regions - the vorteces of the magnetic induction - are not fully penetrated during the entire ramp cycle of a magnet. The model introduced in this paper includes varying current densities inside the filament and calculates the continuous course of the magnetic field over the filament cross-section by means of a differential approach. The model is, therefore, suited to describe also low-field effects such as the peak-shifting, see Fig. 4. The following approach was adopted to derive the mathematical model:

- A differential equation for the course of the magnetic field is derived, based on the equation for the perfectly uniform dipole field produced by a pair of intersection circles, compare Eq. (1).
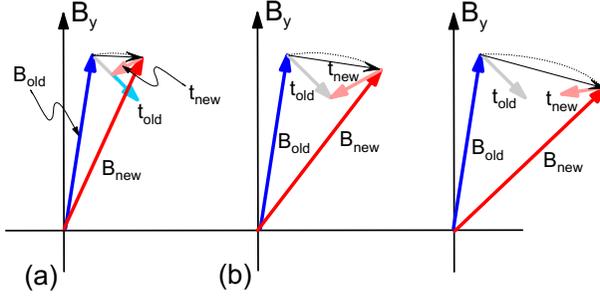
Figure 4. Vector representation of steps of different sizes, following an increase and rotation of the magnetic induction on a filament's surface (dashed line). This plot is to illustrate why, after a large step, (c), the shielding vector $\vec{t}_{new}$ has to point to the arrowhead of $\vec{t}_{old}$: Increasing the step size, the moment of full penetration of the field ($q^* = 1$) is reached in (b). Assume now that the step size is being infinitesimally increased. In order to avoid any incontinuity of the results due to a choice of step sizes, the new shielding vector must point to the arrow head of the old vector. Applying the same reasoning to a larger step, (c), it can be seen that letting $\vec{t}_{new}$ point to the arrow head of $\vec{t}_{old}$ is the appropriate approximation of a series of successive infinitely small steps along the dashed line.

- To obtain a solveable differential equation, the fit function for the critical current density is approximated around the working point.
- A set of differential equations for the x- and a y-components is derived, to describe arbitrary field changes in the transverse plane.
- The shielding induction vector $\vec{t}(q)$ is introduced to describe the course of the induction over the cross-section.
- One differential equation for $t(q) = |\vec{t}(q)|$ is obtained. Solving the equation yields the inverse relation $q(t)$.
- With this solution at hand, the penetration parameter of a new shielding current layer can be determined by solving an equation system, given by the continuity requirements in Eqs.(2-3).
- Given the limits of each current layer and the respective $j_c(B(q))$ relation, the magnetization of the filament can be calculated.

The critical current density as a function of the magnetic induction $B(q) = |\vec{B}(q)|$ is given by the following fit function [4], where $B_c = B_{c_0}(1 - (T/T_{c_0})^{1.7})$:

$$j_c(B(q),T) = \frac{j_c^{ref} C_0 B(q)^{\alpha-1}}{(B_c)^{\alpha}}\left(1 - \frac{B(q)}{B_c}\right)^{\beta}\left(1 - \left(\frac{T}{T_{c_0}}\right)^{1.7}\right)^{\gamma}. \tag{5}$$

The fit parameters for the LHC main-magnet cables are a reference current density $j_c^{ref} = 3 \cdot 10^9$ A/m^2 (at 4.2 K and 5 T), an upper critical field $B_{c_0} = 14.5$ T, a critical temperature of $T_{c_0} = 9.2$ K, a normalization constant $C_0 = 27.04$ T and the fit parameters $\alpha = 0.57$, $\beta = 0.9$ and $\gamma = 2.32$.

For small magnetic inductions $B(q)$, where the persistent currents have the biggest impact, it follows that $j_c(B(q),T) \to \infty$ with $B(q)^{\alpha-1} = B(q)^{-0.43} \sim 1/\sqrt{B(q)}$ for $B(q) \to 0$. For the computation of the induction inside the filament, Eq. (5) is approximated around the working

point $B_{\text{out}} = |\vec{B}_{\text{out}}|$ with the following function (T is constant):

$$j_{\text{c}}(B, T) \sim j_{\text{c}}(B_{\text{out}}) \frac{\sqrt{B_{\text{out}}}}{\sqrt{B(q)}} \equiv \frac{\mathcal{F}(B_{\text{out}})}{\sqrt{B(q)}}. \tag{6}$$

From Eq. (1) for the perfectly uniform dipole field produced by a pair of intersection circles, a system of differential equations for the field change within the filament $d\vec{B}(q)$ can now be derived:

$$\begin{pmatrix} dB_x(q) \\ dB_y(q) \end{pmatrix} = \frac{\mu_0 \mathcal{F}(B_{\text{out}})\mathcal{H}}{\sqrt{B(q)}} \begin{pmatrix} -\cos(\alpha - \beta) \\ \sin(\alpha - \beta) \end{pmatrix} dq. \tag{7}$$

The angles are defined in Fig. 3. The geometry factor $\mathcal{H} = 2r(1 - \ln 2) = 0.614\, r$, where r denotes the filament radius, accounts for the little spaces that are left when a round filament is filled with a series of intersecting circles inscribed in concentric circles (instead of intersecting ellipses that would avoid these spaces but which could not be inscribed in concentric circles, see Fig. 1 (left) and [1]). By setting

$$\underbrace{\begin{pmatrix} B_x(q) \\ B_y(q) \end{pmatrix}}_{\vec{B}(q)} = \underbrace{B_{\text{out}} \begin{pmatrix} \cos(\beta) \\ \sin(\beta) \end{pmatrix}}_{\vec{B}_{\text{out}}} + \underbrace{t(q) \begin{pmatrix} -\cos(\alpha - \beta) \\ \sin(\alpha - \beta) \end{pmatrix}}_{\vec{t}(q)}, \tag{8}$$

and hence

$$B(q) = \sqrt{B_{\text{out}}^2 - 2B_{\text{out}} t(q) \cos(\alpha) + t^2(q)} \tag{9}$$

we can derive from Eq. (7) a differential equation for $t(q)$:

$$dt \sqrt[4]{B_{\text{out}}^2 - 2B_{\text{out}} t(q) \cos(\alpha) + t^2(q)} = \mu_0 \mathcal{F}(B_{\text{out}})\mathcal{H}\, dq. \tag{10}$$

Substituting $u = t(q) \cos \alpha$ yields

$$du \sqrt[4]{u^2 + B_{\text{out}}^2 \sin^2(\alpha)} = \mu_0 \mathcal{F}(B_{\text{out}})\mathcal{H}\, dq \tag{11}$$

A solution for $q(u)$ and hence for $q(B_{\text{out}}, \alpha, t)$ is found by integrating Eq. (11) (using the Mathematica computer program [7]):

$$\begin{aligned} q(B_{\text{out}}, \alpha, t) = \frac{1}{3\,\mu_0 \mathcal{F}(B_{\text{out}})\mathcal{H}} \Bigg[& B_{\text{out}} \cos(\alpha) \left(2\sqrt{B_{\text{out}}} \right. \\ & + \sqrt{B_{\text{out}} |\sin(\alpha)|} \; {}_2F_1 \left. \left(\frac{1}{2}, \frac{3}{4}; \frac{3}{2}; -\cot^2 \alpha \right) \right) \\ & + (t - B_{\text{out}} \cos \alpha) \left(2\sqrt[4]{B_{\text{out}}^2 - 2B_{\text{out}} t \cos \alpha + t^2} \right. \\ & + \sqrt{B_{\text{out}} |\sin \alpha|} \; {}_2F_1 \left. \left(\frac{1}{2}, \frac{3}{4}; \frac{3}{2}; -\frac{(t - B_{\text{out}} \cos \alpha)^2}{\sin^2 \alpha B_{\text{out}}^2} \right) \right) \Bigg], \end{aligned} \tag{12}$$

where ${}_2F_1$ denotes the Gauss' Hypergeometric function. The algorithm for the implementation of ${}_2F_1$ in the program language C was based on [9].

A system of equations can now be established for the problem of finding a relative penetration depth q^* that fulfills the Eqs. (2) and (3) or, equivalently, the problem of finding α_{new}, $|\vec{t}_{\text{new}}(q^*)| = t_{\text{new}}$ and $|\vec{t}_{\text{old}}(q^*)| = t_{\text{old}}$. Obviously it is required that

$$q(B_{\text{new}}, \alpha_{\text{new}}, t_{\text{new}}) = q(B_{\text{old}}, \alpha_{\text{old}}, t_{\text{old}}) = q^*. \tag{13}$$

Moreover, the continuity equation (3) yields

$$B_{\text{new}} \begin{pmatrix} \cos \beta_{\text{new}} \\ \sin \beta_{\text{new}} \end{pmatrix} + t_{\text{new}} \begin{pmatrix} -\cos(\alpha_{\text{new}} - \beta_{\text{new}}) \\ \sin(\alpha_{\text{new}} - \beta_{\text{new}}) \end{pmatrix}$$

$$= B_{\text{old}} \begin{pmatrix} \cos \beta_{\text{old}} \\ \sin \beta_{\text{old}} \end{pmatrix} + t_{\text{old}} \begin{pmatrix} -\cos(\alpha_{\text{old}} - \beta_{\text{old}}) \\ \sin(\alpha_{\text{old}} - \beta_{\text{old}}) \end{pmatrix} . \tag{14}$$

Given the quantities B_{new}, β_{new}, α_{old}, B_{old}, and β_{old}, the system of equations (13) and (14) can be solved for the unknowns α_{new}, t_{new} and t_{old} by means of a Newton algorithm.

Eventually, to find the distribution of the magnetic induction in the filament cross-section, see Eq. (9), the inverse relation of Eq. (12), $t(B_{\text{out}}, \alpha, q^*)$, is required. It is obtained from the Newton algorithm, using K recursions of the form

$$t_{k+1} = t_k - \mu_0 \mathcal{F}(B_{\text{out}}) \mathcal{H} \frac{q(B_{\text{out}}, \alpha, t_k) - q^*}{\sqrt[4]{B_{\text{out}}^2 - 2B_{\text{out}} t_k \cos(\alpha) + t_k^2}}, \tag{15}$$

with $k = 0, \ldots, K$ denoting the index of the iteration step. An appropriate starting value t_0 is required.

From Fig. 3 it is easy to see that N field changes on the outside of the filament effect the creation of $n \leq N$ distinct shielding-current layers between the relative penetration parameter values q_i and q_{i+1}, $1 \leq i \leq n$. The indices i and $i + 1$ correspond to what previously has been subscripted as 'old' and 'new'.

Provided the semi-analytic expression for the magnetic induction $B(q)$ inside the filament, the magnetization due to a layer of shielding currents of critical density $j_c(B(q))$ can be derived. The vector of the entire filament's magnetization $\vec{M}$ equals the geometric sum of the magnetization vectors $\vec{M}_i$ generated by the individual current layers,

$$\vec{M} = \sum_{i=1}^{n} \vec{M}_i = \sum_{i=1}^{n} \int_{q_i}^{q_{i+1}} \vec{m}(q) \, \mathrm{d}q, \tag{16}$$

where $\vec{m}(q)$ denotes the magnetization contribution of a shielding current layer of relative thickness $\mathrm{d}q$ at q.

The direction of $\vec{M}_i$ follows the direction of the shielding vector $\vec{t}_i(q)$, subsequently denoted $\vec{e}_{t_i}$. We obtain for the magnetization of one shielding-current layer

$$\vec{M}_i = \frac{4r}{\pi} \int_{q_i}^{q_{i+1}} j_c(B(q))(1 - q)^2 \, \vec{e}_{t_i} \, \mathrm{d}q$$

$$= \frac{4r\mathcal{F}(B_{\text{out}})}{\pi} \int_{q_i}^{q_{i+1}} \frac{(1 - q)^2}{\sqrt{B(q)}} \vec{e}_{t_i} \, \mathrm{d}q, \tag{17}$$

where $B(q)$ is given in Eq. (9). Generally, Eq. (17) is evaluated numerically. An analytical approximation exists for outer fields being substantially larger than the shielding induction, $|(\vec{B}_{\text{out}})_i| \gg |\vec{t}_i(q_{i+1})|$ which is used whenever possible in order to accelerate the function evaluation.

4. Simulations of a single filament subjected to field changes

The computations in this section are presented for one filament with a radius $r = 3.5 \ \mu\text{m}$ at $T = 1.9 \, \text{K}$. The result is scaled for a strand with a filling factor $\lambda = 1/2.95$ which takes into account the copper to superconductor ratio of the multi-filament wire. Figure 4 presents computations of 1-dimensional field changes between $B_{\text{out}} = 0, 2$ and -2 T. The

model reproduces the typical hysteretic behavior of superconducting filaments, as discussed in [1].

Note that the so-called virgin curve joins the hysteresis slope in the point where the outer field has fully penetrated the filament cross-section (and which is not the point of maximum magnetization). At every point the magnetization vector $\vec{M}$ opposes the applied field change. Consequently the lower branch of the hysteresis loop (up-ramp) is called the shielding branch and the upper branch (down-ramp) is called the trapping branch. It can also be seen that the filament magnetization increases as the outer field tends to zero and decreases for larger outer inductions. This effect is due to the B-dependence of the critical current density $j_c(B, T)$. The fact that the actual peak of the magnetization is shifted from $B = 0$ is discussed in detail in [1].

A similar case for the vector magnetization model is shown in Fig. 6. The left hand side diagram shows a clockwise rotating excitation field that goes through the origin in the B_x/B_y-plane ($\vec{B}_{out} = \vec{0}$). As for the first excitation steps, the filament's response (shown in the right diagram of Fig. 6) is very similar to the 1-dimensional field change. The magnetization opposes exactly the applied field. As the outer field further increases in absolute value, the magnetization decreases due to the field dependence of the critical current density. At full penetration the magnetization curve meets the closed 2d-hysteresis loop. The maximum of $|\vec{M}|$ is reached at the completion of each turn in the excitational field, where $|\vec{B}_{out}|$ is zero.

5. Combining the model with numerical field calculation

For the calculation of field errors in superconducting accelerator magnets with a ferromagnetic yoke, the magnetization model is combined with the coupling method between boundary and finite elements BEM-FEM which accounts for the local saturation effects in the yoke,
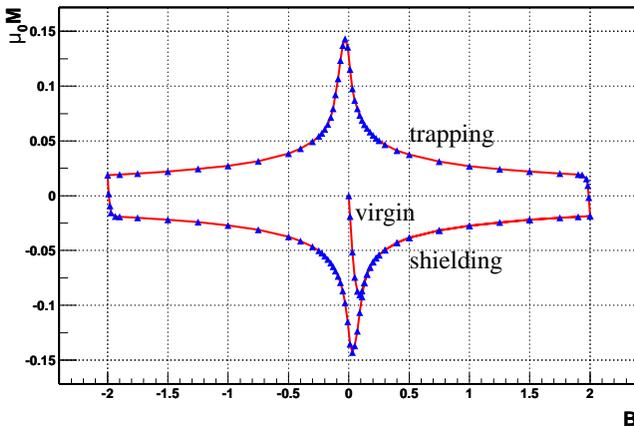
Figure 5. Computed magnetization curve for one filament, scaled by the filling factor λ ($r = 3.5\ \mu\mathrm{m}$, $\lambda = 1/2.95$, $T = 1.9\ \mathrm{K}$). The excitational field is represented on the x-axis, the corresponding magnetization on the y-axis. Note that the maximum of the magnetization is not exactly at zero excitation but slightly shifted from it. This effect is called "peak-shifting" and is confirmed by measurements.
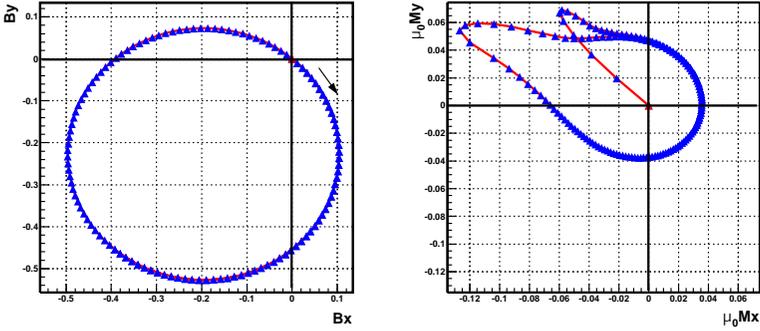
Figure 6. Rotational field excitation of one filament, scaled by the filling factor λ ($r = 3.5\ \mu$m, $\lambda = 1/2.95$, $T = 1.9$ K). Two complete turns have been computed. Left: The external field follows a circular, clockwise path that goes through the origin in the B_x/B_y-plane. Right: The filament's response in terms of magnetization; A virgin curve and a closed hysteresis slope can be identified.

[5]. The BEM-FEM formulation has the major advantage that the coil does not have to be meshed in finite elements since it is positioned in the iron-free BEM domain. The coil can therefore be modeled with the required accuracy taking insulation layers, cable keystoning, and conductor placements into consideration. The fields arising from the source currents in the superconducting coil and those resulting from the induced superconductor magnetization can be calculated analytically by means of the Biot-Savart Law. The field generated by the magnetization of the superconducting strands at each strand position is added to the computed sum of the source fields from the superconducting coil and the contributed magnetic induction which results from the non-linear iron yoke.

In order to calculate global shielding effects in the coil (in particular at low excitational field) the feed back of the persistent currents on the excitational field is calculated by means of an $\vec{M}(\vec{B})$-iteration on the strand level. For that, an algorithm is implemented, that controls the convergence of the x- and y-components of the magnetization vectors seperately, since convergence in magnitude only does not exclude rotation. The combination of the magnetization model with numerical field computation follows the method described in [10] for the 1-dimensional model.

Fig. 7 shows the field plots of the combined octupole decapole spool-piece corrector (MCDO) for the LHC. A somewhat academic excitational cycle is considered (which will serve the purpose of validating the model with measurements). First the decapole is ramped up to 0.25 of its nominal field value. Then the octupole is powered up to its nominal field value (100 A) which creates the asymmetric field in the magnet. Then the octupole field is ramped repeatedly up and down between +100 and -100 A. The latter stages are displayed in Fig. 7. This excitation creates a field change in a particular strand which is displayed on the left hand side of Fig. 8 together with the resulting magnetization on the right.

Fig. 9 shows the B_3 and B_7 field component for the MCDO magnet with the ramp cycle described as above. The field errors are the Fourier series components (in Tesla) of the radial field at a reference radius r_0 of 17 mm. The field in the aperture can be calculated from the
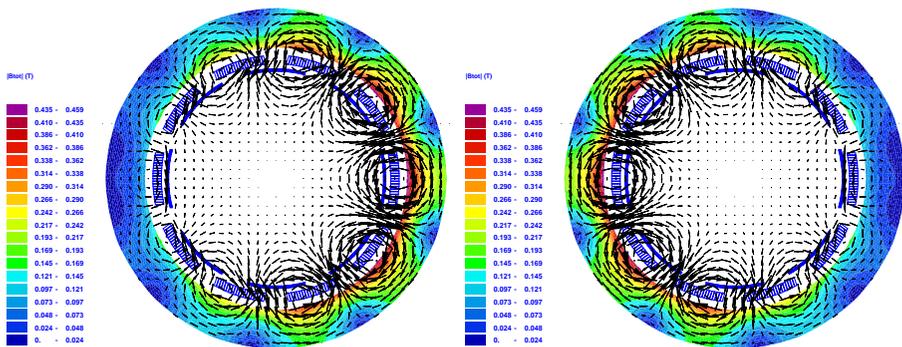
Figure 7. Field plots for the different excitations of the combined decapole octupole corrector magnet. First the decapole (outer layer coil) is ramped up to about 0.25 of its nominal field value. Then the octupole (inner layer coil) is powered up to its nominal field value (100 A) which results in the assymmetric field in the magnet as displayed on the left hand side. Subsequently the octupole field is ramped up and down between +100 and -100 A. This is a somewhat academic excitational cycle but it will serve the purpose of validating the model with measurements.

relation

$$B_y + iB_x = \sum_{n=1}^{\infty}(B_n + iA_n)\left(\frac{z_0}{r_0}\right)^{n-1}, \tag{18}$$

where $z_0 = x_0 + iy_0$ [6]. The B_n are called the normal and the A_n the skew components. Note that, for symmetry reasons, the numerical calculation of the MCDO magnet's field without the persistent current effects does not have any B_3 or B_7 component.
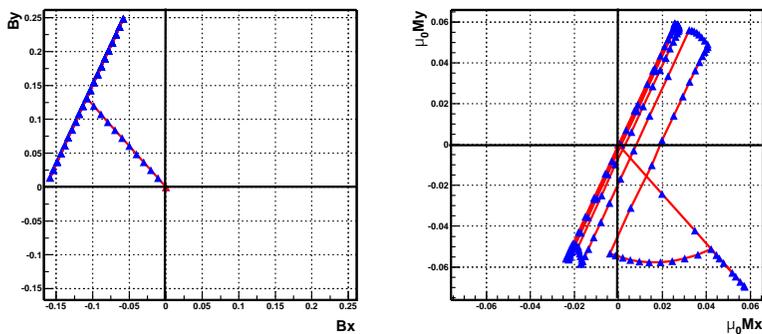
Figure 8. Left: Excitational field of one filament in the MCDO combined corrector magnet. Right: The filament's response in terms of magnetization. The filament in question is situated at the "two o'clock" position in the outer coil (decapole), compare Fig. 7
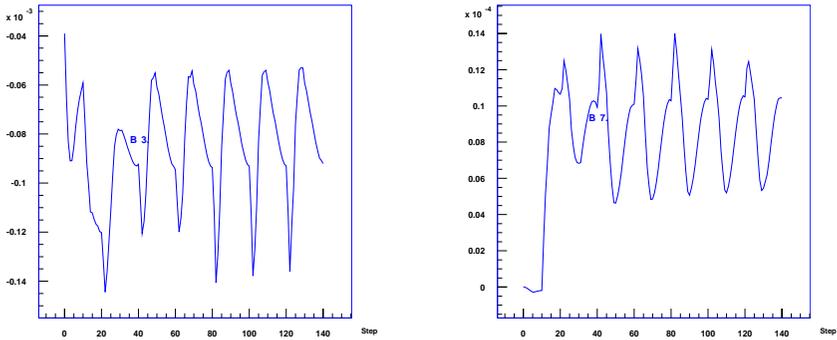
Figure 9. Field errors (left B_3 right B_7 in Tesla at a reference radius of 17 mm) as a function of the excitation step. First the decapole (outer layer coil) is ramped up to about 0.25 of its nominal field value (step 10). Then the octupole (inner layer coil) is powered up to its nominal field value (100 A, step 20). Subsequently the octupole field is ramped up and down between +100 (step 20, 60, 100) and -100 A (step 40, 80, 120).

6. Conclusions

A vector magnetization model for superconducting multi-filamentary wires in the coils of accelerator magnets has been developed. It describes arbitrary excitational cycles, in particular also excitations for which the magnetization is not parallel to the external field. The model has been combined with numerical field computation for the calculation of field errors in magnets with nested coil geometries and local saturation of the ferromagnetic yoke. The material related input parameter is only the critical current density (which can be measured on the strand level). Arbitrary excitational cycles can now be studied and optimized for machine operation.

References

[1] M. Aleksa, S. Russenschuck and C. Völlinger *Magnetic Field Calculations Including the Impact of Persistent Currents in Superconducting Filaments* IEEE Trans. on Magn., vol. 38, no. 2, 2002.
[2] C.P. Bean, *Magnetization of High Field Superconductors*, Review of Modern Physics, vol. 36, 1964.
[3] R. A. Beth, *An Integral Formula for two-dimensional Fields*, Journal of Applied Physics, vol. 38/12, Nov 1967
[4] L. Bottura, *A Practical Fit for the Critical Surface of NbTi*, 16th International Conference on Magnet Technology, Florida, 1999
[5] S. Kurz and S. Russenschuck, *The Application of the BEM-FEM Coupling Method for the Accurate Calculation of Fields in Superconducting Magnets*, Electrical Engineering - Archiv für Elektrotechnik, vol. 82, no. 1, Berlin, Germany, 1999.
[6] The LHC study group, *The Yellow Book, LHC, The Large Hadron Collider - Conceptual Design*, CERN/AC/95-5(LHC), Geneva, 1995.
[7] Wolfram Research, *Mathematica 4.1*
[8] M. Pekeler et al., Coupled Persistent-Current Effects in the Hera Dipoles and Beam Pipe Correction Coils, Desy Report no. 92-06, Hamburg, 1992
[9] W.H. Press et al., *Numerical Recipes in C: The Art of Scienti c Computing*, 2nd edition, Cambridge University Press, 1992
[10] C. Völlinger, M. Aleksa and S. Russenschuck, *Calculation of Persistent Currents in Superconducting Magnets*, Physical Review, Special Topics: Accelerators and Beams, IEEE, New York, 2000
[11] M.N. Wilson, *Superconducting Magnets*, Monographs on Cryogenics, Oxford University Press, New York, 1983.

A simple parallelization of a FMM-based serial beam-beam interaction code

Paul M. Alsing†§, Vinay Boocha†
Mathias Vogt⋆, James Ellison‡ and Tanaji Sen⋄

† Albuquerque High Performance Computing Center,
‡ Department of Mathematics and Statistics
University of New Mexico, Albuquerque NM, 87131
⋆ DESY/MPY, Notkestr. 85, 22607 Hamburg, FRG
⋄ Beam Physics, Fermi National Accelerator Laboratory, Batavia, IL

1. Introduction

The essence of collective simulations of the beam-beam interaction in particle colliders is (1) the propagation of the particles in two counter-rotating bunches between adjacent interaction points (IP) through external fields and (2) the computation of the collective electromagnetic forces that the particles in one bunch exert on each of the particles of the other bunch, localized at the interaction points. The first part is straight forward and can be naively parallelized by assigning subsets of particles to different processors, and propagating them separately. The second part encompasses the core computational challenge which entails the summation of the collective effects of the particles in one beam on the individual particles in the other beam. Typically, for high energy colliders the computation of the collective forces is reduced to a 2 dimensional Poisson problem. If there are N particles in each beam, a brute force calculation involves $\mathcal{O}(N^2)$ computations. For moderate size N, this quickly becomes a prohibitively expensive operation. Fast Multipole Methods (FMM) have been used to reduce the number of calculations to $\mathcal{O}(N)$ [1, 2, 3]. These methods invoke Taylor expansions of forces due to multipole expansions of distant localized groups of particles. At large distances, target particles interact with collective sources, while at near distances, individual particle-particle interactions $(<< N^2)$ are computed directly. In order to perform this computation efficiently, a hierarchal gridded domain structure must be maintained.

FMMs have been parallelized and involve a quad or octet tree structure in which the fine grain "leaves" of the tree record the location of particles in a given region of space [4]. In this spatial domain-decomposition approach, it is crucial that between evolution time steps, the particles in one collection of leaves (owned by a given processor) do not migrate substantially over the whole spatial domain. In such cases, each processor only

§ To whom correspondence should be addressed (alsing@ahpcc.unm.edu)

has to communicate with a suitably defined neighborhood of "nearby" processors and involves the interchange of only a small number of particles close to the boundaries of the domains. However, in beam-beam calculations for circular colliders, this condition is violated, since the propagation of the particles through the external fields between beam-beam interactions potentially causes them to wander over the entire spatial domain. Thus current parallel implementations of FMM are inadequate for such simulations.

In this work we describe a very simple parallelization of a FMM based beam-beam interaction code. We continue to use a serial FMM algorithm, but embed it in a clever parallel communication scheme called Force Decomposition (FD) [5], initially developed for molecular dynamics simulations. Without the FMM, each processor in the FD algorithm would perform an N'^2 brute force calculation, where $N' = N/\sqrt{P}$, with P the total number of processors used. By replacing this brute force computation with a serial FMM algorithm on each processor we reduce the computational complexity to $\mathcal{O}(N') = \mathcal{O}(N/\sqrt{P})$. The advantage of this combined FMM/FD parallel method is that it is extremely easy to implement. The FD method itself involves two basic MPI routines, each called twice per interaction, and basic FMM calculations are performed exactly as in a serial based FMM code, just now on a reduced set of source and target particles.

This paper is organized as follows. In section 2 we discuss the beam-beam ring model. In section 3 we discuss the macro particle tracking method for the beam-beam interaction, and in section 4 a serial implementation of the FMM for the beam-beam kick calculations. In section 5 we discuss the parallel force decomposition (FD) communication scheme. In section 6 we discuss our parallelization of the previous serial code which we denote as FD/FMM. Finally in Section 7 we summarize our findings and indicate directions for future work to increase the order of this speedup.

2. The ring model

We assume a ring with one IP at $\theta = 0$ and two counter-rotating bunches. In this work we only treat head-on collisions , and the reference point at which the distribution is studied is directly *before* the IP ($\theta \bmod 2\pi = 0^-$). In what follows we use the convention that if some parameter or dynamical variable X describes one beam, then X^* describes the other beam.

Let $\psi_n(z)$ and $\psi_n^*(z)$ ($z := (q, p), q = (x, y), p = (p_x, p_y)$) denote the normalized phase space densities of the beams at $\theta = 0^- + 2n\pi$. Then the representation of the one turn map for the unstarred beam from turn n to $n + 1$ is

$$T[\psi_n^*] = A \circ K[\psi_n^*] \tag{1}$$

where A represents the lattice without the beam-beam and $K[\psi]$ is the collective beam-beam kick. In highly relativistic beams the beam-beam force for head-on bunch crossings with zero crossing angle is essentially transverse (the space charge forces within each

bunch are negligible). In this approximation, the collective kick is given by

$$K[\psi](q,p) = \begin{pmatrix} q \\ p + k[\psi](q) \end{pmatrix} \ , \ k[\psi](q) = \zeta \int_{\mathbb{R}^2} g(q - q')\rho(q') \, dq', \qquad (2)$$

where $g = \nabla G$, $G(q) = -\frac{1}{2\pi} \log\left(\sqrt{x^2 + y^2}\right)$ is the Green's function, ζ is a strength parameter and $\rho(q) = \int_{\mathbb{R}^2} \psi(q, p) \, dp$ is the spatial density.

The particle trajectories are propagated turn by turn via

$$z_{n+1} = T[\psi_n^*](z_n) \ , \ z_{n+1}^* = T^*[\psi_n](z_n^*) \qquad (3)$$

and since the maps are measure preserving, the densities evolve via

$$\psi_{n+1}(z_{n+1}) = \psi_n(z_n) \ , \ \psi_{n+1}^*(z_{n+1}^*) = \psi_n^*(z_n^*) \ . \qquad (4)$$

Note that the T and T^* are explicitly distinguished, allowing for different parameter sets describing the starred and the unstarred lattice. Equations (1-4) define a representation of the beam–beam Vlasov-Poisson system using maps.

3. Macro particle tracking

Macro Particle Tracking (see also [2]) is a method for computing time dependent phase space averages of f

$$\langle f \rangle_n := \int_{\mathbb{R}^4} f(z)\psi_n(z) \, dz, \ \langle f \rangle_n^* := \int_{\mathbb{R}^4} f(z)\psi_n^*(z) \, dz, \qquad (5)$$

which can be evaluated as, for example,

$$\langle f \rangle_n = \int_{\mathbb{R}^4} f(z) \, \psi_0\left(M_n^{-1}(z)\right) \, dz \ = \int_{\mathbb{R}^4} f\left(M_n(z)\right) \psi_0(z) \, dz, \qquad (6)$$

where $M_n := T[\psi_{n-1}^*] \circ \ldots \circ T[\psi_0^*]$ is the symplectic n-turn map containing successive collective kicks. Note that the beam-beam kick function can be written as such an average over the beam-beam kernel with q fixed, $k[\psi_n^*](q) = \zeta \ \langle g(q - \cdot) \rangle_n^*$. The macro particle concept enters through approximating the phase space integrals by finite sums over macro particle trajectories. A popular implementation for high dimensional problems is Monte Carlo macro particle tracking (MCMPT) where an initial ensemble of *identical* macro particles is generated randomly at positions z_k according to the initial density ψ_0. Then $\langle f \rangle_n \approx \frac{1}{N} \sum_k f(M_n(z_k))$. Of course, if we approximate the kicks in Eq. (2) by this method then we have approximate trajectories $\eta_k(n) \approx M_n(z_k)$. Thus our final approximation is

$$\langle f \rangle_n \approx \frac{1}{N} \sum_k f(\eta_k(n)) \ . \qquad (7)$$

This will be our procedure in this report. A refined method which captures the behavior outside the beam core more accurately is *weighted* macro particle tracking (WMPT) [2]. Here we start with the initial densities ψ_0 and ψ_0^* defined on an *initial mesh* $\{z_k\}$ and a quadrature formula with weights w_k. Using WMPT to update the approximate trajectories $\eta_k(n)$, the final approximation becomes $\langle f \rangle_n \approx \sum_k f(\eta_k(n))\psi_0(z_k)w_k$.

These procedures use only forward tracking of particles and conservation of probability is guaranteed by construction. In contrast to methods with an explicit mesh (like PF in [2]), naive computation of the collective kick is an $\mathcal{O}(N^2)$ operation. The fast multipole method [1] reduces this to $\mathcal{O}(N)$, albeit with a large order constant.

4. A serial implementation of FMM for the beam-beam kick

The fast multipole method (FMM) [1] is a tree code that allows the computation of the collective force of an ensemble of N charges on themselves to a given accuracy δ with an operations count $\mathcal{O}(N)$ given that the distribution of the ensemble in configuration space is not too irregular. It employs the fact that the force on a particle due to a distant localized "clump" of charge can be given by a *finite order* multipole expansion up to precision δ.

The FMM algorithm successively subdivides an outer rectangle in configuration space occupied by the ensemble until, on the finest level of subdivision, no more than a fixed number (typically 40) of particles are in each box. This leads to a tree structure of boxes containing boxes containing boxes and so on, until the boxes on the finest level finally contain a small number of particles. In the non-adaptive version of the scheme all boxes on the same level have the same number of child boxes. We use the adaptive version in which a box only branches into child boxes if the box itself still contains too many particles. In the upward traversal of the tree (fine → coarse levels) the algorithm first computes the multipole (long distance) expansions for all boxes on the finest level explicitly. The next step is to generate multipole expansions around the center of the parent boxes by translating their children's expansions to the center and adding them up, see Fig.(1). This is repeated at every level so that in the end every box, at every level of mesh refinement, has its own long distance expansion.

The downward traversal of the tree (coarse → fine levels) begins with boxes that are separated from one another by more than one box (i.e. at the 3-rd coarsest level). The potential inside a given box due to the sum of the multipole expansions of all well separated boxes is converted into a Taylor polynomial around the center of that box. Taylor series around the centers of the boxes in the next finer level are generated by shifting the Taylor polynomials of the parent boxes to the centers of their children (Fig.(1)) *and* adding the contributions of the multipole expansions from the well separated boxes (in the child level) which now show up at the boarder of the well separated parents. This procedure is iterated until the finest level. Finally, for each box on the finest level the forces on the particles are computed by evaluating the explicitly given derivative of its local Taylor polynomial at the position of the particle plus a small number of direct Coulomb terms.

The 2-D adaptive routines (`DAPIF2`), used in our simulations, were supplied by Greengard. For more details see Greengard in [1].

FMM was originally invented for dynamics such as space charge where the force of N particles *on themselves* is computed. The beam-beam effect on the other hand consists of
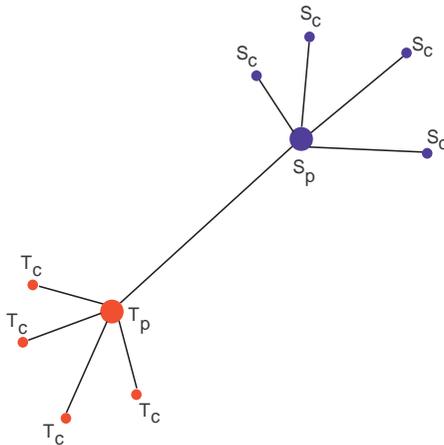
Figure 1. Schematic representation of FMM algorithm (see text). Upward sweep through boxing structure tree (fine $\rightarrow$ coarse levels): Far field multipole expansions (MPE) of individual charges are summed and evaluated at the center of source-child boxes S_c. The MPE of source-children boxes are summed and evaluated at the center of corresponding parent box S_p. Downward sweep through tree (coarse $\rightarrow$ fine levels): The MPE of far sources are evaluated at the center of a target parent box T_p as local Taylor expansions. These local Taylor expansions are shifted and evaluated at centers of target children boxes T_c. At the finest level, forces can be computed on individual target particles.

the collective interaction among two distinct counter-rotating bunches without taking into account the space charge forces *inside* each bunch. In other words one has to compute the forces on N particles of the unstarred bunch exerted by the N^* particles of the starred bunch, *and then in addition* the forces on N^* particles in the starred bunch due to the N particles in the unstarred bunch. Let us assume for simplicity that $N = N^*$. The forces can then be computed at 4 times the computational expense of the space charge force calculations inside a single bunch. First, the $2N$ particles of both beams are considered as *one* ensemble, with the charges of the starred bunch unchanged and the charges of the unstarred bunch set to zero. Then FMM computes the "space charge" force of the $2N$ particles (half of them being uncharged) at the $2N$ spatial coordinates. The kick proportional to the force is applied, however, only to the particles of the unstarred bunch. Then the charges of the unstarred particles are set back to their original values and the charges of the starred particles are set to zero. After having FMM compute the new "space charge" forces, the kicks are applied to the particles of the starred beam only. Thus FMM is called twice with twice as many particles.

5. The parallel force decomposition algorithm

In this section we discuss the Force Decomposition (FD) algorithm as originally envisioned for MD simulations. In the next section we discuss our adaption of the FD algorithm for the beam-beam kick calculations utilizing FMM.

The Force Decomposition (FD) algorithm was developed in 1995 by Plimpton and Hendrickson [5] originally as a parallel method for matrix-matrix multiplication without All-to-All communications (e.g. as used in 2D parallel FFT algorithms when all processors must communicate with each other at the same time during the transpose operation). The authors realized that this communication scheme (described below) could be adapted to short range force molecular dynamics simulations where two-body (and higher multi-body and bonded) forces could be computed more efficiently, over that of the more conventional spatial decomposition algorithms, for particle numbers in the range of $N \sim 10^5 - 10^6$. In molecular dynamics (MD) simulations, the main computational task involves the calculation of the force on particle i due the rest of the N particles, $\mathbf{F}_i = \sum_{j=1}^{N} \mathbf{F}_{ij}$. A brute force serial calculation (which is to be avoided at all costs) would involve an $\mathcal{O}(N^2)$ double-do-loop calculation of the form:

$$
\begin{aligned}
&\texttt{do} \quad i = 1,\, N \\
&\qquad \texttt{do } j = 1,\, N \\
&\qquad\qquad \mathbf{F}_i \leftarrow \mathbf{F}_i + \mathbf{F}_{ij} \\
&\qquad \texttt{enddo} \\
&\quad \texttt{enddo} \ .
\end{aligned}
\tag{8}
$$

The so called Replicated Data (RD) algorithm reduces the order count by distributing the particles amongst the P processors. Thus the outermost (i) do-loop is distributed amongst the processors and each processor computes the forces on its own N/P particles and then updates the position and momentum of these particles. However, in order to perform the next calculation, all N updated particle positions would need to be communicated to all processors; a communication operation that scales as N. A measure of the scalability of an algorithm is *isoscalability IS*, which is the ratio of the communication cost to the computation cost. An ideal algorithm has $IS = 1$, which indicates that larger problems can utilize larger resources in the most efficient manner, i.e. the communication and computation costs scale commensurately. Unfortunately, the RD algorithm has $IS = P$, which implies that the efficiency of the algorithm decreases with the number of processors P, since the communication cost scales as P times the computation cost.

The most common parallel algorithm used in molecular dynamics force calculations is Spatial Decomposition (SD). In SD, the physical domain is divided amongst the P processors, with each processor responsible for the particles within its sub-volume. Thus, computation cost scales as the volume or $V \sim N/P$. For problems with short range potentials which can be treated as negligible after some finite cutoff radius r_c, the communication cost scales as the area $A = V^{2/3} \sim (N/P)^{2/3}$ in the limit of large N/P.

If each processor's sub-volume is on the order of r_c^3, communication takes place between the processor and its nearest neighbors across its boundary areas and the communication cost can approach N/P asymptotically for large N/P. Thus, theoretically $IS \to 1$ for SD, though there can be impediments to reaching this limit [5].

Note that RD divides up the particles while SD divides up the physical domain. RD has the advantage of being geometry free, with ease of coding for any arbitrary dimension of the physical coordinates (i.e. z a 2D, 3D, vector, etc ...), though with the marked disadvantage of poor scalability. The surface area to volume ratio of SD communications makes it worth the added effort of coding for particles sizes typically greater than 10^6.

Force Decomposition (FD) has an isoscalability that lies between RD and SD and has been shown by Plimpton and Hendrickson [5] to out perform SD in the realm of under 10^6 particles. As shown in Fig.(2), FD parallelizes the force matrix $\mathbf{F}_{ij}$ by
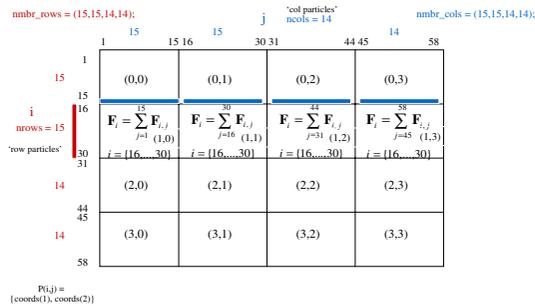
Figure 2. Parallel force decomposition (FD) of force matrix $\mathbf{F}_{ij}$ on a 2D processor grid. Each processor owns approximately $\mathcal{O}(N/\sqrt{P})$ i-particle indices and $\mathcal{O}(N/\sqrt{P})$ j-particle indices. A given processor, e.g. P_{12} computes the *partial* forces on all its i-particles (here $16 - 30$) due to all its j-particles (here $31 - 44$).

creating a processor grid which is two dimensional regardless of the dimension of the coordinates $z = (q, p)$. Each processor, P_{rc}, is responsible for approximately $N/\sqrt{P}$ target or i-particles and $N/\sqrt{P}$ source or j-particles and thus can locally compute the partial forces $\mathbf{F}_i = \sum_{j=1}^{N/\sqrt{P}} \mathbf{F}_{ij}$,

$$\text{do} \quad i = 1, N/\sqrt{P}$$
$$\text{do } j = 1, N/\sqrt{P}$$
$$\mathbf{F}_i \leftarrow \mathbf{F}_i + \mathbf{F}_{ij}$$
$$\text{enddo}$$
$$\text{enddo .} \tag{9}$$

Here the i and j are local indices in a given processor (Note: Fig.(2)-Fig.(4) are labeled with global indices). With the use of neighbor lists (as used in SD) to tag the $\mathcal{O}(N/P)$ non-zero interactions in $\mathbf{F}_i$, the computation cost scales as $\mathcal{O}(N/P)$. Though the FD algorithm can be coded for a non-square number of processors, we discuss it here for a square number. The total number of particles N need not be a multiple of the the number of processors P, as is illustrated in Fig.(2) (which uses $N = 58$ and $P = 16$ for ease of discussion).
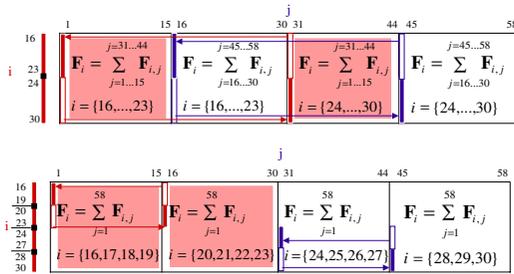
Figure 3. Summation of total forces in a processor grid-row by *recursive halving*(see text). Each processor obtains the total forces on approximately $\mathcal{O}(N/P)$ i-particles in $\log_2(N/\sqrt{P})$ steps. (For processor P_{12} these are i-particles $\{24, 25, 26, 27\}$). This operation is implemented via a call to `MPI_REDUCE_SCATTERV`.

Within each row of the processor grid the goal is to obtain the total force $\sum_{j=1}^{N} \mathbf{F}_{ij}$ for each of the $N/\sqrt{P}$ i-particles in that row. This is accomplished by the following clever *recursive halving* communication scheme, depicted in Fig.(3) for the second processor row of Fig.(2). At the first stage, each processor halves the number of partial forces it has for its i-particles and sends them to a target processor in the same grid-row (top half of Fig.(3)). At the same time this processor receives half of the target processor's partial forces. For example, processor $(1, 0)$ sends its partial forces for $24 \leq i \leq 30$ to processor $(1, 2)$ while receiving the partial forces for $16 \leq i \leq 30$ from this processor. After this first stage, each processor in the grid-row now owns a more complete (yet not fully complete) set of partial forces, but now for only half of its original i-particles. This process is now repeated in $\log_2(N/\sqrt{P})$ stages until each processor in the grid-row owns the total force $\sum_{j=1}^{N} \mathbf{F}_{ij}$, for N/P i-particles. The bottom half of Fig.(3) illustrates the second stage of this process. This recursive halving entails a communication cost that scales as $\mathcal{O}(N/\sqrt{P})$ and is accomplished in MPI by the call `MPI_REDUCE_SCATTERV` with an `MPI_SUM` reduction operation. At this stage each processor updates its N/P coordinates for which it has the total forces (i.e solve $\ddot{\mathbf{x}}_i = \mathbf{F}_i/m_i$).
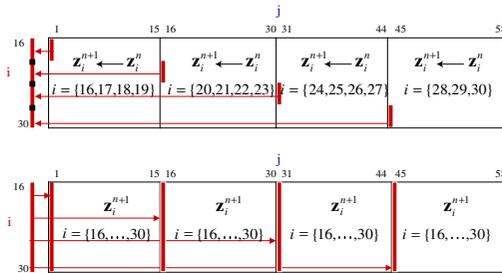
Figure 4. Communication of updated particles positions in a processor grid-row in $\log_2(N/\sqrt{P})$ steps by *recursive doubling* (see text). This operation is implemented by a call to `MPI_ALLGATHERV`.

To return the data to its initial configuration, in order to begin the next calculation, FD invokes a *recursive doubling* of the data within a processor grid-row, Fig.(4) (i.e. the opposite of the recursive halving operation described above). This operation is accomplished by `MPI_ALLGATHERV` and upon its completion, each processor in the grid-row has the updated coordinates for all of its $N/\sqrt{P}$ i-particles.

So far, the FD communication scheme has taken place only within processor grid-rows, with no inter-grid-row communications. However, while the recursive doubling operation above has updated all the i-particle coordinates within a grid-row, each processor still requires the updated coordinates for its j-particles in order to begin the calculations of the partial forces anew. An inspection of Fig.(2) reveals that the *transpose* processor P_{cr} (when we utilize a two-digit designation of the processors in a 2D processor grid) has precisely the i-particles that processor P_{rc} requires for use as its own j-particles. Thus, the last step in the FD communication scheme is for these two processors to perform a point-to-point communication (via `MPI_SENDRECEIVE`) to swap copies of their respective i-particles. Upon completion of this operation, the data has now returned to its original form and we are ready to start a new partial force calculation, as in Fig.(2).

Since the computation cost scales as $\mathcal{O}(N/P)$ and the communication cost scales as $\mathcal{O}(N/\sqrt{P})$, the FD algorithm has an $IS = \sqrt{P}$, which is intermediate between RD and SD. FD retains the ease of coding and the geometry features of RD with a intermediate isoscalability. As $N/\sqrt{P}$ grows large, the double-do-loop in Eq.(9) grows prohibitively large enough to warrant a transition to the SD algorithm with a near optimal isoscalability. As stated earlier, Plimpton and Hendrickson [5] have shown that FD out performs SD in the particle range $N \sim 10^5 - 10^6$.

6. Parallel force decomposition with serial fast multipole method

Our simple parallelization scheme for the beam-beam interaction is now easily described. We employ a FD parallel communication strategy where the on-processor partial force computations in Eq.(9) are replaced by a serial FMM calculation. As pointed out earlier, in the beam-beam interaction the row i-particles and column j-particles in Fig.(2) are in separate beams, say the unstarred and starred beam, respectively. Thus, on each processor, we first compute the total force on the i-particles due to all the j-particles (using FMM) and secondly compute the total force on the j-particles due to all the i-particles (with a second call to FMM). Once the total force on all the i-particles are known we can apply the momentum kick to each of the i-particles and then apply the symplectic map to the next IP (and similarly for the j-particles in the starred beam). What differs from the FD algorithm as discussed in the previous section, is that each beam does not need the updated coordinates of the other beam. This means that we can drop the `MPI_SENDRECEIVE` between transpose processors. Instead, between each IP our algorithm utilizes the pair of routines `MPI_REDUCE_SCATTERV` and `MPI_ALLGATHERV`, (Fig.(3) and Fig.(4)) in each processor grid-row for the accumulation of forces for the i-particles, and then another pair of calls `MPI_REDUCE_SCATTERV` and `MPI_ALLGATHERV` in processor *grid-columns* for the accumulation of forces for the j-particles. We denote this parallelization method FD/FMM. While serial FMM reduces an $\mathcal{O}(N^2)$ computation to $\mathcal{O}(N)$, FD/FMM reduces an $\mathcal{O}((N/\sqrt{P})^2)$ computation to $\mathcal{O}(N/\sqrt{P})$, yielding a $\sqrt{P}$ speedup.

As a proof of principal we have written a test code implementing FD/FMM which functions as follows:

1) Every processor sets up the two lattice maps.

2) Two Monte Carlo ensembles representing the 2 bunches are generated by computing $2N$ times 4 independent Gaussian random variables in transverse phase space and distributing e.g. the unstarred particles along the rows and the starred particles along the columns of the processor grid.

3) The collective beam-beam kicks, starred on unstarred, are computed using the FD/FMM approach along the columns and then along the rows. Here FMM is called once on $2N/\sqrt{P}$ particles in the row cycle and then once again on the $2N/\sqrt{P}$ in the column cycle.

4) Each processor applies the single particle lattice maps A and A^* for its portion of the starred and unstarred particles.

5) Repeat step 3) and 4) for as a given number of turns around the ring.

In our simulations we have achieved the expected $\sqrt{P}$ speedup (defined as the ratio of serial to parallel execution time) for N up to 10^7 particles. In future work, we will incorporate the FD/FMM strategy into the WMPT code and do a detailed comparison with the serial version.

7. Summary and outlook

One of the main advantages of FD/FMM is its ease of coding. We were able to transform the serial FMM code to an FD/FMM code in a couple of days, with most of the work entailing the re-dimensioning of local arrays from $N \rightarrow N/\sqrt{P}$. One can also make use of MPI's Cartesian topology routines to create grid-row and grid-column communicators to facilitate scatter and gather operations in FD algorithm. In all, a total of four MPI communications calls need to be invoked while the original serial FMM code remains unaltered. The serial FMM code is called by each processor to accomplish the force calculation only for the i and j particles that it owns. This is embedded in the efficient parallel communication framework of FD to obtain the total beam-beam kick on each particle.

FD/FMM is limited by its isoscalability, $IS = \sqrt{P}$, which results from its communication cost scaling as $N/\sqrt{P}$. This is inherent in the FD algorithm and so the maximum speedup of $\sqrt{P}$ we achieved cannot be transcended with this method. As pointed out earlier, conventional parallel FMM methods are extremely inefficient in the beam-beam interaction context considered here since the particles move substantially over the physical domain between interactions. Thus FD/FMM offers an easy way to achieve some degree of speed up and ability to utilize a larger number of particles with a minimal amount of coding effort. This work highlights the need for the development of parallel FMM based algorithms for beam-beam interaction codes that can transcend the current spatial decomposition limitations. This will be a focus of our future work. In addition we will explore the application of FD/FMM to the so-called hybrid FMM approach, [2, 3]. This approach significantly reduces the number of points at which FMM has to compute the forces by laying down a PIC like grid and defining a minimum number, n, of particles per cell. Cells with more than n particles per cell are defined to be in the core and are deposited on the cell corners and the other cells are in the halo. The corners of the core cells and the halo particles are then fed to FMM.

8. Acknowledgments

References

[1] L. Greengard, *Thesis*, The Rapid Evaluation of Potential Fields in Particle Systems, The MIT Press, Cambridge, MA (1988);
L. Greengard, L. Rokhlin, *A Fast Algorithm For Particle Simulations*, Journal of Computational Physics **73**, 325-348 1987.
[2] M.Vogt, T.Sen, J.A.Ellison, *Simulations of Three 1-D Limits of the Strong-Strong Beam-Beam Interaction in Hadron Colliders Using Weighted Macro-Particle Tracking* Phys. Rev.

ST Accel. Beams **5**, 024401 (2002) and FNAL Pub-01/096-T (2001); M.Vogt, J.A.Ellison, T.Sen, R.L.Warnock, *Two Methods for Simulating the Strong-Strong Beam-Beam Interaction in Hadron Colliders*, Fermilab Beam-Beam Workshop (2001), Ed. T. Sen and M. Xiao, `http://www-ap.fnal.gov/~meiqin/beambeam01/beambeam01.html`.

[3] F. W. Jones, *Proc. Workshop on Space Charge Physics in High Intensity Hadron Rings*, Ed. A. U. Luccio and W. T. Weng, AIP Conf. Proc. **448**, (1998);

W. Herr, M.-P. Zorzano and F. W. Jones, Phys. Rev. ST-AB **4** 054402, (2001);

W. Herr, M.-P. Zorzano and F. W. Jones, Fermilab Beam-Beam Workshop (2001), Ed. T. Sen and M. Xiao, `http://www-ap.fnal.gov/~meiqin/beambeam01/beambeam01.html`.

[4] M.S. Warren and J.K. Salmon, *A Portable Parallel Particle Program* Computer Physics Communication **87**, 266-290 (1995); J.K. Salmon, M.S. Warren, and G.S. Winckelmans, *Fast Parallel Tree Codes for Gravitational and Fluid Dynamical N-Body Problems*, International Journal of Supercomputer Applications and High Performance Computing **8**, 129-142 (1994).

[5] S. Plimpton, *Fast Parallel Algorithms for Short-Range Molecular Dynamics*, J. Comp Phys. **117**, 1 (1995); B. Hendrickson and S. Plimpton *Parallel Many-Body Simulations Without All-to-All Communications*,Journal of Parallel and Distributed Computing, **27**, 15-25 (1995); S. Plimpton and B. Hendrickson *A New Parallel Method for Molecular Dynamics Simulation of Macromolecular Systems*, J. Comp. Chem **17**, 326 (1996). Alsing, P.M., Kale, R.P., and Fleharty, M.E., *Parallel Molecular Dynamics Visualization using MPI and MPE Graphics*, Proceedings IEEE, Second MPI Developers Conference, Library of Congress #96-76653, **104** (July 1-2, 1996).

Accuracy analysis of a 2D Poisson-Vlasov PIC solver and estimates of the collisional effects in space charge dynamics

A. Bazzani †, **G. Turchetti** ,
C. Benedetti, A. Franchi, S. Rambaldi
Dipartimento di Fisica Università di Bologna, INFN Sezione di Bologna, ITALY

Abstract.
We analyze the accuracy of a 2D Poisson-Vlasov PIC integrator, taking the KV as a reference solution for a FODO cell. The particles evolution is symplectic and the Poisson solver is based on FFT. The numerical error, evaluated by comparing the moments of the distribution and the electric field with the exact solution, shows a linear growth. This effect can be modeled by a white noise in the envelope equations for the KV beam. In order to investigate the collisional effects we have integrated the Hamilton's equations for N charged macro-particles with a hard-core r_H, reducing the computational complexity to $N^{3/2}$. In the constant focusing case we observed that a KV beam, matched or mismatched relaxes to the Maxwell-Boltzmann self consistent distribution on a time interval, which depends on r_H and has a finite limit for $r_H \to 0$. A fully 3D PIC code for short bunches was developed for the the ADS linac design at LNL (Italy). A 3D particle-core model, based on Langevin's equations with the drift and noise terms given by Landau's integrals, has been considered in view of the inclusion of collisional effects in the the Poisson-Vlasov equation.

1. Introduction

High intensity linacs and storage rings are being considered for fundamental research on matter at very high energy density and a variety of applications ranging from neutron spallation sources to nuclear wastes transmutation and inertial fusion. In high intensity accelerators the major problems are the field quality and the control of small losses. In the storage rings, where the number of visited FODO cells can be as high as 10^6, the resonances between the collective motion of the core and nonlinear betatron motion may cause slow diffusion and affect the dynamic aperture. Moreover the collisional effects need to be considered because the changes induced in the distribution may be relevant on a long time scale. The semi-analytical tools available are mainly based on the particle-core models developed to study the behavior of a test particle in a given self consistent field of the core. The application of the frequency map analysis to these models allowed to detect the key role of the mismatch oscillations in the diffusion process which may explain the growth of the halo [1, 2, 3]. The analytical self consistent solutions of the 2D and 3D Poisson-Vlasov (P.V.) equation for the particles distribution and the corresponding electric field are confined to the constant focusing case [4], in which a a linear and nonlinear stability analysis of small perturbations have also been developed [5, 13]. The excitation of the lowest unstable mode was proposed as a possible escape mechanism from the core, generating the halo [6, 7]. For the periodic focusing case

† To whom correspondence should be addressed: `bazzani@bo.infn.it`

numerical PIC solutions are extensively used since the only known analytic solution is KV distribution in the 2D case. We have developed a 2D PIC code for a lattice of identical FODO cells in order investigate its convergence properties, the onset of instabilities and the dynamics of test particles. This program has been undertaken using the KV as a reference solution in the periodic focusing case. Our 2D PIC solver is based on a symplectic integrator and a FFT Poisson solver which allows to impose Dirichelet conditions on a arbitrary closed boundary. For a matched KV solution the error on the electric field and on the moments of the distribution are analyzed as a function of the number K of Fourier components in each dimension and the number of pseudo-particles N. In this case the rms radii, the emittance and the field error exhibit a linear growth. For a fixed K we observe a decrease with N with a possible asymptotic limit; for K fixed the error decreases reaches a minimum and increases again. The linear growth of the moments is related to the accumulation numerical errors (due to the grid size and to the density fluctuations), which behave as a noise in the equations of motion[10]; For a circular machine to keep the error growth sufficiently small is a computational challenge¿ In this case the effect of Coulomb collisions cannot be neglected. To this end we have developed a molecular dynamics algorithm to integrate symplectically a system of N macro-particles, reducing the computational complexity from N^2 to $N^{3/2}$ (with a multipolar expansion of the far field). We have introduced a hard core r_H so that an elastic collision occurs whenever there is a contact. In the absence of hard core the close encounters are resolved by increasing the integration accuracy. It is found that an initial KV distribution evolves towards the self consistent Maxwell-Boltzmann distribution. This relaxation process cannot be obtained within the framework of a mean field equation like P.V. The relaxation time increases as r_H decreases and has a finite limit for $r_H \to 0$. An estimate of the relaxation time for the lattice of identical FODO cells previously considered is given. We have also developed a fully 3D parallel PIC code in order in order to study the evolution of a bunch in the linac ISCL, a project of the INFN laboratories of Legnaro (Italy).. In that case a Neuffer-KV initial distribution evolves numerically, with our PIC solver, towards a Fermi-Dirac distribution whose tails are Maxwellian. [11] We present a 3D particle core model defined by a Langevin equation with a drift and noise given by the Landau integrals in view of an inclusion of the collisions in the PIC code as recently proposed [12].

The plan of the paper is the following: we describe our 2D model in section 2, the PIC solver in section 3, we present the error analysis in section 4, the related Langevin model in section 5, we analyze the MD approach to collisions in section 6, 3D extensions and the related particle-core Langevin model in section 7.

2. The model

2.1. The Hamiltonian

We consider a coasting beam on a lattice of identical FODO cells of length L at the end of which a thin sextupole can be switched on. The Hamiltonian is

$$
\begin{aligned}
H_{ph} = \frac{p_{x\ ph}^2 + p_{y\ ph}^2}{2} + k_{0,x}^{ph}(s_{ph})\frac{x_{ph}^2}{2} - k_{0,y}^{ph}(s_{ph})\frac{y_{ph}^2}{2} + \\
+ \frac{K_{2,ph}}{6}(x_{ph}^3 - 3x_{ph}y_{ph}^2)\delta_L(s_{ph}) + \frac{\xi_{ph}}{2}V_{ph}(x_{ph}, y_{ph})
\end{aligned}
\tag{1}
$$

where $K_{2,ph}$ is the sextupolar gradient, δ_L denotes the periodic Dirac function of period L, ξ_{ph} is the perveance defined by

$$\xi_{ph} = \frac{2qQ}{p_0 v_0 \gamma^2} = \frac{2qI}{mc^3 \beta^3 \gamma^3} \tag{2}$$

and I is the current, Q the total charge per unit length. In order to use dimensionless variables we scale the longitudinal coordinates with L the transverse ones with a. As a consequence dropping the label ph on the scaled variables we have $x_{ph} = ax$, $s_{ph} = Ls$, $p_{x,ph} = p_x a/L$ and $L^2 k_{0,x}^{ph} = k_{0,x}$, $aLK_{2,ph} = K_2$ and $\xi_{ph} L^2/a^2 = \xi$. Finally letting $H_{ph} = a^2 L^{-2} H$ and $V_{ph}(x_{ph}, y_{ph}) = V(x, y)$ the scaled Hamiltonian reads

$$H = \frac{p_x^2 + p_y^2}{2} + k_{0,x}(s)\frac{x^2}{2} - k_{0,y}(s)\frac{y^2}{2} + \frac{K_2}{6}(x^3 - 3xy^2)\delta(s) + \frac{\xi}{2}V(x, y) \tag{3}$$

The self consistent potential satisfies the Poisson equation

$$\Delta V = -4\pi \rho_S(x, y, s) \qquad \rho_S(x, y, s) = \int \rho(x, y, p_x, p_y, s)dp_x dp_y \tag{4}$$

where ρ is the solution of the Liouville equation for the Hamiltonian H namely

$$\frac{\partial \rho}{\partial t} + [\rho, H] = 0 \tag{5}$$

We remark that the emittances scale as well according to $\epsilon_{ph} = \epsilon a^2/L$. Correspondingly the optical functions and the corresponding amplitudes scale according to $\beta_{x,ph} = L\beta_x$ and $A_{x,ph} = \sqrt{\epsilon_{x,ph}\beta_{x,ph}} = aA_x$. The advantage of this choice is that all the quantities become of order 1 choosing a of the order of the transverse size of the beam.

2.2. The FODO cell

It has a unit length, the quadrupoles length is 0.2 and their gradients are $k_{0,x} = k_{0,y} = 12$, the emittances are $\epsilon_x = \epsilon_y = 1$ (tipically $L = 1$ m, $a = 1$ mm). The perveance range we considered is $[1, 10]$, the results presented refer to $\xi = 10$ for which the tune depressions 0.1.

2.3. Envelope equation

In order to analyze the performance of the PIC solver for equations (4) and (5) we consider first a linear lattice ($K_2 = 0$) and K.V. beam whose phase space distribution is

$$\rho = \frac{1}{\pi^2 \epsilon_x \epsilon_y}\delta\left(1 - \frac{X^2 + P_x^2}{\epsilon_x} + \frac{Y^2 + P_y^2}{\epsilon_y}\right) \qquad \rho_S = \frac{1}{\pi A_x A_y}\vartheta\left(1 - \frac{x^2}{A_x^2} - \frac{y^2}{A_y^2}\right) \tag{6}$$

where the normalized coordinates are given by $X = x\beta_x^{-1/2}$, $P_x = (\alpha_x x + \beta_x p_x)\beta_x^{-1/2}$ and the optical functions are related by $\beta_x = A_x^2/\epsilon_x$, $\alpha_x = -A_x A_x'/\epsilon_x$ to the amplitudes $A_{x,y}$ which are the periodic solutions (matched beam) of the envelope equations

$$\frac{d^2 A_x}{ds^2} - \frac{\epsilon_x^2}{A_x^3} + k_{0,x}A_x = \frac{2\xi}{A_x + A_y} \qquad \frac{d^2 A_y}{ds^2} - \frac{\epsilon_y^2}{A_y^3} + k_{0,y}A_y = \frac{2\xi}{A_x + A_y} \tag{7}$$

3. The PIC solver

The numerical solution of Poisson-Vlavov equations was obtained with a PIC code by using N pseudo-particles. A parallel version for the evolution has been developed with MPI.

3.1. Initial distribution

We first change to the linearly normalized coordinate and perform the following transformation

$$X = \sqrt{tu\epsilon_x}\cos(2\pi\phi_x) \qquad Y = \sqrt{t(1-u)\epsilon_y}\cos(2\pi\phi_y) \qquad (8)$$

$$P_x = \sqrt{tu\epsilon_x}\sin(2\pi\phi_x) \qquad P_y = \sqrt{t(1-u)\epsilon_y}\sin(2\pi\phi_y) \qquad (9)$$

where $t \geq 0$ and $u, \phi_x, \phi_y \in [0,1]$. The following relation holds

$$\rho(X, P_x, Y, P_y)dX, dP_x dY dP_y = \rho(t, u, \phi_x, \phi_y)\pi^2\epsilon_x\epsilon_y tdt d\phi_x d\phi_y\, du \qquad (10)$$

For the case of a K.V. distribution we have $\rho = (\pi^2\epsilon_x\epsilon_y)^{-1}\delta(1-t)$ and the distribution is obtained by fixing $t = 1$ and filling uniformly the unit hypecube $[0,1]^3$ in the (u, ϕ_x, ϕ_y) space. The waterbag distribution for which $\rho = (\pi^2\epsilon_x\epsilon_y)^{-1}2\theta(1-t)$ is obtained by filling uniformly the unit hypercube $[0,1]^4$ in the (t^2, u, ϕ_x, ϕ_y) space.

3.2. The evolution scheme

We have considered a second order symplectic evolution scheme . The Hamiltonian (3) can be written as $H = H_0 + H_{sex} + \frac{1}{2}\xi V$ and the numerical scheme consists in replacing the space charge contribution in each interval $s_{i+1} - s_i$ with an impulsive force at s_i (kick approximation), whereas the linear evolution from s_i to s_{i+1} due to H_0 is computed exactly. The sextupole contribution , if any, is also a kick.

3.3. The Poisson solver

The Poisson equation is solved by a spectral method. We choose a rectangle $|x| \leq \frac{1}{2}\Lambda_x$, $|y| \leq \frac{1}{2}\Lambda_y$ whose sides are at least 3 times bigger than the amplitudes A_x, A_y. We consider a grid whose cells have sides $\Delta x = \Lambda_x/K_x$, $\Delta y = \Lambda_y/K_y$ where K_x, K_y are powers of 2 and at any grid point we compute the density either by the nearest grid points (NPG) deposition method or by the cloud in cell (CIC) deposition method. The potential is obtained by a FFT transform and the electric field is computed by a centered difference. To speed up the computation one FFT may be replaced by the solution of a three diagonal system which reduces the computational complexity from $(K \log K)^2$ to $K^2 \log K$. The potential is computed by imposing Dirichelet conditions $V = 0$ at the boundary of the square; these conditions can be imposed on an arbitrary closed curve, by using a Green's function method which doubles the computation time at each step once a $K \times K$ matrix is inverted, for an approximation of the given curve with a polygonal with K vertices.

4. Error analysis

A rigorous analysis of the discretization error introduced by a PIC code with respect to the exact solution is a very hard task. We remark that the approximate solutions depend on the box size Λ, (assuming $\Lambda_x = \Lambda_y = \Lambda$), on the the number K of Fourier components (assuming $K_x = K_y$) which fixes the space resolution $\Delta x = \Lambda/K$, on the particle number N and on the integration step Δs. The exact result is recovered in the limit $\Lambda \to \infty$, $\Delta s \to 0$ and $K, N \to \infty$, once this limit is correctly defined. We have fixed the first two parameters by choosing $\Delta s = 1/100$ (this is the average of Δs_i) and $\Lambda = 20$ so that for the chosen value of the perveance $\xi = 10$ it is larger than 3
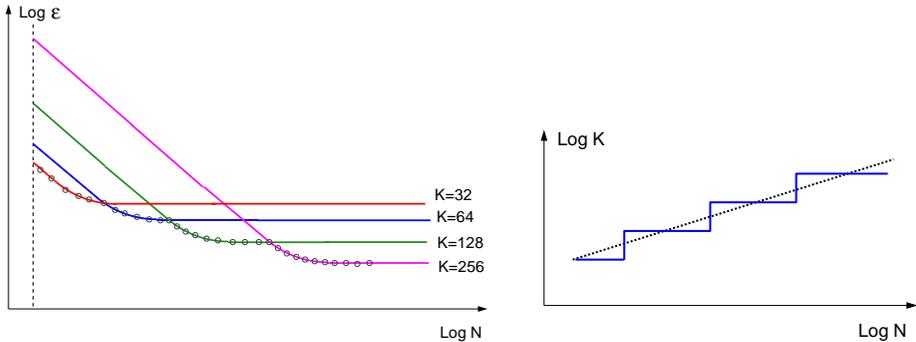
Figure 1. Sketch of the dependence of the error on K and N (left). Sketch of the behavior of K_{opt} as a function of N and corresponding power law $K_{opt} = C\,N^{\beta}$ continuous interpolation.

times the amplitudes. The error introduced by the longitudinal step is of order $(\Delta s)^2$ and we checked this is the case on the solution of envelope equation we computed by integrating (7) with the same procedure (splitting method with exact integration of the linear part and kicks for the space charge force). The error dependence on K and N has been examined. We investigate the double limit by letting first $N \to \infty$ and considering the its dependence on K. Indeed N can be increased up to 10^6 and occasionally 10^7, whereas K is limited to a few values ranging from 32 to 512. We have checked that by increasing N the error decreases up to a limit value, which can be further reduced by increasing K. As consequence if we fix N and vary K, the error first decreases reaches a minimum and then increases. The error we consider refers to the electric field and to the moments of the distribution. Since the error growth with s is linear, see figures 2 and 3, we have computed the slope and studied its dependence on N and K. For fixed K the slope appears to follow a power law $N^{-\alpha}$ with an exponent α slightly smaller than 1 before the asymptotic region is reached; the asymptotic limit decreases with K. Fixing N at a low value the error increases with K. As a consequence for fixed N the error decreases, reaches a minimum and then increases again, as shown qualitatively by figure 1. The optimal value of K is a stepwise function of N for which we propose a power law interpolation $K = c\,N^{\beta}$.

4.1. Scaling law

A power law is found if we assume that the optimal condition is reached when the relative error on the density, given by the statistical fluctuation, is equal to uncertainty in the space resolution. Denoting by n_P the average number of particles per cell in the core and by ℓ the core diameter such condition is $\Delta n_P / n_P \simeq \Delta x / \ell$. Letting η the fraction of the box occupied by the core, the diameter of the core is given by $\ell \simeq \eta^{1/2}\,\Lambda$. The side of the cell is $\Delta x = \Lambda / K$ so that the uncertainty in the position within the core is $\Delta x / \ell = 1/(K\,\eta^{1/2})$ For a uniform distribution the number of occupied cells is $\eta\,K^2$ and the average number of particles per cell is $n_P = N/(\eta\,K^2)$. As a consequence since the statistical fluctuation is $\Delta n_P = \sqrt{n_P}$ by equating the position uncertainty to the cell population uncertainty $n_P^{-1/2} = \Delta x / \ell$ we obtain

$$N \simeq \eta^2\,K^4 \tag{11}$$
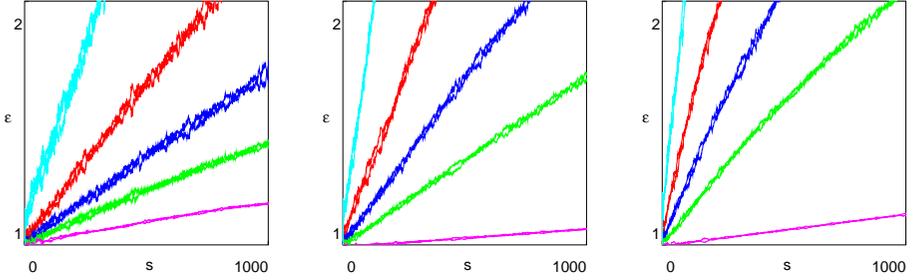
Figure 2. Variation of the emittances $\bar{\epsilon}_x$, $\bar{\epsilon}_y$ with the iteration number n for different values of the number of Fourier components $K = 64$ (left frame), $K = 128$ (central frame), $K = 256$ (right frame). In each frame the x and y emittance have the same color which changes with the particles number: $N = 10^3$ (light blue), $N = 2.5\,10^3$ (red), $N = 5\,10^3$ (blue), $N = 10^4$ (green), $N = 10^5$ (purple).

4.2. Numerical experiments

We have analyzed the error dependence on K and N on a model of FODO cell with a high value of the perveance. This choice was made since we observed that the behavior of the error was clearly detectable for a moderate number of iterations $s_{max} = 10^3$. Indeed the most relevant effect of discretization is a numerical noise which causes a linear growth of the moments of the distributions and of the error in the electric field. The growth rate increases with the perveance. We consider the FODO cell of section 2 with a perveance $\xi = 10$. The bare and depressed tune are $\nu_{0\,x} = \nu_{0\,y} = 61.8^o$ and $\nu_x = \nu_y = 6.5^o$, the bare amplitudes $A_{0\,x} = 1.21$, $A_{0\,y} = 0.81$, the core amplitudes $A_x = 3.65$, $A_y = 2.53$. The number of longitudinal steps per FODO cell used is 100, the periodic solution is found by a Newton's method and the accuracy of the symplectic integration procedure agrees with the theoretical estimate $(\Delta s)^2 = 10^{-4}$. The size of the square where Poisson equation is solved is $\Lambda = 20$ so that fraction occupied by the core is $\eta = 0.07$. The relative error on the electric field is defined by

$$\frac{\Delta E}{E} = \frac{\left(\int_{D_*} [(E_x - E_{x\ ex})^2 + (E_y - E_{y\ ex})^2]\, dx\, dy\right)^{1/2}}{\left(\int_{D_*} [E_{x\ ex}^2 + E_{y\ ex}^2]\, dx\, dy\right)^{1/2}} \tag{12}$$

where the integrals are extended to the $D_* = [-2.5, 2.5]^2$ a domain almost entirely within the core. In this way we can ignore errors due to the non-zero field on the boundaries.

In figure 2 we show the variation of the emittances defined by $\bar{\epsilon}_x = 4\,(<x^2><p_x^2> - <xp_x>^2)^{1/2}$, with the number s of visited FODO cells. It is evident that the growth is linear with s and that the slope decreases with N. When K increases for a fixed N, the slope also increases, except for for the highest value $N = 10^5$, shown in figures 2, for which the minimum occurs at $K = 128$. In figure 3 we consider the variation of the error in the electric field $\Delta E/E$ with the iteration number s. We observe that the error growth with s is still linear and that the slope decreases with N for a fixed K. We have also fitted a line $a + bn$ to the error growth in order to examine the dependence of the coefficient b on N and K. The scaling law suggests that increasing K by 2 the optimum N should increase by 2^4, a result compatible with our numerical observations. For $K = 64$ the optimum N is below 10^5 a value above which the error does not decrease
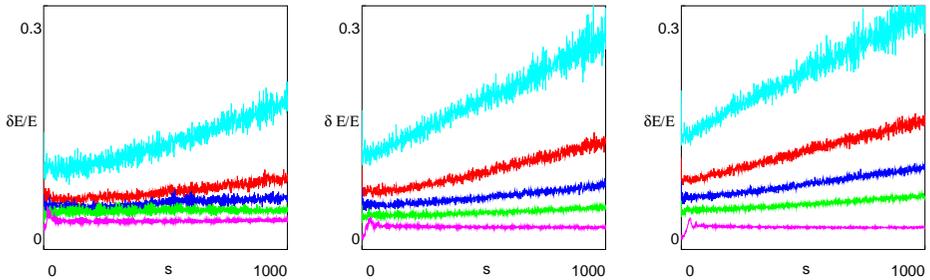
Figure 3. Variation of error on the electric field $\delta E/E$ with the iteration number s for different values of the number of Fourier components $K = 64$ (left frame), $K = 128$ (central frame), $K = 256$ (right frame). In each frame the color which changes with the particles number: $N = 10^3$ (light blue), $N = 2.5\,10^3$ (red), $N = 5\,10^3$ (blue), $N = 10^4$ (green), $N = 10^5$ (purple).
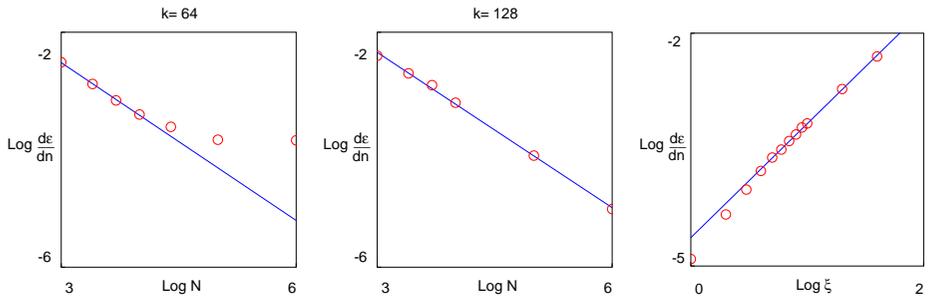
Figure 4. Variation of the slope b of the emittance $\bar{\epsilon}_x = a + bs$, determined by least square fit, with respect to N, for different values of the number of Fourier components: $K = 64$ (left frame), $K = 128$ (central frame) . Variation of the slope with the perveance ξ for $N = 10^4$ and $k = 128$ (right frame).

appreciably. This is shown by figure 4 where dependence of the emittance slope on N is shown for a fixed value of K. The saturation for $K = 64$ is evident, whereas it is not reached for $K = 128$ up to $N = 10^6$. In the same figure we show the variation of the slope with the perveance which is monotonic and approximately given by $b \propto \xi^{3/2}$. The results described above refer to the NPG deposition method. We have repeated the same simulations with CIC deposition method and the difference was typically within 10%, without any significant variation for the dependence on K and N described above.

5. The Langevin equation

We consider a stochastic model suitable to describe the linear rise of the errors with number of visited cells s. We assume that the fluctuations in the particle number in every cell is equivalent to a white noise in the equations of motion. This is by no means related to the physical collisions which preserve asymptotically the kinetic energy. Referring

still to a KV beam the equations of motion become

$$\frac{d^2x}{ds^2} + k_x\,x = \eta\frac{dw}{ds} \qquad\qquad \frac{d^2y}{ds^2} + k_y\,x = \eta\frac{dw}{ds} \qquad (13)$$

where $w(s)$ denotes a Wiener process and the the depressed gradients k_x, k_y are defined by $k_x(s) = k_{0\,x} - 2\xi\,(\,A_x(A_x+A_y))^{-1}$ and similarly for k_y. The amplitudes A_x and A_y are solutions of the envelope equation

$$\frac{d^2 A_x}{ds^2} - \frac{\epsilon_x^2}{A_x^3} + k_x(s)A_x = 0 \qquad\qquad \frac{d^2 A_y}{ds^2} - \frac{\epsilon_y^2}{A_y^3} + k_y(s)A_y = 0 \qquad (14)$$

We recall that when $\eta = 0$ the following relations hold

$$<x^2> = \frac{A_x^2}{4} \qquad <p_x^2> = \frac{1}{4}\frac{\epsilon_x^2}{A_x^2} + \frac{1}{4}\Big(\frac{dA_x}{ds}\Big)^2 \qquad <x\,p_x> = \frac{1}{4}\,A_x\frac{dA_x}{ds} \qquad (15)$$

By taking the average also on the stochastic process we find that the second moments grow linearly with s with a slope proportional to η^2.

This can be easily proven by direct computation in the smooth focusing approximation and zero space charge limit. In the symmetric case $k_{0,x}(s) = k_{0,y}(s)$ are replaced by a constant ω_0^2 and the fluctuations change the x and y emittance ϵ_0 into $\epsilon(s) = \epsilon_0 + 2\eta^2\,s/\omega_0$ if one neglects oscillating terms. As a consequence since $A^2 = \epsilon(s)/\omega_0$ we see that $d\epsilon^2/ds = 4\epsilon\,\eta^2/\omega_0 = 4\eta^2\,A^2$. More generally the amplitudes and the emittances satisfy the system of equations (14), where k_x, k_y are replaced by ω_0^2 and

$$\frac{d\epsilon_x^2}{ds} = 4\,\eta^2 A_x^2 \qquad\qquad \frac{d\epsilon_y^2}{ds} = 4\,\eta^2 A_y^2 \qquad (16)$$

Assuming these equations still hold in presence of space charge, one one has to replace $k_x(s)$, $k_y(s)$ with $\omega_2(s) = \omega_0^2 - \xi/A^2(s)$. In order to find an approximate solution for a matched beam we observe that the numerical solution shows that $A(s)$ varies slowly so that in (16) one can replace $A^2(s)$ with $A^2(0) = \epsilon_0/\omega(0)$ namely

$$\epsilon(s) = \Big(\epsilon_0^2 + 4\epsilon_0\frac{\eta^2\,s}{\omega(0)}\Big)^{1/2} \simeq \epsilon_0 + 2\frac{\eta^2\,s}{\omega(0)} \qquad (17)$$

In (17) the linear approximation holds for $4\eta^2 s/\epsilon_0\omega(0) \ll 1$ and $\omega(0) = \epsilon_0/A^2(0)$ where

$$A^2(s) = \frac{\xi}{2\omega_0^2} + \frac{\epsilon(s)}{\omega_0}\Big(1 + \frac{\xi^2}{4\epsilon^2(s)\omega_0^2}\Big)^{1/2} \qquad (18)$$

The errors on the emittance and the amplitude are given by (17) and (18) whereas for the electric field we have $\mathbf{E} = \mathbf{x}\,\xi/A^2(s)$ and the error is computed according to (12). The result for $\xi \gg 1$ simplifies and setting $\gamma = 2\eta^2/\omega(0)$ we have

$$\epsilon(s) = \epsilon(0) + \gamma s \qquad A^2(s) = A(0)^2 + \frac{\gamma}{\xi\,A(0)}s \qquad \frac{\delta E}{E} = \Big|1 - \frac{A^2(s)}{A^2(0)}\Big| = \frac{2\gamma}{\xi A^2(0)}s \qquad (19)$$

In our model using the smooth focusing approximations we obtain $A = 2.95$, to be compared with the average of the amplitude over one period $<A_x> = <A_y> = 3.08$. The estimate of the slope for the electric field error growth is 0.022 times the slope of the emittance growth; fitting that data of our simulations we find a range of values between 0.018 and 0.03 which is compatible with the estimate.
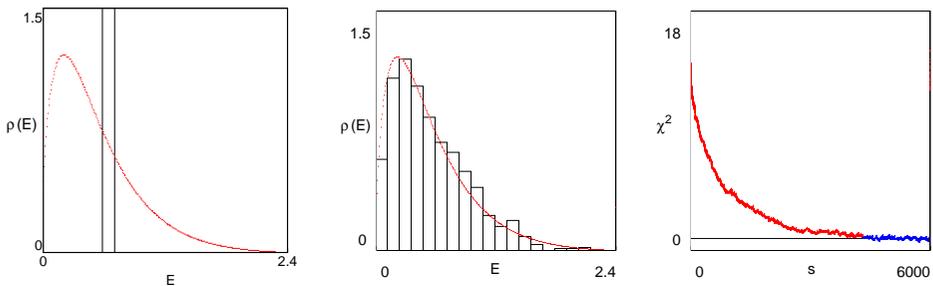
Figure 5. Energy density for particles without hard core at $s = 0$ (left) and $s = 6000$ (center). The red curve is the self consistent Maxwell-Boltzmann (MB) distribution. Right frame: The χ^2 for the discrepancy of the phase distribution ρ with respect to the MB distribution.

6. Coulomb oscillators

In order to estimate the effect of collisions we have developed a 2D molecular dynamics (MD) code. In this preliminary stage the smooth focusing approximation was chosen because the thermodynamic equilibrium distribution is known. The equations of motion for a system of N Coulomb oscillators are

$$\frac{d^2 x_i}{ds^2} + \omega_0^2 x_i = \frac{\xi}{N} \sum_{j \neq i} \frac{x_i - x_j}{r_{ij}^2} \qquad \frac{d^2 y_i}{ds^2} + \omega_0^2 y_i = \frac{\xi}{N} \sum_{j \neq i} \frac{y_i - y_j}{r_{ij}^2} \tag{20}$$

and we integrate them with a second order symplectic integrator. A more accurate integration is reserved to the subset of colliding particles (whose distances r_{ij} are below an assigned value λ_{coll} smaller than the average distance $N^{-1/2}$ which in turn is smaller than the Debye radius, which for $\xi \sim 1$ is $1/3$ the beam size.

$$R_D^2 = \frac{k_B T}{4\pi q^2 \rho_s} = \frac{A^2}{2\xi} \left\langle \frac{p_x^2 + p_y^2}{2} \right\rangle = \left(\frac{A}{4}\right)^2 \frac{2\omega\,\epsilon_0}{\xi} \tag{21}$$

We have also considered a model in which the macro-particles have a hard core r_H, so that elastic collisions occur when $r_{ij} = 2r_H$. These collisions are treated with compiling a list for each step and linearly interpolating the elastically colliding particles. The simulations refer to a unit cells with bare phase advance $\omega_0 = 1$ emittances $\epsilon_0 = 2$ and perveance $\xi = 2$ so that $\omega = 0.64$. The second order moments, for a matched initial K.V. distribution, remains constant within the statistical error. The distribution changes and evolves to a Maxwell-Boltzmann self consistent distribution $\rho = Z^{-1} e^{-H/kT}$ where $H = \frac{1}{2}(p^2 + \omega_0^2 r^2) + \frac{\xi}{2} V(r)$ and V is the self consistent potential. The relaxation time depends on the hard core radius and reaches a finite limit for $r_H \to 0$. In figure 5a and 5b we show the initial and final distribution in energy, when the equilibrium condition has been reached. The result is the same for any value of the hard core. In figure 5c we show instead the evolution of the distance from the equilibrium distribution, measured by the χ^2 (averaged on a small time interval to reduce the fluctuation). The hard core accelerates the relaxation process as expected and the dependence of the relaxation time on r_H is smooth.

7. 3D model and its applications

7.1. Scaling law for the PIC solver

An analogous error analysis of our 3D PIC solver is a more difficult task. In the periodic focusing a self-consistent analytic solution of the Vlasov problem to be used as reference solution is not available. The coupling between longitudinal and transverse dynamics may also produce a fast relaxation of the bunch into a stationary distribution. Looking at the emittance behavior it is therefore difficult to distinguish the physical and numerical effects. A generalization of the scaling law given in sec. 4.1 reads

$$N \simeq \eta^{5/3} \ K^5 \tag{22}$$

where η is the fraction of the cube occupied by the core. With the parameters of the TRASCO-ISCL linac [8], a first set of simulations has been recently carried out [9].

7.2. Collisional Particle in Core model

The collisional effects within the Debye radius are modeled by using the Landau approach: the collisions are supposed to be 'soft', binary and of short range. We introduce the drift and the diffusion coefficients

$$a_i(\mathbf{x}, s) = \eta^2 \frac{\partial \overline{\rho}_1}{\partial p_i} \qquad D_{i\,k}(\mathbf{x}, s) = \frac{\eta^2}{2} \frac{\partial^2 \overline{\rho}_{-1}}{\partial p_i \partial p_k} \tag{23}$$

where $\overline{\rho}_n(\mathbf{x}, s)$ are the Rosenbluth potentials obtained by integrating the phase space density over a Debye sphere of center $\mathbf{x}$, whose volume is V_D

$$\overline{\rho}_n(\mathbf{p}, \mathbf{x}, s) = \int \frac{\rho_D(\mathbf{p}', \mathbf{x})}{\|\mathbf{p} - \mathbf{p}'\|^n} d\mathbf{p}' \qquad \rho_D(\mathbf{p}', \mathbf{x}, s) = \frac{1}{V_D} \int_{\|\mathbf{x}-\mathbf{x}'\| < R_D} \rho(\mathbf{x}', \mathbf{p}, s) \ d\mathbf{x}' \tag{24}$$

where $j, k = 1, 2, 3$ denotes the coordinates in the bunch reference frame, ρ is the normalize (to 1) phase space beam distribution and η is related to the perveance ξ by

$$\eta^2 = 2\pi \frac{\xi^2}{N} \log \Lambda \qquad \xi = \frac{2 \, e^2 \, N}{m v_0^2} \tag{25}$$

$\Lambda = \log(2/\theta_{\min})$ being here the Coulomb logarithm, with $\theta_{\min}$ the minimum scattering angle within the Debye sphere. The integrals defined above should be computed numerically. However assuming an ellipsoidal uniform distribution (not self consistent), they can been explicitly evaluated in terms of elliptic functions. The single particle equations are Langevin equations

$$\frac{d^2 x_i}{ds^2} + \omega_{0\,i}^2 \, x_i = \xi E_i \, + \, a_i \, + \, \sqrt{D_{i,k}} \, \frac{d \, w_k}{ds} \tag{26}$$

where w_i is a Wiener process, $E_i^{s.c.}$ the mean space charge field (see ref. [3]), produced by a unit charge. Using the average parameters of the TRASCO-ISCL linac and the constant focusing approximation, we have performed two simulations (matched and mismatched case). $5 \cdot 10^4$ pseudo-particles are generated following the K.V. distribution and tracked for 500 focusing cells. The emittance growth observed in both case is of about 2%. The halo formation is limited in the matched case, while reaches the 20% in the mismatched case, as shown in fig 6a. In fig. 6b is shown the final energy spread around the mean value defined as

$$< E >= \left\langle \sum_{i=1}^{3} \frac{A_i^2 p_i^2}{\epsilon_i^2} + \frac{x_i^2}{A_i^2} \right\rangle \tag{27}$$
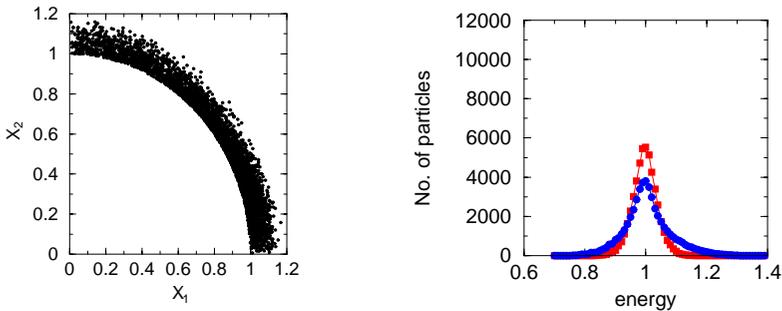
Figure 6. Formation of a halo in due to the collision terms (left). Spread of the energy after a 500 FODO cells (right) for the matched (red) and mismatched (blue) case.

where a_i are the bunch amplitudes.

We recall that the Particle in Core model is able to reproduce the single particle instabilities, but unable to observe the collective instabilities since its space charge field is not self consistent. On the contrary the PIC model is self consistent but is not optimized for the single particle instabilities, since the single particle contributes are averaged on the grid points. All these problems could be avoided by using a fully 3D Molecular Dynamics model, whose computational complexity is too high to be implemented with enough resolution.

8. References

[1] M. Lagniel Necl. Instr. Meth. A **345** (1994) 405.
[2] A. Bazzani, M. Communian, A. Pisent Part. Accel. **63** (1999) 79.
[3] M. Communian, A. Pisent, A. Bazzani, S. Rambaldi, G. Turchetti Phys. Rev. Special Topics - Accelerators and Beams Vol. 4, 124201 (2001)
[4] R. L. Gluckstern, A. V. Fedotov, S. Kurennoy, R, Ryne Phys. Rev. **58**, 4977 (1998)
[5] I. Hofmann, Phys. Rev. E **57** (1998) 4713.
[6] R. L. Gluckstern Phys. Rev. Letters **73**, 1247 (1994)
[7] R. L. Gluckstern, W. H. Cheng, S. S. Kurennoy, H. Ye Phys. Rev E **54**, 6788 (1996)
[8] http://trasco.lnl.infn.it
[9] A. Franchi et al. , LINAC conference, Korea 2002
[10] J. Struckmeier, Phys. Rev. **E 54**, 830 (1996)
[11] G. Turchetti, S. Rambaldi, A. Bazzani, M. Communian, A. Pisent *3D solutions of the Poisson-Vlasov equations for a charged plasma and particle core model in a line of FODO cells* - EPJ submitted (2002)
[12] Ji Qiang, R. D. Ryne, S. Habib *Self-consistent Langevin simulation of Coulomb collisions in charged particle beams* technical paper, SC2000 conference, Dallas (2000) http://www.SC2000.org/proceedings/techpapr/papers/pap224.pdf
[13] R. Davidson Phys. Rev. Letters **81**, 991 (1998)

The COSY language independent architecture: porting COSY source files

L. M. Chapin†, J. Hoefkens†§ and M. Berz†

† Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA

Abstract. As part of the project of porting the code COSY INFINITY[1] to a language independent architecture, a tool is being developed that translates COSY language to C++ language. The tool will allow the use of C++ classes for all elementary operations of COSY, which represent a wrapper around high-performance COSY C source. This will allow those unfamiliar with the COSY language access to the power of COSY.

1. The COSY language independent architecture and the COSY syntax

Recently the COSY system [1, 2, 3, 4, 5, 6] has been ported to a language independent platform to accommodate the use in various modern and widely used programming environments. In particular, the languages supported include FORTRAN 77, C, as well as the object oriented environments of FORTRAN 90 and C++. This is achieved in the following way. The original FORTRAN 77 based language environment consists mainly of operations on a suite of COSY objects, most notably the DA (Differential Algebra) objects that have proven very useful for beam physics purposes [7]. These objects can traditionally be utilized either within the environment of a FORTRAN precompiler[8, 9, 10], or within the object oriented language environment of the COSY scripting tool[1], which will be discussed in detail below. Support in the C world is achieved by the use of a FORTRAN-to-C precompiler, and the FORTRAN source has been adjusted to achieve reliable cross-compilation to the C environment that results in efficient code. To assure future reliability and reproducibility of this porting operation, the C source code of the cross-complier is maintained together with the FORTRAN package. The use within C++ and FORTRAN90 are achieved by a suite of light wrappers providing C++ and FORTRAN90 objects for the COSY objects [11]. Great care is taken to maintain the high performance of these data types, which is achieved by merely representing the objects via pointers into COSY's original memory management system. Altogether this approach allows the rendering of COSY objects in four different language environments, while developers need only to write and maintain the FORTRAN 77 based source code in COSY.

Besides the use of data types of COSY in other languages, it is also desirable to port high-level code written in the COSY scripting language to other languages. The COSY language is an object oriented scripting tool that supports dynamic typing. It is recursive and allows nesting, local variables, and procedures. Its syntax concepts are similar to those of PASCAL, but in addition COSY has object oriented features. All type checking is done at run time, not at compile time.

Most commands of the COSY language consist of a keyword, followed by expressions and names of variables, and terminated by a semicolon. The individual entries and the

§ Currently at GeneData AG, CH-Basel 4016, Switzerland.

semicolon are separated by blanks. The exceptions are the assignment identifier ":=", and the call to a procedure, in which case the procedure name is used instead of the keyword. Line breaks are not significant; commands can extend over several lines, and several commands can be in one line. Everything contained within a pair of curly brackets "{" and "}" is ignored. Each keyword and each name consist of up to 32 characters. The case of the letters is not significant. The basic COSY commands consist of the following commands:

```
BEGIN        END
PROCEDURE    ENDPROCEDURE
FUNCTION     ENDFUNCTION
VARIABLE
LOOP         ENDLOOP
WHILE        ENDWHILE
IF    ELSEIF  ENDIF
```

as well as the assignment command. (For more details, see p. 18 ff in [1], and p. 27 for a complete list.) In the following we describe a tool that converts code written in COSY syntax to C++.

2. Outline of the method

The conversion between COSY syntax and C++ syntax is accomplished by means of a program written in Perl [12]. The program begins by reading in all of the lines of the source .fox COSY file and placing each one, delimited by a semicolon, into an array called @Lines, with a separate entry for each line. The array is then processed and various changes are made to each line, and the order is changed in accordance with C++ standard syntax. As an example of this first type of change, a COSY loop construct "LOOP I 1 TWOND 2 ;" would be changed to the C++ equivalent "for (I = 1 ; I <= TWOND ; I+=2)".

The second type of change includes the unnesting of COSY functions and returning of corresponding C++ functions in the appropriate order. The modified contents of that array are then outputted to a new C++ file, which can then be read through and even modified. The resulting C++ source will be fully compilable and, at execution time, will exactly duplicate the performance of the original COSY source.

3. COSY intrinsic functions and procedures

All of the COSY intrinsic functions and procedures can be accessed through the COSY class. The converter program gives calls in the proper format; in COSY, calls may be made with no parentheses or commas between the arguments, but they are necessary in C++. A few of the names are slightly different (normally with different capitalization) to avoid conflict with C++ reserved words, such as "int." The renaming is based on a translation table (pp. 44-48 in [1]). For example: "SIN(X)" becomes "Cosy sin (x)" and "PROCEDURE MEMALL Y" becomes "void memall (y)".

4. COSY commands

The Programming Manual lists the keywords in COSY (p. 27 in [1]). The simple commands are reformatted with a one line statement when they are encountered. The commands that are

complicated enough to need more than one line of manipulation have their own subprograms that are called when a line is found to contain a command. The line is sent to the subprogram as an argument. Each subprogram breaks the line up and takes pieces as it needs them, processes the line, and returns it to its place. For example, upon encountering a line containing the word 'WRITE' such as "WRITE 6 'Hello' ;", the program would call the following Perl subprogram.

```perl
sub write
{
   my($write) = $_[0] ;
   #creates a local variable for the argument sent, i.e. The line

   $write =~ s/\b&\b/+/g ;
   #changes the Cosy concatenation symbol for the C++ symbol

   $write =~ s/(?i)write 6/cout << /i ;
   #changes the Cosy output command for the C++ outstream

   $write =~ s/'/"/g ;
   #changes Cosy's single quotes around strings
   #to double quotes for C++

   $write = $write." ; \n" ; #returns the modified line
}
```

Here $_[0] is the default variable for the first argument sent to the subprogram in Perl (the arguments are stored as entries to the array @_ and are accessed by $_[0], $_[1], $_[2], etc. for each argument).

The subprograms for the other commands are generally much more complicated, but follow the same basic idea.

5. User-declared procedures and functions

In COSY, following conventional PASCAL philosophy, it is permitted to nest functions and procedures within each other infinitely deep and to even entangle the functions and procedures. For example, it's fine to declare:

```
PROCEDURE PROTEST1 X Y ;
   PROCEDURE PROTEST2 Z ;
      FUNCTION FUNTEST1 A ; FUNTEST1 := A +1 ; ENDFUNCTION ;
      Z := FUNTEST1(Z) ; ENDPROCEDURE ;
   PROTEST2 Y ; X := X+Y ; WRITE 6 X ;
   ENDPROCEDURE ;
```

In C++, functions cannot be nested, and the function signatures should be declared at the beginning of the converted program and the functions should be defined at the end. Thus, a user COSY code must be reshaped before being converted to C++. "PROCEDURE"s may be regarded as void functions for the purposes of C++. The subprograms of the conversion program that reshape the "PROCEDURE"s are very similar to those that reshape "FUNCTION"s, with the exception of being of type void and thus not returning anything. They are treated together for reshaping purposes.

(i) The conversion program reads through the array until it first finds the keyword "ENDFUNCTION" or "ENDPROCEDURE," assigning the variable 'toggle' to "FUNCTION" or "PROCEDURE" depending on which is found first.

(ii) The index of the end 'toggle' line is recorded and the search reverses and looks for the closest preceding 'toggle,' recording its index.

(iii) All of the array entries between the 'toggle' index and the end 'toggle' index are inclusively removed from the array and concatenated into a single string (with the entries separated by semicolons ";").

(iv) The string is passed to the conversion 'toggle' subprogram, which takes care of formatting details. The nested structure is preserved by appending the names of any functions or procedures the function is nested within to the end of the name.

(v) The formatted function is then stored in a separate array and the process is repeated, assigning 'toggle' the value of "FUNCTION" or "PROCEDURE" each time depending on the next one found, until all of the "END" 'toggle' s are removed from the @Lines array.

(vi) Each function is then split back into separate entries by semicolons and added to the end of the @Lines array.

As a result of this process, the reshaped intermediate code of the example above would look like this:

```
FUNCTION FUNTEST1_PROTEST2_PROTEST1 A ;
           FUNTEST1 := A +1 ;
ENDFUNCTION ;

PROCEDURE PROTEST2_PROTEST1 Z ;
           Z := FUNTEST1(Z) ;
ENDPROCEDURE ;

PROCEDURE PROTEST1 X Y ;
           PROTEST2 Y ;
           X := X+Y ;
           WRITE 6 X ;
ENDPROCEDURE ;
```

The subprograms of the converting program that format the "FUNCTION"'s and "PROCEDURE"'s differ only slightly. The following describes the formatting "FUNCTION" subprogram in detail.

(i) The conversion subprogram for functions takes all of the lines that were between the command "FUNCTION" and the command "ENDFUNCTION" concatenated into one line as an argument.

(ii) The line is immediately divided up into local variables for function name, arguments, and the rest of the lines.

(iii) An opening curly bracket is appended to the arguments. If there is more than one argument, commas are placed between them. The set of arguments is placed in parentheses and each argument is typed as 'Cosy&'. This is a reference to the actual variable. An 'f' for function or a 'p' for procedure is placed

at the beginning of the name to establish uniqueness. Thus, for example, " FUNCTION FUNTEST1_PROTEST2_PROTEST1 A " in COSY has become "Cosy fFUNTEST1_PROTEST2_PROTEST1 (Cosy& A)".

(iv) A COSY variable with the same name as the function is declared, preceded by an "i", because, in COSY, functions deal with and return such a variable, which is declared only in naming the function. "Cosy iFUNTEST1_PROTEST2_PROTEST1 (1)," is the return variable for the above example.

(v) The rest of the lines are returned to the function and the final curly bracket closes it.

(vi) The names of any variables declared in the function (including the name variable and arguments) are placed in a string that becomes the "namespace" of the function. The "namespaces" that are visible to the function are declared using "using namespace" [13]. See the section on variables below.

(vii) The entire array @Lines holding the rest of the program is then queried for calls to the function, which are reformatted to call " $functionName" with parentheses surrounding and commas separating the arguments and the full function name.

(viii) All keywords are searched for and dealt with by their individual subprograms.

(ix) A function signature is also generated to be placed at the beginning of the output program so that the function may be placed at the end of the main output program in the style of C++. The function is then returned to the @Lines array.

The resulting C++ code for this example would be:

```
Cosy fFUNTEST1_PROTEST2_PROTEST1 (Cosy& A) ;
void pPROTEST2_PROTEST1 (Cosy& Z) ;
void pPROTEST1 (Cosy& X, Cosy& Y) ;

namespace PROTEST1
{
        Cosy X (1) ;
        Cosy Y (1) ;
}

namespace PROTEST2_PROTEST1
{
        Cosy Z (1) ;
}

namespace FUNTEST1_PROTEST2_PROTEST1
{
        Cosy iFUNTEST1_PROTEST2_PROTEST1 (1) ;
        Cosy A (1) ;
}

Cosy fFUNTEST1_PROTEST2_PROTEST1 (Cosy& A)
{
        using namespace PROTEST1 ;
        using namespace PROTEST2_PROTEST1 ;
        using namespace FUNTEST1_PROTEST2_PROTEST1 ;
        iFUNTEST1_PROTEST2_PROTEST1 = A +1 ;
```

```
        return iFUNTEST1_PROTEST2_PROTEST1 ;
}

void pPROTEST2_PROTEST1 (Cosy& Z)
{
        using namespace PROTEST1 ;
        using namespace PROTEST2_PROTEST1 ;
        Z = fFUNTEST1_PROTEST2_PROTEST1 (Z) ;
}

void pPROTEST1 (Cosy& X, Cosy& Y)
{
        using namespace PROTEST1 ;
        pPROTEST2_PROTEST1 (Y) ;
        X = X+Y ;
        cout << X ;
}
```

6. Variables

COSY variables have no declared types. They can be declared with an unlimited number of arguments that will simply increase the dimensionality of the variable (while defining the dimensions).

In treating the variables of COSY programs, most of the variables could be given the type "Cosy" from the COSY class. However, many functions explicitly require integers, and the C++ version of a COSY array is based on an integer array. For these functions, it must be determined if a numerical integer or a COSY variable is passed (both equally valid options in COSY, but not acceptable in C++). In the case of a COSY variable, the variable must be cast into an integer. This is done by using the COSY class's "toDouble" Cosy to double casting function [1] and casting the resulting double as an integer in the normal C++ fashion.

In consideration of scope, COSY variables are visible inside the program segment in which they are declared as well as in all other program segments inside it. If a variable has the same name as one visible from a higher level, its name and properties override the name and properties of the higher level variable of the same name for the remainder of the block [1]. This priority is being dealt with in C++ by creating a "namespace"[13] for each function and procedure and giving access to the "namespace" in order of priority to all of the appropriate program segments. The C++ command "using namespace"[13] overrides variables in the same manner as COSY priority. The rule for granting access is that: if a "namespace"'s full name (with the nesting extensions) appears in the name of the block, the block has the right to that "namespace". This "namespace" convention is still being implemented, but appears to be a promising solution that allows for variables to maintain their short, original names while preserving scope and avoiding a mass of global variables. In COSY, the actual variable is passed to a function and not just a copy, as is common in C++. To account for this, all variables are passed by reference to functions in the converted code [13].

7. Array access

To access arrays conveniently using all of the arguments used to call an array in COSY, two C++ functions were written that are appended to the end of the converted program and used throughout : "GetIt" and "SetIt". These functions take advantage of member access functions of the COSY class [1]. The [] access is not implemented in the COSY class and so is not presently available in the conversion.

8. Examples

To provide a more comprehensive example of the conversion process, we list here the results of the translation of an entire procedure from the code COSY.FOX, a part of the beam physics environment of COSY [2]. The procedure, as coded in COSY language, has the following form:

```
PROCEDURE DL L ;                        {FIELD-FREE DRIFT}
   VARIABLE I 1 ; VARIABLE J 1 ;
   IF LUM#1 ; WRITE 6 ' *** ERROR, CALL UM BEFORE ELEMENT' ;
      QUIT 0 ; ENDIF ;
   DLACT MAP L ; IF NRAY#0 ; UMS ; DLACT MSC L ; ENDIF ;
   LOCSET 0 L 0 L 0 0 0 ; IF CONS(L)>0 ; DR := 0 ; ENDIF ;
   LOOP I 1 3 ; LOOP J 1 3 ; SSCR(I,J) := 0*DD(1) ; ENDLOOP ;
   SSCR(I,I) := 1+0*DD(1) ; ENDLOOP ; UPDATE 0 1 1 ;
   ENDPROCEDURE ;
```

After conversion, the C++ code will have the following form:

```
void pDL (Cosy& L) ;

namespace DL
{
        Cosy J (1) ;
        Cosy I (1) ;
        Cosy L (1) ;
}

void pDL ( Cosy& L)
{
        using namespace DL ;
        /*FIELD-FREE DRIFT*/
        if( LUM!=1 )
        {
                cout <<  " *** ERROR, CALL UM BEFORE ELEMENT "  ;
                break 0  ;
        }
        DLACT ( MAP, L )  ;
        if( NRAY!=0 )
        {
                UMS()   ;
```

[10] M. Berz. Differential algebra precompiler version 3 reference manual. Technical Report MSUCL-755, Michigan State University, East Lansing, MI 48824, 1990.

[11] J. Hoefkens. *Verified Methods for Differential Algebraic Equations*. PhD thesis, Michigan State University, East Lansing, Michigan, USA, 2001.

[12] E. Siever and S. Spainhour and N. Patwardhad. PERL in a Nutshell : A Desktop Quick Reference. O'Reilly & Associates, Inc., CA, 1999.

[13] The C++ Resources Network, http://www.cplusplus.com, 2000-2003.

Simulation Issues for RF Photoinjectors

E. Colby, V. Ivanov, Z. Li, C. Limborg
Stanford Linear Accelerator Center

Abstract. Accurate simulation of radiofrequency photoinjector performance remains a challenge in predicting the performance of future linear colliders and next-generation light sources. Calculated performance consistently and significantly exceeds measured performance. This discrepancy can be traced to two principal sources: measurement uncertainty and unrealistic assumptions and approximations made in the numerical calculations. The dynamics involved make the calculation challenging: electrons are accelerated from rest to relativistic velocities in millimeters. Strong wakefield and space charge effects require that fields be accurately represented on very small distance scales over large volumes. These issues will be discussed, the results of detailed code-to-code comparisons for tracking and particle-in-cell codes will be shown, and recommendations for further tests and improvements will be made.

1. Introduction

The RF photoinjector is an electron source that is widely used for particle accelerators and radiation sources. The ultimately performance of these devices is often directly linked to the beam quality produced by the electron source. Consequently, accurate simulation of photoinjector performance is an important part of designing and improving these devices.

Simulations have routinely predicted beam quality (in particular, the transverse emittance) that is significantly better than has been measured on real devices. There are two principal reasons for this: first, simulation codes make approximations of one kind or another and hence do not accurately reproduce the physics, and second, measurement uncertainty can be quite large for the figures-of-merit generally used to describe injector performance. It is the purpose of this paper to assess the importance of certain specific approximations made in a widely used tracking code, Parmela[1], and to discuss means for assuring that measurement and simulation are comparable.

Before proceeding with a discussion of the physics and simulation issues, it is worth commenting on the second source of the discrepancy, measurement uncertainty. Virtually all measures of beam quality from electron injectors depend on reducing a video image of the beam to statistical quantities. This image has added noise from dark current (field emission current not related to the photoemitted beam) and x-ray hits to the video system, resulting in bright pixels in the video image. Background image subtraction and despeckling algorithms generally are effective in dealing with these problems. However, the process of reducing the remaining beam profile to statistical quantities is generally

either accomplished by computing RMS quantities directly or by least-squares-fitting a particular distribution (e.g. a Gaussian) to the measured profile. Both procedures make use of the squares of deviations—either from a mean or from the fitted curve—and are therefore quite sensitive to the outlying data points. This has led to a large collection of filtering techniques (thresholding, low pass filtering, and so on) to suppress the effects of outliers, through which a fair amount of subjectivity and variation can enter the measurements.

Two actions are recommended to aid in comparing experimental measurements to simulation, and in comparing data with other laboratories. First, the fitting procedures, including background subtraction, despeckling algorithms, baseline subtraction, filtering and fitting should be presented, in detail, together with the reduced data. Second, for comparison with simulation, the experimental measurement should be simulated in as much detail as is possible, and the simulated measurement compared against the actual measurement. Many simulation codes have canned algorithms for computing emittance, spot size, and so on, but these are often too idealized to be directly comparable. For example, if quadrupole scan emittance data is to be compared to simulation, beam propagation should be simulated at the same quad strengths used in the actual measurement and the simulated spot sizes compared with the actual data.

2. Photoinjector physics

The dynamics of rf photoinjectors pose a challenge to desktop-scale computer simulation for several reasons. The electron beam is produced essentially at rest by photoemission and accelerated in very strong electric fields ($\sim 10^8$ V/m) to relativistic velocities in millimeters. Within this very short period of time, space charge forces rapidly expand the bunch by almost an order of magnitude in volume, resulting in rapid emittance growth in all dimensions. In addition, acceleration is so rapid that the image charges on the emission surface remain nearly fixed in position, viewed from the bunch reference frame, resulting in strong retarding forces on the emitted bunch. During this same period, the first emitted electrons are accelerated to a significant fraction of the speed of light before the last electrons are emitted giving rise to a time- and space-dependent focussing of the tail of the bunch from the induced magnetic field.

Rf photoinjectors produce short, dense bunches because the acceleration gradient is high enough that the electron bunch produced closely follows the spatial distribution of the laser pulse that produced it. The photoemission process is complicated by surface roughness effects and surface contaminants and is generally poorly understood and specific to each photoinjector cathode. Variations in emitted current density result in additional electrostatic potential energy being stored in the charge distribution than if the emission had been uniform, and this added energy generally results in increased emittance as the distribution evolves. Accurate determination of beam quality therefore depends on using measured current density functions for the cathode in question.

Alignment and harmonic content in the fields of the rf cavities and the external focussing system can also strongly impact performance. While simulating misalignment of components is routine, handling harmonic content requires 3D computations, is more troublesome, and is less frequently tackled.

3. Photoinjector simulation

Numerous suitable codes exist for tracking and particle-in-cell simulation of rf photoinjectors. Parmela is the most widely used code for simulating rf guns, having been developed and extensively used at Los Alamos, and used in the design and analysis of the much-copied Brookhaven gun. It is presently used as the most detailed model for the LCLS gun[2], motivating this study. Published benchmarking of photoinjector codes is comparatively rare in the literature, and is a critical step to predicting performance.

Some of the more significant approximations made in Parmela include

1. Space charge forces are strictly electrostatic (test problem 1)
2. No wakefield effects are included except image charge effects on the cathode
3. Retardation effects are neglected (test problem 1)

In addition, common simulation practice often makes the following idealizations:

4. The microwave excitation in all cavities has the same amplitude
5.   All injector components possess strictly axisymmetric, aligned fields (test problem 2)
6. Electron emission process is highly idealized (test problem 3)

In light of the strong velocity sheer present during emission, approximations (1) and (3) are invalid and were studied further. Approximation (1) has already received close attention in the context of electron beams in uniform motion[3] but not in the presence of large velocity sheer. Wakefield effects are also significant, but require detailed PIC modeling and will be studied and published in a subsequent paper. Field balance problems (4) have been studied and published elsewhere[4]. The approximation of rf cavity fields as axisymmetric (5) is also potentially significant and was studied. Nonuniform emission current density (6) has been studied with high spatial frequency checkerboard patterns[5] and measured laser profile data[6], and has been studied here from the vantage point of higher order mode content in the laser pulse.

Three test problems have been constructed to assess the impact of approximations (1), (3), (5), and (6). These problems are chosen to have parameters similar to the LCLS gun, but to be as abstracted and simple as possible, and in the case of test problem 1 to be amenable to fast, unambiguous calculation with very different simulation methods (e.g. tracking and PIC codes).

4. Test problems with results

4.1.   Test problem 1: Electrostatic and retardation-free approximations

The purpose of this problem is to test the electrostatic approximation of space charge forces and the influence of relativistic retardation effects during emission. This approximation is invalid in situations where the beam radiates (bends, strong quads), and at emission (where internal velocity sheer occurs, see Figure 1). This problem tests the latter circumstance.

Conceptually, the problem is similar to the emission process in the LCLS gun, but with rf effects suppressed by choosing a low working rf frequency. A bunched beam is

emitted from a perfect conductor under a strong accelerating field and tracked until the mean velocity reaches <β>=0.9. Radial RMS spot size, normalized emittance, bunch length, momentum spread, momentum/coordinate correlation function and <β> are computed as a function of <z>. Bunch space charge fields (E_z, E_r, $B_θ$) are plotted shortly after the entire bunch has been emitted.
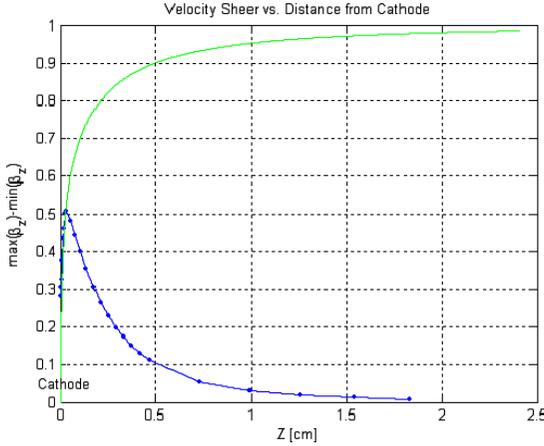
Figure 1. Velocity sheer (max($β_Z$)-min($β_Z$)) and mean bunch velocity <$β_Z$> versus mean bunch position <z>.

The details of the problem are: an acceleration gradient of E_z=100 MV/m (peak on cathode) is used, driven at 100 MHz. The electron bunch has a charge Q= 1nC, and is uniformly distributed in space and time in a 1 mm radius x 10 ps long cylinder. The beam is launched with 1 eV energy, strictly longitudinal, and is otherwise cold. (i.e. no transverse momentum, no energy spread) The beam is launched at the crest of the rf wave. No magnetostatic focussing fields are to be applied. Grid density and number of macroparticles should be chosen to achieve results that do not depend on the grid size or number of macroparticles.

Space charge fields are to be plotted at a single time ("snapshot fields") for the region around and within the bunch for PIC codes, and at the position of each macroparticle for tracking codes. E_r, E_z, and $B_θ$ should be plotted as a function of z. Also, compute: <β>, $ε_r$, $σ_r$, $ε_z$, $σ_z$, $σ_p$, Corr(z,p), and <z> every picosecond and record the 5D (or 6D) phase space when <β>=0.9. Sums over the distribution are to be performed for all quantities taken at a single time ("snapshot" emittances), not a single z location. RMS definitions of all quantities are to be used:

$$\langle x \rangle \equiv \sum_i^N q_i x_i / \sum_i^N q_i$$

$$\sigma_r = \sqrt{\langle r^2 \rangle - \langle r \rangle^2}, \mu_r = \langle r \rangle, \mu_{pr} = \langle p_r \rangle, p_r = \gamma\beta_r$$

$$\varepsilon_r \equiv \sqrt{\langle (r - \mu_r)^2 \rangle \langle (p_r - \mu_{pr})^2 \rangle - \langle (r - \mu_r)(p_r - \mu_{pr}) \rangle^2}$$

$$\sigma_z = \sqrt{\langle z^2 \rangle - \langle z \rangle^2}, \sigma_p = \sigma_{\gamma\beta_z} = \sqrt{\langle (\gamma\beta_z)^2 \rangle - \langle \gamma\beta_z \rangle^2}$$

$$Corr(z, p) \equiv < z\gamma\beta_z > /(\sigma_z \sigma_p)$$

Results of this test problem are displayed in Figures 2 and 3 below. The Los Alamos version of Parmela (referred to here as "Parmela-lanl") and the UCLA version of Parmela[6] (referred to here as "Parmela-ucla") are compared against Magic-2D[7] and Maxwell-T[8]. Space charge field strengths are displayed for Parmela-ucla and for Magic-2D in Figure 2. Particle tracking is completed for all four codes and is displayed in Figure 3.

There is good agreement of electric field strengths at the bunch extremities, but since Parmela uses the electrostatic approximation, B_θ is zero everywhere, in stark contrast to the computed values shown from Magic-2D. Note that field plots from Magic-2D are plotted as functions of r and z, whereas fields from Parmela are plotted as functions of the macroparticle coordinates. The fields at the outer radius of the bunch compare reasonably well, with Magic-2D giving $E_r \sim 6.3$ MV/m, and Parmela giving $E_r \sim 5.9$ MV/m. The total space charge and image charge fields at the tail of the bunch also compare reasonable well, with Magic-2D giving $E_z \sim 23.8$ MV/m, and Parmela $E_z \sim 23$ MV/m. Magic-2D shows a maximum magnetic field of $B_\theta \sim 0.015$T, which gives the same approximate focusing force as a radial electric field of $E_r \sim 3.4$ MV/m, and results in a sizable reduction in the effective radial space charge force when the bunch still has large velocity sheer.
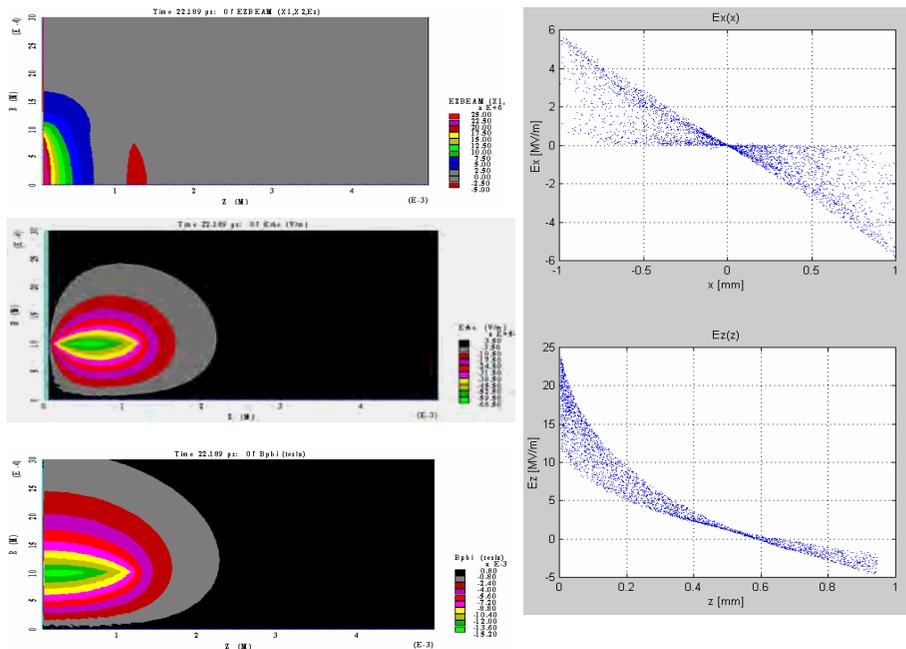
Figure 2. Space charge and image charge fields for test problem 1, evaluated by Magic-2D (left) and Parmela-ucla (right). Upper left: E_z, middle left: E_r, lower left: B_θ. Upper right: $E_x(x)$, lower right: $E_z(z)$.

Plots of the moments, spot sizes, and emittances are shown in figure 3 below. Agreement is surprising good amongst the four simulation codes, which use very different calculation methods. Still more surprising is that the transverse emittances are consistently better in the PIC code simulations, which include the time-and-space dependent effects of the sheer-induced magnetic field, and also correctly handle

retardation effects with the cathode image charges. Agreement on the spot sizes, momentum spread, and momentum/phase correlation are all excellent.

It is reasonable to conclude from these results that omission of retardation effects and the velocity sheer fields in Parmela result in quite small errors in the spot size, bunch length and momentum spread, and to at most a ~20% overestimation of the transverse emittances. It is likely that some of this discrepancy is traceable to the very different computation methods used and not to the physics approximations. Finally, it is noteworthy that the two Parmela variants agree well with each other, and the two PIC codes agree reasonably with each other, suggesting that the differences are indeed algorithmic in origin.
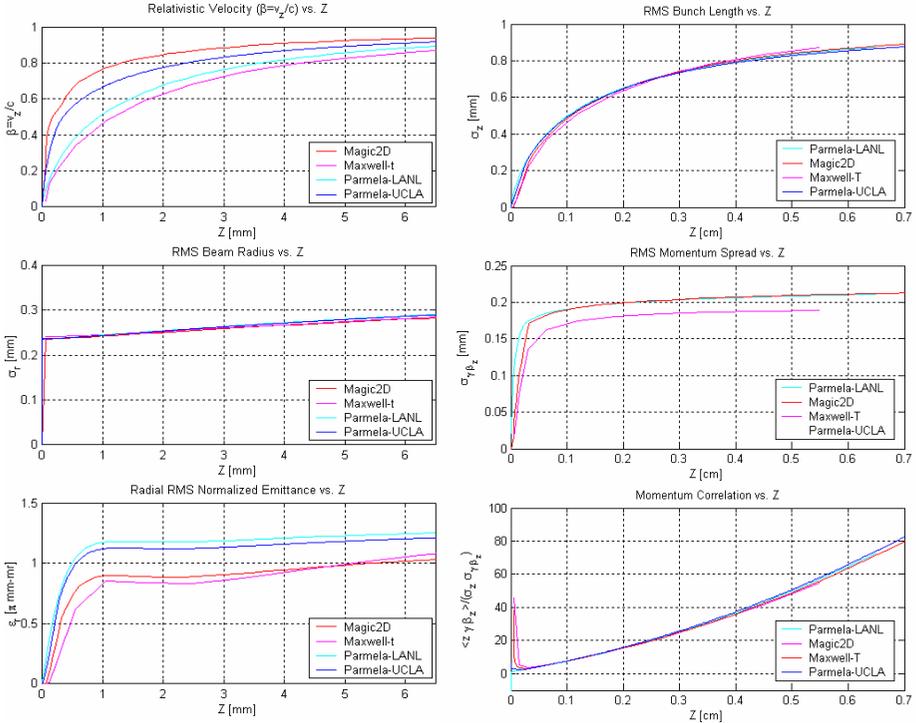
Figure 3. Transverse (left) and longitudinal (right) beam quantities. Upper left: $\langle\beta\rangle$, middle left: σ_r, lower left: ε_r. Upper right: σ_z, middle right: σ_p, lower right: $Corr(z,p)$.

4.2. Test problem 2: Axisymmetric microwave field approximation

This second test problem is designed to address the importance of approximating the non-axisymmetric rf fields of the LCLS gun as axisymmetric. Efforts have been made to suppress the dipole asymmetry of the fields, leaving the quadrupole as the leading error term.

Conceptually, this test problem is quite specific to the geometry of the LCLS gun[2]. Beam is emitted and propagated through the 1.6 cell s-band LCLS gun, stopping at the exit of the gun, but using two different maps to represent the rf fields. The first map is strictly axisymmetric, generated by revolving a 2D map (generated by Superfish[9]) about the z-axis. The second map is fully 3D, generated from a fully 3D gun model

computed with Mafia[10], including the laser ports, and the power couplers, each of which induce quadrupole field structure.

Specifically, the gun gradient will be taken to be $E_Z=100$ MV/m, the frequency $f_{rf}=2856$ MHz, and the geometry will be that of the LCLS gun. Solenoid focusing is to be used with a peak on-axis field strength of 2.7 kG. The electron bunch will be 1mm radius by 10ps length, uniformly distributed, but should have zero charge, to suppress space charge effects. The beam is to be launched with only a longitudinal velocity corresponding to 1 eV kinetic energy. The grid density and macroparticle number should be chosen to give results independent of either. Care must be exercised that the two field maps are equivalent in resolution and spatial extent.

Compute and plot the "difference fields", defined as the difference of the 3D and 2D maps on a component-by-component basis: $\Delta E_x=E_X(3D)-E_X(2D),\ldots, \Delta B_z=B_Z(3D)-B_Z(2D)$. Compute ε_r, σ_r, at the exit of the gun and compare.
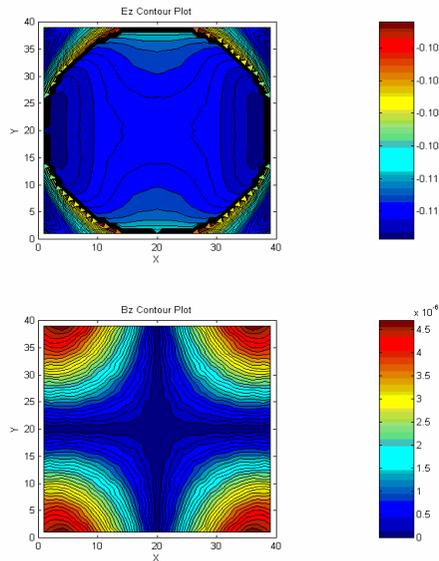
Figure 4. Difference fields ΔE_z (top) and ΔB_z (bottom).

Computation of the beam transverse emittance (defined in test problem 1 above) with the 2D and 3D maps is summarized in Table 1. For nominal LCLS beam parameters, there is no statistically significant emittance increase for the 3D field map case (N=10000, observed increase is 0.8%). As a check of this negative result, the initial beam size was increased five-fold and the simulations run again. The much larger beam size significantly increases the rf-induced emittance growth. This case yielded a 10% increase in the emittance due to the added multipole errors, and is also displayed in Table 1.

The conclusion is that the rf cavity asymmetries induced by the input power coupler and laser ports do not significantly impact the emittance.

Table 1. Transverse emittance and spot size at gun exit for the 2D vs 3D comparison.

	Exit Spot Size		Exit Emittance		Exit Energy $[m_ec^2]$	
	2D	3D	2D	3D	2D	3D
R=1mm	0.849 mm	0.878 mm	0.387 π μ	0.390 π μ	10.75	11.06
R=5mm	3.626 mm	3.589 mm	3.319 π μ	3.689 π μ	10.85	11.01

4.3.   Test problem 3: Uniform current emission approximation

The purpose of this test problem is to test sensitivity of beam transverse emittance to variations in the emitted current density. Variations arise from several factors: laser nonuniformity, cathode roughness leading to variations in Shottky enhancement, and cathode contamination leading to variations in quantum efficiency.

Conceptually this specific problem addresses nonuniformities of the laser, as would be caused by higher-order modes being generated in the laser by, scattering, variations in amplifier gain with position, optics damage, and so on. Electron beam distributions are derived by standard rejection techniques to have Gauss-Laguerre form[11].

Parmela-lanl will be used for this case as it has an advanced 3D mesh-based space charge calculation method. The gradient will be E_Z=120 MV/m, frequency f_{rf}=2856 MHz, and geometry that of the LCLS gun[2]. Solenoid focusing is to be used with a peak on-axis fields strength of 2.7 kG. The bunch charge will be Q= 1nC, and distributed approximately uniformly in time with a 10 ps length and 0.7 ps risetime, but with transverse distributions specified by the Gauss-Laguerre eigenfunctions with waist parameter w_o=0.6*sqrt(2) mm and a distribution cutoff of r=1.2 mm. The laser striking the cathode is presumed to be focussed on the cathode, so the phase front radius of curvature R=∞, and the Guoy phase angle ϕ_o=0. Launch energy: 1 eV, strictly longitudinal, with 0.4 π μ thermal emittance added. The number of macroparticles, and mesh density should be chosen as required to achieve results that do not depend on either. Propagate the beam to the exit of the gun, and compute the transverse emittance at this location (using the definitions given in test problem 1).
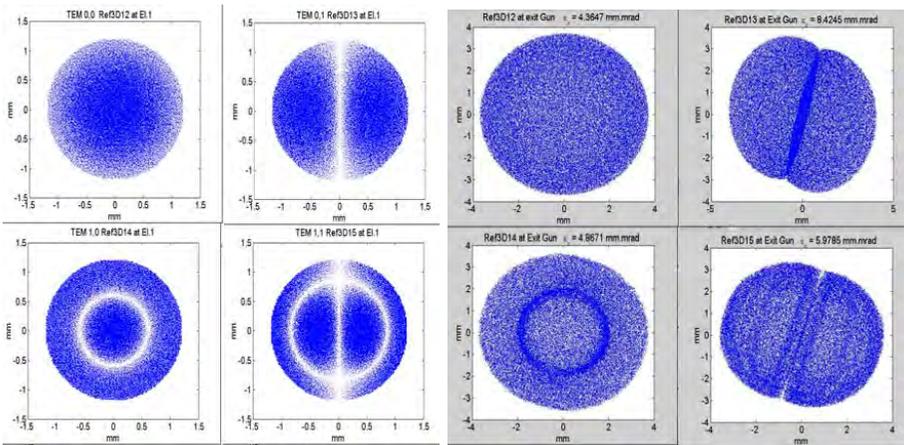
Figure 5. Transverse charge distributions at the cathode (left) and at the exit of the gun (right).

Transverse beam profiles both at launch and at the gun exit are shown in Figure 5 below. Washout of the initial distributions has clearly begun, and the action of the solenoid focussing are visible as a rotation of the distribution about the axis.

Transverse emittances for the lowest four Gauss-Laguerre modes are summarized in Table 2 below. The cases computed here correspond to excitation of single laser modes, which gives some indication of what types of distortion (azimuthal, low spatial frequency) are most damaging to the beam quality. In reality a large collection of these modes combine to make the actual laser profile used to illuminate the cathode, and realistic simulations will require examining more realistic combinations of these modes, which will be the subject of future work.

Table 2. Transverse emittance at exit of gun for lowest four Gauss-Laguerre modes.

Laser Mode	Exit Emittance
Radially uniform distribution	2.84 π μ
TEM_{00} (Gaussian)	4.37 π μ
TEM_{01} (azimuthal modulation only)	8.42 π μ
TEM_{10} (radial modulation only)	4.87 π μ
TEM_{11} (both radial and azimuthal modulation)	5.98 π μ

5. Conclusions

These three test problems address four of the approximations commonly made in simulating rf photoinjector performance. The electrostatic and retardation-free approximations are most dramatically violated shortly after emission, but as test problem 1 shows, the time during which these effect apply is so short that their inclusion has little impact on beam quality. The approximation of the rf fields as strictly axisymmetric is also reasonable for the symmetrized case of the LCLS gun. Accurate modelling of injector performance, however, depends directly on the fidelity with which the electron distribution is initially produced. Detailed measurements of emission current density over the active area of the cathode is essential data that has to be incorporated into injector simulation before accurate results can be expected.

During the course of these studies, several difficulties arose from the very different nature of the simulation codes used, and from the challenge of getting exactly comparable results from these codes.

Benchmarking codes and conducting fully integrated multi-code simulations are and will remain essential tasks for making reliable simulations of rf photoinjectors and for estimating the overall performance of systems which use rf photoinjectors. We therefore offer these recommendations with the hope of making both processes more efficient and reliable:

Recommendation 1. Adopt the Self-Describing Data Sets (SDDS)[12] format for all input and output files whose primary user is another program. Often-edited input files, such as beamline descriptions or run control files, or output files intended solely for human consumption, should remain in easily understood formats that need not be standardized. Interchange between users of different simulation codes, either for comparison purposes or for continuing a complex multi-step simulation process, will be greatly facilitated by the adoption of a standard format.

Recommendation 2. Devise abstracted, minimal test problems which reveal quantitatively the importance of the various approximations made in the simulations, and *publish these problems and their results with sufficient detail to be exactly replicated*. Three such problems are presented here.

Studies of wakefield effects will address the importance of one of the last remaining approximations made to the physics, and will be conducted for the LCLS gun in the near future. Other issues, specifically numerical in origin, also deserve attention such as the integrity of field maps and the symplecticity and convergence of the integrator.

6. Acknowledgements

References

[1] Parmela is "Phase And Radial Motion in Electron Linear Accelerators", and is maintained and distributed by the Los Alamos Accelerator Code Group (LAACG), contact laacg@lanl.gov.

[2] LCLS Design Report, SLAC-R-521, December (1998).

[3] H. Liu, "Analysis of space charge calculation in Parmela and its application to the CEBAF FEL Injector design", CEBAF-PR-93-008, (1993).

[4] C. Limborg, *et al*, "New Optimization for the LCLS Photo-Injector", Proceedings EPAC 2002, Paris, France, June 3-7 (2002).

[5] F. Zhou, "Experimental characterization of emittance growth induced by the nonuniform transverse laser distribution in a photoinjector", *Phys. Rev. Special Topics—AB*, **5**, 094203, (2002).

[6] E. Colby, "Design, Construction, and Testing of a Radiofrequency Photoinjector for the Next Linear Collider", PhD Thesis, U.C.L.A., (1997).

[7] "User-Configurable MAGIC for electromagnetic PIC calculations," *Computer Physics Communications* **78** (1995) 54-86. Magic-2D is a commercially produced code maintained by the Mission Research Corporation, http://www.mrcwdc.com.

[8] Maxwell-T is a 2.5D PIC code created and maintained by V. Ivanov, contact: ivanov@slac.stanford.edu.

[9] K. Halbach and R. F. Holsinger, "SUPERFISH -- A Computer Program for Evaluation of RF Cavities with Cylindrical Symmetry," Particle Accelerators 7 (4), 213-222 (1976). Superfish is maintained and distributed by the LAACG, contact: laacg@lanl.gov.

[10] M. Bartsch, *et al*, "MAFIA release 3: the new version of the general purpose electromagnetic design code family", DESY Internal Report M-90-05K, Deutsches Elektronen-Synchrotron, Hamburg, Germany (1990). MAFIA is a commercially marketed code maintained by CST GmbH, contact: info@cst.de.

[11] P. Goldsmith, *Quasioptical Systems*, p. 27, IEEE Press, New York, (1998).

[12] M. Borland, "Application Programmer's Guide for SDDS Version 1.4", APS LS Note. See http://www.aps.anl.gov/asd/oag/manuals/SDDStoolkit/node2.html.

Enhancements to iterative inexact Lanczos for solving computationally large finite element eigenmode problems

J. F. DeFord and B. L. Held

Simulation Technology & Applied Research, Inc.

Abstract. The inexact Lanczos method implemented in Omega3P is a highly efficient technique for solving large generalized eigenmode problems on parallel computers. For large finite element problems the iterative solution of the shifted linear system using the preconditioned conjugate gradient (PCG) method dominates the computational time. Recent work has shown that the convergence rate of PCG can be greatly enhanced by introducing scalar degrees of freedom representing the electric potential. This paper presents the implementation and application of the inexact Lanczos method with A-V basis functions in the parallel OM3P solver. Results for large RF cavity problems (> 1 million DOFs) demonstrate significant reductions in computational time (more than a factor of 10 in some cases) as compared to the original implementation. The application of a deflation algorithm that improves the performance of the procedure on mode clusters is also illustrated.

1. Introduction

Large eigenmode problems are commonly encountered in accelerator component design, particularly in the analysis and optimization of RF cavities. To provide optimal guidance in the design process of next generation cavities eigenfrequency error must be less than one part in 10^4. Moreover, examination of field stress-related issues typically requires very fine conformal finite-element meshes in various areas of a cavity. These two realities lead to very large finite-element meshes (often exceeding 10^6 elements), with correspondingly large numbers of unknowns ($10^6 - 10^7$ or more). Given the typical matrix fill of between 25 and 60 non-zeros per row, memory needed for matrix storage alone can easily exceed 10 GB, and computational requirements are such that parallel computers are needed for problem solution.

Recent work by Sun, et al. [1] has lead to a novel method for the efficient solution of large eigenmode problems on parallel computers, and this method has been implemented in the Omega3P solver. The method, called ISIL (Inexact Shift and Invert Lanczos), uses a standard shift-invert Lanczos process to solve the generalized eigenvalue problem, except that the solution of the shifted linear system is done iteratively only to low precision (10^{-2} residual). The inexact Lanczos process will usually stagnate before the desired eigenvector residual is obtained, at which point the Jacobi Orthogonal

Component Correction (JOCC) procedure from the Jacobi-Davidson method is used to improve the residual of the mode.

With an appropriate choice of linear solver, such as PCG, ISIL scales well to many processors. For large problems the computational time is dominated by the time required to solve the shifted linear system. The performance of PCG on matrices generated using the finite element method in electromagnetics is strongly dependent on the particular problem and, of course, the preconditioner. Scalability requirements limit the options for preconditioners, with some of the more effective strategies on serial machines involving factorization of the coefficient matrix being less desirable when the problem is to be solved on a parallel computer. Given these constraints, the Gauss-Seidel method is a good choice for preconditioner, with a number of steps from 1 to 5 depending on the situation (each step requires essentially two matrix-vector multiplies). With this linear solver, however, it can still take many thousands of iterations to reduce the initial residual by a factor of 100. Indeed, even in some cases involving relatively small meshes, it is possible to exceed 50,000 iterations without suitable convergence.

Recent work in the finite element community has shown that the poor convergence of the PCG method on matrices obtained for electromagnetics problems is a result of two factors [3]: static modes in null space of the curl operator, and oversampling of relatively low frequency valid modes. Multigrid methods are being investigated for solution of the latter problem [4], and use of the vector-scalar (so-called A-V) basis set has been shown to effectively mitigate the former. In this paper we investigate the utility of the A-V basis set in connection with ISIL.

2. A-V basis set

In the A-V finite element formulation the electric field is given by

$$E = A + \nabla\Phi \tag{1}$$

where the vector potential is represented using a standard vector basis set used in the E-formulation that has unknowns associated with edges and faces of the mesh. The scalar potential is supported by nodal degrees of freedom. As noted in [2] it is not necessary to form the resulting coefficient matrix explicitly, and the generalized eigenvalue matrix equation can be written in the form

$$A'x = \lambda B'x \tag{2}$$

where

$$A' = [C^T A C] \tag{3}$$

$$B' = [C^T B C] \tag{4}$$

$$C = [I\ G] \tag{5}$$

The A and B matrices in Eqns. 3 and 4 are the contributions to the standard E-formulation finite-element coefficient matrix from the curl-curl and dot product terms of the functional, respectively. Assuming the solution vector x is ordered such that the n vector unknowns all appear first, followed by the m nodal unknowns, then the I matrix is the n x n identity matrix, and G (n x m) has a maximum of two non-zeros per row.

As noted in previous work the coefficient matrix that is obtained using the A-V basis set is rank deficient. Although this is not directly relevant to our application, it may be the cause of divergent behavior in PCG that is sometimes observed if the desired solution residual for the shifted linear system is small enough. In practice one does not run into

this problem with ISIL since the stopping point residual is relatively large, but it can limit the utility of the basis set (at least as we are using it) in the exact Lanczos process.

3. Parallel implementation of A-V basis

ISIL using the A-V basis set has been implemented in the OM3P parallel eigensolver. While the authors of [2] suggest a relatively sophisticated decomposition of the matrix to allow incomplete Cholesky preconditioning, we instead employ the matrices in their original form. OM3P uses the PETSc library [5] for its parallel data structures and numerical solvers, and the A' and B' matrices of Eqn. 2 are represented as symbolic matrices in this library. In this form one can provide functions that perform the required operations (such as matrix-vector multiplication) for the symbolic matrix, and it is treated as a regular distributed matrix by the PETSc solvers.

In a parallel implementation it is advantageous to explicitly form both C and C^T. The single instance of each matrix is then referenced by both the A' and B' symbolic matrices. Supported in this way the A-V basis set introduces a 5%-15% increase in unknown count (depending on the order of the vector basis set), and a similar increase in aggregate memory requirements for the matrices. The primary disadvantage of the symbolic representation of the A' and B' matrices is that there are three vector scatters required per matrix-vector multiplication step in PCG (one for each matrix in the product), as opposed to one if they were explicitly formed. However, as compared to roughly a factor of two increase in memory requirements for explicit formation of A' and B', and a comparable increase in the amount of computation required for a matrix-vector multiplication, it was seen as a worthwhile trade-off to leave them in symbolic form.

The implementation of the Gauss-Seidel preconditioner also requires some special care for the symbolic A-V matrices. Since a single Gauss-Seidel iteration requires the same vector information as a regular matrix-vector multiplication, this operation has the same communication requirements, that is, three vector scatters per step. To avoid this additional communication cost we perform the preconditioning using only the original E-formulation matrices (A and B), requiring only one scatter operation per step.

4. Deflation in OM3P

Deflation methods are used to combat the tendency of the Lanczos method to produce duplicate eigenpairs. The Hessenberg matrix that is generated by the Lanczos and Arnoldi processes has the property that its eigenvalues rapidly converge to the extreme eigenvalues of the original system. The deflation/locking procedure diagonalizes a particular row/column of the Hessenberg, leaving the corresponding eigenvalue on the diagonal. The row/column are the then removed from the matrix, and the Lanczos/Arnoldi process continues. Performing this operation as eigenpairs converge, together with ensuring that new Lanczos vectors are orthogonal to the existing subspace, effectively bars duplicates from forming. This procedure also allows the unblocked Lanczos process to extract degenerate eigenpairs.

There are a variety of methods for implementing deflation. We chose a relatively simple technique described in a book by Dongarra, et al. [6], called the Orthogonal Deflating Transformation. It was not clear to the authors at the outset whether the usual methods would work in ISIL, because only approximate information about the Krylov subspace is being generated. In practice we found that using the simple Dongarra method in ISIL can still allow duplicates to be generated if the shifted linear system is only solved to 10^{-2} residual tolerance. However, lowering the tolerance to 10^{-3} eliminates the problem in all cases we have considered, and so OM3P uses the latter tolerance by default. It often takes 1.5 - 2 times as long to reach residuals of 10^{-3} as compared to 10^{-2},

so there is some motivation to consider deflation methods that work reliably at the latter residual level, but the authors have not investigated this possibility.

5. Numerical examples

We have run several examples that illustrate the behavior of the A-V basis set as implemented in OM3P compared with the original implementation ISIL using the E-formulation basis. Comparisons were made using a variety of meshes generated for a single cell of the RDDS-1 cavity shown in Fig. 1.

Figure 1. Single cell of the Round Damped Detuned Structure RDDS-1 cavity proposed for the NLC (1/8 model). Photograph courtesy of K. Ko, SLAC.

Results for a range of element counts in 1.5 cells of the RDDS-1 cavity are summarized in Table 1 and Fig. 2. The meshes were fairly uniform. Experience has shown that widely varying element sizes can cause additional ill-conditioning of the matrix that can adversely affect PCG, although this problem seems to be substantially worse using the standard vector basis set than it is with the A-V basis set. All cases were run without JOCC so they only compare the behavior in the Lanczos part of the algorithm.

Table 1. Average number of PCG iterations required to reach a residual of 10^{-3} for standard vector basis (E) and for the A-V basis for various element counts in 1.5 RDDS-1 cavities. The final residuals shown are the values obtained when the inexact Lanczos process stagnates.

Element Count	DOF Count		Average PCG Iterations Per Lanczos Step	Final Residual (No JOCC)
30,179	E	38,770	2837	2.09×10^{-4}
	A-V	43,780	126	9.41×10^{-7}
55,467	E	62,425	2199	2.31×10^{-4}
	A-V	71,125	151	4.79×10^{-7}
232,441	E	267,437	1739	1.69×10^{-4}
	A-V	306,122	187	1.05×10^{-7}
331,490	E	381,809	2173	1.48×10^{-4}
	A-V	436,764	209	1.04×10^{-7}
756,781	E	876,835	2077	1.19×10^{-4}
	A-V	1,004,351	277	3.84×10^{-8}

The residuals shown in Table 1 illustrate another characteristic of A-V when used with ISIL – A-V routinely generates substantially better residuals than the E-formulation at a similar point in the Lanczos process, and for every case listed in the table it took fewer Lanczos steps to reach stagnation with A-V than in the corresponding E formulation case.
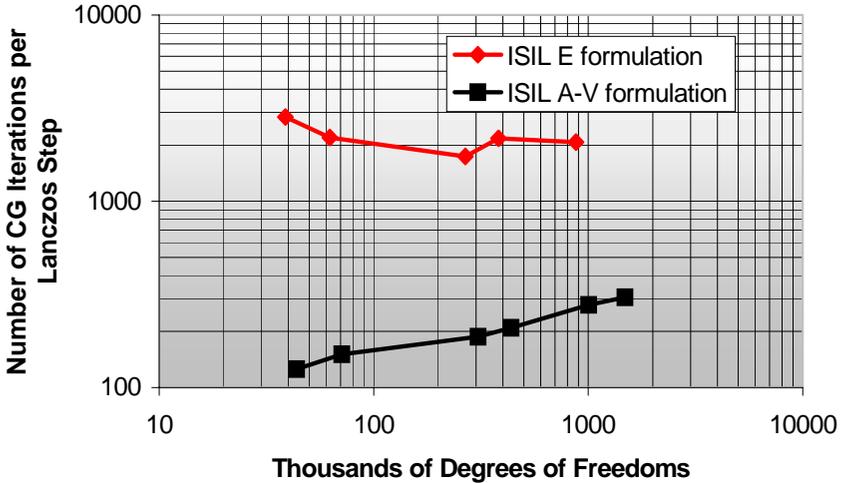
Figure 2. Comparison of the number of PCG iterations per Lanczos step for the regular E-formulation vector basis set used in the original implementation of ISIL, and the A-V basis set used in OM3P. The geometry used for these problems was 1.5 cells of the RDDS-1 structure shown in Fig. 1.

To illustrate the use of deflation in ISIL, we analysed a 5-cavity structure that was designed and built at the National Superconducting Cyclotron Laboratory [7]. The structure has several modes clustered around 800 MHz, and in Table 2 we compare the performance of ISIL with deflation to ISIL without deflation but with a sophisticated re-shifting mechanism used in an early version Omega3P.

The mesh for this example contained about 190K elements, and the model is essentially 2-D (run as a narrow wedge in 3D). The structure and fields for one of the modes are shown in Fig. 3. The tabulated results show that while deflation allowed all of the modes to be extracted in only 9 Lanczos steps, the re-shifting strategy actually missed one of the modes and required more the twice the number of Lanczos steps. The increase in the step count comes about because Lanczos is restarted after each mode when the shift point is changed.

Table 2. Comparison of mode cluster extraction for ISIL using deflation with ISIL using reshifting. An "x" in the column indicates the corresponding mode was found.

Method	Mode Number					Total Number of
	1	2	3	4	5	Lanczos Steps
ISIL w/ reshifting	x	x		x	x	20
ISIL w/ deflation	x	x	x	x	x	9
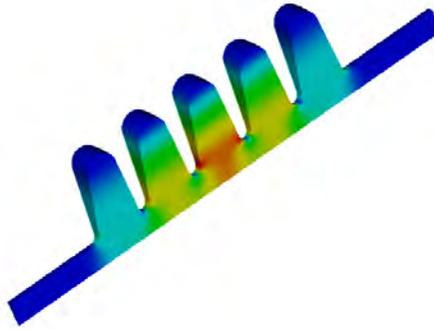
Figure 3. Electric field magnitude for the fundamental mode at 794 MHz in the five-cell cavity. Model courtesy of F. Marti, NSCL.

Using OM3P we have also run this problem with 4.76M elements (6.6M DOFs) on six processors hooked up on a standard interoffice network of 2.2 GHz Xeon processors. It took 8 hrs of elapsed time to extract the first mode, and a total of 20 hrs to extract all five modes.

6. Conclusions

The use of the A-V basis set has been shown effective at reducing the computational work associated with solving linear systems arising from the application of the finite-element method in electromagnetics. In this paper we demonstrate that this basis set may be applied in the ISIL method, leading to dramatic reductions in the number of PCG iterations that are required to obtain solution to the shifted linear system. Also illustrated is the application of a standard deflation technique to ISIL for a mode cluster extraction problem, showing superior performance to a re-shifting algorithm.

References

[1] Y. Sun, N. Folwell, Z. Li, and G. Golub, "High precision accelerator cavity design using the parallel eigensolver Omega3P," *Proceedings of The Annual Review of Progress in Applied Computational Electromagnetics 2002*.

[2] R. Dyczij-Edlinger, G. Peng, and J. F. Lee, "A fast vector-potential method using tangentially continuous vector finite elements," *IEEE-MTT*, **46**, June, 1998, pp.863-868.

[3] R. Dyczij-Edlinger and O. Biro, "A joint vector and scalar potential formulation for driven high frequency problems using hybrid edge and nodal finite elements," *IEEE-MTT*, **44**, Jan., 1996, pp. 15-23.

[4] Y. Zhu and A. C. Cangellaris, "Hybrid multilevel/multigrid potential preconditioner for fast finite-element modelling," *IEEE Microwave and Wireless Comp. Let.*, **12**, Aug., 2002, pp. 290-292.

[5] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, "Efficient management of parallelism in object oriented numerical software libraries," *Modern Software Tools in Scientific Computing*, Editors: E. Arge and A. M. Bruaset and H. P. Langtangen, Birkhauser Press, 1997, pp. 163-202.

[6] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, Philadelphia, 2000.

[7] J. Popielarski, T. Grimm, W. Hartung, L. Kempel, F. Marti, "Higher order mode analysis of multicell axisymmetric superconducting cavities through measurements and numerical calculations," this proceedings.

New numerical methods for solving the time-dependent Maxwell equations

H. De Raedt, J.S. Kole, K.F.L. Michielsen, and M.T. Figge

Applied Physics - Computational Physics‡, Materials Science Centre
University of Groningen, Nijenborgh 4, NL-9747 AG Groningen, The Netherlands

Abstract. We review some recent developments in numerical algorithms to solve the time-dependent Maxwell equations for systems with spatially varying permittivity and permeability. We show that the Suzuki product-formula approach can be used to construct a family of unconditionally stable algorithms, the conventional Yee algorithm, and two new variants of the Yee algorithm that do not require the use of the staggered-in-time grid. We also consider a one-step algorithm, based on the Chebyshev polynomial expansion, and compare the computational efficiency of the one-step, the Yee-type and the unconditionally stable algorithms. For applications where the long-time behavior is of main interest, we find that the one-step algorithm may be orders of magnitude more efficient than present multiple time-step, finite-difference time-domain algorithms.

1. Introduction

The Maxwell equations describe the evolution of electromagnetic (EM) fields in space and time [1]. They apply to a wide range of different physical situations and play an important role in a large number of engineering applications. In many cases, numerical methods are required to solve Maxwell's equations [2, 3]. A well-known class of algorithms is based on a method proposed by Yee [4]. This finite-difference time-domain (FDTD) approach owes its popularity mainly due to its flexibility and speed while at the same time it is easy to implement [2, 3].

A limitation of Yee-based FDTD techniques is that their stability is conditional, depending on the mesh size of the spatial discretization and the time step of the time integration [2, 3]. Furthermore, in practice, the amount of computational work required to solve the time-dependent Maxwell equations by present FDTD techniques [2, 3, 5, 6, 7, 8, 9, 10, 11, 12] prohibits applications to a class of important fields such as bioelectromagnetics and VLSI design [2, 13, 14]. The basic reason for this is that the time step in the FDTD calculation has to be relatively small in order to maintain stability and a reasonable degree of accuracy in the time integration. Thus, the search for new algorithms that solve the Maxwell equation focuses on removing the

‡ http://www.compphys.org/

conditional stability of FDTD methods and on improving the accuracy/efficiency of the algorithms.

2. Time integration algorithms

We consider EM fields in linear, isotropic, nondispersive and lossless materials. The time evolution of EM fields in these systems is governed by the time-dependent Maxwell equations [1]. Some important physical symmetries of the Maxwell equations can be made explicit by introducing the fields $\mathbf{X}(t) \equiv \sqrt{\mu}\mathbf{H}(t)$ and $\mathbf{Y}(t) \equiv \sqrt{\varepsilon}\mathbf{E}(t)$. Here, $\mathbf{H}(t) = (H_x(\mathbf{r}, t), H_y(\mathbf{r}, t), H_z(\mathbf{r}, t))^T$ denotes the magnetic and $\mathbf{E}(t) = (E_x(\mathbf{r}, t), E_y(\mathbf{r}, t), E_z(\mathbf{r}, t))^T$ the electric field vector, while $\mu = \mu(\mathbf{r})$ and $\varepsilon = \varepsilon(\mathbf{r})$ denote, respectively, the permeability and the permittivity. Writing $\mathbf{Z}(t) = (\mathbf{X}(t), \mathbf{Y}(t))^T$, Maxwell's curl equations [2] read

$$\frac{\partial}{\partial t}\mathbf{Z}(t) = \begin{pmatrix} 0 & -\frac{1}{\sqrt{\mu}}\nabla \times \frac{1}{\sqrt{\varepsilon}} \\ \frac{1}{\sqrt{\varepsilon}}\nabla \times \frac{1}{\sqrt{\mu}} & 0 \end{pmatrix} \mathbf{Z}(t) \equiv \mathcal{H}\mathbf{Z}(t). \tag{1}$$

It is easy to show that $\mathcal{H}$ is skew symmetric, i.e. $\mathcal{H}^T = -\mathcal{H}$, with respect to the inner product $\langle \mathbf{Z}(t)|\mathbf{Z}'(t)\rangle \equiv \int_V \mathbf{Z}^T(t) \cdot \mathbf{Z}'(t)\,d\mathbf{r}$, where V denotes the system's volume. In addition to Eq.(1), the EM fields also satisfy $\nabla \cdot (\sqrt{\mu}\mathbf{X}(t)) = 0$ and $\nabla \cdot (\sqrt{\varepsilon}\mathbf{Y}(t)) = 0$ [1]. Throughout this paper we use dimensionless quantities: We measure distances in units of λ and expresss time and frequency in units of λ/c and c/λ, respectively.

A numerical algorithm that solves the time-dependent Maxwell equations necessarily involves some discretization procedure of the spatial derivatives in Eq. (1). Ideally, this procedure should not change the basic symmetries of the Maxwell equations. We will not discuss the (important) technicalities of the spatial discretization (we refer the reader to Refs. [2, 3]) as this is not essential to the discussion that follows. On a spatial grid Maxwell's curl equations (1) can be written in the compact form [11]

$$\frac{\partial}{\partial t}\mathbf{\Psi}(t) = H\mathbf{\Psi}(t). \tag{2}$$

The vector $\mathbf{\Psi}(t)$ is a representation of $\mathbf{Z}(t)$ on the grid. The matrix H is the discrete analogue of the operator $\mathcal{H}$. The formal solution of Eq. (2) is given by

$$\mathbf{\Psi}(t) = e^{tH}\mathbf{\Psi}(0) = U(t)\mathbf{\Psi}(0), \tag{3}$$

where $U(t) = e^{tH}$ denotes the time-evolution matrix. If the discretization procedure preserves the underlying symmetries of the time-dependent Maxwell equations then the matrix H is real and skew symmetric, implying that $U(t)$ is orthogonal [15]. Physically, the orthogonality of $U(t)$ implies conservation of energy.

There are two, closely related, strategies to construct an algorithm for performing the time integration of the time-dependent Maxwell equations defined on the grid [16]. The traditional approach is to discretize (with increasing level of sophistication) the derivative with respect to time [16]. The other is to approximate the formally exact solution, i.e. the matrix exponential $U(t) = e^{tH}$ by some time evolution matrix

$\tilde{U}(t)$ [16, 17]. We adopt the latter approach in this paper as it facilitates the construction of algorithms with specific features, such as unconditional stability [17].

If the approximation $\tilde{U}(t)$ is itself an orthogonal transformation, then $\|\tilde{U}(t)\| = 1$ where $\|X\|$ denotes 2-the norm of a vector or matrix X [15]. This implies that $\|\tilde{U}(t)\mathbf{\Psi}(0)\| = \|\mathbf{\Psi}(0)\|$, for an arbitrary initial condition $\mathbf{\Psi}(0)$ and for all times t and hence the time integration algorithm defined by $\tilde{U}(t)$ is unconditionally stable by construction [16, 17].

We now consider two options to construct the approximate time evolution matrix $\tilde{U}(t)$. The first approach yields the conventional Yee algorithm, a higher-order generalization thereof, and the unconditional schemes proposed in Ref.[11]. Second, the Chebyshev polynomial approximation to the matrix exponential [18, 19, 20, 21, 22, 23]. is used to construct a one-step algorithm [24, 25].

2.1. Suzuki product-formula approach

A systematic approach to construct approximations to matrix exponentials is to make use of the Lie-Trotter-Suzuki formula [26, 27]

$$e^{tH} = e^{t(H_1 + \ldots + H_p)} = \lim_{m \to \infty} \left(\prod_{i=1}^{p} e^{tH_i/m} \right)^m, \tag{4}$$

and generalizations thereof [28, 29]. Expression Eq. (4) suggests that

$$U_1(\tau) = e^{\tau H_1} \ldots e^{\tau H_p}, \tag{5}$$

might be a good approximation to $U(\tau)$ if τ is sufficiently small. Applied to the case of interest here, if all the H_i are real and skew-symmetric $U_1(\tau)$ is orthogonal by construction and a numerical scheme based on Eq. (5) will be unconditionally stable. For small τ, the error $\|U(t = m\tau) - [U_1(\tau)]^m\|$ vanishes like τ [29] and therefore we call $U_1(\tau)$ a first-order approximation to $U(\tau)$.

The product-formula approach provides simple, systematic procedures to improve the accuracy of the approximation to $U(\tau)$ without changing its fundamental symmetries. For example the matrix

$$U_2(\tau) = U_1(-\tau/2)^T U_1(\tau/2) = e^{\tau H_p/2} \ldots e^{\tau H_1/2} e^{\tau H_1/2} \ldots e^{\tau H_p/2}, \tag{6}$$

is a second-order approximation to $U(\tau)$ [28, 29]. If $U_1(\tau)$ is orthogonal, so is $U_2(\tau)$. Suzuki's fractal decomposition approach [29] gives a general method to construct higher-order approximations based on $U_2(\tau)$ (or $U_1(\tau)$). A particularly useful fourth-order approximation is given by [29]

$$U_4(\tau) = U_2(a\tau)U_2(a\tau)U_2((1 - 4a)\tau)U_2(a\tau)U_2(a\tau), \tag{7}$$

where $a = 1/(4 - 4^{1/3})$.

In practice an efficient implementation of the first-order scheme is all that is needed to construct the higher-order algorithms Eqs.(6) and (7). The crucial step of this approach is to choose the H_i's such that the matrix exponentials $\exp(\tau H_1)$, ..., $\exp(\tau H_p)$ can be calculated efficiently. This will turn the formal expressions for $U_2(\tau)$ and $U_4(\tau)$ into efficient algorithms to solve the time-dependent Maxwell equations.

2.2. One-step algorithm

The basic idea of this approach is to make use of extremely accurate polynomial approximations to the matrix exponential. We begin by "normalizing" the matrix H. The eigenvalues of the skew-symmetric matrix H are pure imaginary numbers. In practice H is sparse so it is easy to compute $\|H\|_1 \equiv \max_j \sum_i |H_{i,j}|$. Then, by construction, the eigenvalues of $B \equiv -iH/\|H\|_1$ all lie in the interval $[-1, 1]$ [15]. Expanding the initial value $\mathbf{\Psi}(0)$ in the (unknown) eigenvectors $\mathbf{b}_j$ of B, Eq. (3) reads

$$\mathbf{\Psi}(t) = e^{izB}\mathbf{\Psi}(0) = \sum_j e^{izb_j}\mathbf{b}_j\langle\mathbf{b}_j|\mathbf{\Psi}(0)\rangle, \tag{8}$$

where $z = t\|H\|_1$ and the b_j denote the (unknown) eigenvalues of B. There is no need to know the eigenvalues and eigenvectors of B explicitly. We find the Chebyshev polynomial expansion of $U(t)$ by computing the expansion coefficients of each of the functions e^{izb_j} that appear in Eq. (8). In particular, as $-1 \leq b_j \leq 1$, we can use the expansion [30] $e^{izb_j} = J_0(z) + 2\sum_{k=1}^{\infty} i^k J_k(z)T_k(b_j)$, where $J_k(z)$ is the Bessel function of integer order k, to write Eq. (8) as

$$\mathbf{\Psi}(t) = \left[J_0(z)I + 2\sum_{k=1}^{\infty} J_k(z)\hat{T}_k(B)\right]\mathbf{\Psi}(0). \tag{9}$$

Here $\hat{T}_k(B) = i^k T_k(B)$ is a matrix-valued modified Chebyshev polynomial that is defined by $\hat{T}_0(B)\mathbf{\Psi}(0) = \mathbf{\Psi}(0)$, $\hat{T}_1(B)\mathbf{\Psi}(0) = iB\mathbf{\Psi}(0)$ and the recursion

$$\hat{T}_{k+1}(B)\mathbf{\Psi}(0) = 2iB\hat{T}_k(B)\mathbf{\Psi}(0) + \hat{T}_{k-1}(B)\mathbf{\Psi}(0), \tag{10}$$

for $k \geq 1$. As $\|\hat{T}_k(B)\| \leq 1$ by construction and $|J_k(z)| \leq |z|^k/2^k k!$ for z real [30], the resulting error vanishes exponentially fast for sufficiently large K. Thus, we can obtain an accurate approximation by summing contributions in Eq. (9) with $k \leq K$ only. The number K is fixed by requiring that $|J_k(z)| < \kappa$ for all $k > K$. Here, κ is a control parameter that determines the accuracy of the approximation. For fixed κ, K increases linearly with $z = t\|H\|_1$ (there is no requirement on t being small). ¿From numerical analysis it is known that for fixed K, the Chebyshev polynomial is very nearly the same polynomial as the minimax polynomial [31], i.e. the polynomial of degree K that has the smallest maximum deviation from the true function, and is much more accurate than for instance a Taylor expansion of the same degree K. In practice, $K \approx z$.

In a strict sense, the one-step method does not yield an orthogonal approximation. However, for practical purposes it can be viewed as an extremely stable time-integration algorithm because it yields an approximation to the exact time evolution operator $U(t) = e^{tH}$ that is exact to nearly machine precision [24, 25]. This also implies that within the same precision $\nabla\cdot(\mu\mathbf{H}(t)) = \nabla\cdot(\mu\mathbf{H}(t=0))$ and $\nabla\cdot(\varepsilon\mathbf{E}(t)) = \nabla\cdot(\varepsilon\mathbf{E}(t=0))$ holds for all times, implying that the numerical scheme will not produce artificial charges during the time integration [2, 3].

3. Implementation

The basic steps in the construction of the product-formula and one-step algorithms are best illustrated by considering the simplest case, i.e. the Maxwell equations of a 1D homogeneous problem. From a conceptual point of view nothing is lost by doing this: the extension to 2D and 3D nonhomogeneous problems is straigthforward, albeit technically non-trivial [11, 12, 24, 25].

We consider a system, infinitely large in the y and z direction, for which $\varepsilon = 1$ and $\mu = 1$. Under these conditions, the Maxwell equations reduce to two independent sets of first-order differential equations [1], the transverse electric (TE) mode and the transverse magnetic (TM) mode [1]. As the equations of the TE- and TM-mode differ by a sign we can restrict our considerations to the TM-mode only. The magnetic field $H_y(x,t)$ and the electric field $E_z(x,t)$ of the TM-mode in the 1D cavity of length L are solutions of

$$\frac{\partial}{\partial t}H_y(x,t) = \frac{\partial}{\partial x}E_z(x,t) \quad , \quad \frac{\partial}{\partial t}E_z(x,t) = \frac{\partial}{\partial x}H_y(x,t), \tag{11}$$

subject to the boundary condition $E_z(0,t) = E_z(L,t) = 0$ [1]. Note that the divergence of both fields is trivially zero.

Following Yee [4], to discretize Eq.(11), it is convenient to assign H_y to odd and E_z to even numbered lattice sites. Using the second-order central-difference approximation to the first derivative with respect to x, we obtain

$$\frac{\partial}{\partial t}H_y(2i+1,t) = \delta^{-1}(E_z(2i+2,t) - E_z(2i,t)), \tag{12}$$

$$\frac{\partial}{\partial t}E_z(2i,t) = \delta^{-1}(H_y(2i+1,t) - H_y(2i-1,t)), \tag{13}$$

where we have introduced the notation $A(i,t) = A(x = i\delta/2, t)$. The integer i labels the grid points and δ denotes the distance between two next-nearest neighbors on the lattice (hence the absence of a factor two in the nominator). We define the n-dimensional vector $\boldsymbol{\Psi}(t)$ by

$$\boldsymbol{\Psi}(i,t) = \begin{cases} H_y(i,t), & i \text{ odd} \\ E_z(i,t), & i \text{ even} \end{cases}. \tag{14}$$

The vector $\boldsymbol{\Psi}(t)$ contains both the magnetic and the electric field on the lattice points $i = 1, \ldots, n$. The i-th element of $\boldsymbol{\Psi}(t)$ is given by the inner product $\boldsymbol{\Psi}(i,t) = \mathbf{e}_i^T \cdot \boldsymbol{\Psi}(t)$ where $\mathbf{e}_i$ denotes the i-th unit vector in the n-dimensional vector space. Using this notation (which proves most useful for the case of 2D and 3D for which it is rather cumbersome to write down explicit matrix representations), it is easy to show that Eqs.(12) and (13) can be written in the form (2) where the matrix H is given by

$$H = \begin{pmatrix} 0 & \delta^{-1} & & & \\ -\delta^{-1} & 0 & \delta^{-1} & & \\ & \ddots & \ddots & \ddots & \\ & & -\delta^{-1} & 0 & \delta^{-1} \\ & & & -\delta^{-1} & 0 \end{pmatrix} = \delta^{-1}\sum_{i=1}^{n-1}\left(\mathbf{e}_i\,\mathbf{e}_{i+1}^T - \mathbf{e}_{i+1}\mathbf{e}_i^T\right). \tag{15}$$

We immediately see that H is sparse and skew-symmetric by construction.

3.1. Yee-type algorithms

First we demonstrate that the Yee algorithm fits into the product-formula approach. For the 1D model (15) it is easy to see that one time-step with the Yee algorithm corresponds to the operation

$$U_1^{Yee}(\tau) = (I + \tau A)(I - \tau A^T) = e^{\tau A} e^{-\tau A^T}, \tag{16}$$

where

$$A = \delta^{-1} \sum_{i=2}^{n-1}{}' \left(\mathbf{e}_i \, \mathbf{e}_{i-1}^T - \mathbf{e}_i \, \mathbf{e}_{i+1}^T \right), \tag{17}$$

and we used the arrangements of H and E fields as defined by Eq.(14). We use the notation $\sum'$ to indicate that the stride of the summation index is two.

Note that since $A^2 = 0$ we have $e^{\tau A} = 1 + \tau A$ *exactly*. Therefore we recover the time-step operator of the Yee algorithm using the first-order product formula approximation to $e^{\tau H}$ and decomposing $H = A - A^T$. However, the Yee algorithm is second-order, not first order, accurate in time [2, 3]. This is due to the use of a staggered grid in time [2, 3]. To perform one time step with the Yee algorithm we need to know the values of $E_z(t)$ and $H_y(t + \tau/2)$, not $H_y(t)$. Another method has to supply the H_y-field at a time shifted by $\tau/2$.

Within the spirit of this approach, we can easily eliminate the staggered-in-time grid at virtually no extra computational cost or progamming effort (if a conventional Yee code is available) by using the second-order product formula

$$U_2^{Yee}(\tau) = e^{\tau A/2} e^{-\tau A^T} e^{\tau A/2} = (I + \tau A/2)(I - \tau A^T)(I + \tau A/2). \tag{18}$$

The effect of the last factor is to propagate the H_y-field by $\tau/2$. The middle factor propagates the E_z-field by τ. The first factor again propagates the H_y field by $\tau/2$. In this scheme all EM fields are to be taken at the same time. The algorithm defined by $U_2^{Yee}(\tau)$ is second-order accurate in time by construction [17]. Note that $e^{\tau A/2}$ is not orthogonal so nothing has been gained in terms of stability. Since $[U_2^{Yee}(\tau)]^m = e^{-\tau A/2}[U_1^{Yee}(\tau)]^m e^{+\tau A/2}$, we see that, compared to the original Yee algorithm, the extra computational work is proportional to $(1 + 2/m)$, hence negligible if the number of time steps m is large.

According to the general theory outlined in Sec.2, the expression

$$U_4^{Yee}(\tau) = U_2^{Yee}(a\tau)U_2^{Yee}(a\tau)U_2^{Yee}((1 - 4a)\tau)U_2^{Yee}(a\tau)U_2^{Yee}(a\tau), \tag{19}$$

defines a fourth-order accurate Yee-like scheme, the realization of which requires almost no effort once U_2^{Yee} has been implemented. It is easy to see that the above construction of the Yee-like algorithms holds for the much more complicated 2D, and 3D inhomogeneous case as well. Also note that the fourth-order Yee algorithm U_4^{Yee} does not require extra storage to hold field values at intermediate times.

3.2. Unconditionally stable algorithms

Guided by previous work on Schrödinger and diffusion problems [17], we split H into two parts

$$H_1 = \delta^{-1} \sideset{}{'}\sum_{i=1}^{n-1} \left(\mathbf{e}_i \ \mathbf{e}_{i+1}^T - \mathbf{e}_{i+1} \mathbf{e}_i^T \right), \tag{20}$$

$$H_2 = \delta^{-1} \sideset{}{'}\sum_{i=1}^{n-2} \left(\mathbf{e}_{i+1} \mathbf{e}_{i+2}^T - \mathbf{e}_{i+2} \mathbf{e}_{i+1}^T \right). \tag{21}$$

such that $H = H_1 + H_2$. In other words we divide the lattice into odd and even numbered cells. According to the general theory given above, the first-order algorithm is given by $\tilde{U}_1(\tau)$. Clearly both H_1 and H_2 are skew-symmetric block-diagonal matrices, containing one 1×1 matrix and $(n-1)/2$ real, 2×2 skew-symmetric matrices. As the matrix exponential of a block-diagonal matrix is equal to the block-diagonal matrix of the matrix exponentials of the individual blocks, the numerical calculation of $e^{\tau H_1}$ (or $e^{\tau H_2}$) reduces to the calculation of $(n-1)/2$ matrix exponentials of 2×2 matrices. Each of these matrix exponentials only operates on a pair of elements of $\boldsymbol{\Psi}(t)$ and leaves other elements intact. The indices of each of these pairs are given by the subscripts of $\mathbf{e}$ and $\mathbf{e}^T$. Using the $\tilde{U}_1(\tau)$ algorithm it is easy to construct the unconditionally stable, higher-order algorithms $\tilde{U}_2(\tau)$ and $\tilde{U}_4(\tau)$, see Eq.(6) and Eq.(7).

3.3. One-step algorithm

The one-step algorithm is based on the recursion Eq.(10). Thus, the explicit form Eq.(15) is all we need to implement the matrix-vector operation (i.e. $\boldsymbol{\Psi}' \leftarrow H\boldsymbol{\Psi}$) that enters Eq.(10). The coefficients $J_k(z)$ (and similar ones if a current source is present) should be calculated to high precision. Using the recursion relation of the Bessel functions, all K coefficients can be obtained with $\mathcal{O}(K)$ arithmetic operations [31], a neglible fraction of the total computational cost for solving the Maxwell equations.

Performing one time step amounts to repeatedly using recursion (10) to obtain $\hat{T}_k(B)\boldsymbol{\Psi}(0)$ for $k = 2, \ldots, K$, multiply the elements of this vector by the appropiate coefficients and add all contributions. This procedure requires storage for two vectors of the same length as $\boldsymbol{\Psi}(0)$ and some code to multiply such a vector by the sparse matrix H. The result of performing one time step yields the solution at time t, hence the name one-step algorithm. In contrast to what Eq.(10) might suggest, the algorithm does not require the use of complex arithmetic.

4. Numerical experiments

Except for the conventional Yee algorithm, all algorithms discussed in this paper operate on the vector of fields defined at the same time t. We use the one-step algorithm (with a time step $\tau/2$) to compute $E_z(\tau/2)$ and $H_y(\tau/2)$. Then we use $E_z(0)$ and $H_y(\tau/2)$ as the initial values for the Yee algorithm. In order to permit comparison of the final
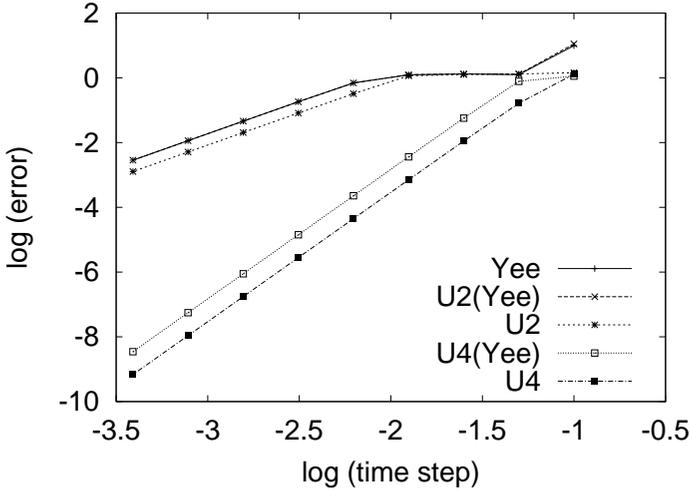
Figure 1. The error $\|\tilde{\boldsymbol{\Psi}}(t) - \hat{\boldsymbol{\Psi}}(t)\|/\|\hat{\boldsymbol{\Psi}}(t)\|$ at time $t = 100$ as a function of the time step τ for five different FDTD algorithms, plotted on a double logarithmic scale. The initial values of the EM fields are random, distributed uniformly over the interval [-1,1], on a grid of $n = 5001$ sites with $\delta = 0.1$ (corresponding to a physical length of 250.05). $\hat{\boldsymbol{\Psi}}(t)$ is the vector obtained by the one-step algorithm $\kappa = 10^{-9}$, using $K = 2080$ matrix-vector operations $\boldsymbol{\Psi}' \leftarrow M\boldsymbol{\Psi}$. The results of the Yee and U_2^{Yee} algorithm lie on top of each other. Lines are guides to the eye.

result of the conventional Yee algorithm with those of the other methods, we use the one-step algorithm once more to shift the time of the H_y field by $-\tau/2$. This procedure to prepare the initial and to analyse the final state of the Yee algorithm does in fact make the results of the Yee algorithm look a little more accurate than they would be if the exact data of the $\tau/2$-shifted fields were not available.

We define the error of the solution $\tilde{\boldsymbol{\Psi}}(t)$ for the wave form by $\|\tilde{\boldsymbol{\Psi}}(t) - \hat{\boldsymbol{\Psi}}(t)\|/\|\hat{\boldsymbol{\Psi}}(t)\|$ where $\hat{\boldsymbol{\Psi}}(t)$ is the vector of EM fields obtained by the one-step algorithm. Thereby we have already assumed that the one-step algorithm yields the exact (within numerical precision) results but this has to be demonstrated of course. A comparison of the results of an unconditionally stable algorithm, e.g. $\tilde{U}_4$ with those of the one-step algorithm is sufficient to show that within rounding errors the latter yields the exact answer. Using the triangle inequality $\|\boldsymbol{\Psi}(t) - \hat{\boldsymbol{\Psi}}(t)\| \leq \|\boldsymbol{\Psi}(t) - \tilde{\boldsymbol{\Psi}}(t)\| + \|\tilde{\boldsymbol{\Psi}}(t) - \hat{\boldsymbol{\Psi}}(t)\|$ and the rigorous bound $\|\boldsymbol{\Psi}(t) - \tilde{\boldsymbol{\Psi}}(t)\| \leq c_4\tau^4 t\|\boldsymbol{\Psi}(0)\|$ [17], we can be confident that the one-step algorithm yields the numerically exact answer if i) this rigorous bound is not violated and ii) if $\|\tilde{\boldsymbol{\Psi}}(t) - \hat{\boldsymbol{\Psi}}(t)\|$ vanishes like τ^4.

¿From the data in Fig.1 we conclude that the error of algorithm $\tilde{U}_4$ vanishes like τ^4, demonstrating that the one-step algorithm yields the numerically exact result. The high precision of the one-step algorithm also allows us to use it for genuine time stepping with

arbitrarily large time steps, this in spite of the fact that strictly speaking, the one-step algorithm is not unconditionally stable.

If the initial EM field distribution is random then, for sufficiently small τ, algorithm $\widetilde{U}_2$ is more accurate than the two second-order accurate Yee algorithms, as is clear from Fig.1 [32]. However, this conclusion is biased by the choice of the model problem and does not generalize. For the largest τ-values used in Fig.1, the Yee and U_2^{Yee} algorithm are operating at the point of instability, signaled by the fact that the norm of $\mathbf{\Psi}(t)$ grows rapidly, resulting in errors that are very large. If the initial state is a Gaussian wave packet that is fairly broad, the Yee-type algorithms are much more accurate than the unconditionally stable algorithms employed in this paper (results not shown). The data of Fig.1 clearly show that for all algorithms, the expected behavior of the error as a function of τ is observed only if τ is small enough.

The answer to the question which of the algorithms is the most efficient one crucially depends on the accuracy that one finds acceptable. The Yee algorithm is no competition for $\widetilde{U}_4$ if one requires an error of less than 1% but then $\widetilde{U}_4$ is not nearly as efficient (by a factor of about 6) as the one-step algorithm. Increasing the dimensionality of the problem favors the one-step algorithm [24, 25]. These conclusions seem to be quite general and are in concert with numerical experiments on 1D, 2D and 3D systems [25]. A simple theoretical analysis of the τ dependence of the error shows that the one-step algorithm is more efficient than any other FDTD method if we are interested in the EM fields at a particular (large) time only [24, 25]. This may open possibilities to solve problems in computational electrodynamics that are currently intractable. The Yee-like algorithms do not conserve the energy of the EM fields and therefore they are less suited for the calculation of the eigenvalue distributions (density of states), a problem for which the $\widetilde{U}_4$ algorithm may be the most efficient of all the algorithms covered in the paper.

The main limitation of the one-step algorithm lies in its mathematical justification. The Chebyshev approach requires that H is diagonalizable and that its eigenvalues are real or pure imaginary. The effect of relaxing these conditions on the applicability of the Chebyshev approach is left for future research.

In this paper we have focused entirely on the accuracy of the time integration algorithms, using the most simple discretization of the spatial derivatives. For practical purposes, this is often not sufficient. In practice it is straightforward, though technically non-trivial, to treat more sophisticated discretization schemes [2, 12] by the methodology reviewed is this paper.

Acknowledgments

References

[1] M. Born and E. Wolf, *Principles of Optics*, (Pergamon, Oxford, 1964).

[2] A. Taflove and S.C. Hagness, *Computational Electrodynamics - The Finite-Difference Time-Domain Method*, (Artech House, Boston, 2000).

[3] K.S. Kunz and R.J. Luebbers, *Finite-Difference Time-Domain Method for Electromagnetics*, (CRC Press, 1993).

[4] K.S. Yee, IEEE Transactions on Antennas and Propagation **14**, 302 (1966).

[5] See http://www.fdtd.org

[6] F. Zheng, Z. Chen, and J. Zhang, IEEE Trans. Microwave Theory and Techniques **48**, 1550 (2000).

[7] T. Namiki, IEEE Trans. Microwave Theory and Techniques **48**, 1743 (2001).

[8] F. Zheng and Z. Chen, IEEE Trans. Microwave Theory and Techniques **49**, 1006 (2001).

[9] S.G. Garcia, T.W. Lee, S.C. Hagness, IEEE Trans. and Wireless Prop. Lett. **1**, 31 (2002).

[10] W. Harshawardhan, Q. Su, and R. Grobe, Phys. Rev. E **62**, 8705 (2000).

[11] J.S. Kole, M.T. Figge and H. De Raedt, Phys. Rev. E **64**, 066705 (2001).

[12] J.S. Kole, M.T. Figge and H. De Raedt, Phys. Rev. E **65**, 066705 (2002).

[13] O.P. Gandi, *Advances in Computational Electrodynamics - The Finite-Difference Time-Domain Method*, A. Taflove, Ed., (Artech House, Boston, 1998).

[14] B. Houshmand, T. Itoh, and M. Piket-May, *Advances in Computational Electrodynamics - The Finite-Difference Time-Domain Method*, A. Taflove, Ed., (Artech House, Boston, 1998).

[15] J.H. Wilkinson, *The Algebraic Eigenvalue Problem*, (Clarendon Press, Oxford, 1965).

[16] G.D. Smith, *Numerical solution of partial differential equations*, (Clarendon Press, Oxford, 1985).

[17] H. De Raedt, Comp. Phys. Rep. **7**, 1 (1987).

[18] H. Tal-Ezer, SIAM J. Numer. Anal. **23**, 11 (1986).

[19] H. Tal-Ezer and R. Kosloff, J. Chem. Phys. **81**, 3967 (1984).

[20] C. Leforestier, R.H. Bisseling, C. Cerjan, M.D. Feit, R. Friesner, A. Guldberg, A. Hammerich, G. Jolicard, W. Karrlein, H.-D. Meyer, N. Lipkin, O. Roncero, and R. Kosloff, J. Comp. Phys. **94**, 59 (1991).

[21] T. Iitaka, S. Nomura, H. Hirayama, X. Zhao, Y. Aoyagi, and T. Sugano, Phys. Rev. E **56**, 1222 (1997).

[22] R.N. Silver and H. Röder, Phys. Rev. E **56**, 4822 (1997).

[23] Y.L. Loh, S.N. Taraskin, and S.R. Elliot, Phys. Rev. Lett. **84**, 2290 (2000); *ibid*. Phys.Rev.Lett. **84**, 5028 (2000).

[24] H. De Raedt, K. Michielsen, J.S. Kole, and M.T. Figge, in Computer Simulation Studies in Condensed-Matter Physics XV, eds. D.P. Landau et al., Springer Proceedings in Physics, (Springer, Berlin, in press).

[25] H. De Raedt, K. Michielsen, J.S. Kole, and M.T. Figge, submitted to IEEE Antennas and Propagation, LANL preprint: physics/0208060

[26] M. Suzuki, S. Miyashita, and A. Kuroda, Prog. Theor. Phys. **58**, 1377 (1977).

[27] H.F. Trotter, Proc. Am. Math. Soc. **10**, 545 (1959).

[28] H. De Raedt and B. De Raedt, Phys. Rev. A **28**, 3575 (1983).

[29] M. Suzuki, J. Math. Phys. **26**, 601 (1985); *ibid* **32** 400 (1991).

[30] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions*, (Dover, New York, 1964).

[31] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes*, (Cambridge, New York, 1986).

[32] This also explains why the unconditionally stable algorithms $\widetilde{U}_2$ and $\widetilde{U}_4$ yield more accurate eigenvalue distributions than the Yee algorithm [11].

Beam simulation tools for GEANT4 (and neutrino source applications)

V. Daniel Elvira† , Paul Lebrun† , and Panagiotis Spentzouris† §

† Fermilab, P.O. Box 500, Batavia, IL 60555, USA

Abstract. Geant4 is a tool kit developed by a collaboration of physicists and computer professionals in the High Energy Physics field for simulation of the passage of particles through matter. The motivation for the development of the Beam Tools is to extend the Geant4 applications to accelerator physics. Although there are many computer programs for beam physics simulations, Geant4 is ideal for modelling a beam going through material or a system with a beam line integrated into a complex detector. There are many examples in the current international High Energy Physics programs, such as studies related to a future Neutrino Factory, a Linear Collider, and a very Large Hadron Collider.

1. Introduction

Geant4 is a tool kit developed by a collaboration of physicists and computer professionals in the High Energy Physics (HEP) field for simulation of the passage of particles through matter. The motivation for the development of the Beam Tools is to extend the Geant4 applications to accelerator physics. The Beam Tools are a set of C++ classes designed to facilitate the simulation of accelerator elements such as r.f. cavities, magnets, absorbers. These elements are constructed from the standard Geant4 solid volumes such as boxes, tubes, trapezoids, or spheres.

A variety of visualization packages are available within the Geant4 framework to produce an image of the simulated apparatus. The pictures shown in this article were created with Open Inventor [3], which allows direct manipulation of the objects on the screen, with perspective rendering via the use of light.

Although there are many computer programs for beam physics simulations, Geant4 is ideal to model a beam passing through a material or to integrate a beam line with a complex detector. There are many such examples in the current international High Energy Physics programs.

§ Correspondence should be addressed to V. Daniel Elvira (daniel@fnal.gov)

2. A brief introduction to Geant4

Geant4 is the object oriented C++ version of the Geant3 tool kit for detector simulation developed at CERN. It is currently being used in many fields, such us HEP, space exploration, and medicine.

As a tool kit, Geant4 provides a set of libraries, a `main` function, and a family of initialization and action classes to be implemented by the user. These classes are singlets, and their associated objects are constructed in `main`. The objects contain the information related to the geometry of the apparatus, the fields, the beam, and actions taken by the user at different times during the simulation. The Geant4 library classes start with the `G4` prefix. The example described in this section, called `MuCool`, uses only some of the many available user classes.

2.1. Detector and field construction

The detector and field geometry, properties, and location are implemented in the constructor and methods of the `MuCoolConstruct` user class, which inherits from `G4VUserDetectorConstruction`. In the `Construct()` method the user does the initialization of the electromagnetic field and the equation of motion. There are a variety of Runge-Kutta steppers to select from, which perform the integration to different levels of accuracy. Next comes the detector description, which involves the construction of solid, logical, and physical volume objects. They contain information about the detector geometry, properties, and position, respectively. Many solid types, or shapes, are available. For example, cubic (box) or cylindric shapes (tube), are constructed as:

```
G4Box(const G4String& pName, G4double pX, G4double pY, G4double pZ);
G4Tubs(const G4String& pName, G4double pRMin, G4double pRMax,
       G4double pDz, G4double pSPhi, G4double pDPhi);
```

where a name and half side lengths are provided for the box. Inner, outer radii, half length, and azimuthal coverage are the arguments of a cylinder (tube). A logical volume is constructed from a pointer to a solid, and a given material:

```
G4LogicalVolume(G4VSolid* pSolid, G4Material* pMaterial,
                const G4String& name)
```

The physical volume, or placed version of the detector is constructed as:

```
G4PVPlacement(G4RotationMatrix *pRot, const G4ThreeVector &tlate,
              const G4String& pName, G4LogicalVolume *pLogical,
              G4VPhysicalVolume *pMother, G4bool pMany, G4int pCopyNo);
```

where the rotation and translation are performed with respect to the center of its "mother" volume (container). Pointers to the associated logical volume, and the copy number complete the list of arguments.

2.2. Physics processes

Geant4 allows the user to select among a variety of physics processes which may occur during the interaction of the incident particles with the material of the simulated apparatus. There are electromagnetic, hadronic and other interactions available like: "electromagnetic", "hadronic", "transportation", "decay", "optical", "photolepton_hadron", "parameterisation". The different types of particles and processes are created in the constructor and methods of the `MuCoolPhysicsList` user class, which inherits from `G4VUserPhysicsList`.

2.3. Incident particles

The user constructs incident particles, interaction verteces, or a beam by typing code in the constructor and methods of the `MuCoolPrimaryGeneratorAction` user class, which inherits from
`G4VUserPrimaryGeneratorAction`.

2.4. Stepping actions

The `MuCoolSteppingAction` user action class inherits from `G4UserSteppingAction`. It allows the performance of actions at the end of each step during the integration of the equation of motion. Actions may include killing a particle under certain conditions, retrieving information for diagnostics, and others.

2.5. Tracking actions

The `MuCoolTrackingAction` user action class inherits from `G4UserTrackingAction`. For example, particles may be killed here based on their dynamic or kinematic properties.

2.6. Event actions

The `MuCoolEventAction` user action class inherits from `G4UserEventAction`. It includes actions performed at the beginning or the end of an event; that is, immediately before or after a particle is processed through the simulated apparatus.

3. Description of the beam tools classes

This Section is devoted to explain how to simulate accelerator elements using the Beam Tools. Brief descriptions of each class and constructor are included.

3.1. Solenoids

The Beam Tools provide a set of classes to simulate realistic solenoids. These are `BTSheet`, `BTSolenoid`, `BTSolenoidLogicVol` and `BTSolenoidPhysVol`.

- The `BTSheet` class inherits from `G4MagneticField`. The class objects are field maps produced by an infinitesimally thin solenoidal current sheet. The class data members are all the parameters necessary to generate analytically a magnetic field in r-z space (there is φ symmetry). No geometric volumes or materials are associated with the `BTSheet` objects. `GetFieldValue` is a concrete method of `BTSheet` inherited from `G4Field`, through `G4MagneticField`. It returns the field value at a given point in space and time.

- The `BTSolenoid` class inherits from `G4MagneticField`. The class objects are field maps in the form of a grid in r-z space, which are generated by a set of `BTSheet`. The sheets and the `BTSpline1D` objects, containing the spline fits of B_z and B_r versus z for each r in the field grid, are data members of `BTSolenoid`. No geometric volumes or materials are associated with `BTSolenoid`. The field at a point in space and time is accessed through a `GetFieldValue` method, which performs a linear interpolation in r of the spline fit objects.

- The `BTSolenoidLogicVol` class defines the material and physical size of the coil system which is represented by the set of current sheets. A `BTSolenoid` must first be constructed from a list of current `BTSheets`. The `BTSolenoid` object is a data member of `BTSolenoidLogicVol`. The `BTSolenoidLogicVol` class constructor creates `G4Tubs` solid volumes and associated logical volumes for the coil system, the shielding, and the empty cylindric regions inside them. Only the logical volumes are constructed here. No physical placement of a magnet object is done.

- The `BTSolenoidPhysVol` class is the placed version of the `BTSolenoidLogicVol`. It contains the associated `BTSolenoid` object as a data member, as well as the pointers to the physical volumes of its logical constituents.

Figure 1 shows a group of four solenoidal copper coil systems modeled with four infinitesimally thin sheets equally spaced in radius.
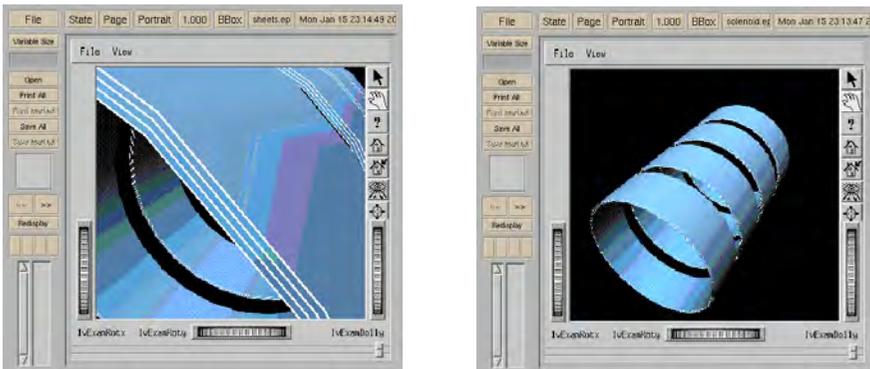
Figure 1. Left: a solenoidal copper coil system modeled with four infinitesimally thin sheets equally spaced in radius. Right: array of four solenoids separated by gaps.

3.2. Magnetic field maps

The Beam Tools also allow to simulate generic field maps using the `BTMagFieldMap` and `BTMagFieldMapPlacement` classes.

- `BTMagFieldMap` class inherits from `G4MagneticField`. The constructor reads the map information from an `ASCII` file containing the value of the field at a set of nodes of a grid. No geometric objects are associated with the field. The field at a point in space and time is accessed through a `GetFieldValue` method, as in the case of the solenoid.

- The `BTMagFieldMapPlacement` class is a placed `BTMagFieldMap` object. Only the field is placed because there is no coil or support system associated with it.

3.3. r.f. systems: pill box cavities and field maps

This section explains how to simulate realistic r.f. systems using Pill Box cavities. The Beam Tools package provides the classes: `BTAccelDevice`, `BTPillBox`, `BTrfCavityLogicVol`, `BTrfWindowLogicVol`, and `BTLinacPhysVol`.

- `BTAccelDevice.hh` class is abstract. All accelerator device classes are derived from this class, which inherits from `G4ElectroMagnetField`.

- The `BTPillBox` class inherits from `BTAccelDevice` and represents single $\pi/2$ Pill Box field objects. No solid is associated with `BTPillBox`. The time dependent electric field is computed using a simple Bessel function. It is accessed through a `GetFieldValue` method. The field is given by:

$$E_z = V_p \, J_0 \left(\frac{2\pi\nu}{c} r \right) sin(\phi_s + 2\pi\nu t) \tag{1}$$

$$B_\varphi = \frac{V_p}{c} J_1 \left(\frac{2\pi\nu}{c} r \right) cos(\phi_s + 2\pi\nu t) \tag{2}$$

where V_p is the cavity peak voltage, ν the wave frequency, ϕ_s the synchronous phase, and $J_{0,1}$ the Bessel functions evaluated at $\left(\frac{2\pi\nu}{c} r \right)$.

- The `BTrfMap` class also inherits from `BTAccelDevice`. The class objects are electromagnetic field maps which represent an r.f. cavity. In this way, complex r.f. fields can be measured or generated and later included in the simulation. The field map, in the form of a grid, is read in the `BTrfMap` constructor from an `ASCII` file. The `BTrfMap` object is a field, with no associated solid. A `GetFieldValue` method retrieves the field value at a point in space and time.

- The `BTrfCavityLogicVol` class constructor creates solid and logical volumes associated with the r.f. field classes. In the case of a map, a vacuum cylinder ring represents its limits. In addition to geometric and material parameters of the cavity, the class contains field and accelerator device information.

- The `BTrfWindowLogicVol` class is used with `BTCavityLogicVol` to create the geometry and logical volume of r.f. cavity windows, including the support structure, which may be placed to close the cavity iris at the end caps.

- The `BTLinacPhysVol` class is a placed linac object. A linac is a set of contiguous r.f. cavities, including the field, the support and conductor material, and windows. The `BTLinacPhysVol` constructor is overloaded. One version places a linac of Pill Box cavities and the other places field maps.

Fig. 2 shows a Pill Box cavity (in red) with windows. It also shows a cooling channel where solenoids are embedded in large low frequency cavities. Since the beam circulates inside the solenoid, the cavity is represented by a field map (in red) restricted to a cylindric volume with radius slightly smaller than the inner radii of the magnets.
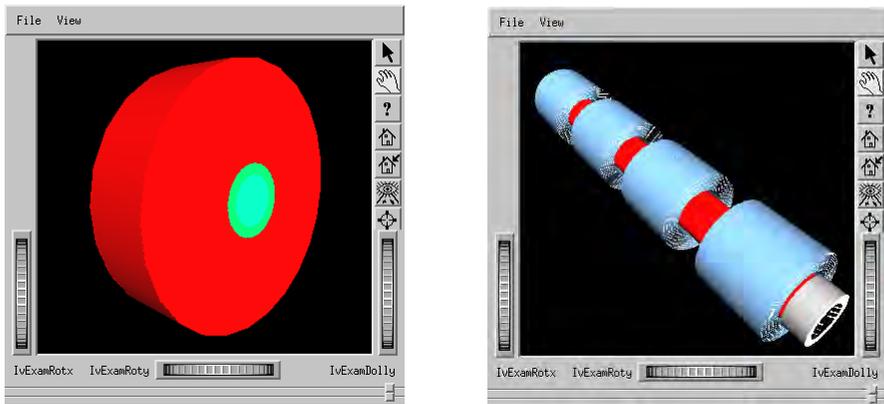
Figure 2. Left: a Pill Box cavity (in red), with windows (green). Right: low frequency cooling channel. The red cylinders are the dummy software structure representing the limits of the electric field maps.

3.4. Tuning the r.f. cavity phases

One of the critical elements of an accelerator simulation is the "r.f. tuning". Each cavity must be operated at the selected synchronous phase at an instant coincident with the passage of the beam. The r.f. wave must be therefore synchronized with the beam, more specifically, with the region of beam phase space that the user needs to manipulate. For this, there is the concept of a reference particle, defined as the particle with velocity equal to the phase velocity of the r.f. wave. If the kinematic and dynamic variables of the reference particle are set to values which are coincident with the mean values of the corresponding variables for the beam, the r.f. system should affect the mean beam properties in a similar way it affects the reference particle.

The Beam Tools allow the use of a "reference particle" to tune the r.f. system before processing the beam. The time instants the particle goes through the phase center of

each cavity are calculated and used to adjust each cavity phase to provide the proper kick, at the selected synchronous phase.

3.5. Absorbers

The Beam Tools provide a set of classes to simulate blocks of material in the path of the beam. The constructors create the solid, logical, and physical volumes in a single step. They are all derived from the abstract class of absorber objects `BTAbsObj`.

- `BTCylindricVessel` is a system with a central cylindric rim, and two end cup rims with thin windows of radius equal to the inner radius of the vessel. The material is the same for the vessel walls and windows, and the window thickness is constant. The vessel is filled with an absorber material.

- Two classes are available to simulate absorber lenses: `BTParabolicLense` and `BTCylindricLense`. The first one is a class of parabolic objects with uniform density, and the second a cylinder object with the density decreasing parabolically as a function of radius. From the point of view of the physics effect on the beam, both objects are almost equivalent. The `BTParabolicLense` is built as a set of short cylinders. The radius is maximum for the central cylinder and reduces symmetrically following a parabolic equation for the others in both sides. The `BTCylindricLense` object is built from concentric cylinder rings of the same length, different radius, and different densities to mimic a real lens.

The gray cylinder in Fig. 3 is a schematic representation of a liquid hydrogen vessel with aluminum walls and windows. Figure 3 also shows a set of six parabolic lenses in the center of a complex magnetic system. The lenses are placed to mitigate the effect of the decrease in $\langle p_z \rangle$ at large radii in a magnetic field flip region, using an emittance exchange mechanism.

Wedge absorbers are also useful in some cases. They can be easily constructed using the Geant4 trapezoid shape `G4Trap`.

4. Applications to neutrino factory feasibility studies

The neutrino beam in a Neutrino Factory would be the product of the decay of a low emittance muon beam. Muons would be the result of pion decay, and pions would be the product of the interaction of an intense proton beam with a carbon or mercury target. Thus the challenge in the design and construction of a Neutrino Source is the muon cooling section, aimed to reduce the transverse phase space by a factor of ten, to a transverse emittance of approximately $\varepsilon_x \sim 1$ cm.

The ionization cooling technique uses a combination of linacs and light absorbers to reduce the transverse emittance of the beam, while keeping the longitudinal motion under control. There are two competing terms contributing to the change of transverse
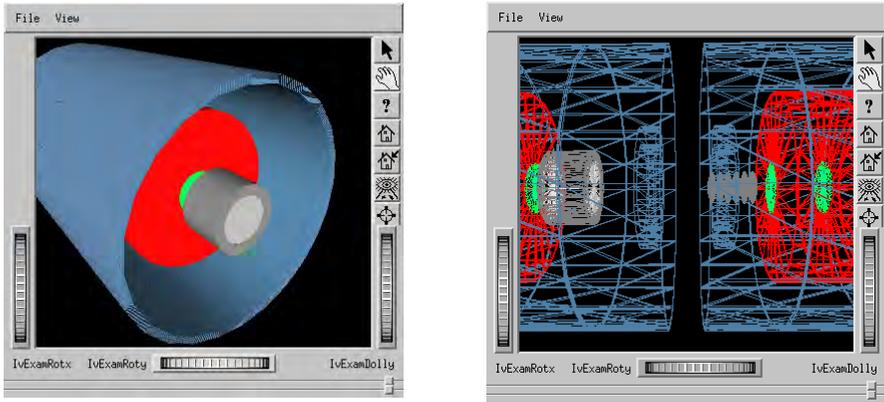
Figure 3. Left: cooling unit cell composed of a solenoid (blue), surrounding the r.f. system (red) and the cylindric absorber vessel (gray). Right: six parabolic lenses (gray) inside a complex magnetic field.

emittance ε_x along the channel. One is a cooling term, associated with the process of energy loss, and the other is a heating term related to multiple scattering.

4.1. The double flip cooling channel

The double flip cooling channel is a system consisting of three homogeneous solenoids with two field-flip sections. The first flip occurs at a relatively small magnetic field, B=3 T, to keep the longitudinal motion under control. The field is then increased adiabatically from -3 to -7 T, and a second field flip performed at B=7 T. Figure 4 shows a side view of a lattice unit cell, consisting of a six 201 MHz Pill Box cavities linac and one liquid hydrogen absorber, inside a solenoid. Details on the design and performance of this channel are available in Ref. [6].

4.2. The helical channel

The helical channel cools both in the transverse and longitudinal directions. The lattice is based on a long solenoid with the addition of a rotating transverse dipole field, lithium hydride wedge absorbers, and 201 MHz r.f. cavities. Figure 4 shows a side view of the helical channel, including the wedge absorbers, idealistic (thin) r.f. cavities, and the trajectory of the reference particle. The design details and performance of this channel are described in Ref. [7].

4.3. The low frequency channel

This is a design based on 44/88 MHz r.f. technology. A unit cell is composed of four solenoids embedded in four r.f. cavities, followed by a liquid hydrogen absorber. Figure 2 shows a unit cell of the low frequency channel, including the solenoids, the absorber,

and the relevant section of the r.f. field map (inside the magnets). More information about this channel may be found in Ref. [5].

4.4. Other systems

Among other simulations performed with the Beam Tools for Geant4 we may cite: the *Alternate Solenoid Channel (sFoFo)* [8], and a *High Frequency Buncher/Phase Rotator* scheme for the neutrino factory [9, 10].
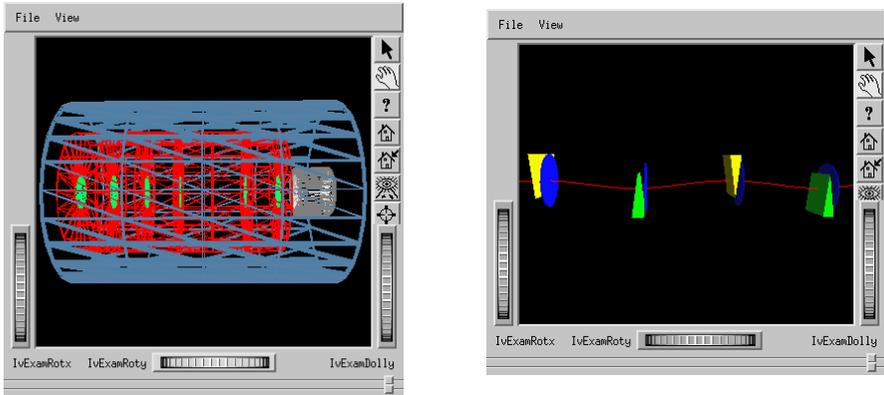
Figure 4. Left: side view of the double flip channel unit cell, including the solenoid, the six Pill Box cavities, and the absorber. Right: image of the helical channel, including the wedge absorbers (yellow and green), idealistic thin r.f. cavities (blue), and the trajectory of the reference particle (red).

5. Summary

The Beam Physics Tools for Geant4 are used in numerous accelerator studies, reported in conference proceedings and proposals. Geant4 is especially suited to systems where accelerators, shielding, and detectors must be studied jointly with a simulation. The Beam Tool libraries, a software reference manual, and a user's guide, are available from the Fermilab Geant4 web page [11].

Acknowledgments

References

[1] See Geant4 home page at: wwwinfo.cern.ch/asd/geant4/geant4.html.

[2] See Root home page at: http://root.cern.ch/root.

[3] Open Inventor. Registered trademark of Silicon Graphics Inc.

[4] See ZOOM home page at:
http://www.fnal.gov/docs/working-groups/fpcltf/Pkg/WebPages/zoom.html.

[5] "Pseudo-Realistic GEANT4 Simulation of a 44/88 MHz Cooling Channel for the Neutrino Factory", V. D. Elvira, H. Li, P. Spentzouris. MuCool note #230, 12/10/01.
http://www-mucool.fnal.gov/notes/noteSelMin.html

[6] "The Double Flip Cooling Channel", V. Balbekov, V. Daniel Elvira *et al.* Published in PAC2001 proceedings, Fermilab-Conf-01-181-T.

[7] "Simulation of a Helical Channel Using GEANT4", V. D. Elvira *et al.* Published in PAC2001 proceedings, Fermilab-Conf-01-182-T.

[8] "Feasibility Study 2 of a Muon Based Neutrino Source", S. Ozaki *et al.* BNL-52623, Jun 2001.

[9] "High Frequency Adiabatic Buncher ", V. Daniel Elvira, MuCool note #253, 8/30/02.
http://www-mucool.fnal.gov/notes/noteSelMin.html

[10] "Fixed Frequency Phase Rotator for the Neutrino Factory", N. Keuss and V. D. Elvira, MuCool note #254, 8/30/02.
http://www-mucool.fnal.gov/notes/noteSelMin.html

[11] See the Fermilab Geant4 web page at: http://www-cpd.fnal.gov/geant4.

Muon cooling rings for the ν factory and the $\mu^+\mu^-$ collider

**Yasuo Fukui (UCLA), David B. Cline (UCLA),
Alpher A. Garren (UCLA), and Harold G. Kirk (BNL)**

Abstract. We designed two storage ring models for 6 dimensional phase space cooling of intense muon beam for the neutrino factory and the muon collider, by using the SYNCH code. We demonstrate the 6 dimensional muon phase space cooling with the first 8 cell cooling ring model by using a tracking simulation code, ICOOL, with distributed Lithium lenses with the β at 30 cm, wedge absorbers, and RF cavities, with the muon momentum at 500 MeV/c. Phase space cooling is done by the transverse ionization cooling, and the wedge absorbers contribute to the emittance exchange from the longitudinal phase space to the transverse phase space. The second muon cooling ring has a 1.25 m long Lithium lens with the β at 1 cm in a straight section of a race track ring. Tracking simulation study is in progress to demonstrate the 6 dimensional phase space cooling of muon beam.

1. Introduction

In order to reduce the 6 dimensional phase space of muons within their lifetime, the ionization cooling is considered to be one of the most promising methods, where both transverse and longitudinal momenta are reduced due to the energy loss in absorbers, and the only longitudinal components of the muon momenta are restored through the accelerating fields of RF cavities. The multiple Coulomb scattering contributes to heat the transverse phase space. And the normalized transverse equilibrium depends on material kinds and the transverse beta function where an absorber is located. Initially, we designed muon cooling rings with quadrupole magnets, RF cavities, wedge absorbers of liq. H_2. [1] In order to increase the acceptance and to reduce the circumference, we then designed muon cooling rings with zero-gradient dipole magnets with edge focusing, RF cavities, and liq. H_2 wedge absorbers. Work is still in progress in improving the cooling performance in muon cooling rings.

Lithium lens is an active focusing element with energy absorber function at the same time. With β at 1 cm with high current density Lithium lenses, the normalized transverse emittance can be at 100-200 mm · mrad, which is low enough for a $\mu^+\mu^-$ collider.

By using the muon cooling rings and repeat the phase space cooling multiple times, the cost of the muon cooling channel can be reduced for the Neutrino Factory designs, and by using Lithium Lens with small beta in a storage ring, the 6 dimensional phase space cooling can be achieved for the muon collider designs.

2. Phase space cooling in a muon cooling ring

Figure 1 shows a schematic diagram of a ν Factory and a $\mu^+\mu^-$ collider. Two stages of muon phase space cooling rings are used in the figure.

In the longitudinal phase space, a wedge absorber in a dispersive region can reduce the energy spread of muons, where the straggling of the dE/dx and the slope of the dE/dx as a function of the muon momenta contributes to heat the longitudinal phase space by widening the energy spread. The wedge absorbers perform the emittance exchange from the longitudinal

phase space to the transverse(horizontal) phase space, due to the change of the muon trajectories in a storage ring when relative muon momenta were changed through absorbers in dispersive regions. Table 1 lists a comparison of an electron damping ring and a muon cooling ring in elements of damping, excitation, and the partition numbers in the transverse phase space and in the longitudinal phase space. Partition numbers in the Robinson's theorem are also listed. The muon cooling ring with wedge absorbers is similar to the well known electron damping rings in the damping and excitation terms.

Figure 1 shows a schematic diagram of a ν Factory and a $\mu^+\mu^-$ collider. Two stages of muon phase space cooling rings are used in the figure.
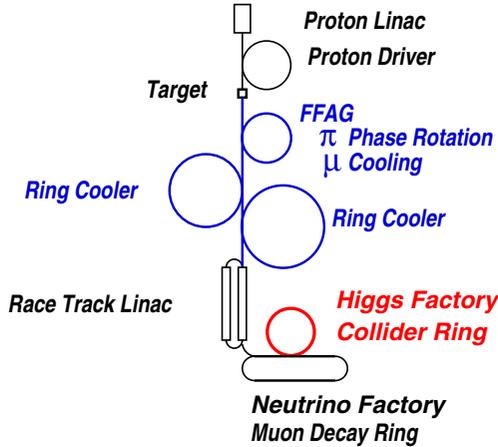
Figure 1. A schematic diagram of a ν Factory and a $\mu^+\mu^-$ collider

We designed an eight cell ring, with eight 45 degree bending cells, with a storage ring design code, SYNCH [3], by using two sets of zero-gradient dipole magnets with edge focusing. Figure 2 shows a schematic diagram of the eight cell muon cooling ring, and a blow-up of a half of a 45 degree bending lattice. A circumference is 28.8 m and a radius is 4.6 m. 2.5 cm long liq.H_2 wedge absorber and 7 cm long Lithium lens with field gradient of 3.3 Tesla/m, a set of two 0 gradient dipole magnets with bending angles of 44 degree and -22 degree with edge focusing, and a half of a 201 MHz RF cavity is shown in a half of a 45 degree cell. The cell length is 3.6 m. Figure 3 shows β_x, β_y and η in a 45 degree bending cell in the SYNCH modeling. The maximum β_x is 1.8 m, and the maximum β_x is 2.7 m at the outside dipoles. At the center of the 45 degree cell where the liq.H_2 wedge absorbers and Lithium lenses are placed, the minimum β_x, β_y are 30 cm each, and the maximum η, the dispersion, is 38 cm.

SYNCH is a linear matrix program to design a storage ring. It does not have any of the following: acceleration through RF cavities, dE/dx energy loss and straggling, multiple Coulomb scattering in the absorbers or in the Lithium lenses, and the effect of nonlinear fields on particle tracking. We use the fitted values of the ring parameters of SYNCH as input parameters of the ICOOL [4], a tracking code with nonlinear field configurations.

Table 2 lists parameters of the eight cell muon cooling ring. Figure 4 shows evolution of the normalized transverse emittances, $\epsilon_{nx}, \epsilon_{ny}$, and the normalized longitudinal emittance, ϵ_{nz} and a merit factor as a function of the path length along the central trajectory. Here, x is the horizontal coordinate and its positive direction goes outside of the muon cooling ring, y is the vertical coordinate, and the z goes along the central trajectory with beam. With the minimum β at 30 cm at the wedge absorbers and the Lithium lenses, the expected normalized vertical
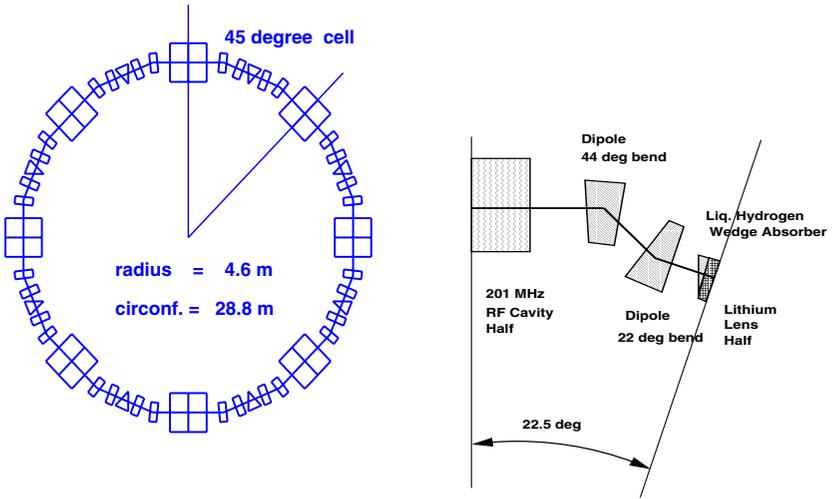
Figure 2. A top view of a muon cooling ring and a schematic diagram of a half of a 45 degree bending cell
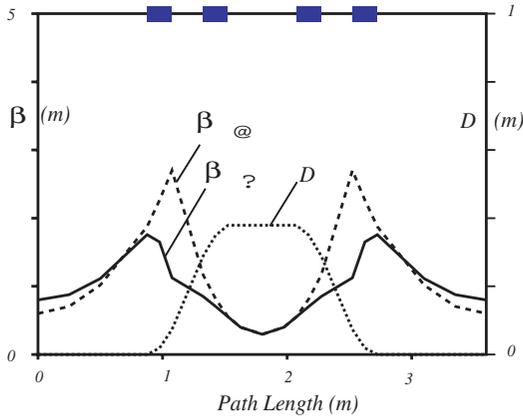
Figure 3. β_x, β_y and η as a function of z in a 45 degree bending cell

equilibrium emittance $\epsilon_{ny,equi}$ is 1.9 mm·rad, and we obtained 2.3 mm·rad in the ICOOL simulation, which is close enough to the expected number.

A merit factor is defined as a ratio of the initial normalized 6 dimensional emittance to the final normalized 6 dimensional emittance, multiplied by the muon transmission, without including the muon decay factor. With the average muon momentum at 500 MeV/c, the average survival factor of muons due to decay at z at 1200 m is 69 %. The merit factor in this simulation is around 10 at z at 1200 m, which corresponds to 42 turns.
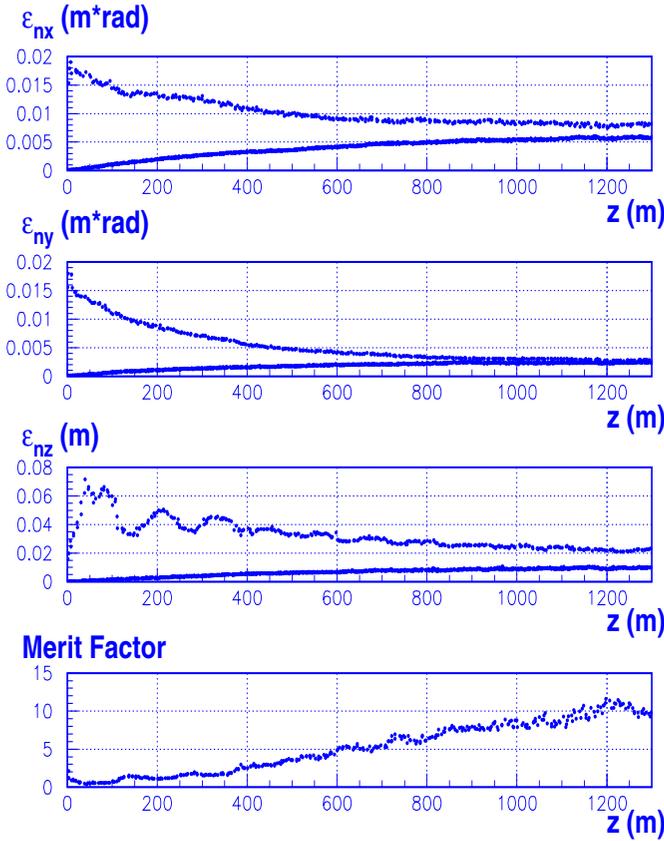
Figure 4. x, y, z normalized emittances and a merit factor as a function of z in the ICOOL tracking simulation on the 8 cell muon cooling ring

3. A muon cooling ring with a Lithium lens with β at 1 cm

We designed a race track ring model where a 1.08 m Lithium lens with β at 1 cm, sandwiched by two 9 cm long matching Lithium lens with β at 5 cm, is installed in a straight section. Table 2 lists parameters of the race track muon cooling ring. Figure 5 shows the schematic diagram of the race track ring. The circumference is 64.8 m, the straight sections are 18.0 m each. Figure 6 shows β_x, β_y inside two matching Lithium lenses and a central Lithium lens. Matching of the β function through a matching Lithium lens is done by:

$$\beta_{match} = \sqrt{\beta_{in} \cdot \beta_{out}}, \qquad \lambda_{osci}/2 = \pi/2 \cdot \beta_{match}$$

where $\beta_{match}, \beta_{in}, \beta_{out}$, and λ_{osci} are an equilibrium beta function of the matching Lithium lens, beta functions outside the matching Lithium lens, and the wave length of the beta oscillation in the matching lens.

The set of Lithium lenses is connected to a special matching lattice which is shown in Figure 7. Maximum β_x and β_y are 10.0 m and 12.9 m, respectively, in a quadrupole magnet

close to the Lithium lens set. Figure 8 shows β_x and β_y in four 4.50 m long straight cells on the other side of the straight section. No Lithium lens is installed here. Figure 9 shows β_x, β_y, and η in the whole race track muon cooling ring, including the Lithium lens insert.
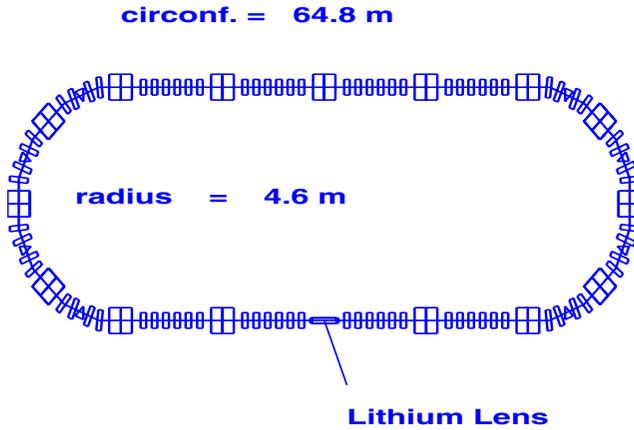
Figure 5. A schematic diagram of a race track muon cooling ring with a Lithium lens with the β at 1 cm
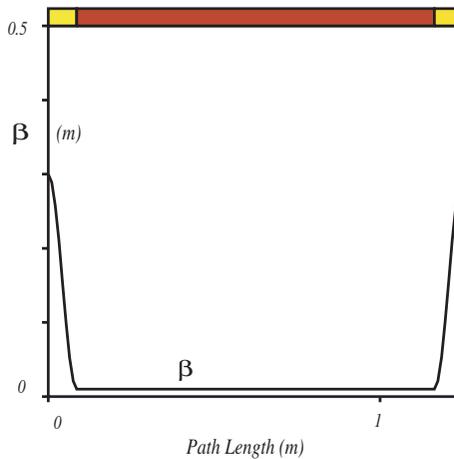
Figure 6. β_x, β_y as a function of z in two matching Lithium lenses and in a central Lithium lens where the minimum β is 1 cm.
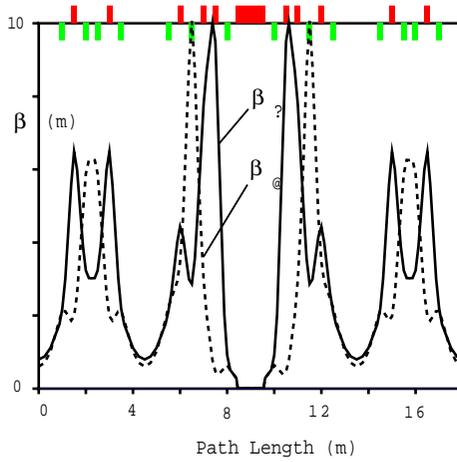
Figure 7. β_x, β_y as a function of z in a straight section with a Lithium lens at the β at 1 cm
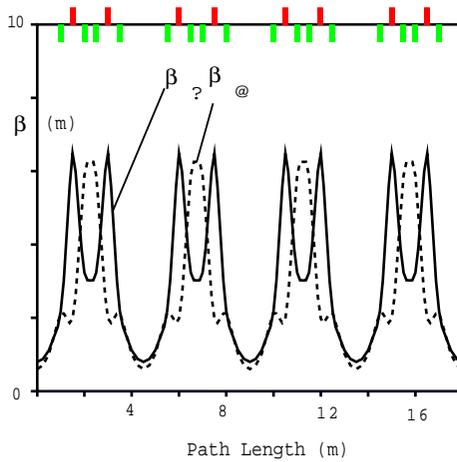
Figure 8. β_x, β_y as a function of z in a straight section without a Lithium lens

4. Conclusion

We designed an eight-cell muon cooling ring using SYNCH, a storage ring design code, with zero-gradient dipole magnets with edge focusing. We demonstrated the 6 dimensional muon phase space cooling in the eight cell ring with a tracking simulation code, ICOOL, with liq. H_2 wedge absorbers, Lithium lenses with β at 30 cm, RF cavities to compensate the z component of the muon momentum.

We designed a race track muon cooling ring with 1 m long Lithium lens with β at 1 cm, with the SYNCH code. Study is in progress to obtain the 6 dimensional muon cooling in this cooling ring.
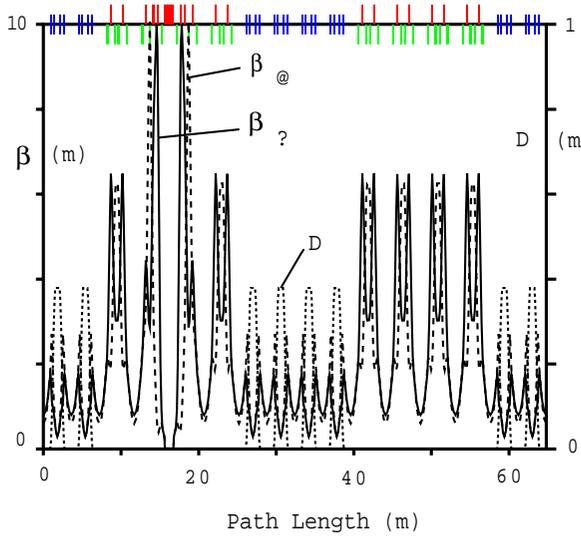
Figure 9. β_x, β_y and η as a function of z in a race track ring

Table 1. Comparison of an electron damping ring and the Muon Cooling ring

		e **Damping Ring**	
phase space	x	y	z
Damping	x' synch.rad. +RF	y' synch.rad. +RF	synch.rad. $\Delta E \propto E^4$
Excitation	x-x' orbit change		quantum fluct. $\propto E^{3.5}$
Partition #	$(1 - \mathcal{D})$	1	$2 + \mathcal{D}$

	μ **Cooling Ring with Wedge Absorbers**		
phase space	x	y	z
Damping	x' Ion.Cooling	y' Ion.Cooling	$\Delta E \propto E$ in Wedge
Excitation	x-x' orbit change mult.scat.	mult.scat.	$\frac{dE}{dx}$ straggling $\propto E^2$
Partition #	2-d	2	d

Acknowledgments

The authors wish to thank R. B. Palmer, R. Fernow, A Sessler, M. Zisman, J. Wurtele, D. Neuffer, and V. Balbekov, for useful discussions, and their help and encouragement.

Table 2. Parameters of an 8 cell muon cooling ring and a racetrack muon cooling ring

	8 Cell Ring	Racetrack Ring	
muon momentum	500	500	MeV/c
Circumference	28.8	64.8	m
straight section length		36.0	m
Structure of half cell	2 dipoles with edges	2 dipoles with edges	
Number of bending cells	8	8	
Number of straight cells	0	8	
bend cell length	3.6	3.6	m
straight cell length		4.5	m
max. β bending cell	1.8/2.7	1.8/2.7	m
min. β bending cell	0.30	0.30	m
max. η bending cell	0.38	0.38	m
max. β straight cell		10.0/12.9, 6.5/6.3	m
min. β straight cell		0.30	m
max. Quad grad		22.7	Tesla/m
length of Lithium lens	59	125	cm/turn
equilib. β in Lithium lens	0.30	0.01/0.05	m
dB_ϕ/dr of Lithium lens	0.034	167/5.5	Tesla/cm
length of liq. H_2 wedge absorber	0.40	0.40	m/turn
energy loss	57	115	MeV/turn
dipole length	0.2	0.2	m
dipole bend angles	44.2, -21.7	44.2, -21.7	degree
dipole edge angles	30/-3, -11/-11	30/-3, -11/-11	degree
dipole magnetic field	6.5, -3.2	6.5, -3.2	tesla
RF cavity length	1.0	1.0	m
number of RF cavities	8	15	
RF frequency	201	201	MHz
total drift space	21.4	44.8	m/turn
Cell tunes bend cell	0.72/0.70	0.72/0.70	
Cell tunes straight cell		0.39/0.47, 0.77/1.22	
Ring tunes	3.09/3.75	2.87/2.80	
chromaticities bend cell	-0.86/-0.69	-0.86/-0.69	
chromaticities straight cell		-2.75/-2.49, -9.29/0	
momentum compaction	-0.062	-0.062	
transition gamma	4.02	4.02	

References

[1] H. G. Kirk, D. B. Cline, Y. Fukui, A. A. Garren, *Progress Toward A Muon Ring Cooler*, SNOWMASS-2001-M101, Jun 2001. 5pp., Prepared for APS / DPF / DPB Summer Study on the Future of Particle Physics (Snowmass 2001), Snowmass, Colorado, 30 Jun - 21 Jul 2001.

[2] M. M. Alsharo'a, et. al., the NuMu Collaboration, *Status of Neutrino Factory and Muon Collider Reasearch and Developmemt and Future Plans*, FERMILAB-PUB-02-149-E, Jul 2002. 103pp, submitted to Phys. Rev. ST Accel. Beams

[3] A. A. Garren, A. S. Kenney, E. D. Courant, A. D. Russel, and M. J. Syphers. *SYNCH - A Compuetr System for Synchrotron Design and Orbit Analysis, User's Guide*, SSCL-MAN-0030, 1993

[4] R. Fernow, *A Simulation Code for Ionization Cooling of Muon Beams*, Part. Accel. Conf., Edts., A. Luccio and W. MacKay, Proc. 1999, p. 3020.

Status of the Los Alamos Accelerator Code Group

Robert W. Garnett

Los Alamos National Laboratory

Abstract. Since 1986, the Los Alamos Accelerator Code Group (LAACG) has received funding from the U.S. Department of Energy. The LAACG is a national resource for members of the accelerator community who use and/or develop software for the design and analysis of particle accelerators, and beam transport systems. We distribute accelerator design and simulation codes such as the POISSON/SUPERFISH group of codes, PARMELA, PARMILA, TRACE 3-D, and others via the world-wide web, and maintain an online compendium of accelerator codes at http://laacg1.lanl.gov/laacg/componl.html. We presently have over 1300 registered users world-wide using our codes and accessing our website. An update of recent LAACG activities and projects for the upcoming year will be discussed. This work is supported by the U. S. Department of Energy Office of Science, Division of High Energy Physics.

1. Brief history of the Los Alamos Accelerator Code Group

Since 1986, the Los Alamos Accelerator Code Group (LAACG) has received funding from the U.S. Department of Energy. The LAACG is a national resource for members of the accelerator community who use and/or develop software for the design and analysis of particle accelerators, and beam transport systems. We distribute accelerator design and simulation codes via the world-wide web and maintain an online compendium of accelerator codes at http://laacg1.lanl.gov/laacg/componl.html. Figure 1 shows the website layout for the code compendium. We presently have over 1300 registered users world-wide using our codes and accessing our website. The code group resides in the Accelerator Physics and Engineering Group, LANSCE-1, at Los Alamos National Laboratory. Initially, funding was received to maintain and document a standard version of POISSON/SUPERFISH, a collection of programs for calculating static magnetic and electric fields, and radio-frequency electromagnetic fields in either 2-D Cartesian coordinates or axially symmetric cylindrical coordinates. Designers of modern high-energy physics machines use POISSON/SUPERFISH to design magnets, accelerator cavities, and other beam-line components. Significant improvements in the accuracy and ease of use of the POISSON/SUPERFISH group of codes have resulted due to the past efforts of members of the LAACG. In the early 1990s the LAACG added support of other workhorse accelerator design codes: PARMELA, PARMILA and TRACE 3-D. At about the same time, due to rapidly increasing desktop computing capabilities, the code group began migration of these programs from older mainframe computers to PC desktop machines. Now all the supported codes run on PC desktop machines.

Figure 1. Code compendium website layout.

2. Impact of the LAACG

The codes maintained by the LAACG are currently being used in the design and simulation of accelerators and beam delivery systems presently funded through several divisions of the DOE Office of Science and other funding sources. Many of these projects are of national importance such as the Spallation Neutron Source (SNS) now being constructed at Oak Ridge National laboratory, at SLAC for LCLS and NLC, at BNL for RHIC, at universities such as UCLA, the University of Maryland, the University of Indiana, and Michigan State University, and several companies in industry including Advanced Energy Systems and AccSys Technology, Inc.

A review of major conference and workshop proceedings (e.g. the International Linac Conference, the Particle Accelerator Conference, the European Particle Accelerator Conference, the International Conference on the Application of Accelerators in Research and Industry, and others) shows that, in any given year, several hundred papers are published for which the research presented involved codes developed and maintained by the LAACG. Several of these publications also include comparisons between codes, indicating that similar alternate software is being developed at other institutions [1]. However, these codes are always benchmarked against the LAACG codes. Another trend that is visible is the development of software at other institutions that basically provides graphics interfaces and uses the LAACG codes as the calculation tools (or uses output from the LAACG codes as input for secondary calculations).

3. Present LAACG activities

Present LAACG activities include: (1) maintenance and enhancement of certain widely used software packages, including POISSON/SUPERFISH, PARMILA, PARMELA, PARMTEQ, and TRACE 3-D; (2) consultation, education, and gathering/dissemination of information related to computational accelerator physics; (3) the distribution of software and documentation via the Internet; and (4) maintenance of the official LAACG website. The continued high-level use of the above-mentioned codes, which have over 1300 registered users world-wide, demonstrates the great value of the LAACG to the national and international accelerator communities. Additionally, every year, the code group is pleased to grant the requests of instructors at universities and at special accelerator schools (USPAS, CERN) to use LAACG codes (POISSON/SUPERFISH, PARMELA, and TRACE 3-D) in the classroom as part of their graduate-level instruction in accelerator physics and engineering.

Many improvements to the codes have resulted through national and international collaborations with our users and through regular feedback regarding how the codes are serving the needs of the community. Future code enhancements such as including the capability to model low-energy superconducting accelerator structures could impact the design of facilities such as the Rare Isotope Accelerator (RIA) now being planned by collaboration between Argonne National Laboratory and Michigan State University, and the Accelerator Demonstration Test Facility (ADTF) being planned as part of the Advanced Accelerator Applications Program (AAA). Codes like PARMELA have already been used to enhance the performance of existing machines such as the Stanford Linear Collider [2]. Additionally, these codes may be required to help design the next- generation light source.

Recently, in order to reach some of our future goals, we have begun to collaborate with members of the Accelerator Controls and Automation Group (LANSCE-8) in our division and have added a member of that group to the LAACG. Members of this group will help us to eventually modernize our workhorse codes by allowing modularization of beam-physics calculations through object-oriented programming methods. Modularization will increase the application flexibility of these codes. Additionally, we are working towards standardizing the input format for our distributed codes. This will also increase the application flexibility of the codes.

4. The LAACG design codes

POISSON/SUPERFISH [3] – *is a 2-D code package of more than 30 programs for the design of RF cavities, magnet components, electrostatic elements, and waveguide devices*. An over-relaxation method is used to solve the generalized Poisson's equation in two dimensions. Eigenfrequencies and fields for arbitrarily shaped two-dimensional waveguides in Cartesian coordinates and three-dimensional axially symmetric RF cavities in cylindrical coordinates can be determined. The package contains codes to generate a very accurate triangular mesh adjusted to conform to the problem geometry, to plot the fields and to evaluate auxiliary quantities of interest in the design of drift-tube linac (DTL) cavities, coupled-cavity linac (CCL) cells, radio-frequency quadrupole (RFQ) cavities and other devices. For example, the code calculates transit-time factors, power losses, and the effect of perturbations. Several codes are included for automatically tuning DTL, CCL, and RFQ cavities by iterating on a selected portion of the geometry.

PARMILA [4] – *is a multi-particle design and transport code for ions historically used to design drift-tube linacs (DTLs).* The name comes from the phrase, "***Phase and***

Radial Motion in Ion Linear Accelerators". The code has been extended to also design coupled-cavity linacs, and elliptical-cavity superconducting linac structures. A "drift-kick" method is used to transform the beam, represented by a collection of particles, through the linac to study the beam dynamics performance of the design.

PARMELA [5] – *is a multi-particle beam dynamics code used primarily for electron-linac beam simulations.* The name comes from the phrase, "*Phase and Radial Motion in Electron Linear Accelerators.*" It is a versatile code that transforms the beam, represented by a collection of particles, through a user-specified linac and/or transport system. It includes several space-charge calculation methods. Particle trajectories are determined by numerical integration through the fields. This approach is particularly important for electrons where some of the approximations used by other codes (e.g. the "drift-kick" method commonly used for low-energy protons) would not hold. PARMELA works equally well for either electrons or ions although is computationally slower due to the numerical integrations. PARMELA can read field distributions generated by the POISSON/SUPERFISH group of codes. Members of the code group won a LANL 2000 Distinguished Copyright Award for this code.

PARMTEQ [6] – *and several other RFQ design codes comprise this group of codes and are used to design high-performance radio-frequency quadrupole (RFQ) linacs.* PARMTEQ is an acronym for "*Phase and Radial Motion in a Transverse Electric Quadrupole*". The codes have been experimentally verified in some detail by working hardware at Los Alamos and at other laboratories around the world. As we learn more about linac performance, both experimentally and theoretically, we continue to update these codes. Partial and complete RFQ design-code distributions are available. A partial distribution contains the codes necessary to design the RFQ vane profile and analyze the beam performance including the effects of higher order multipole field components and image charges. A complete distribution also includes the code VANES and several related programs, which generate and analyze machine instructions for numerically controlled machining of the vanes. Multi-particle simulations of the RFQ design are also possible with these codes.

TRACE 3-D [7] – *is an interactive first-order beam-dynamics program that calculates the envelopes of a bunched beam, including linear space-charge forces, through a user-defined transport system.* It provides an immediate graphics display of the envelopes and the phase-space ellipses in three dimensions. This code is extremely useful for laying out beam transport lines and for determining beam matching parameters.

5. FY 2002 highlights

The LAACG suite of PC codes (PARMELA, PARMILA, RFQ Codes, POISSON/SUPERFISH, and TRACE 3-D) continued to be supported and enhanced in FY 2002. A description of some of these activities follows. More detailed accounts can be found in the "Changes" files distributed with the codes, available on the LAACG fileservers [8]. Additionally, some updating of information and links in the online code compendium was completed.

POISSON/SUPERFISH – (highlights from more than 40 code modifications)
At user request, the POISSON/SUPERFISH codes now support an unlimited number of materials. Program VGAPLOT was retired and replaced by the Windows plotting program WSFPLOT, which includes several hard copy choices, arrow and circle plots to show field amplitude and direction, numbered axes, and other features. New utility programs update user preferences in an "INI file," and convert experimental

measurements of cavity frequencies between ambient conditions and vacuum at operating temperature. Modifications to point selection and harmonic polynomial functions improved the accuracy of the field interpolator. Several new features were added to the suite of RF-tuning programs. At the suggestion of a user at LBNL, we added new problem variables to the POISSON and PANDIRA harmonic analysis that allow off-axis placement of the analysis circle. We fixed some reported problems in the postprocessors SF7 and Force. All codes now use the robust programming practice that declares "IMPLICIT NONE" in every program unit, an extensive overhaul that allowed us to discover and work around a serious bug inherent to the Windows 98 and Windows ME operating systems.

PARMELA – (highlights from more than 20 code modifications)
Version 3, which was released last year, has proven robust enough so that we have discontinued support of version 2. The 3-D space-charge routine has new limits on the mesh aspect ratio while still allowing high-energy electrons to be treated correctly. Several beam-line elements have additional options, some at user request. Drifting particles whose velocity falls to zero are now treated properly in the presence of a static electric field. The code now writes multiple plot files for large problems (> 2 GB), eliminating a code crash. The plotting code, PARGRAF, includes new options, including color-coded particle densities, output of time-step emittance, and beam-size data. Input of Twiss parameters now uses the same convention as the programs PARMILA and TRACE 3-D.

PARMILA – A new 3-D space-charge algorithm has been added to more accurately simulate the performance of beams with high-aspect ratios. The code group benchmarked performance of PARMILA against four other codes as a part of an international collaboration on linac-design techniques. New features and plotting routines have also been added to help evaluate linac designs, and aid in the development and understanding of machine-commissioning techniques. PARMILA was used to design the SNS linac that is presently under construction and is being modified for use during its commissioning. PARMILA has become a large code. An effort to separate the accelerator design and beam dynamics functions of the code is presently underway with the goal of improving the ease of use of the code.

TRACE 3-D – A more robust data entry routine was added. This routine parses the namelist-like entries for syntax errors and reports the specific error, saving users considerable time debugging an input file.

IMPACT [9] – Many enhancements were made to the Impact code (an object-oriented three-dimensional parallel particle-in-cell code for studying high intensity beam dynamics) to increase its applicability to a wider range of problems and to improve performance. The code treats mean-field Coulomb interactions along with the dynamics of charged particles accelerator electromagnetic fields. The code includes two types of integrators, one based on a Hamiltonian approach and another based on the direct integration of the Lorentz-force equations. The code's 3-D parallel Poisson solvers treat six types of boundary conditions. The Lorentz-force based solver has been modified to have a time-step sub-cycle for the space-charge force calculation. This allows several steps, using different external fields, to use the same space-charge forces, which speeds up the Lorentz integrator. External field transformation functions needed for error studies using the Lorentz integrator with all six boundary conditions were added. The restart capability (portable to all high performance computers) was modified so that each processor in a multi-processor run will read from a file. The random sampling of an initial distribution was improved. A new beam-line element, EMfld, provides users the flexibility of using complex external-focusing and -

accelerating structures in the simulation. We used the code to study a "cross-plane resonance" phenomena discovered by Ingo Hofmann [10], which may lead to halo amplitudes significantly larger than previously expected.

POSTER – This graphics postprocessor is currently under development by C. Thomas Mottershead and is a new addition to the LAACG suite of codes. We anticipate making a version of this code available on the web soon. POSTER converts data files into Postscript form for plotting. It also includes several high-performance data processing/analysis capabilities. Its primary utility is to layout linacs, circular accelerators and beam lines that have been designed by the popular accelerator design codes. However, it has also been used to visualize results from multiparticle simulations and actual measured beam data. Figure 2 shows two examples of the sophisticated data processing and plotting capabilities of POSTER. Figure 2A shows a layout of the LANSCE Proton Storage Ring that is produced directly by reading a MARYLIE output file. All dimensions of the ring layout including the specific locations of each of the beam line elements are accurately portrayed to scale by the program. Figure 2B is a plot of the real-space transverse cross-section of a beam containing a halo. In this example the graded logarithmic color scale indicates halo beam fraction decades. This postprocessor will be extended to provide a seamless interface to PARMILA, PARMELA, TRACE 3-D, MARYLIE, DIMAD, TRANSPORT, and TEAPOT. POSTER will read the input files for these codes and generate accurate physical layouts of beam lines and accelerator lattices. An online version of a users manual will also be available. POSTER has been used rather extensively for beam-line layouts completed as part of the Advanced Hydrotest Facility Project, as well as others.

CODE COMPARISONS [11] – In an attempt to validate our linac-design codes we undertook a program to benchmark the performance of 5 codes: PARMELA, PARMILA, IMPACT, and two codes developed by K. Crandall at TechSource, LINAC and PARTRAN, against each other by simulation of the beam dynamics in the high-intensity SNS linac design as the basis of comparison. The comparison showed all 5 codes to be in close agreement. This exercise, involving an international collaboration, laid the initial groundwork for standardization of the I/O of all the codes to facilitate further collaborations and comparisons in the future.

6. Future plans

Present funding levels for the code group do not allow any new initiatives to be undertaken. However, it is anticipated that the future level of DOE support may increase. As a result, we are considering several new initiatives for the future. These are discussed below along with our ongoing activities.

6.1 Ongoing activities

Code Development and Distribution – The LAACG will continue to develop, maintain, and distribute the following codes: POISSON/SUPERFISH, PARMILA, PARMELA, PARMTEQ, and TRACE 3-D. A major area of emphasis will be to upgrade all source codes to the Fortran 90 standard. The POISSON/SUPERFISH suite of codes has already been upgraded to the Fortran 90 standard. A near-term goal will be to complete conversion of the PARMILA code, with conversion of all the codes in the next few years. Conversion of all the codes to this standard is a long-term goal of the LAACG
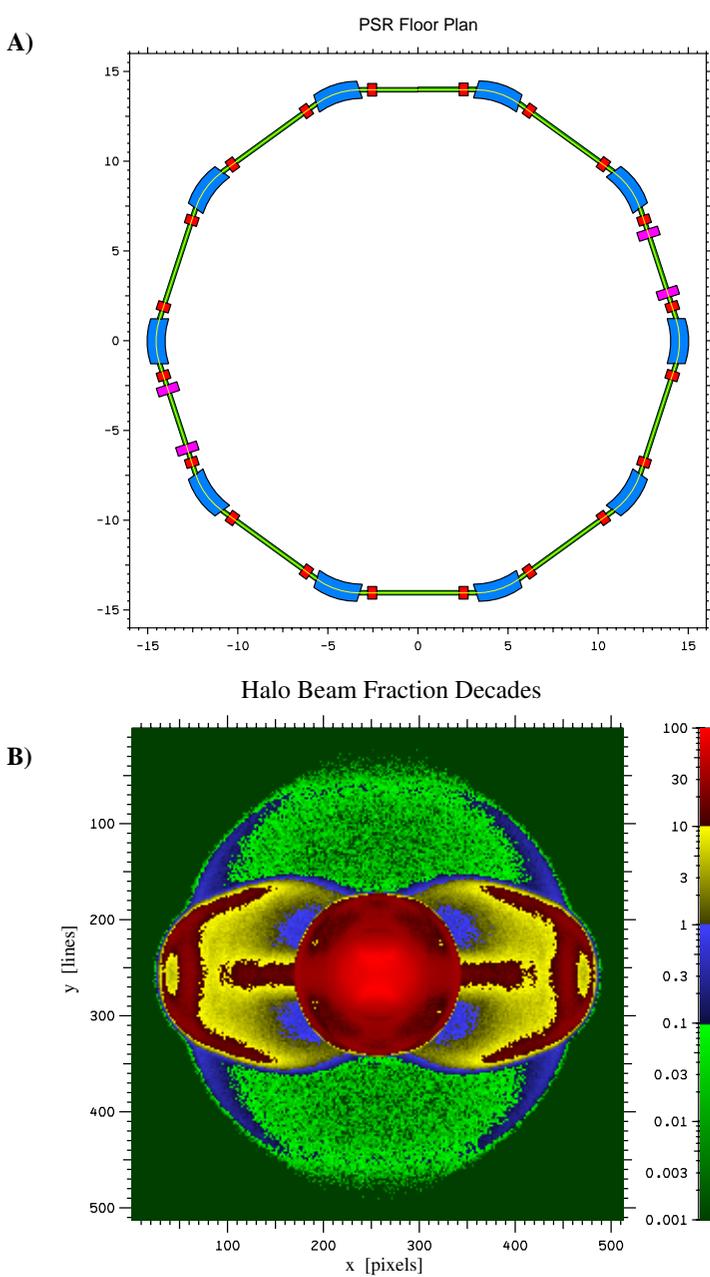
Figure 2. POSTER plotting examples: A) LANSCE Proton Storage Ring layout, B) Plot of the real-space transverse cross-section of a beam containing a halo where the graded logarithmic color scale indicates halo beam fraction decades.

and will facilitate the eventual release of Linux versions of all these codes, thereby increasing our user base significantly. Another major area of emphasis will be to improve the user interfaces of these codes. Other code development activities will include:

> POISSON/SUPERFISH – Windows versions of the general-purpose plotting codes will be released (A beta-test version has been very well received in the community.), the ability to use file editors while in a major application will be implemented, several obsolete features no longer needed on fast, modern computers will be eliminated, and the 600-page online users manual will be revised.

> PARMELA – The next major release will include 3-D field maps for RF and static fields, and coherent synchrotron radiation effects for electron beams in bends. PARMELA will likely be the first of the beam dynamics codes released as either a Linux or Windows version.

> TRACE 3-D – The next release will include new Windows dialog boxes with tabular entry of variables, replacing old-style cryptic text commands.

> RFQ CODES – Work is underway to bring the source codes for these programs up to modern standards. The next versions will use robust coding in Fortran 90 similar to that already implanted in POISSON/SUPERFISH. In addition, two important features will be added to the beam-dynamics simulation code PARMTEQ: a 3-D space charge routine, and implementation of the dynamic aperture effect.

Web Site Maintenance / Compendium of Accelerator Codes – The LAACG will continue to maintain the official web site and online compendium of accelerator codes. At present, the online compendium is incomplete with many pages not yet containing the required descriptive text or appropriate links. We expect to be able to correct all major compendium deficiencies during FY 2003, although improvement of the web site and updating the compendium is expected to be an ongoing activity of the LAACG. Acquisition of a new server running the Linux operating system is also a goal for FY 2003. Switching to a Linux-based server will allow improved capabilities such as online searching along with implementation of software not presently available under the MS-Windows environment to track our web site traffic.

6.2 New initiatives

External Advisory Panel – As a means of better setting priorities for the LAACG and to better serve our user community, we will establish an external advisory panel. A volunteer panel comprised of users from several other national laboratories and universities will be formed. One to two meetings per year via e-mail or phone conference will be used to address issues and set priorities to best utilize the LAACG funding.

Universal Input Format – Through recent interactions with many of the international users of our distributed codes, some consensus has been reached as to what a universal input file format might look like. The impetus for standardizing an input file format would be to allow ease of simulation of the same problem using various codes throughout the world. In particular, this would be extremely useful in

expediting code benchmarking. We expect to specifically define this format with the help of our users. Our final goal is to, as much as is possible, implement this input format with all of our distributed codes.

Superconducting (SC) Accelerating Structure Modeling – At present, modeling of SC elliptical accelerating cavities is included in the PARMILA code. These cavities have historically been used in electron accelerators (β=1) and will soon be used in the high-energy section of the Spallation Neutron Source (SNS) proton linac. However, there is increasing national and international interest in the use of low-β accelerating structures such as spoke resonators that could be used to eliminate the majority of normal-conducting accelerating structures in a proton or ion linac, for example, for systems like accelerator transmutation of waste, etc. LANSCE-1 presently has codes used for modeling the beam dynamics of these low-β accelerating structures. We plan to either incorporate the modeling of these cavities into the PARMILA code or to develop our present codes to a state that would eventually allow distribution through the LAACG. This work would be carried out as part of our present code-development activities. An additional future project will be the modeling of other types of quarter-wave and half-wave resonators that are now also becoming of interest for low-β SC applications. Interest in these cavities exists since they will be used for the proposed RIA Project.

Code Modernization – A long-term goal of the LAACG is to eventually move away from codes having a monolithic, procedural architecture. This is typical of most legacy codes, making them difficult to maintain, upgrade, and distribute. We believe that porting the simulation algorithms and numerical techniques to a modern, modular, and object-oriented architecture has the potential to significantly reduce the overhead associated with code maintenance. Additionally, we believe this approach will increase the range of applications and ease of use of the codes. A major new area of application for this modern modular programming approach is the implementation of sophisticated beam modeling algorithms as part of an accelerator control system. Major advances in control system sophistication related to beam tailoring and fault recovery/management will be required for operation of complex accelerator systems. Example applications for such a system include an energy-producing waste transmutation plant where tolerances to accelerator faults is very low and beam down-time must be reduced significantly below what is presently tolerated in research facilities, significant improvements in beam control required to reach luminosity goals in future colliders, and the very tight tolerances required for successful operation of the next 4th-5th generation light sources. As a first step, we plan to modernize/modularize the beam dynamics calculations for a drift-tube linac as an initial test case. Simulation results will be benchmarked against the PARMILA code. A specific part of the modularization process is documentation of the physics approximations and numerical methods used in the codes. This is particularly urgent since many of the key individuals having developed these codes over the last 10-15 years will retire within the next 2-5 years. We will begin this documentation process in FY 2003.

IMPACT Code Development – Past LAACG funds were used to support the development of advanced computing applications. We would like to continue to develop the IMPACT code in collaboration with R. Ryne and J. Qiang, the developers of this code and both now at Lawrence Berkeley National Laboratory, into another work-horse user-friendly code that can eventually be distributed through the LAACG. Specifically, we would work on making the code portable for use on both large multi-processor clusters and smaller desktop systems. Improvement of user interfaces and documentation would also be a goal.

3D Space-charge model for GPT simulations of high-brightness electron bunches

S.B. van der Geer, O.J. Luiten
Eindhoven University of Technology, The Netherlands

M.J. de Loos
Pulsar Physics, The Netherlands

G. Pöplau[*]**, U. van Rienen**
Rostock University, Germany

Abstract. For the simulation of high-brightness electron bunches, a new 3D space-charge model is being implemented in the General Particle Tracer (GPT) code. It is based on a non-equidistant multigrid solver, allowing smooth transitions from a high to a low-aspect ratio bunch during a single run. The algorithm scales linearly in CPU time with the number of particles and the insensitivity to aspect ratio ensures that it can be used for a variety of applications. Tracking examples and field comparisons with an analytical model will be shown.

1. Introduction

Applications such as SASE-FELs require high quality electron bunches with an emittance of the order of 1 micron and a peak current of 1 kA [1]. The standard method to create such bunches is to use a rf-photogun to produce relatively long bunches of several ps length, and magnetically compress them to the required current following acceleration.

A typical example of the bunch compression scheme is the first part of the DESY TTF, where a bunch of several ps is accelerated to 17 MeV and subsequently compressed [2]. This compression stage however causes Coherent Synchrotron Radiation (CSR) degrading the transverse emittance and hence reducing bunch brightness [3].

One of the alternatives to this route are acceleration techniques based on pulsed DC technology. Attainable fields of 1 GV/m can keep a bunch sufficiently short to reach the current of 1 kA without the need for downstream compression and thus avoiding degradation of bunch quality. An example of DC acceleration is the DC/RF scheme pioneered at Eindhoven University of Technology [4]. Here a 50 fs FWHM laser pulse is

---

used to create a 200 fs electron bunch at 10 MeV in a two-stage process. First the bunch is accelerated in a 1 GV/m field to 2 MeV. This is followed by a 2.5 cell S-band rf-booster to increase the energy to 10 MeV.

The dynamic behaviour of the space-charge fields for these approaches is very different. This can easily be seen by comparing the aspect ratio A in the rest-frame of the bunch, defined as $A=R/(\gamma L)$ with R the bunch radius, L the length and γ the Lorentz factor. As shown in Table 1, the compression scheme has an aspect ratio of near unity, for a 1 mm radius, at 1 MeV. At that energy, the 'TU/e' bunch still has an aspect ratio far larger than one, i.e. a 'pancake' geometry.

Table 1. Aspect ratio in the rest frame for various energies and bunch-lengths measured in the lab frame.

	0.1 MeV	1 MeV	10 MeV
100 fs	50.85	11.98	1.62
1 ps	5.09	1.20	0.16
10 ps	0.51	0.12	0.02

The remainder of this paper focuses on a new 3D space-charge model in the GPT code [5,6]. Section 2 shows space-charge fields for bunches with varying aspect ratios. Section 3 explains the algorithm of the new space-charge model, where the accuracy and scalability is shown in section 4. The first tracking tests are presented in section 5.

2. Space-charge fields

Different aspect ratios pose different challenges to the space-charge routines in simulation codes. To demonstrate this, **Figure 1** shows the transverse and longitudinal electric fields for various low aspect ratio (cigar) bunches. This is the typical regime for most electron accelerators. For very long bunches with $A<1/100$, the longitudinal fields become non-linear at the head and the tail of the bunch. In many simulations however, these effects can safely and conveniently be ignored because they only affect a relatively small part of the bunch and do not affect the overall results significantly.
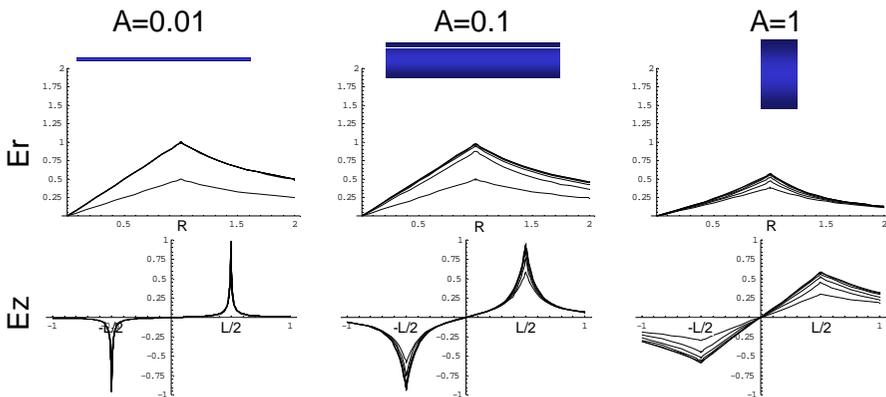
Figure 1. Transverse (top) and longitudinal (bottom) electrostatic fields in cylindrically symmetric low aspect ratio ('cigar') bunches. The different lines in each plot represent different positions in the bunch. All plots are created with uniform charge density.

High aspect ratio ('pancake') bunches have very different fields as shown in **Figure 2**. There the longitudinal fields are always relatively linear. However, when $A>100$, the transverse fields become non-linear near the edge of the bunch. This non-linearity is hard to tackle analytically and numerically because it is very ill described by a truncated Taylor expansion. In many simulations this difficult effect is ignored because bunches with an aspect ratio of 10 or more are quite uncommon in accelerators, see Table 1.

For the simulation of high brightness electron bunches, where every anomaly is important, the non-linear fields in both the high and the low aspect ratio bunches need to be modelled with great precision. Especially the dynamic behaviour of the short bunches is challenging because the aspect ratio varies significantly during the first stages of acceleration.
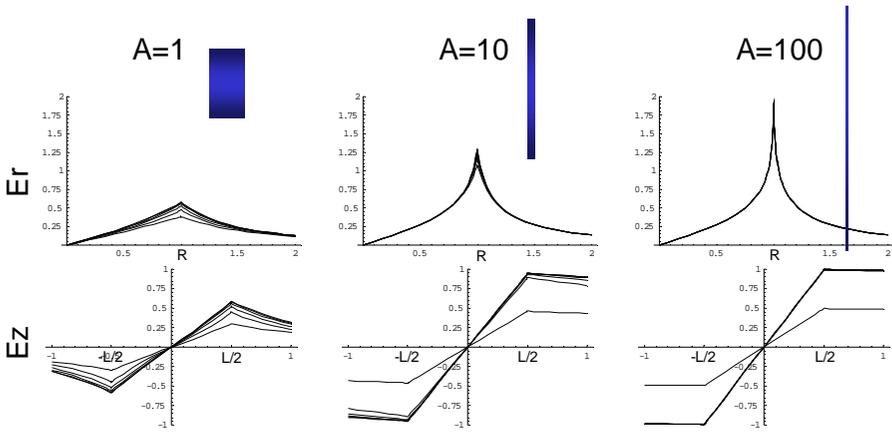
Figure 2. Transverse (top) and longitudinal (bottom) electrostatic fields in uniformly charged cylindrically symmetric high aspect ratio ('pancake') bunches. The different lines in each plot represent different positions in the bunch. All plots are created with uniformly distributed constant total charge.

3. The new space-charge model

3.1. *Requirements*

To be able to track high-brightness electron bunches through accelerators like the TU/e DC-RF scheme, or the DESY-TTF, the new space-charge model should be applicable to a wide parameter range in aspect ratio. At least aspect rations varying between 0.01 and 100 are required.

Furthermore, the model should be very accurate, since the design of demanding applications such as SASE-FELs relies on very accurate simulation tools. It is difficult to exactly state how accurate the model should be, but in any case the final simulation results for emittance, energy spread and bunch length must be accurate to a few percent or less.

Because not all external fields in an accelerator are cylindrically symmetric, the space-charge model has to be 3D. Furthermore, cathode impurities and laser anomalies often start a non-cylindrically symmetric beam on a photocathode. To simulate this process, again a 3D space-charge model is required.

Finally, the new space-charge routine should be fast and robust to make it useful as a design tool. The aim is to make it fast enough to allow scanning and optimisation of various parameters on a standard PC.

3.2. *Algorithm*

The chosen algorithm for the new space-charge model in GPT is a mesh-based Poisson solver in the rest-frame of the bunch. First the bunch is transformed to the frame with no average momentum. In this rest-frame, only electrostatic forces are present as long as the velocities in this frame are non-relativistic. The bunch-length in the rest-frame is longer by the Lorentz factor γ.

Subsequently, a Cartesian mesh is generated in a box around the bunch. The distribution of the meshlines is non-equidistant and adapted to the projected charge density. This is essential for the accuracy and speed of the routine. More information about the meshing is presented in section 3.3. The non-equidistant mesh is used to store an approximation of the charge density on the corners of the mesh. This information is obtained using a linear distribution of the charge of every particle over the eight corners of its enclosing meshbox.

The charge density on the mesh is fed into a Poisson solver to obtain the potential. A state-of-the-art multigrid Poisson solver [7,8] has been constructed for the non-equidistant meshes described in subsection 3.3. It scales linearly in CPU time with the number of meshnodes. Selectable Dirichlet or open boundary conditions allow the simulation of bunches within pipes with rectangular cross section, bunches near a cathode and a bunch in open space.

The resulting potential is interpolated and differentiated using a 2nd order interpolation scheme to obtain the electric field in the rest-frame of the bunch at all particle coordinates. This electric field is transformed back to the laboratory frame to obtain the electric and magnetic fields. The GPT kernel combines these fields with the external fields in the tracking engine.

The chosen algorithm implies that the velocity in the rest-frame of the bunch is non-relativistic. This assumption is not always true for long bunches in the first stages of acceleration. There, the situation is possible that new electrons are emitted with an energy in the eV level while the front of the bunch has already been accelerated to relativistic velocities. This is a known limitation of the routine reducing the applicability.

3.3. *Adaptive meshing*

To reduce the number of meshlines needed, an adaptive meshing technique is used. The purpose of this scheme is to both reduce 'wasted' CPU time in empty volume and reduce numerical errors by choosing more meshlines in regions with higher charge density.

To control the 'adaptiveness' of the meshing, a control parameter *fn* is introduced. This parameter defines the maximum difference in size between neighbouring meshnodes as

shown in **Figure 3**. When *fn*=0, the mesh-spacing is equidistant, resulting in the most stable setting for the Poisson solver. Increasing *fn* typically increases the overall efficiency because less nodes are required to obtain the same accuracy. A *fn*-value of 100%, allowing neighbouring mesh spacing to double in size, seems to be a good compromise for a relatively large parameter range. Further increasing *fn* typically leads to a regime where the Gauss-Seidel scheme does not always converge [9]. Obviously, different applications with different accuracy requirements and different charge density distributions have different optimal settings for *fn*.
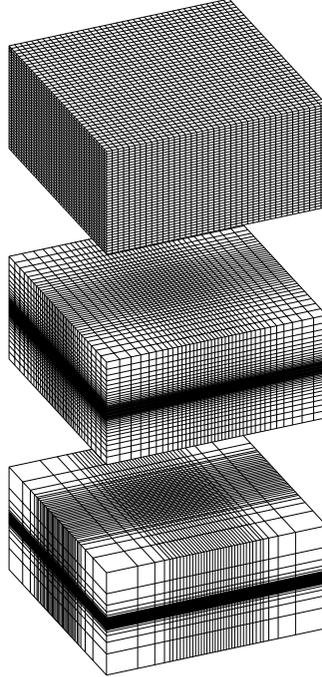
Figure 3. Different adaptive meshes for a uniformly charged hard-edge cylindrical bunch. The *fn*=0 (top) setting results in an equidistant mesh. When *fn*=20% (middle) more meshlines are chosen at the position of the bunch with smooth transitions into empty space resulting in very stable solver performance. The *fn*=100% case (bottom) results in very aggressive adaptive meshing, choosing almost all meshlines at the position of the bunch.

4. Analytical test-cases of a hard-edged cylinder

To test the accuracy of the new space-charge model, the field calculations have been compared to various analytical test cases [10]. In this section we describe the tests against analytical equations for a uniformly charged hard-edged cylindrically symmetric bunch as function of aspect ratio, number of particles and number of meshnodes.

As shown in **Figure 4**, the dimensionless field error E is below 10% for a range in aspect ratio varying between 0.01 and 100 for 100,000 meshnodes, where E is defined as:

$$E = \frac{\left| \mathbf{E}_{analytical} - \mathbf{E}_{simulation} \right|}{\left| \mathbf{E}_{analytical} \right|} \tag{1}$$

The aspect ratios 0.01 to 10 produce much better results with E-values in the range of a few percent starting from 20,000 meshnodes. Although an error of a few percent may sound quite high, it should be noted that these tests represent worst-case scenarios. In actual tracking simulations, as presented in section 5, all hard-edges become smooth in a few timesteps. This produces much higher average accuracy of the space-charge routine, allowing aspect ratios over 100 and increasing the practical applicability of the algorithm beyond the range plotted in **Figure 4**.
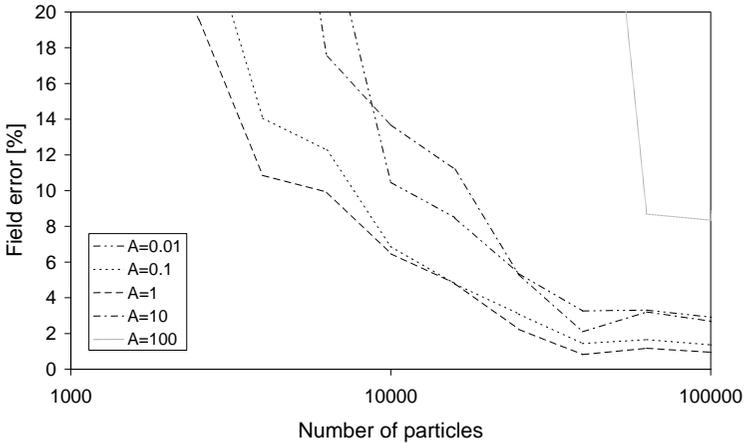
Figure 4. Field-accuracy as function of the number of particles. The number of meshnodes is chosen equal to the number of particles with $fn=100\%$.

The Rostock multi-grid routine should ideally scale as O(N) as function of number of meshnodes in terms of CPU time. As comparison, FFT based Poisson solvers scale as O($N \log(N)$) and particle-particle interaction is an O(N^2) process. The actual performance of the Poisson solver is very close to this ideal case, independent on aspect ratio, as shown in **Figure 5**. The typical scaling is as the number of meshnodes to the power 1.1.
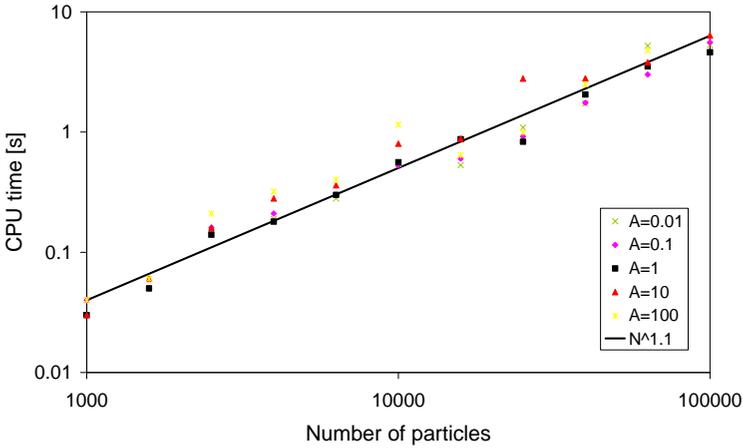
Figure 5. CPU time as function of number of particles for various aspect ratios. The number of meshnodes is chosen equal to the number of particles with *fn*=100%. The solid line is an exponential fit through all points and scales with the number of meshnodes to the power 1.1.

5. Tracking tests

5.1 Exploding bunch

The first tracking test of the new space-charge routine was a simulation of an exploding hard-edged cylindrically symmetric bunch during 100 ps, as shown in **Figure 6**. The initial bunch charge of 1 nC is uniformly distributed over a 'pillbox' with an initial radius R=1 mm and a bunchlength L=0.1 mm. The Lorentz factor γ is 5.
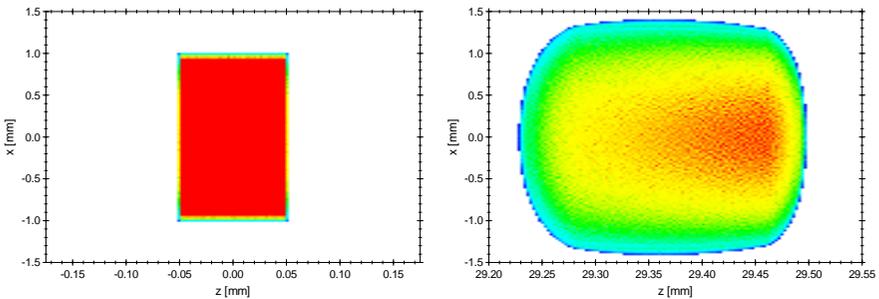
Figure 6. One million particles are tracked during 100 ps on a mesh with 64x64x64 meshnodes.

Figure 7 shows a comparison of bunch-length and emittance of the new 3D space-charge routine with the well-tested 2D space-charge model of GPT. When the number of particles is increased, there is smooth convergence in both routines to the same values. For this specific scenario, about 10,000 particles are sufficient for convergence of the 3D model.
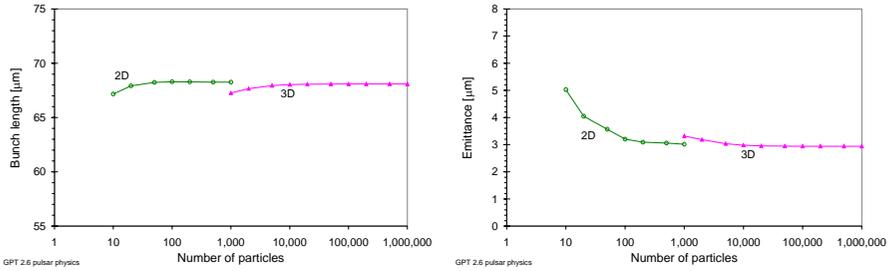
Figure 7. Final bunchlength and emittance values for an exploding R=1 mm, 1 nC bunch after 100 ps. Both 2D and 3D results are shown as function of number of particles.

5.2. *1 GV/m diode*

To test the new space-charge routine in GPT for a scientifically relevant case, it has been used to simulate the 1 GV/m diode as is currently under construction at the Eindhoven University of Technology [11]. The set-up consisting of an anode with a circular opening and a flat cathode is shown schematically in **Figure 8**. During 1 ns, a 2 MV pulse is applied between the cathode and the anode, resulting in a DC acceleration field of 1 GV/m. This field is used to accelerate an R=0.5 mm, 100 pC electron bunch photo-extracted from the cathode surface by a 50 fs FWHM laser. Although the peak-current is over one kA at initiation, the bunch-quality is maintained due to the high acceleration field.
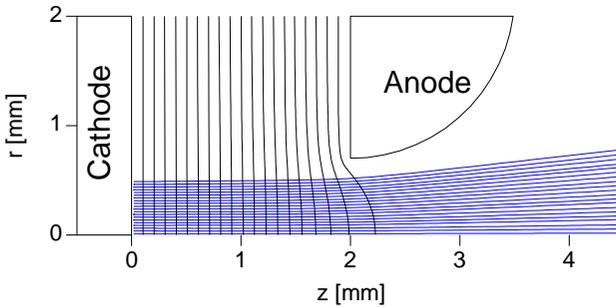
Figure 8. Schematic set-up of the 1 GV/m diode. Shown are equipotential lines and sample trajectories.

The simulation results shown in **Figure 9** are a snapshot in time at z=4.5 mm. The large number of particles, 100,000, produces for more detailed results compared to the previously obtained 2D results [11].
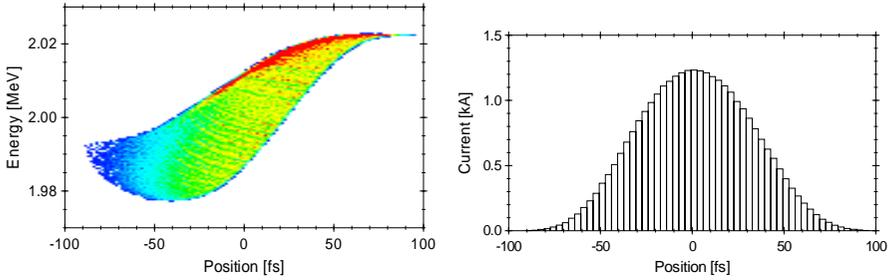
Figure 9. Raw simulation results of the 1 GV/m diode with 100,000 particles on a 60x60x60 mesh with *fn*=100%. Shown are (left) the energy/position correlation and (right) the current.

Because the set-up is fully cylindrically symmetric, it can again be compared with the 2D space-charge model of GPT. Because the 2D model does not make use of a transformation to a rest-frame, the accuracy is not affected by relative energy spread. As a result, this comparison verifies that for this specific case the electrostatic assumption in the rest-frame is valid.
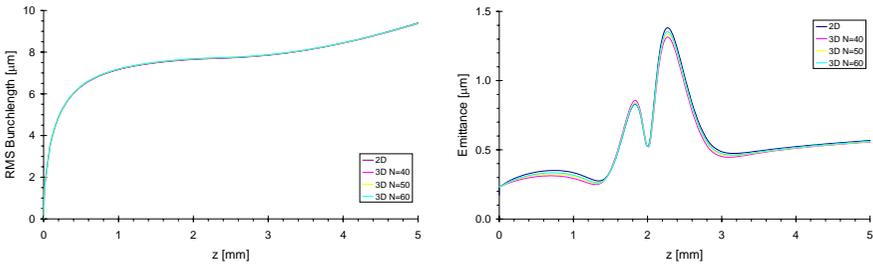
Figure 10. Comparison between the GPT 2D model and the new 3D mesh-based route applied to the TU/e 1GV/m diode set-up. The different lines indicate different number of meshlines in the 3D model.

6. Conclusions

A new 3D mesh-based space-charge model has been implemented in the General Particle Tracer (GPT) code. The routine is based on a multi-grid Poisson solver to obtain the electrostatic potential in the rest frame of a charged particle bunch on a non-equidistant adaptive mesh. Lorentz transformation and interpolations are used to obtain the electromagnetic fields in the laboratory frame. The multi-grid Poisson solver scales linearly in terms of CPU time as function of the number of meshnodes over a wide range of bunch parameters.

When compared to analytical expressions for a uniformly charged cylindrically symmetric bunch, the average field error is below 10% over a range of 6 orders of magnitude in aspect ratio. Tracking tests for an exploding bunch and a 1 GV/m diode show perfect agreement with 2D simulation results.

The accuracy, speed and applicability over a wide parameter range make the new routine ideally suited for the simulation of 3D space-charge effects in high-brightness electron bunches.

7. References

[1]   M.J. van der Wiel, Conf. Proc. 2nd ICFA Future Accelerator Workshop on 'The Physics of High-Brightness Beams', Los Angeles (1999).

[2]   S. Schreiber, Conf. Proc. ICFA Adv. Accel. Workshop on 'The Physics and Applications of High Brightness Electron Beams', Sardinia, Italy (2002).

[3]   B. Carlsten and T. Raubenheimer, Phys. Rev. E 51, 1453 (1995).

[4]   M.J. de Loos, S.B. van der Geer, F.B. Kiewiet, O.J. Luiten, M.J. van der Wiel, Proc. 2002 European Particle Accel. Conf., Paris, France, p. 1831.

[5]   S.B. van der Geer, M.J. de Loos, Proc. 1998 European Particle Accel. Conf., Stockholm, Sweden, p. 1245.

[6]   GPT User Manual, Pulsar Physics, De Bongerd 23, Soest, The Netherlands, www.pulsar.nl.

[7]   G. Pöplau, U. van Rienen, Proc. 3rd Conf. on Scientific Computing in Electrical Engineering 2000, edited by U. van Rienen, D. Hecht and M. Günther, LNSCE (Springer, Berlin, 2001), p. 169.

[8]   G. Pöplau, U. van Rienen, M.J. de Loos, S.B. van der Geer, Conf. Proc. ICAP, Lansing, Michigan, USA (2002).

[9]   W. Hackbusch, *Multi-Grid Methods and Applications*, Springer, Berlin, 1985.

[10]   O.J. Luiten, *to be published*.

[11]   S.B. van der Geer, M.J. de Loos, J.I.M. Botman, O.J. Luiten, M.J. van der Wiel, Phys. Rev. E 65, 046501 (2002).

S-parameter-based computation in complex accelerator structures: Q-values and field orientation of dipole modes[†]

Hans-Walter Glock, Karsten Rothemund, Dirk Hecht, Ursula van Rienen

Institut für Allgemeine Elektrotechnik, Universität Rostock

D – 18059 Rostock, Germany

Abstract. For structures with very fine details or slight geometrical deviations, an approach fully based on discretization frequently causes problem dimensions which are much too big even for powerful computing hardware. In this paper, two such situations are studied: 1.) simulation of an entire 9-cell TESLA cavity, fully equipped with HOM- and Input-couplers, the latter demanding for 3D-computation; 2.) chains of cavity cells being slightly elliptically deformed with different axis orientation. Both simulations were performed using a scheme which we denote as Coupled-S-Parameter-Calculation (CSC). CSC applies a segmentation in several sections of the entire structure. The scattering properties of every segment need to be computed with field solving codes, resulting in (multi-dimensional) frequency dependent S-matrices. Then CSC is used both for the concatenation of the S-matrices and - if the whole system is entirely closed - to determine its eigen-resonances. In our first example we take profit of the rotational symmetry of the inner part - the TESLA cavity, which therefore can be calculated in 2D. The scattering properties of the couplers at both cavity ends are computed separately. Then the CSC result of the overall system S-matrix is used to derive the Q-values of each single resonance. The study of dipole modes in deformed cavities uses the ability of CSC to calculate the frequencies and the distribution of waveguide mode amplitudes of eigenmodes. It is shown that the orientation of the fields experiences an angular shift along a chain of cells having elliptical deformations of different orientation. Here several parameter sets were analyzed in order to gain some experience on the phenomenon. In this study we took advantage of the analytical description of the cell-to-cell rotation thus leading to very short cpu times.

1. Introduction

Resonators used for accelerating particles were probably with the first objects being studied with numerical codes for electromagnetic fields. Nevertheless their geometric complexity often still demands for simplifications, even on powerful modern computers.

[†] work supported by DESY, Hamburg, Germany

Two typical assumptions are the neglection of structure appendices, especially couplers, and the idealization of pure rotational symmetry. In this paper we are going to describe the application of a method, denoted here as Coupled-S-Parameter-Calculation (CSC), on resonators where none of both simplifications are possible. CSC is based on the partitioning of the entire structure in segments, being small enough to calculate their rf-scattering properties – the frequency-dependent S-matrices - on standard hardware with commonly used field solving codes like MAFIA, CST Microwave Studio™ [1], etc. Afterwards the calculation scheme of CSC is used to combine the segments to build up the entire structure, independent from its topology and the number of waveguide modes needed to describe the segment´s interactions. CSC is able to handle both kind of structures: a. with open ports, delivering the overall S-matrix, and b. completely closed resonators, calculating their eigenfrequencies and internal wave amplitudes. For a detailed description of CSC and its theory see [2, 3].

The first example is the TESLA-9-cell structure [4] (section 3) fully equipped with Higher Order Mode- (HOM-) and input-couplers, that showed to have resonances with unexpected high quality factors in the 2.6 GHz-range [5]. A similar study was presented in [6]. Here three different arrangements of HOM- and input-couplers were analyzed and the according Q-factors were determined. The latter step was performed by applying a fitting procedure (section 2) on the S-parameter-spectra found with CSC. A significant variation of the Q-values was found here.

In the second part (section 4) focus is laid on structures that are derived from ideally circular symmetric geometries, but experienced slight elliptical deformations. Two types of structures are considered: one denoted as "waist-structure", having a big circular cross section at the connecting planes and a small inner cylinder of elliptical shape; the second being derived from an inner cell of the TESLA-resonator with a circular waist, but a slightly elliptical body. For both structure types eigenresonances of chains built from 3, 4 and 6 cells were studied, the cells being skewed one to the next by angles between 0° and 90°. Front-to-end differences of wave polarizations are calculated for all arrangements and a certain set of dipole modes. Furthermore CSC-results of an unskewed 9-element-chain of the TESLA-type cell is compared with reference frequencies [7]. The 4-element chain is investigated in further detail, regarding the fields in all cutting planes.

2.   Q - value determination from S-parameters

For rf devices with resonant behaviour a set of resonance frequencies and Q-values are the main quantities to characterize the object. We developed a procedure to extract these quantities from previously calculated S-parameter curves. It is assumed, that the S(f) curves follows the complex relation

$$S(f) = \sum_{k=1}^{N} \frac{a_k}{2\pi i f - p_k} \tag{1}$$

with the frequency f, the number N of poles and the coefficients a_k and p_k describing the pole. The resonance frequency f_k of the k-th pole and its Q-value are given by equations (2a) and (2b).

$$f_k = \text{Im}\{p_k\}/2\pi \qquad\qquad Q_k = -\text{Im}\{p_k\}/2\,\text{Re}\{p_k\} \tag{2a, 2b}$$

If p_k is determined from the S-parameter curve, both parameters are known. The relation

$$S(f) = \frac{a_k}{2\pi i f - p_k} + R_k$$

(3)

holds in the vicinity of a certain pole k, if we assume, that all other contributions in the vicinity of that pole can be summarized in a frequency-independent residual R_k. Furthermore $S_j(f_j)$ has to be known for a sufficient number z of different f_j in the frequency range near the pole. This leads to an overdetermined set of equations of the form

$$2\pi \, i \, f_j \, R_k + S_j(f_j) \, p_k - (p_k R_k - a_k) \; = \; 2\pi \, i \, f_j \, S_j(f_j)$$

(4)

Or, in matrix-vector notation:

$$M = \begin{pmatrix} 2\pi f_1 & S_1 & -1 \\ & & \\ 2\pi f_z & S_z & -1 \end{pmatrix}; \quad v = \begin{pmatrix} 2\pi i S_1 f_1 \\ \\ 2\pi i S_z f_z \end{pmatrix}; \quad x = \begin{pmatrix} R_k \\ p_k \\ p_k R_k - a_k \end{pmatrix} \Rightarrow M x = v$$

(5)

In the minimal least square sense this system has the explicit solution $x = \left(\overline{M}M\right)^{-1} M^+ v$ with the adjoint matrix $\overline{M}$ of M which directly yields the parameters searched for, i.e. p_k, a_k and R_k - the latter serving as a reliability check.

The success of this procedure strongly depends on the selection of frequency points used for the overdetermined system. Taking a span of a few dB on one side of a resonance curve, starting from the maximum, usually gives good results. In special cases like the spectrum shown in figure 3 with resonances of strongly varying Q-values, an additional step needs to be performed by the user: Then, in a first run it is necessary to subtract the contribution of the low-Q-resonances determined from the S-parameters, afterwards repeating the procedure with the remaining signal. Like CSC, this procedure was implemented using Mathematica™ [8].

3.   The TESLA-9-cell cavity

Experimental investigation of the HOM-properties of the TESLA-9-cell resonator (figure 1) [5] proved the existence of at least one mode with a surprisingly high Q-value. This gave reason for intense numerical investigations, which finally were able to explain the observed phenomenon (see especially [6], where a method very similar to CSC was applied).

Figure 1. Cavity for particle acceleration as used in the TESLA Test Facility at DESY [4]. The circular symmetry of the nine cells is lost when the couplers attached at both ends of the cavity are taken into account. (Picture courtesy to DESY.)

Our S-parameter calculations were performed with an arrangement similar to a test stand set-up used for measurements at DESY (figure 2): a TESLA-9-cell-cavity, equipped with upstream HOM-coupler and downstream HOM-input-coupler combination. The beam pipes at both ends of the cavity are shortened at a distance of L2,u = 101.4 mm and L2,d = 65.4 mm measured from the middle plane of the HOM-coupler. The scattering parameters of the TESLA-cavity were calculated from modal coefficients [9, 10], which were determined by eigenmode calculations. This approach is advantegeous in case of objects with significant resonant-like behaviour, which would cause extremely long time domain calculations [6]. The S-parameters of the shorts and beam pipes were calculated analytically, the HOM-coupler and HOM-input-coupler segments were modelled in CST Microwave Studio™. To compare the effects of geometry variations on S-parameters and Q-values the following set-ups were computed:

a) input coupler empty and closed, HOM-coupler in original orientation
b) input coupler empty and closed, HOM-coupler mirrored, as suggested by M. Dohlus [10] to ensure better coupling to all dipole mode orientations
c) input coupler with inner conductor shortened, HOM-coupler in original orientation

The coaxial ports of both HOM-couplers were left open.
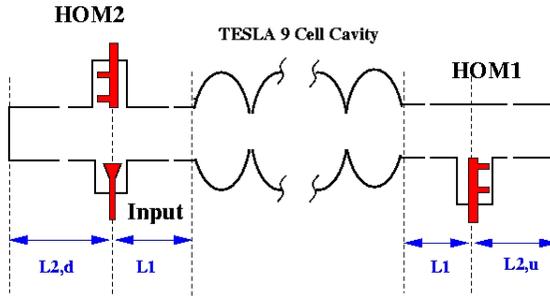
Figure 2. Geometry set-up as used for CSC calculation.
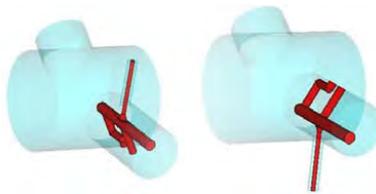
Figure 3. HOM-coupler in original (left) and mirrored (right) version. In arrangements a) and b) the input-coupler port has no inner conductor and is closed; in arrangement c) a shortened inner conductor is added. The figure shows the geometry as modelled by CST Microwave Studio™.

For all three set-ups CSC was used to compute the transmission coefficients from HOM1 to HOM2 in the range of the second dipole passband (2.47 GHz – 2.58 GHz), showing eight high-Q resonances and eight well known low-Q resonances (see figure 4). The spectra of the Q-values of all resonances were extracted from the CSC-calculated $|S_{21}|$ values as shown in figure 4 for the three different set-ups a) to c) as described before. A very narrow frequency sampling of the S-parameters is needed in the vicinity of the high-Q resonances.
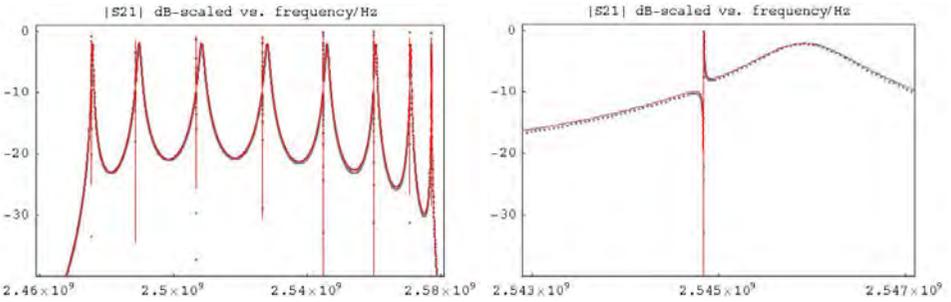
Figure 4. Left: Full spectrum of CSC calculated transmission values ($|S_{21}|$, grey dots) and poles according to equ. (1), (solid line) of set-up a), right: detail of high-Q and low-Q resonance at a frequency of about 2.545 GHz
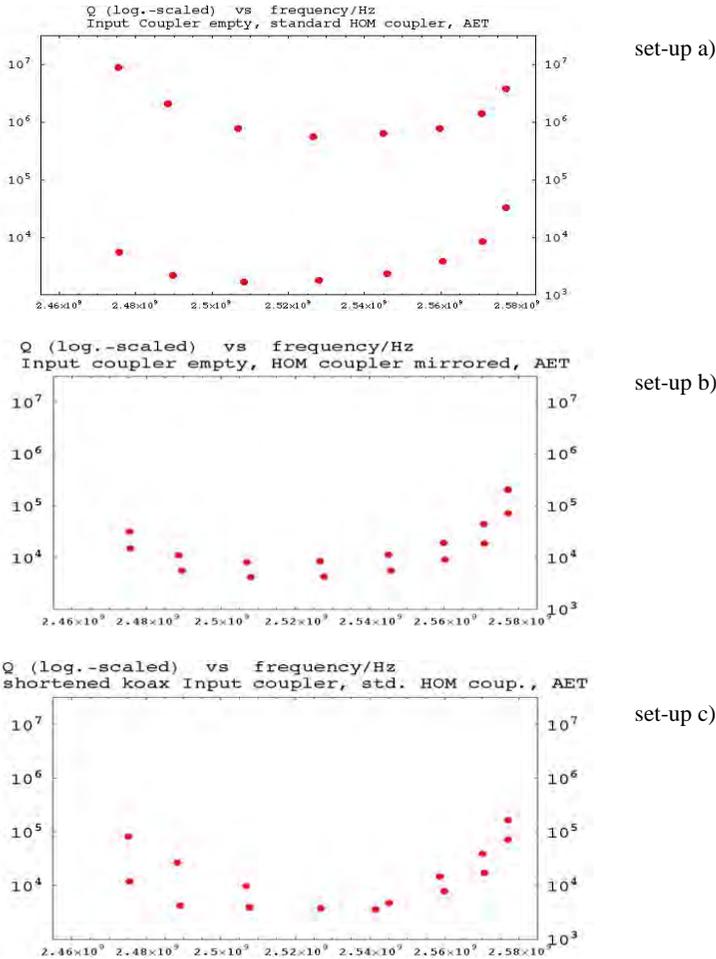
Figure 5. Spectra of Q-values extracted from CSC calculated $|S_{21}|$ values for the three different set-ups a) to c)

This can be provided either by analytic calculation (short, waveguide, cavity with modal expansion) or by interpolation of the component´s S-parameters (HOM-/Input-couplers which have weak frequency dependence). These Q-values of the high-Q resonances are strongly depending on the HOM-coupler orientation and on the presence of the inner conductor in the input coupler. For set-up a) HOM-resonances with extremely high Q-values ($>10^6$) can be indicated in the TESLA cavity with HOM-couplers. With the mirrored HOM-coupler these Q-values are reduced by about two orders of magnitude. But also if the (shortened) inner conductor at the combined HOM-input-coupler is present, the Q-values decrease by nearly the same order.

4.   Orientation of dipole-polarization in skewed chains

Most resonators used for the acceleration of charged particles are designed to be of ideally circular cross section in the transverse plane. Nevertheless, deviations from that ideal shape may occur after installation due to tolerances, improper handling, external forces, etc. Therefore it is worth to gain better understanding of the influence of weak deviations from rotational symmetry. Elliptical deformations as they are used here seem to be a proper first order approximation of more complicated imperfections. Since there is an overwhelming variety of possible combinations of deformations, it was decided to study first a very regular set-up, consisting of chains with a constant skewing angle between neighbouring cells.

Even though a single elliptic structure cannot couple waves oriented orthogonal to each other, it will scatter both polarizations differently. Then, if two or more segments with different orientations are combined, multiple reflections in the chain will lead to an overall coupling of polarizations. This influences both the polarization crosstalk even if deformation and skewing angle are kept very small [2], and the resonant fields and eigenfrequencies as well, if such a chain is closed. In either case a method like CSC is by far better suited to calculate the effect of the rotation than direct 3D discretization. In CSC, rotation is described by a virtual scattering element without reflection, having transmission submatrices, that rotate the incident wave amplitudes by a given angle φ:

$$\begin{pmatrix} 0 & 0 & \cos\varphi & \sin\varphi \\ 0 & 0 & -\sin\varphi & \cos\varphi \\ \cos\varphi & -\sin\varphi & 0 & 0 \\ \sin\varphi & \cos\varphi & 0 & 0 \end{pmatrix} \tag{6}$$

In order to start with a very simple geometry a symmetrical waist segment (figure 6) was designed. The outer diameters and the connecting cones both are ideally circular, whereas the inner part is of a slightly elliptical shape (ellipticity = 1.022). Three to six of those segments were combined. Deformations of the real accelerating structure are more likely to affect those regions with large radii. In order to describe this, a deformation model of a single inner TESLA cell (figure 7) was introduced. The curve of revolution that describes an ideal rotational TESLA cell is divided in three parts: the narrow waist **I** follows an ellipse, the inner part **III** is given as circular segment, between them exists a linear connection **II**. Transversal elliptical deformation is applied similarly: no deformation of the (mechanically stable) waist (**I**); maximum ellipticity in the inner part (**III**); ellipticity linearly growing from zero to maximum in the linear part (**II**). MAFIA was used to discretize the cell as stack of 56 slices either of circular or elliptical cross section and about 2 mm thickness in main axis direction.
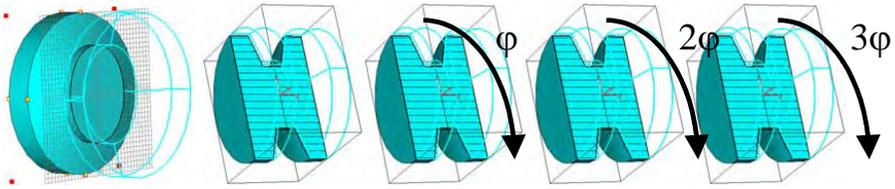
Figure 6. Single waist segment (ellipticity = 1.022), calculated in 3D with 86,000 mesh points in CST Microwave Studio™ (left). Several segments were combined, each rotated to the next by a certain angle φ (right).
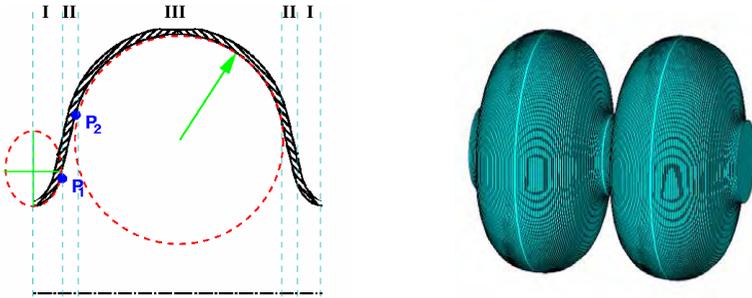
Figure 7. Cross section of a single cell used for S-matrix calculation and double cell arrangement for the check of concatenation results. Discretization with MAFIA, 150,000 mesh points per cell. The ellipticity is too weak to be seen in the picture.
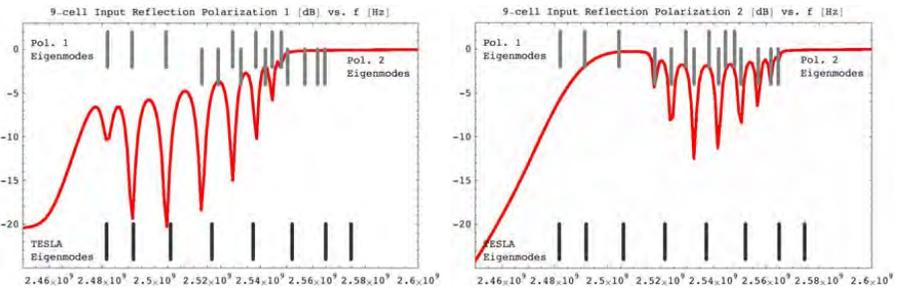
Figure 8. CSC-calculation of input reflection of nine identical unskewed but elliptic cells for both polarizations, together with frequencies from CSC eigenmode calculation (grey lines, distinguished with respect to the according polarization found) and TESLA resonator eigenmodes (black lines) as listed in [7]. (Note: In [7] the TESLA resonator had modified end cells.)

Even though a chain of nine identical elliptic cells does not correctly resemble the TESLA-9-cell cavity with its special end cells, differing from the inner ones, such a chain was calculated without rotation using CSC. The comparison of the results both from CSC-S-parameter and CSC-eigenmode calculations shows a good agreement with resonance frequencies simulated in 2D for an ideal circular resonator [7] as shown in figure 8. The main difference is the existence of twice the number of resonances due to the splitting in two polarizations.

118
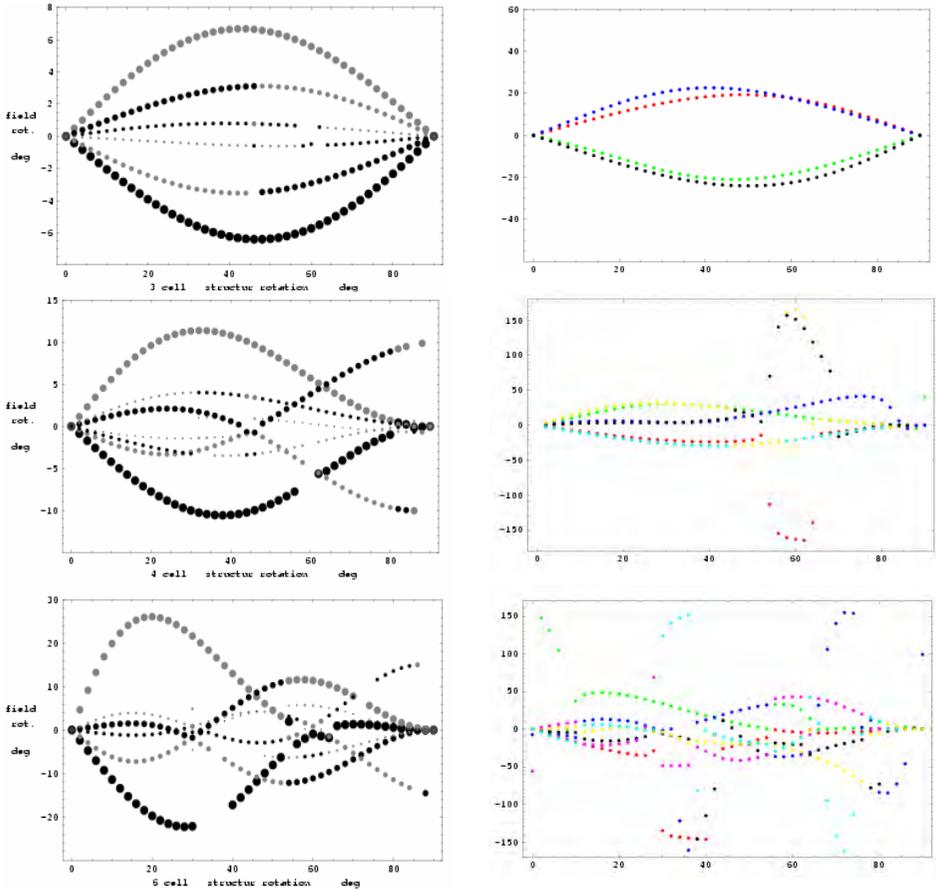
Figure 9. Difference of field orientation angle from left to right of cavity chain depending on cell-to-cell skewing angle for (up to down) 3, 4 and 6 segments. Left column: waist segments, ellipticity = 1.022, frequency range f = 1.31GHz - 1.69 GHz, right column: resonator cell, ellipticity = 1.01, f = 2.45 GHz - 2.60 GHz. Different colour/dot thickness indicates different mode index. Continuous curves are frequently composed by dots from modes with different indices, due to changes in the order of frequency. Gaps are caused by the limited frequency range used for calculations.

Next we consider chains of three, four and six identical segments, each segment rotated about a certain angle compared to its neighbour. The chain is shortened with planes at both ends. Again the only field solving calculation needed for this analysis is the calculation of the single segment's S-matrix. In the S-matrices both polarizations of the TE11 circular waveguide modes are considered. The monopole TM01 and higher waveguide modes are omitted, because of their very weak coupling to the TE11 modes. Therefore only dipole-like eigenmodes of the chains are calculated.
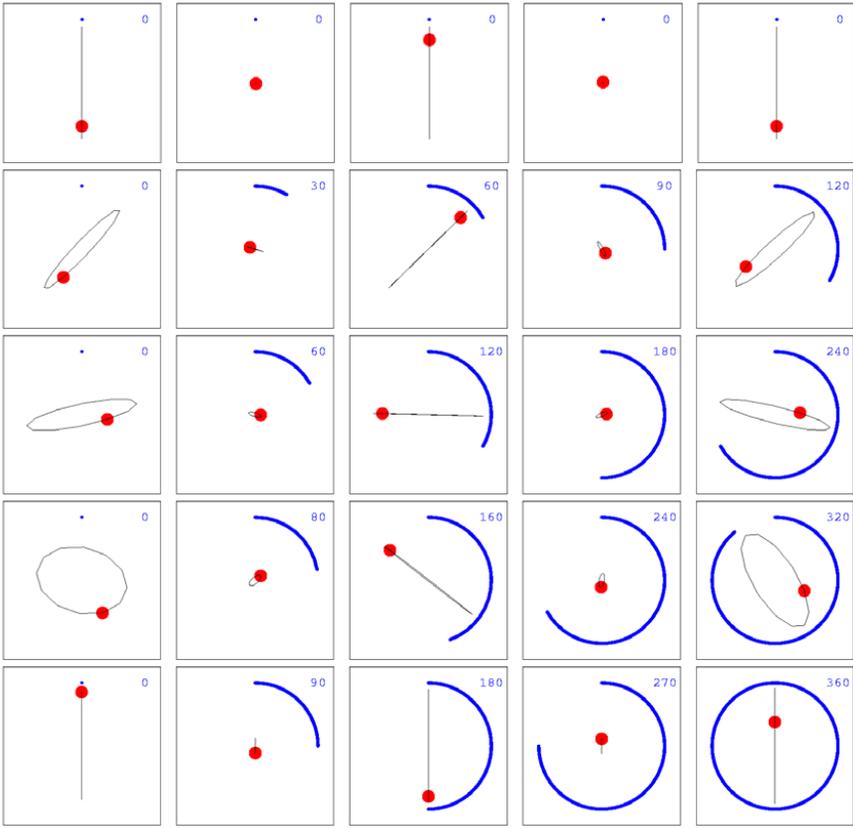
Figure 10. Magnetic field polarization of a certain mode at the cutting planes in a resonator of four TESLA cells for skewing angles between 0 and 90 degrees.
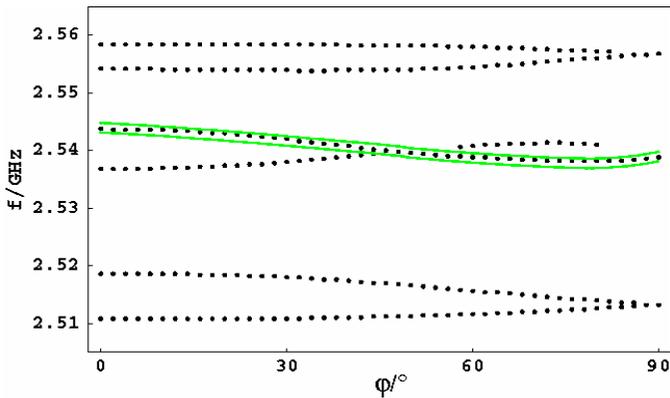
Figure 11. Dependence of the resonant frequencies of a closed four element TESLA chain on the skewing angle. Mode patterns displayed in figure 10 correspond to the modes starting at 2.545 GHz / 0°, specially marked here.

120

Introducing a nonvanishing angular shift between single cells of a multicell structure, which is assumed to be constant along the chain, showed to cause a different orientation of the incident dipole mode waves at both ends of the chain. The orientation is calculated as $\operatorname{atan}(A_y/A_x)$ using the wave amplitudes A_x and A_y that belong to the resonances and that are found directly by CSC. Here the difference of wave angles at beginning and end of the skewed chain is analyzed (Fig. 9).

For the chain of four TESLA-like cells a further analysis of the field orientation in all cutting planes was undertaken by superposition of the forward and backward waves. A representation by the field vector´s path during one rf period seemed to be most appropriate (Fig. 10). As shown there, fields of elliptic polarization do appear for certain skewing angles, whereas pure linear polarizations are valid for 0° and 90° skewing angle.

5.   Conclusions

The segmentation of long or complex structures by means of Coupled-S-parameter-Calculation proved to be a method that may significantly reduce numerical effort. It is a versatile tool to calculate properties of resonators equipped with couplers, if there is an appropriate partitioning possible.

Due to CSC´s semi-analytic character it allows for easy computation of polarizational crosstalk and eigenmodes in skewed chains. Those show a complicated dependence on the skewing angle, especially linear polarization is lost. The longer the chains are the more complicated are the dependencies of field rotation on the skewing angle.

References

[1]   MAFIA V. 4.106 and CST Microwave Studio[TM] Version 3 CST GmbH Bad Nauheimer Str. 19 D-64289 Darmstadt
[2]   Glock H-W Rothemund K van Rienen U 2002 IEEE Transactions on Magnetics Vol 38 Issue 2 Part 1   1173 -1176
[3]   Glock H-W Rothemund K van Rienen U 2001   accepted for publication in Proceedings of the GAMM-Workshop Computational Electromagnetics 26-28 Jan 2001 Kiel to appear in Springer Series "Lecture Notes in Computational Science and Engineering"
[4]   TESLA Technical Design Report 2001 DESY 2001-011 ECFA 2001-209 TESLA Report 2001-23 TESLA-FEL 2001-05
[5]   Baboi N Dohlus M Magne C Mosnier A Napoli O Glock H-W 2000 Proceedings EPAC 2000 Vienna Austria   1107 – 1109
[6]   Dohlus M Wipf S G Kaljuzhny V   TESLA Report 2002-05 DESY Hamburg
[7]   Wanzenberg R TESLA Report 2001-33 DESY Hamburg
[8]   Mathematica™ V4.1 Wolfram Research Champaign IL USA
[9]   Dohlus M Schuhmann R Weiland T 1999 Int. J. Numer. Model. 12 41-68
[10]   Dohlus M private communication

AHF booster tracking with SIMPSONS

D. E. Johnson† and F. Neri‡

† SAIC/InterTech Resources Development, Inc., Dallas, TX 75219
‡ Los Alamos National Laboratory, Los Alamos, NM 87544

Abstract. The booster lattice for the Advanced Hydrotest Facility at Los Alamos was tracked in 3-D with the program SIMPSONS, using the full, symplectic lattice from TEAPOT, the full set of magnet and misalignment errors, and full space-charge effects. The only corrections included were a rough closed-orbit correction and chromaticity correction. The lattice was tracked for an entire booster cycle, from multi-turn injection through acceleration to the top energy of 4 GeV, approximately 99,000 turns. An initial injection intensity of 4×10^{12}, injected in 25 turns, resulted in a final intensity of 3.2×10^{12} at 4 GeV. Results of the tracking, including emittance growth, particle loss, and particle tune distributions are presented.

1. Introduction

The booster lattice for the proposed Advanced Hydrotest Facility (AHF) at Los Alamos was tracked in 3-D with the program SIMPSONS [1], using the full, symplectic lattice from TEAPOT [2], the full set of magnet and misalignment errors, and full space-charge effects. The only corrections included were a rough closed-orbit correction and chromaticity correction. The lattice was tracked for an entire booster cycle, from multi-turn injection through acceleration to the top energy of 4 GeV, approximately 99,000 turns. An initial injection intensity of 4×10^{12} protons, injected in 25 turns, resulted in a final intensity of 3.2×10^{12} at 4 GeV. Results of the tracking, including emittance growth, particle loss, and particle tune distributions are presented.

2. Booster design

The booster for the AHF at Los Alamos is a 9-period, rapid-cycling synchrotron with circumference of 261 m, operating between 157 MeV and 4 GeV, injecting into a 50 GeV Main Ring. It is filled from an H⁻ linac, and is designed to operate at 5 Hz with an average intensity in excess of 2×10^{12} protons per pulse. Lattice functions for this machine are shown in Fig. 1. Fig. 2 shows a 3-D layout of the ring.

B9L
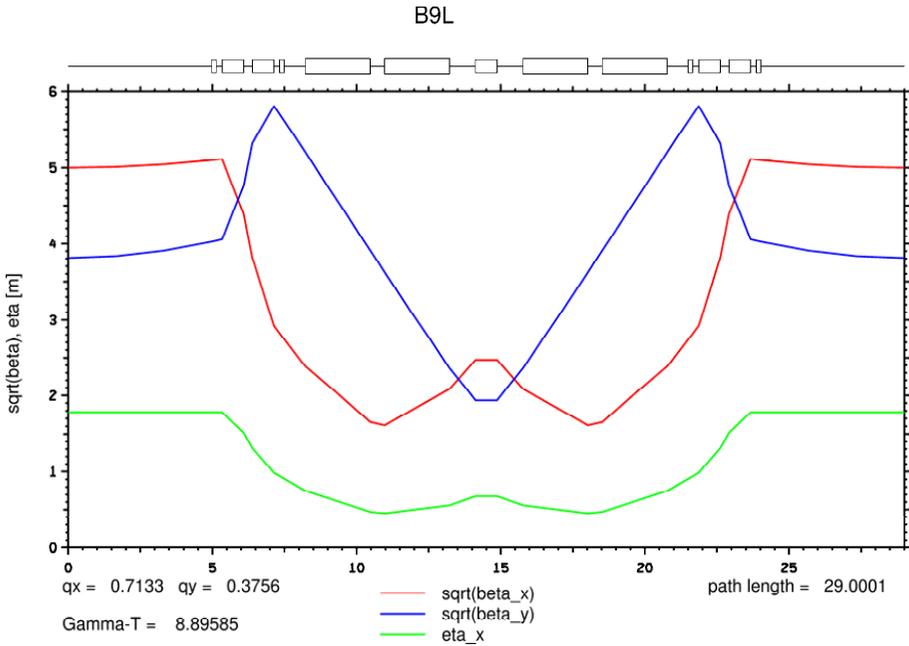
Figure 1. AHF Booster Lattice Functions for 1/9 of the ring
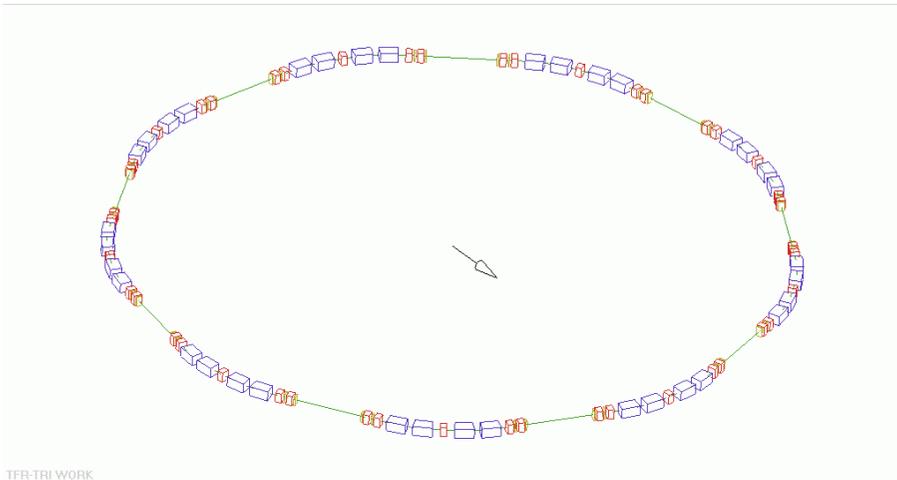
Figure 2. 3-D layout of the AHF Booster

3. TEAPOT

TEAPOT tracking for the booster has been done to specify the error content of the magnets. We use the magnets described in the Fermilab Main Injector Design Handbook [3], scaled for energy and aperture. We discuss below how the errors were determined for the booster magnets.

3.1 Dipole errors

The dipole errors in the FMI Handbook are shown in Table 1.

Table 1. Dipole errors (part in 10^{-4} at 1 inch)

Harmonic Number	Normal Systematic	Skew Systematic	Normal Random	Skew Random
1	0.737	-	10.251	-
2	0.06	-	0.8	-
3	-0.6	0	0.18	0.12
4	0.04	0.03	0.06	0.03
5	0.33	0	0.05	0.05
6	-0.01	-0.03	0.05	0.04
7	-0.03	0	0.05	0.05

The errors in Table 1 are given in term of parts in 10^{-4} of the main field at one-inch radius from the beamline center. Because TEAPOT uses the very different units of fractions at one meter, the numbers appearing in the input files are very different. Table 2 gives the dipole errors in TEAPOT units. Note that the harmonic number we use is one unit larger that the index used in TEAPOT. Our Harmonic number 1 skew multipole is TEAPOTS' a_0, *etc.*

Table 2. Dipole errors (TEAPOT units)

Harmonic Number	Normal Systematic	Skew Systematic	Normal Random	Skew Random
1	0.74E-04	-	0.10E-02	-
2	0.16E-03	-	0.21E-02	-
3	-.42E-01	0	0.19E-01	0.12E-01
4	0.16E+00	0.12E+00	0.25E+00	0.12E+00
5	0.53E+02	0	0.80E+01	0.80E+01
6	-.63E+02	-.19E+03	0.32E+03	0.25E+03
7	-.75E+04	0	0.12E+05	0.12E+05

Since we do not have a booster dipole design specified, the booster runs were done with errors obtained by scaling the FMI dipoles, scaled to reflect the larger gap (4 inches *vs.* 2 inches). The process involves two steps. First the same errors are assumed at a proportionately larger radius for the larger magnet. Then the multipoles at a different radius are obtained using the formula

$$b_m(R) = \left(\frac{R}{r}\right)^{(m-1)} b_m(r).$$

(1)

Equation 1 relates the multipole errors for the same dipole at different radii.

Using Equation 1 we can derive TEAPOT multipoles for 4" magnets dipoles from the values of Table 1.

Table 3. 4" Booster Dipole errors at injection (TEAPOT units)

Harmonic Number	Normal Systematic	Skew Systematic	Normal Random	Skew Random
1	0.74E-04	-	0.10E-02	-
2	0.79E-04	-	0.11E-02	-
3	-.10E-01	0	0.47E-02	0.31E-02
4	0.20E-01	0.15E-01	0.31E-01	0.15E-01
5	0.33E+01	0	0.50E+00	0.50E+00
6	-.20E+01	-.59E+01	0.99E+01	0.79E+01
7	-.12E+03	0	0.19E+03	0.19E+03

3.2   Quadrupole errors

Table 4 summarizes the errors in the quadrupoles we obtained from scaling the errors for the FMI quadrupoles from the Design handbook.

Table 4. Original FMI Quadrupole errors (part in 10^{-4} at 1 inch)

Harmonic Number	Normal Systematic	Skew Systematic	Normal Random	Skew Random
2	-	-	24	-
3	-0.51	1.08	2.73	1.85
4	1	-2.05	1.02	2.38
5	0.03	-0.75	1.12	0.47
6	-1.49	0.43	0.63	0.70
7	0.21	-	0.64	0.44
8	1.14	-	0.64	-
9	-0.19	-0.07	0.12	0.16
10	-0.77	-0.12	0.06	0.07

The errors of Table 4 are very large, especially for the high-order multipoles. There is some suspicion that the values reported are upper limits, because such high values are somehow unphysical.

3.3   Revised quadrupole errors

Table 5 summarizes the quadrupole errors that were agreed on after an analysis of what kind of errors can be realistically obtained. The numbers are obtained (sometimes with scaling) from actually produced magnets and represent what can be achieved without increasing the quadrupole cost.

The formula used for scaling the quadrupole errors is

$$b_m(R) = \left(\frac{R}{r}\right)^{(m-2)} b_m(r). \tag{2}$$

We have considered two possible apertures for the booster dipoles. The larger quadrupoles have an 8" diameter, twice the dipole gap. The nominal quadrupole diameter is only 1.5 times the dipole gap. 6" quadrupole values will be shown in the tables, but 8" tracking results will be included in the tracking results. Using Equation 2 and converting to TEAPOT units, the quadrupole errors are shown in Table 6.

We should again point out that the index used by TEAPOT is one less than the harmonic number we use.

Table 5. Revised Quadrupole errors (part in 10^{-4} at 1 inch)

Harmonic Number	Normal Systematic	Skew Systematic	Normal Random	Skew Random
2	0	-	12	-
3	0.2	0.3	1	1
4	-0.5	0.1	1	0.3
5	-0.1	-0.1	0.15	0.1
6	0.4	-0.1	0.1	0.1
9	-	-	-	-
10	-	-	-	-

Table 6. Revised Booster 6" Quadrupole Errors at Injection (TEAPOT units)

Harmonic Number	Normal Systematic	Skew Systematic	Normal Random	Skew Random
2	0	-	0.12E-02	-
3	0.52E-03	0.79E-03	0.26E-02	0.26E-02
4	-.34E-01	0.69E-02	0.69E-01	0.21E-01
5	-.18E+00	-.18E+00	0.27E+00	0.18E+00
6	0.19E+02	-.47E+01	0.47E+01	0.47E+01
9	-	-	-	-
10	-	-	-	-

4. Injection

25-turn, multi-turn injection was used for the booster. The incoming beam was painted in the four transverse dimensions in a standard method. The beam was also painted longitudinally by modulating the momentum coming out of the linac and by chopping the head and tail of the linac beamlet. The number of particles injected on each turn varied, corresponding to the length of the chops. A total intensity of 4×10^{12} protons were injected. The resulting pulses just matched the waiting rf bucket and the bucket was uniformly filled from low to high momentum. The injection was done on-the-fly at the bottom of a sine ramp into an accelerating bucket. The fully-injected beam was then tracked for approximately 99,000 turns until it reached its extraction energy of 4 GeV

Neither the transverse painting nor the number of injected turns were fully optimized. Looking at the final beam loss & emittance blow-up, an injection scheme with more turns and a larger transverse displacement would be advantageous. The number of turns possible from the linac will have to be studied more closely.

126

In the simulation, 20,335 macroparticles were used, and space-charge kicks were applied 4 times per cell. A run of 99,000 turns required approximately 500 hours of CPU time on a 1.5 GHz dedicated machine. A uniform beampipe radius of 7.5 cm was taken for the definition of lost particles. This may need to be somewhat reduced in order to account for the beampipe thickness. The rf profiles and brho curves were derived by L. Rybarcyk [4].

Macroparticle distributions for the injection process are presented in Fig. 3.
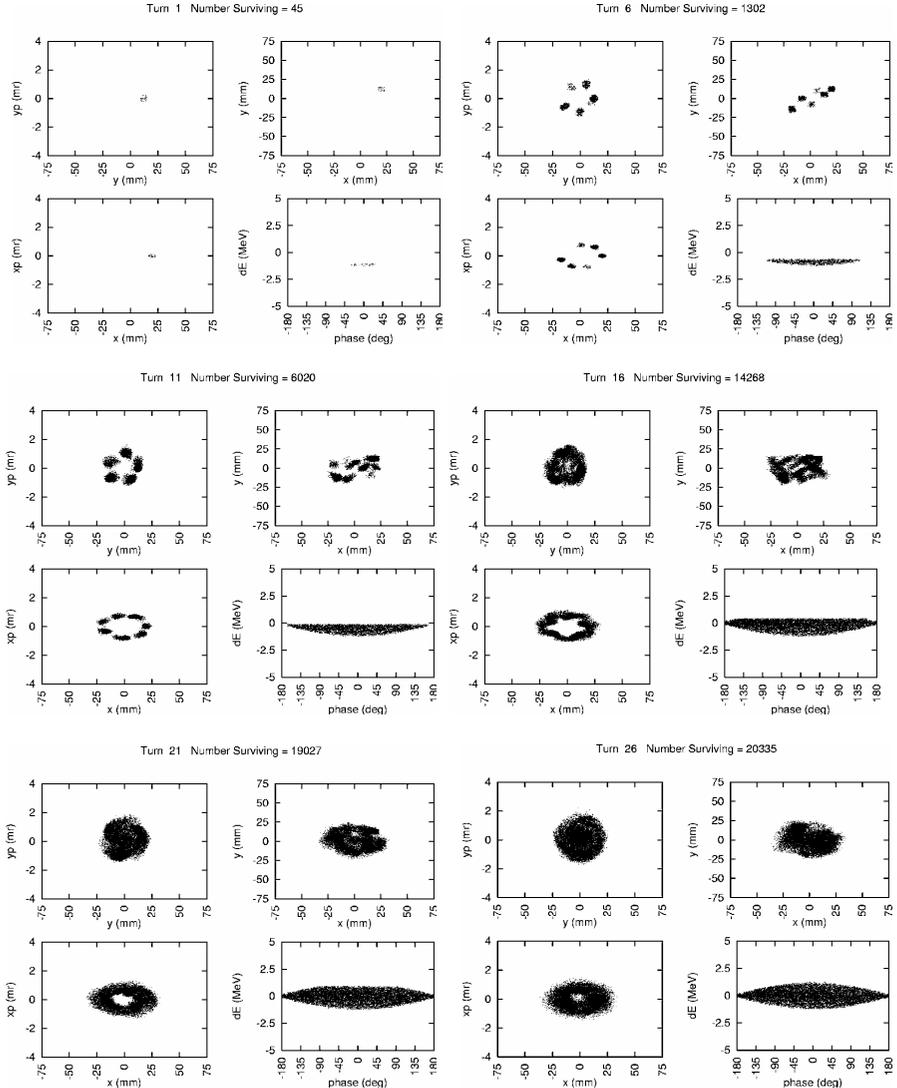
Figure 3. B9L Injection Macroparticle Distributions.

5. Full booster cycle tracking

The Booster was tracked for the full acceleration cycle of approximately 99,000 turns. The beam exhibited both emittance growth and beam loss for the first few thousand turns, then settled down and was accelerated smoothly for the remainder of the cycle. Typical macroparticle plots during the acceleration cycle are presented in Fig. 4 and plots representing the full beam behavior during the cycle are shown in Fig. 5. As can be seen from the emittance plots, the injection parameters need to be better optimized. The resultant beam at the end of the acceleration cycle has suffered an approximate 22% loss from the initial injected beam, or a reduction from 4.07×10^{12} total particles to 3.17×10^{12}.
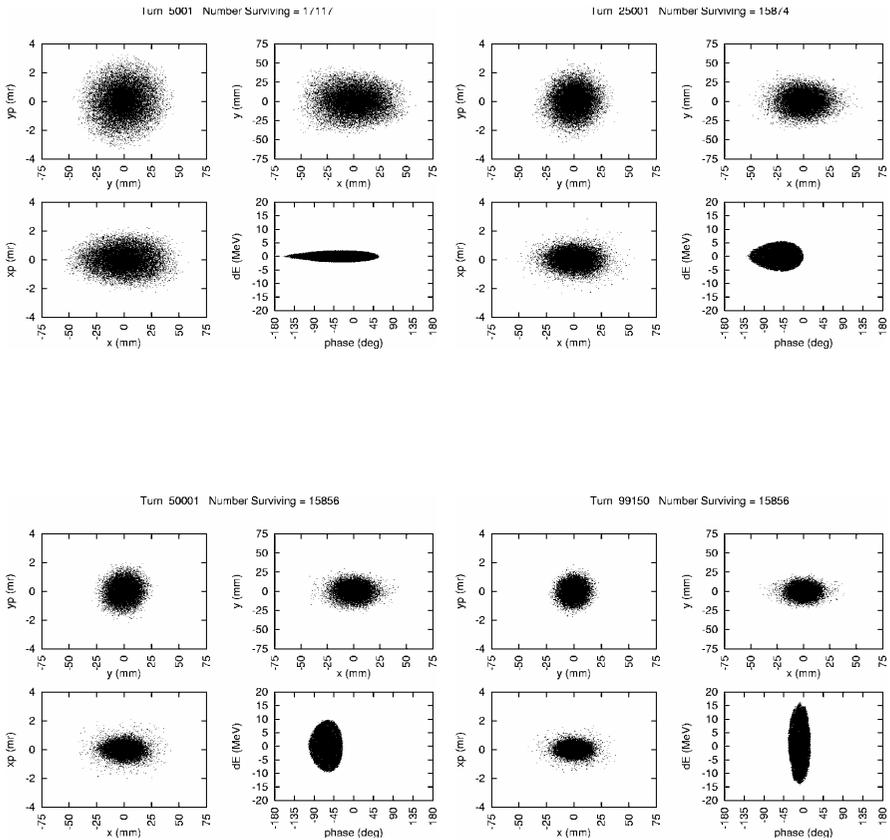
Figure 4. B9L Macroparticle Distributions during the Acceleration Cycle.
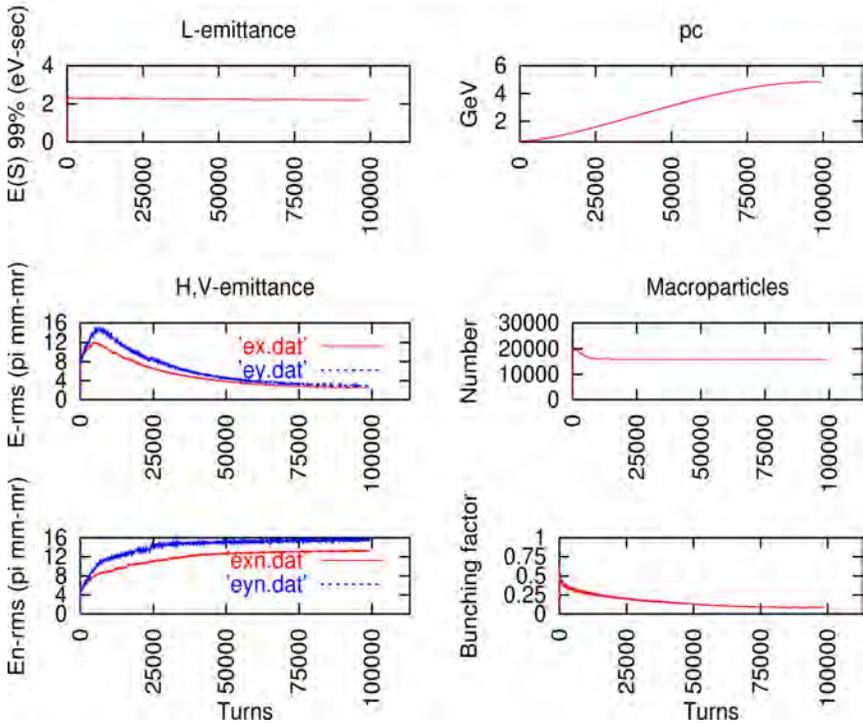
Figure 5. B9L Beam parameters During the Acceleration Cycle.

6. Booster Tune-Density Plots

During the booster tracking, Fast Fourier Transformations (FFT) were performed on the macroparticle distributions periodically. Each FFT scanned the entire 20,000 macroparticle distribution over 128 turns and calculated the individual macroparticle tunes. These tunes were then plotted, using the program POSTER [5]. The resultant tune plots are presented in Fig. 6. Horizontal and vertical tunes are plotted on the X & Y axis. In these plots, the solid and dashed blue lines represent regular and skew systematic resonances, while the gray lines indicate simple error resonances.
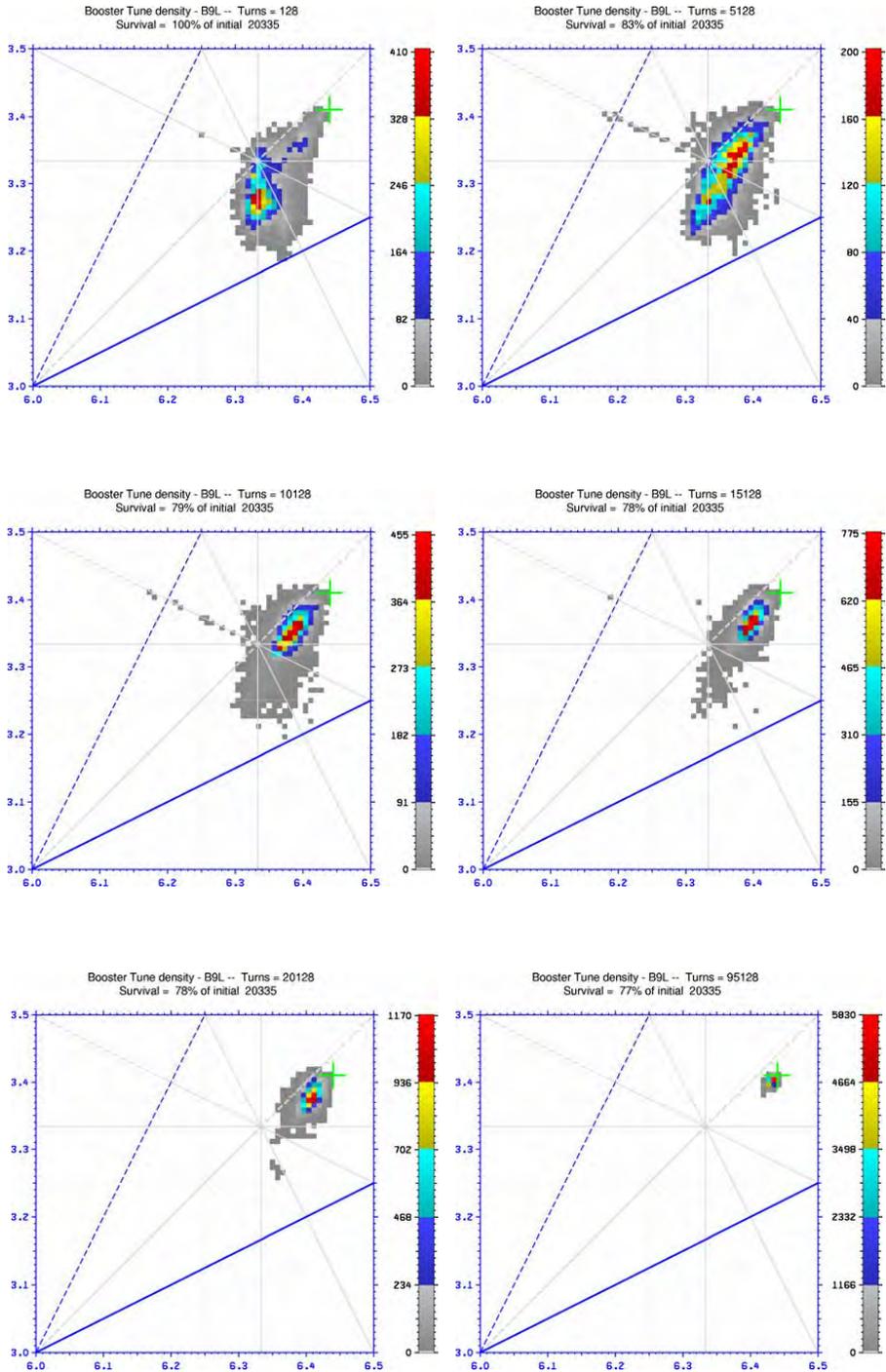
Figure 6. B9L Tune-Density Plots.

The green cross indicates the injection tune point. The color scale to the right of each plot denoted the point densities plotted. Note that the scale changes from plot to plot. It is interesting to note that some particles initially stream out along the normal resonance line $Q_x + 2Q_y = 13$. As can be seen from the plot of the first 128 turns, the beam core crosses this resonance. During the crossing, increased beam loss is observed. An improved injection painting will reduce this resonance-line crossing; however, it would be prudent to correct this resonance with a set of harmonic sextupole correctors. Again, it should be noted that this simulation contains all of the individual magnet systematic and random errors, as described in the above section, as well as magnet misplacement and misalignment errors. The $Q_x + 2Q_y = 13$ resonance is driven only by the random magnet errors and those induced by space-charge forces, as this is not a systematic resonance, and as such is not driven by the chromaticity-correcting sextupoles, to first order.

7. Conclusions

For the first time, tracking in 3-D through an entire synchrotron acceleration cycle has been possible, including magnet errors, misalignments and space charge.

References

[1]   S. Machida, **SSCL-570**, SSC Laboratory, 1992.
[2]   L. Schachinger & R. Talman, *Particle Accelerators*, **22**, pp. 35-56, 1987.
[3]   Technical Design Handbook, Fermilab, 1995.
[4]   L. Rybarcyk, private communication.
[5]   C. T. Mottershead, private communication.

Parallel simulation algorithms for the three-dimensional strong-strong beam-beam interaction

Andreas C. Kabel†

Stanford Linear Accelerator Center,
2575 Sand Hill Road, Menlo Park, CA 94025 ‡

Abstract. Adding a third dimension to the simulation of the strong-strong beam-beam effect can, by mechanisms such as the hourglass effect and phase averaging, significantly change the results of the simulations. However, CPU requirements for such simulations are beyond the reach of single-processor machines. We discuss a set of parallel algorithms for the treatment of the problem, based on particle-mesh methods and rapid elliptic solvers. Particle tracking, particle-mesh deposition, and field calculation are three tasks which have to be parallelized; they have competing requirements for localization of the tasks on the parallel machines. We discuss different possible parallelization schemes. We present a code based on spatial domain decomposition in the longitudinal direction, which shows good scaling behavior for the case of storage rings with small synchrotron tunes.

1. Introduction

The strong-strong beam-beam effect is one of the most important effects limiting the luminosity of ring colliders. Little is known about it analytically, so most studies utilize numeric simulations. The two-dimensional realm is readily accessible to workstation-class computers (cf.,e.g.,[1, 2]), while three dimensions, which add effects such as phase averaging and the hourglass effect, require vastly higher amounts of CPU time. Thus, parallelization of three-dimensional simulation techniques is required; in the following, we consider several approaches, emphasizing computational aspects such as scalability. While the correctness of the code has been checked for analytically accessible cases, physical benchmarking and comparison with experimental results will be subject of a forthcoming paper.

2. The Pool algorithm

A solver for the weak-strong beam-beam problem is embarrassingly parallelizable: a pool of test particles, representing the weak beam, is distributed among processors. In each step, the field of the immutable strong beam is calculated, and the phasespace coordinates of the weak beam's particles are updated accordingly. Inter-process communication is only necessary for the calculation of collective quantities, such as the moments of the weak beam's particle distribution and luminosity.

---

† E-Mail: andreas.kabel@slac.stanford.edu

2.1. Two dimensions

For the strong-strong problem, both beams are represented by test particles. We are now presented with two parallelizable problems: the application of fields to the particles and the calculation of the fields due to the current particle distribution. The latter problem is typically approached by a Particle-on-lattice (POL) algorithm: A discretized, two-dimensional charge density $\rho_{\vec{i}}$ is used to calculate the equally discretized electric potential $\phi_{\vec{i}}$ by solving a discretized version of Poisson's equation. A particle is deposited on $\rho_{\vec{i}}$ ($\vec{i}$, $\vec{k}$,... are two-dimensional integer vectors, corresponding to positions $x = x_0 + h_x i_1, y = y_0 + h_x i_1$ on the grid) by adding charge to the lattice sites nearest to it with distance-depending weights. There are a number of choices for these weights, we use a third-order momentum-conserving approach[3]. For the two-dimensional sub-problem, we make use of algorithms introduced and validated in [1, 2].

Generalizing the weak-strong approach, we divide the particles into pools local to processors. For the two-dimensional problem, a single grid is placed in the interaction point, perpendicular to the directions of motion. A parallelized solver step now looks like this:

(i) (local) collect $\rho_{\vec{i}}$ from test particles

(ii) (global) add all $\rho_{\vec{i}}$ from particle pools

(iii) (global) distribute sum $\rho_{\vec{i}}$ to solver processors

(iv) (local) calculate $\vec{E}_{\vec{i}}$ from $\rho_{\vec{i}}$

(v) (global) distribute $\vec{E}_{\vec{i}}$ to particle pools

(vi) (local) apply $\vec{E}$ to test particles

(vii) (local) transport particles around the complete ring or to the next interaction point (IP)

Each step, of course, needs to be executed for both bunches. There are alternatives to steps 5 and 6, which are discussed below.

The transport operation involves all of the particle dynamics between IPs; for a proton machine, it can be as simple as the application of a symplectic 6×6 matrix to the particles' phasespace coordinates; for an electron machine, the transformation also has to include damping and excitations due to synchrotron radiation processes.

As the deposition and field interpolation step involves the same particles on the same grid, only one interpolation step has to be done, the results of which (namely, the coordinates of the nearest grid points and the weight vectors) can be stored with the particle. As the interpolation step is quite cheap, but storage of the results with each particle needs substantial memory, cache locality effects may make it cheaper to recalculate these quantities on real machines.

2.2. Three dimensions

For the solution of the three-dimensional problem, we divide the bunches into n_z longitudinal slices of equal lengths h_z and slice numbers $i_z \in [0, n_z)$, 0 representing the leading slice. We now need to place (for both bunches) $2n_z - 1$ grids numbered $1 - n_z \ldots n_z - 1$ at positions $\frac{1 - n_z}{2}, \ldots, \frac{n_z - 1}{2}$ around the IP.

For each encounter of bunches, we now need to execute $2n_z - 1$ steps, numbered $s \in [0, 2n_z - 2]$. In step s, particles in slice $k \in [\max(0, s - n_z + 1), \min(s, n_z - 1)]$ are deposited on grid $\pm(k - s)$ (sign according to the direction of flight). All updated grids are then used to calculate fields, and the resulting fields are applied to the opposing bunch's particles

longitudinally nearest to the respective grid. Each bunch encounter thus consists of n_z^2 slice encounters, each involving one two-dimensional deposit/solve/kick step.

Note that the longitudinal particle-to-grid assignment needs to be done only once per bunch encounter, as the longitudinal motion is 'frozen' in the IP.

To move the particles onto the slice, we can either simply project to the slice (i. e. just set the z-coordinate to the grid position) or transport the particle to the grid by applying the transfer matrix of a drift space of appropriate length. Our code leaves the choice to the user.

When the bunch length is comparable to the β function in the IP, the hourglass effect becomes significant, meaning that the grids far away from the IP have to accommodate a larger bunch diameter than the grids close to it. To optimize resolution, we scale the grid resolutions according to $h_{x,y} \propto \sqrt{1 + \frac{z^2}{\beta_{x,y}^{*\,2}}}$.

2.3. Field calculation

The field calculation is based on a convolution algorithm. The discretized charge distribution $\rho_{\vec{i}}$ is convoluted with the discretized Green's function $G_{\vec{i}-\vec{k}} \propto \log \sum_i \frac{(i_i - k_i)^2}{h_i^2}$ for the two-dimensional Coulomb problem. The convolution can be done efficiently by Fast Fourier transforming ρ, doing a point-wise multiplication with the Fourier transform $\tilde{G}$ of G, and transforming back. If we choose a lattice of dimensions $L = [0, 2h_x n_x) \otimes [2h_y n_y)$, but restrict the support of ρ to $L' = [0, h_x n_x) \otimes [0, h_y n_y)$, the $(2n_x, 2n_y)$ periodicity of $\tilde{G}$ will not modify the potential in L', i. e. the method will obtain the correct potential for open boundary conditions. This is the famous Hockney trick [3].

To avoid the singularity of the Green's function at the origin, we choose a natural smoothing prescription: we shift the Green's function by $\frac{1}{2}\vec{h}$, such that $G_0 = 0$, and evaluate the fields at a position shifted by $-\frac{1}{2}\vec{h}$.

Figure 1 shows the x-component of the electric field as a result of the particle deposition, Poisson solver, and interpolation algorithm; the lattice has a size of $512 \otimes 512$ and spans $\pm 15\sigma$ (so there are 100 lattice points used for the interpolation), and the charge density was created by depositing 10^6 Gauss-distributed (for random number generation, see below) particles from a round beam onto the lattice.

The Green's function is pre-calculated at program start. Note that using different lattice resolution for lattices on different z positions means that different Green's functions for each pair of grids centered around each IP have to be pre-calculated, since G obeys no simple scaling law for the case of $\beta_x^* \neq \beta_y^*$.

The convolution approach makes it easy to handle non-congruent lattices, i. e. the case of non-concentric beams ('parasitic crossings'); all that is needed is the convolution with an appropriately shifted Green's function. Similarly, $\tilde{G}$ can be precalculated for the case of the lattice spacings of the charge deposition and the field evaluation lattice not being the same. We plan to implement these generalizations in a future version of our code.

Parallelizing the convolution method amounts to parallelizing the local multiplication with $\tilde{G}$, which is trivially done, and parallelizing the Fast Fourier Transform. For the latter, we use the high-performance, open-source FFT library 'FFTW'[4], which provides an MPI compliant parallel version. The parallel FFT works by dividing the field and charge matrices into slices consisting of adjacent rows. Each solver process then does a one-dimensional fast-Fourier transform of all row vectors in its assigned slice. In an all-to-all communication step, the charge matrix is transposed, and the row transform step is repeated. A subsequent transposition is unnecessary, as we can account for it by using $\tilde{G}^\top$ instead of $\tilde{G}$. The inverse transformation of $\tilde{\rho}^\top \star \tilde{G}^\top$ works the same way, again leaving out one transposition and thus
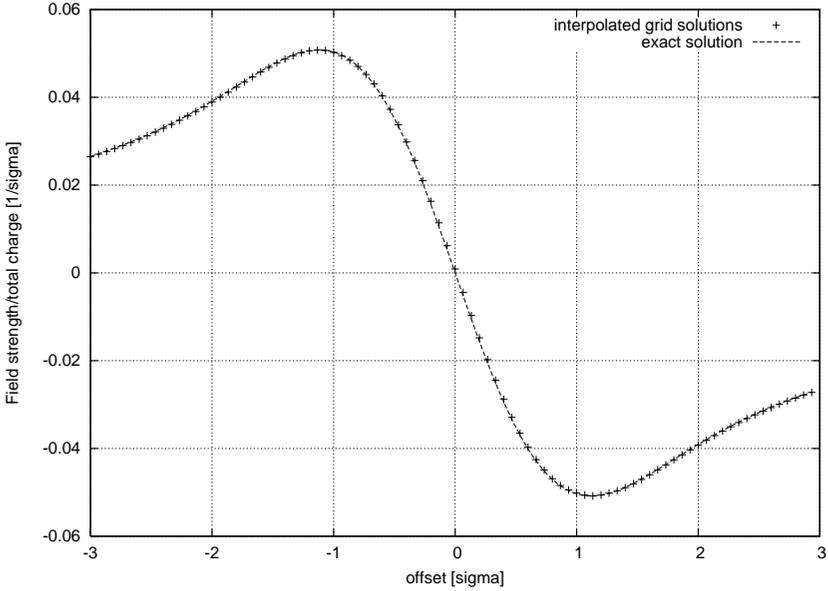
Figure 1. Field calculation vs. analytic results

returning a correctly oriented potential matrix ϕ:

$$\rho_{ik} \xrightarrow[FFT]{} \rho_{i\tilde{k}} \xrightarrow{\top} \rho_{\tilde{k}i} \xrightarrow[FFT]{} \rho_{\tilde{k}\tilde{i}} = \tilde{\rho}_{ik}^{\top} \xrightarrow[\tilde{G}^{\top}\star]{}$$

$$(\tilde{G}\star\tilde{\rho})_{ik}^{\top} = \tilde{\phi}_{ik}^{\top} = \phi_{\tilde{k}\tilde{i}} \xrightarrow[FFT^{-1}]{} \phi_{\tilde{k}i} \xrightarrow{\top} \phi_{i\tilde{k}} \xrightarrow[FFT^{-1}]{} \phi_{ik} \quad . \quad (1)$$

Calculation of the electric field is done by discretized differentiation with an appropriate weight algorithm[2]. In theory, the discretized differentiation algorithm could be absorbed into the fourier-transformed Green's function, in practice, this solution turns out to be slower, as the CPU time for the backwards fourier transform roughly doubles (one needs two real-to-real transforms instead of one for ϕ).

There are two basic choices for the application of kicks:

(i) Particle-scattering methods: the particles' coordinates are scattered to the appropriate local support, the fields are sampled and the appropriate kicks applied on the owner process of the local support. The particles can either be retained by the owner process of the local support, or the kicks can be transferred back to the particle's owner process. The former solution, however, involves more data transfers (complete particle state in one direction vs. two real numbers back and forth)

(ii) Field-gathering methods: each process gathers the fields of the entire support and applies them to the particles owned by it. As this requires constant time for a given grid resolution and number of processors, this will be the more time-efficient solution beyond a certain number of particles; we choose this solution in our code.

2.4. Improvements

2.4.1. Loose coupling between rings   We can make use of an additional symmetry property of the system: as there are two rings involved, we split the processors into two subgroups, each assigned to one of the bunches. The only communication necessary between these subgroups is then the exchange of the charge density $\rho_{\bar{i}}$, which can be done after collecting it to the root process of the solver. Thus, only a single pair of communicators between processes assigned to different bunches is necessary.

The advantage of this procedure is due the hardware configuration of the computer system available to us. On the IBM SP at NERSC, 16 processors share a node and can communicate via shared memory. Communications between nodes will be over a fast network, but still be substantially slower. Thus, it is advantageous to limit the distribution of the Poisson solver, which will involve a large amount of all-to-all communications, to one node. By the bisection of the problem the communications overhead penalty will start to set in at 32 processors instead of 16 processors.

2.4.2. Parallel two-dimensional steps   Communications overhead can be further reduced by using another possible parallelization. As soon as the longitudinal slice number $n_z > 1$, each encounter will involve the independent encounters of several slices. Particle deposition and solving the Poisson equation can then be done in parallel, making it possible to keep both local to a single node by setting the number of processors in a solver to $n_{sp} < n_p/2$. However, not every encounter step involves the encounter of an integer multiple of n_{sp}, so ... solvers will have been idling during one encounter. The optimum choice for n_{sp} depends on the hardware setup and the ratio of CPU time usage for solving the Poisson equation and particle-grid-dynamics, resp., so it has to be found experimentally for each given number of particles and grid size.

Tables 2.4.2 and 2.4.2 show typical computation times for two different setups; the column is the number of processors used for a single poisson solver. (e.g, 32 processors and 4 processors/solver would handle 32/2/4=4 deposit/solve/kick-cycles for each bunch in parallel, leaving a maximum of 6 processors idle)

Table 1. Pool algorithm: $3.1 \cdot 10^5$ particles, $128 \times 64 \times 31$ grid, time (in s) for 5 encounters

Proc./Solver	Number of Processors						
	2	4	8	16	32	64	128
1	292.4	154.0	85.8	53.4	42.8	41.5	46.3
2	–	175.0	94.6	56.9	40.6	39.1	43.0
4	–	–	98.7	58.0	40.3	36.8	38.3
8	–	–	–	62.3	42.7	35.4	37.0
16	–	–	–	–	47.1	39.8	39.0
32	–	–	–	–	–	94.3	
64	–	–	–	–	–	–	n/a

3. Adding a Particle Manager: The Slice algorithm

While the algorithm described above allows for a great flexibility with respect to variable computer parameters such as number of processors, number of processors in a fast sub-cluster etc., its performance for a higher number of processors is somewhat disappointing.

Table 2. Pool algorithm: $2.5 \cdot 10^4$ particles, $256 \times 256 \times 5$ grid, time (in s) for 10 encounters

Proc./Solver	Number of Processors						
	2	4	8	16	32	64	128
1	94.4	58.7	41.2	39.0	39.9	41.6	43.0
2	–	56.4	36.6	27.9	26.6	28.0	29.6
4	–	–	31.4	22.8	18.6	19.3	20.4
8	–	–	–	20.6	16.5	15.2	16.0
16	–	–	–	–	15.2	14.4	14.4
32	–	–	–	–	–	16.2	16.4
64	–	–	–	–	–	–	n/a

This is, of course, due to the choice of a common "pool" of particles, shared among all processors, with no attempt at localization in physical space, making it necessary to communicate calculated fields to all processors. An obvious modification would be to assign particles of a certain, limited part of configuration space uniquely to one processor or group of processors, making field communication local to that processor or group of processors.

The obvious drawback is the need to re-distribute particles among processors, as the dynamics will move a particle from one processor's responsibility into another's. Care must be taken not to lose in particle management communication what was gained by saving field communication.

3.1. Longitudinal domain decomposition

Consequently, particles should be assigned to processors according to their longitudinal coordinate: The longitudinal dynamics in a storage ring usually is much slower than the transverse one, meaning that a relatively small number of particles will change processors during a single turn.

A complication arises from the fact that a simple equidistant slicing will lead to a very uneven distribution of particles, leaving most of the tracking work to the processors responsible for the center slices. This can be remedied by chosing a Hirata-type slicing [5], choosing borders $\zeta_i, \zeta_0 = \infty, \zeta_{z_n} = -\infty$ such that the number of particles in $[\zeta_i, \zeta_{i+1}) = N_p/n_z$; the encounter points between slices i and k are chosen at a distance $\frac{z_i - z'_k}{2}$ from the IP, where z_i, z'_k are the centers of gravity of the slices in the respective bunch. Again, the grids' resolutions are scaled according to z_i and β.

3.2. The Wraparound algorithm with idle cycles

Another problem is the fact that the processors assigned to the head of the bunch will be idle after the centers of the bunches have passed each other (Fig. 2; slices atop each other are in a deposit/solve/kick step, single slices are idle. Each slice is on one or more CPUs) .

We can, however, apply the transfer map of the lattice up to the next IP (or to the beginning of the same IP) in the first of these idle steps, and do the next collision in the next step (Fig. 4)

A complication arises from the fact that particles from the trailing slices may move into a slice when they are transported through the lattice after they have encountered their last collision partner. This can be partially cured by inserting a 'hiatus', i.e., a slice, after having been transported, waits for one or several additional idle steps, leaving CPUs unused, for
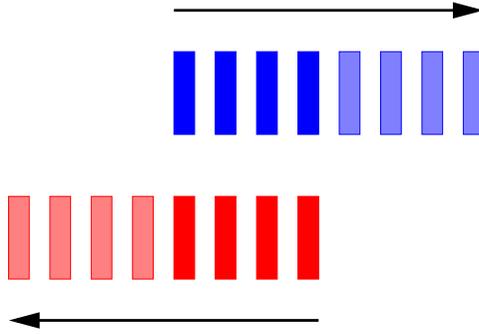
Figure 2. Idle CPUs in a longitudinal domain composition

Apply Transfer Map
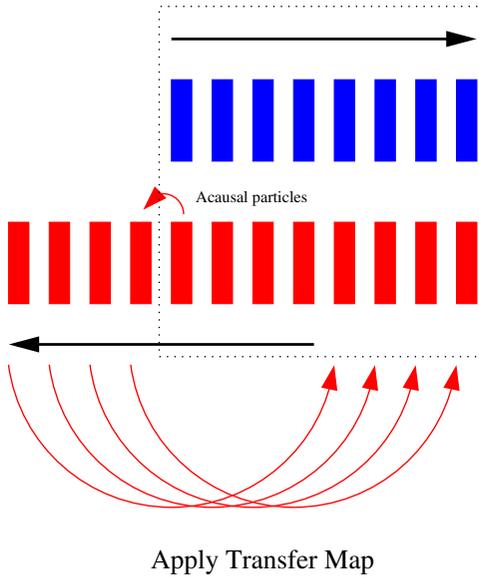
Figure 3. Wraparound algorithm for longitudinal domain composition

particles from the slices behind it to catch up. Fig. 3 shows this setup. But even then, further catch-up events are unavoidable: For any phase advance $\Delta\mu > 0$ there will be a particle with a single particle action high enough to translate the phase advance into a sufficiently high spatial advance.

Assuming a matched distribution (i. e., $\rho(p,q) = \tilde{\rho}(H(p,q))$) and a quadratic Hamiltonian, and scaling the canonical variables to $\sigma_{p,q} = 1$, the number of particles moving from a slice $q_1 < q \leq q_2$ to a slice $q_3 < q \leq q_4$ in two-dimensional phasespace during a phase advance $\Delta\phi = 2\pi\Delta\nu$ (or vice versa, as the distribution is invariant under rotations) is given by the integral over the parallelogram obtained by overlapping one slice with the other, rotated slice:

$$\Delta N = N_P \int_{q_3}^{q_4} \mathrm{d}q \int_{q_1/\sin\Delta\phi + q\cot\Delta\phi}^{q_2/\sin\Delta\phi + q\cot\Delta\phi} \mathrm{d}p\rho(H(p,q)) \quad . \tag{2}$$

Apply Transfer Map
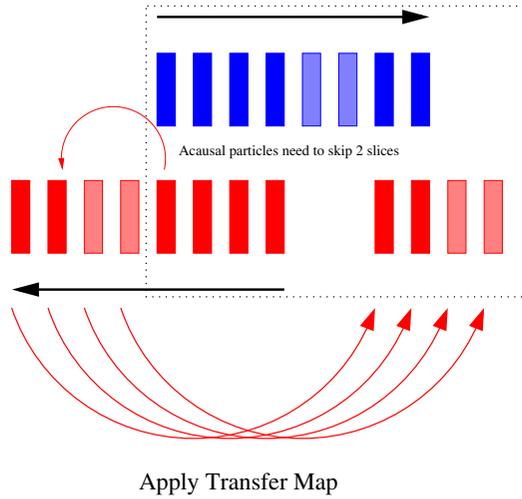
Figure 4. Wraparound algorithm with a hiatus of 2 for longitudinal domain composition

For a gaussian distribution, this integral has to be evaluated numerically.

The acausal leakage rate can now be calculated by use of 2, a plot of the maximum acausal leakage rate per turn vs. the number of inserted idle cycles is given in Fig. 5. The synchrotron tunes are $0.04, 0.02$, and 0.00072 (PEP II HER, PEP II LER, and Tevatron, resp.), the number of slices is 11. For the Tevatron, the leakage rate is completely benign even for just 1 idle cycle, resulting in near-optimal CPU utilization.
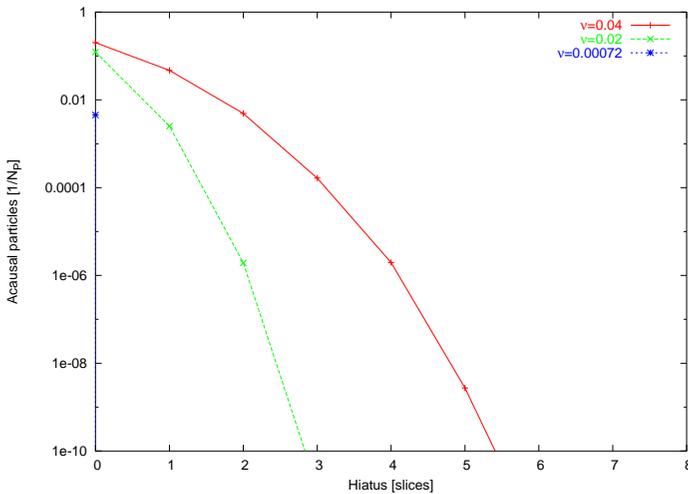
Figure 5. Acausal leakage rate for typical machines and a longitudinal decomposition into 11 slices

For the sake of clarity, we give the wraparound algorithm in some detail for a

configuration of 5 slices and a hiatus of 2. We get the following table of operations:

Table 3. Sequence of parallel operations for 5 slices and a hiatus period of 2

Slice/Step	0	1	2	3	4	5	6
0	K_4L	F_{01}	$F_{02}B_{02}$	K'_0F_{03}	K'_1F_{04}	K'_2	K'_3
1	K_3	K_4LF_{01}	$F_{02}B_{02}$	$F_{03}B_{13}$	K'_0F_{04}	K'_1	K'_2
2	K_2	K_3	$K_4LF_{02}B_{02}$	$F_{03}B_{13}$	$F_{04}B_{24}$	K'_0	K'_1
3	K_1	K_2	K_3	$K_4LF_{03}B_{13}$	$F_{04}B_{24}$	B_{34}	K'_0
4	K_0	K_1	K_2	K_3	$K_4LF_{04}B_{24}$	B_{34}	1

Here, K_i stands for a kick due to slice i of the opposing bunch, K'_i for a kick due to slice i of the opposing bunch in the next IP, L for the transport to the next IP, F_{ik} for a forward re-shuffling operation involving slices $[i,k]$ (k being the originating slice), B_{ik} for a backward re-shuffling operation involving slices $[i,k]$ (i being the originating slice).

Each K operator involves the transport to the appropriate encounter point and the (un)projection on/off the slice before/after the actual kicks step. Each L operator involves the transport into the IP. Thus, each sequence of steps expects and releases the particles in completely overlapping bunches, transported into the IP. Note that the arrangement above has related collective operations occurring in the same step number on each processor and leaves no collective operation open after the encounter.

The B operator's scope is different from the F operator: while we have to accept the occasional causality-violating particle being transported forward, we can avoid causality violations in the other direction by transporting no further backwards than to the youngest slice in hiatus. Particles not belonging there will move out when it is this slice's turn to be originator of a backwards re-shuffling operation.

The arrangement above takes into account that

(i) $B_{ii} = F_{ii} = 1$
(ii) A B operation should originate as late as possible (i. e., immediately before K'_0) to avoid having to transport particles twice
(iii) An F operation should originate as early as possible (i. e., immediately after L) to minimize causality violations.

We have test-run the code on the NERSC facility; this time, we observe an almost linear behavior of the CPU time vs. CPU number (Fig. 6). Due to the much more favorable localization of particles on CPUs as opposed to the pool algorithm, a breakdown of this behavior will not set in before the most communication-intensive process, field solving, is distributed among more than 16 CPUs. Thus, for a typical slice number of 32, we expect this point to be reached for 1024 CPUs.

3.3. Bunch setup and random number generation

The simulation codes makes use of pseudo-random numbers during the initial setup of the particles and, in the case of electrons, for the simulation of noise induced by synchrotron radiation.

It is desirable that the code's result is independent of the number of processors used; this means that not only the initial conditions should depend only on the number of particles used (and possibly on a free 'seed' parameter), but also that the history of synchrotron radiation effects should be the same for a given particle, whatever processor it is assigned to.
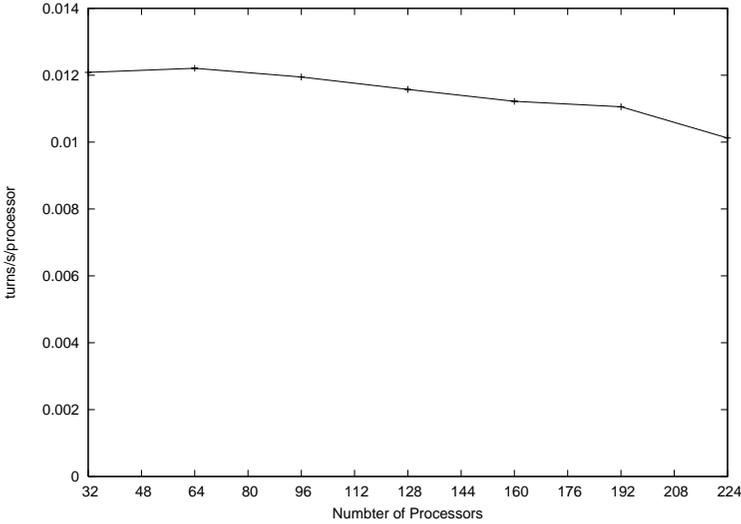
Figure 6. Scaling of a simulation run on NERSC; $n_x = n_y = 64, n_z = 16, v_s = 7.2 \cdot 10^{-4}$

In the case of the pool approach, this can be realized by assigning each particle a unique tag number $\in [0, N)$. Then, each processor uses an identical pseudo random number generator (RNG) and interleaves its output, such that particle i's phasespace coordinates are initialized with elements $x_{6i}, \ldots, x_{6i+5}$ of the random number generator's sequence. Other elements (belonging to other processors' particles) are discarded, which wastes CPU time, but makes inter-process communication between the RNGs unnecessary. A similar approach is used for the generation of synchrotron radiation noise.

This approach would not work in the slice approach, as particles are exchanged between processors, possibly making it necessary to use random number sequence elements already discarded.

A simple solution to this difficulty is to have each particle carry its own unique RNG. We use a 64-bit linear congruential generator, which generates the sequence

$$X_{k+1} = aX_k + p \mod m \quad , \tag{3}$$

where a is an integer multiplicator which can be chosen the same for each generator, $m = 2^{64}$, p is a prime number, and $x_k = 2^{-64}X_k \in [0, 1)$. The generator can be made unique by chosing p the ith prime for the ith generator. Normal-distributed numbers y_k can be generated pairwise by the prescription $y_k + iy_{k+1} = e^{2\pi i x_{k+1}} \sqrt{-2\log(1 - x_k)}$.

Each generator thus has to carry its state X (a 64-bit number) and the parameter p (for which a 32-bit number is sufficient for typical number of particles used), only slightly increasing the payload for a particle data structure.

4. Conclusion and outlook

A naive implementation of a strong-strong beam-beam solver on parallel machines lead to disappointing results due to high inter-process communications. We have developed an alternative algorithm which decouples the problem as far as possible, making use of the slow dynamics of the system in the longitudinal phase plane and the binary symmetry of

the problem. We have observed an almost linear CPU time scaling on the NERSC facility. The code is transparent with respect to CPU number.

Further improvements improvements will include:

- Nonvanishing crossing angles
- Offset beams (parasitic crossings)
- Dynamically adapting grid resolution
- Dynamic reslicing, keeping the particle number in each slice constant
- Different number of longitudinal partitions in different IPs (for handling of crossings with low phase advance: the required z resolution will be much smaller)

Acknowledgements

References

[1] Y. Cai, A. Chao, S. Tzenov, and T. Tajima. Simulation of the beam-beam effect in e^+e^- storage rings with a method of reduced region of mesh. *PRSTA*, 4:011001, 2001.

[2] Y. Cai. Simulation of beam-beam effects in e^+e^- storage rings. *SLAC-PUB-8811*, 2001.

[3] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles*. Institute of Physics Publishing, 1988.

[4] M. Frigo and S. G. Johnson. Fftw: An adaptive software architecture for the fft. In *1998 ICASSP conference proceedings*, page 1381. http://www.fftw.org.

[5] H.Moshammer K.Hirata and F.Ruggiero. *Particle Accelerators*, 40:205, 1993.

A parallel code for lifetime simulations in hadron storage rings in the presence of parasitic beam-beam interactions

Andreas C. Kabel†, Y. Cai

Stanford Linear Accelerator Center,
2575 Sand Hill Road, Menlo Park, CA 94025 ‡

Abstract. When two beams of charged particles share the same beam pipe, traveling in opposite directions, nonlinear forces are introduced due to the presence of beam-beam forces. These forces may excite incoherent resonances, eventually leading to particle loss and thus to a finite beam lifetime. Beam-beam interaction will occur as head-on collisions in the designated interaction points of the machine as well as as parasitic crossings along the circumference of the machine. In the case of protons, damping mechanisms are absent; this can lead to very long time scales for the dynamics of this system.

We developed a code which models a proton/anti proton storage ring as a sequence of symplectic transformations due to the lattice and momentum kicks due to weak-strong beam-beam interaction. The relevant information for the setup is extracted from a MAD description file of the machine and translated to speed-optimized C++ code. By distributing test particles among computers running in parallel, we are able to track them for a relatively long time: On a 64-processor cluster, and for a storage ring with 100 parasitic crossings, we are able to calculate beam lifetimes in the range of several hours. We present typical results and discuss possible applications.

1. Introduction and motivation

1.1. Beam-beam effects

Beam-beam effects play an important role in the design and operation of storage ring colliders. A close encounter of charged particle beams traveling in opposite directions will lead to strong transverse forces exerted on the respective other bunch. Depending on the ratio of the forces, one distinguishes between the weak-strong (in which one beam can be considered as unaffected by the force of the other) and strong-strong regime (in which both beams influence each other with comparably sized effects).

The beam-beam effect will lead to linear effects due to the additional focusing (or defocusing) forces near the center of the beam, resulting in tune shifts and new effective Twiss functions. Non linear components of the forces lead to amplitude dependent tune shifts, thus introducing a tune spread. Other non-linear effects include the flip-flop effect.

The most prominent occurrence of beam-beam forces is in the design interaction point (IP), in which the beams collide head-on, leading to maximum transverse forces.

Interaction points other than the design IP are referred to as 'parasitic crossings'. In a circular machine with N bunches circulating, the opposing beam will see $2N$ crossings (provided the beams share the same beam pipe)

1.2. Beam-beam effects in the Tevatron§

In its current setup, the Tevatron operates with 36×36 proton/anti-proton bunches, which share the beam pipe in a helical scheme. During the injection stage, which takes place at 150GeV, all 72 interaction points are parasitic.

During injection, a relatively fast decay of the anti-proton current is observed; the lifetimes obtained by fitting to an exponential decay behavior are of the order of 1 hour. In the absence of the proton beam, the anti-proton lifetime is well beyond 20 hours. The proton beam itself is unaffected, having a vastly higher charge. Thus, it seems justified to assume that the lifetime behavior is due to weak-strong beam-beam effects; the particle loss might be caused by diffusion processes or incoherent resonances, driving a particle to the physical aperture of the machine (which is, due to the need to open the helix wide to mitigate beam-beam effects, very narrow)

In the rest of the paper, we discuss methods and a parallel code to simulate this behavior.

2. Lifetime calculations

The beam lifetimes we would like to simulate are of the order of 1hour. For the Tevatron, this corresponds to a loss rate of $< 5.8 \cdot 10^{-9} N/$turn, where N is the number of particles in a weak bunch.

We choose a direct particle tracking approach to lifetime calculations, as opposed to the usual dynamic aperture calculations. As we are dealing with a proton machines, synchrotron radiation damping mechanisms are absent, so there is no natural limit on the number of turns one needs to track. Assuming that we need to observe 10^2 particles being lost during the tracking procedure to get decent statistics, we need to simulate $> 1.7 \cdot 10^{10}$ Particles · Turns. As we are interested in the injection stage, where we assume that all parasitic crossings will contribute to the dynamics of the beam, $> 1.2 \cdot 10^{12}$ weak-strong interactions need to be simulated. An aggravating factor for the calculation is the dependence of the beam-beam interaction pattern a bunch experiences on the longitudinal position within a train and on the cogging stage it is currently in. Thus, the computational effort necessary is clearly out of the range of single processor machines.

3. A parallel weak-strong code

Parallelization clearly is needed to tackle the problem described above. But even when parallelized, the computational demands of a proton tracking code for lifetime calculations are pushing the limits of the computing resources available to us. Thus, a carefully designed, speed optimized code is necessary; only the physics relevant to the problem should be included.

The code `PlibB` (Parallel Lifetime Calculations with beam-Beam) was written with this goal in mind. It is not an optics code; rather it assumes a machine description prepared for it using MAD8[], using a defined convention of marking the interaction points within the beamline.

Using the MAD input file, a PERL script then runs MAD several times to

- calculate Twiss parameters for the proton and anti-proton beams at the IPs
- calculate the closed orbit of both beams

§ This section is based on personal communications by N. Gelfand, V. Lebedev, P. Lebrun, T. Sen, M. Syphers, V. Shiltsev, and M. Xiao, to whom we are deeply indebted. Any inaccuracies in this section, however, should be attributed to the author.

- calculate the linear transfer matrices for the anti-proton beam between interaction points with respect to the closed orbit . This requires the undocumented trick of doing an 'empty match' on the RMATRIX of a beamline subsection. Furthermore, MAD had to be modified to output the transfer matrix with full internal precision to ensure symplecticity.

The Twiss parameters and transfer matrices are then used by the script to generate a C++ program file, containing an alternating sequence of beam-beam kick elements and linear transformations, as well as a single 'aperture' element (see below). This file is then compiled and linked into the main code.

4. Analytic treatment of the beam-beam kick

The integrated transverse kick of a gaussian bunch can be expressed by the Bassetti-Erskine Formula[1]:

$$
F_y + iF_x = \frac{Nr_{class}}{\gamma} \sqrt{\frac{2\pi}{\sigma_x^2 - \sigma_y^2}} \left[\mathrm{w}\left(\frac{x+iy}{\sqrt{2(\sigma_x^2 - \sigma_y^2)}} \right) - \exp\left(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2} \right) \mathrm{w}\left(\frac{\sigma_x^2 x + i\sigma_y^2 y}{\sigma_x \sigma_y \sqrt{2(\sigma_x^2 - \sigma_y^2)}} \right) \right] \quad (1)
$$

where $\sigma_{x,y}$ are the RMS sizes of the strong beam, and $\mathrm{w}(z)$ is the complex error function:

$$
\mathrm{w}(z) = e^{-z^2} \mathrm{erfc}(iz) \quad (2)
$$

The evaluation of w turns out to be the most time-critical component of the code. Thus, we surveyed a number of implementations, (for a comprehensive list, see [2]); the fastest we found are

- The Chiarella-Matta-Reichel algorithm ([3]). This is an exponentially converging series; stripped down to 10^{-5} precision, whree only 6 rational functions need to be summed. Its most prominent downsides, namely, bad accuracy near the origin and near the zero of the denominator of the summands can be avoided by following the prescription given in [4]: there are two pre-calculated coefficient series for slightly different poles, of which we chose the more distant one for a given argument. Also, we switch to a Padé approximation for $|z| \ll 1$
- Brute force: a pre-calculated 2-D lookup table and bilinear interpolation. This is good for 10^{-6} accuracy. The method works very well on the IBM SP, which has a large data cache; on the PC platform, the first method turns out to be faster.

The implementations were checked with a reference implementation [5]. Numerical experiments showed that a 10^{-5} relative accuracy or better everywhere is sufficient for typical simulation lengths; lifetime estimates (see below) are in excellent agreement with code using 10^{-16} accuracy implementations; test particles begin to show a visible deviation from the reference implementation not before $10^5 \ldots 10^6$ turns.

4.1. Other optimizations

Although conventional wisdom has it that hand-optimizing floating-point code is hardly worthwhile, this is not true for modern CPU architectures in which FP operations are of the

execution speed of integer arithmetic and where cache locality effects and vectorizability can have a substantial impact on the execution speed.

We concentrated our efforts on speeding up the particle transport and the beam-beam kick on the architectures available to us (INTEL Xeon processor and IBM SP). In our code, each particle carries two 6d-vectors of phasespace coordinates and a flag to switch between them. This way, load-to-store operations and storing of intermediate results can be avoided when updating coordinates. For the linear transport on the INTEL platform, we use hand-coded SSE instruction, making use of the parallelizability of FP operations on Intel. On the IBM SP, we use the array of phasespace coordinate vectors as a $6 \times N$ matrix, multiplying it by the transfer matrix using the optimized ESSL library routines. For the implementation of the Matta-Reichel complex error function, care was taken to exploit vectorizability of the inner loops (e.g., no direct summations of intermediate results, but store them in an array and sum later)

By these methods, we gained about a factor of 3 in speed over an initial version of the code. Currently, the speed achieved is 10^6 (kicks,transforms)/second on the Intel Xeon and $6.7 \cdot 10^5$ (kicks,transforms)/second on the IBM SP.

4.2. Parallelization

The Weak-Strong is embarrassingly parallelizable; a parallel code merely assigns different chunks of particles of a common particle pool to each processor. Communications between processes is only necessary when collective quantities (lost particles, beam sizes, ...) are calculated.

4.3. Simplified distributions

Usually, one would not expect particles in the core of the beam to be lost. Therefore, one can use 'de-cored' distributions where particles near the center of the distribution are left out, thus increasing the effective speed of the simulation. Different strategies are possible. Clearly, a simple cutoff in each phase space dimension is not sufficient, as it will not be a matched distribution. In action space, different cutoffs are possible , the limiting cases of which are (with an obvious notation referring to phase space coordinates normalized to unity)

- An all-dimensional cutoff, i. e., a $2d$-hypersphere of radius R cut out: $\rho(\vec{x}) \propto \Theta(|\vec{x}| - R)e^{-|\vec{x}|^2/2}\mathrm{d}(|\vec{x}|^{2d})\mathrm{d}\Omega$; the ratio of particles within that distribution is

$$N/N_0 = e^{-R^2/2} \sum_{i=0}^{d-1} \frac{R^{2i}}{2^{2i}i!}$$

- pairwise radial cutoff, i. e., the direct product of distributions with a disc of radius R cut out: $\rho(\vec{x}) \propto \prod_i \Theta(x_i^2 + p_i^2 - R^2)e^{-(x_i^2+p_i^2)/2)}\mathrm{d}^d x \mathrm{d}^d p$, here

$$N/N_0 = (1 - \exp(-R^2/2))^n$$

For $d = 3$ (six dimensional phase space), the latter choice (which we use in our code) eliminates more particles for typical choices ($R \approx 1\sigma \ldots 2\sigma$) of the cutoff.

5. Lifetime estimates

In the tracking procedure, the particles are tracked through the sequence of transfer elements. In the aperture element, the action of the particle (with respect to the Twiss parameters at the
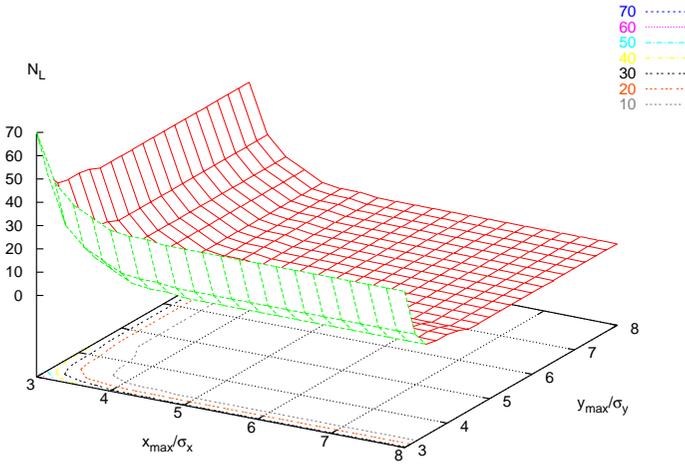
Figure 1. Typical 'waterfall' plot for $N_{Lost}(I_x, I_y; t = t_{fixed})$

aperture's position) is determined. The action is recorded in the particle's data structure if it is greater than actions recorded before. Thus, after completion of each turn, one has a record of the maximum action for each particle.

Scanning through I_x, I_y-action space, one can count the number of particles beyond a certain I_x, I_y. Assuming that this action pair represents the physical aperture of the machine, one gets a number $N_{\text{Lost}}(I_x, I_y; t)$ (where t represents the number of turns), which is a monotonously falling function with respect to I_x, I_y for fixed t and a monotonously rising function with respect to t for fixed I_x, I_y. In order not to slow down the code, this scan is typically run after each $10^3 \dots 10^4$ turns. The unstable particles are identified and their tag number printed out; in a subsequent run, the code can be run with only the unstable particles in the initial distribution; phase space dumps then are used to help; identify the instability mechanism. Looking at N_{Lost} for fixed I_x, I_y, one can observe typical lifetime patterns: a quick loss of particles (due to fast resonances or mismatching effects), going over into a linear behavior for large turn numbers. Extrapolating the linear behavior, one can estimate the lifetime, assuming a certain aperture, by looking at the slope of the linear part of N_{Lost}. An advantage of this procedure is, of course, the fact that it allows to check the connection between physical aperture and lifetime for a number of apertures with a single run.

Figures 1–3 illustrate the method for a toy example (the tevatron collision lattice, run at injection energy and with the closest parasitic crossings around the IPs included). In this example, a fifth-order resonance was the mechanism driving the particle loss.

For the real Tevatron lattice, we were, so far, unable to find lifetimes compatible with the experimental observations; simulated particle loss rates were much smaller (compatible with $> 50h$ lifetimes) than observed. Our simulations, however, display the right signature both in the temporal behavior of particle loss (a fast-loss stage and a linear asymptotic beahvior) and in the relative lifetimes of bunches in different cogging stages (the first bunch suffering most losses).
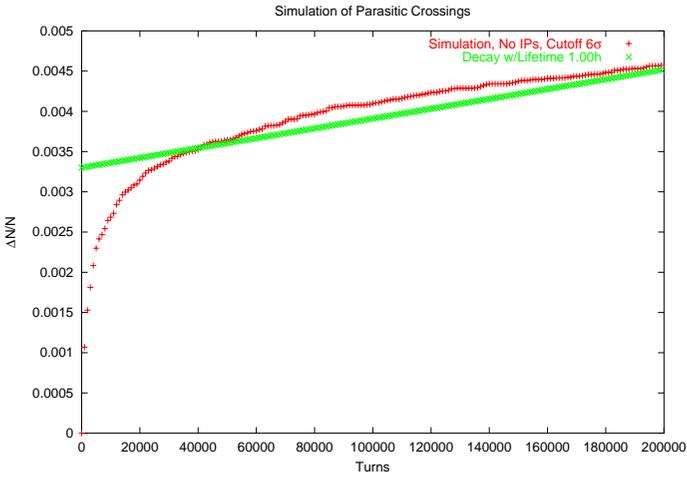
Figure 2. Typical loss plot for $N_{Lost}(I_x = I_{x,fixed}, I_y = I_{y,fixed}; t)$
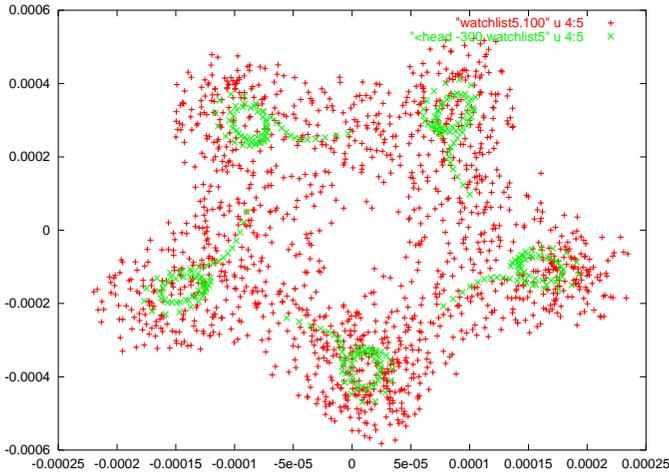
Figure 3. Poincaré section of instable particles in the toy example

This might be due to inaccurate depiction of the actual physics in our model (see discussion below) or due to our insufficient knowledge of machine parameters. Further simulations, including parameter scans, are required.

6. Conclusion and further directions

Through parallelization and optimized implementation of tracking procedures, the `PlibB` code is able to calculate finite lifetimes caused by parasitic beam-beam crossings based on the

tracking simulation of $\approx 10^{10}$ particles·turns, which, as it produces measurable quantities, can be viewed as a distinct advantage over dynamical aperture results produced by other tracking codes. While we are able to reproduce the signatures of anti-proton loss in the injection stage of the Tevatron, the actual lifetimes obtained by our code are much longer than actually observed. A possible cause of this might be the beam-physical austerity of our code, which only considers beam-beam elements and linear transport elements. In its current stage, the code misses elements of beam physics that could turn out to be important:

- Magnet nonlinearities/chromaticities: This might be the single most important physics element we are missing. The natural chromaticity of the Tevatron due to sextupole elements is extremely high (it is compensated down to 8 units, however). While we have implemented the effective global lattice chromaticity, the high local sextupole content may lead to strong non linear effects. A full implementation of this effect would lead either to a high number ($\approx 10^2 \ldots 10^3$) of non-linear kick elements (which would preserve symplecticity) or to high-order differential algebra mappings. Either method will dramatically slow down the code.

  If the full, high order map is not a viable option for speed reasons, we still can 'fudge' in chromaticity. MAD is used to calculate $\xi_i = \partial \mu_i / \partial \delta$ at all interaction points. Then, $\delta \mu = \delta(\xi_{i+1} - \xi_i)$ is the phase advance a particle with momentum deviation δ has on top of that due to the transfer matrix.

  The phase advance can be applied by sandwiching the transfer matrix between two thin pseudo-quadrupoles with appropriate strengths (independent strengths for x and y). A generalized quadrupole of δ-dependent strength results from a Hamiltonian potential $V = \kappa_x \delta x^2 + \kappa_y \delta y^2$. To preserve symplecticity, the coordinate l, conjugate to δ, needs to be kicked by $\Delta l = -2(\kappa_x x^2 + \kappa_y y^2)$.

  While this procedure certainly preserves symplecticity, we encountered unreasonably high loss rates when implementing it with the current Tevatron lattice. This might be due to the fact that we apply the kicks at fixed relative phase locations, namely the parasitic crossings; a smoother (or random) distribution of the kicks might get rid of this phenomenon.

- Finite length of beam-beam interaction: When transporting a particle to the longitudinal position of a beam-beam interaction (which might deviate from the one of orbit particles due to a longitudinal offset of the weak particle), the surrounding element is assumed to be a drift space. This is generally true for head-on collisions in the IPs, but not for parasitic crossings in the Tevatron. Given the length of the bunches, this effect might be important.

- Coupling between planes is not consistently considered. While the motion of the particles due to linear elements is fully 6-D coupled, the strong beam's transverse distribution is assumed to have its principal axes along x and y; in reality, however, it can be tilted by some degrees. This sould be fairly easy to implement; all that is required is the absorbtion of an appropriate rotation matrix into the neighboring linear transfer elements.

References

[1] G. A. Erskine M. Bassetti. *CERN-ISR-TH/80-06*, 1980.

[2] J. A. C. Weideman. *SIAM J. Numer. Anal.*, 31:1497, 1994.

[3] F. Matta and A. Reichel. *Math. Comp.*, 25:339, 1971.

[4] B. M. Gammel. http://www.matpack.de.

[5] W. Gautschi. *SIAM J. Numer. Anal.*, 7:187, 1970.

Using macroparticles with internal motion for beam dynamics simulations[*]

M. Krassilnikov, T. Weiland

FB18-TEMF, Technische Universität Darmstadt, Germany

Abstract. In order to simulate the beam dynamics in particle accelerators the Ensemble Model has been developed [1,2]. The Model divides a particle beam in a set of sub-beams or Ensembles. Besides the motion of the macroparticle center in phase space an internal motion inside Ensemble is considered. Within linear approximation for acting forces one can possible to build a compact self-consistent model based on 6 first- and 21 second order moments of the distribution function. This reduces the number of required macroparticles drastically. It was shown [3], that even with one Ensemble the simulation of the beam sizes and correlations in photo-injector yields good agreement with conventional beam dynamics codes, while the Ensemble Model has demonstrated significant advantage in computation time. In contrary to the conventional macroparticles the sizes and correlations of the Ensemble change in accordance with gradients of applied forces. The space charge model for the Single Ensemble Model (SEM) is based on the homogeneously charged ellipsoid (known as K-V distribution [4]). Such an approach, being very efficient for the SEM, has significant difficulties for application to Multi Ensemble Model (MEM). As an alternative approach the 6D Gaussian distribution has been considered, and Multi-Centered Gaussian Expansion for the space charge force calculation has been studied [5]. The V-Code, based on the Ensemble Model, has been developed for the on-line beam dynamics simulations [3,6,7].

1. Introduction

Numerical algorithms, based on macroparticles techniques, are very useful tools for the simulation of beam dynamics in particle accelerators. The beam phase space is represented by a set of macroparticles which are traced individually through a beamline, using external accelerating and focusing fields. In the case of intense and/or low energy charge particles beams not only external electromagnetic fields determine motion of the individual macroparticle but also the internal space charge fields. In order to reproduce smooth particle distribution function large number of macroparticles has to be taken into consideration. Charge assignment algorithms with smooth shape function can reduce the number of required macroparticles, but unfortunately not so drastically and they need modelling parameters, such as macroparticle size, which is fixed and under circumstances can produce some unphysical effects [8].

The Ensemble Model [1,2] represents a particle beam as a set of subbeams or Ensembles, which are described not only by position of the Ensemble center but also by different correlations in phase space. These correlations, (i.e. Ensemble rms sizes) obey time equations with corresponded Lorentz force gradient, this implies internal motion within the Ensemble. Therefore the Ensemble collective parameters (i.e. rms sizes) develop in time consistently. Superposition of the distribution function of Ensembles yields smooth beam distribution even with smaller number of macroparticles (Ensembles), so the number of required macroparticles can be reduced drastically.

2. Ensemble Model

The Ensemble Model represents a particle beam as a superposition of Ensembles. Main equations are derived from Vlasov equations for the distribution function

$$\frac{d\psi}{d\tau} = \frac{\partial \psi}{\partial \tau} + \frac{\partial \psi}{\partial \vec{r}} \frac{\vec{p}}{\gamma} + \frac{\partial \psi}{\partial \vec{p}} \frac{\vec{F}}{mc^2} = 0, \tag{2.1}$$

where $\gamma = \sqrt{1 + \vec{p} \cdot \vec{p}}$ is normalized energy, $\vec{p} = \vec{P}/mc$ is normalized momentum, $\tau = ct$ and $\vec{F}$ is applied force.

2.1. Ensemble Parameters

Using the normalizing condition for the distribution function $\int \psi(\tau, \vec{r}, \vec{p}) d\vec{r} d\vec{p} = 1$,

the first order moments are

$$\langle \xi \rangle = \int \xi \, \psi(\tau, \vec{r}, \vec{p}) \, d\vec{r} d\vec{p}; \quad \langle p_v \rangle = \int p_v \, \psi(\tau, \vec{r}, \vec{p}) \, d\vec{r} d\vec{p} \tag{2.2}$$

where $\xi, v = \{x, y, z\}$. The second order moments are given by

$$M_{\xi v} = \langle \Delta \xi \cdot \Delta v \rangle = \int \Delta \xi \cdot \Delta v \, \psi(\tau, \vec{r}, \vec{p}) d\vec{r} d\vec{p},$$

$$M_{\xi p_v} = \langle \Delta \xi \cdot \Delta p_v \rangle = \int \Delta \xi \cdot \Delta p_v \, \psi(\tau, \vec{r}, \vec{p}) d\vec{r} d\vec{p}, \tag{2.3}$$

$$M_{p_\xi p_v} = \langle \Delta p_\xi \cdot \Delta p_v \rangle = \int \Delta p_\xi \cdot \Delta p_v \, \psi(\tau, \vec{r}, \vec{p}) d\vec{r} d\vec{p},$$

where $\Delta \xi = \xi - \langle \xi \rangle, \Delta v = v - \langle v \rangle$.

2.2. Main Assumptions

The Ensemble Model assumes that for the applied force $\vec{F}$ and a Ensemble parameter μ

$$\langle \mu \cdot \operatorname{div}_{\vec{p}} \vec{F} \rangle = 0; \quad \mu = \{\xi, p_v, \Delta \xi \cdot \Delta v, \Delta \xi \cdot \Delta p_v, \Delta p_\xi \cdot \Delta p_v\}, \tag{2.4}$$

what is equivalent to the Ensemble phase area (normalized emittance) invariance [1]. Moreover, the Ensemble energy spread is not very large, so the inverse normalized energy can be expanded as

$$\gamma^{-1} = \gamma_m^{-1} - \gamma_m^{-3} \sum_n \left[\langle p_n \rangle \Delta p_n + \tfrac{1}{2} \left(\Delta p_n^2 - M_{p_n p_n} \right) \right], \tag{2.5}$$

where the mean energy γ_m includes energy spread $\gamma_m^2 = 1 + \sum_n \left[\langle p_n \rangle^2 + M_{p_n p_n} \right]$.

2.3. Time Equations

After averaging the Vlasov equation (2.1) gives (here Ensemble parameter μ has the same meaning as in (2.4))

$$\frac{\partial \langle \mu \rangle}{\partial \tau} = \left\langle \frac{\partial \mu}{\partial \vec{r}} \frac{\vec{p}}{\gamma} \right\rangle + \left\langle \frac{\partial \mu}{\partial \vec{p}} \frac{\vec{F}}{mc^2} \right\rangle . \tag{2.6}$$

Using Lorentz force expansion till linear terms

$$\frac{\vec{F}}{mc^2} = \mathbf{F}\left(\langle \vec{r} \rangle, \langle \vec{p} \rangle \right) + \hat{\mathbf{F}}^X \cdot \Delta \vec{r} + \hat{\mathbf{F}}^P \cdot \Delta \vec{p} \ , \tag{2.7}$$

where matrices

$$\hat{\mathbf{F}}^X_{ij} = \frac{1}{mc^2} \frac{\partial \vec{F}_i}{\partial r_j} \bigg|_{\langle \vec{r} \rangle, \langle \vec{p} \rangle} \ , \quad \hat{\mathbf{F}}^P_{ij} = \frac{1}{mc^2} \frac{\partial \vec{F}_i}{\partial p_j} \bigg|_{\langle \vec{r} \rangle, \langle \vec{p} \rangle} \ , \tag{2.8}$$

one can obtain 6 time equations for the first order moments $\langle \vec{r} \rangle, \langle \vec{p} \rangle$:

$$\frac{\partial \langle \vec{p} \rangle}{\partial \tau} = \mathbf{F}\left(\langle \vec{r} \rangle, \langle \vec{p} \rangle \right); \quad \frac{\partial \langle \vec{r} \rangle}{\partial \tau} = \hat{\mathbf{W}} \cdot \langle \vec{p} \rangle, \tag{2.9}$$

and 21 equations for the second order moments:

$$\frac{d\hat{\mathbf{M}}_{pp}}{d\tau} = \hat{\mathbf{F}}^X \cdot \hat{\mathbf{M}}_{xp} + \hat{\mathbf{F}}^P \cdot \hat{\mathbf{M}}_{pp} + \left(\hat{\mathbf{F}}^X \cdot \hat{\mathbf{M}}_{xp} + \hat{\mathbf{F}}^P \cdot \hat{\mathbf{M}}_{pp} \right)^T ,$$

$$\frac{d\hat{\mathbf{M}}_{xp}}{d\tau} = \hat{\mathbf{V}} \cdot \hat{\mathbf{M}}_{pp} + \hat{\mathbf{M}}_{xx} \cdot \left(\hat{\mathbf{F}}^X \right)^T + \hat{\mathbf{M}}_{xp} \cdot \left(\hat{\mathbf{F}}^P \right)^T , \tag{2.10}$$

$$\frac{d\hat{\mathbf{M}}_{xx}}{d\tau} = \hat{\mathbf{M}}_{xp} \cdot \hat{\mathbf{V}} + \hat{\mathbf{V}} \cdot \hat{\mathbf{M}}_{xp}^T .$$

Elements of the auxiliary matrices $\hat{\mathbf{W}}, \hat{\mathbf{V}}$, used in (2.9) and (2.10) are

$$W_{ij} = \frac{1}{\gamma_m} \left\{ \delta_{ij} - \frac{\hat{\mathbf{M}}_{p_i p_j}}{\gamma_m^2} \right\}; \quad V_{ij} = \frac{1}{\gamma_m} \left(\delta_{ij} - \frac{\langle p_i \rangle \langle p_j \rangle}{\gamma_m^2} \right). \tag{2.11}$$

3. 1D Ensemble Model: Two-Stream Instability Simulation

The simplest physical plasma microinstability is that when two equal streams of electrons flow throw each other [8]. Simplified one-dimensional Ensemble model can be applied to simulate such an instability. Equations (2.9) and (2.10) can be simplified, the Ensemble center motion obeys the equations:

$$\frac{\partial \langle p_z \rangle}{\partial \tau} = F_z \left(\langle z \rangle, \langle p_z \rangle \right); \quad \frac{\partial \langle z \rangle}{\partial \tau} = \frac{1}{\gamma_m} \left\{ 1 - \frac{M_{p_z p_z}}{\gamma_m^2} \right\} \langle p_z \rangle, \tag{3.1}$$

the second order moments time equations are

$$\frac{dM_{p_z p_z}}{d\tau} = 2 \cdot \left(F_{zz}^X \cdot M_{zp_z} + F_{zz}^P \cdot M_{p_z p_z} \right); \quad \frac{dM_{zz}}{d\tau} = \frac{2}{\gamma_m} \left(1 - \frac{\langle p_z \rangle^2}{\gamma_m^2} \right) \cdot M_{zp_z}$$

$$\frac{dM_{zp_z}}{d\tau} = \frac{1}{\gamma_m} \left(1 - \frac{\langle p_z \rangle^2}{\gamma_m^2} \right) \cdot M_{p_z p_z} + F_{zz}^X \cdot M_{zz} + F_{zz}^P \cdot M_{zp_z} . \tag{3.2}$$

For the nonrelativistic case $p_z \ll v/c$, the electrostatic approximation for the field calculation can be applied $F_z = -\partial \Phi / \partial z$, $F_{zz}^X = -\partial^2 \Phi / \partial z^2$, $F_{zz}^P = 0$.

Parameters were used $\mu = \omega_p \cdot DT = 0.25$; $v_0 / v_T = 25$; $v = \lambda_D / H = 0.4$; $L/H = 64$. Here ω_p is frequency of beam plasma oscillations, initial beams have Maxwellian distribution

with mean velocities $\pm v_0$ and thermal velocity v_T, $\lambda_D = v_T / \omega_p$ is the Debye length. Numerical parameters are time step DT and cell width H. The system has length L, more appropriate boundary conditions for the present example are periodic conditions. Simulations were done with 10 Ensembles per cell (640 total number), initial Ensemble sizes were about cell size, velocity spreads were chosen in order to satisfy Maxwellian distribution, no initial correlations M_{zp_z} have been introduced.

Poisson equation has been solved for the field calculations. The Ensemble Model makes possible to simulate smooth charge density distribution even with small number of macroparticles in calculation cell. By choosing Gaussian charge distribution function with time dependent parameters $\langle z \rangle(\tau), M_{zz}(\tau)$, a contribution of an Ensemble to grid charge density can be expressed in terms of error function.

The phase space plots for different times are shown in Figures 1-2a). The interaction of the beams causes bunching of the macroparticles, leading to potential wells which further enhance the bunching. Figures 1-2b) illustrate an evolution of the field and field gradient, which smoothness is provided by distributed macroparticles with varying sizes (Figures. 1-2c)). Field gradient affect firstly the Ensembles velocity spreads (correlated and uncorrelated), what is depicted in Figures 1-2d).
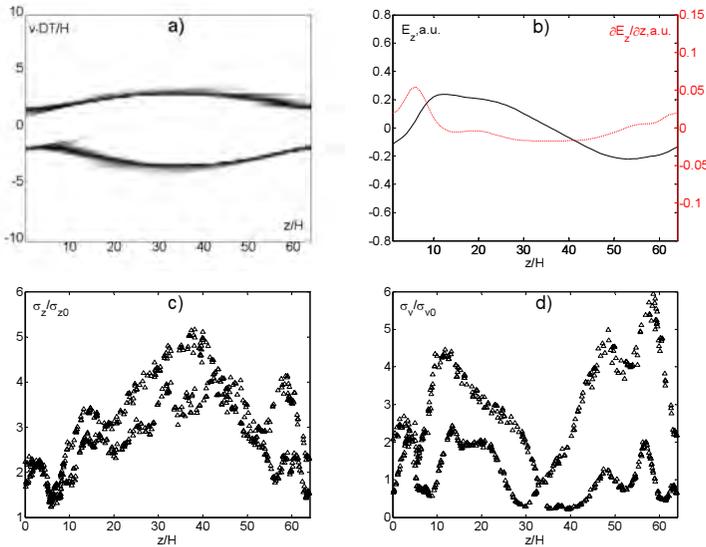
Figure 1. Two-stream instability, quasi-linear regime at time $\tau = 35 \cdot DT$. a) Phase space plot; b) Field (solid line) and field gradient (dotted red line); c) Ensembles rms sizes; d) Ensembles rms velocity spreads.

Since dimensions of an Ensemble becomes large enough (for instance, Ensemble rms size is about characteristic length of field nonlinearity), the "Ensemble multiplying" procedure can be applied. This procedure divides a "large" Ensemble in several distributed in phase space smaller Ensembles with moments satisfying the invariance of the phase area (or Ensemble emittance). In this manner nonlinear phenomena, resulting in beams heating (emittance growth), can be simulated more accurately.

It should be noted, that the main difference between the Ensemble Model and the conventional Particle-In-Cell algorithm is that the Ensemble size changes consistently with a local field gradient. The distribution function is reproduced with more accuracy. Moreover the Ensemble Model implies momentum spread within one Ensemble, whereas the conventional PIC code consider macroparticles without internal motion.
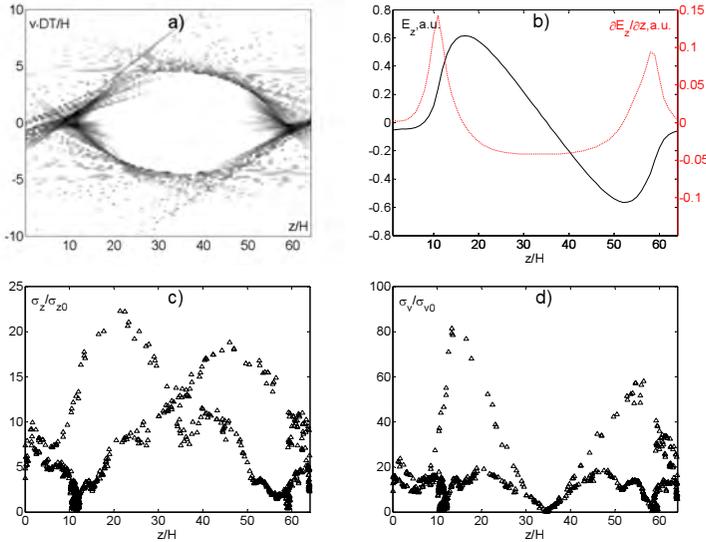
Figure 2. Two-stream instability, strongly nonlinear regime at time $\tau = 55 \cdot DT$. a) Phase space plot; b) Field (solid line) and field gradient (dotted red line); c) Ensembles rms sizes; d) Ensembles rms velocity spreads.

4. 3D Single Ensemble Model (SEM)

Since the Ensemble Model implies internal motion in macroparticle, even with one Ensemble collective effects in beam dynamics can be simulated [3]. In the case of the intense particle beams not only forces due to external electromagnetic fields act on an Ensemble but also the internal space charge forces. In this section we shall discuss application of SEM to the simulation of beam dynamics in accelerators and in particular in photoinjector.

4.1. SEM: Space Charge Implementation

The space charge implementation makes an Ensemble charge distribution function an issue. The rigorous problem consists of determining the stationary charge distribution (which does not explicitly depend on time), which corresponds to the linear applied forces. The distributions in which the forces are linear and the phase space areas remain constant is known as microcanonical distribution. A homogeneous ellipsoidal beam distribution, known as K-V distribution [4] leads to a perfect linear space charge force within the beam radius. The space charge model for the Single Ensemble Model (SEM) is based on the homogeneously charged ellipsoid. The Lorentz force between two moving charged particles is

$$\vec{F} = \frac{e}{\gamma} \left(\vec{p} \left(\vec{p} \vec{R} \right) + \vec{R} \right) \cdot \left(\left(\vec{p} \vec{R} \right)^2 + R^2 \right)^{-3/2}, \tag{4.1}$$

where $\vec{R}$ is a radius-vector between two-particles. To calculate a resulting force at small offset from the homogeneously charged ellipsoid it is necessary to integrate (4.1) over thin shell of uncompensated charges [2]. The integration yields

$$\vec{F} = \hat{G} \left(\gamma \frac{2\sigma_z}{\sigma_x + \sigma_y} \right) \cdot \frac{eQ}{\gamma^2} \cdot \frac{\vec{r} - \langle \vec{r} \rangle}{V_G}, \tag{4.2}$$

where $\vec{R} = \vec{r} - \langle \vec{r} \rangle$, σ_ξ - rms ellipsoid sizes, $V_G = (\det \hat{\mathbf{M}}_{\xi v})^{-1/2}$ is a ellipsoid geometrical volume, the geometrical factor $\hat{G}$ is

$$\hat{G}_{ij}(u) = (1 - \exp(u)) \cdot (\delta_{ij} + \delta_{iz} \cdot (\gamma/u - 1)). \tag{4.3}$$

Formula (4.2) has been implemented for the calculation of space charge force contribution to the matrices (2.8).

4.2. *Photoinjector Simulations*

RF-gun, being a remarkable object for simulations with PIC and tracking codes, has been simulated with V-Code, based on the SEM [3,6,9]. For the comparison with V-Code a PIC code MAFIA TS2 [10] and tracking code ASTRA [11] have been used. Results of comparison are shown in Fig.3.
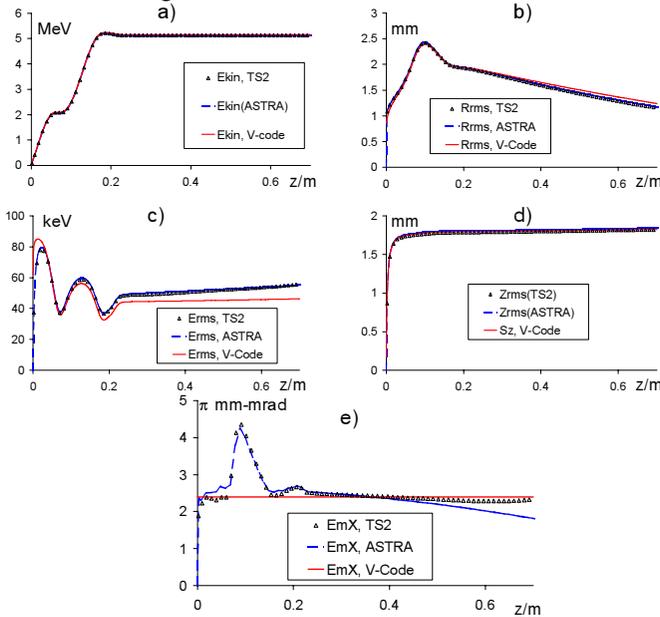
Figure 3. Simulations of beam dynamics in rf-gun using different codes: MAFIA TS2, ASTRA, V-Code. a) Beam kinetic energy; b) RMS beam radius; c) RMS beam energy spread; d) RMS bunch length; e) RMS normalized transverse emittance.

Beam parameters, including collective (such as beam sizes, energy and momentum spreads) simulated with conventional tools and V-Code (which uses one Ensemble for beam dynamics simulation) are in good agreement, whereas V-Code demonstrates significant advantage in calculation time. Such an advantage of the Ensemble Model in simulation time has been used for the practical purposes. A beam line data base of the V-Code has been designed in order to be compatible with a given accelerator control system. One of the main principle of the V-Code is capability of on-line beam dynamics simulations [6]. A dedicated Alignment Utility based on V-Code solver, has been developed and applied to the misalignment study and Beam Based Alignment of the TTF rf-gun [7].

As it was already mentioned under assumptions of the Ensemble Model, the phase area (normalized emittance) of the separate Ensemble is invariant, and Fig. 3e) illustrates the difference in emittance development. Emittance changing as well as additional energy

spread growth are caused by nonlinear space charge field and correlation within an electron beam. Because the SEM takes only linear part of applied forces, for the nonlinear effects simulations Multi Ensemble Model (MEM) has to be developed.

5. Multi Ensemble Model (MEM).

External fields can be implemented in Multi Ensemble Model (MEM) by a simple extension of the SEM approach, but the space charge implementation in MEM needs special considerations.

A model of homogeneously charged ellipsoid is very useful for the calculation of the space charge force gradient at the center of the Ensemble. But in the case of several Ensembles it is necessary to calculate not only a space charge gradient at the center of the driving Ensemble, but also Lorentz force and its gradient at positions of others Ensembles. The most probable macroparticles configuration is a set of overlapping Ensembles (the distance between two Ensembles is less than their rms sizes).

5.1. MEM: Space Charge Implementation

The problem of space charge implementation for the case of several Ensembles is to find a Lorentz force (with gradient) from driving Ensemble at the position of a test Ensemble. A smooth Ensemble charge distribution function provides the field continuity. The Gaussian distribution is the most probable candidate for it:

$$\psi(\vec{r},\vec{p}) = (2\pi)^{-3} \det \hat{\mathbf{M}}^{-1} \exp\left(-\Delta\vec{\varsigma}^{T} \cdot \hat{\mathbf{M}}^{-1} \cdot \Delta\vec{\varsigma}/2\right) \tag{5.1}$$

where $\Delta\vec{\varsigma} = \vec{\varsigma} - \langle\vec{\varsigma}\rangle$, $\vec{\varsigma} = \{\vec{r},\vec{p}\}$ and $\hat{\mathbf{M}}$ is 6×6 matrix of the second order moments.

To find a fields generated by an Ensemble with charge distribution (5.1) several methods are available.

1. Direct Integration implies straightforward calculations [12]:

$$\vec{E}(\vec{r},t) = -\frac{\partial\varphi}{\partial\vec{r}} - \frac{1}{c}\frac{\partial\vec{A}}{\partial t}, \quad \vec{B}(\vec{r},t) = \operatorname{rot}\vec{A},$$

$$\left[\varphi,\vec{A}\right](\vec{r},t) = \int d\vec{r}' dt' \left[1,\vec{\beta}(t')\right] \cdot \frac{Q\psi(\vec{r}',\vec{p}',t')}{|\vec{r}-\vec{r}'|} \delta\left(t-t'-\frac{|\vec{r}-\vec{r}'|}{c}\right). \tag{5.2}$$

This approach includes many physical effects (such as a synchrotron radiation), but multidimensional integration and singularity makes it slow and complicated.

2. Using Poisson Solver with Lorentz transformations assumes no momentum spreads are taken into account. After coordinates transformation (which corresponds to the matrix $\hat{\mathbf{M}}$ diagonalization) and integration (5.1) in pulse, we have for charge distribution function:

$$\psi(\tilde{x},\tilde{\sigma}_x,\tilde{y},\tilde{\sigma}_y,\tilde{z},\tilde{\sigma}_z) = \prod_{i=x}^{z} \psi_G(\Delta\tilde{\xi}_i,\tilde{\sigma}_{\xi_i}), \tag{5.3}$$

where $\psi_G(u_i,\sigma) = \exp\left(-u^2/2\sigma^2\right)/\sqrt{2\pi\sigma^2}$ is one dimensional Gaussian distribution. In the Ensemble rest system the Poisson equation has to be solved:

$$\tilde{\Delta}\varphi = -4\pi Q \cdot \psi(\tilde{x},\tilde{\sigma}_x,\tilde{y},\tilde{\sigma}_y,\tilde{z},\tilde{\sigma}_z). \tag{5.4}$$

In general case the z axis does not coincide with direction of vector $\langle\vec{\beta}\rangle = \langle\vec{p}\rangle/\sqrt{1+\langle\vec{p}\rangle\cdot\langle\vec{p}\rangle}$, so the coordinates transformation can be performed as

$$\tilde{\vec{r}} = \hat{\mathbf{T}} \cdot \left(\vec{r} - \langle\vec{r}\rangle\right), \tag{5.5}$$

where $\hat{\mathbf{T}}$ is a transformation matrix:

$$\hat{\mathbf{T}} = \hat{\mathbf{\Phi}}_M \cdot \hat{\mathbf{\Gamma}} \cdot \hat{\mathbf{\Phi}}_\beta; \quad \hat{\mathbf{\Phi}}_\beta \cdot \langle \vec{\beta} \rangle = \left\{ 0, 0, \left| \langle \vec{\beta} \rangle \right| \right\}^T; \quad \hat{\mathbf{\Gamma}}_{ij} = \delta_{ij} \cdot [1 - \langle \vec{\beta} \rangle \cdot \langle \vec{\beta} \rangle \delta_{iz}]^{-1/2}. \tag{5.6}$$

Matrix $\hat{\mathbf{\Phi}}_M$ has to be found from the diagonalization of the matrix $\hat{\mathbf{M}}^G$:

$$\left[\hat{\mathbf{\Phi}}_M^T \cdot \hat{\mathbf{M}}^G \cdot \hat{\mathbf{\Phi}}_M \right]_{ij} = \tilde{\sigma}_i^2 \delta_{ij} \tag{5.7}$$

The equation (5.4) with boundary conditions (exact or asymptotic), can be solved using grid based methods, which give solution at all points of discretization, what significantly exceeds needs of the Ensemble Model. Moreover most of these methods are valid for any arbitrary right part of the Poisson equation, so the regularity of the Gaussian distribution is not used.

3. One of the algorithms, based on distribution function expansion is <u>Multi-Centered Gaussian Expansion</u> (MCGE). Various types of expansions in series of Chebyshev-Hermite polynomials currently are used in astrophysics for weakly non-normal distributions [13]. From the other side, a multi-centered Hermite expansion basis is successfully used for more accurate Vlasov-Poisson simulations [14]. Multi-Centered Gaussian Expansion (MCGE) is based on the expansion of the Ensemble charge density in distributed basis functions with known solutions of the field equation.

5.2. Multi-Centered Gaussian Expansion (MCGE)

For symmetrical basis function

$$\psi_{nml} = \psi_G(x - x_n, \sigma_0) \psi_G(y - y_m, \sigma_0) \psi_G(z - z_l, \sigma_0) \tag{5.8}$$

electric field can be calculated analytically:

$$\vec{E}_{nml}(x, y, z) = \frac{q_{nml}}{\sigma_0^2} \cdot S\left[\frac{|\vec{r}_{nml}|}{\sqrt{2}\sigma_0} \right] \cdot \frac{\vec{r}_{nml}}{|\vec{r}_{nml}|} \tag{5.9}$$

where $S(x) = x^{-2} \cdot \left\{ \mathrm{erf}(x) - 2/\sqrt{\pi} \cdot x \cdot \exp(-x^2) \right\}$ and $\vec{r}_{nml} = \{x - x_n, y - y_m, z - z_l\}$ is a radius-vector from the basis function ψ_{nml} center to the observation point, q_{nml} is a weight part of the Ensemble charge.

Exact distribution function (5.3) is approximated by a series

$$\psi_{ex} \approx \psi_{appr} = \sum_{n,m,l} w_{nml} \cdot \psi_{nml}, \tag{5.10}$$

where $w_{nml} = Q / q_{nml}$. Unfortunately the basis (5.8) is not orthogonal, but factorization of the weights $w_{nml} = w_n^x \cdot w_m^y \cdot w_l^z$ reduces three dimensional weights problem to three independent one dimensional ($\xi = x, y, z$):

$$\sum_j w_j^\xi \cdot \exp\left(-\frac{(\xi - \xi_j)^2}{2\sigma_0^2} \right) \approx \frac{\sigma_0}{\sigma_\xi} \cdot \exp\left(-\frac{\xi^2}{2\sigma_\xi^2} \right) \tag{5.11}$$

Standard procedure yields a system of linear equations for weights w_j^ξ, the matrix elements can be calculated analytically.

The main approximation parameters are: 1) α is normalized distance between two basis functions ($\xi_j = \alpha j \sigma_0$); 2) $K_\xi = \alpha \sigma_0 N_\xi / \sigma_\xi$ is asymptotic parameter, determined by N_ξ - a number of terms in sum (5.10) for truncation; 3) σ_0 is rms basis function size. By choosing $\sigma_0 = \min \sigma_\xi = \sigma_\nu$ we obtain $N_\nu = 0$ and $w_0^\nu = 1$. Centers of the basis functions are located at the plane $\xi_\nu = 0$. If all three dimensions of the Ensemble are significantly

different triangular truncation in 2D sums can be used to reduce computation time. An example of Ensemble distribution function approximation is shown in Fig. 4. A proper approximation parameter set can be found by parameter scan, the optimal parameters yielded a discrepancy less than 0.1% are $\alpha \quad 1 \div 1.5$, $K_\zeta > 3$.

Using MCGE a space charge field at the point of interest (x, y, z) can be calculated by summing up of (5.9):

$$\vec{E}(x, y, z) = \sum_{nml} \vec{E}_{nml}(x, y, z), \tag{5.12}$$

moreover a sum with index equal ν is degenerated, so it can be simply omitted. Other advantage of the MCGE is the capability of analytical expressions for derivatives of the space charge field, what is necessary for matrices (2.8) computation.
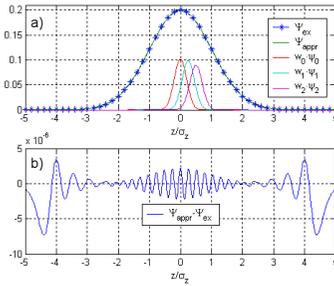
Figure 4. Distribution function approximation by multi-centered Gaussian basis functions. Approximation parameters: $\alpha = 1.25$, $K_z = 4.0$ $(N_z = 16)$, $\zeta_z = \sigma_z / \sigma_0 = 5.0$. a) Exact and approximated distribution function and first three weighted basis functions. b) Approximation discrepancy.

5.3. *MCGE Calculations of Space Charge Fields of a Single Ensemble*

Electric field of a round $(\sigma_x = \sigma_y = \sigma_\perp)$ Gaussian distribution calculated with conventional Poisson solver (solid lines) and using MCGE (markers) is shown in Fig. 5a,b for different $\zeta = \sigma_z / \sigma_\perp$ values. MCGE gives formulas for electric field:

$$E_r(r, z = 0) = \frac{eQ}{\sigma_r^2} \sum_{n=-N_z}^{N_z} w_n^z \cdot S\left[\frac{\sqrt{r^2 + \alpha^2 n^2 \sigma_r^2 / 2}}{\sigma_r}\right] \cdot \frac{r}{\sqrt{r^2 + \alpha^2 n^2 \sigma_r^2 / 2}};$$

$$E_z(r = 0, z) = \frac{eQ}{\sigma_r^2} \sum_{n=-N_z}^{N_z} w_n^z \cdot S\left[\frac{\gamma z}{\sigma_r} - \frac{\alpha n}{\sqrt{2}}\right]. \tag{5.13}$$
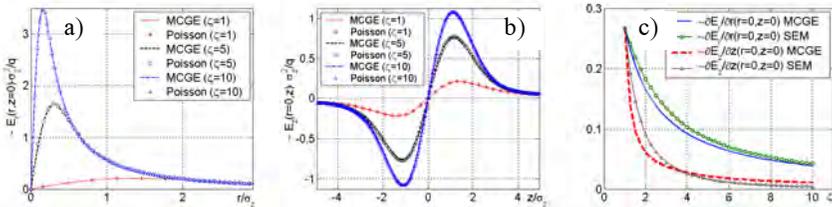
Figure 5. Radial (a) and longitudinal (b) electric field and field gradient (c) of round Gaussian Ensemble, calculated using MCGE in comparison with conventional Poisson solver.

The results are in very good agreement, 100×100 mesh has been used for Poisson solver, whereas the MCGE for the fields calculation treats $N_z(\zeta) = 1 \div 30$ terms in sum (5.12) (for the round beam two sums in (5.12) are degenerated). Field gradients $(\partial E_r / \partial r; \partial E_z / \partial z)$ at the driving Ensemble center are calculated using Single Ensemble space charge analytical model (4.2) and MCGE are shown in Fig. 5c).

Field gradient calculated using analytical approximation (4.2) is

$$\left. \frac{\partial E_r}{\partial r} \right|_{\substack{r=0 \\ z=0}} = \left(1 - \exp\left[\gamma \frac{\sqrt{2}\sigma_z}{\sigma_r} \right] \right) \cdot \frac{eQ}{\gamma^2 \sigma_r^2 \sigma_z}; \quad \left. \frac{\partial E_z}{\partial z} \right|_{\substack{r=0 \\ z=0}} = \left(1 - \exp\left[\gamma \frac{\sqrt{2}\sigma_z}{\sigma_r} \right] \right) \cdot \frac{eQ}{\gamma^2 \sigma_z^2 \sigma_r}, \qquad (5.14)$$

whereas the correspondent expressions calculated within MCGE are

$$\left. \frac{\partial E_r}{\partial r} \right|_{\substack{r=0 \\ z=0}} = \frac{eQ}{\sigma_r^3} \cdot \sum_{n=-N_z}^{N_z} \frac{2w_n^z}{\alpha^3 n^2} \cdot \left\{ \sqrt{2}\, \mathrm{erf}\left[\frac{\alpha n}{\sqrt{2}} \right] - 2\alpha n \exp\left[-\frac{\alpha^2 n^2}{2} \right] \right\};$$

$$\left. \frac{\partial E_z}{\partial z} \right|_{\substack{r=0 \\ z=0}} = \frac{eQ}{\sigma_r^3} \cdot \sum_{n=-N_z}^{N_z} \frac{4\gamma w_n^z}{\alpha^3 n^2} \cdot \left\{ 2\alpha n \left(2 + \alpha^2 n^2 \right) \exp\left[-\frac{\alpha^2 n^2}{2} \right] - \sqrt{2}\, \mathrm{erf}\left[\frac{\alpha n}{\sqrt{2}} \right] \right\}. \qquad (5.15)$$

The analytical model (4.2) demands less calculations, but it can be applied for the calculation space charge field gradient only at the driving Ensemble center, the MCGE is capable of gradient calculation at any point. The MCGE algorithm demonstrates good approximation properties, moreover linear momentum spread in Ensemble can be simulated by introducing of corresponding momentum to a Gaussian basis function.

6. Conclusions

The Ensemble Model, being alternative to conventional macroparticles models, has been developed for the large scale and fast beam dynamics simulations. In contrary to conventional macroparticle algorithms the Ensemble Model implies an internal motion within a macroparticle, the Ensemble correlations in phase space change in time in consistency with local Lorentz force gradient.

Nonlinear effects can be simulated using Multi Ensemble Model. The beam charge induced field calculation is an issues of the Model, several approaches can be applied. The Multi-Centered Gaussian Expansion demonstrates good agreement with conventional space charge routines, whereas the MCGE has advantages in implementation as a space charge routine into MEM.

References

[1]   A. Novokhatski and T. Weiland, PAC'99, New York, March 1999
[2]   A. Novokhatski and T. Weiland, ICAP'98, Monterey, Sept. 1998.
[3]   M. Krassilnkov et al., ICAP 2000, Darmstadt, Sept. 2000
[4]   I.M. Kapchinskiy "Theory of Resonance Linear Acceler.", Harw.Ac.Pub., 1985.
[5]   M. Krassilnikov, T. Weiland, LINAC'02, Gyeongju, August 2002.
[6]   A. Novokhatski et al., LINAC 2000, Monterey, August 2000.
[7]   R. Cee et al., PAC 2001, Chicago 2001.
[8]   Birdsall, Langdon. Plasma Physics via Computer Simulation. McGr.-Hill, 1985
[9]   S. Setzer et al., PAC 2001, Chicago 2001.
[10]   CST GmbH, Bad Nauheimer Str. 19, 64289 Darmstadt, Germany
[11]   ASTRA User Manual, http://www.desy.de/~mpyflo/Astra_documentation/
[12]   J.D. Jackson, Classical Electrodynamics, Willey, N.Y.
[13]   S. Blinnikov, R. Moessner, Astron. Astrophys. Suppl. Ser. 130, 193-205 (1998).
[14]   J.W. Schumer, J. P. Holloway, DPP 38th Ann. Meet. 11-15 Nov. 1996, Denver.

The software ModeRTL for simulation of radiation processes in electron beam technologies

V. T. Lazurik, V. M. Lazurik, G. F. Popov, Yu V. Rogov

Kharkiv National University, Kharkiv, P.O.Box 60, 61052,Ukraine,

Abstract. The software ModeRTL (Modeling of Radiation-Technological Line) was developed by authors for simulation of radiation processes, that used as an irradiation source of the pulsed or continuous type of electron accelerators with electron energy in the range from 0.1 to 20 MeV. The ModeRTL is intended for: a choice of optimum layout of the radiation equipment at realization of specific radiation-technological process; an optimization of parameters of radiation treatment of products in view of features of radiation-technological process; a validation of the chosen parameters. A physical basis, mathematical aspects, and operation of the software ModeRTL in some radiation technologies will be discussed in this paper.

1. Introduction

At present the electron beam (EB) and X-ray (bremsstrahlung) processing based on electron accelerators are widely used in different industrial radiation technologies, such as sterilization of medical devices, in particular, for mail sterilization; food irradiation; advanced composites modification; wire and cable cross-linking; bulk polymer modification; polymerization of monomers and grafting on monomer onto polymers; tire and rubber pre-cure treatment; purification of water and gas wasters, and others.

An absorbed dose of electron and X-ray beams within the irradiated product is one of the most important characteristic for all radiation-technological processes. Irradiator operating regimes for execution of a radiation-technological process depend on a large number of factors to be specific for this process [1]. These parameters can be obtained both by experimental methods and at use of computer models of process and special computational methods. Experimental determination of these parameters, in particular, the absorbed dose distribution of the X-rays and electrons within of the heterogeneous materials demands considerable efforts and much time. The computer simulation and calculation of process of electron/gamma–matter interaction permit solving these problems accounting of both a radiation process and a processing system specificity at much lesser time and expenditure of labour.

Necessity of further development of methods of dosimetric support is connected, first of all, with continual diversification of objects under radiation treatment, and with sophistication of irradiation modes. Besides, the conventional methods of dosimetry within the irradiated materials do not ensure the data gaining accuracy to be necessary for modern radiation technologies in the case of heterogeneous systems irradiation. Lastly, there are no methods permitting to solve effectively the problems of optimum choice of irradiation modes for the vast list of practically important cases.

Now there are a few powerful universal packages such as ITS [2], EGS [3],

GEANT [4], PENELOPE [5] for simulation of electron and photon transport through complex multi-element constructions. These packages permit obtaining by Monte Carlo method numerical data sets to solve practical problems.

These packages are just the sets of program blocks accelerating notable coding of Monte Carlo methods for simulation of transport of ionizing radiation through construction. To realize a computer experiment, that one could interpret as the physical one, the scientific elaboration of the physical experiment model is necessary, as well as model error evaluation of results of computer simulation.

The analysis of the artifacts appearance under the use of these packages shows that any universal computer package, even the most reliable and thoroughly tried one, such as ITS, EGS or GEANT, can not be used without continual detail expert control of obtained results. Therefore, for simulation of irradiation processes with use of a complex of equipment, a detail theoretical investigation is necessary. In particular, one needs construct models reflecting adequately variety of effects and phenomena observing by passing and interacting of ionizing radiation with all elements of this complex.

Note that, due to specific features of Monte Carlo method, data to be obtained on the basis of process simulation do not permit applying to them standard mathematical devices and methods of search of optimum parameters of irradiation process. It decreases largely the practical worth of the method for solving of problems of radiation technology. In this connection, development of semiempirical models of phenomena and corresponding them analytical formulas for well-grounded and physically correct handling of simulation results have a special interest.

Authors for solving above mentioned problems have developed the software tools kit that is intended for a simulation of irradiation process on the radiation-technological lines using scanning electron and/or X-ray (bremsstrahlung) beams. This tools kit includes the following programs the EMID, SIRTEL, ModeRTL, RT-Soft, RadCad, X-ray-Soft [1, 6].

The main feature of these programs is the use of hybrid scheme of calculation using the formulas of analytical models and simulation of transport of electron and gamma radiation by a Monte-Carlo method.

The architecture and operation of the software ModeRTL (Modeling of Radiation-Technological Line) which was developed by authors for simulation of radiation processes, that used as an irradiation source of the pulsed or continuous type of scanning electron beams will be discussed in detail in this report.

2. The software ModeRTL consideration

The software ModeRTL is a complex of physical and mathematical methods included in an uniform program shell. This software uses the Monte Carlo simulation and calculations based on analytical model of irradiation processes, gives an opportunity to compare calculated results with each other and to compare distributions of absorbed dose of electrons in different materials with the world-wide data base. It provides high verification of calculated results and decisions accepted on this base.

The software ModeRTL ensures a fast, scientifically-justified choice of parameters and modes of operations of a radiation-technological line for optimization of irradiation process by scanning electron beams of the products of radiation technologies. The choice is carried out in view of features of the used equipment of the RTL, configuration of the irradiated product and requirements of technology to conditions of irradiation. It is reached due to a combination in one software of a set of various modern methods:

• Hybrid scheme of calculation using the formulas of analytical models and simulation of transport of electron and gamma radiation by Monte-Carlo method.
• Problem-oriented interface supplied with specially developed system of the expert control of the input information.
• Pliable scheme of the assignment conformed geometrical and physical models for each functional element of simulated system.
• Expert control of the input and output information for prevention of occurrence of artefacts and problem-oriented cognitive visualization of calculation results.
• Special theoretical models of process and analytical relations for handling and representation of the statistical information obtained by a Monte-Carlo method.
• Methods of the solution of optimization problems for multiparameter system at presence of implicit conditions and restrictions on tolerance region of parameters.

The software ModeRTL is a program shell containing five various functional modules and technology processing data base. The interaction between functional modules and data bas is carried out by means of a set of service blocks. The software architecture is schematically shown in Figure1.
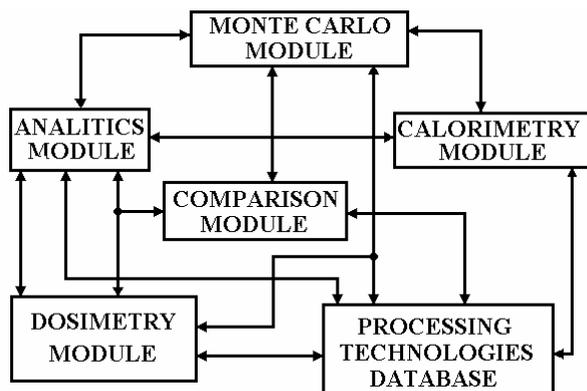
Figure 1. A schematic view of the software ModeRTL architecture.

The program shell contains the following functional modules:

The Analytics module - implements semiempirical models for calculation of a spatial distribution of a dose and integral characteristics of action of scanning electron beam on irradiated objects.

The Monte Carlo module - implements methods of statistic trials for calculation of a spatial dose distribution and integrated characteristics of action of scanning electron beam on irradiated product.

The Comparison module - implements methods of mathematical physics for handling and comparative analysis of calculation results obtained in modules Analytics (deterministic data) and Monte Carlo (statistic data). This module ensures procedure of a choice of optimum modes of irradiation and estimation of reliability of the accepted solutions on the basis of comparison of calculation data for various parameters of the radiation equipment and at use of various models for realization of calculations.

The Calorimetry module - implements numerical methods of evaluation of spatial distribution of radiation-induced temperature and analytical estimations of integral characteristics of a heat transmission for process of cooling of the irradiated products in a thermostable environment.

The Dosimetry module - implements methods of mathematical physics for entering,

processing and comparative analysis of experimental dosimetry data with calculation results, and with the world-wide data base.

The processing technologies database consists of a user guide, an archive database, and a dynamic database.

The user guide contains the detailed description, the rules and instruction for users - "how to get results", which allows them successfully to work with all modulus of the software.

The archive database stores geometrical and operational characteristics of all construction elements of RTL and their mutual layout, detailed description and characteristics of the current radiation-technological process, the optimum parameters of electron beam, the peculiarities and limitations for radiation treatment parameters of the current radiation-technological process, the calibration data for monitoring equipment, the parameters of irradiated materials, the material and size of the package for irradiated product. The archive database stores table data from the world-wide data base for comparison theirs with a calculated absorbed dose data and with an experimental data [7].

The dynamic database stores input files with monitoring data received in the course of current radiation-technological process from beam monitor system, a scanner control system, and a conveyor line system; output files with reconstructed data of critical process parameters which are used for determining of an operation value of the absorbed dose within irradiated product; data related with permissible level of uncertainties and deviations between operation and limited values of absorbed dose for irradiated product.

The ModeRTL program shell ensures parallel operation of the basic functional modules, that enables essentially to reduce a latency period of results gained by a Monte Carlo method during a choice of optimum modes of an irradiation and estimation of reliability of the accepted solutions.

3. Physical basis and mathematical aspects of the software ModeRTL

The program ModeRTL uses a combination of two methods for calculation of dose field in an object irradiated by electrons: the formulas of analytical models and simulation of transport of electron and gamma radiation by a Monte-Carlo method.

Analytic calculation. The principles.

It is supposed, that the process of scanning ensures a quite high uniformity of the electrons flux on all surface of the target. It is achieved in a case, when the scan frequency f, the target driving velocity V and effective beam diameter d satisfy the following condition $V << d \cdot f$ and the width of a scanning zone W_S exceeds the width of the target W_T on magnitude, more than on $2d$ i.e. $W_S > W_T + 2d$. Within the framework of this supposition, the spatial dose distribution in the target does not depend on scan frequency and beam diameter. Besides, the model does not take into account boundary and edge effects at description of the spatial dose distribution.

The semiempirical model.

In semiempirical model, the analytical relations for dependence of dose $D(x,E)$ on depth x in a semi-infinite target uniformly irradiated with the normally incident beam of monoenergetic electrons with energy E are used [8]. The computational scheme for an one-dimensional spatial dose distribution $D(x,E)$ was implemented in the program EMID [6]. The comparison of results of analytical calculations with results of simulation by the Monte Carlo method by the program ITS shows a small data discrepancy (less than 3 %) for energies of electrons E from 0.1 МэВ up to 20 МэВ and materials of a target with the atomic numbers Z from 4 up to 92 [6]. The

generalization of one-dimensional model is carried out in the supposition of a smallness of an electron cross deviation in comparison with its path length in substance.

The analytical expressions for calculation of two-dimensional spatial dose distribution can be submitted in a form

$$D(x, y, E) = \frac{10^3 I}{W_S V} \frac{W_T \cdot K_S(y)}{(W_T + 2x \cdot tg\theta_M)} \frac{D(x/\cos\theta, E)}{\cos\theta} \, , \tag{1}$$

$$\cos\theta = \sqrt{\frac{W_T^2}{W_T^2 + 4y^2 tg^2\theta_M}} \, ,$$

where I is the beam current (μA), V - the conveyor velocity (cm/sec), W_S - the width of scanning (cm), W_T - the target width (cm), θ_M - the maximum angle of incidence of the beam on the target (radian), $K_S(y)$ - the- normalized electron flux intensity on target surface ($x = 0$) in point y, defined by the mode of operations of a scanning system.

As it is supposed in the formula (1), the target is located in area $0 \le x < \infty$ and scanning is executed along axis Y symmetrically in relation to a point $y = 0$ within interval $-\frac{W_S}{2} < y < \frac{W_S}{2}$.

At the sawtooth shape of a time dependence of a current in deflecting magnets of the scanner and small angles of the beam deflection, the function $K_S(y)$ looks like:

$$K_S(y) = \frac{W_T W_S \cdot tg\theta_M}{W_T^2 + 4y^2 tg^2\theta_M} \left(arctg\left(\frac{W_S \cdot tg\theta_M}{W_T}\right) \right)^{-1}. \tag{2}$$

The absorbed dose within the irradiated target with account the cover box thickness h, the energy spectrum ΔE and angular $\Delta\theta$ distributions of electrons in the beam $D(x, y, E, h, \Delta E, \Delta\theta)$ is calculated by formula:

$$D(x, y, E, h, \Delta E, \Delta\theta) = \frac{1}{4\Delta E\Delta\theta} \int_{E-\Delta E}^{E+\Delta E} dE' \int_{\theta-\Delta\theta}^{\theta+\Delta\theta} D(x, y', E', h)d\theta' \, , \tag{3}$$

where $x_h = h\rho_s/\rho$, ρ_s, ρ - density of substance of object and cover, correspondingly, the variables y, y' are related to variables θ, θ' by equation (1).

The atomic number Z^* and the atomic weight A^* to be used for compounds and mixtures are given by the following formulas:

$$Z^* = \left(\sum_{i=1}^{M} \frac{w_i Z_i^2}{A_i} \right) / \left(\sum_{i=1}^{M} \frac{Z_i}{A_i} \right), \qquad A^* = \left(\sum_{i=1}^{M} \frac{w_i Z_i^2}{A_i} \right) / \left(\sum_{i=1}^{M} \frac{Z_i}{A_i} \right)^2, \tag{4}$$

where w_i denotes the fraction by weight of the i-th constituent element with atomic number Z_i and atomic weight A_i.

Monte Carlo simulation. Physical model.
This is a few decades as the Monte Carlo method applies successfully to calculate spatial electron dose distributions in various objects, therefore physical models of calculation are well known today. The following elementary processes are traditionally taken into account: elastic scattering of electron on atom, inelastic collision of electron with atomic structure, generation of electron and generation of bremsstrahlung quantum. For realization of computer experiment on transition electrons through matter, a scheme of grouping of collisions is used.

A scheme of grouping of collisions is implemented with the following parameters: the critical energy E_e of electron formation, the critical energy E_p of bremsstrahlung quantum formation and critical angle θ_C of elastic scattering of electron on atoms. For construction of the electron trajectory, a free parameter E_S of the scheme is specified. This parameter - energy of stopping of electron - is determined by field of applicability of used descriptions of elementary processes.

The necessity of introducing of the averaging domain for calculation of spatial distributions of physical quantities is the feature of the Monte Carlo method. For calculation of the spatial dose distribution, sizes of a domain of averaging of energy, which is transmitted in the act of electron- matter interaction, are specified.

The features of MC method.

The specially designed analytical interrelations are used for the reasonable choice of free parameters of the physical model for adjusting statistical and model errors of computer experiment and minimizing time of calculation for obtaining of data with a given accuracy.

The feature of implemented statistical estimation of the dose distribution is the use of the method of translations in some spatial domains of the object. This method is used in regions, where variation of magnitude and direction of the electron flux leads to dose variation to be smaller than the established model error. The sizes of these regions are determined according to expert equations, on the basis of parameters of irradiation process and the established model error for obtained results.

4. The software ModeRTL operation

The software ModeRTL was used on the radiation-technological line (RTL) based on pulsed electron accelerator with scanning electron beams and a conveyor line at stages of planning, starting-up and adjustment works of radiation facility and realization of the different radiation-technological processes. As a source of the scanning electron beams, a linear accelerator of the type Electronica - U003" is used with the following characteristics: electron energy 5-8 MeV; electron beam current up to 0.5 mA; mean beam power up to 5 kW; pulse duration 1-4 microsecond; pulse frequency 1-250 Hz; scanning frequency of electromagnetic scanner 1-8 Hz. Schematic layout of the RTL major components is shown on Figure 2. Scanning system of the RTL has the modes of operation, that creates the triangular irradiation treatment field in target material.

The software ModeRTL was tested in the following radiation processes: sterilization of medical devices and polymer composite materials formation [11]. These technological processes differ by a method of a product irradiating. In the process of medical devices sterilization the product is irradiated on the continuously moving conveyor, and conveyor speed governs the absorbed dose in the product. In the process of polymer composite materials (PCM) formation the irradiated compound are stationary in the irradiated zone, and irradiation time governs the absorbed dose in compound. The set of required dose for full radiation-induced polymerization of compound is carried out in the time interval from 1 to 20 minutes in the stationary position. Results of calibration measurement by Cellophane DPTs-2/25 dosimetry films of the depth-dose distribution within irradiated product are in agreement with calculated results by Analytical and Monte Carlo methods of the software ModeRTL.

167

Electron beam e^-

Scan magnet

Scan horn

Window end of the scan horn

Width of scanning

Distance scan - conveyor

e^- e^- e^-

Boundary Center

Thickness of packaging

Irradiated target

Thickness of target

Thickness of additional packaging

Width of target

Conveyor

Geometrical scheme of a radiation technological line with a triangular - shaped scan horn
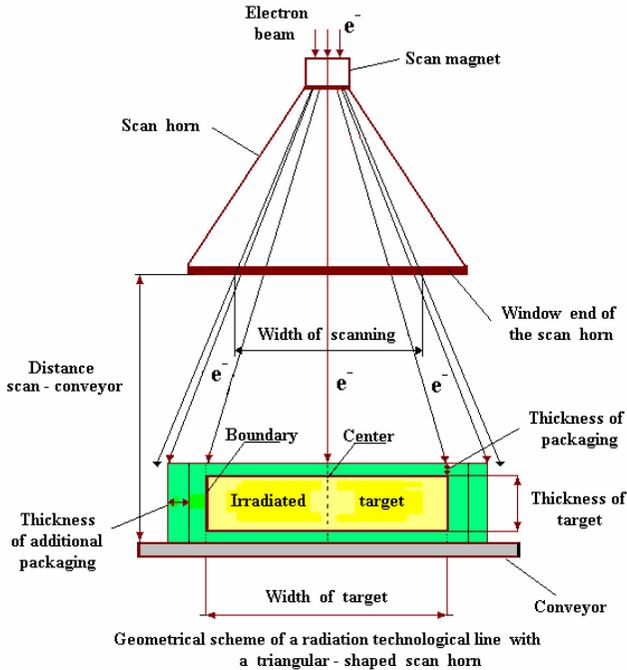
Figure 2. Schematic layout of the RTL major components.

The software ModeRTL decides the following problems in current technological processes: carries out a choice of optimum layout of the construction elements of RTL and an irradiated product, optimizes configurations of packing; optimizes of parameters of radiation treatment for irradiated product in view of features of radiation-technological process; calculates a depth-dose, charge and temperature distributions at one- and two-sided irradiation of material by scanning electron beam, the average dose, the coefficient of utilization of electron beam energy; in case of necessity carries out of the equalization the depth-dose distributions in irradiated materials with help of special filters, one- and two-sided irradiation; evaluates an economic parameters of a planned work.

The processing rate and absorbed dose distribution within of the irradiated materials depend on a lot of parameters of the radiation facility of RTL and characteristics of target material. *Input data* for the program ModeRTL are the following:

• *Parameters of electron beam*: electron energy, spectrum energy, beam current, beam diameter and spatial distribution of the electron beam intensity, pulse duration and repetition frequency in pulsed accelerators.

• *Parameters of scanning system*: modes of operation, the triangular or non-diverging irradiation treatment field in target material; beam and form of current in magnet of scanning system; repetition frequency of scanning; angular distribution of electron beam at the outlet of a scanning system.

• *Parameters of conveyor line*: speed and geometrical characteristics of the line.

• *Parameters of irradiated product*: geometrical characteristics of the irradiated product; elemental composition of the target material and weight concentrations of component; material and size of the covering for irradiated product.

• *Regimes of irradiation*: one- and two-sided irradiation.

For searching optimum solutions, the program ModeRTL *calculates and represents the following functions and parameters:* in convenient for comparative analysis graphic and tabular forms, the spatial distributions of absorbed electron dose and temperature in object irradiated by scanning electron beam; maximum, minimum and average values of absorbed electron dose; total energy transmitted to the target; factor of utilization of electron beam energy; electron ranges; relative deviations of maximum and minimum values from the average value for dose profiles at center and boundary of the target; root-mean-square deviation of dose distribution from average dose in the target, and other important characteristics.

Figure 3 represents the calculation results of equalization of the depth-dose distribution within the PCM target and determining of the optimal thickness of the PCM target treated by electron beams which was partly overlap by of the special Al filter. The predicted by the program ModeRTL of the optimal thickness of PCM wares treated by electron beams at one-and two-sided irradiation have a good agreement with experimental data. We choose the optimal thickness of PCM that corresponds to the most uniform depth-dose distribution of the electron beams.
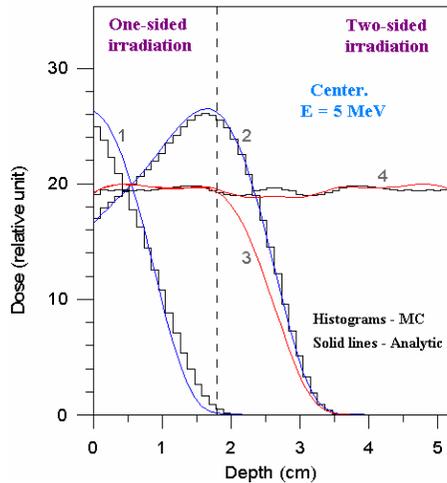
Figure 3. Depth-dose curves in the target with the use of the special Al filter for electrons beam (curve 1), without filter (2) and with converting of the beam part (curve 3 - one-sided irradiation, curve 4 - two-sided irradiation).

The comparison of the depth-dose distributions within an irradiated samples of PCM obtained both by experimental and calculated with Mode RTL program was carried out. The depth-dose distribution for an electron beam in samples was measured by Cellophane DPTs-2/25 dosimetry film placed within the sample of PCM along and perpendicularly of an electron beam axis. The absorbed dose within the range of 20-250 kGy was determined by the measurements of optical density of the film at λ =515 nm. Experiments were made for a wide beam (diameter D_b = 4 cm) and a narrow beam (D_b = 1 cm). A comparison of experimental dosimetry data and results obtained by Analytical and Monte Carlo methods are shown in Figure 4.

In a first case (curve 1) a beam diameter is approximately equivalent to R_o (continuous slowing-down approximation range) and dose distribution in target is close

5. Conclusion

The software ModeRTL was used for simulation of irradiation process on radiation technological lines (RTL) incorporating an electron accelerator with a scanner of electron beam and a conveyor. The main feature of the program is the use of hybrid scheme of calculation using the formulas of analytical models and simulation of transport of electron and gamma radiation by a Monte-Carlo method.

The software ensures a prompt, scientifically-justified choice of parameters and modes of operations of a radiation-technological line for optimization of process of irradiation of products of radiation technologies. The choice is carried out in view of features of the used equipment of the line, configuration of the irradiated product and requirements of technology to conditions of irradiation.

The use of the program ModeRTL for simulation and calculations of electron and gamma irradiation transport within irradiated products has allowed essentially to reduce the volume of routine dosimetric measurements of an absorbed dose within materials, irradiated with scanning electron beams, at stages of planning, starting-up and adjustment works of radiation facility and realization of the radiation-technological processes.

The ModeRTL program is available for Windows -98/NT/ME/XP PC' s, is written in the Delphy language, has a convenient user interface which users can use intuitively, and can be easily adapted to PC-based control system for all industrial RTL with scanning electron beams.

References

[1]   Lazurik V T, Lazurik V M , Popov G F , Rogov Yu and Korenev S A 2000 Proc. of the PCaPAC DESY Hamburg Germany p 4 CD-Rom
[2]   Habbleib J A, Kensek R P, Mehlhorm T A, Valdez G, Seltzer S M and Berger M J (1992) Report SAND 91-1634, Sandia Nat. Labs. /Reprinted, (1994) pp 1-132.
[3]   Nelson R, Hirayama H and Rogers D W O 1985 The EGS4 code system. Stanford Linear Accelerator Center Report SLAC-265, Stanford, CA
[4]   Brun R, Bruyant F, Maire M, McPherson A L, 1987 GEANT-3 Data handling division DD/EE/84-1 CERN
[5]   Baró J, Sempau J, Fernández-Varea J M and Salvat F 1995 Nuclear Instruments and Methods in Physics Research vol 100 p 31
[6]   Lazurik V M, Tabata T, Lazurik V T 2001 Radiation Physics and Chemistry vol 60 pp 161-162
[7]   Tabata T, Andreo P and Ito R 1994 Atomic Data and Nuclear Data Tables 56 105-31
[8]   Tabata T, Andreo P and Shinoda K 1998 *Radiation Physics and Chemistry* 53 p 205
[9]   Alouani Bibi F Lazurik V and Yu. Rogov. 1999Canadian Journal of Physics vol 77 N. 2 pp 127-136
[10]   Lazurik V T , Tabata T , Moskvin V 1998 IEEE Trans. Nucl. Sci. 45(3) pp 626-631
[11]   Avilov A, Deryuga V, Popov G 1999 PAC'99 4 pp 2549-2551
[12]   Lazurik V T, Lazurik V M , Popov G F , Rogov Yu 2002 Abstract book of the PCaPAC Frascaty Italy p 32
[13]   Kalinichenko A I, Kresnin Yu A, Popov G F 1997 Proc. of the ICALEPCS'97 Beijing China pp 550-554

Computational aspects of the trajectory reconstruction in the MAGNEX large acceptance spectrometer

A. Lazzaro†‡*, A. Cunsolo†‡, F. Cappuzzello†, A. Foti‡§,
C. Nociforo†‡•, S. Orrigo†‡, V. Shchepunov#, J. S. Winfield†, M. Allia†

† INFN – Laboratori Nazionali del Sud, Catania, Italy.

‡ Dipartimento di Fisica e Astronomia, Università di Catania, Catania, Italy.

§ INFN – Sezione di Catania, Catania, Italy.

Flerov Laboratory, Joint Institute of Nuclear Research, Dubna, Russia

Abstract. The ion optics of a large-acceptance magnetic spectrometer is discussed. General techniques based on a minimum of multi-purpose magnetic elements are described. A trajectory reconstruction algorithm is used to strongly enhance the energy resolving power. The influence of straggling of ions and finite detector resolutions on the achievable energy, mass and angular resolutions is studied using a library of routines (COSYMAG) which combines the calculation possibilities offered by the two codes COSY INFINITY and SRIM2000. The simulations show that a very high order of the trajectory reconstruction should be used in order to achieve the required energy resolution in the full phase space. This entails high precision in the magnetic field reconstruction. Different methods for the interpolation of the grid are tested especially with regard to the application of Differential Algebraic techniques.

1. Introduction

In the past a variety of techniques have been set up to solve the problem of the correction of aberrations, usually based on the use of corrective magnetic elements. When a large acceptance magnetic spectrometer has to be built, these techniques cannot be conveniently applied for a number of reasons [1]. First, a considerable number of corrective lenses is needed to compensate a large number of aberrations, which cannot be neglected due to the large acceptance. Secondly, the large beam envelope implies magnetic elements with a high section to length ratio. This creates a long fringe field fall-off which could induce strong high-order effects. So a large acceptance magnetic spectrometer should be conceived in such a way that the number of magnetic elements should be as small as possible. As a consequence, a large number of aberrations will

---
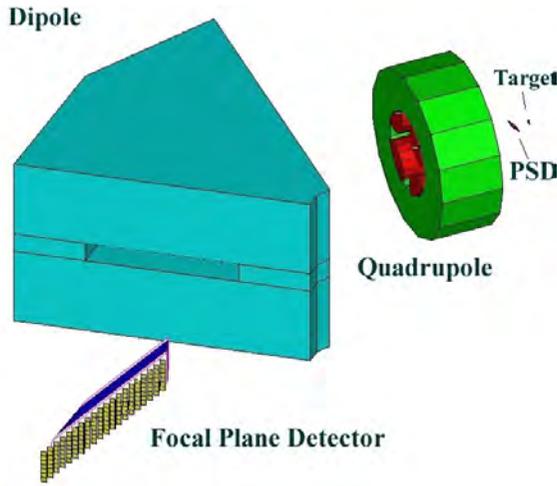
Figure 1. The MAGNEX spectrometer schematic layout, showing the quadrupole and dipole magnets and the detection system.

remain uncorrected and "software" techniques should be applied to reduce their detrimental effects on the energy resolution.

The MAGNEX spectrometer, presently under construction at the LNS Catania, is an innovative device designed to offer an high energy and mass resolution in a very large accepted phase space, thus allowing the exploration of the "new nuclear physics" connected with the use of radioactive ion beams from the EXCYT ISOL facility [2,3]. It will also be available for experiments at low to intermediate energy with stable beams accelerated from the Tandem Van de Graaff and the K-800 superconducting Cyclotron.

As shown in Fig. 1, MAGNEX consists of two magnetic elements, a quadrupole and a multipurpose bending magnet, the former to provide focusing strength in the vertical plane and the latter both momentum dispersion and horizontal focus by field boundary not perpendicular to the reference trajectory at the entrance and exit. Such a non-normal field boundary is often referred to as a "rotated pole face". A Focal Plane Detector (FPD) is located at the focal surface in order to reconstruct the ion's trajectory and impact position at the focal plane, while simultaneously identifying the nuclear species in terms of nuclear charge and mass.

Table 1. Main features of the large acceptance spectrometer MAGNEX.

Angular acceptance	~ 50 msr
Momentum acceptance	± 10 %
Energy resolving power	1000 – 2000
Mass resolution	1/200
Maximum magnetic rigidity	1.8 Tm
Threshold energy	~ 0.5 MeV/u

In addition, a Position Start Detector (PSD) is placed after the target both to generate a start signal for Time Of Flight (TOF) and to measure the vertical angle. In Table 1 the main features of the spectrometer are listed.

2.   The trajectory reconstruction algorithm

2.1 Differential Algebraic Techniques

The recent advent of the trajectory reconstruction algorithm [4] has made it possible to face the problem of the software correction of aberrations from a different and more fruitful point of view. In general the differential equations, describing the motion of ions throughout the spectrograph, are solved by perturbative methods and the Taylor coefficients of the flow linking the initial phase space with the final one are obtained. In this way by measuring the positions and direction of the ions at the focal surface one can reconstruct the full trajectory and consequently the scattering angle and the initial momentum. The most efficient perturbative technique relies on the use of the so-called Differential Algebras (DA) [5]. In this mathematic environment the integration of differential equations becomes a simple algebraic task and very high-order of the perturbation series can be treated.

2.2 Trajectory reconstruction in the MAGNEX spectrometer

In our work we have studied the application of the trajectory reconstruction algorithm and DA-techniques to the case of MAGNEX. A first question connected with the use of Differential Algebraic techniques is about the order of the reconstruction. To answer this, a realistic simulation of the algorithm itself should be carried out taking into account the straggling effects that the ions undergo when they cross the active volume of the detectors and the finite resolution of the measurements.

At the energies where MAGNEX should operate (0.5 - 8 MeV/u), straggling phenomena cannot be neglected and the effectiveness of the trajectory reconstruction algorithm could be seriously reduced if a careful shaping of the magnetic fields is not carried out. Indeed, in order to achieve a given momentum resolving power with the trajectory reconstruction algorithm, the effective image size at the focal surface should be less than the product of the dispersion and the given momentum resolution. Applying the standard formulas for the errors propagation, this requirement can be expressed by the following inequality:

$$\Delta x^{eff} = \sqrt{\left(\Delta x_f\right)^2 + \left(\frac{\partial x_f}{\partial x_i} \cdot \Delta x_i\right)^2 + \left(\frac{\partial x_f}{\partial \theta_f} \cdot \Delta \theta_f\right)^2 + \left(\frac{\partial x_f}{\partial y_f} \cdot \Delta y_f\right)^2 + \left(\frac{\partial x_f}{\partial \varphi_f} \cdot \Delta \varphi_f\right)^2} \leq \frac{\partial x_f}{\partial \delta} \cdot \mathfrak{R}_p$$

(1)

Note that the derivatives that appear in the left-hand side of Eq. (1) can be expressed as a function of the aberrations whereas the measurement errors depend both on the straggling effects and finite detector resolution. Such a coupling between the straggling, finite detector resolution and optical aberrations puts severe constraints on these latter once the former two cannot be reduced below certain practical limits. Consequently an appropriate approach [6] for the hardware minimisation of aberrations needs to be adopted.

2.3 Minimisation of the aberration effects for the MAGNEX spectrometer

It is well known that minimisation of a single aberration can induce higher order effects that cannot be neglected in a large acceptance devices. On the other hand, for the design of a spectrograph the goal is not the minimisation of single aberration value
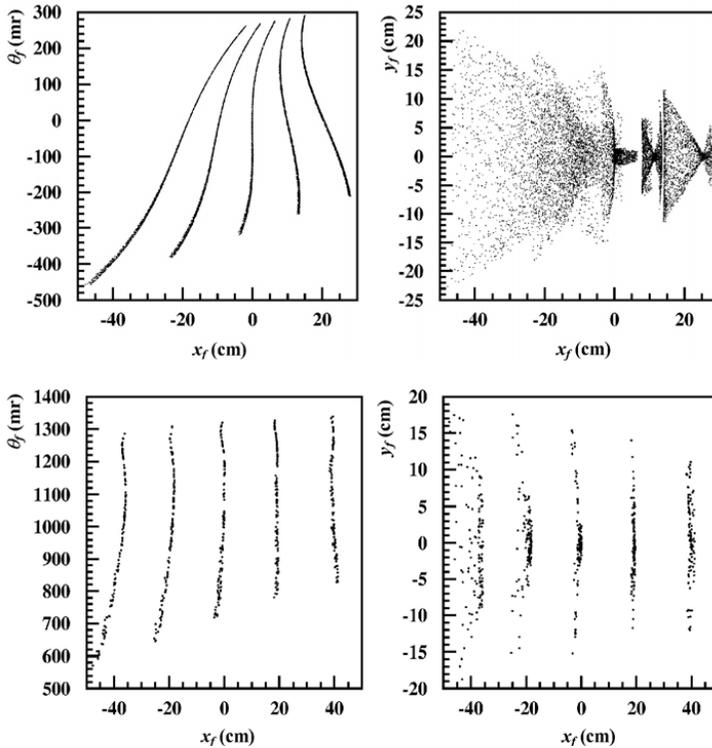
Figure 2. Two-dimensional projection, (x_f, θ_f) and (x_f, y_f), of the final phase space before (top) and after (bottom) the last step of the optimisation procedure (see text).

but the minimisation of physical quantities such as the straggling and resolution - aberration coupling, the focal plane size and the focal plane angle.

The first step of our algorithm [6] was to deduce analytical formulas describing these physical quantities in terms of the aberrations. In this manner we directly monitored the main optical properties of different possible layouts for the spectrograph. High precision was not requested at this level, therefore only aberrations up to 3rd order were considered. The program ZGOUBI [7] was used to calculate the particle trajectories through the spectrometer and extracting aberrations. In a second step the physical quantities of interest have been minimized using ZGOUBI fitting routines where some geometrical quantities has been left as parameters to be fixed within given intervals. As a result the general layout of the spectrometer was defined in a way that intrinsically guarantees reasonable optical properties. A final improvement has been achieved by a careful shaping of the dipole Effective Field Boundaries (EFBs) . In particular the latter have been modelled with 8th order polynomials and through powerful fitting routines, similar to the ones previously mentioned, the coefficients have been extracted.

In Fig. 2 the two-dimensional projections of the final phase space at the focal surface are plotted. A set of particles spanning all the phase space for five different values of the assigned fractional deviation momentum, was generated by the MonteCarlo routines of

ZGOUBI and tracked through the spectrometer. Before the 3rd step of the optimisation procedure, a strong enlargement of the final image is observed, mainly due to the non-linear chromatic effects and the uncorrected aberrations. After modelling of the EFBs the situation improves dramatically. The size of the image at the focal plane is strongly reduced and chromatic effects are negligible. This result entails a good compensation for the coupling effects we discussed before.

In conventional spectrometers for nuclear physics the actual orbit of each particle throughout the system is not traced. Nevertheless using a suitable set of detectors, it is possible to simultaneously determine the positions and angles of each particle at the focal surface. We can consider the final particle positions and angles as functions of the initial ones and fractional deviation momentum. If one neglects the initial horizontal coordinates the initial observables can be expressed as a function of the final ones as it follows:

$$\begin{cases} \theta_i = F_1(x_f, \theta_f, y_f, \varphi_f) \\ y_i = F_2(x_f, \theta_f, y_f, \varphi_f) \\ \varphi_i = F_3(x_f, \theta_f, y_f, \varphi_f) \\ \delta = F_4(x_f, \theta_f, y_f, \varphi_f) \end{cases} \qquad (2)$$

The Eqs. (2) represent the principle of the trajectory reconstruction algorithm. This standard procedure cannot be conveniently applied in the MAGNEX case due to its small vertical angular magnification [1], which would require a too high precision in the final vertical angle measurement. Then we use a modified trajectory reconstruction algorithm based on the measurement of the initial vertical angle instead of the final one.

The high order transfer and the relative inverse maps for the modified trajectory reconstruction algorithm have been calculated using COSY INFINITY (Vers.8.1) [8].

2.4 COSYMAG simulations

COSYMAG is a FORTRAN-g77 library of routines that has been developed by us in order to test the performance of MAGNEX based on the trajectory reconstruction algorithm [9].

A MonteCarlo simulator generates an arbitrary number of particles with initial conditions randomly distributed in the studied phase space. The straggling effects in the target and in the PSD (Fig. 1) are calculated using the code SRIM2000 [10]. A random noise is added to the coordinates of the particles after they cross the detector, in order to account for the detector's intrinsic resolution. The particles are tracked through the spectrometer using the maps given by COSY INFINITY. The order of tracking is directly set from the order of map used in the calculation. At the focal surface the FPD (Fig. 1) measures the coordinates of the particles in two different planes in order to extract horizontal and vertical positions and angles. Straggling in the window and detector gas is accounted for by SRIM2000 calculations. Noise is then added to the effective coordinates to simulate the intrinsic FPD position and angular resolutions.

The final reconstructed coordinates are then used to get the initial conditions by applying the inverse maps. A comparison with the simulated initial observables is possible at this stage and the energy, mass, angular resolution are calculated based on statistical analysis of the reconstructed distributions.
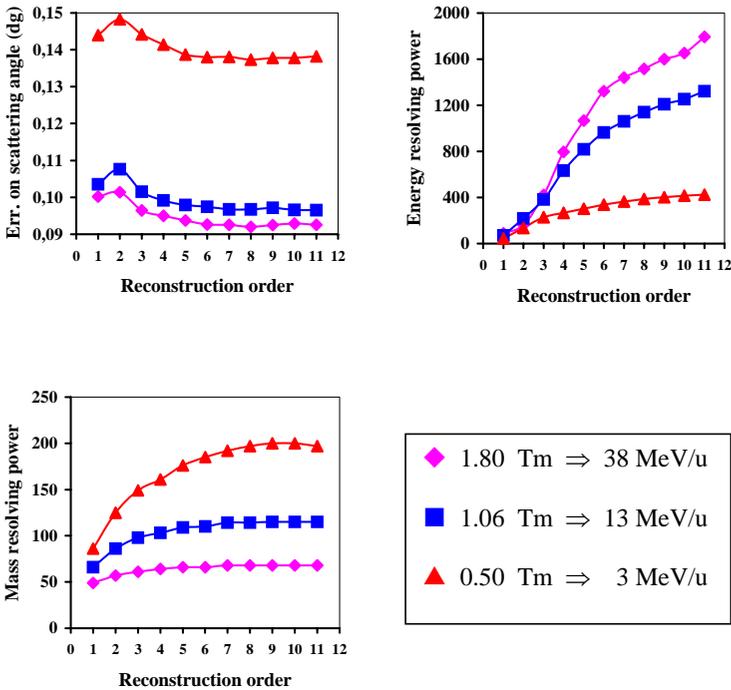
Figure 3. Behaviour of the mean error on the scattering angle measurement and of the mean energy and mass resolving power as a function of the reconstructive order for the three magnetic rigidities studied for ^{16}O.

In the simulations different ions and energies have been considered. In each case the behaviour of the angular, energy and mass resolution with respect to the reconstructive order was studied. Some results for the ^{16}O at three different magnetic rigidities are shown in Fig. 3. For simplicity we only show results averaged on the initial phase space. As can be seen, the error on the resolution on angle measurement is almost constant, starting from the 4th order. Also the mass resolving power is not critically dependent on the order of the reconstruction, except for the lower energies. The behaviour of the energy resolving power is much different. At higher energies it strongly depends on the reconstruction order and no saturation seems to be reached even at the 11th order, which is the highest we studied. Saturation is observed only at low energy, where the straggling phenomena become predominant. Results obtained for other ions look quite similar to those obtained in the ^{16}O case. Thus we conclude that, in the energy range of interest for us, trajectory reconstruction up to 10th order or even more is needed to reach the optimal energy resolving power. As consequence DA - techniques should be used to calculate the MAGNEX map. Obviously, this kind of trajectory reconstruction also implies an high level of accuracy in the reconstruction of the magnetic field.

3. Magnetic field reconstruction

The reconstruction of magnetic field for ray reconstruction purposes is constrained by two main requirements. First of all one has to consider that every measured field contains a level of uncertainty due to the experimental resolution of the devices used and stray background fields. Second one is interested in interpolation models that maintain the symplecticity of the field in order not to violate Maxwell equations and consequently disturb the calculated trajectories. As a result one needs to find an interpolation model that is weakly influenced by errors while preserving physical meaning to the whole algorithm.

3.1 2D-interpolation of measured field data

For magnetic devices with mid-plane symmetry the knowledge of the field on the median plane usually guarantees a detailed reconstruction of the whole 3D field. We expect that this would not be enough in the case of the MAGNEX dipole due to the large vertical acceptance (18 cm).

To have a quantitative answer we tested the technique firstly using a 2D polynomial interpolation to the 4th order for a 2D grid whose values were obtained by TOSCA [11]. A random noise (1 ‰, 0.5 ‰ and 0.25 ‰ compared to the maximum strength) was added to the data in order to simulate the experimental errors in the measurement. The full 3D field was then calculated by analytical extrapolation (4th order for vertical component and 3rd order for the transversal ones) from the medium plane data, using the standard ZGOUBI algorithm. Then a MonteCarlo simulation, similar to that described in subsection (2.3), was done in order to observe the effect of the noise on the image at the focal plane. Some results are shown in Fig.4. One observes that even the least noise, which corresponds to practical limit for standard measurement devices, can strongly deteriorate the image at the focal plane which compromise the effectiveness of the trajectory reconstruction algorithm.
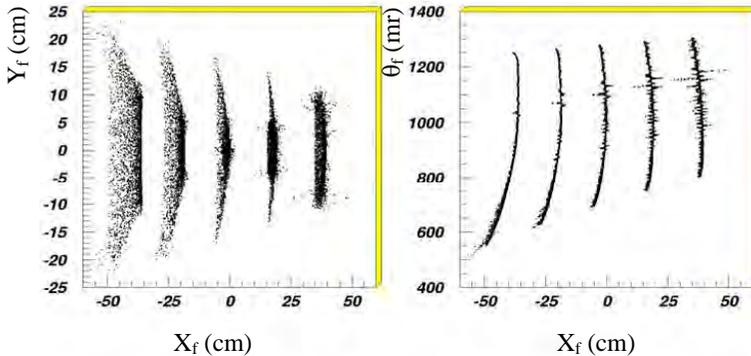
Figure 4. A similar plot as Fig.2. In the MonteCarlo the dipole field has been interpolated by polynomial functions. The noise level on the grid points was 0.25 ‰.

Much better results are obtained using a gaussian wavelets model for the 2D interpolation of the mid-plane data [4]. This leads to a smoother field distribution and consequently to a strong attenuation of the influence of noise on the images, as shown in Fig.5. Nevertheless, a residual noise is still present if the off medium plane field is extrapolated
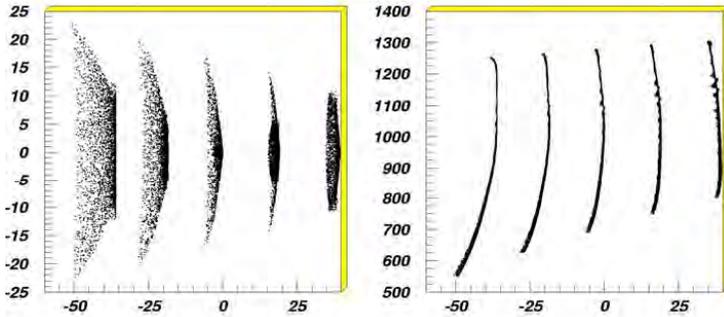
Figure 5. A similar plot as Fig.2. In the simulation the dipole field has been interpolated by gaussian functions. The noise level on the grid points was 0.25 ‰.

as explained before, that could limit the achievable energy resolution. So in our case a full 3D interpolation algorithm is desirable to reconstruct the magnetic vector field. Therefore, we need to know the three components of the field in a convenient number of planes above or below the medium one.

3.2 Reliability of field data interpolation

The large acceptance condition has important consequences to the field measurement and therefore to the interpolation algorithms. One needs accurate measurements with small errors (0.5 ‰ or less) but one should consider that the total number of measurements has the practical limit of the time needed. In the case of MAGNEX the beam envelope inside the dipole covers a volume of about $6*10^5$ cm^3. Measurements with grid step sizes of less than 1 cm in each plane would require several months of uninterrupted work for a single field of excitation and are consequently prohibitive. So one must consider at least 1 cm step size for the mid-plane measurements while even larger step sizes must be used in the vertical direction.

To investigate the reliability of gaussian wavelets representation of such a field we used a 2D grid for the medium plane calculated by TOSCA for a maximum field of 6625 Gauss. A step size of 1 cm was used for the grid, giving a total of 83160 points. The data were then interpolated for different values for the width parameter of the gaussian functions. At each point of the grid the interpolated values were compared with the TOSCA ones in order to get the absolute and fractional discrepancy. In Fig. 6, the distribution of the absolute error across the grid is shown for a width parameter of the gaussians equal to 1.4 (corresponding to a standard deviation of 1.4 cm). A worsening of the interpolation is observed in the fringe field region of the dipole where the field varies rapidly. Discrepancies up to 10 Gauss are obtained, which are too high for a good reconstruction of the field. Moreover, the sharp oscillations in the fringe field region could induce noise in the focal plane images.

Different values for the width parameter of the gaussians have been used. As indicators of the quality of the interpolation we calculate for each case the mean of the fractional error distribution and its standard deviation.
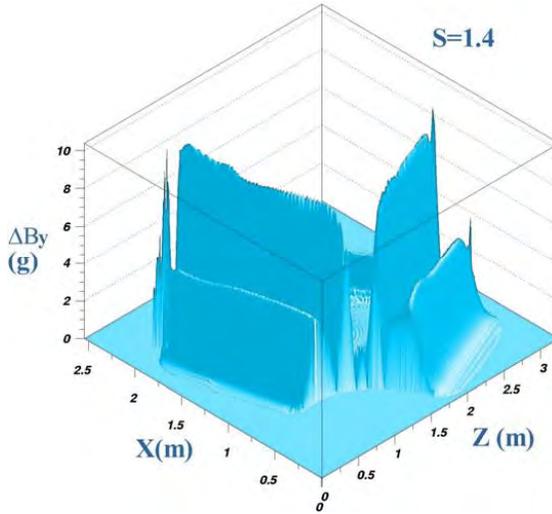
Figure 6. Space distribution of the discrepancy between TOSCA field data and the gaussian interpolation for the dipole mid-plane. The step size for the TOSCA data was 1 cm while the width parameter for the gaussian wavelets was 1.4.

As shown in Fig.7 better conditions are found for width parameter around one where a mean value about 2 Gauss and a standard deviation of 3 Gauss are observed. Worse values are obtained for larger step sizes of the grid. Simulations show that for values of the Mean Absolute Error (MAE) larger than 0.8 Gauss the overall energy resolving power of the spectrometer is sensibly influenced by the precision of the ray reconstruction algorithm [9].
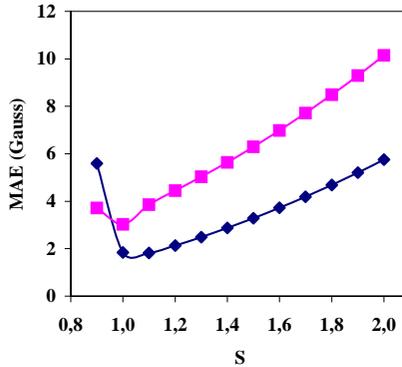
Figure 7. The Mean Absolute Error (MAE) of the 2D gaussian wavelets interpolation for the mid-plane dipole field versus the width parameter. ◆: mean of the distribution ; ■: standard deviation.

We conclude that gaussian interpolation method cannot be used for a good reconstruction of the magnetic field in the MAGNEX spectrometer. An alternative approach [12], presently under study, is based on a modified charge density method and allows the use

of measurements in several planes resulting in a global Maxwellian field that suppresses local measurement inaccuracies.

4.   Conclusions

A large acceptance magnetic spectrometer has been designed and it is presently under construction at the LNS, Catania. High energy and mass resolving power are expected due to the innovative design. To test the trajectory reconstruction algorithm together with the magnetic layout, we made a complete simulation of the whole spectrometer taking into account the straggling effects and the finite resolution of the detectors. These simulations show that high reconstructive orders are needed to achieve the required energy resolution. Under these conditions the standard methods to interpolate magnetic field cannot be applied. Instead one needs an interpolation based on the construction of an analytical function that is a solution of the Maxwell's equations such that the boundary values problem can be applied. In this case, the uniqueness theorem ensures that the solution found is the one we are looking for. Moreover, this analytical function has to be infinitely differentiable, such that DA techniques can be applied, and should provide an efficient smoothing of the experimental errors.

References

[1]   Cunsolo A. et al. 2002 Nucl. Instr. & Meth. A 484 56-83.
[2]   Ciavola G. et al. 1997 Nucl. Instr. & Meth. B126 258.
[3]   http://www.lns.infn.it/excyt/index.html.
[4]   Berz M. Modern Map Methods in Particle Beam Physics (1999, Academic press, New York).
[5]   Berz M. et al. 1993 Phys. Rev. C vol. 47 537.
[6]   Cunsolo A. et al. 2002 Nucl. Instr. & Meth. A 481 48-56.
[7]   Meot F. and Valero S. ZGOUBI user's guide Version 3 (SATURNE, LNS/GT/93).
[8]   Berz M. and Makino K. (2002) COSY INFINITY Version 8.1, MSUHEP-20704, Department of Physics and Astronomy, Michigan State University.
[9]   Lazzaro A. Ph.D. Thesis "The large acceptance and high resolution ray-tracing magnetic spectrometer MAGNEX" (2002) University of Catania.
[10]   Ziegler J.F. Biersak J.P. (2000) SRIM-2000 version 39 (copyright IBM corporation, Yorktown, NY, USA).
[11]   TOSCA (1999) Static Field Analysis OPERA-3D user guide (Vector Fields, 24 Bankside Kidlington, Oxford, England).
[12]   Degenhardt R. and Berz M. (1999) Nucl. Instr. & Meth. A 427 151-156.

Study of RF coupling to dielectric loaded accelerating structures

Wanming Liu, Chunguang Jing and Wei Gai
High Energy Physics Division, Argonne National Laboratory

Abstract. A new method of coupling rf to an X-band dielectric loaded accelerating structure is described here. In common with a technique developed earlier [1] we taper the dielectric to obtain efficient coupling, but we also incorporate a mode conversion section that results in compactness, flexibility and improved resistance to breakdown at high power. Numerical results demonstrate that enhanced microwave transmission can be achieved with a shorter dielectric taper section. The dielectric taper can also be modified without redesign of the metal structure hosting the tapered section, resulting in time and cost savings during the development cycle.

1. Introduction

Dielectric loaded accelerating structures were proposed in the early 1950's [2]. Since then, this class of device has been studied both theoretically and experimentally [3-6]. The advantages and potential problems of using dielectrics are discussed in the above references and summarized in [1]. Some potential long-term challenges of using dielectric material in a high power RF environment are breakdown and thermal heating. One practical problem that has arisen during prototyping the dielectric accelerator is the difficulty of efficiently coupling RF power into the structure [1, 6]. One scheme for solving this problem proposed and studied in [1] by P. Zou consists of a combination of a side coupling slot and a tapered dielectric layer near the slot. Simulations and low power test results show that the scheme is adequate; high power tests however were unsuccessful due to rf breakdown in the vicinity of the coupling slot [7].

To solve this problem, we have adapted a new coupling scheme proposed by Tantawi and Nantista [8] that uses a TE-TM mode converter as coupling structure. A similar technique was also studied by I. Syratchev for use in the CLIC accelerating structure [9]. Our new scheme is shown in Figure 1. A transition section is used to convert the TE mode (from the rectangular WR90 feed waveguide) to a TM mode (in a cylindrical copper waveguide). A tapered dielectric section is then used to transmit RF power into the dielectric accelerator section. This scheme separates the dielectric loaded accelerator from the coupling structure by a tapered section. Such scheme makes the coupler independent of the dielectric properties, and the highest fields are developed in the accelerator section. Because the coupler is implemented in a section of circular waveguide, the aperture of the coupling slot is much larger than in the old scheme [1] so that the peak value of the EM field at the coupling aperture is much smaller than that of the old scheme under the same input power.

One could also separate the structure shown in Figure 1 into several modules: 1) Coupling section; 2) Dielectric tapered section and 3) Dielectric accelerator section. This would greatly simplify the experimental implementation of high power testing since each of the three components could be built and tested separately.
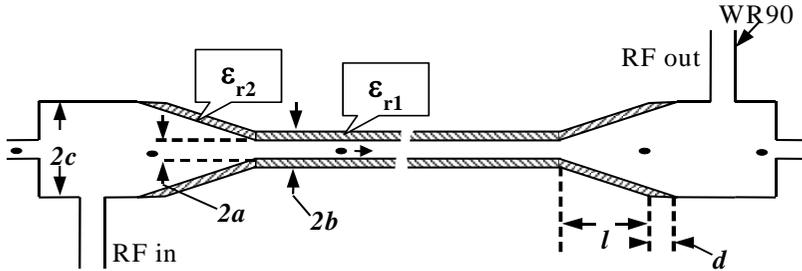
Figure 1. Example of an 11.424 GHz Dielectric Structure incorporating the new TE-TM mode conversion technique

In this paper, we will concentrate on details of the design of the tapered sections only. The properties of the acceleration section have been studied in detail previously [1, 6]. The coupling section and a simple taper section design are discussed in [10]. For this simple scheme, the length of the taper section should vary with the dielectric constant, with higher permittivity requiring a longer taper. Another disadvantage of the old scheme is that the dielectric taper is not tunable. To improve the performance of the taper section, a new scheme is proposed and discussed here.

2. Taper section conceptual design

As shown in figure 2, the new taper design is defined by the geometric parameters a, b, c, t_1, t_2 and t_3. a is the inner radius of the dielectric accelerating section, b is the outer radius of the section and c is the radius of the circular waveguide of the output port of the TE-TM converter. Unlike our old scheme (shown in figure 3), this design is much more flexible. There are 3 independently adjustable geometrical parameters, t1, t2 and
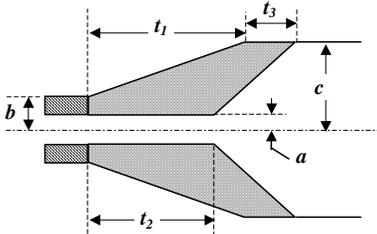
Figure 2. The new dielectric taper concept
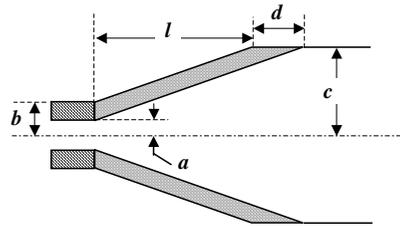
Figure 3. The old dielectric taper concept

t3. This makes it possible for us to change the dielectric taper without any change to the metal structures. This property also makes it tunable. Another advantage of this new

design is that we can achieve acceptable transmission using a relatively shorter taper section especially when the dielectric constant is high.

3. EM simulation and results

As the mode pattern of TM01 inside the dielectric loaded accelerating section will change with the frequency, it is nearly impossible to get correct results in time domain simulation if we define a waveguide port at the end of a partially dielectric loaded waveguide. So what we do is to define only one waveguide port at the end of the ordinary circular waveguide and put an absorbing boundary condition at the other end. By doing this, we can obtain the reflection parameter S11 from the simulation. The software we use is Microwave Studio™ [11].
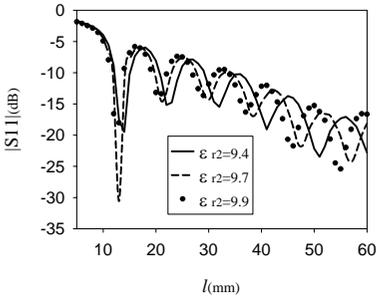
Figure 4. S11 of taper section, old scheme, $\varepsilon_r = 9.4$, 9.7, 9.9.
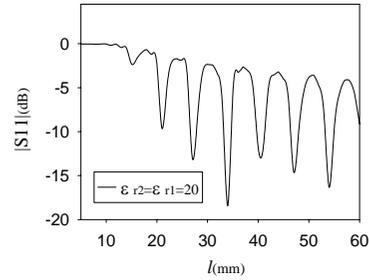
Figure 5. S11 of taper section, old scheme, $\varepsilon_r = 20$
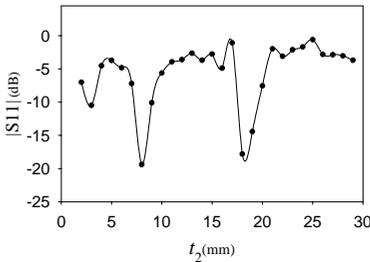
Figure 6. S11 of taper section, new scheme, $\varepsilon_r = 20$, $t_1 = 30mm$, $t_3 = 15mm$
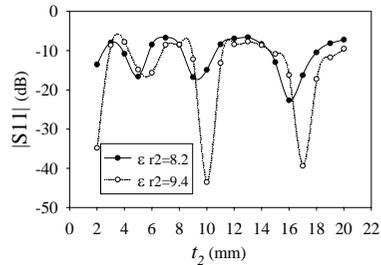
Figure 7. S11 of taper section, new scheme, $\varepsilon_r = 9.4$, 8.2, $t_1 = 40mm$, $t_3 = 14.25mm$

We have used the old scheme to design tapered sections for an X-band dielectric loaded accelerator structure [10]. The dielectrics used in these two accelerators are 9.4 and 20 respectively. Figure 4 gives S11 as a function of taper length for a structure with permittivities in the range of 9.4-9.9. We would be able to achieve good transmission when l is over 37 mm. Figure 4 also shows that if the dielectric used for the taper is slightly different from which we used in the accelerating section, the EM property of taper section will change. But as the envelope of these curves decreases fast while the taper length increases, so that this effect can be ignored when l is greater than 40 mm.

Figure 5 gives the S11 dependence on taper length for ε_r=20. The bandwidth of each relative minimum is narrow and the envelope of the curve decreases very slowly as a function of taper length. This makes the performance of taper section very sensitive to machine error and material properties, making it difficult to achieve our goal with a reasonable length of the taper section. For the new scheme, as shown in figure 6, the bandwidth of the minima is relatively wider than those of old scheme. This makes the structure less sensitive to machine error and material properties. Because the new scheme is tunable simply by adjusting the geometry of the taper, it is easy to make adjustments during prototyping.

Figure 7 gives the S11 of taper section under new scheme for the X band dielectric loaded traveling wave accelerator using dielectric with $\varepsilon_r = 9.4$. Comparing to the results in figure 4, figure 7 shows that with the same length of taper section, the new scheme can achieve better performance even though the dielectric constant of the taper section changes to 8.2. We can still use the same metallic structure designed for 9.4 by simply changing t_2 of the taper which has nothing to do with the profile of the copper jacket.

4. Summary

A new concept for a dielectric taper section is proposed here for application in X-band dielectric loaded accelerating structures. Compared with the earlier concept, this new scheme is more compact and flexible. Numerical results show that enhanced microwave transmission can be achieved with a shorter dielectric taper section. As there are 3 independent geometric parameters in the new scheme, one can change any of these 3 parameters to change the EM properties of the taper section. Under this scheme, the dielectric taper could be changed without redesign of the metal structure hosting the dielectric taper. This new scheme can reduce time and costs during the development phase. This work is supported by DOE, High Energy Physics Division, Advanced Technology Branch under the contract W-31-109-ENG-38.

References

[1] P. Zou, W. Gai, R. Konecny, X. Sun, T. Wong and A. Kanareykin, Rev. Scientific Instruments, **71**, 2301-2304 (2000)

[2] G. Flesher and G. Cohn, AIEE Trans. 70, 887 (1951)

[3] R. Kenigs, M. Jones, and W. Gai, Particle Accel. **24,** 223 (1989) and W. Gai et al, Physical Rev. Lett, **61**, 24, p2756 (1988)

[4] T-B Zhang, J. Hirshfield, T. Marshall, and B. Hafizi, Phys. Rev. E **56**, 4447(1997) and T. B. Zhang, T. C. Marshall, M. A. LaPointe, J. L. Hirshfield, and A. Ron, Phys. Rev. E **54**, 1918-1929 (1996)

[5] W. Gai, R. Konecny, and J. Simpson, in Proceedings of Particle Accelerator Conference, edited by M. Comyn, M.K. Craddock, M. Reiser and J. Thomson, pp. 636-638, 1997,

[6] J.-M. Fang and T.C. Marshall, M.A. LaPointe and J.L. Hirshfield, "An Experimental Test of a Microwave Inverse Cerenkov Accelerator (MICA)", Proceedings of the 2001 Particle Accelerator Conference, Edited by P. Lucas, S. Webber, Chicago, IL, p4240

[7] J. Power et al, in Proceedings of Advanced Accelerator Concepts: Tenth Workshop

[8] S. Tantawi and C. Nantista, private communication

[9] I. Syratchev, CLIC Note 505, CERN 2001, unpublished

[10] W. Liu and W. Gai, in Proceedings of Advanced Accelerator Concepts: Tenth workshop

[11] Microwave Studio™, Version 3.4, CST America, http://www.cst-america.com

On the map method for electron optics

Zhixiong Liu

Department of Electronics, Peking University, Beijing 100871, China

Emial: zxliu@pku.edu.cn

Abstract. In this paper the modern map method for electron optics has been discussed in detail and third order aberrations of several electron lenses have been computed through the method. A cross-check of the numerical results has been made with those evaluated by using the electron optical aberration integrals. They are very consistent with each other. In conclusion, the map method is not only concise and efficient, but very precise for electron optical aberration analysis and COSY INFINITY is an excellent programming code for such methods.

1. Introduction

Modern map methods have been widely used in particle beam physics and accelerators with great success [1] and COSY INFINITY [2,3] is an arbitrary order beam simulation and analysis computer code that is suitable for such methods. However, the feature of electron optics is that a rotating coordinate frame is always used in order to simplify the analytical expression of the paraxial trajectory equation of rotationally symmetric electron optical systems. As a result, the aberration of lenses is always expanded in rotating coordinates and must be corrected by the object magnetic immersion (OMI) effect if the object is immersed in the lens magnetic field [4]. Therefore, there exist three types of differential algebraic (DA) descriptions for electron optical aberrations.

By using the DA descriptions for aberrations developed in this work, numerical computation of third order aberrations of electrostatic, magnetic, and combined electromagnetic lenses has been carried out and the computational results have been cross-checked with those evaluated through the electron optical aberration integrals. In addition to references [5,6], the present work has also proved that the map method has advantages in conciseness, efficiency, and high precision and that COSY INFINITY is an excellent programming code for such methods. It is expected that modern map methods would become a powerful tool for aberration analysis in electron optics, especially for high order aberrations.

2. Three types of DA descriptions for electron optical aberrations

2.1. The DA description in fixed coordinates

For this type of the DA description the general electron trajectory equation [7,8] is used, which has the form of

$$X'' = \frac{\rho^2}{2\Phi}\left(\frac{\partial\phi}{\partial X} - X'\frac{\partial\phi}{\partial Z}\right) + \frac{\eta\rho^2}{\sqrt{\Phi}}(\rho B_Y - Y'B_t),$$

$$Y'' = \frac{\rho^2}{2\Phi}\left(\frac{\partial\phi}{\partial Y} - Y'\frac{\partial\phi}{\partial Z}\right) + \frac{\eta\rho^2}{\sqrt{\Phi}}(-\rho B_X + X'B_t),$$

$$\eta = \sqrt{\frac{e}{2m}}, \qquad \rho = \sqrt{1 + X'^2 + Y'^2}, \qquad B_t = \frac{1}{\rho}(B_Z + X'B_X + Y'B_Y).$$

(1)

Note that in this context we use the uppercase letters, X and Y, to represent the position coordinates in the fixed coordinate frame. For aberration analysis the transfer map is found by tracking Eq.(1) from the object plane to the image plane by means of a DA integrator. Then, the DA description in fixed coordinates for third order aberrations is expressed as

$$\Delta X_{3i} = \sum_{k,l,m,n=0,1,2,3}^{k+l+m+n=3} M_f(1, klmn) X_o^k X_o'^l Y_o^m Y_o'^n,$$

$$\Delta Y_{3i} = \sum_{k,l,m,n=0,1,2,3}^{k+l+m+n=3} M_f(3, klmn) X_o^k X_o'^l Y_o^m Y_o'^n,$$

(2)

where $M_f(1, klnm)$ and $M_f(3, klmn)$ are the third order transfer map elements in fixed coordinates and subscripts "i" and "o" signify the object and image planes. Since the DA description in fixed coordinates is only a special case of the DA description in rotating coordinates for aberrations of electron optical rotationally symmetric systems, we will not discuss it in detail and concentrate more on the latter.

2.2. The DA description in rotating coordinates

In electron optics the rotating transformation of coordinates [7,8] are

$$X = x \cos \theta - y \sin \theta,$$

$$Y = x \sin \theta + y \cos \theta,$$

$$X' = (x' - \theta' y) \cos \theta - (y' + \theta' x) \sin \theta,$$

$$Y' = (x' - \theta' y) \sin \theta + (y' + \theta' x) \cos \theta,$$

$$Z = z, \qquad \theta = \frac{\eta}{2} \int_{z_o}^{z} \frac{B(z)}{\sqrt{V(z)}} dz, \qquad \theta' = \frac{\eta}{2} \frac{B(z)}{\sqrt{V(z)}},$$

(3)

where θ is the rotation angle of the rotating coordinate frame relative to the fixed one and the lowercase letters, x, y, and z, imply the rotating coordinates in the context.

At the object plane $\theta_o = 0$ we have

$$X_o = x_o, \qquad\qquad Y_o = y_o,$$

$$X'_o = x'_o - \theta'_o y_o, \qquad Y'_o = y'_o + \theta'_o x_o,$$

(4)

$$\theta'_o = \frac{\eta}{2} \frac{B(z_o)}{\sqrt{V(z_o)}},$$

For the third order aberration there is the transformation

$$\Delta X_{3i} = \Delta x_{3i} \cos \theta_i - \Delta y_{3i} \sin \theta_i,$$

$$\Delta Y_{3i} = \Delta x_{3i} \sin \theta_i + \Delta y_{3i} \cos \theta_i.$$

(5)

Similarly, the DA description in rotating coordinates for third order aberrations takes the form

$$\Delta x_{3i} = \sum_{k,l,m,n=0,1,2,3}^{k+l+m+n=3} M_r(1, klmn) x_o^k x_o'^l y_o^m y_o'^n,$$

$$\Delta y_{3i} = \sum_{k,l,m,n=0,1,2,3}^{k+l+m+n=3} M_r(3, klmn) x_o^k x_o'^l y_o^m y_o'^n,$$

(6)

where $M_r(1, klmn)$ and $M_r(3, klmn)$ are the third order transfer map elements in rotating coordinates. Comparing Eq. (6) with the electron optical expansion of the third order aberration in Glaser's notation, we immediately obtain the corresponding relationships between the map elements and aberration coefficients, which are shown in Table 1. Furthermore, combining Eqs. (2) and (4-6), the expressions of $M_r(1, klmn)$ and $M_r(3, klmn)$ have been found as the function of $M_f(j, klmn)$ for $j = 1$ or 3 as well as θ_i and θ'_o. Below, we only write those simplest relationships defined as the DA description in rotating

coordinates for the third order aberration coefficients,

$$B = M_r(1, 0300) = B_1 \cos\theta_i + B_2 \sin\theta_i,$$

$$F = M_r(1, 1002) = F_1 \cos\theta_i + F_2 \sin\theta_i,$$

$$C = \tfrac{1}{2} M_r(1, 1011) = \tfrac{1}{2}(C_1 \cos\theta_i + C_2 \sin\theta_i),$$

$$D = M_r(1, 0120) = D_1 \cos\theta_i + D_2 \sin\theta_i,$$

$$E = M_r(1, 3000) = E_1 \cos\theta_i + E_2 \sin\theta_i, \tag{7}$$

$$f = \tfrac{1}{3} M_r(3, 1002) = \tfrac{1}{3}(F_2 \cos\theta_i - F_1 \sin\theta_i),$$

$$c = \tfrac{1}{2} M_r(3, 1011) = \tfrac{1}{2}(C_2 \cos\theta_i - C_1 \sin\theta_i),$$

$$e = M_r(3, 3000) = E_2 \cos\theta_i - E_1 \sin\theta_i.$$

where $B_1, B_2, \ldots\ldots$, and E_2 have the form

$$B_1 = G_3[M_f(1, 0300), 0, 0, 0, \theta_o'],$$

$$B_2 = G_3[M_f(3, 0300), 0, 0, 0, \theta_o'],$$

$$F_1 = G_3[M_f(1, 1002), 3M_f(1, 0003), 0, 0, \theta_o'],$$

$$F_2 = G_3[M_f(3, 1002), 3M_f(3, 0003), 0, 0, \theta_o'],$$

$$C_1 = G_3[M_f(1, 1011), -M_f(1, 1101) + 2M_f(1, 0012), -2M_f(1, 0102), 0, \theta_o'],$$

$$C_2 = G_3[M_f(3, 1011), -M_f(3, 1101) + 2M_f(3, 0012), -2M_f(3, 0102), 0, \theta_o'], \tag{8}$$

$$D_1 = G_3[M_f(1, 0120), -2M_f(1, 0210), 3M_f(1, 0300), 0, \theta_o'],$$

$$D_2 = G_3[M_f(3, 0120), -2M_f(3, 0210), 3M_f(3, 0300), 0, \theta_o'],$$

$$E_1 = G_3[M_f(1, 3000), M_f(1, 2001), M_f(1, 1002), M_f(1, 0003), \theta_o'],$$

$$E_2 = G_3[M_f(3, 3000), M_f(3, 2001), M_f(3, 1002), M_f(3, 0003), \theta_o'],$$

$$G_3(\alpha_0, \alpha_1, \alpha_2, \alpha_3, \theta_o') = \alpha_0 + \alpha_1 \theta_o' + \alpha_2 \theta_o'^2 + \alpha_3 \theta_o'^3.$$

2.3. The DA description in hybrid coordinates

The object magnetic immersion effect has been discussed in reference [4] in detail. What is emphasized here is that the OMI correction of aberration coefficients becomes very simple when the map method in hybrid coordinates is employed. For the third order aberration it is expressed as

$$\Delta x_{3i} = \sum_{k,l,m,n=0,1,2,3}^{k+l+m+n=3} M_h(1, klmn) X_o^k X_o'^l Y_o^m Y_o'^n,$$

$$\Delta y_{3i} = \sum_{k,l,m,n=0,1,2,3}^{k+l+m+n=3} M_h(3, klmn) X_o^k X_o'^l Y_o^m Y_o'^n, \tag{9}$$

where $M_h(1, klmn)$ and $M_h(3, klmn)$ are the third order transfer map elements in hybrid coordinates. The corresponding relationships between the map elements and aberration coefficients are similar to

those in Table 1. Combining Eqs. (2), (5), and (9), we obtain all the third order aberration coefficients which have been OMI corrected as follows:

$$B_m = M_h(1,0300) = M_f(1,0300)\cos\theta_i + M_f(3,0300)\sin\theta_i,$$

$$F_m = M_h(1,1002) = M_f(1,1002)\cos\theta_i + M_f(3,1002)\sin\theta_i,$$

$$C_m = \tfrac{1}{2}M_h(1,1011) = \tfrac{1}{2}[M_f(1,1011)\cos\theta_i + M_f(3,1011)\sin\theta_i],$$

$$D_m = M_h(1,0120) = M_f(1,0120)\cos\theta_i + M_f(3,0120)\sin\theta_i,$$

$$E_m = M_h(1,3000) = M_f(1,3000)\cos\theta_i + M_f(3,3000)\sin\theta_i,$$ \hfill (10)

$$f_m = \tfrac{1}{3}M_h(3,1002) = \tfrac{1}{3}[M_f(3,1002)\cos\theta_i - M_f(1,1002)\sin\theta_i],$$

$$c_m = \tfrac{1}{2}M_h(3,1011) = \tfrac{1}{2}[M_f(3,1011)\cos\theta_i - M_f(1,1011)\sin\theta_i],$$

$$e_m = M_h(3,3000) = M_f(3,3000)\cos\theta_i - M_f(1,3000)\sin\theta_i;$$

3. Computational examples

Hutter's electrostatic [9,10], Glaser's magnetic [7,8], and Ximen's combined electromagnetic [11] lenses have been chosen for computation and COSY INFINITY has been used for programming. However, in the present work some global variables, functions, and procedures for electron optics have been added to the program COSY.FOX [12].

The axial potential and magnetic induction distributions of these lenses respectively have the form

$$\text{Hutter's electrostatic lens:} \qquad V(z) = V_0\exp(K\arctan\tfrac{z}{d}),$$

$$\text{Glaser's magnetic lens:} \qquad B(z) = \frac{B_0}{1+\frac{z^2}{d^2}},$$

$$\text{Ximen's combined electromagnetic lens:} \hfill (11)$$

$$V(z) = V_0\exp(K\arctan\tfrac{z}{d}),$$

$$B(z) = \frac{B_0\exp(\frac{K}{2}\arctan\frac{z}{d})}{1+\frac{z^2}{d^2}}.$$

For each lens Eq. (1) has been tracked from the object plane to the image planes by using an eighth order Runge-Kutta integrator [13] under the given lens parameters and magnification. Then, the third order aberration coefficients have been calculated according to Eqs. (7) and (10). The numerical results are respectively shown in Tables 2, 3, and 4, together with those evaluated through the aberration integrals [7,8] and Mathematica [14].

4. Discussion and conclusion

The DA description in fixed coordinates for aberrations, Eq. (2), reflects realistic aberrations at the image plane for all electron optical rotationally symmetric systems, including electrostatic, pure magnetic and combined electromagnetic lenses, while the aberrations described by the DA description in rotating coordinates, Eq. (6), are aberrations at the corresponding rotating image plane without the inclusion of the OMI effect. As for the DA description in hybrid coordinates, Eq. (9), it is meant by the DA description in rotating coordinates with the inclusion of the OMI effect. So the relation between the DA descriptions in fixed and hybrid coordinates is a rotating transform.

For electrostatic round lenses there is no difference between the fixed and rotating coordinates nor the OMI effect, so the three types of DA descriptions are identical and the numerical results calculated through Eqs. (7) and (10) should be equal, as is shown in Table 2. In fact, because both θ_i and θ_o' vanish in the case of an electrostatic lens, we have

$$B = B_m = M_f(1,0300) = M_r(1,0300) = M_h(1,0300),$$

$$F = F_m = M_f(1,1002) = M_r(1,1002) = M_h(1,1002),$$

$$C = C_m = \tfrac{1}{2}M_f(1,1011) = \tfrac{1}{2}M_r(1,1011) = \tfrac{1}{2}M_h(1,1011),$$

$$D = D_m = M_f(1,0120) = M_r(1,0120) = M_h(1,0120),$$

$$E = E_m = M_f(1,3000) = M_r(1,3000) = M_h(1,3000),$$

$$f = f_m = \tfrac{1}{3}M_f(3,1002) = \tfrac{1}{3}M_r(3,1002) = \tfrac{1}{3}M_h(3,1002),$$

$$c = c_m = \tfrac{1}{2}M_f(3,1011) = \tfrac{1}{2}M_r(3,1011) = \tfrac{1}{2}M_h(3,1011),$$

$$e = e_m = M_f(3,3000) = M_r(3,3000) = M_h(3,3000),$$

$$(12)$$

Therefore, all the third order aberration coefficients of electrostatic lenses can be directly extracted from the map elements in fixed coordinates expressed in Eq. (12). It has also been revealed in Table 2 that only isotropic aberrations exist in electrostatic lenses and the anisotropic aberration coefficients, f, c, and e, are all equal to zero, which is consistent with theory of electron optics.

From Tables 2, 3 and 4 it is clear that all the aberration coefficients calculated through the map method are in excellent agreement with those evaluated by using the aberration integrals and OMI correction formulas, the relative errors being very small. In the meantime, it has been shown that the OMI effect does influence all aberrations to a certain extent except for spherical aberration and isotropic coma. As a result, consideration of the OMI effect is of importance if all the third order aberrations are investigated in the case of oject magnetic immersion.

In conclusion, the DA description in hybrid coordinates is the most important among the above three types of the DA descriptions; the modern map method has the advantage of conciseness and efficiency in aberration analysis, with high precision remaining unchanged; and COSY INFINITY is an excellent computer code for such methods. In addition to the electron optical canonical aberration theory [15,16], it would be expected that the modern map method will become another important method for high order aberration analysis in electron optics.

Acknowledgements

References

[1] Berz M 1999 Modern Map Methods in Particle Beam Physics (San Diego: Academic)

[2] Berz M 1990 Nucl Instr Meth A **298** 473

[3] Makino K and Berz M 1999 Nucl Instr Meth A **427** 338

[4] Ximen J Y and Liu Z X 2000 Optik **111** 355

[5] Wang L, Tang T T, Cheng B J and Cai J 1999 Optik **110**) 408

[6] Cheng M, Tang T T and Yao Z H 2001 Optik **112** 250

[7] Hawkes P W and Kasper E 1989 Principles of Electron Optics (London: Academic)

[8] Ximen J Y 1986 Aberration Theory in Electron and Ion Optics (New York: Academic)

[9] Hutter R G E J 1945 Appl Phys **16** (1945) 678

[10] Liu Z X 2002 Nucl Instr Meth A **488** 42

[11] Ximen J Y and Liu Z X 2000 Optik **111** 75

[12] Berz M 2000 COSY INFINITY Version 8 Tech Report MSUCL-1173 MSU USA

[13] Hartmann B, Berz M and Wollnik H 1990 Nucl Instr Meth A **297** 343

[14] Wolfram S 1996 The MATHEMATICA Book (Cambridge: Cambridge University)

[15] Ximen J Y 1991 Adv Electronics and Electron Phys **81** 231

[16] Ximen J Y 1996 Adv Imaging and Electron Phys **97** 359

Table 1. Relationships between the map elements and third order aberration coefficients in the rotating coordinate frame.

Map elements	Aberr. coeff.	Map elements	Aberr. coeff.
$M_r(1,3000)$	E	$M_r(3,3000)$	e
$M_r(1,2100)$	$2C+D$	$M_r(3,2100)$	c
$M_r(1,1200)$	$3F$	$M_r(3,1200)$	f
$M_r(1,0300)$	B	$M_r(3,0300)$	0
$M_r(1,2010)$	$-e$	$M_r(3,2010)$	E
$M_r(1,1110)$	$-2c$	$M_r(3,1110)$	$2C$
$M_r(1,0210)$	$-3f$	$M_r(3,0210)$	F
$M_r(1,2001)$	c	$M_r(3,2001)$	D
$M_r(1,1101)$	$2f$	$M_r(3,1101)$	$2F$
$M_r(1,0201)$	0	$M_r(3,0201)$	B
$M_r(1,1020)$	E	$M_r(3,1020)$	e
$M_r(1,0120)$	D	$M_r(3,0120)$	$-c$
$M_r(1,1011)$	$2C$	$M_r(3,1011)$	$2c$
$M_r(1,0111)$	$2F$	$M_r(3,0111)$	$-2f$
$M_r(1,1002)$	F	$M_r(3,1002)$	$3f$
$M_r(1,0102)$	B	$M_r(3,0102)$	0
$M_r(1,0030)$	$-e$	$M_r(3,0030)$	E
$M_r(1,0021)$	$-c$	$M_r(3,0021)$	$2C+D$
$M_r(1,0012)$	$-f$	$M_r(3,0012)$	$3F$
$M_r(1,0003)$	0	$M_r(3,0003)$	B

Table 2. Third order aberration coefficients computed through the map method and aberration integrals for the Hutter's electrostatic immersion lens with the lens parameters: $V_0 = 100$ V, $K = 1$, and $d = 0.01$ m and under the condition of $M = -1000$.

Aberr. coeff.	Map method − Eq. (7)	Aberr. integrals	Rel. error
B (m)	$-8.38764500096 \times 10^1$	$-8.38764500091 \times 10^1$	5.96126×10^{-12}
F	$-1.98510655925 \times 10^3$	$-1.98510653299 \times 10^3$	1.32285×10^{-8}
C (m^{-1})	$-4.91180246598 \times 10^4$	$-4.91180246615 \times 10^4$	-3.46105×10^{-11}
D (m^{-1})	$-6.40984204833 \times 10^4$	$-6.40984204850 \times 10^4$	-2.65216×10^{-11}
E (m^{-2})	$-1.48053226042 \times 10^6$	$-1.48053226042 \times 10^6$	$0.$
f	$0.$	$0.$	$0.$
c (m^{-1})	$0.$	$0.$	$0.$
e (m^{-2})	$0.$	$0.$	$0.$

Aberr. coeff.	Map method − Eq. (10)	OMI corrected	Rel. error
B_m (m)	$-8.38764500096 \times 10^1$	$-8.38764500091 \times 10^1$	5.96126×10^{-12}
F_m	$-1.98510655925 \times 10^3$	$-1.98510653299 \times 10^3$	1.32285×10^{-8}
C_m (m^{-1})	$-4.91180246598 \times 10^4$	$-4.91180246615 \times 10^4$	-3.46105×10^{-11}
D_m (m^{-1})	$-6.40984204833 \times 10^4$	$-6.40984204850 \times 10^4$	-2.65216×10^{-11}
E_m (m^{-2})	$-1.48053226042 \times 10^6$	$-1.48053226042 \times 10^6$	$0.$
f_m	$0.$	$0.$	$0.$
c_m (m^{-1})	$0.$	$0.$	$0.$
e_m (m^{-2})	$0.$	$0.$	$0.$

Table 3. Third order aberration coefficients computed through the map method and aberration integrals for the Glaser's bell-shaped magnetic lens with the lens parameters: $V = 1000$ V, $B_0 = 0.01$ T, and $d = 0.01$ m and under the condition of $M = -1000$.

Aberr. coeff.	Map method $-$ Eq. (7)	Aberr. integrals	Rel. error
B (m)	$-1.22774962370 \times 10^2$	$-1.22774962355 \times 10^2$	1.22175×10^{-10}
F	$-3.24099974570 \times 10^3$	$-3.24099931553 \times 10^3$	1.32728×10^{-7}
C (m^{-1})	$-7.42852841026 \times 10^4$	$-7.42852840963 \times 10^4$	8.48082×10^{-11}
D (m^{-1})	$-1.42976644081 \times 10^5$	$-1.42976644075 \times 10^5$	4.19650×10^{-11}
E (m^{-2})	$-3.25193084150 \times 10^6$	$-3.25193084149 \times 10^6$	3.07498×10^{-12}
f	$-9.38230480762 \times 10^2$	$-9.38230480760 \times 10^2$	2.13153×10^{-12}
c (m^{-1})	$-4.33646936433 \times 10^4$	$-4.33646936433 \times 10^4$	$0.$
e (m^{-2})	$-8.74371281419 \times 10^5$	$-8.74371281418 \times 10^5$	1.14369×10^{-12}

Aberr. coeff.	Map method $-$ Eq. (10)	OMI corrected	Rel. error
B_m (m)	$-1.22774962370 \times 10^2$	$-1.22774962355 \times 10^2$	1.22175×10^{-10}
F_m (m^{-1})	$-3.24099974570 \times 10^3$	$-3.24099931553 \times 10^3$	1.32728×10^{-7}
C_m (m^{-1})	$-7.98375704337 \times 10^4$	$-7.98375704277 \times 10^4$	7.51526×10^{-11}
D_m (m^{-1})	$-1.26319785088 \times 10^5$	$-1.26319785081 \times 10^5$	5.54148×10^{-11}
E_m (m^{-2})	$-3.13011678803 \times 10^6$	$-3.13011678110 \times 10^6$	2.21397×10^{-9}
f_m	$-4.45639641034 \times 10^2$	$-4.45639641092 \times 10^2$	-1.30150×10^{-10}
c_m (m^{-1})	$-1.73579776896 \times 10^4$	$-1.73579811414 \times 10^4$	-1.98860×10^{-7}
e_m (m^{-2})	$-3.38107934296 \times 10^5$	$-3.38107934322 \times 10^5$	-7.68984×10^{-11}

Table 4. Third order aberration coefficients computed through the map method and aberration integrals for the Ximen's combined bell-shaped electromagnetic lens with the lens parameters: $V_0 = 100$ V, $B_0 = 0.01$ T, $K = 1$, and $d = 0.01$ m and under the condition of $M = -1000$.

Aberr. coeff.	Map method $-$ Eq. (7)	Aberr. integrals	Rel. error
B (m)	$-2.49579806656 \times 10^0$	$-2.49579806621 \times 10^0$	1.40236×10^{-10}
F	$-4.13402045499 \times 10^1$	$-4.13402014037 \times 10^1$	7.61051×10^{-8}
C (m^{-1})	$7.33956465391 \times 10^4$	$7.33956465390 \times 10^4$	1.36229×10^{-12}
D (m^{-1})	$-2.44272558705 \times 10^5$	$-2.44272557783 \times 10^5$	3.77447×10^{-9}
E (m^{-2})	$-3.63429552342 \times 10^6$	$-3.63429552348 \times 10^6$	-1.65095×10^{-11}
f	$-4.39205425904 \times 10^2$	$-4.39205425904 \times 10^2$	$0.$
c (m^{-1})	$-8.50459338269 \times 10^3$	$-8.50459338286 \times 10^3$	-1.99891×10^{-11}
e (m^{-2})	$-1.64076435079 \times 10^7$	$-1.64076435079 \times 10^7$	$0.$

Aberr. coeff.	Map method $-$ Eq. (10)	OMI corrected	Rel. error
B_m (m)	$-2.49579806656 \times 10^0$	$-2.49579806621 \times 10^0$	1.40236×10^{-10}
F_m	$-4.13402045629 \times 10^1$	$-4.13402014037 \times 10^1$	7.64169×10^{-8}
C_m (m^{-1})	$-1.57685763474 \times 10^3$	$-1.57685764232 \times 10^3$	-4.80703×10^{-9}
D_m (m^{-1})	$-1.93550461832 \times 10^4$	$-1.93550452387 \times 10^4$	4.87986×10^{-8}
E_m (m^{-2})	$-3.27206878871 \times 10^6$	$-3.27206872180 \times 10^6$	2.04488×10^{-8}
f_m	$-7.60603390810 \times 10^1$	$-7.60603391317 \times 10^1$	-6.66576×10^{-10}
c_m (m^{-1})	$3.52562042045 \times 10^3$	$3.52561950111 \times 10^3$	2.60760×10^{-7}
e_m (m^{-2})	$-1.07246671507 \times 10^6$	$-1.07246685034 \times 10^6$	-1.26130×10^{-7}

Aspects of parallel simulation of high intensity beams in hadron rings

Alfredo U. Luccio § **and Nicholas L. D'Imperio**

Brookhaven National Laboratory. C-AD Department. Upton, NY 11973, USA †

Abstract. A PIC code to simulate high intensity beams in hadron circular accelerators, *BNL-Orbit*, has been implemented on parallel computers already in 1999. The issues of 3-dimensional tracking for long bunches of high energy hadrons and the relevant approximations are discussed, as well as solvers for the space charge problem in the presence of walls.

1. Introduction

The code *Orbit* [1] has been designed for PIC tracking of a particle beam in a high intensity and high energy circular hadron accelerator. In the code, space charge forces are continuously calculated and applied to the individual macroparticles of the herd as transverse momemtum kicks and as longitudinal energy kicks.

The Brookhaven edition of the code *Orbit*, or *BNL-Orbit* was made fully MPI [2] parallel, typically running on a Unix Linux platform. The parallelization of *Orbit*, with space charge calculation done is structured around the concept of longitudinally partitioning the beam into segments, the number of which can be varied. The parallelization possesses the capability to efficiently handle almost any beam configuration ranging from coasting beams of uniform longitudinal density with lengths equalling the circumference of the machine to bunched beams of non-uniform longitudinal density with lengths a small fraction thereof.

2. Split Operator. 3-D Treatment of Long Bunches

In *Orbit* the propagation of the beam is controlled by a Split Operator technique. At each stage the herd is transformed through maps calculated for a bare lattice, followed by the application of space charge kicks. Maps are provided by an optical program as MAD [3] and are arranged sequentially along the circumference of the machine. Space charge kicks are applied at certain locations in the lattice: "SC nodes".

In PIC simulation the independent variable can be either time t, or space s. While time can be a natural choice, because the space charge interaction must be calculated

§ To whom correspondence should be addressed (luccio@bnl.gov)
† Work performed under the auspices of the U.S.Department of Energy

with the position of all the macros at the same time, in a cyclic accelerator, where particle traverse the same position in the lattice many times, the position is a convenient clock. *Orbit* uses s as the independent variable.

To solve the space charge problem in the presence of accelerator chamber walls, we use a pre-calculated impedance budget, or direct calculation of the electromagnetic field from charges and currents in the beam and image charges and currents induced on the wall. Impedances are useful to represent lump properties of the entire wall structure, direct calculation is done to address in detail local wall properties.

Given beam and wall image charge and current distribution, and assuming steady state current flow, the electromagnetic problem can be solved via two partial elliptic differential equations, Poisson Law and Ampere Law (in the Coulomb's gauge)

$$\nabla^2 \Phi(P) = -\frac{\rho(Q)}{\epsilon_0}, \quad \nabla^2 \vec{A}(P) = -\frac{\vec{j}(Q)}{\mu_0}, \tag{1}$$

with ρ the beam charge distribution, and $\vec{j}$ the current distribution at a source point Q. The solution would yield the scalar electric potential Φ and the magnetic vector potential $\vec{A}$ in each field point P, from which space charge kicks are calculated.

To find ρ, one should bin the macroparticles on a suitable grid according to their position coordinates (x, y, z), and to find $\vec{j}$ one should bin the herd according to the momentum coordinates $(p_x, p_y, \Delta p/p)$.

For long bunches, as it is common in circular accelerators like synchrotrons, *Orbit* makes at the present the customary approximative assumption that the flow of beam current is parallel to the walls, thus representing the partial compensation between space charge repulsion and space current attraction by only solving the Poisson equation and multiplying the kicks by a factor $1/\gamma^2$. Then, transverse momentum kicks and longitudinal energy kicks assume the form

$$\frac{\delta p_\perp}{p} = \wp \frac{\partial \phi}{\partial r} L_T, \quad \frac{\delta \Delta E}{E} = \beta^2 \wp \frac{\partial \phi}{\partial z} L_s, \tag{2}$$

where L_T and L_s are the "lengths" of a kick, and $\wp$ the *perveance*

$$\wp = \frac{4\pi \lambda q h r_0}{\Delta x \beta^2 \gamma^3 m_0}, \tag{3}$$

with λ the longitudinal current density, h the harmonic number and Δx the (Cartesian) grid mesh size.

In principle, equations (1) should be solved in 3 Dimensions. For a numerical treatment the herd is binned to a grid of points and the equation is solved on that grid by finite difference methods. The approximation for 3-D solvers for high energy beams of very small aspect ratio (diameter/length $< 10^{-3}$) must sensibly take into account the following facts:

• it is impractical and unnecessary to make the longitudinal grid step as small as the transverse,

• the longitudinal space charge distribution varies only smoothly along the beam,

• the longitudinal motion of particles within the beam is much slower than the transverse motion,

• because of Relativity, the high energy flattens the field produced by the beam into a transverse disk, so much that the interaction between particles at different longitudinal positions vanishes very quickly with the distance,

• it is essential that before the interaction is calculated all particles are considered at the same time, to allow a correct calculation between adjacent longitudinal grid partitions.

According to the above, the transverse grid is terminated at the wall boundary, and the longitudinal grid that covers the whole length of the beam bunch consists of beam segments, long enough that the average density in each segment, the transverse aspect ratio of the segment, and the wall configuration around the segment can be considered constant. A perturbative treatment, not yet fully implemented in the code, is indicated in Section6.

In *Orbit*, in approximation 0, we then write for the beam space charge

$$\rho(x, y, z) = \rho_u(x, y)\, \rho_z(z). \tag{4}$$

with $\rho_z(z)$ a constant within a segment. This approximation simplifies the problem, since we can now only solve the transverse Poisson problem simultaneously in each segment by parallel computation.

Fig. 1 [4] shows a comparison of the longitudinal energy kick, calculated as above along the beam, with the prediction by the standard expression for a beam of radius a in a round pipe of radius b, for the same conditions.

$$(\Delta E)_{SC} \propto Z_0 \frac{\lambda'}{2\gamma^2} \left[1 + 2\ln\frac{b}{a} + f(r) \right] \tag{5}$$

where Z_0 is the impedance of free space and 'λ' the charge gradient along the beam.

3. Problem Decomposition and Load Balancing

In two dimensions the space charge calculations in *Orbit* take place at given points called nodes, situated around the circumference of the ring. The herd of macro particles, or macros, which compose the beam, arrive at each node independent of time and are transversely represented as a flat disk. The particles are then binned onto a 2-d mesh from which the potential is calculated using a sparse LU solver.The directional derivative of the potential is the respective component of the force which is then applied to each particle as a kick proportional to the length of the space charge element. Proceeding in this manner the entire ring is traversed for a given number of turns.

In a trivial parallelization for 2-D the herd would be evenly subdivided over the number of processors. The processors then separately track their particles around the ring. At a space charge node, the processors bin their particles onto a local mesh and then communicate the local meshes to a global mesh. The calculations then proceed on the global mesh as before. This scheme scales linearly.

In three dimensions a different approach must be used. The simulation can no longer be independent of time, because to correctly represent the longitudinal interactions between adjacent beam segments, all segments must be populated by macros all
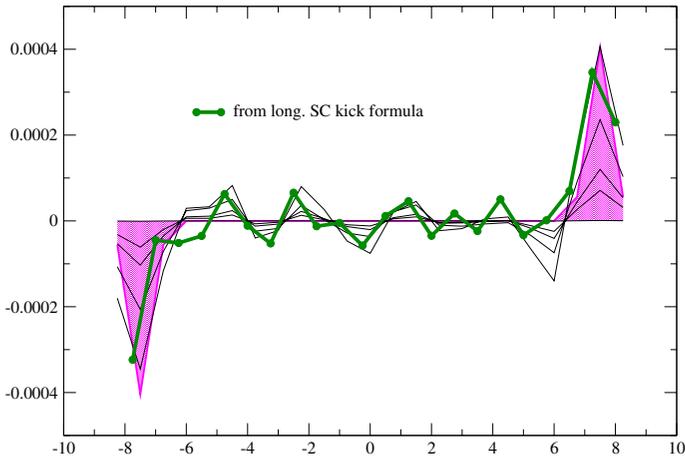
Figure 1. Longitudinal SC energy kick in a 9-slice beam (AGS). Each line: distribution of kick for various x, y. Thick line: standard equation

considered at the same time. Therefore, at each space charge node the beam, still represented as a flat disk up to this point, is expanded longitudinally to bring each particle to their appropriate position at a given time.

In this case the parallelization is not trivial any more. The beam can be divided into longitudinal segments whose boundaries are delineated by the space charge elements in the ring. The processes each take a number of these segments and do all calculations independently. The problem is still decomposed by subdividing the herd, however, the subdivision of the herd is dependent on the longitudinal locations of the particles.

In a typical run for a ring with K space charge elements, N total macros, and P processes, N/P macros would initially be injected into the ring by each process. Each macro has no constraint regarding its longitudinal position upon injection and therefore may be found in any of the K elements. When the first space charge element is encountered, the processes synchronize and expand their respective herds longitudinally. The ring is spatially decomposed along its length and so each process is assigned K/P space charge elements with one of the processes taking the remainder. The macros are then exchanged among the processes based on their longitudinal positions. After the exchange each process contains all the macros in the global herd that belong to its K/P slices. The communication involved is large only for the first space charge element as the synchrotron motion is relatively slow and particles will infrequently migrate between processes. The processes then do a 2-d transverse space charge calculation for each of

their K/P slices before collapsing the beam and continuing to track. The process repeats itself at the next space charge element though with less particle exchange.

This idea works well for a uniform beam. If the beam is not longitudinally uniform the simulation will not be efficiently load balanced, as one process may have many more macros than another. Therefore, several factors must be considered when decomposing the problem over the process domain.

The computational burdens which have the greatest effect on the performance of the code are dependent on two variables. The number of space charge elements over which the Poisson equation must be solved and the number of macros in the herd. Therefore, rather than simply dividing the the number of elements evenly among the processes it is more efficient to consider the number of elements assigned as a function of the number of macros contained within them. Load balancing is an important part of the MPI implementation of *BNL-Orbit*, that dynamically calculates an optimal decomposition scheme between each space charge element, as suggested by figure 2. For this, we use a genetic algorithm that finds out the fittest configuration to optimize run time.
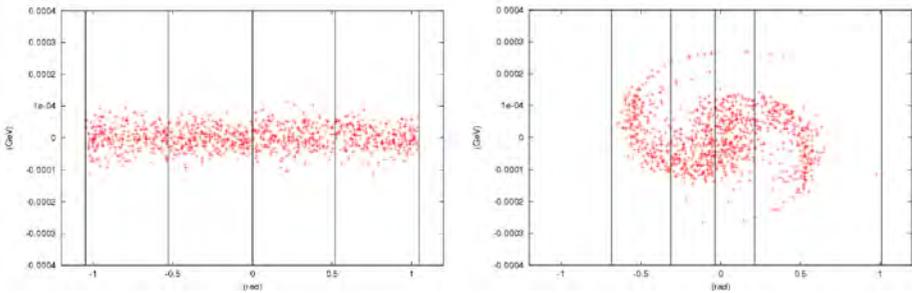
Figure 2. Process decomposition superimposed on the longitudinal phase space of a herd at (a) injection and (b) at a later time. Example for the SIS [5]

4. Details of Longitudinal Beam segmentation

The longitudinal beam segmentation is established using as a guide the shape of the beam envelope represented by the square root of the twiss functions β. The length of a segment is a fraction of a β-wavelength, as shown schematically in figure 3. The local accelerator chamber profile is associated to each segment. When the herd, that can be
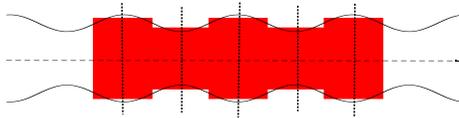
Figure 3. Slicing a beam. The wavy lines represent the envelope of the beam (*beta-wave*). Dashed vertical lines represent planes where the Poisson equation is solved.

imagined as a flat disk, reaches a SC node at a longituinal position s_{SC}, all the macros are there at different times. To calculate space charge kicks, the beam bunch is then reconstructed bringing each macro to the position s at the (center of the) segment where it was or will be at a common time, using the transfer maps between s and s_{SC}. After kicks are calculated and applied to the individuual macros, the beam is again flattened and transfered to the next node. Figures 4 show a gaussian beam bunch thus expanded in a FODO channel. Transfer maps for the expansion are for the *bare lattice*, within the
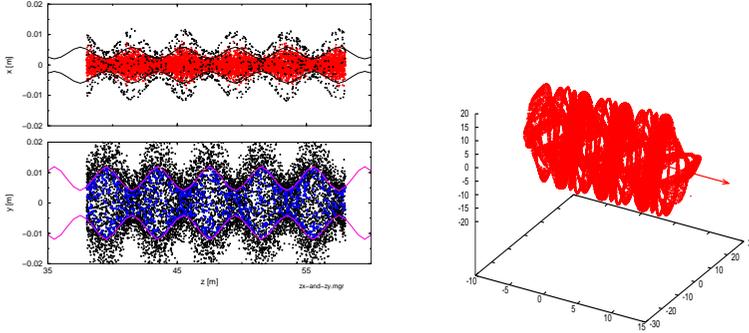
Figure 4. Expanded Gaussian beam and its 3D rendition.In the lhs figure the beam envelope is also shown

very approximation that leads to the concept of split operators.

5. 2D Poisson Solvers

The differential formulation of Poisson Law is in equation (1), the integral expression for the potential in a field point P is

$$\Phi(P) = \frac{1}{4\pi\epsilon_0\gamma^2} \int \frac{\rho(Q)}{r} dQ. \tag{6}$$

calculated by integration over the distribution at the source point Q with a Green function $G(r) = 1/r$, $r = |P - Q|$.

In an integral formulation the image charge distribution on the walls is part of the *input* of the problem and must be calculated in advance, conversely, in a differential formulation the image is part of the *solution*.

Equation (6) can be solved by direct integration (Brute Force) or the integral can be reduced to a convolution between the FFT transforms of $\rho(Q)$ and $G(r)$.

$$\Phi(r) = Const \times FFT^{-1}\left(\tilde{G}(\omega) * \tilde{\rho}_\perp(\omega)\right).$$

Both BF and FFT are implemented in *BNL-Orbit*.

The differential Poisson Equation, including boundary condition on the walls (Dirichlet condition for the function or Neumann condition for its derivative at the walls) can be numerically dealt with in various ways. At the present *BNL-Orbit* is
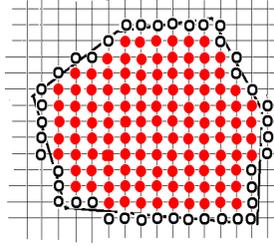
Figure 5. Solving with perfectly conducting walls

limited to perfectly conducting walls, where it is $\Phi(P_{walls}) = 0$. The field outside is zero and, by Gauss's Theorem, the total wall (image) charge is equal to the beam charge.

Let us discretize equation (1) on a $M \times N$ Cartesian grid with equal spacing in the transverse coordinates x, y. The equation and its solution (implicit sum on subscript and superscript indeces).can be written as

$$-4\pi\rho_{ij} = \mathcal{L}_{ij}^{kl}\Phi_{kl}, \quad \Phi(P) = -\tfrac{1}{4\pi}\mathcal{L}^{-1}\rho(Q). \tag{7}$$

Use the second order expression for the second partial derivative (in x)

$$\frac{\partial^2\Phi}{\partial x^2} = \frac{1}{h^2}\left(\Phi_{i-1,j} - 2\Phi_{i,j} + \Phi_{i+1,j}\right), \tag{8}$$

and write the Laplacian matrix ∇^2 in discrete form over a Cartesian grid that extends to the wall

$$\mathcal{L}_{ij}^{kl} = -4\delta_i^k\delta_j^l + \delta_{i+1}^k\delta_j^l + \delta_{i-1}^k\delta_j^l + \delta_i^k\delta_{j+1}^l + \delta_i^k\delta_{j-1}^l. \tag{9}$$

The set (7) is a system of linear equations. Figure 5 schematically suggests how to achieve a solution for perfectly conducting walls. Walls are mapped to n empty dots, and the interior to m full dots. The system of equations is exactly determined, with $n + m$ known quantities, i.e. $\Phi = 0$ at the n empty dots and ρ at the m full dots, and $m + n$ unknowns, i.e. m values of Φ to be calculated at the full dots and ρ_{image} at the n empty dots.

The solution is found in *BNL-Orbit* by using one of two alternate methods, depending on the problem: (i) by LU decomposition and (ii) by iteration.

(i) The Laplacian is a symmetric band sparse matrix. To solve the sytem of equations of the form $Ax = b$ we use the LU decomposition for sparse matrices. By replacing the coefficient matrix A by its LU factorization we have $Ax = (LU)x = L(Ux) = Ly$, where $y = Ux$; This reduces one equation, $Ax = b$, to two equations, $Ly = b$ and $Ux = y$. Using forward substitution, one calculates y, after which x is calculated using backward substitution. The LU factorization of a symmetric band sparse matrix is itself a symmetric band sparse matrix and so is done only once at the beginning of the program and stored in memory. This is very efficient for small to moderate grids where storage of the sparse LU factorization is manageable [6].

(ii) Instead of LU Decomposition, the discretized Poisson's can be solved by iteration. From equations (7) and (9) obtain

$$\Phi_{i,j} = \frac{1}{4} \left(\Phi_{i-1,j} + \Phi_{i,j+1} + \Phi_{i+1,j} + \Phi_{i,j-1} - \rho_{i,j} \right), \tag{10}$$

and solve this by iteration, starting with a guess. At iteration k it is

$$\Phi_{i,j}^{k+1} = \frac{1}{4} \left(\Phi_{i-1,j}^k + \Phi_{i,j+1}^k + \Phi_{i+1,j}^k + \Phi_{i,j-1}^k - \rho_{i,j} \right).$$

Since the beam density generally evolves slowly from one space charge node to the next, iterative techniques benefit through more rapid convergence. Several techniques were applied to the problem including: Successive Over Relaxation (SOR), SOR with Chebychev acceleration, and Conjugate Gradient (CG). Preconditioned CG was also considered but memory considerations precluded its use. As expected, CG showed the most rapid convergence, however, the basic algorithm requires more operations than either SOR technique and was therefore less efficient in this case. It was also found that basic SOR was most efficient for small grids ($N < 128$) while SOR with Chebychev acceleration was most efficient for large grids ($N > 128$). The number of iterations required for convergence to 8 significant digits was approximately 4N for both SOR methods using optimal relaxation parameters.

6. Approximations for the Longitudinal Dimension

In first approximation, as discussed before, we solve the 3D problem by segmenting the beam longitudinally and solving the 2D equation (1) simultaneously in each segment. In this approach, the longitudinal space charge forces will be calculated simply by taking the differences of the potential at a given radius between adjacent slices. Still, within this model we can do something better that takes into account details of the longitudinal distribution. A perturbative approach is the following

In 3D, using the decomposition of equation (4), Poisson becomes

$$\Phi_z \nabla_\perp^2 \Phi_u + \Phi_u \frac{\partial^2 \Phi_z}{\partial z^2} = -\frac{1}{\epsilon_0} \rho_z \rho_u \tag{11}$$

To 0-th order both $\rho_z(z)$ and $\Phi_z(z)$ are piece-wise constant

$$\begin{cases} \rho_z(z) \approx \rho_\parallel \\ \Phi_z(z) \approx \Phi_\parallel \end{cases}, \frac{\partial^2 \Phi_z}{\partial z^2} \approx 0.$$

Equating z functions and (x, y) functions on both sides, this yields

$$\Phi_\parallel = \rho_\parallel, \quad \nabla_\perp^2 \Phi_u = -\frac{1}{\epsilon_0} \rho_\perp(x, y).$$

That says that an approximate solution to the Poisson equation is obtained by solving for Φ_u in the transverse space, using the transverse charge density, and then multiply the result by a constant longitudinal $\Phi_\parallel$, or

$$\Phi^{(0)}(x, y, z) = \Phi_\parallel \Phi_u(x, y).$$

A better solution may be found by a perturbative method. With ϕ a small longitudinal potential, write

$$\Phi_z^{(1)}(z) = \Phi_\parallel + \phi(z).$$

and insert this into equation (11)

$$\left(\Phi_\parallel + \phi(z)\right)\nabla_\perp^2 \Phi_u + \Phi_u \frac{\partial^2 \phi(z)}{\partial z^2} = -\frac{1}{\epsilon_0}\rho_z \rho_u,$$

to find, after cancelling out the lowest order terms

$$\frac{\partial^2 \phi(z)}{\partial z^2} + \omega^2 \phi(z) = 0, \quad \text{with} \quad \omega^2 = -\frac{\rho_u}{\epsilon_0 \Phi_u}. \tag{12}$$

The complete solution is

$$\Phi_z^{(1)}(z) = \Phi_\parallel + \frac{1}{\omega}\frac{\partial \Phi_z}{\partial z}(0)\cos(\omega z), \quad with \quad \Phi_z(z=0) = \Phi_\parallel, \tag{13}$$

Note that (i) the frequency ω in equation (12) is a (weak) function of (x,y). e.g., for a Gaussian shaped beam, the transverse charge density and transverse potential have similar shape. (ii) The derivative of the longitudinal potential in the center of the segment is approximately proportional to the longitudinal variation of current in the beam at that location -remember that at the lowest order the longitudinal potential is equal to the longitudinal charge density- This has nothing to do with the transverse size of the beam but with its longitudinal phase space profile. (iii) The previous observation is consistent with the impedance models, where the longitudinal space charge kick on the particles is proportional to the charge per unit length in the beam.

This approach is under numerical experimentation.

Acknowledgments

References

[1] J.D.GALAMBOS, J.A.HOLMES, D.K.OLSEN A.LUCCIO and J.BEEBE-WANG: *Orbit User's Manual Vers. 1.10.* Technical Report SNS/ORNL/AP 011, Rev.1, 1999.
[2] W.GROPP, E.LUSK, A.SKJELLUM: *Using MPI.* MIT Press, 1998.
[3] H.GROTE and F.CH.ISELIN: *The MAD program, Vers.8.19.* Technical Report CERN/SL/90-13, Geneva, Switzerland, 1996.
[4] A.U.LUCCIO, N.L.D'IMPERIO, J.BEEBE-WANG: *Space-Charge Dynamics Simulated in 3-D in the Code ORBIT.* Proceedings of EPAC2002, Paris, France, 2002.
[5] B.FRANCZAK: *SIS Parameter List.* Technical Report GSI-SIS-TN/87-13, sept 1987.
[6] W.HAGER: *Applied Numerical Linear Algebra.* Prentice Hall, 1988.

High-order beam features and fitting quadrupole-scan data to particle-code models †

W. P. Lysenko‡, R. W. Garnett‡, J. D. Gilpatrick‡, J. Qiang‡§, L. J. Rybarcyk‡, R. D. Ryne‡§, J. D. Schneider‡, H. V. Smith‡, L. M. Young‡ and M. E. Schulze‖

‡ LANL, Los Alamos, NM 87545, USA
‖ General Atomics, Los Alamos, NM 87544, USA

E-mail: wpl@lanl.gov

Abstract. Quadrupole scans in the HEBT of the 6.7 MeV LEDA RFQ were analyzed to characterize the RFQ output beam. In previous work, profiles measured by the wire scanner were fit to models (beam parameterizations and HEBT simulations) to determine the transverse Courant-Snyder parameters α, β, and ε at the RFQ exit. Unfortunately, at the larger quadrupole settings, the measured profiles showed features that were not present in any of our simulations. Here we describe our latest analysis, which resulted in very good fits by using an improved model for the RFQ output beam. The model beam was generated by the RFQ simulation code TOUTATIS. In our fitting code, this beam was distorted by linear transformations that changed the Courant-Snyder parameters to whatever values were required by the nonlinear optimizer while preserving the high-order features of the phase-space distribution. No new physics in the HEBT was required to explain our quad-scan results, just an improved initial beam. High-order features in the RFQ output beam apparently make a significant difference in behavior downstream of the RFQ. While this result gives us increased confidence in our codes, we still have a mystery: exactly what high-order features in the beam are responsible for the the strange behavior downstream. Understanding this phenomenon may be helpful to understanding our halo-experiment data. We have begun to study this by comparing higher-order moments of the TOUTATIS distribution with other distributions.

1. Introduction

1.1. Quadrupole scans

During commissioning of the 6.7 MeV Low-Energy Demonstration Accelerator (LEDA) radio-frequency quadrupole (RFQ), we used a four-quadrupole high energy beam transport (HEBT) line to transport the beam from the RFQ exit to the beam stop. Quadrupole scans in the HEBT were used to characterize the transverse phase space at the RFQ exit. In this procedure, only the two quadrupoles immediately downstream of the RFQ exit were used. Quadrupole Q1 focuses in the y-direction and Q2 focuses in x. For characterizing the beam in the x-direction, Q2 was varied and the beam was observed at the wire scanner, which was about 2.5 m downstream, just before the beam stop. The strength of Q1 was fixed at a value that ensured that the beam was contained in both directions for all values of Q2. For characterizing the y-direction, Q1 was varied with Q2 fixed.

For both the x- and y-scans, as the quadrupole strength is increased from its minimum to its maximum value (we used about 10 settings in both cases), the beam size at the wire
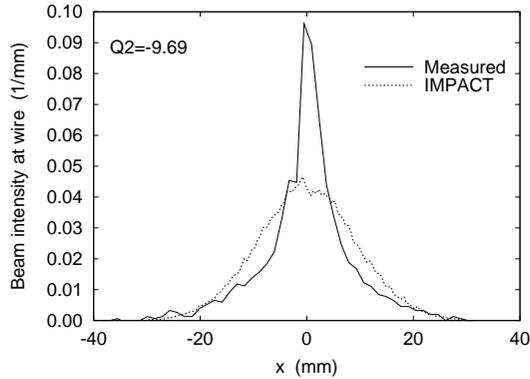
Figure 1. Comparison of a measured *x*-profile and an IMPACT simulation for Q2=−9.69 T/m. The initial beam for the simulation is a truncated Gaussian having Courant-Snyder parameters corresponding to the LINAC fit to the rms beam widths.

scanner goes through a minimum. At the minimum, the beam has a waist at the wire-scanner position. For larger quadrupole strengths, this waist occurs somewhere between the RFQ and the wire scanner. In this experiment, the wire-scanner profiles (beam intensities as functions of *x* or *y*) were recorded for each quadrupole setting. Although quad scans were done for several currents, we present results here for the highest current (nearly 100 mA) case.

1.2. Fitting to model of beam and HEBT

To determine the phase-space properties of the beam at the RFQ exit, we have to fit our data to some model that describes the behavior of the beam in the HEBT under quad-scan conditions. A model consists of two parts: a representation of the beam at the RFQ exit and a means of computing the beam at the wire-scanner position, given this beam as input. The problem is to find an input beam that best fits our data. We used input beams parameterized by the Courant-Snyder parameters α, β, and ε in the three directions. The initial beam parameters for the longitudinal direction were taken from the RFQ simulations (there was little coupling between the three directions). For computing the evolution of the beam in the HEBT, we used various simulation codes.

2. Previous results

2.1. Fit to LINAC rms sizes

Using the LINAC code and a uniform-in-4-D input distribution as our model, we could find a set of α, β, and ε values that produced a good fit to the rms beam size as a function of quadrupole gradient[1, 2]. However, for the larger quadrupole gradients, for the situation in which the beam waist is upstream of the wire scanner, the simulated and the measured beam profiles look quite different. The measured profiles had shoulders (triangular tails) that did not appear in any of the simulations. The agreement was especially poor in the *x*-direction. Figure 1 compares the measured and simulated profiles for one of the larger Q2 values, Q2=−9.69 T/m. Because of the inability to reproduce the measured profiles, we did not believe fits to this model could be used to accurately determine the Courant-Snyder parameters of the RFQ beam.

2.2. Fit to IMPACT profile shapes

In an attempt to improve our fitting procedure, we made two changes. The first change was to use the IMPACT code[3] to determine the evolution of the beam in the HEBT. IMPACT is a 3-D particle-in-cell (PIC) code with nonlinear space charge. The input beam was a truncated Gaussian parameterized by the usual Courant-Snyder parameters. The second change was to use all the profile data, not just the rms widths. For the x-scans, for each of the 11 values of Q2 and for each of the 51 x-positions of the wire, the difference between the measured intensity and the simulated intensity at the wire positions was computed. It is the sum of the squares of these 561 differences that was minimized by varying the values of α_x, β_x, and ε_x of the input beam (beam at RFQ exit). Unfortunately, this improved fitting procedure still failed to reproduce the shoulders in the profiles at the wire scanner position for the larger quadrupole gradients[4].

The beams we were using in the fitting procedures described above were uniform or truncated Gaussians in 4-D phase space. We also did IMPACT simulations (no fitting) using collections of particles generated by the RFQ simulation code PARMTEQM[5], which was used to design this RFQ. In addition, we investigated various distortions of the input phase-space distributions. In no case did our simulations exhibit the shoulders on the profiles that were seen in the measurements for the larger quadrupole gradients.

3. Improved input-beam model

Our latest improvement, which finally got good fits to the profiles, consisted of using the RFQ output beam generated by the TOUTATIS code[6] as the input beam for the IMPACT simulations. In the new fitting code, this beam (a collection of coordinates in phase space) was distorted by linear transformations that changed the Courant-Snyder parameters to whatever values were required by the nonlinear optimizer, while preserving the high-order features of the original phase-space distribution. The transformation between the initial coordinates (x_i, x_i') and the final coordinates (x_f, x_f') was

$$\begin{pmatrix} x_f \\ x_f' \end{pmatrix} = \sqrt{\frac{\varepsilon_f}{\varepsilon_i}} \begin{pmatrix} \sqrt{\dfrac{\beta_f}{\beta_i}} & 0 \\ \dfrac{\alpha_i - \alpha_f}{\sqrt{\beta_i \beta_f}} & \sqrt{\dfrac{\beta_i}{\beta_f}} \end{pmatrix} \begin{pmatrix} x_i \\ x_i' \end{pmatrix}, \tag{1}$$

where $(\alpha_i, \beta_i, \varepsilon_i)$ are the Courant-Snyder parameters of the initial beam and $(\alpha_f, \beta_f, \varepsilon_f)$ are those of the final beam.

Figure 2 shows the data flow for the latest fitting code. Data files are represented by rectangular boxes and processes by boxes with rounded corners. The part of the figure inside the dashed lines correspond to a normal IMPACT simulation (no fitting to data). The initial particle file is *partcl_start.data*, which in the present case is the output of the TOUTATIS RFQ simulation. This distribution is transformed using equation (1) by the GENSIM code using new Courant-Snyder parameters stored in the file *beam.dat* to generate the file *partcl.data*, which is used by IMPACT as the initial beam. The optimizer process QSCANFIT looks at the final particle coordinates in file *fort.9*, which describes the beam at the wire-scanner location. This is done for all quadrupole settings. The error relative to the measured data is then determined. The nonlinear optimizer in the QSCANFIT process suggests new Courant-Snyder parameters, which are passed to the file *beam.dat* to use in the next iteration.

We started our new fitting calculation with a TOUTATIS beam having Courant-Snyder parameters determined by our previous LINAC fits to the rms widths. We found, to our
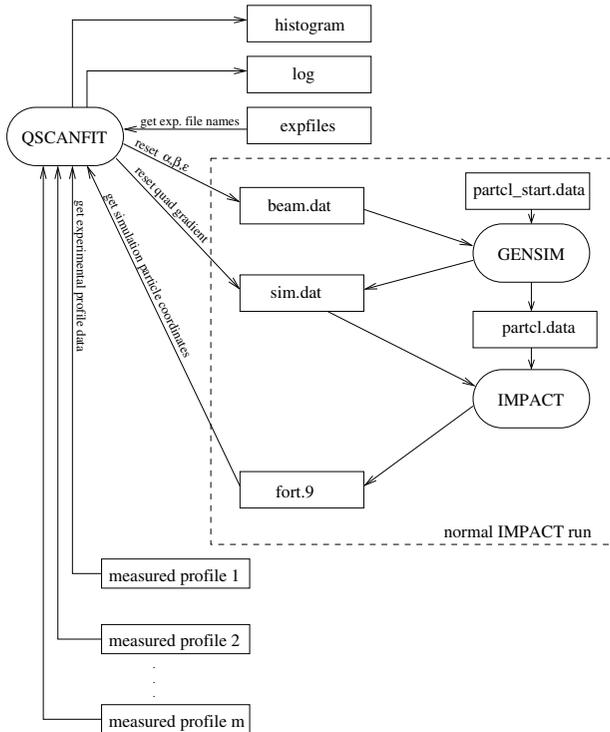
Figure 2. Data flow for fitting the quad-scan measured profiles to the TOUTATIS/IMPACT model. Data files are represented by rectangular boxes and processes by boxes with rounded corners.

surprise, that the optimizer could not find better Courant-Snyder parameters than this initial guess. The reason was that the simulation with this initial beam accurately reproduced all the structure of the measured profiles, including the shoulders on the profiles for the larger quad settings.

Figures 3 and 4 shows these results for the x-scan. We show the x-scans because it was this direction that gave the poorer fits in our previous work. The figure shows the measured and simulated profiles at the wire scanner for ten different Q2 values. We see how the beam width decreases as the strength of the quadrupole is turned up and then starts to increase again. At this point (see case for Q2=-7.70 T/m), shoulders (triangular tails) appear in the profiles. These tails were not present in any of our previous simulations that did not use the TOUTATIS beam as a starting point. Compare figure 1 to the third graph in figure 4. The old simulation did a very poor job of reproducing the shape of the distribution. It is important to remember that the only difference between the old and the new simulations is that the higher-order features of the initial beams are different. Both initial beams have exactly the same second moments (Courant-Snyder parameters). We repeated some of the TOUTATIS simulations with reduced and zero space charge. While this changed the beam size at the wire scanner substantially, the shoulders on the profiles remained. It is clear the behavior in the tails of the distribution is caused by the initial beam and not generated in the HEBT.
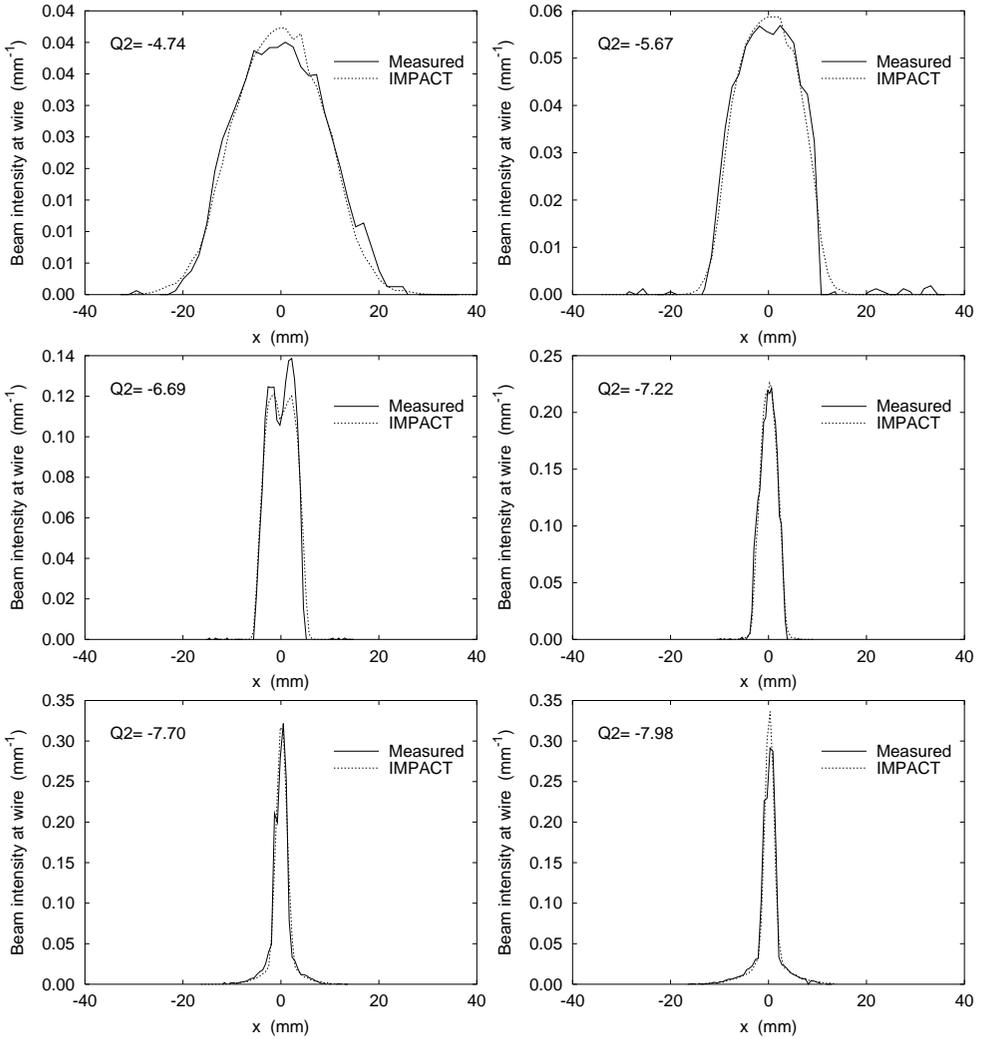
Figure 3. Comparison of measured profiles in the *x*-direction with IMPACT simulations for various values of Q2. The initial beam for all these simulations was an RFQ exit beam generated by TOUTATIS and distorted by a linear transformation to have Courant-Snyder parameters corresponding to those determined by fitting rms widths to LINAC simulations. Fitting by IMPACT to the details of the profiles did not improve these already good fits. Continued in figure 4.
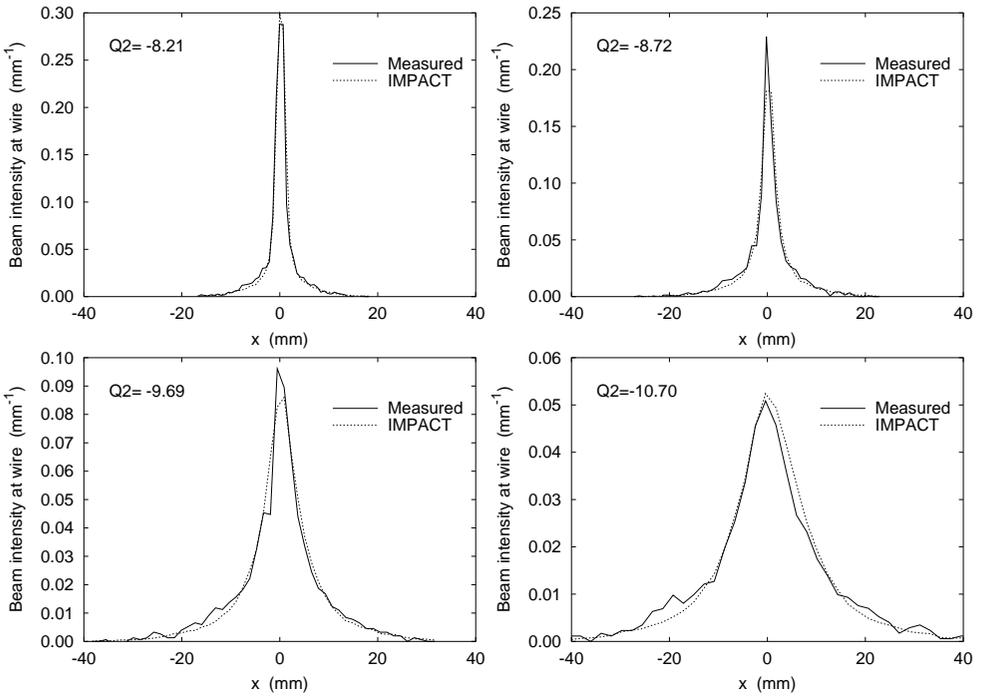
Figure 4. Continuation of figure 3. Some more Q2 values.

Table 1. Courant-Snyder parameters at the RFQ exit (unnormalized).

	α_x	β_x (mm/mrad)	ε_x (mm·mrad)	α_y	β_y (mm/mrad)	ε_y (mm·mrad)
Prediction (PARMTEQM)	1.59	0.398	2.03	-2.74	0.726	2.04
Prediction (TOUTATIS)	1.99	0.464	1.68	-3.63	0.904	1.75
Measured (LINAC rms fit)	1.79	0.358	2.11	-2.48	0.892	2.62

Table 1 shows the Courant-Snyder parameters for the LINAC rms fit. Also shown are the predictions from the PARMTEQM and TOUTATIS codes.

4. Discussion

In summary, we have seen that using a TOUTATIS beam as the basis for the input-beam model correctly reproduces the previously mysterious shoulders in the wire-scanner profiles. We have also seen that there is little feed-down from higher order. Our older rms fits generated good values for the second moments (Courant-Snyder parameters) even though those simulations got the higher-order features wrong.

The beam from the TOUTATIS simulation of the RFQ contains higher-order features that are not in the uniform, truncated Gaussian, or even the PARMTEQM output beams. It appears that features of the beam seen in the HEBT have their origins in the RFQ or
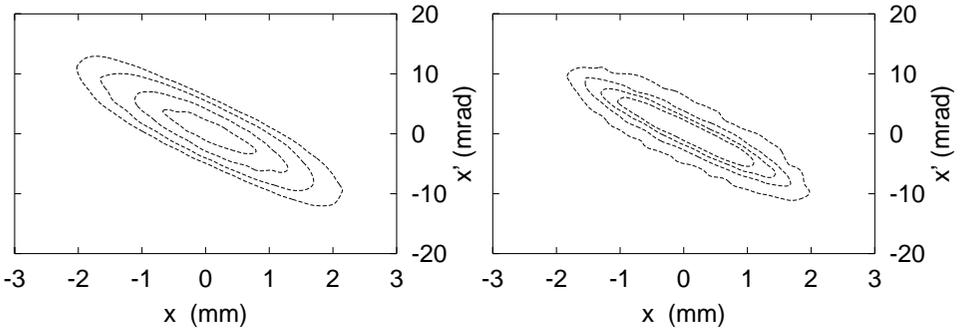
Figure 5. Contours of equal density in phase space for the PARMTEQM (left) and the TOUTATIS (right) RFQ exit distributions in the *x*-direction. Both of these beams have the same second moments.

perhaps even upstream of the RFQ. The practical consequence of this is that we have to be careful in preparing beams because high-order features can significantly influence behavior downstream. The good news is that no new physics was required to explain our quad-scan results, just a better input beam. The simulation codes accurately reproduce our experimental results. Although the quad-scan procedure differs from the ordinary HEBT operation or beam transport in a linac, the physics regime is still similar. We felt it was important that the beam behavior we observed in the experiment be seen in the simulations. We now believe we have a believable characterization of the RFQ output beam, but this is of secondary importance (quad scans are probably not a good way to measure the LEDA RFQ beam properties). The fact that the simulation codes correctly predict beam behavior increases our confidence in the design work that is based on our codes.

Of course, there is still a mystery. Exactly what high-order features in the RFQ output beam are causing the shoulders in the wire-scanner profiles? This should be investigated because it may be related to halo generation in linacs having its origin upstream of the RFQ exit. In particular, an understanding of this phenomenon may help us better understand our halo-experiment data[7].

In the TOUTATIS code, the space charge and external (rf) electric fields are calculated numerically with a multigrid finite-difference method using the actual vane geometry. This provides a more accurate representation of the fields in the region outside a cylinder of radius equal to the minimum aperture than the expansions used in PARMTEQM. Also, TOUTATIS uses the actual vane geometry to determine which particles are lost by striking the walls instead of the circular cylinder used in PARMTEQM. (The latter feature has been incorporated into the latest version of PARMTEQM and the resulting beams are now more similar to the TOUTATIS results.) Apparently, the details of the motion of particles in the RFQ near the periphery of the beam are responsible for the interesting behavior we observed in the quadrupole-scan experiments in the LEDA HEBT.

Figure 5 compares the PARMTEQM and TOUTATIS beams at the RFQ exit. Both beams have been distorted to have the Courant-Snyder parameters α, β, and ε to correspond to that of the LINAC rms fit. The contours shown for both beams are for phase-space densities of 0.005, 0.015, 0.030, and 0.050 (mm·mrad)$^{-1}$. The PARMTEQM distribution is smoother than the TOUTATIS distribution because it has more particles (93k particles compared to 27k). There is no obvious feature that explains why only the TOUTATIS beam leads to the shoulders in the profiles at the wire scanner.

One way to analyze the high-order features of the beam is by higher moments. One well-

Table 2. Invariant kurtosis for some distributions.

Distribution	k_2
uniform	2.31
Gaussian	3.46
PARMTEQM	2.96
TOUTATIS	4.46

known technique for 1-D distributions is to look at the kurtosis k, which is the fourth moment of the distribution, normalized by the square of the second moment:

$$k = \frac{<x^4>}{<x^2>^2}.$$ (2)

This quantity has value 2 for a uniform distribution, 3 for a Gaussian distribution, and higher values for more peaked distributions. Often, the kurtosis is defined with a 3 subtracted from the ratio above making the kurtosis zero for a Gaussian distribution.

For phase space (x, p), we have two dimensions for one degree of freedom. If we want to extend the definition of kurtosis to phase space we need to also account for correlations between x and p. A reasonable definition is something like the halo variable in reference [8]

$$k_2 = \frac{\left(<x^4><p^4> - 4<x^3 p><xp^3> + 3<x^2 p^2>^2\right)^{1/2}}{<x^2><p^2> - <xp>^2}.$$ (3)

The numerator and denominator are both moment invariants, which are functions of moments that are preserved for linear motion. The denominator is the square of the usual rms emittance. Other moment invariants exist and may also possibly be useful for describing halo. The advantage of a definition like that of (3) is that its value is the same anywhere in a beamline where the motion is linear. Thus the halo cannot hide just by being observed at some particular point in the beamline. (Of course, this is only approximate if nonlinearities are involved.) Because of this property, we can think of k_2 as some kind of invariant kurtosis. Table 2 shows the value of k_2 for some distributions. Notice that the TOUTATIS distribution has a fairly high value of the invariant kurtosis. It may be useful to study if high kurtosis is an indicator of susceptibility to halo generation.

References

[1] Lysenko W P *et al* 2000 *Proc. of the XX Int. Linac Conf. (Monterey)* SLAC-R-561 pp 809–811
[2] Schulze M E *et al* 2000 *Proc. of the XX Int. Linac Conf. (Monterey)* SLAC-R-561 pp 575–577
[3] Qiang J *et al* 2000 *J. Comput. Phys.* **163** 434
[4] Lysenko W P *et al 2001 Proc. of the 2001 Particle Accelerator Conf. (Chicago)* IEEE-01CH37268 pp 3051–3053
[5] Crandall K R *et al RFQ Design Codes*, *Los Alamos National Laboratory Report* LA-UR-96-1836 (Revised November 17, 2001)
[6] Duperrier R 2000 PRST-AB **3** 124201
[7] Garnett R W *et al* to be published in *Proc. of the XXI Int. Linac Conf.*
[8] Allen C K and Wangler T P submitted to *Phys Rev* ST:AB

Muon beam ring cooler simulations using COSY INFINITY

Carlos O. Maidana†, Martin Berz† and Kyoko Makino‡

† Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA

‡ Department of Physics, University of Illinois at Urbana-Champaign, 1110 W. Green Street, Urbana, IL 61801-3080, USA

E-mail: `maidanac@msu.edu, berz@msu.edu, makino@uiuc.edu`

Abstract.   In this paper we present simulations using COSY INFINITY to study the behavior of muon beams in a ring cooler designed by V. Balbekov[1]. Because of the substantial transversal emittance, the nonlinearities play a very important limiting role that must be understood and controlled well, which leads to the requirement of high order computation. We describe the system, the approaches for the simulations of the large aperture solenoids and magnetic sectors, and we show the nonlinear transfer maps as well as tracking simulations for different field models, and compare with other methods based on various approximations.

PACS numbers: 02.60.Cb, 05.45.-a, 29.27.-a, 29.27.Eg, 41.85.-p, 41.85.Ja, 41.85.Lc

1. The ring and the simulation approaches

Various designs and ideas have been developed for cooling of short lived muon beams in neutrino factories and muon colliders[2]. The concept of cooling is based on ionization through material[3, 4, 5, 6], and to reduce cooling time, normally the system has a combined structure, consisting of absorbing material, accelerating cavities and guiding magnets[2, 7]. Because of the huge transversal emittance of muon beams, the consideration of nonlinear effects is an essential component in an earlier design stage. Lately, several designs of ring coolers have been considered because of the ability to utilize cooling sections repeatedly, and the additional potential for transversal and longitudinal emittance exchange. In this paper, we analyze a muon beam ring cooler designed by V. Balbekov[1]. The layout of the system is shown in Figure 1.
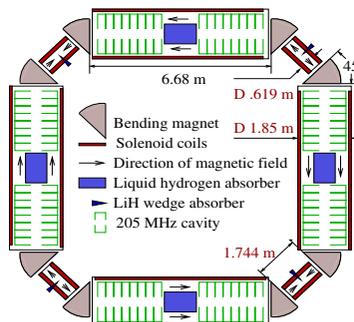
Figure 1. The layout of the tetra muon cooler designed by V. Balbekov[1].
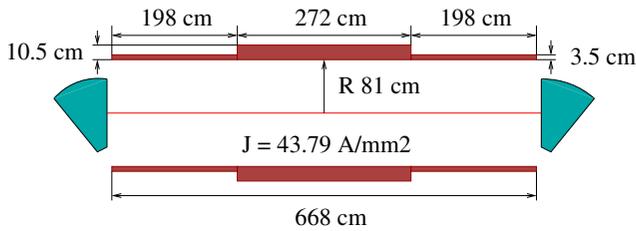
Figure 2. The parameters of the long straight solenoidal section[1].
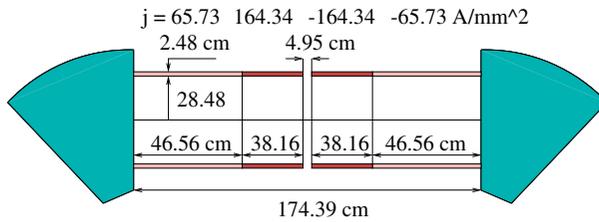
Figure 3. The parameters of the short straight solenoidal section[1].
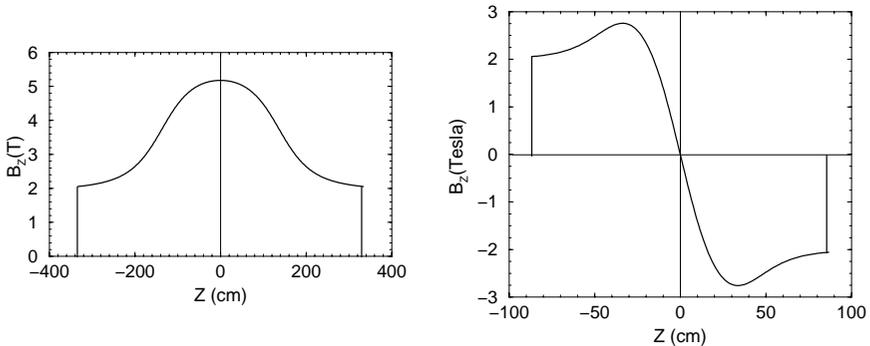
Figure 4. The hard edge model of the axial magnetic field in the long straight solenoidal section (left) and in the short straight solenoidal section (right), assuming the solenoid coils extend to infinity[1].

The ring consists of eight straight sections dominated by solenoids and eight inhomogeneous bending magnets[1]. The four long straight sections have absorbing material and accelerating cavities inside the solenoids, thus the aperture is very large. The parameters of the solenoids in the long section are shown in Figure 2. The four short straight sections have wedge absorbers to allow for transversal and longitudinal emittance exchange in the middle, where the longitudinal magnetic field component flips direction. The parameters of the solenoids in the short section are shown in Figure 3. Balbekov uses a hard edge model for

all magnets, and for simplicity of design purposes, it is usually adequate to assume that the coil of the magnets extend to infinity[1]. The profiles of the longitudinal component of the axial fields are shown in Figure 4.

First, if the length of the solenoids is finite, the field profiles differ significantly. Second, in the long section, due to the huge aperture, the fringe field extension is exceedingly long[8]. Third, the hard edge model is somewhat unrealistic even for the case of ferromagnetic yokes designed to block the fall-off of the fields[1] because they apparently need to have an aperture large enough for passage of the muon beams that have large transversal emittance. Considering these, we study the effects due to the different treatment of the fields using the code COSY INFINITY[9]. The perhaps most realistic field treatment in the design stage is to assume finitely long solenoids as indicated in Figures 2 and 3 without assuming the presence of ferromagnetic yokes. Such field profiles are shown in Figures 5 and 6, including the outside fringe regions. The code COSY INFINITY allows the nonlinear treatment of such solenoidal fields including outside fringe field effects[9, 8]. For the purpose of comparison, we also study the hard edge model of the fields. Balbekov uses the following linear kicks applied to the transversal components of momentum to recover the most important edge field effect, namely the induced overall rotation of the particles:[10]

$$\Delta p_x = \frac{C}{2} B_z y, \quad \Delta p_y = -\frac{C}{2} B_z x, \tag{1}$$

where $\vec{p}$ is in MeV/c, x and y are in meter, B_z is the longitudinal component of the axial field at the edge in Tesla, and $C = 299.79245$. We also use the same linear kicks when the hard edge model is used.

2. Transfer maps of solenoidal sections

We compare the effects of the different treatment of the solenoidal fields in the long and short straight sections. A long section consists of three solenoidal parts, and a short section consists of four solenoidal parts, with the longitudinal field flipping direction in the middle. Both the long and the short sections are designed to have stronger current toward the middle. Due to the flip of the field direction and the relatively small aperture, the short section is more readily treatable by various approximations.

2.1. Short straight section

We compute the nonlinear transfer maps of the short section for different field models with the beam kinetic energy of 250 MeV. We list the transfer maps of the hard edge model of infinitely long solenoids first. For the purpose of comparison, we show the map without and with the linear kicks (1). Below, parts of the nonlinear transfer maps are shown in the notation of COSY INFINITY[9]. We observe that the linear x, $a(= p_x/p_0)$ terms and the linear y, $b(= p_y/p_0)$ terms are almost decoupled when the kicks are applied, while they are coupled without the presence of kicks. Thus, the linear kick approximation recovers the main point of the linear motion and one of its important physical properties.

In the subsequent excerpts from transfer maps, the four columns represent final horizontal position x (in meter), final horizontal slope a, final vertical position y (in meter), and final vertical slope b, as a polynomial in the initial conditions. The exponents of the polynomial are listed in the last column; for example, "4100" corresponds to the initial horizontal position raised to the fourth power, and the initial horizontal slope raised to the first. The top lines of the map represent the linear motion, and corresponds to the well-known transfer matrix (although the latter is usually shown as the transpose of our format).

214
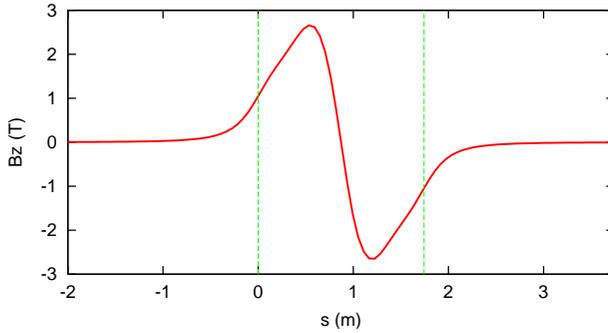
Figure 5. The axial magnetic field profile of the short section with finitely long solenoids and fringe fields.

Hard edge model with infinitely long solenoids (no linear kicks)

x_f	a_f	y_f	b_f	xayb
-0.1467136	-1.796260	0.9193886	-0.2657072	1000
1.015304	-0.1467136	0.3060451E-11	0.9193886	0100
...	...	...	...	....
-15.71585	27.23532	-23.72338	-33.63860	5000
-36.02640	5.753202	-21.46530	-55.14127	4100
-38.39075	16.29484	-33.88081	-55.16902	3200
-33.14696	-7.116152	-14.53667	-36.65477	2300
-14.27604	2.840011	-11.47237	-15.47623	1400
-7.567237	-3.099152	-1.462962	-6.833978	0500

Hard edge model with infinitely long solenoids with linear kicks

-0.1467136	-0.9637261	-0.2022731E-03	0.5845774E-04	1000
1.015304	-0.1467136	0.3060451E-11	-0.2022730E-03	0100
...	...	...	...	....
-2.968705	1.388108	1.284507	1.099203	5000
-12.74829	-1.191336	1.970436	4.189763	4100
-22.23560	-5.817135	-0.2996768	5.761626	3200
-22.28744	-10.62269	-3.195642	4.297247	2300
-15.60109	-7.532831	-4.618497	1.461128	1400
-7.567237	-4.424200	-1.462962	0.1989097E-01	0500

We now list the same parts of the map of the hard edge model of finitely long solenoids. As seen in Figure 5, the edge field strength is about half of that with infinitely long solenoids. Thus, the map differs from the previous case, and the (x,x) and (a,a) terms show an obvious difference.

Hard edge model with finitely long solenoids with linear kicks

0.3762572E-01	-0.9144481	0.1324007E-03	0.9123892E-05	1000
1.092008	0.3762572E-01	0.1216993E-10	0.1324008E-03	0100
...	...	...	...	....
-4.559878	2.194351	1.925709	2.734348	5000
-14.03917	1.790312	1.247951	8.100432	4100

```
-26.15120    -0.5027561    -4.741542     9.421870    3200
-29.55137    -6.150214     -9.833677     4.913186    2300
-20.98499    -6.094498     -8.584871    -0.5031165   1400
-8.419566    -5.823382     -1.776100    -0.7673755   0500
```

We compare this map with the one computed for finitely long solenoids with correct outside fringe field consideration without using the linear kicks. These two maps agree well, confirming that the kick approach works well so far.

Finitely long solenoids with correct fringe field consideration

```
0.9113584E-02-0.9101484     0.2707143E-04-0.8741688E-06  1000
  1.098631      0.9113584E-02-0.1597370E-05 0.2707097E-04  0100
    ...           ...           ...            ...         ....
-4.919054      1.095873      0.6296333      1.603693     5000
-14.58589      0.6387705    -0.2037065      4.302134     4100
-24.12332     -1.227331     -3.197581       3.442199     3200
-26.81068     -7.088700     -4.268833     -0.1562313     2300
-19.59752     -9.127604     -2.390249     -2.335453      1400
-8.224955     -7.060173      0.1989972     -0.9178104    0500
```

2.2. Long straight section

We performed the same study for the long straight section with the beam total energy of 250 MeV (the kinetic energy of 144.32 MeV). The hard edge model is used for infinitely long solenoidal field, and for finitely long solenoidal field, where the edge field strength is again about half of that with infinitely long solenoids. The computed transfer maps are compared to the one with the correct outside fringe field consideration. Since no good agreement was found between those three maps even in linear terms, we list only a part of the linear terms below.

Hard edge model with infinitely long solenoids with linear kicks

```
0.7201144     0.6140963     0.4256817     0.3629942    1000
-0.3623067    0.7201318    -0.2141606     0.4256523    0100
```

Hard edge model with finitely long solenoids with linear kicks

```
-0.1764031E-02 0.4682935E-01-0.3214589E-01 0.9457639   1000
-0.5217216E-01-0.1420209E-02 -1.053667    -0.3216291E-01  0100
```

Finitely long solenoids with correct fringe field consideration

```
0.2334781     0.7546891     0.8859026E-01 0.2042114    1000
-1.157656     0.2462930    -0.3132503     0.4123110E-01  0100
```

The main reason for the disagreement between the hard edge model and the correct fringe field treatment is due to the limitation of the linear kick approximation in (1). Comparing the field profiles between the short section in Figure 5 and the long section in Figure 6, the edge field strength is almost the same, namely about 1 Tesla, but the extension of the outside fringe fields behaves differently. The outside fringe fields of the short section vanish rapidly, but those of the long section cannot vanish even for a very long distance. By setting the inner radius of the solenoids to 1/10 of the original radius while keeping all the other parameters fixed, we computed the linear maps for the hard edge model and the correct fringe field treatment for finitely long solenoids, and we found reasonable agreement.
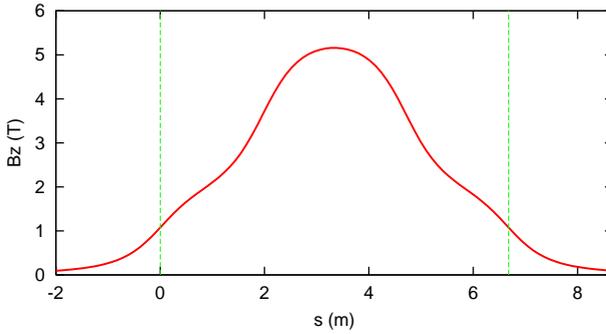
Figure 6. The axial magnetic field profile of the long section with finitely long solenoids and fringe fields.
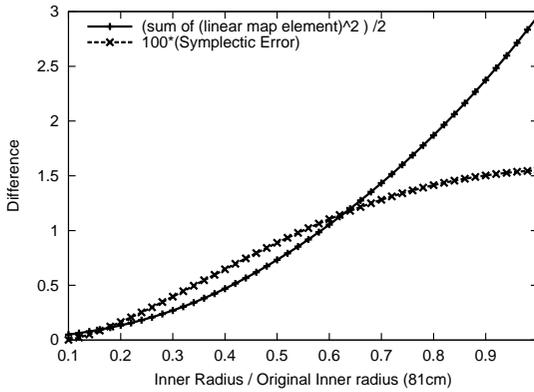
Figure 7. The disagreement of the linear transfer maps between the hard edge model and the correct fringe field treatment for the finitely long solenoids of the long section as functions of the aperture.

**Hard edge model with finitely long solenoids with linear kicks
(The inner radius is 1/10 of the original size.)**

```
-0.1162300E-01-0.1163328E-01  0.2318096       0.3343426       1000
 0.9834308E-01-0.4517004E-02 -2.826397        0.2320568       0100
```

**Finitely long solenoids with correct fringe field consideration
(The inner radius is 1/10 of the original size.)**

```
-0.5651780E-02-0.8030099E-02  0.2324166       0.3343818       1000
 0.6789750E-01-0.5511159E-02 -2.827323        0.2324200       0100
```

Figure 7 shows the correlation of the agreement of the maps for various aperture sizes. The difference in the sum of the square of linear terms and the difference in the symplectic error are plotted as functions of the ratio of the inner radius to the original size.

3. Dynamics through the magnets in the ring

Since one half of the ring characterizes the whole system as seen in Figure 1, we compute the transfer map of one half of the ring to study the beam dynamics through many revolutions in the ring. Scanning the energies of the reference particle shows that under the presence of dipole fringe fields and solenoid fringe fields, the linear motion is frequently unstable, suggesting the need to re-fit the optical properties of the ring. To illustrate the performance, we thus restricted ourselves to the design energy of 250 MeV total (=kinetic energy + muon mass energy). Figures 8 and 9 show tracking for various cases. Figure 8 shows the hard edge model of the solenoid with finitely long solenoids in the linear kick approximation. The left picture shows tracking at order 9 with a hard edge model bending magnet, while the right picture shows the effect of using a realistic bending magnet fringe field, which here leads to unstable linear motion. Figure 9 shows the finitely long solenoids with correct fringe field
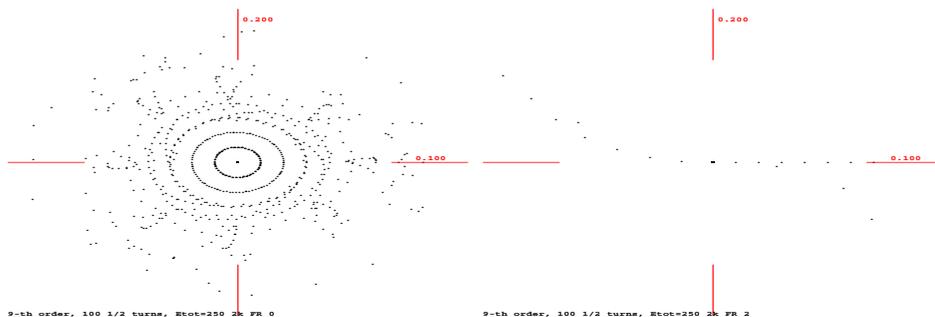
Figure 8. Tracking 50 revolutions at reference energy of $E_{tot} =250$ MeV with finitely long solenoids in hard edge kick approximation, using hard edge dipole fields (left) and realistic dipole fringe fields (right).
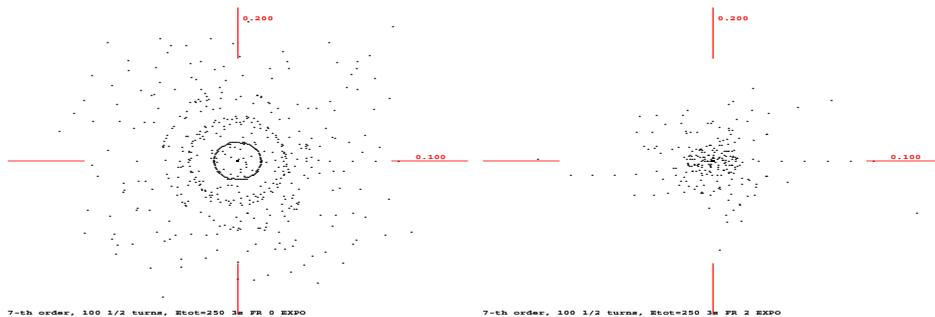
Figure 9. Tracking 50 revolutions at reference energy of $E_{tot} =250$ MeV with finitely long solenoids with correct fringe field consideration, using hard edge dipole fields (left) and realistic dipole fringe fields (right).

consideration; again the left picture shows the situation for a hard-edge bending magnet, while the right picture shows the effects of a realistic bending magnet fringe field.

Particle tracking in the x-a phase space is done for 50 full revolutions in the ring, and the Poincare sections are in the middle of the short straight solenoidal section, i.e. the upper left corner of Figure 1. In Figures 8 and 9, the horizontal axis is the horizontal position x in meter, and the vertical axis is the horizontal slope $a = p_x/p_0$. The particles with the initial positions 1, 2, ..., 7 cm are tracked in the ring, and the ends of lines showing the axes in the pictures are ± 0.1 meter in x and ± 0.2 in a.

Acknowledgments

References

[1] V. Balbekov. Ring cooler progress. The Muon Collider and Neutrino Factory Notes MUC-NOTE-COOL-THEORY-0246, Fermi National Accelerator Laboratory, 2002. See http://www-mucool.fnal.gov/notes/notes.html.

[2] S. Ozaki et al. for the Muon Collaboration. Feasibility study-II of a muon-based neutrino source. Technical Report 52623, Muon Collider Collaboration, BNL, 2001.

[3] A. N. Skrinsky and V. V. Parkhoumchuk. Methods of cooling beams of charged particles. *Soviet Journal of Particles and Nuclei*, 12:223–247, 1981.

[4] D. Neuffer. Principles and applications of muon cooling. *Particle Accelerators*, 14:75–90, 1983.

[5] Ya. S. Derbenev. The self-cooling of charged particle beams in a straight line. Technical report, Department of Physics, University of Michigan, 1991.

[6] D. Errede, M Berz, C. J. Johnstone, K. Makino, and A. van Ginneken. Stochastic processes in muon ionization cooling. *Nuclear Instruments and Methods*, in print, 2003.

[7] K. Makino, M. Berz, D. Errede, and C. J. Johnstone. High order map treatment of superimposed cavities, absorbers, and magnetic multipole and solenoid fields. *Nuclear Instruments and Methods*, in print, 2003.

[8] K. Makino and M. Berz. Solenoid elements in COSY INFINITY. In these proceedings.

[9] M. Berz and K. Makino. COSY INFINITY Version 8.1 - user's guide and reference manual. Technical Report MSUHEP-20704, Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, 2001. See also http://cosy.pa.msu.edu.

[10] V. Balbekov. Private communication, 2002.

Solenoid elements in COSY INFINITY

K. Makino†and M. Berz‡

† Department of Physics, University of Illinois at Urbana-Champaign, 1110 W. Green Street, Urbana, IL 61801-3080, USA
‡ Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA

E-mail: `makino@uiuc.edu, berz@msu.edu`

Abstract. We describe a full array of solenoidal elements in the high order transfer map computation code COSY INFINITY, starting from a loop coil to any superposition of thick straight solenoids. Since the fringe field of coils extends very far longitudinally, and at the same time contains various nonlinearities due to the longitudinal dependence of the field, accurate but fast field computation is necessary. In COSY, the 3D fields along the integration of transfer map through such an element are computed using a Differential Algebra based PDE solver, which is very fast and only requires information about the analytical axial potential. By examples, we illustrate the feature of each solenoidal element and how to simulate realistic beamlines containing combinations of solenoids and other elements.

1. Introduction

The differential algebraic (DA) methods [1, 2] allow the efficient computation and manipulation of high order Taylor transfer maps. When integrating transfer maps through electromagnetic fields, the full 3D fields are computed as part of each integration time step using DA PDE (partial differential equation) solvers. First, we address the mechanism of the method of DA fixed point PDE solvers, and as will be seen, the method is very compact and fast, and only requires the analytical axial potential for solenoidal elements.

After developing the theoretical background, we illustrate a variety of solenoidal elements available in COSY INFINITY [3], and study their features. Compared to multipole electromagnetic elements as dipoles, quadrupoles and so forth, the fringe fields of solenoids extend for a long distance. Particularly because of this long extension of the fringe fields, in practice it is important to be able to efficiently combine the fields consisting of several solenoidal coils, which are also treated with the DA PDE solvers. This often even simplifies the simulation efforts due to the shortened fringe fields created by the cancellation of fields of counteracting coils, as will be seen in an example from a muon beam cooling cell [4]. At last we show some examples of very atypical uses of standard electromagnetic elements, producing solenoidal fields from non-solenoidal elements, or producing bending fields from solenoidal elements. Such beam optical systems are particularly important in several components of neutrino factory designs [4].

2. DA fixed point PDE solvers

The idea of differential algebraic (DA) methods [1, 2, 5] is based on the observation that it is possible to extract more information about a function than its mere values on computers. One can introduce an operation T denoting the extraction of the Taylor coefficients of a pre-specified order n of the function $f \in C^n(R^v)$. In mathematical terms, T is an equivalence relation, and the application of T corresponds to the transition from the function f to the equivalence class $[f]$ comprising all those functions with identical Taylor expansion in v variables to order n; the classes are apparently characterized by the collection of Taylor coefficients. Since Taylor coefficients of order n for sums and products of functions as well as scalar products with reals can be computed from those of the summands and factors, the set of equivalence classes of functions can be endowed with well-defined operations, leading to the so-called Truncated Power Series Algebra (TPSA) [6, 7]. More advanced tools address the composition of functions, their inversion, solutions of implicit equations, and the introduction of common elementary functions[1]. For treatment of ODEs and PDEs, the power of TPSA can be enhanced by the introduction of derivations ∂ and their inverses ∂^{-1}, corresponding to the differentiation and integration on the space of functions, resulting in the Differential Algebra $_nD_v$. This structure allows the direct treatment of many questions connected with differentiation and integration of functions, including the solution of the ODEs $d\vec{x}/dt = \vec{f}(\vec{x}, t)$ describing the motion and PDEs describing the fields [5].

To any element $[f] \in {}_nD_v$ we define the depth $\lambda([f])$ as

$$\lambda([f]) = \begin{cases} \text{Order of first nonvanishing derivative of } f & \text{if } [f] \neq 0 \\ n+1 & \text{if } [f] = 0 \end{cases}.$$

In particular, any function f that does not vanish at the origin has $\lambda([f]) = 0$.

Let $\mathcal{O}$ be an operator on the set $M \subset {}_nD_v^m$, where $_nD_v^m$ is the set describing vector functions $\vec{f} = (f_1, ..., f_m)$ from R^v to R^m. Then we say that $\mathcal{O}$ is contracting on M if for any $\vec{a}$, $\vec{b} \in M$ with $\vec{a} \neq \vec{b}$,

$$\lambda(\mathcal{O}(\vec{a}) - \mathcal{O}(\vec{b})) > \lambda(\vec{a} - \vec{b}).$$

In practical terms this means that after application of $\mathcal{O}$, the derivatives in $\vec{a}$ and $\vec{b}$ agree to a higher order than before application of $\mathcal{O}$. For example, the antiderivation ∂_k^{-1} is a contracting operator. Contracting operators satisfy a fixed point theorem:

Theorem 1 *(DA Fixed Point Theorem) Let $\mathcal{O}$ be a contracting operator on $M \subset {}_nD_v$ that maps M into M. Then $\mathcal{O}$ has a unique fixed point $a \in M$ that satisfies the fixed point problem $a = \mathcal{O}(a)$. Moreover, let a_0 be any element in M. Then the sequence $a_k = \mathcal{O}(a_{k-1})$ for $k = 1, 2, ...$ converges in finitely many steps (in fact, at most $(n+1)$ steps) to the fixed point a.*

The fixed point theorem is of great practical usefulness since it assures the existence of a solution, and moreover allows its exact determination in a very simple way in finitely many steps. The proof of the theorem can be found in [1]. The DA fixed point theorem has many useful applications, in particular a rather straightforward solution of ODEs and PDEs [5].

The direct availability of the derivation ∂ and its inverse ∂^{-1} allows to devise efficient numerical PDE solvers of any order. The DA fixed point theorem allows one to solve PDEs iteratively in finitely many steps by rephrasing them in terms of a fixed point problem. The details depend on the PDE at hand, but the key idea is to eliminate differentiation with respect to one variable and replace it by integration. As an example, consider the rather general PDE

$$a_1 \frac{\partial}{\partial x}\left(a_2 \frac{\partial}{\partial x}V\right) + b_1 \frac{\partial}{\partial y}\left(b_2 \frac{\partial}{\partial y}V\right) + c_1 \frac{\partial}{\partial z}\left(c_2 \frac{\partial}{\partial z}V\right) = 0,$$

where a_1, a_2, b_1, b_2, c_1 c_2 are functions of x, y, z. The PDE is re-written as

$$V = V|_{y=0} + \int_0^y \frac{1}{b_2} \left\{ \left(b_2 \frac{\partial V}{\partial y} \right) \bigg|_{y=0} - \int_0^y \left[\frac{a_1}{b_1} \frac{\partial}{\partial x} \left(a_2 \frac{\partial V}{\partial x} \right) + \frac{c_1}{b_1} \frac{\partial}{\partial z} \left(c_2 \frac{\partial V}{\partial z} \right) \right] dy \right\} dy.$$

The equation is now in fixed point form. Now assume the derivatives of V and $\partial V / \partial y$ with respect to x and z are known in the plane $y = 0$. If the right hand side is contracting with respect to y, the various orders in y can be calculated by mere iteration.

As a particularly important example, consider the Laplace equation. It can be represented in general curvilinear coordinates [8, 9]. In the special case of a curvilinear coordinate system, the Laplace equation is obtained as [8, 9]

$$\triangle V = \frac{1}{1+hx} \frac{\partial}{\partial x} \left[(1+hx) \frac{\partial V}{\partial x} \right] + \frac{\partial^2 V}{\partial y^2} + \frac{1}{1+hx} \frac{\partial}{\partial s} \left(\frac{1}{1+hx} \frac{\partial V}{\partial s} \right) = 0.$$

In the case of a straight section, where $h = 0$, it reduces to nothing but the Cartesian Laplace equation. The fixed point form of the Laplace equation in the planar curvilinear coordinates is

$$V = V|_{y=0} + \int_0^y \left(\frac{\partial V}{\partial y} \right) \bigg|_{y=0} dy$$
$$- \int_0^y \int_0^y \left\{ \frac{1}{1+hx} \frac{\partial}{\partial x} \left[(1+hx) \frac{\partial V}{\partial x} \right] + \frac{1}{1+hx} \frac{\partial}{\partial s} \left(\frac{1}{1+hx} \frac{\partial V}{\partial s} \right) \right\} dy dy.$$

In this form, the right hand side has the interesting property that, regardless of what function V is inserted, the parts not depending on y are reproduced exactly, since all integrals introduce y dependence. Because of the integral operation, for a given choice of x and s and considering only the y dependence, the right hand side is contracting. In COSY INFINITY [3], the planar curvilinear Laplace equation is solved by the following very compact code

```
POLD := P ;
HF := 1+H*DA(IX) ;
HI := 1/HF ;
LOOP I 2 NOC+2 2 ;
   P := POLD - INTEG(IY,INTEG(IY,
      HI*( DER(IX,HF*DER(IX,P)) + DER(IS,HI*DER(IS,P)) ) )) ;
   ENDLOOP ;
```

Here the boundary condition $V|_{y=0} + \int_0^y (\partial V/\partial y)|_{y=0} dy$ is provided through the incoming form of P, which is obtained using the DA expression in COSY. The DA fixed point iteration converges to the solution potential P in finitely many steps. DA(IX) represents the identity for x, NOC is the current transfer map computation order, and DER(I,...) and INTEG(I,...) correspond to the DA derivative and the DA anti-derivative operations with respect to the variable specified by the first argument I, namely "∂_{x_I}" and "$\int_0^{x_I} dx_I$". The full 3D field is derived from the solution potential P, using the elementary DA derivations ∂_x, ∂_y and ∂_s. In coded form, we have

```
BX := DER(IX,P) ;
BY := DER(IY,P) ;
BZ := DER(IS,P) ;
```

The advantages of the method are:

- Only the field in the midplane is needed
- The resulting field will always satisfy the stationary Maxwell equations

- The method works to any order

Another important coordinate system often suitable for computations under consideration are the cylindrical coordinates, in which the Laplace equation takes the simple form

$$\triangle V = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial V}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 V}{\partial \phi^2} + \frac{\partial^2 V}{\partial s^2} = 0.$$

If V does not depend on ϕ, namely V is rotationally symmetric, as in solenoid magnets, the fixed point form of the Laplace equation is simplified to

$$V = V|_{r=0} - \int_0^r \frac{1}{r}\int_0^r r\frac{\partial^2 V}{\partial s^2}drdr,$$

and the right hand side is contracting with respect to r. Since we are only interested in cases in which $V(s,r)$ is expressed in DA, if $\partial^2 V/\partial s^2$ is nonzero, the integral $\int_0^r r\partial^2 V/\partial s^2 dr$ contains r to a positive power. Thus, the factor $1/r$ in the outer integral simply lowers the power of r by one, and the right hand side of the fixed point form can be evaluated in DA without posing trouble. To perform the DA fixed point iteration for the purpose of obtaining the full potential $V(s,r)$, one only needs to prepare the on-axis potential expression $V(s,r)|_{r=0}$ as the boundary condition.

3. Single coil solenoid elements

We showed in the last section that for solenoid magnets, the DA PDE solver only requires an analytical expression of the potential on axis. In this section, we provide the on-axis field and potential of some solenoidal elements in the code COSY INFINITY [3] and discuss their features. In the following, R is the radius of the coil, R_1 and R_2 are the inner and outer radii of the coil if non-zero thickness is considered, I is the current, n is the number of turns per meter, and the coil extends from $s = 0$ to $s = l$. While the on-axis forms are easily obtained, the out of axis forms can usually not be represented in closed form as they involve elliptic integrals; thus the ability of the DA PDE solver to generate the power series representation of the full 3D field to any order is very useful. Once the on-axis field $B_z(s)$ is known, an on-axis potential $V(s)$ can be determined via $V(s) = \int B_z(s)ds$. It is customary to omit the minus sign known for the electric case for magnetic scalar potential.

The first solenoid element is the current loop, consisting of a thin circular wire of radius R carrying the current I.

Current loop (COSY element CMR)

$$B_{z,\mathrm{CMR}}(s) = \frac{\mu_0 I}{2R}\frac{1}{\left[1 + (s/R)^2\right]^{3/2}}, \quad V_{\mathrm{CMR}}(s) = \frac{\mu_0 I}{2R}\frac{s}{\sqrt{1 + (s/R)^2}}.$$

The derivation of $B_{z,\mathrm{CMR}}(s)$ can be found in various text books on electromagnetism, for example, see eq. (5.40) (with $\theta = 0$) in [10].

The next element is a thin coil extending from $s = 0$ to $s = l$, made up of a single layer of thin wire carrying current I with n windings per meter.

Thin solenoid (COSY element CMSI)

$$B_{z,\mathrm{CMSI}}(s) = \frac{\mu_0 In}{2}\left(\frac{s}{\sqrt{s^2 + R^2}} - \frac{s-l}{\sqrt{(s-l)^2 + R^2}}\right),$$

$$V_{\mathrm{CMSI}}(s) = \frac{\mu_0 In}{2}\left(\sqrt{s^2 + R^2} - \sqrt{(s-l)^2 + R^2}\right).$$

$B_{z,\text{CMSI}}(s)$ can be obtained by integrating individual current loops positioned from $s = 0$ to $s = l$ as $B_{z,\text{CMR}}(s) = n \cdot \int_0^l B_{z,\text{CMR}}(s-x)dx$. Derivations of $B_{z,\text{CMSI}}(s)$ can also be found in various text books, for example, see problem 5.2 in [10], where $\cos\theta_1 = s/\sqrt{s^2 + R^2}$ and $\cos\theta_2 = -(s-l)/\sqrt{(s-l)^2 + R^2}$ in our case. It is worth observing that in the middle, we have

$$B_{z,\text{CMSI}}(l/2) = \frac{\mu_0 In}{2} \cdot \frac{l}{\sqrt{(l/2)^2 + R^2}},$$

and if $l \gg R$, the field approaches the expected asymptotic value $B_{z,\text{CMSI}}(l/2) \to \mu_0 In$.

The next element is a thick coil extending longitudinally from $s = 0$ to $s = l$, and radially from $r = R_1$ to $r = R_2$, wound out of wire with a winding density n and carrying current I.

Thick solenoid (COSY element CMST)

$$B_{z,\text{CMST}}(s) = \frac{\mu_0 In}{2(R_2 - R_1)} \left[s\log\left(\frac{R_2 + \sqrt{R_2^2 + s^2}}{R_1 + \sqrt{R_1^2 + s^2}}\right) - (s-l)\log\left(\frac{R_2 + \sqrt{R_2^2 + (s-l)^2}}{R_1 + \sqrt{R_1^2 + (s-l)^2}}\right) \right],$$

$$V_{\text{CMST}}(s) = \frac{\mu_0 In}{4(R_2 - R_1)} \left[s^2\log\left(\frac{R_2 + \sqrt{R_2^2 + s^2}}{R_1 + \sqrt{R_1^2 + s^2}}\right) - (s-l)^2\log\left(\frac{R_2 + \sqrt{R_2^2 + (s-l)^2}}{R_1 + \sqrt{R_1^2 + (s-l)^2}}\right) \right.$$

$$\left. + R_2\sqrt{R_2^2 + s^2} - R_1\sqrt{R_1^2 + s^2} - R_2\sqrt{R_2^2 + (s-l)^2} + R_1\sqrt{R_1^2 + (s-l)^2} \right].$$

The derivation of the field for the thick solenoid is similar in spirit to the derivation of $B_{z,\text{CMSI}}(s)$. In fact, we have $B_{z,\text{CMST}}(s) = 1/(R_2 - R_1) \cdot \int_{R_1}^{R_2} B_{z,\text{CMSI}}(s, R)dR$. The factor $1/(R_2 - R_1)$ is necessary to maintain the meaning of n as the number of windings per meter, i.e. In is the total current per meter. We observe that in the middle of the solenoid, we have

$$B_{z,\text{CMST}}(l/2) = \frac{\mu_0 In}{2(R_2 - R_1)} l\log\left(\frac{R_2 + \sqrt{R_2^2 + (l/2)^2}}{R_1 + \sqrt{R_1^2 + (l/2)^2}}\right).$$

If $l \gg R_1, R_2$, the log part in the right hand side of $B_{z,\text{CMST}}(l/2)$ above is approximated as follows:

$$\log\left(\frac{R_2 + l/2}{R_1 + l/2}\right) \approx \log\left[\left(1 + \frac{2R_2}{l}\right)\left(1 - \frac{2R_1}{l}\right)\right] \approx \log\left[1 + \frac{2}{l}(R_2 - R_1)\right] \approx \frac{2}{l}(R_2 - R_1).$$

So the field approaches the asymptotic value $B_{z,\text{CMST}}(l/2) \to \mu_0 In$.

Traditionally, also various other approximate representations of fields have been used (see for example [11, 12, 13, 14]) that are based on particularly simple forms for the fields or potentials; of these approximations, we have implemented two. One of them is the Glaser lens, which is frequently used to approximately describe a lens made of a coil with finite but short length and finite but small thickness.

Glaser lens (COSY element CML)

$$B_{z,\text{CML}}(s) = \frac{B_0}{1 + (s/R)^2}, \quad V_{\text{CML}}(s) = B_0 R \arctan(s/R).$$

The other frequently used approximation is for an extended coil of length l of small thickness of the form.
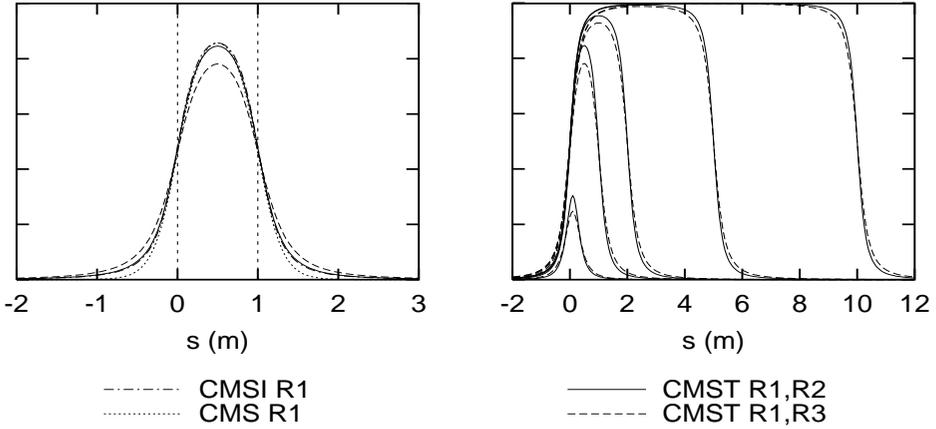
Figure 1. The axial field profile $B_z(s)$ of various COSY solenoid elements. Left: Comparison between the thin element CMSI, the tanh approximation element CMS, and the thick element CMST. The length is $l = 1$m. Right: Comparison of different lengths $l = 0.3$m, 1m, 2m, 5m and 10m for CMST. The (inner) radius is $R_1 = 0.3$m. For CMST, the outer radii $R_2 = 0.33$m and $R_3 = 0.5$m are compared. The field strength is scaled relative to the asymptotic value $\mu_0 In$.

Thin solenoid (tanh approximation) (COSY element CMS)

$$B_{z,\text{CMS}}(s) = \frac{B_0}{2\tanh(l/2R)}\left[\tanh(s/R) - \tanh((s-l)/R)\right], \quad B_{z,\text{CMS}}(l/2) = B_0.$$

$$V_{\text{CMS}}(s) = \frac{B_0}{2\tanh(l/2R)}R\left[\log(\cosh(s/R)) - \log(\cosh((s-l)/R))\right].$$

Here the hyperbolic tangents are used as simple approximations for the rise and fall-off of the field at $s = 0$ and $s = l$, respectively.

As the analytical expressions of the on-axis field $B_z(s)$ and the potential $V(s)$ indicate, the profiles of s-dependence are characterized by the ratio of the length l and the aperture R. Figure 1 shows the axial field profile $B_z(s)$ of the elements CMSI, CMS, and CMST of length $l = 1$m with the radius $R_1 = 0.3$m and the outer radius $R_2 = 0.33$m or $R_3 = 0.5$m (Left), and the field profile of CMST of different lengths $l = 0.3$m, 1m, 2m, 5m and 10m (Right). The field strength is scaled relative to the asymptotic value $\mu_0 In$, and B_0 for CMS is given by $B_{z,\text{CMSI}}(l/2)$. As the length study picture shows, many realistic solenoids do not even reach maximum fields close to the asymptotic value $\mu_0 In$.

The tanh approximation as in the element CMS is commonly used because the on-axis field drops more swiftly in the fringe region compared to the pure theoretical fields as CMSI and CMST, which simplifies the simulation effort. On the other hand, the discrepancy from the actual field becomes very large particularly for sufficiently thick solenoids, which are important in practice because of their ability to provide high field strength. Figure 2 shows the full 3D field distributions $B_z(s,r)$ and $B_r(s,r)$ of the thick element CMST of length $l = 1$m with the radii $R_1 = 0.3$m and $R_2 = 0.33$m. The full 3D field is derived only from the on-axis potential $V(s)$ via the DA fixed point PDE solver.

Some matrix elements of fifth order transfer maps of these solenoid elements are listed below in COSY notation for comparison, showing the differences in the linear and nonlinear behavior. Similar to before, the length is $l = 1$m, the (inner) radius is $R = R_1 = 0.3$m, and the
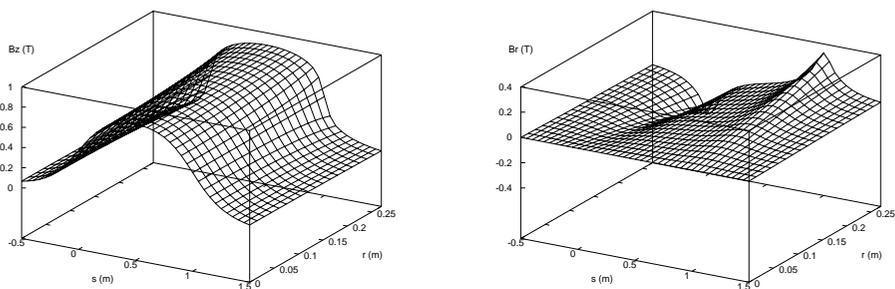
Figure 2. The full 3D field distributions $B_z(s,r)$ (Left) and $B_r(s,r)$ (Right) of the thick element CMST (l =1m, R_1 =0.3m, R_2 =0.33m) derived only from the on-axis potential $V(s) = \int_s B_z(s)$ using the DA fixed point PDE solver. The field strength is scaled relative to the asymptotic value $\mu_0 In$.

outer radii used in CMST are R_2 =0.33m and R_3 =0.5m. The magnet strength is adjusted to have $\mu_0 In = 1$Tesla, and B_0 for CMS is scaled to agree with $B_{z,\mathrm{CMSI}}(l/2)$.

```
CMSI:
      x_f              a_f              y_f              b_f           xayb
   0.7937713      -0.1431056      -0.4573031       0.8243452E-01    1000
   0.8436961       0.7938225      -0.4860026      -0.4572141        0100
      ...             ...             ...             ...           ....
  -0.4911018E-01 -0.4490695      -0.3866000      -0.8418896         0014
   0.2409343      -0.6962362E-01  0.1265599      -0.2574890         0005

CMS:
   0.8152097      -0.1403695      -0.4257566       0.7331028E-01    1000
   0.8629072       0.8152097      -0.4506673      -0.4257566        0100
      ...             ...             ...             ...           ....
  -0.2386703E-01 -0.3563119      -0.3615158      -0.7512712         0014
   0.2331341      -0.5329836E-01  0.1522912      -0.2406150         0005

CMST with R1, R2:
   0.7953672      -0.1399478      -0.4582230       0.8061561E-01    1000
   0.8446078       0.7954184      -0.4865284      -0.4581341        0100
      ...             ...             ...             ...           ....
  -0.4865521E-01 -0.3911095      -0.3711714      -0.7356460         0014
   0.2426523      -0.5600330E-01  0.1324498      -0.2326217         0005

CMST with R1, R3:
   0.8034393      -0.1239508      -0.4628757       0.7140094E-01    1000
   0.8494714       0.8034915      -0.4893318      -0.4627851        0100
      ...             ...             ...             ...           ....
  -0.4406906E-01 -0.2077903      -0.3039852      -0.3947231         0014
   0.2456954      -0.1146626E-01  0.1444507      -0.1507266         0005
```

The thinner case of CMST with R_1 and R_2 agrees with the map of CMSI to approximately two digits. On the other hand, the map of CMS agrees with that of CMSI to approximately only one digit. This again shows that the tanh approximation element CMS has to be used with care.

4. Multiple coil solenoids

If a system consists of several solenoids, it is often crucial to be able to treat the whole solenoidal system as one element with superimposed solenoidal field, because the fringe field extension is particularly long for solenoids. In this section, we present such an example from muon beam ionization cooling systems. The example is a 2.75m sFOFO muon beam ionization cooling cell in Muon Feasibility Study II [4]. There are three coils in the cell, and the starting position of each coil is 0.175m, 1.21m, and 2.408m. The outer two coils are 0.167m long with the inner and outer radii 0.33m and 0.505m and the current density is 75.2A/mm^2. The middle one is 0.33m long with the radii 0.77m and 0.85m and the current density 98.25A/mm^2 [4]. The pictures in Figure 3 show the coil layout and the superimposed axial field profile $B_z(s)$ as well as the full 3D field distributions of $B_z(s,r)$ and $B_r(s,r)$ that
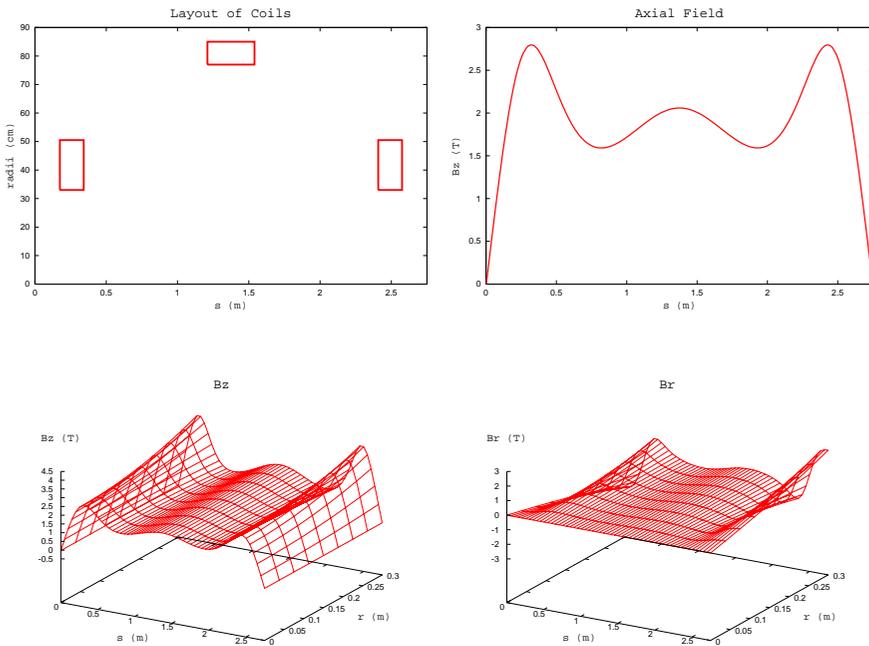
Figure 3. The coil layout, the superimposed axial field profile $B_z(s)$, and the full 3D field distributions of $B_z(s,r)$ and $B_r(s,r)$ of a 2.75m sFOFO cell [4]. $B_z(s,r)$ and $B_r(s,r)$ are derived only from the on-axis potential using the DA fixed point PDE solver.

are obtained via the DA fixed point PDE solver. Since the thickness of coils is very large, the superimposed field maintains high strength throughout the cell except for the ends of the cell, where the axial field drops to zero due to the alternating field direction in the preceding and following cells. In fact, the drop of the axial field to zero simplifies the computation of high order transfer maps by beginning and ending the computation at a zero crossing of the field, although of course the original reason for the design need for field flips is to enhance cooling efficiency [4].

The muon beam cooling cell has accelerating cavities and absorbers situated inside the solenoids. We can treat such systems with the transfer map in a split operator framework approach by slicing the cell into short pieces so that the effects of each element can be superimposed by inserting a short negative drift [5]. For example, the 2.75m sFOFO cell is sliced into about 80 pieces [5].

5. Vertical solenoidal fields and misalignment

The modern concept of designing beam optical systems is to perform each of the common tasks of bending, focusing, and nonlinear correction by separate elements; but there are situations in which this simple concept would lead to significant sacrifices. For example, the beam optical systems for rare and short-lived particles often require complicated setups including combined function electromagnetic elements to manipulate the beam efficiently, an example of which has been provided above. Sometimes misalignment by displacement and tilts of regular kind of single function elements can achieve the necessary combined fields. The complication brought by the misalignment has to be studied carefully, because sometimes it may lead to unexpected beam dynamics.

We show an example using a design of the $60°$ arc cell of a compact muon storage ring [4]. The cell was designed to achieve a very high degree of compactness using half overlapping coils as shown in Figure 4. The double layered part has a strong dipole field of 7 Tesla, and the single layered part has a dipole field component of around half that strength. In addition, the latter region exhibits a skew quadrupole field which is used for focusing purposes, as well as small high order multipole components introduced mostly because of the limited horizontal width of the coils. The longitudinal magnet layout produces a longitudinal field component, breaking midplane symmetry, and the on-axis longitudinal field, in other words the solenoidal filed, is as strong as 2.2 Tesla; Figure 4 shows the strength as a function
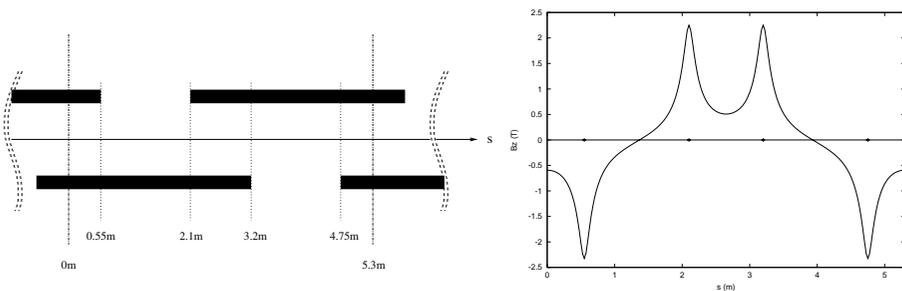
Figure 4. The longitudinal magnet layout (Left) and the on-axis field profile of solenoidal field component $B_z(s)$ (Right) of a design of the $60°$ arc cell of a compact muon storage ring [4].

of position s. By using the technique discussed in the previous section for superimposed solenoidal fields, the effects of the solenoidal field of the $60°$ arc cell can be included in the beam dynamics study.

Due to limitations of space, we refrain from providing details about the results of simulations of the resulting particle dynamics; various such results are given in [4].

Another interesting example of misalignment is a design of a muon beam cooling ring using short solenoids with large aperture [15] that are tilted horizontally to deflect the reference orbit of the beam and overall lead to the possibility of operating without dipoles [16]. Utilizing the COSY commands for misalignment, it is also possible to analyze such bending beamlines consisting of only solenoidal elements.

Acknowledgments

References

[1] M. Berz. *Modern Map Methods in Particle Beam Physics*. Academic Press, San Diego, 1999.

[2] M. Berz. Differential algebraic description of beam dynamics to very high orders. *Particle Accelerators*, 24:109, 1989.

[3] M. Berz and K. Makino. COSY INFINITY Version 8.1 - user's guide and reference manual. Technical Report MSUHEP-20704, Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, 2001. See also http://cosy.pa.msu.edu.

[4] S. Ozaki et al. for the Muon Collaboration. Feasibility study-II of a muon-based neutrino source. Technical Report 52623, Muon Collider Collaboration, BNL, 2001.

[5] K. Makino, M. Berz, D. Errede, and C. J. Johnstone. High order map treatment of superimposed cavities, absorbers, and magnetic multipole and solenoid fields. *Nuclear Instruments and Methods*, in print, 2003.

[6] M. Berz. The new method of TPSA algebra for the description of beam dynamics to high orders. Technical Report AT-6:ATN-86-16, Los Alamos National Laboratory, 1986.

[7] M. Berz. The method of power series tracking for the mathematical description of beam dynamics. *Nuclear Instruments and Methods*, A258:431, 1987.

[8] K. Makino. *Rigorous Analysis of Nonlinear Motion in Particle Accelerators*. PhD thesis, Michigan State University, East Lansing, Michigan, USA, 1998. Also MSUCL-1093.

[9] K. Makino and M. Berz. Perturbative equations of motion and differential operators in nonplanar curvilinear coordinates. *International Journal of Applied Mathematics*, 3,4:421–440, 2000.

[10] J. D. Jackson. *Classical Electrodynamics*. Wiley, New York, 1975.

[11] W. Glaser. *Grundlagen der Elektronenoptik*. Springer, Wien, 1952.

[12] X. Jiye. *Aberration Theory in Electron and Ion Optics*. Advances in Electronics and Electron Physics, Supplement 17. Academic Press, Orlando, Florida, 1986.

[13] P. W. Hawkes and E. Kasper. *Principles of Electron Optics*, volume 1-2. Academic Press, London, 1989.

[14] K. G. Steffen. *High Energy Beam Optics*. Wiley-Interscience, New York, 1965.

[15] J. S. Berg, R. C. Fernow, and R. B. Palmer. An alternating solenoid focused ionization cooling ring. The Muon Collider and Neutrino Factory Notes MUC-NOTE-COOL-THEORY-0239, Fermi National Accelerator Laboratory, 2002. See http://www-mucool.fnal.gov/notes/notes.html.

[16] J. S. Berg, R. C. Fernow, and R. B. Palmer. Private communication, 2002.

Experience during the SLS commissioning

M. Muñoz, M. Böge, J. Chrin, A. Streun

Paul Scherrer Institut - SLS

Abstract. During the commissioning and first years of SLS operations we have made extensive use of an accelerator machine model. The model is based on a C port of the original TRACY-2 Pascal code. This model is connected to the applications in the control room through a CORBA interface. This approach has facilitated easy integration of the accurate machine model with the operation of the machine, and allows us to use the well-tested TRACY-2 procedures online. A good example of this approach is the orbit correction system, but several other utilities have also been developed.

1. Description of the SLS

The Swiss Light Source (SLS) is the most recent of a series of 3rd generation light sources to be commissioned [1] and that has come in operation. It is located at the Paul Scherrer Institut. The Swiss Government approved the SLS project in September 1997. The construction of main building started in June 1998 and the commissioning of the machine began in January 2000, and was finished successfully in August 2001. At present over 70% of operation time is schedule for user experiments.

An initial set of four insertion devices have been installed in the storage ring to cover users demands in the range of 10 eV to 40 keV photon energy:

1. The high field wiggler W61 (5-40 keV), for material science.
2. The in-vacuum undulator U24 (8.14 keV) for protein crystallography.
3. The electromagnetic twin undulator UE212 (8-800 eV) for surface and interface spectrography.
4. The Sasaki/APPLE II type twin undulator UE56 (90 eV-3 keV) for surface and interface microscopy.

The light source consists of a 100 MeV linac, a novel type of full energy booster synchrotron, and the main storage ring. The storage ring is a 12 cell TBA, with a circumference of 288 m, providing a natural emittance of 5 nm at 2.4 GeV. The current mode of operations is 250 mA of stored current in top-up mode, and it is planned to switch to 400 mA top-up in the 3rd quarter of 2003.

To reach such a successful status, a number of innovative design features and subsystem were incorporated in the project: digital power supplies, digital BPM systems, advanced booster design, flexible injection system (these last features allows for top-up injection as the routine mode of operation).

The SLS storage ring consists of 12 Triple-Bend Achromats (TBA with 8°/14°/8° bends) with six short straight sections of 4 meters, three medium sections of 7 m and three long ones of 11 m. Figure 1 shows the optical functions for a $1/6^{th}$ of the ring.
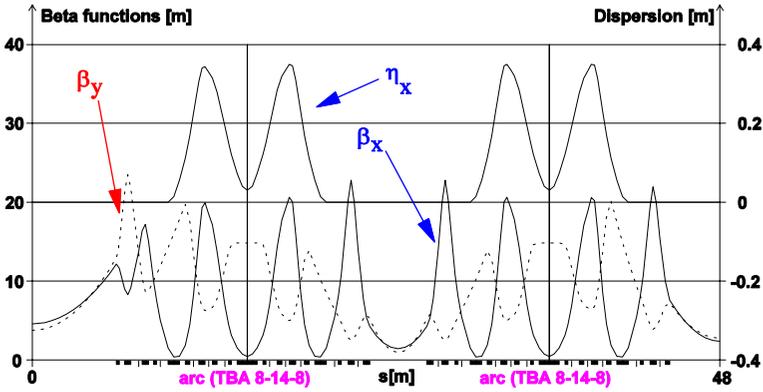
Figure 1. Storage ring optics for a 6th of the ring (L/2-TBA- S-TBA- M/2)

2. Beam dynamics tools

During the design and commissioning of the SLS we used and developed tools for simulation and operation of the light source. We present a short review of the most important ones.

2.1 OPA

The basic design tool used at the SLS is the OPA program. Is a TurboPascal code, written by A. Streun [2] based on a code developed by the DELTA-group at the University of Dortmund. Additional procedures and algorithms were taken from TRACY-2 and implemented. In particular, J. Bengtsson added procedures for optimize the sextupole configuration in order to reduce non-linear effects.

OPA is a very user-friendly program, very useful for start a storage ring design from scratch. It provides a clean graphical interface, run in almost any DOS machine and does not require a lot of memory (640 kB is enough) or CPU (a 80286 processors is enough). In spite of the low requirements, OPA provides good tracking calculation of the optical functions and other basic tools with either with a matrix model or a 4th order simplectic integrator [5] at a very good speed and accuracy.

However the program is not perfect and some features are missing: it does not include simulation of errors in the magnetic lattice, does not include skew quadropoles or vertical bending magnet nor high order multipolar content in the magnets (only sextupoles are simulated) and do not include synchrotron motion (only optics for fixed dp<>0 available). Excluding these shortcomings (that are not very important for the design of a synchrotron light source), the list of features is very complete and the strength of the programs lies in the simplicity and ease of use, the accurate simulation of the particle motion (good prediction of dynamic apertures, tune shift with amplitudes and others). All this makes OPA a very convenient tool to do the initial design of a light source.

2.2 C-TRACY-2

The main tracking and simulation tool used at the SLS is a C port of the TRACY-2 code. The original TRACY-2 code was written by J. Bengtsson [3], and consisted in a Pascal-S interpreter and a set of Pascal routines for simulation. Due to the need to migrate from VMS to Linux system, the Pascal source was converted to a C library and the Pascal-S interpreter was removed. In that way the original Pascal-S input program is replaced by a C main that is then linked with the TRACY-2 library [4].

The conversion was performed utilizing the GNU p2c and f2c tools. The main work was spent on the reorganization of routines and data structures. Most of the routines available in the Pascal-S version are still there, although some of them have slightly changed parameter lists, and some bugs were corrected. The biggest changes was the addition of routines for performing orbit correction using SVD (**S**ingular **V**alue **D**ecomposition) tools, that are the base of the Fast Orbit Feedback system for the SLS.

The conversion from the limited capabilities of the Pascal-S interpreter to a full-fledged C-library offers some advantages: the more obvious one is speed in the calculations, as we have eliminated the overhead due to the compiler. However, the increase of flexibility proved to be most useful one during the commissioning of the machine. We were able to interface the TRACY-2 library to the SLS control system using CORBA as a middleware.

TRACY-2 implements the usual matrix mode calculations or the 4th order Ruth' symplectic integrator [5]. It includes optical function calculations with either method, particle tracking with synchrotron radiation and RF cavities, offers a very good implementation of magnetic errors and in general includes everything that is required to simulate the running of the SLS facility (for example modules for girder motion or the routines used for orbit correction in the control room).

However the code is not perfect, and some work is still required in some areas. In particular a better simulation of the insertion devices is required, and tools for evaluation of one-turns maps are missing.

2.3 The CORBA system

The high level control system employed at the SLS is EPICS [6] with Linux consoles. A detailed description of it can be found in [7]. It became clear very early in the planning of the SLS commissioning that it would be very useful to have a machine model integrated in the SLS control system. As described with more details in reference the solution chose to implement the access to model is based in the CORBA (**C**ommon **O**bject **R**equest **B**roker **A**rchitecture) system architecture [8].
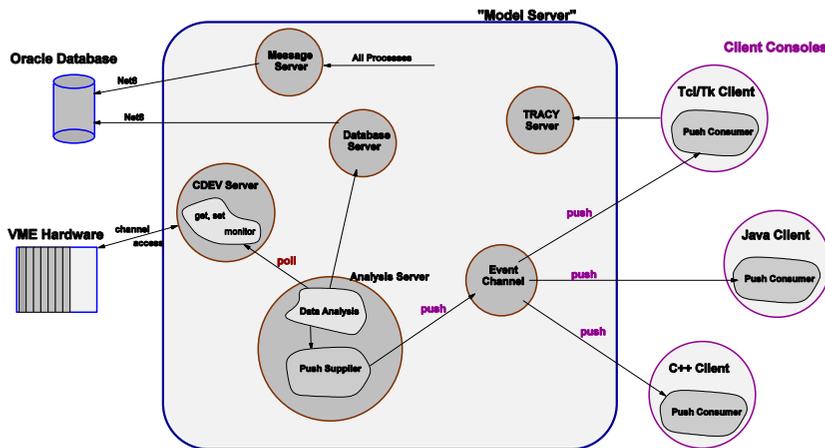
Figure 2. Conceptual design of the client-server software components.

CORBA is a standard that provides a mechanism for defining interfaces between distributed components. Its most distinguished assets are platform independence, in so far as the platform hosts a CORBA **O**bject **R**equest **B**roker (ORB) implementation, and language independence, as ensured through the use of the **I**nterface **D**efinition **L**anguage

(IDL). The latter feature is of particular interest to SLS beam dynamics API developers as it provides for the option between high-level application languages, in particular Tcl/tk and Java.

Figure 2 shows the implementation of the client/server software components. Three main components are visible: a database server, to access the SLS Oracle database; the CDEV server, which allow high-level applications to access the control system; and the TRACY servers. Those ones provide access to the machine model to the diverse high-level applications.

Developing TRACY servers is not a difficult task. Once the functionality has been implemented using TRACY-2 and tested off-line, writing a C++ wrapper and integrating the CORBA functionality is an easy task. Examples of TRACY serves in use are the closed orbit correction system, an optics model for the transfer lines and a tune measurement system.

3. Model/Real machine agreement

We used the lattice files of TRACY-2 and the measured magnet calibration factors to evaluate the required currents of the magnets. We also developed some applications that allow us to change the tunes of the machine using the same procedures used in TRACY-2 (the SLS lattice is a complex one, and to change the tunes in a controlled way it is required to adjust all the quadrupoles families).

Using the theoretical model values, the machine presented a stable orbit at the first injection, with only some small adjustements in the tunes.

1. A vertical tune shift of +0.2, due to TRACY-2 using a wrong model for the edge focusing in bending magnets. This produced a distributed reduction of the focusing strength.
2. Some residual beta-beat of up to 15%, due to differences between the individual members of a magnet family and the model used for the family.
3. A discrepancy on the energy of the machine (1% higher that it should be). The reason for that is still unclear, but it is probably due to a wrong dipole calibration.

More important, the non-linear components (sextupole families) were also set to the theoretical values. In the SLS case, eight families of sextupoles are employed to provide the required dynamic aperture and energy acceptance required. If the model is not good trying to set them to the correct values could be a time consuming and difficult task. In our case, the predicted values of the sextupoles provided the model predicted chromaticity. Applications to change the chromaticity and the sextupoles settings following the predictions of the model were also developed and produced the predicted effect.

3.1 Correcting the beta beat

An example of the good agreement between the machine model and the reality is the measurement and correction of the beta-beat. At the SLS, each one of the 174 quadrupoles is independently powered by a high-resolution power supply. This allows us to measure the beta functions in each of them, using the tune shift associated to a current change. This was already a test of the integration between diverse high-level applications.

After the tune change in function of the current was measured [9] and the beta functions in each one evaluated (see ref for more details), the ideal machine model was fitted to provide the same perturbation (using only errors in the quadrupoles). This provided us with a correction scheme for the beta beat. Figure shows the residual beta-beat after the first correction.
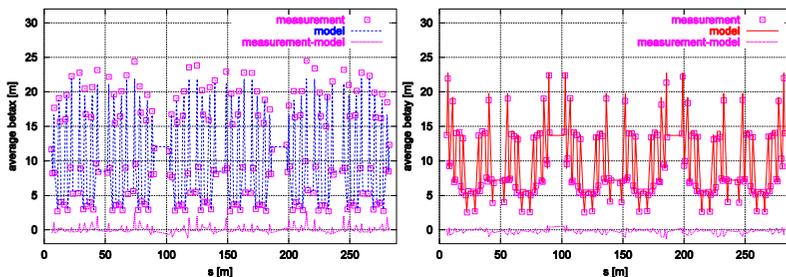
Figure 3. Measured average beta functions (squares) at the location of 174 quadrupoles in comparison to the model of the unperturbed optics (solid lines).

4. Conclusions

The success of the SLS commissioning shows the power of the approach used for machine simulation and high level application writing: start with a simple but accurate design program; refine the design using the same algorithm used later in commissioning with a more powerful and complex tool; and incorporate this tool in the high level applications used in the control room. The use of CORBA as the glue between the different components in the high level application has probed to be a good idea, allowing the writers to choose a language appropriate for the user interface (Java, Tcl/tk) and keeping the power of the TRACY-2 C library.

For more information on the commissioning, check our report collection at our web server (http://slsbd.psi.ch/) or contact the author at Marc.Munoz@psi.ch. The web site provides also access to the software packages described in this paper.

References

[1]   M Böge., *First Operation of the Swiss Light Source*, presented at EPAC'02, Paris, June 2002

[2]   A.Streun, http://slsbd.psi.ch/~streun/opa/opa.html.

[3]   J. Bengtsson, *TRACY-2 User's Manual*, SLS Internal Document (1997)

[4]   M. Böge, *Update on TRACY-2 Documentation*, SLS Internal Note, SLS-TME-TA-1999-0002 (1999)

[5]   E. Forest and R.D. Ruth, *Fourth Order Symplectic Integration*, Physica D, vol. 43(1), pp. 105-117 (1990)

[6]   L. Dalesio et al, *The Experimental Physics and Industrial Control System Architecture: Past, Present and Future*, Proc. 4th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'93), Berlin, Germany, Oct. 18-23, 1993, Editors: W. Busse, M.C. Crowley-Milling, Nucl. Instr. Meth. A352 (1994) 179; EPICS, http://www.aps.anl.gov/epics/

[7]   S. Hunt *et al.*, *The Control and Data Acquisition System of the Swiss Light Source*, Proc. 7th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'99), 4-8 Oct. 1999, Trieste, Italy, p. 615

[8]   M. Böge, J. Chrin, M. Muñoz, A. Streun, *Commissioning of the SLS using CORBA Based Beam Dynamics App-lications,* Proc. 2001 Particle Acc. Conf. (PAC 2001), 18-22 June 2001, Chicago, USA, p. 292

[9]   M. Böge, A. Streun, V. Schlott, *Measurement and Correction of Imperfections in the SLS Storage Ring*, presented at EPAC'02, Paris, June 2002

Tracking particles in axisymmetric MICHELLE models

Eric M. Nelson†§ **and John J. Petillo‡**

† Los Alamos National Laboratory, MS B259, Los Alamos, NM 87545
‡ SAIC, Suite 130, 20 Burlington Mall Road, Burlington, MA 01803

Abstract. A novel particle tracker for 3d unstructured grids has been developed and employed in the finite element gun code MICHELLE. The particle tracker has now been extended to handle 2d planar and axisymmetric models. The unstructured 2d grids may contain triangle and/or quadrilateral elements and the element shapes may be linear, quadratic or cubic. In this paper we describe the particle tracking algorithm for axisymmetric models.

The 2d algorithm is similar to the 3d algorithm. A particle's position is computed in the enclosing element's local coordinate system, and the particle's momentum is computed in the global coordinate basis. For axisymmetric models we also integrate the angular velocity to obtain the azimuthal angle. We continue to use the cartesian basis for the momentum; we do not compute the momentum in the cylindrical coordinate basis. This provides for acceptable behavior near the axis without any exceptional treatment, but there is the cost of an additional two or three coordinate transformations in the equations of motion.

We also demonstrate the importance of scaling the azimuthal step size with the relative spatial step size when one wants to efficiently improve the accuracy of the particle tracker.

1. Introduction

MICHELLE [1] is a 3d gun and collector simulation code. A gun code attempts to self-consistently compute the emission and trajectories of charged particles in an electrostatic field. The charged particles contribute to the electrostatic field, and the electrostatic field contributes to the forces acting on the charged particles. One cycle of a gun code consists of a Poisson solve followed by particle tracking. Charge deposition during particle tracking provides a source term for the next cycle's Poisson solve. Ideally, a self-consistent solution is obtained after a modest number of cycles.

Users requested a 2d version of MICHELLE that includes MICHELLE's numerous sophisticated models for particle emission. These models include space-charge-limited and temperature-limited emission algorithms for electron gun cathodes, and secondary emission algorithms with material-specific databases for collectors. The users also requested compatible application-specific preprocessing and postprocessing capabilities. The development of the 2d version of MICHELLE was thus motivated.

The 2d finite element field solver for Poisson's equation is typical in many respects. An unstructured grid of triangular and quadrilateral elements is employed. Unstructured grids are atypical in gun codes. For example, the popular 2d finite element gun codes DEMEOS [2] and TRAK [3] rely on structured triangular grids. But unstructured grids do appear frequently in other applications. The elements' shape and basis functions may

§ enelson@lanl.gov

be linear, quadratic or cubic. The order of the shape functions for individual elements is independent of the uniform order of the basis functions, which is another relatively distinct feature of our finite element field solver.

The particles' equations of motion are simple and well-behaved in 3d and 2d planar geometries. In 2d axisymmetric geometries the equations of motion are not so simple and they are not necessarily well-behaved, particularly near the axis. In this paper we describe MICHELLE's 2d axisymmetric particle tracker, its evolution, and our initial experience with the tracker.

2. The axisymmetric particle tracker

MICHELLE employs a novel algorithm for tracking particles on unstructured grids. The novel algorithm tracks particles element by element through the unstructured grid, stopping and then restarting at each element boundary crossing. Within each element the particles are tracked in the element's local coordinate system using a Runge-Kutta integrator. After each step a polynomial representation of the trajectory is constructed and checked for intersections with the element boundary. The novel algorithm intends to track particles robustly, rapidly and accurately despite the presence of (1) field discontinuities at element boundaries and (2) quadratic and higher order element shapes. The novel algorithm is described in more detail in [1]. Our experience so far with 3d models suggests the novel algorithm is meeting these expectations and performing very well.

The novel algorithm tracks particles in the element's local coordinate system. For 2d problems the local coordinates are (u, v). For axisymmetric models the corresponding global coordinates are (z, r). The local coordinates are easily mapped to global coordinates. The tracker also integrates the particle's angular velocity $d\phi/dt$ to obtain the azimuthal position ϕ.

Note that the transformation from global coordinates (z, r) to local coordinates (u, v) is not simple for the higher order element shapes and for general linear quadrilateral elements. It is only simple for linear triangular elements and special (e.g., orthogonal) quadrilateral elements. At this time we prefer to avoid such transformations. This decision excludes, for example, the otherwise attractive Boris algorithm in cylindrical coordinates [4][5]. However, the Boris algorithm does behave very nicely near the axis, so perhaps we will reconsider our prohibition on global to local coordinate transformations and adopt some of the Boris algorithm's features in a future version of the code.

The novel particle tracker computes the momentum in the global coordinate basis. For 2d axisymmetric models there are two viable choices: a cylindrical basis with momentum components $(\gamma\beta_z,\ \gamma\beta_r,\ \gamma\beta_\phi)$; or a cartesian basis with momentum components $(\gamma\beta_x,\ \gamma\beta_y,\ \gamma\beta_z)$. Both choices are discussed in the following subsections, but we ultimately chose the cartesian basis because it behaves better when particles are near the axis.

2.1. Computing momentum in the cylindrical basis

The initial implementation of the 2d axisymmetric particle tracker computed the momentum in the cylindrical basis. This was motivated by a desire for speed and a desire to mitigate possible problems with gridlock. Regarding speed, using the cylindrical basis instead of the cartesian basis avoids two coordinate transformations. For either basis the electric field and the particle velocity are computed in the cylindrical basis. For

the cartesian basis, the electric field must be transformed from the cylindrical to the cartesian basis, and the particle velocity must be transformed from the cartesian to the cylindrical basis. The magnetic field, if present, contributes a third transformation when computing the momentum in the cartesian basis. These are not difficult transformations, but the individual steps of the particle tracker will be faster without them.

Gridlock [6] occurs when a particle is on the boundary between two elements with opposing transverse (relative to the shared boundary) forces and negligible transverse momentum. The consequent numerous element boundary crossings degrade the performance of the novel particle tracker. The particle can get stuck on the boundary. A simple element swell algorithm has effectively mitigated any gridlock problems that have arisen so far. More sophisticated algorithms for mitigating gridlock (which have yet to be implemented) decompose the momentum into parallel and transverse components. Computing the momentum in the cylindrical basis facilitates this decomposition.

The relativistic equations of motion for the momentum in the cylindrical basis are

$$\frac{d}{dt}\left(\gamma\beta_z\right) = \frac{q}{mc}\left(E_z + v_r B_\phi - v_\phi B_r\right) \tag{1}$$

$$\frac{d}{dt}\left(\gamma\beta_r\right) = \frac{q}{mc}\left(E_r + v_\phi B_z - v_z B_\phi\right) + \gamma\beta_\phi\frac{d\phi}{dt} \tag{2}$$

$$\frac{d}{dt}\left(\gamma\beta_\phi\right) = \frac{q}{mc}\left(v_z B_r - v_r B_z\right) - \gamma\beta_r\frac{d\phi}{dt} \tag{3}$$

where the angular velocity is $d\phi/dt = v_\phi/r$, and the MKSA system of units is employed.

The second term in equations (2) and (3) makes integration of the equations of motion difficult for some particles. The angular velocity can by very large and vary rapidly for particles near the axis. An error-estimating adaptive integrator would be helpful in such cases, but MICHELLE does not currently employ such integrators.

Two types of behavior arise when the integrator poorly samples the variation of the angular velocity. In mild cases there will be a large error in the particle's angle, but otherwise the momentum is not bad. The momentum is not necessarily bad because the particle strikes the axis (i.e., the global coordinate r passes through zero) without a significant change in momentum, at which time the transverse momentum is converted to a purely outward radial momentum. Any prior angular momentum has disappeared. This by itself might be okay for some axisymmetric models.

In worse cases, the integrator happens to sample the equations of motion when the particle is very close to the axis. The angular velocity will be very large, and the second terms in equations (2) and (3) lead to a large and obviously unphysical change in the momentum. The particle blasts off radially from the axis with a large transverse momentum. Note that global (as opposed to adaptive) refinement of the time step may fix an instance of this egregious behavior, but similiar behavior will likely arise for another particle in another location. The error does not appear to be bounded by the time step.

Such flagrant violations of physical law led us to abandon this approach in favor of computing the momentum in the cartesian basis. It is conceivable that the problems with computing the momentum in the cylindrical basis can be overcome with a modest amount of work. Aside from employing an error-estimating adaptive integrator, one can also switch from the cylindrical basis to the cartesian basis when the particle is in an element adjacent to the axis. Another fix, which would only address the egregious cases, is to artificially limit the angular velocity.

Perhaps the MICHELLE development effort will return to this matter in the future, but for now this initial implementation should be treated as yet another example

of premature optimization distracting a code development effort. Fortunately, the distraction was brief.

2.2. Computing momentum in the Cartesian basis

The relativistic equations of motion for the momentum in the cartesian basis are

$$\frac{d}{dt}(\gamma\beta_x) = \frac{q}{mc}(E_r\cos\phi + v_y B_z - v_z B_y) \tag{4}$$

$$\frac{d}{dt}(\gamma\beta_y) = \frac{q}{mc}(E_r\sin\phi + v_z B_x - v_x B_z) \tag{5}$$

$$\frac{d}{dt}(\gamma\beta_z) = \frac{q}{mc}(E_z + v_x B_y - v_y B_x) \tag{6}$$

where the transformation of the transverse electric field has been made explicit. The transverse field components of the magnetic field must also be transformed from the cylindrical basis to the cartesian basis,

$$B_x = B_r\cos\phi - B_\phi\sin\phi \tag{7}$$
$$B_y = B_r\sin\phi + B_\phi\cos\phi. \tag{8}$$

The worst case behavior when computing the momentum in the cylindrical basis does not occur in this case. Sampling an anomalously large angular velocity when a particle passes close to the axis does not lead to a corresponding anomalously large momentum error. But the behavior is still not necessarily ideal. There will be an anomalous but bounded momentum and position error due to the azimuthal angle ϕ being an effectively arbitrary number.

The position error, in a 3d sense, will be a fraction of the particle's radius. This can be made arbitrarily small by reducing the time step, but it is also bounded by the radial element size. The momentum error is due to the forces from the transverse fields being in an arbitrary direction. These forces are bounded, so this error can also be made arbitrarily small by reducing the time step. The fact that the transverse fields are typically small near the axis also helps.

When particles strike the axis, the momentum becomes directed radially outward, by definition, and any prior azimuthal component of momentum disappears. This aspect of the particle tracker's behavior is the same whether the momentum is computed in the cylindrical basis or the cartesian basis.

2.3. Time step selection

The integrator employed in MICHELLE is not an error-estimating adaptive integrator, but the time step is still chosen to accomodate the particle's local environment. For each step the time step is the minumum of (1) a specified maximum time step, (2) the time step computed from a specified spatial step size relative to the element size, (3) a specified fraction of the cyclotron period and (4) a specified fraction of the rotation period about the axis. The last criterion is equivalent to an azimuthal step size, and it is present only in the axisymmetric case. The first three criteria are present in all three cases: 3d, 2d planar and 2d axisymmetric. The specified fraction for criterion (4) is the same as the specified fraction for criterion (3).

We typically reduce the relative spatial step size to improve the accuracy of the particle tracker. But it is also important to reduce the azimuthal step size when a particle's momentum is mostly azimuthal. The difficulty in this case is that the relative spatial step size only weakly bounds the azimuthal step size. The initial $d\phi/ds$ is very
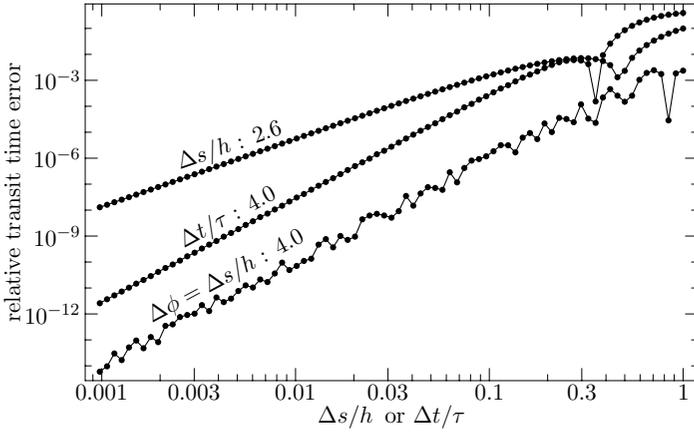
Figure 1. Convergence of the transit time τ with respect to three different step size parameters for a test case with a purely azimuthal initial momentum and using a 4th-order Runge-Kutta integrator with cubic interpolation at element boundary crossings. Each curve is labeled by the step size parameter and the convergence rate. The convergence rate is degraded when varying the relative spatial step size $\Delta s/h$ alone. The convergence rate is restored when the azimuthal step size $\Delta\phi$ scales with $\Delta s/h$.

large, perhaps infinite, so the azimuthal step $\Delta\phi$ is large even though the relative spatial step $\Delta s/h$ is small. The large azimuthal step degrades the accuracy of the momentum computation, particularly in the cylindrical basis.

Figure 1 shows some results from a simple non-relativistic parallel plate test case. A particle starts at the cathode plate with a purely azimuthal momentum and is accelerated axially in a constant electric field until the particle encounters the outer radial boundary. There are no field errors induced by the uniform 2×2 grid employed in this test case. The momentum was computed in the cylindrical basis.

The expected convergence rate of the transit time error for the 4th order Runge-Kutta integrator is not observed when varying only the relative spatial step size; the slope of the curve is only 2.6, not 4.0. Similarly degraded convergence is observed for the final position and momentum. The degraded convergence rate does not depend on the initial azimuthal momentum (assuming its not negligible). Selecting a fixed time step restores the convergence rate, as does making the azimuthal step size proportional to the relative spatial step size.

3. Conclusion

We have described the implementation and evolution of the particle tracker employed in axisymmetric MICHELLE models. The novel particle tracking algorithm for unstructured grids has been modified for axisymmetric models. The momentum is still computed in the cartesian basis in order to improve the behavior of particles near the axis. Some tests revealed the importance of setting the angular step size proportional to the relative spatial step size.

At this time the particle tracker appears to be acceptable for gun and collector simulations. However, we are still in the early stages of exercising the MICHELLE

finite element gun code, particularly the 2d version. We expect the code to continue to evolve as our experience grows.

Acknowledgements

References

[1] J. Petillo, et al. 2002 "The MICHELLE 3D Electron Gun and Collector Modeling Tool: Theory and Design," IEEE Trans. Plasma Science, vol. 30, no. 3, June 2002.
[2] R. True, "A General Purpose Relativistic Beam Dynamics Code," 1993 Computational Accelerator Physics Conf. (CAP'93), AIP Conf. Proc. 297, pp. 493–499.
[3] S. Humphries Jr., "Finite-element Methods for Electron Gun Design," 1996 Int. Conf. on Plasma Science (ICOPS'96). See also `http://www.fieldp.com/`.
[4] J. P. Boris, "Relativistic plasma simulation—optimization of a hybrid code," Proc. Fourth Conf. Numerical Simulation of Plasmas, 2–3 November 1970, Washington, D.C., pp. 3–67.
[5] C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation*, Institute of Physics Publishing, 1991.
[6] Eric M. Nelson, "Analysis of Simple Gridlock," Los Alamos National Laboratory report LA-UR-00-4266, 9 August 2000.

Beam dynamics problems of the muon collaboration: ν-factories and μ+-μ- colliders

David Neuffer

Fermilab, Batavia IL 60510

Abstract. We present some of the outstanding beam dynamics problems that are being explored by the muon collaboration, which is studying potential uses of cooled muons in neutrino factories and muon colliders. These problems include the search for more efficient muon capture and cooling methods, improved acceleration options, and more affordable neutrino factory scenarios. Recent progress in ring coolers enables longitudinal cooling and may extend cooling performance to the level needed for μ+-μ- colliders.

1. Introduction

The relatively long lifetimes of muons, coupled with their properties as high-mass leptons, make them a candidate for collection, acceleration and storage scenarios in high-energy physics facilities. The Muon collaboration is studying these possibilities, initially for a future μ+-μ- collider [1,2], and more recently for a ν-factory. The collider concept is displayed in fig. 1, and the ν-factory concept [3,4] is displayed in fig. 2.

The ν-Factory feasibility studies [3,4] have established that a high-intensity ν-Factory could be built within present technological capabilities, but that its expected cost would be a bit larger than the currently limited resources of US high-energy physics. The key high-cost systems are the μ-capture and rf rotation systems, the beam cooling, and the acceleration components. The present R&D efforts are focussed on developing more affordable alternatives and improvements to these systems; improved performance with reduced cost would make the facility a leading candidate for the next world high-energy physics facility.

This R&D requires extensive use of computational accelerator physics (CAP). These efforts are focussed on improving the $\pi \rightarrow \mu$ collection scenarios, and the cooling concepts, as well as developing acceleration scenarios. Promising approaches in each of these areas are developing; for example, ring cooler approaches have demonstrated "emittance exchange" required for simultaneous longitudinal and transverse cooling. These are discussed below. Initial experiments on possible target systems and on initial cooling systems are in progress and also require CAP support.

2. Muon production and capture problems

An R&D priority is the development of an optimal μ production and capture system. Fig. 3 shows the current muon collaboration baseline design. It requires a high intensity proton source (up to 10^{14} ~20 GeV protons per pulse, operating at ~15Hz). Intense

proton bunches from this source are sent onto a liquid-metal target, producing a large number of π's which are captured within a strong-focussing (~20T) solenoidal field. The captured π's decay in the following transport. This design has been developed through extensive use of CAP studies, most notably using the particle interaction and production code MARS of N. Mokhov et al.,[5] but also using magnet design and target material response codes. Variations on this scenario continue to be studied, searching for lower cost and/or higher performance. Variations in the driver p-beam, different targets (solid Cu or C or …), varying capture optics (Li lens, magnetic horns, quads, etc.), and differing following μ-transport systems can be considered and studied, using CAP.

2.1 μ-bunching and phase-energy rotation options

Following initial capture, the muon beams must be matched into the following cooling and/or acceleration systems. The initial step is to reduce the energy spread. In the collider studies [1, 2] this is done by phase-energy rotation using low-frequency rf (~30 MHz), which is matched into a low-frequency initial cooling system. For the ν-factory scenarios, [3,4] an induction linac is used to decelerate the high-energy "head" and accelerate the low-energy "tail" of a muon bunch, obtaining a long bunch (~30-100m) with small energy spread (~10MeV). This is trapped into a train of 200 MHz bunches, which is then injected into a 200 MHz cooling system. (~200 MHz rf systems may be an optimum in cost and acceptance for cooling.) Both of these methods require development and construction of large and expensive novel acceleration systems, with gradients and total voltages substantially larger than currently available.

More recently, a variant capture and phase-energy rotation system using only ~200MHz rf has been proposed.[6] In this variant, the muons first drift, lengthening into a long bunch with a high-energy "head" and a low-energy "tail". Then, the beam is transported thtough an "adiabatic buncher", a section of rf cavities that gradually increase in gradient and decrease in frequency (from ~300 to ~200MHz). The rf wavelength is fixed by requiring that reference particles at fixed energies remain separated by an integer number of wavelengths. This forms the beam into a string of bunches of differing energies (see fig. 4). Following the buncher, the beam is transported through a high-gradient fixed-frequency (or slightly varying) rf section that aligns the bunches to (nearly) equal central energies, suitable for injection into a fixed-frequency ~200 MHz cooling system.

This high-frequency bunching and phase-energy rotation uses present technology and should be much more affordable than low frequency options. Much more simulation and optimization study is needed to determine whether it traps sufficient useable muons for cooling and acceleration.[7] Complete, realistic simulations for a neutrino factory have not yet been performed.

3. Muon cooling challenges

The muon bunches must then be compressed in size to fit within the neutrino factory acceleration and storage systems. This requires beam cooling, and the cooling and acceleration must occur within a fraction of a muon lifetime. (More cooling would be required for a high-luminosity μ^+-μ^- Collider.) The cooling method that can be fast enough is "ionization cooling."[8,9] In ionization cooling the muons pass through a material medium, losing both transverse and longitudinal momentum, followed by reacceleration in rf cavities, regaining longitudinal momentum. The net effect is loss of transverse momentum, obtaining transverse cooling. (Energy loss in wedges at nonzero dispersion can add longitudinal cooling.) Configurations for cooling have been developed by the muon collaboration; they require high-gradient rf as well as strong

focusing. A particular configuration that has been developed is a "FOFO" cell which relies on high-field solenoids for focusing, and places the cell tunes between resonances to obtain focusing over a broad energy range.[10]

3.1 Simulation tools for muon cooling

Detailed simulation is required for the verification and optimization of possible cooling systems, and these systems present many novel features for simulation codes. The optimum μ cooling energy is ~200 MeV ($\gamma \cong 2$—3) so the dynamics is neither nonrelativistic nor fully relativistic. Also the μ's have large transverse momenta, and nonparaxial motion must be considered. Ionization cooling requires energy loss in materials; simulation codes must include the full complexities of particle-material interactions (multiple scattering, energy-loss straggling, correlations, etc.). Strong focusing to small beam sizes and high-gradient reacceleration are required; simulations must include all nonlinear field effects due to realistic rf cavities, solenoidal fields, quad, dipole and fringe fields, etc. All kinematic effects must be included, including x-y coupling and angular momentum effects of solenoidal fields, and chromatic effects from large-momentum spreads.

New simulation codes have been developed, and other codes have been adapted to simulate ionization cooling. These include:

    a. ICOOL, a new simulation code developed by R. Fernow and collaborators.[11]
    b. Geant4, based on particle in matter and fields simulations for particle physics detectors.[12]
    c. COSY, a general purpose beam dynamics code that can use its flexible framework to encompass cooling problems.[13]

Other single user simulations have also been used, as well as analytical models.[14]

A critical need is for simulations that can vary parameters within the cooling systems to develop optima; COSY, in particular, has significant capabilities for this application.

3.2 Ring coolers

An important problem for cooling is the development of systems that can obtain longitudinal as well as transverse cooling. These systems require dispersion with wedge absorbers to cause higher energy particles to lose more energy than lower energy particles, and effectively combining this with the transverse cooling has been a challenge.

Recently, a number of approaches using a "ring cooler" type of design[15-20] have been developed and studied in CAP simulations. Fig. 5 shows a particular ring cooler[17] and simulation results from another case[19]. The ring coolers are designed for multiturn cooling and can cool both transversely and longitudinally by up to a factor of ten in each of the 3 dimensions. Detailed simulations, including accurate modelling of dipoles, fringe fields and wedge absorbers are needed to confirm these designs.

These designs are a large step toward the goal of achieving the cooling needed for a collider.[20] The multiturn cooling systems could also be much more affordable than the single pass systems used in the ν-factory feasibility studies.

4. Acceleration scenarios and variations

4.1 Recirculating linac acceleration

The baseline schemes for the ν-factory and μ^+-μ^- colliders use recirculating linacs (RLAs) as the primary engines for muon acceleration. In a RLA, the beam is transported through a linac for multiple turns of acceleration, with a separate return transport for each turn. The simulation program OPTIM was used to develop the most recent ν-Factory scenario.[21] In the present ν-Factory scenarios, only ~4 turns of recirculating

acceleration are included and each of the recirculating transports requires relatively large apertures to accept the very large μ-beams. While less expensive than a single linac, the acceleration is somewhat more expensive than desired; cost reduction is desired

4.2 FFAG acceleration scenarios

Mori et al. have suggested using fixed-field alternating gradient (FFAG) accelerators for a ν-factory.[22] In an FFAG the beam is accelerated over the full energy gain within a single fixed magnetic field transport, avoiding the multiple return arcs an the RLA. A critical problem is to maintain synchronous acceleration over many turns; low-frequency or multiharmonic rf may be needed. Typical systems use ~10 turns of acceleration.

4.3 Very rapid cycling synchrotron scenarios

More recently, an accelerator scenario using a very-rapid-cycling synchrotron (VRCS) to accelerate muons from 4 to 20 GeV has been proposed.[23] In a VRCS the magnetic field is ramped as the μ-energy increases. The critical constraint here is that acceleration must be completed before μ decay. For the reference case, acceleration requires 35μs or 12 turns of a 900m circumference VRCS ring. Beam dynamic studies are needed to determine if particle stability is maintained (longitudinal and transverse) through the acceleration. Critical problems exist in developing the fast ramping magnets. The VRCS concept should be somewhat easier in higher-energy acceleration; the acceleration time can be increased as the muon lifetime increases.

All of these acceleration alternatives require substantial computational physics for verification, optimization and development. Improved alternatives may result from these studies.

5. Experimental studies and support

The Muon collaboration has initiated experimental studies.[24] At BNL and CERN studies of liquid metal (Hg) jet targets have begun, and at BNL liquid jet target interactions with a high-intensity proton beam have been studied. At Fermilab, rf studies of high-gradient cavities for μ-cooling have proceeded. Design, construction and tests of liquid hydrogen cooling absorbers are in progress. Internationally, a collaboration for an international cooling experiment (MICE) based at the Rutherford Appleton Laboratories-ISIS μ beam line has been formed, with plans to transport and track beam through typical cooling segments (see Fig. 6) and to compare measured with predicted cooling performance.[25] These initiatives need substantial CAP support and development, both in component design and complete system optimization.

6. Discussion and future directions

We have presented some of the critical R&D challenges of the muon collaboration, particularly in muon targetry, capture, cooling, and acceleration. Many other challenges have not been described here; for example, the design of the μ-storage ring for the ν-factory and the collider ring for a μ⁺-μ⁻ collider. Also, the best future applications for stored muons will depend on the currently developing status of particle physics. Many opportunities for computational accelerator physics applications exist within the present R&D, and computational physics initiatives can lead to dramatic changes and improvements in this program. Some of that research is presented in this conference in papers from M. Berz, D. Elvira, D, Errede, Y. Fukui, C. Johnstone, C. Maidana, K. Makino, P. Stoltz, and others [26].
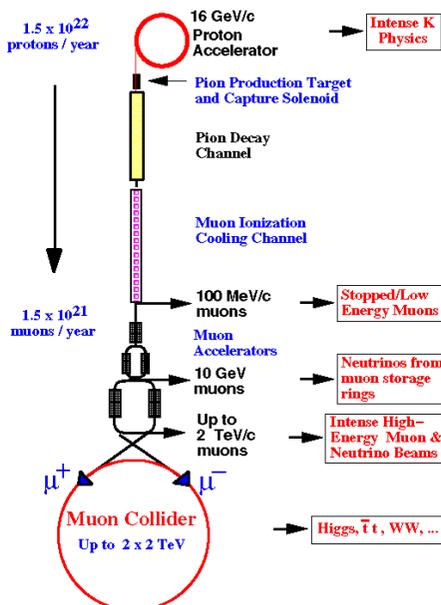
Figure 1. A schematic view of a μ^+-μ^- collider facility, showing a high-intensity proton source which would produce pions. π decay produces μ's which are cooled to collider intensities, and accelerated. Counterrotating μ^+ and μ^- bunches are inserted into a storage ring for high-luminosity collisions.
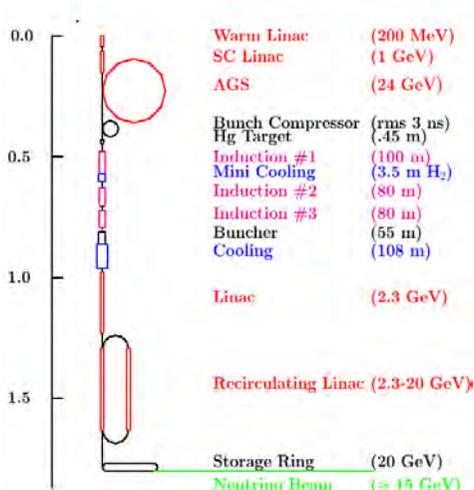
Figure 2. A schematic view of a μ-storage ring ν-factory. A proton driver produces μ's which are cooled and accelerated to 20 GeV and inserted in a storage ring. μ-decay ($\mu \rightarrow e + \nu_\mu + \nu_e^*$) in the storage ring straight section provides collimated electron and muon neutrino beams suitable for long baseline oscillation experiments.
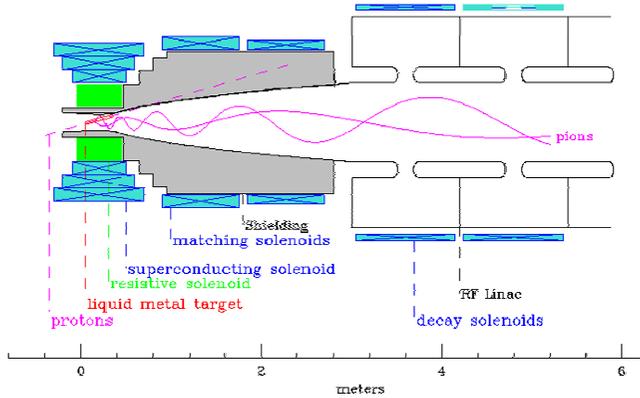
Figure 3. Possible target and initial capture system. A high intensity proton beam hits a liquid metal target inside a high-field solenoid, capturing π's which are focussed into a lower field solenoidal transport system for π→μ +ν decay; the μ's are captured and cooled.
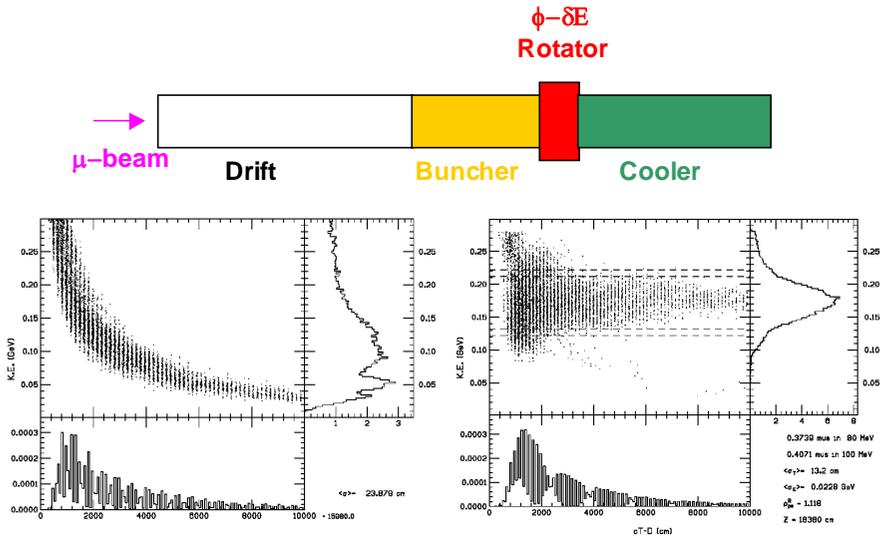
Figure 4. "High-frequency" buncher and phase rotation consisting of a ~100m drift followed by a ~ 60m "buncher" in which the beam is transported through a 300-200 MHz rf system which forms the beam into a string of bunches, which are then aligned in energy by a ~200 MHz high-gradient rf system (φ-δE rotator). The lower left figure shows simulation results of beam at the end of the buncher and the lower right shows beam at the end of the φ-δE rotator.
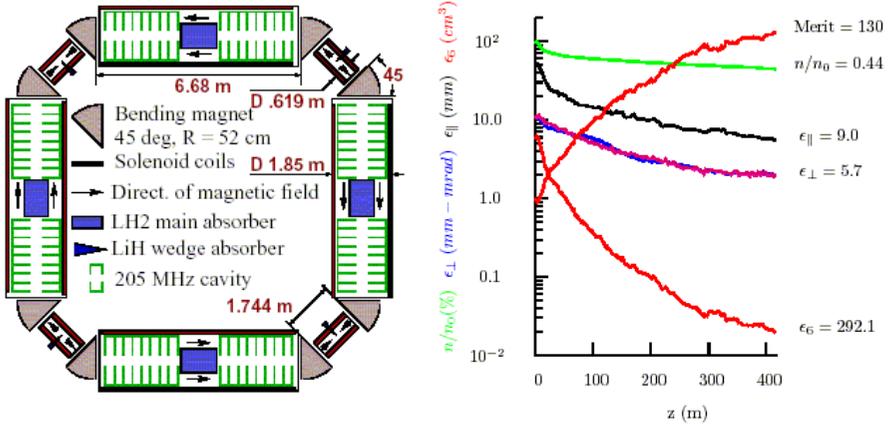
Figure 5. On the left a 4-sided "ring cooler" is displayed.[from ref. 17]. On the right simulation results from a (different) 12-sided ring cooler [from ref. 19], are shown, with the simulations showing reduction of transverse emittances by a factor of 5.7 and of longitudinal emittance by a factor of 9 after ~10 turns.
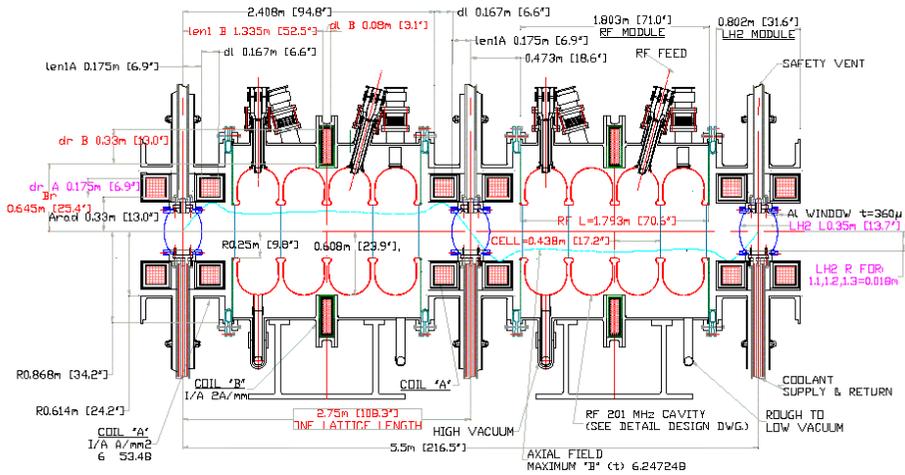
Figure 6. Overview of a cooling cell (engineering design), such as may be used in the MICE experiment as a prototype for a ν-Factory cooling system. The cell includes rf cavities, magnetic focusing coils, and hydrogen absorbers.

References

[1] "$\mu^+\mu^-$ Collider - A Feasibility Study", BNL-52503, Fermi-Lab-Conf.-96-092, LBNL-38946 (1996), Proc. Snowmass 96 workshop (1997)., see also http://www.fnal.gov/projects/muon_collider/, and links from there.

[2] C. M. Ankenbrandt et al., Phys. Rev. Special Topics Accel. Beams 2, 081001 (1999).

[3] N. Holtkamp and D. Finley, ed. "A Feasibility Study of a Neutrino Source Based on a Muon Storage Ring", Fermilab, April 15th, 2000.

[4] J. Gallardo, S. Ozaki, R. Palmer, and M. Zisman, ed., "BNL ν-Factory Feasibility Study", BNL 52623, June, 2001.

[5] N. V. Mokhov, "The MARS Code System User's Guide", Fermilab-FN-628 (1995); N. V. Mokhov and O. E. Krivosheev, "MARS Code Status", Fermilab-Conf-00/181 (2000); http://www-ap.fnal.gov/MARS/

[6] D. Neuffer, A, Van Ginneken, "High-Frequency Bunching and ϕ-δE Rotation for a Muon Source," Proc. 2001 Part. Acc. Conf., Chicago, p. 2029 (2002).

[7] D. Elvira and N. Keuss, MUCOOL-253, MUCOOL-254, August 2002.

[8] A. N. Skrinsky and V.V. Parkhomchuk, Sov. J. Part. Nucl., 12, 223 (1982).

[9] D. Neuffer, Particle Accelerators 14, 75 (1983).

[10] R. Fernow, "Classification of Periodic Alternating Solenoid Lattices," MuCOOL-244, April 2002.

[11] R. Fernow, "A Simulation Code for Ionization Cooling of Muons", Proc. 1999 Part. Acc. Conf., New York, p. 3020, http://pubweb.bnl.gov/people/fernow/icool/.

[12] V. Daniel Elvira, P. Lebrun and P. Spentzouris, "Beam Simulation Tools for GEANT4(BT-v1.0)", MuCOOL-260, July 2002, http://www-cpd.fnal.gov/geant4/

[13] M. Berz and K. Makino, "COSY INFINITY, version 8.1", MSUHEP-20704, May 2001.

[14] C. X. Wang and K.-J. Kim "Linear theory of ionization cooling in 6D phase space", MUCOOL-240, April 2002.

[15] V. Balbekov and A. Van Ginneken, "Ring Cooler for Muon Collider", in Proc. Physics Potential and Development of μ^+-μ^- Colliders AIP: Conf. Proc. 441, p. 310(1998).

[16] V. Balbekov et al. "Muon Ring Cooler for the Mucool Experiment, Proc. 2001 Part. Acc. Conf., Chicago, p. 3867 (2002).

[17] V. Balbekov, "Ring Cooler Progress", MuCOOL-246, May 2002.

[18] J. Berg, R. Fernow, R. Palmer, "An Alternating Solenoid Focused Ionization Cooling Ring", MuCOOL-239, March 2002.

[19] R. B. Palmer, R. Fernow, S. Berg, "Simulations of RFOFO Ring Emittance Exchange", unpublished (2002).

[20] Y. Fukui, D. B. Cline, A. Garren, and H. Kirk, "Muon cooling Rings for the ν-Factory and μ^+-μ^- Collider", MuCOOL-262, November 2002.

[21] A. Bogascz and V. Lebedev, "Recirculating Linacs for a Neutrino factory- Arc Optics Design and Optimization", Nucl. Inst. and Meth. A 472, 528 (2001).

[22] Y. Mori, Proc. European Particle Accelerator Conference, Paris, p.278 (2002).

[23] S. Berg, A. Garren, R. Palmer, G. Rees, D. Summers and Y. Zhao, "A Pulsed Muon Synchrotron for a Neutrino Factory", MUCOOL Note 259, October, 2002.

[24] S. Geer, "Neutrino factory designs and R&D", MuCOOL-249, July 2002.

[25] A. Blondel et al., (MICE collaboration), "An International Muon Ionization Cooling Experment (MICE)," http://hep04.phys.iit.edu/cooldemo/micepsiloi.pdf.

[26] M. Berz and K. Makino, eds., Proc. ICAP2002.

Simulation of electron cloud multipacting in solenoidal magnetic field[*]

A. Novokhatski[**] **and J. Seeman**

Stanford Linear Accelerator Center, Stanford University, CA 94309

Abstract. A simulation algorithm is based on a numerical solution of the Vlasov equation for the distribution function of electron cloud density in a cylinderical vacuum chamber with solenoidal magnetic field. Algorithm takes into consideration space charge effects. This approach considerably improves the simulation of multipacting effects thanks to a better approximation of the secondary emission mechanism. Simulation studies were carried for the SLAC B-factory vacuum chamber for different bunch patterns and solenoidal field. Space charge and the magnetic field limit the maximum density of the electron cloud. Magnetic resonant damping of multipacting was found in special cases of positron beam parameters and magnetic field amplitude.

1. Introduction

The electron cloud at PEP-II in the low-energy positron ring is built up from multipacting electrons in the straight section vacuum chamber and secondary emission of electrons from the vacuum antechamber in the arcs. Placing solenoidal magnetic fields around the ring successfully reduced multipacting and damped the electron cloud instability [1]. PEP-II has an upgrade plan that is leading toward higher luminosity by doubling the number of bunches and decreasing the spacing between bunches by a factor of 2 [2]. Here we describe the attempt to understand the possible effect from a new bunch pattern on electron cloud multipacting using the results of a computer simulation.

2. The model of electron cloud

The physical picture of the multipacting process leads us to use the phase distribution function for the best description of the electron cloud and for a precise modelling of secondary electron emission. It is worth noting that the usual approach of particle tracking can not accurately describe the secondary electron yield, as a lot of particles just needed for satifyring the boundary condition. The energy distribution of electrons which are emitted from a surface bombarded with primary electrons has a narrow peak of order 5 eV for "true'" secondary electrons, on the other hand, more over to have secondary emission yield more than one electron, the primary electrons must have tens or even hundreds of electron volts. This means the emitted electrons have to be accelerated by the

---

field of positron bunches up to these energies in order to build up the multipacting process. In this way the initial angular distribution of emitted electrons does not play any considerable role, as the force vector from a positron bunch determines the vector of electron momentum. Without multipacting the number of primary electrons, coming from photoemission is not enough to create a considerable back action on the positrons.

In the straight sections of PEP-II the vacuum chamber is round and made from stainless steel. When a positron bunch is moving along the axis of a round tube it's electric field is radial and it also gives a radial kick to the cloud electrons. If the surface of the tube wall is azimuthally homogeneous (secondary emission yield is the same everywhere), then we can imagine that the electron cloud will also be azimuthally homogeneous. This means that we need only a two-dimension phase distribution function of radius and radial velocity for a complete description of the electron cloud in a straight section.

3. Vlasov equation and electromagnetic forces

The phase distribution function $\Psi(r,V)$ describes the density of the electron cloud on the phase plane of radius and radial momentum (velocity), as shown in Fig. 1.
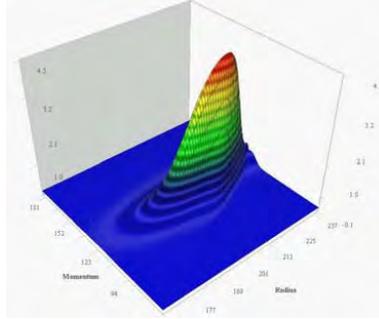
Figure 1. Phase distribution function on the phase plane of radius and radial momentum

This function obeys the Vlasov equation

$$\frac{\partial}{\partial t}\Psi + \frac{\partial}{\partial V}\Psi \times \frac{dV}{dt} + \frac{\partial}{\partial r}\Psi \times \frac{dr}{dt} = 0 \qquad \frac{dV}{dt} = \frac{F}{m_e} \qquad \frac{dr}{dt} = V$$

where F is the force acting on an electron from a positron bunch field, solenoidal magnetic field and electric field of other electrons (space charge force), m_e is the mass of an electron. Electric field of a positron bunch of the Gaussian shape is

$$E_{bunch}(r,t) = \frac{c}{2\pi r} \times \frac{Z_0 I_{bunch}^+}{f_{rev}} \times \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(ct)^2}{2\sigma^2})$$

where I_{bunch}^+ is the positron current per one bunch, σ is the positron bunch length, f_{rev} is the revolution frequency of particles in the positron ring and $Z_0 = 120\pi$ Ohm. The radial force from a solenoid is

$$F_{sol}/m_e = r*\Omega^2 *\left(\left(\frac{a}{r}\right)^4 *\left(1-\frac{\varphi_0}{\Omega}\right)^2 - 1\right)$$

where a is the radius of the vacuum tube and $\Omega = \frac{e}{2mc}H$ is the Larmor frequency, $a\varphi_0$ is initial azimuthal velocity at the wall. Azimuthal motion is invariant

$r^2(\varphi - \Omega) = const = a^2(\varphi_0 - \Omega)$. The space charge field has radial and longitudinal components. In the case of periodic series of positron bunches, when $E(t,z,r) = E(\tau,0,r)$ $\tau = t - z/c$ these components are:

$$E_r^{sc} = \frac{Z_0 c}{2} * e\bar{n} * \frac{a^2}{r} * \frac{\int_0^r (n + \frac{1}{Z_0 c^2}\frac{\partial}{\partial \tau}E_z)r'dr'}{\int_0^a nrdr}$$

$$E_z^{sc}(r,\tau) = Z_0 * \int_r^a j_r(r',\tau)dr'$$

where $\bar{n} = \frac{2}{a^2}\int_0^a nrdr$ is the space average electron cloud density and j_r is the radial electron cloud current.

Using these formulas it is very easy to estimate the acceleration (energy gain) of the cloud electrons from a kick of a positron bunch

$$W = \frac{m_e c^2}{2} * \left(\frac{Z_0 I_{bunch}^+}{m_e c f_{rev} 2\pi a}\right)^2$$

and the average electron cloud density

$$\bar{n}e = 2\frac{I_{bunch}^+}{\pi a^2 f_{rev}\lambda_{RF} N}$$

The spacing between positron bunches is equal to RF wavelength λ_{RF} multiplied by the spacing number N.

For typical PEP-II parameters:

$$I_{bunch}^+ = 2mA,\ f_{rev} = 136kHz,\ a = 47.5mm,\ \lambda_{RF} = 63cm,\ N = 2$$

we have the following estimations:

$$W = 30.2\,eV \qquad \bar{n} = 2.06*10^{13}\ m^{-3}.$$

This means that an accelerated electron after returning back to the wall can produce more secondary particles (according to the secondary emission yield curve, presented in Fig.2). Saturated cloud density is of the order of the of residual gas density in the vacuum chamber ($p \sim 10^{-9}$ Torr).

4. Computer algorithm

A double-step semi-implicit finite-difference scheme with a diffusion parameter α is to model a numerical solution of the Vlasov equation

$$22\Psi_{i,k}^{n+1} + \Psi_{i,k+1}^{n+1} + \Psi_{i,k-1}^{n+1} = 22\Psi_{i,k}^n + \Psi_{i,k+1}^n + \Psi_{i,k-1}^n - 24\frac{\Delta t}{\Delta r}V_j\left(\Psi_{i,k+1/2}^{n+1/2} - \Psi_{i,k-1/2}^{n+1/2}\right)$$

$$a\Psi_{i,k}^{n+1} + \Psi_{i+1,k}^{n+1} + \Psi_{i-1,k}^{n+1} = b\Psi_{i,k}^n + c(\Psi_{i+1,k}^n + \Psi_{i-1,k}^n) - d\frac{\Delta t}{\Delta V}F_k\left(\Psi_{i+1/2,k}^{n+1/2} - \Psi_{i-1/2,k}^{n+1/2}\right)$$

$$a = \frac{22+2\alpha}{1-\alpha} \quad b = \frac{22-2\alpha}{1-\alpha} \quad c = \frac{1+\alpha}{1-\alpha} \quad d = \frac{24}{1-\alpha}$$

This scheme has a very good dispersion relation

$$\sin\frac{\omega\,\Delta t}{2} = \frac{1\,2}{1\,1} V \frac{\Delta t}{\Delta r} \frac{\sin\dfrac{k\,\Delta r}{2}}{1 + \dfrac{1}{1\,1}\cos(k\,\Delta r)}$$

Wave vector is linear with frequency $\omega \cong Vk$ in the region up to half the mesh-size frequency. Small value of the diffusion parameter $\alpha = 0.005$ being needed to compensate oscillations at the mesh-size frequency.

Boundary conditions are described by secondary electron emission probability functions $P(\varepsilon_{in}, \varepsilon_{out})$

$$\Psi_{in}(r=a) * V_{in} = \Psi_{out}(r=a) * V_{out} \times P(\varepsilon_{in}, \varepsilon_{out})$$

The probability function is a combination of secondary emission yield and a spectrum of secondary electrons. Experimental data is used for this function specification.

The code was written in Fortran 90 using graphical library "Array Viewer". A typical number of mesh points is 500*500. Typical computer time (1GHz PC) for a $1\mu\sec$ mulpacting process is 12 hours.

5. Secondary emission functions

Secondary emission yield for stainless steel, as a function of the energy of the primary electron is presented in Fig2
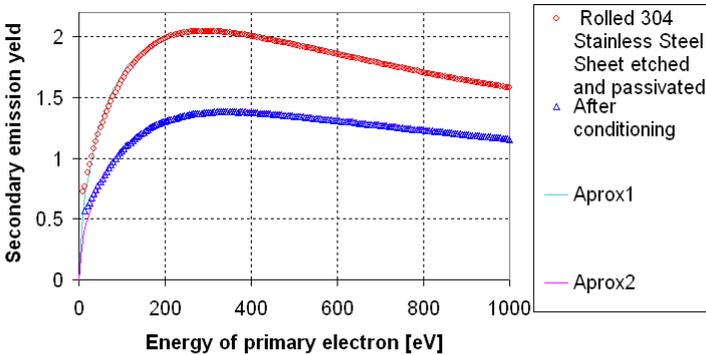
Figure 2. Secondary electron emission yield.

The measurements were done at SLAC by R.E.Kirby [3,4]. The smaller yield curve shows the reduction of the secondary emission yield with conditioning. We can suggest that the real yield is somewhere between this curves, however in simulations we take usually use the worst case. Approximation curves used in simulations are also shown. We extrapolate to zero yield for zero energy of the primary electrons.

The energy spectrum of the secondary electrons includes inelastically backscattered and elastically reflected electrons [5]. We use R.E.Kirby results of the spectrum measurement for two primary electron energies (Fig.3) to extrapolate spectrums to other energies.
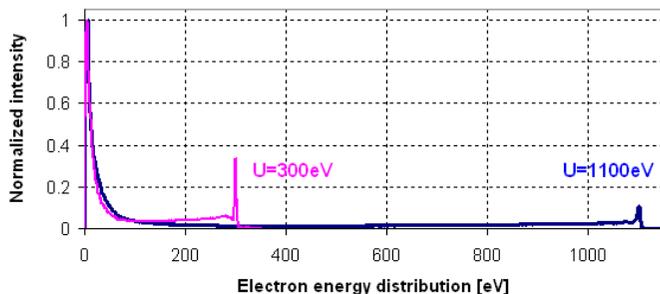
Figure 3. Normalized spectrum of secondary electrons for normal incident primary electrons of 300eV and 1100eV from clean stainless steel.

6. Multipacting at small solenoidal fields

First simulations were carried for small values of a solenoidal field to study the growth rates of cloud density due to multipacting. We start with some initial distribution of electrons then let positron bunches appear periodically in time and watch how the electron cloud density changes in time. Fig.4 shows the dynamics of cloud density with a positron train of a bunch spacing by 2 and bunch current of 2mA for different values of solenoidal field. Logarithm functions of density are shown on the right side together with linear approximations. At the beginning, the density increases exponentially, but saturates due to the action of space charge forces. Higher solenoidal fields not only decrease the growth rate, but also change the growth function from pure exponential to a square root dependance of time.
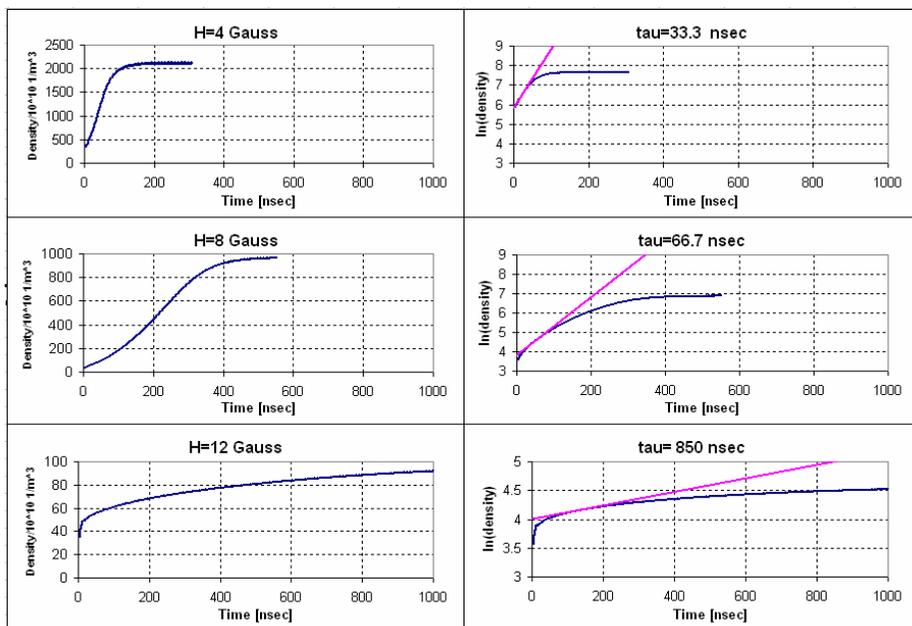
Figure 4. Dynamics of electron cloud density for different values of solenoidal field. Logarithm functions of density are shown on the right side.

Electron cloud distribution on the phase plane (image plot) is shown on Fig.5. Particles travel from chamber axis to the wall in the upper half part of the plot and from the wall to the axis in the lower half part. In the phase plot, radial momentum is measured in equivalent of electron volts. The radial distribution is shown in the right part of Fig.5. The picture shows the cloud at the moment between two positron bunches. Solenoidal magnetic field is 8G. It is possible to see that the cloud comprises five secondary bunches.
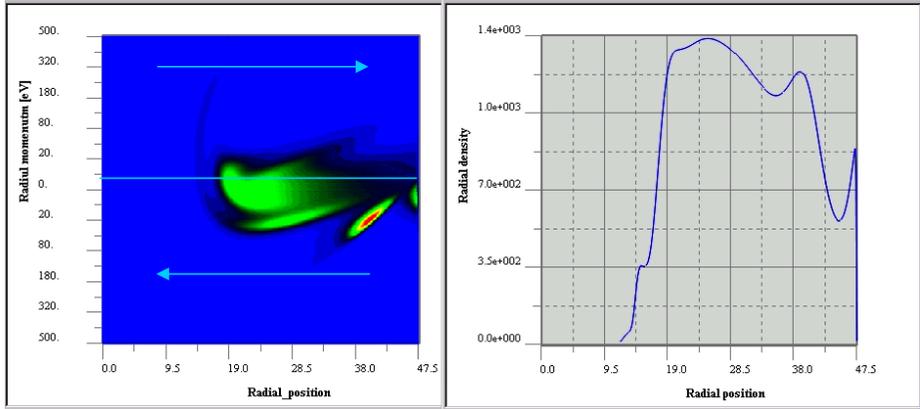
Figure 5. Electron cloud on the phase plane. Light blue line depicts zero velocity line. Arrows show the directions of particle motions. Radial cloud distribution is on the right.

7. Main resonance

While studying the behaviour of the electron cloud for different solenoidal fields we found a strong resonance. This resonance happens when the time interval between the positron bunches is equal to the flight time of the secondary particles back to wall. The flight time is mainly determined by the solenoidal field H and partially by the cloud size and intensity. Naturally the resonance depends strongly upon the secondary emission function. The resonance is the boundary between completely different behaviours of the electron cloud. Multipacting happens when the flight time is a little bit smaller than the positron time interval; when the solenoidal field is a little bit higher than the resonance field $H > H_{res}$. And there is no multipacting if $H < H_{res}$. Fig.6 demonstrates this effect. A difference of the solenoidal field of only 1 Gauss completely changes the behaviour of the cloud.
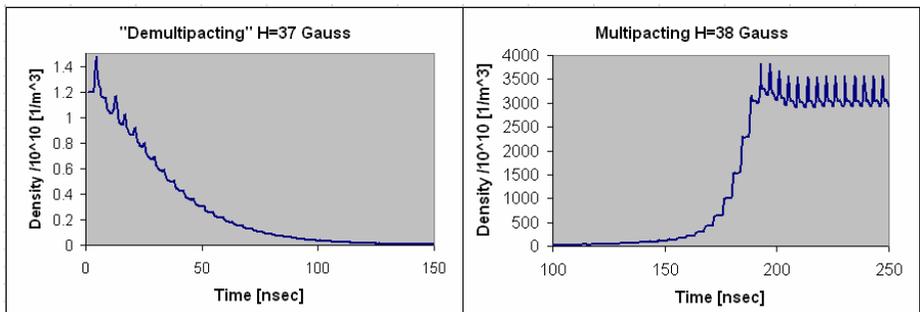
Figure 6. Cloud density behaviour in the resonance region.

Corresponding phase photos (high plots) of the clouds are shown at Fig.7. Clouds are "shot" just before positron bunch arrives.
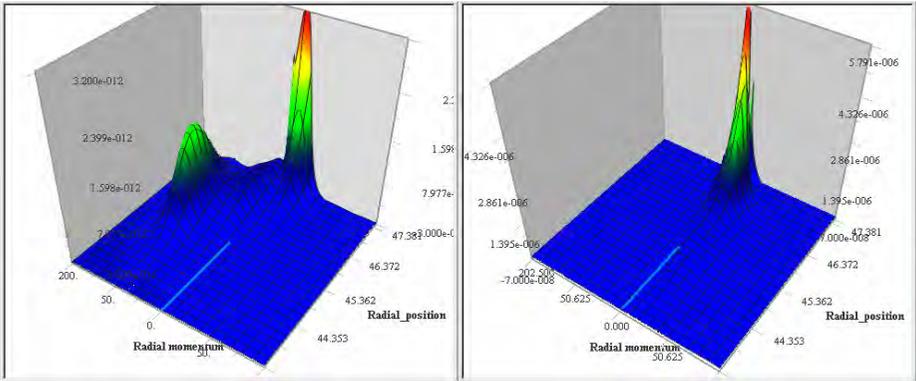
Figure 7. Electron clouds on the phase plane just before a positron bunch arrives. The left picture is for a solenoidal filed H= 37G and the right picture for H=38G. The light blue lines depict the zero velocity line.

Secondary particles, previously produced by high-energy particles are ready to be accelerated by the next positron bunch. After acceleration they will come back to the wall and produce more new particles. It is possible to see that "right" cloud mainly consists of secondary particles. However "left" cloud has additional high-energy peak, which will be decelerated by positron bunch and will arrive at the wall with very little energy and hence will not produce new secondary particles. Therefore the density will go down and finally "left" cloud will waste away.

8. Other resonances

We can assume that there can be some other resonances. A resonance can also happen if the flight time of the secondary particle is equal to an integer number of time intervals between positron bunches.
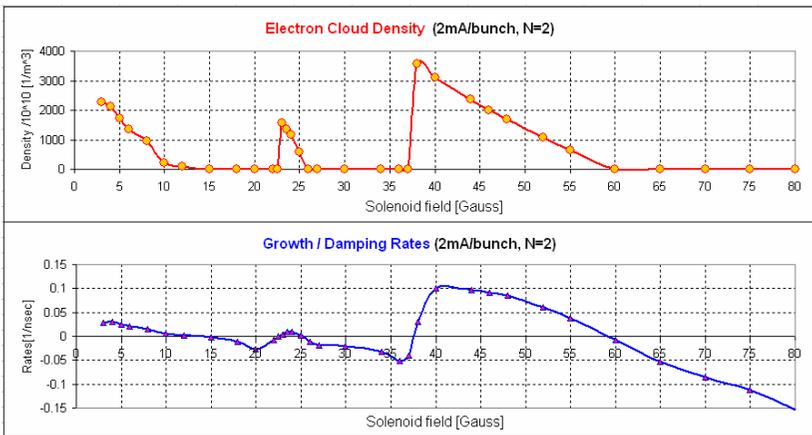
Figure 8. Electron cloud saturated density and growth/damping rates as a function of the solenoidal field for a positron train with a bunch spacing by 2.

These resonances happen at smaller values of solenoidal field. In our case we have a second resonance at the solenoidal field of 23G. Other resonances are in the region below 10 G. Fig.8 shows the saturated values of the electron cloud as a function of the solenoidal field and growth/damping rates. Negative values mean that the cloud wastes away after some time. There is no multipacting when solenoidal field is more than 60 G. Clear regions are also in the gap of 26-36 G and 14-22 G.
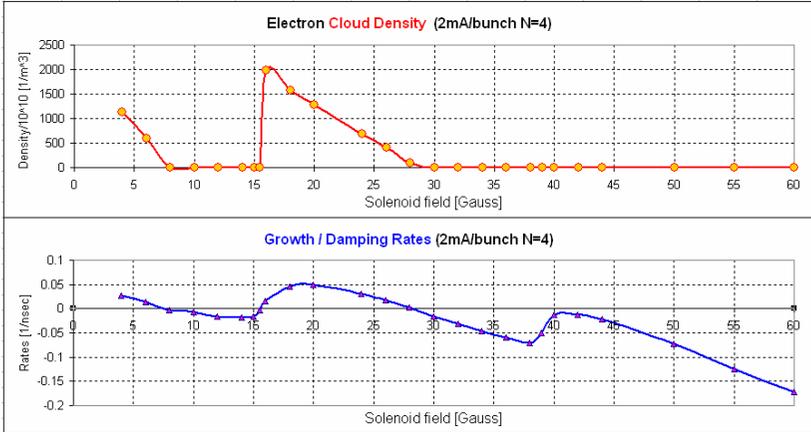
Figure 9. Electron cloud saturated density and growth/damping rates for a positron train with bunch spacing by 4.

For comparison with Fig.9 we present analog curves for the electron clouds with a positron train with spacing by 4. The main resonance is moved to 16 G, other resonances are in the region below 8 G. No multipacting after 30 G and in the gap 8-15 G. It is interesting to note that in the region of 39 G there is a jump in the damping rate. It is possible to suggest that it is a half integer resonance: the forced frequency from the field of positron bunches is two times smaller than the repetition rate of the secondary electron emission.
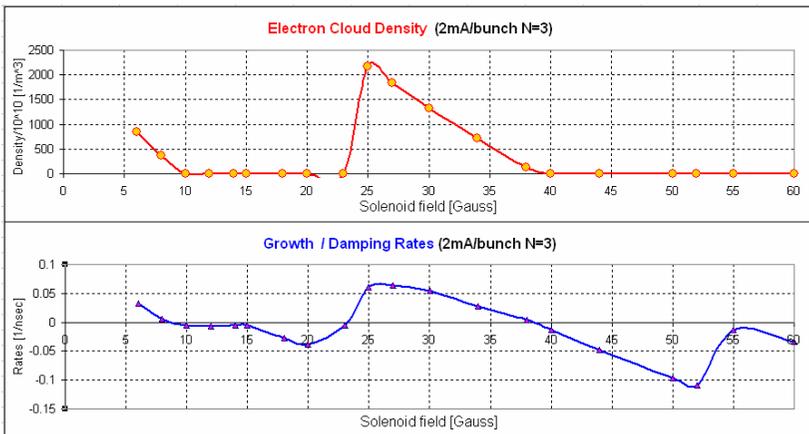
Figure 10. Electron cloud saturated density and growth/damping rates for a positron train with bunch spacing by 3.

In the intermediate case when the positron bunch train has a spacing of 3 RF buckets the resonance value for solenoidal field is 25 G and zones with no multipacting after 40 G and between 10-24 G. Half integer resonance is in the region of 53 G (Fig.10).

9. Longitudinal electron cloud field

In order to build electron cloud, positron bunches have to use some amount of their kinetic energy. The field, which is responsible for the energy transformation, is longitudinal electric field. When the cloud is already built, this longitudinal field acts as an oscillating force on the cloud electrons and gives, at the same time, an additional energy variation inside the positron bunches. The head of the positron bunch is accelerated and the tail is decelerated. This action of the longitudinal field is similar to the action of RF fields in a cavity. As a result the positron bunches will have different lengths throughout the train. This effect can be checked in experiment. Fig. 11 shows a longitudinal field together with the positron bunch shape. The solenoidal field is 38G, bunch spacing is two RF buckets. The energy variation along a positron bunch is more than 100 V on 1 m of vacuum chamber. The total length of all straight sections in the positron storage ring is around 740 m, so the total effect of the electron cloud can be of order of 74 kV, which is equivalent to 185 kV of RF voltage at 476 MHz. It is easy to make an analytical estimation for this effect. The additional variation of energy in a positron bunch for one meter of vacuum chamber due to the electron cloud is

$$\Delta W / l = \frac{2}{\lambda_{RF} * N} m_e c^2 * \left(\frac{Z_0 I_{bunch}^+}{m_e c f_{rev} 2\pi a} \right)^2$$

This formula gives the same result, as we got from the computer simulations.
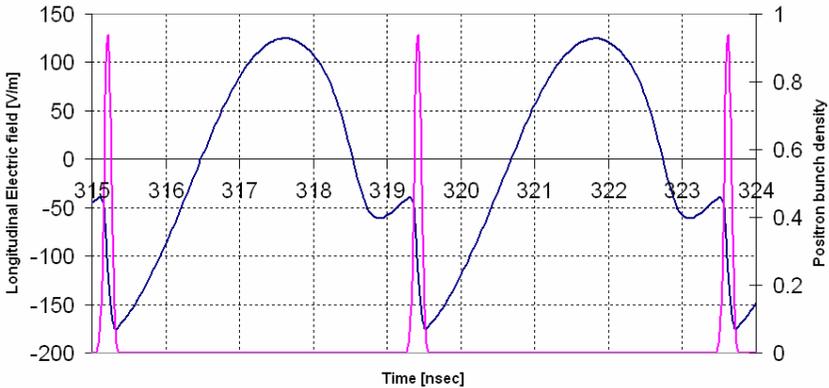
Figure 11. Longitudinal electric field in electron cloud. Solenoidal field is 38 G, bunch spacing is two RF buckets.

10. Conclusions

Computer simulations show that increasing the number of positron bunches by a factor of two and keeping the same current per bunch means that the solenoidal field needs to be doubled in order to keep the same electron cloud density. Currently PEP-II solenoids in streight sections have neaby 30 gauss [1], so we need to increase the field up to 60

gauss. Fortunately there are regions that are "free of multipacting" at smaller values of solenoidal field where the electron cloud density can not get very high. This prediction of low cloud density can be checked in experiment.

Acknowledgement

The authors thank R.E.Kirby for important information about secondary emission process and presented files of SEY measurements. We thank A.Kulikov who first pointed out this problem and M.Sullivan for fruitful discussions and very useful remarks.

References
[1]   A.Kulikov, A.Fisher, S.Heifets, J.Seeman, M.Sullivan, U.Wienands, W.Kozanecki, "The Electron Cloud Instability at PEP-II B-Factory", PAC'2001, Chicago, June 2001, p. 1903
[2]   J.Seeman et al., "PEP-II Status and Future Plans", EPAC 2002, Paris, June 2002, p.434
[3]   R.Kirby, Private communication
[4]   R.E.Kirby and F.K.King, "Secondary electron emission yield from PEP-II accelerator materials", NIM A 469(2001) p.1
[5]   P.A.Redhead, J.P.Hobson, E.V.Kornelsen, "The Physical Basis of Ultrahigh Vacuum", AIP, New York, 1993, p134.

RF heating in the PEP-II B-factory vertex bellows

A. Novokhatski, S. Weathersby

Stanford Linear Accelerator Center, Stanford University, CA 94309

Abstract. A study of RF modes is carried out for the complicated geometry of the shielded vertex bellows near the IP region of the PEP-II B-factory. Calculations indicate several monopole, dipole and quadrupole modes can exist in the bellows structure near the experimentally observed frequency region of 5 GHz. The observed modes are correlated to the bellows heating and are excited by both electron and positron beams. These modes can heat the bellows by coupling through the RF shield fingers.

1. Introduction

The PEP-II B-Factory collides 1.7 A of 3 GeV positrons with 1.0 A of 10 GeV electrons, in trains of several hundred bunches. The bunch length is 1.3 cm. Within the BaBar detector near the IP, where the two beams share a common vacuum chamber, anomalous heating is observed at thermocouples situated on a shielded bellows structure at the juncture of a beryllium beam pipe with a copper vacuum chamber. The heating is determined to be due to higher order modes [1]. A signal from a nearby BPM was used to measure the spectrum of the fields excited by the beams. These observations reveal several high Q modes correlated with the bellows temperature. Computer calculations in 2D and 3D with the MAFIA [2] program have found eigenmodes which can exist in the bellows structure near the observed frequencies. Further calculations are performed modeling the vacuum chamber and bellows together in 3D as coupled cavities from which coupling parameters are obtained for the dipole and quadrupole modes. From the coupling parameters and the loss factor for the beam chamber an estimation of the power dissipated in the bellows can be made. This study complements and extends the pioneering work carried out by Stan Ecklund et al. [1] during the last physics run.

2. The bellows structure

Figure 1 gives an overview of the bellows area. Positrons are incident from the left, electrons from the right. The IP is located roughly 20 cm to the right of the bellows. The bellows are shielded from the beam chamber by a series of 16 metal fingers azimuthally separated by gaps or slots. The variations in the beam pipe vertical cross section couple the beam field and high order transverse modes. It is the transverse modes which are expected to couple into the bellows through slots between the fingers. The slots are made to vary in length from 10 to 13 mm in response to thermal expansion and contraction of surrounding structures. During a recent maintenence period the vertex chamber was extracted from the BaBar detector and the slots were found to be fully extended at 13 mm.
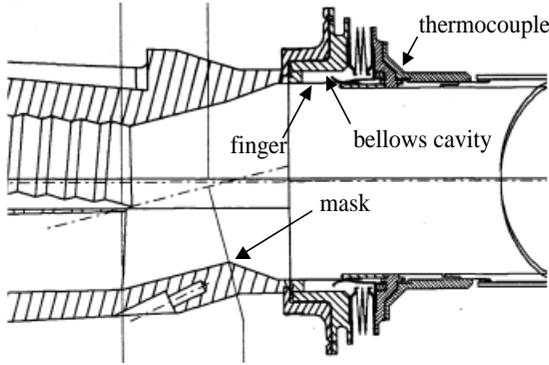
Figure 1. Bellows structure and vacuum chamber showing synchrotron masks and site of thermocouple with high temperature readings.
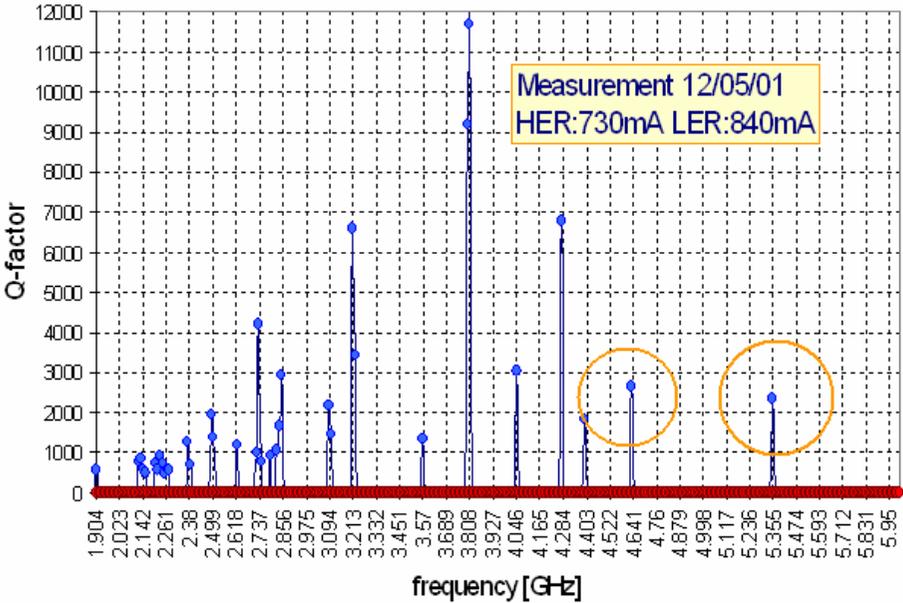
Figure 2. FFT of a gated oscilloscope BPM button signal in the gap between trains. The circled peaks show amplitude correlation with the thermocouple temperature.

3. Spectrum measurements

The colliding bunch trains have an intervening gap of about 350 ns required for the ramp-up of an abort kicker and the clearing of ions for the electron ring. A beam position monitor button electrode located 50 cm from the bellows provides a signal to a HP54120 high frequency oscilloscope and a R&S gated spectrum analyzer. The signal is gated in time to coincide with the gap where no beam is present. An FFT of the oscilloscope

signals is shown in figure 2. To calculate the equivalent Q-factor the amplitudes are measured in small time windows at the beginning and the end of the gap.

The same spectrum was also observed with the gated spectrum analyzer. Within a forest of peaks in the 5 GHz region several modes are identified as having amplitudes correlated with the bellows temperature. The amount of thermal power dissipated in the bellow vs. time was calculated from the temperature data. In figure 3 this calculation is compared with the amplitude of one of the peaks.

05/03/02 B-side

f=5593274 —— Heat Ampl.
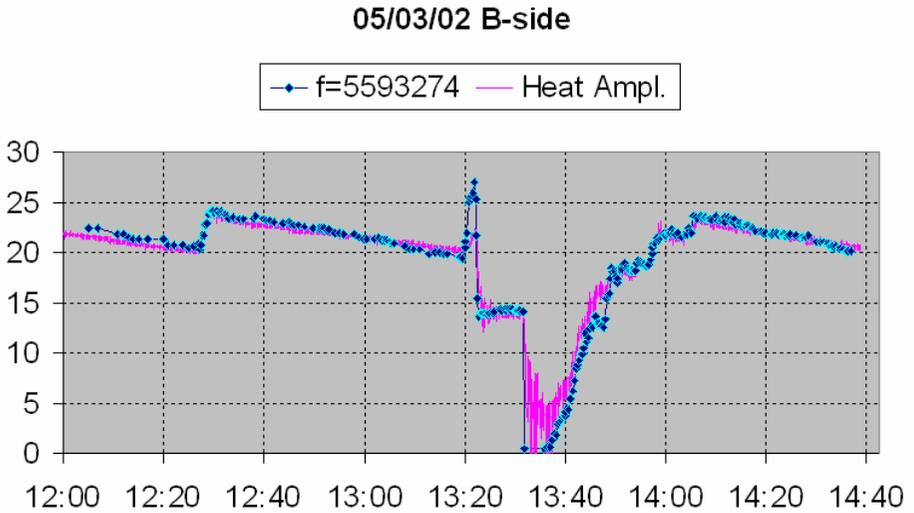
Figure 3. Estimated thermal power (pink) from thermocouple data and measured peak amplitude (blue) of a 5.6 GHz mode vs. time.

The figure includes data from a beam abort in which one of the beams was lost followed by a subsequent loss of the remaining beam. The slow rise after the beam loss reflects filling the rings back to nominal currents. The data indicate contributions from both beams to the heating and mode power.

4. A simple 3D coaxial model of the bellows

To understand the structure of possible RF modes in the bellow cavity we start with a simple model of a coaxial cavity with a rather large inner conductor. The effect of the vacuum chamber and slots as well as the bellows convolutions are ignored. Eigenmodes of such a structure are easily visualized using MAFIA. The cavity has an inner and outer radius of 26.5 cm and 29.6 cm respectively and a length 22.3 cm. The field patterns in the bellows cavity identify the modes as monopole, dipole and quadrupole according to the azimuthal variation period. In figure 4 the length of the cavity is exaggerated to illustrate the field patterns more clearly for the dipole case.
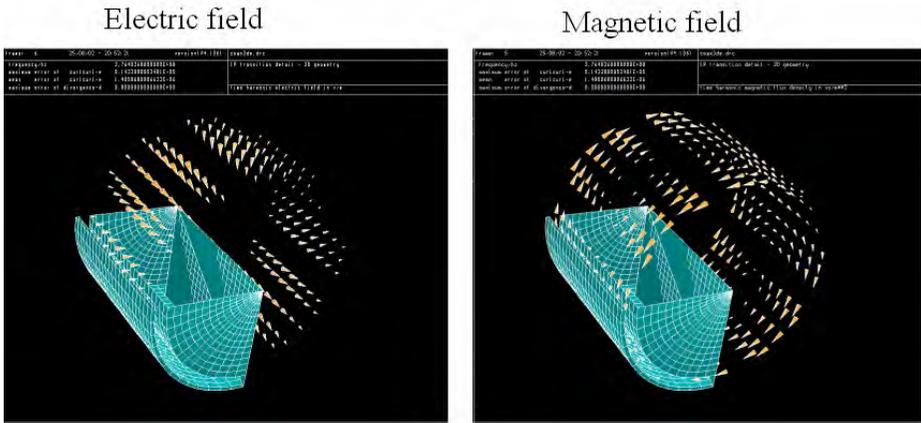
Figure 4. Electric and magnetic fields of the dipole eigenmode for the 3D coaxial model of the bellows given by MAFIA. The length of the cavity is exaggerated to illustrate the field patterns.

In the monopole case the magnetic field lines are exclusively azimuthal. For the dipole case the magnetic field shows two areas of circulation around the electric field nodes and assume mainly longitudinal directions near the electric field anti-nodes. We expect the wall currents and therefore the heating to exhibit this pattern. The quadrupole mode exhibits the same features.

5. Detailed 2D results

Two dimensional eigenmode calculations for the bellows cavity with more detailed geometry finds dipole and quadrupole frequencies shown in table 1. The monopole frequency is computed using a proprietary code NOVO [4]. Dipole and quadrupole modes are obtained with MAFIA. The fingers and beam chamber are not modeled in this calculation. The magnetic field energy density for the monopole, dipole and quadrupole modes integrated over the azimuthal angle ϕ are shown in figures 5-7. There is high energy density at the bottom of the bellows convolutions for the dipole and quadrupole case. This magnetic energy density corresponds to large longitudinal magnetic flux.

The azimuthal variation of the magnetic fields have the same characteristics as the simple coaxial cavity model discussed above. In figures 8 and 9 the dipole magnetic fields are plotted for two different azimuthal planes: $\phi=0$ and $\phi=\pi/2$. At $\phi=0$ the magnetic field is azimuthal while at $\phi=\pi/2$ the magnetic field lines have a longitudinal return path mainly along the bottom of the bellows.

Table 1. Eigenmode frequencies for the 2D bellows cavity. The monopole mode was calculated with the proprietary code NOVO [4].

4.74 GHz	Monopole
5.46 GHz	Dipole
6.19 GHz	Quadrupole
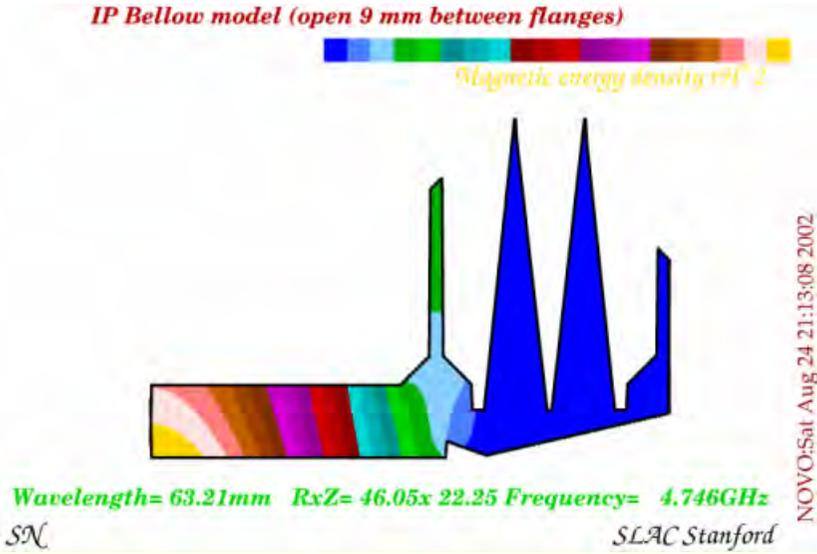
Figure 5. Monopole 4.75 GHz magnetic energy density.
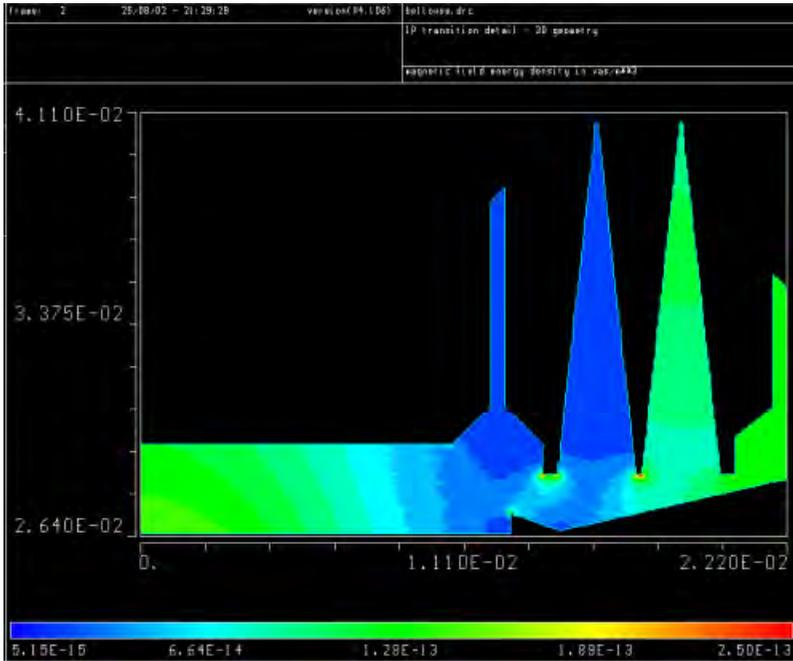
Figure 6. Dipole 5.46 GHz magnetic energy density.

264
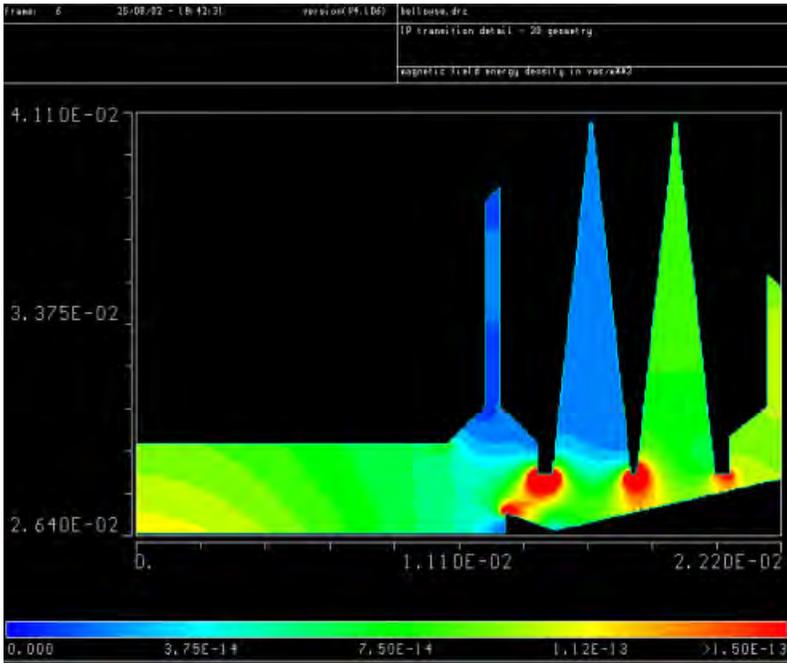
Figure 7. Quadrupole 6.19 GHz magnetic energy density. The scale is truncated at the high end to enhance the density pattern in the hot spots.
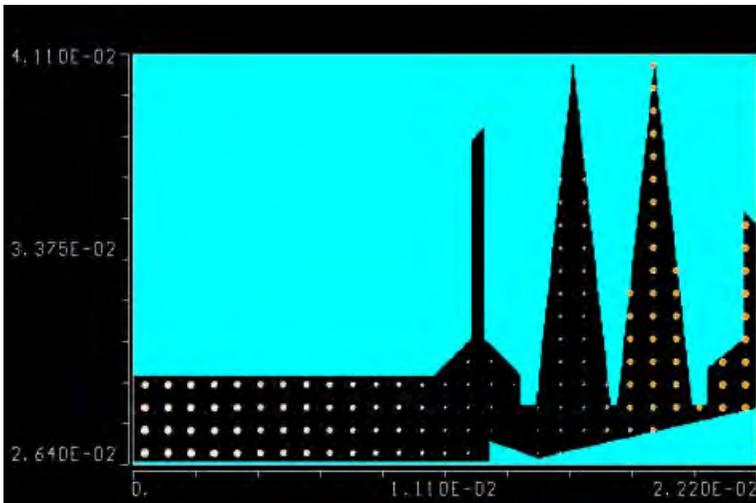
Figure 8. Dipole mode magnetic field at φ=0 is azimuthal.
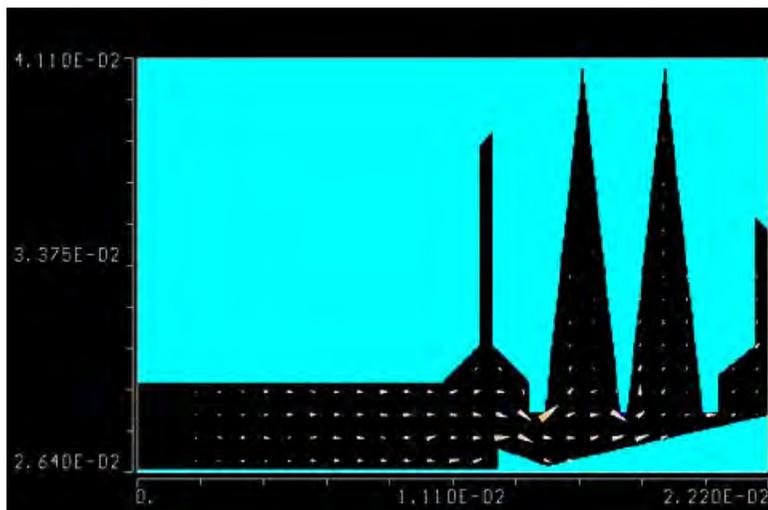
Figure 9. Dipole mode magnetic field at $\phi = \pi/2$ is mainly longitudinal.

6. Coupling studies

So far we have determined the existence of modes in the bellows cavity which are likely responsible for the observed heating. To examine a possible excitation mechanism we now construct an inner beam cavity with slots in its outer walls linking the outer bellows cavity in its simple coaxial configuration. The evolution of a particular mode in each cavity is observed as the longitudinal dimension of the beam cavity is varied. The bellows cavity dimensions are held fixed in these studies. Furthermore, the slots are kept at the center of the longitudinal dimension of the beam cavity, to minimize the effect of the cavity walls. This construction is shown in figure 10.
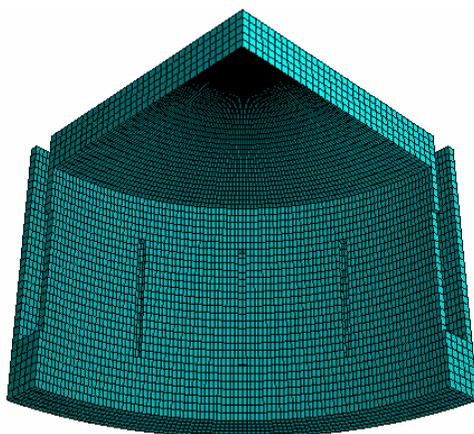
Figure 10. Geometry for the dipole coupling studies showing the beam cavity, bellows cavity and slots. Only half of this geometry is used in the quadrupole studies.

266

Within the regime where no coupling is present, the bellows cavity frequency remains constant while the beam cavity frequency changes with the change in length. If there is no coupling, the change in the beam cavity frequency will have no effect on the fixed frequency of the bellows cavity. If there is coupling, one expects a shift in the bellows cavity frequency as the beam cavity frequency approaches the bellows cavity frequency. The closest approach of the two frequencies yields the degree of coupling. The larger the closest approach, the larger the coupling. We give the coupling α as the minimum frequency separation over the average frequency.

6.1 Boundary considerations

Symmetry is used to keep computational costs low. The longitudinal boundaries are considered to be conductive endplates. For the quadrupole mode only $1/8^{th}$ of the structure is necessary in the azimuthal dimension. The azimuthal boundaries are electric at $\phi=0$ and magnetic at $\phi=\pi/4$. The dipole simulation requires $1/4^{th}$ of the geometry with the same boundary conditions. The electric field for a beam cavity quadrupole mode computed with MAFIA is shown in figure 11.
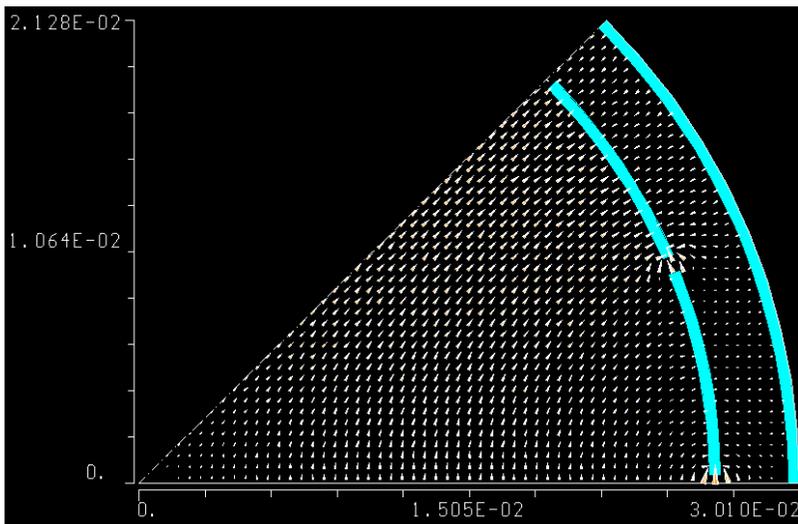
Figure 11. Quadrupole electric field for the beam cavity computed with MAFIA.

6.2 Quadrupole coupling

For a given slot size, frequencies of the quadrupole mode for both cavities and their difference are plotted as a function of beam cavity length in figure 12. As the chamber length is decreased from –4 mm, the upper bellows mode which is constant outside the coupling region begins to shift as the lower cavity frequency nears. A non-zero minimum difference in frequency of 229 MHz indicates the degree of coupling for the quadrupole modes. We give the quadrupole coupling α as the percent of the minimum frequency separation to the average mode frequency which is roughly 2% for this case.

Monopole mode coupling is not considered since the magnetic field lines are purely azimuthal, however, if the beam axis and the cavity axis are not parallel, a longitudinal coupling component will be introduced.

To extrapolate, we investigate coupling for different slot widths. The minimum frequency separation as a function of slot size for the quadrupole modes is shown in figure 13, where the horizontal axis is the fractional width of the real slot width of .81 mm. The data is fitted with an analytic expression derived from electric and magnetic polarizabilities of small apertures [3] which is given in terms of slot width w and length l and proportional to

$$\frac{1}{\ln(4l/w)-1}.$$

Both the data and fit are fairly linear and flat in the region of interest. Practically speaking, reduction of coupling for the quadrupole modes by narrowing the slots is not feasible.
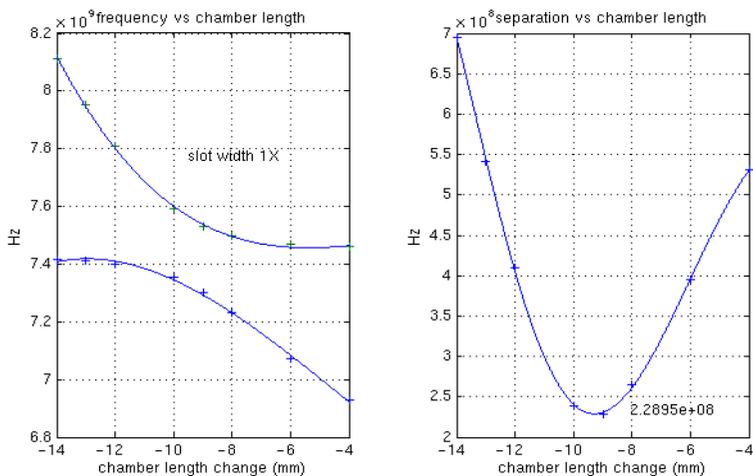
Figure 12. Quadrupole mode frequency for the beam and bellows cavities and their difference as a function of beam cavity length for a slot width of 0.81 mm. The minimum separation is 229 MHz.
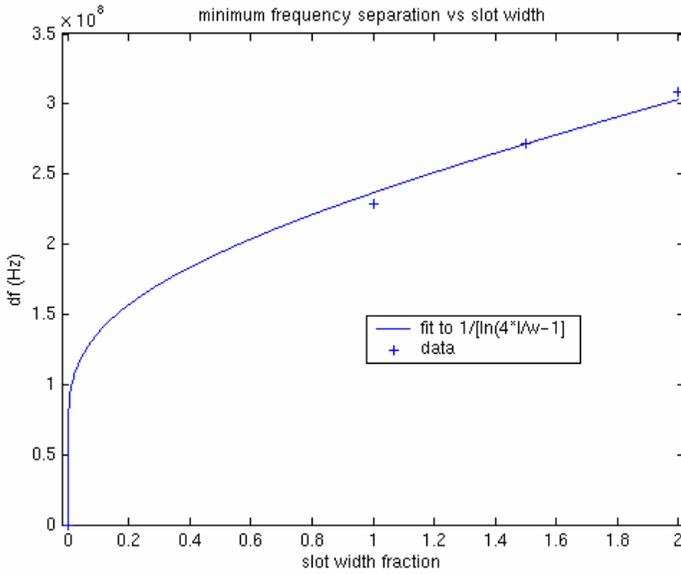
Figure 13. Minimum mode separation vs. fractional slot width. The data is fit to an analytic expression derived from the polarizability of small apertures.

6.3 Dipole coupling

Figure 14 is a plot of the dipole field in the beam chamber and bellows cavity obtained from a MAFIA eigenmode calculation. The results for the dipole case is shown in figure 15 and indicates a smaller coupling. The minimum frequency separation is 75 MHz yielding a 1% coupling.
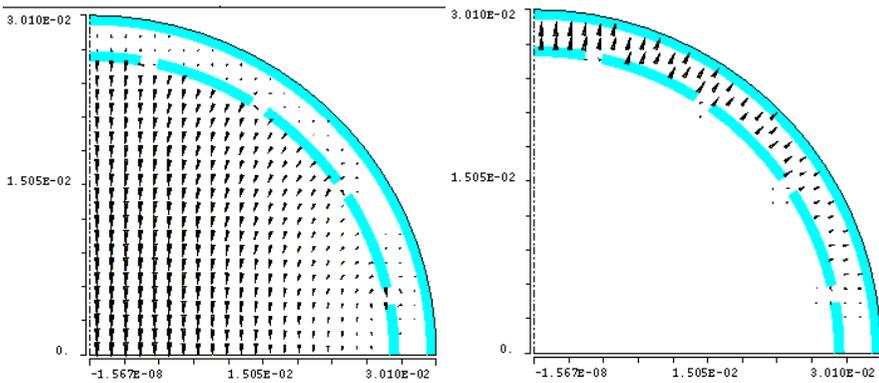
Figure 14. Dipole electric field for the beam and bellows cavity calculated from MAFIA.
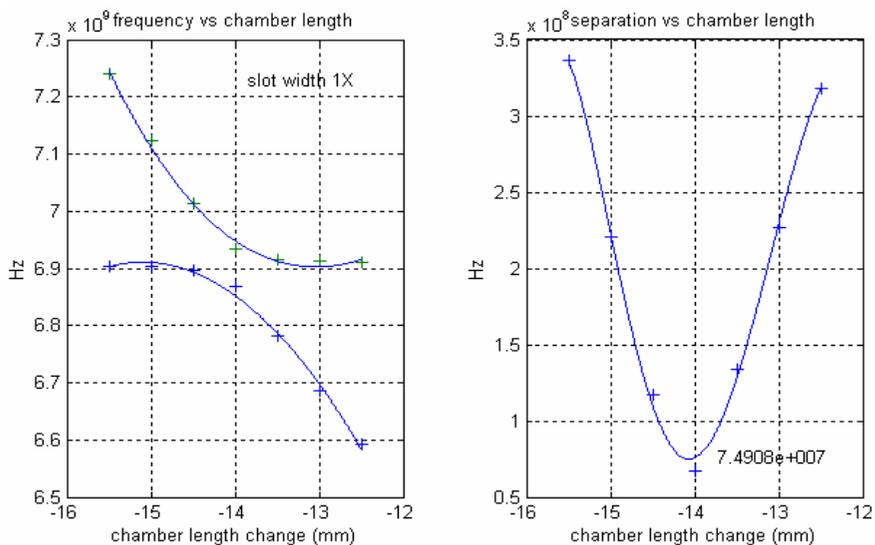
Figure 15. Dipole mode frequency for the beam and bellows cavities and their difference as a function of beam cavity length for a slot width of 0.81 mm.

7. Power in the bellows

One can estimate the power dissipated in the bellows cavity, given the loss factor for the actual beam pipe structure, the currents, and the coupling constants α from the above calculations. First we calculate the loss factor.

The loss factor for the complex beam pipe geometry of figure 1 is calculated with several methods. The nature of the problem is three dimensional and so is suited for MAFIA's 3D time domain solver. A one coulomb gaussian bunch is propagated through the mesh of figure 16. This gives the longitudinal wake shown in figure 18 which when convoluted with the gaussian beam of bunch length 1.3 cm yields a loss factor $\kappa = .06$ V/pC.
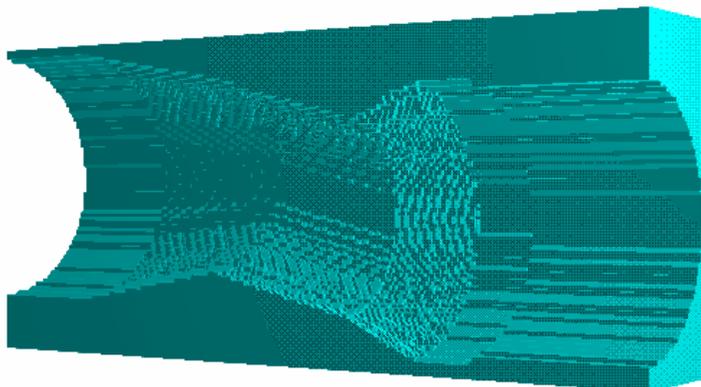
Figure 16. Mesh for MAFIA wake potential calculation.

One can make 2D calculations with the NOVO[4] wake field program by taking the top and bottom profiles of the beam pipe separately as axially symmetric structures as illustrated in figure 17. The NOVO program directly computes κ =0.031 and 0.01 V/pC respectively for the top and bottom profiles of the beam chamber. A comparison of the two methods is shown in figure 18.
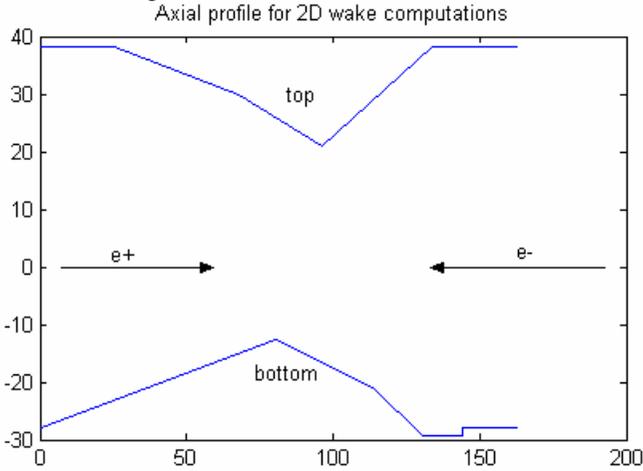
Figure 17. Templates for axially symmetric 2D NOVO loss factor calculations.
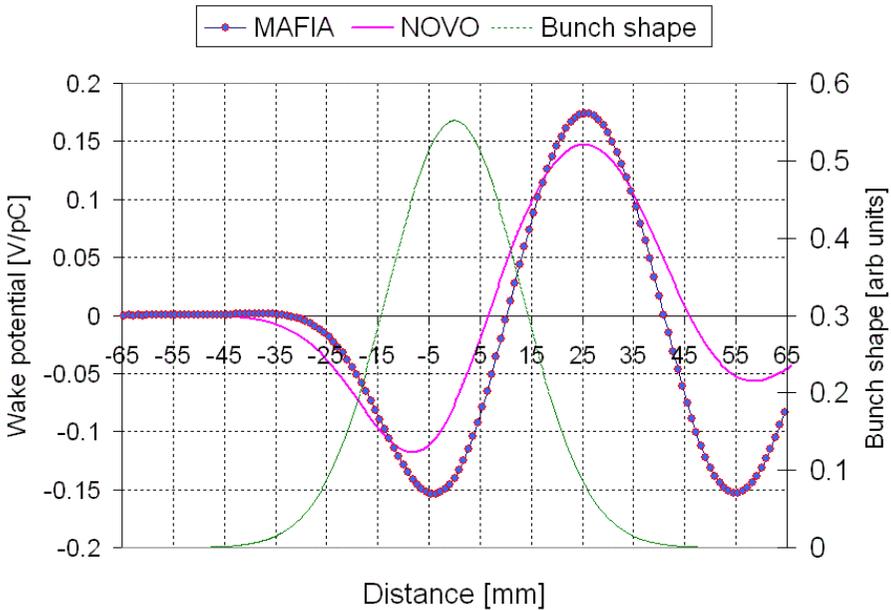
Figure 18. Longitudinal wake potentials for a gaussian bunch (green) of length 1.3 cm. The dotted line is from the 3D MAFIA time domain solver. Solid red is from the 2D NOVO code using axially symmetric top profile.

We make the assumption that all the beam generated fields are scattered by the masks, taper and axial offsets for the complicated beam pipe geometry of figure 1. We assume this irregular geometry will convert the beam fields into transverse modes which can couple into the bellows. PEP-II typically runs with currents of $I_+ = 1.7$ A and $I_- = 1.0$ A respectively for the positrons and electrons at a revolution frequency of f=136 kHz with N=837 bunches. In this case we can estimate the beam pipe power as

$$P_{beam} = \frac{\kappa}{fN}(I_+^2 + I_-^2)$$

which is about 2 kW given the MAFIA 3D loss factor and 1 kW for the NOVO 2D loss factor.

To estimate the fraction of the beam power coupling into the bellows cavity we further assume all power incident in the bellows is dissipated there. We can write for the bellows power:

$$P_{bellow} = \alpha P_{beam}.$$

With a 1% coupling α=.01 (dipole mode)   this corresponds to 20 W and 10 W respectively for the MAFIA 3D and NOVO 2D loss factors, 40 W and 20 W respectively for the 2% (quadrupole mode) coupling. These results are consistent with other independent estmations for similar geometries[5]. If the bunch length is decreased by a factor of 2, the loss factor and hence the power will go up by a factor of 4.

8. Conclusions

Bellows heating is observed to be correlated with RF mode power in the 5 GHz region. Higher order modes near this frequency are shown to exist in the bellows cavity and can be induced by the beams via coupling through the shield fingers. Calculations show the quadrupole and dipole modes as being likely candidates, with coupling a strengths of about 1 to 2%. Depending on the value obtained for the loss factor for the beam chamber, an estimation of 20 to 40 Watts of power is dissipated in the bellows by the quadrupole mode and roughly half as much by the dipole mode. During an extended down time additional cooling has been installed for the bellows as well as additional thermocouples. The thermocouples are located radially around the bellows in hopes of verifying the existence of the quadrupole or dipole mode in the thermal profile.

References

[1]   F. Decker, S. Ecklund, A. Fisher, A. Kulikov, M. Sullivan "High Order Mode Heating Observations in the PEP-II Interaction Region", SLAC. PAC-2001-RPPH142, SLAC-PUB-9372, Aug 2001. 3pp. Presented at IEEE Particle Accelerator Conference (PAC2001), Chicago, Illinois, 18-22 Jun 2001.

[2]   The MAFIA collaboration "User Guide", CST GmbH, Darmstadt, Germany.

[3]   Bruno W. Zotter and Semyon A. Kheifets, "Impedances and Wakes in High - Energy Particle Accelerators", 1998, World Scientific.

[4]   A. Novokhatski, M. Timm and T. Weiland, "Transition Dynamics of the Wake Fields of Ultra Short Bunches", Proceeedings of the 1998 International Computational Accelerator Physics Conference.

[5]   S. Heifets, private communication.

Optimization of RFQ structures

A. D. Ovsiannikov

Saint-Petersburg State University, Russia

Abstract. Previously a new approach to optimization of RFQ structures was proposed [4-8]. Corresponding code was developed to realize this approach. In this paper, a new version of this code is considered. Different RFQ structures, obtained as result of optimization, are also considered.

1. Introduction

The theory of linear accelerators design was developed long ago. But, in connection with wide implementation of accelerator technology in various manufacturing processes, modern problems appear to designers. And, by our opinion, it is impossible to solve some of them through frames of traditional approaches to accelerating and focusing structures parameters calculations. At the present time, problems of constructing of compact, high-current machine with high acceleration speed and minimal losses are especially important. At this high intensity and small emittance beams with prescribed phase and energy characteristics are required.

One possible path for solving of such a type of problems is the optimization approach. In the optimization code BDO–RFQ developed by Saint-Petersburg State University for accelerating structures with RFQ, wide use of various mathematical models of particles dynamics, sequential execution of optimization with more complicated models at each consecutive stage, employment of computer modelling opportunities and user-friendly interface are provided.

2. Mathematical models of optimization

The following mathematical model of longitudinal motion optimization described by a system of integro-differential equations is considered:

$$\frac{dx}{dt} = f(t, x, u), \tag{1}$$

$$\frac{dy}{dt} = F_1(t, x, y, u, du/dt) + \\ \int_{M_{t,u}} F_2(t, x, y, z_t)\rho(t, z_t)dz_t = F(t, x, y, u, du/dt), \tag{2}$$

$$\frac{d\rho}{dt} = -\rho(t, y) \cdot div_y F(t, x, y, u, du/dt), \tag{3}$$

with initial conditions that

$$x(0) = x_0, \tag{4}$$

$$y(0) = y_0 \in M_0, \tag{5}$$

$$\rho(0, y(0)) = \rho_0(y_0), \quad y_0 \in M_0. \tag{6}$$

Here $t \in T_0 = [0,T]$ — is independent variable, $x \in R^n$ and $y \in R^m$ — vectors of phase variables, $u = u(t)$ — r- dimensional control vector function, T — is fixed number, $\rho_0(y_0)$ is some non-negative continuous function. The set $M_{t,u}$ is a shift of set M_0 along trajectories of system (2) under fixed control $u = u(t)$ and according program motion $x(t)$, $div_y F = \sum_{i=1}^m \partial F / \partial y_i$.

Equation (1) describes a dynamics of a synchronous particle with initial condition (4). In this paper, we will consider this motion as the program one. Equation (2) is a motion equation in deviation from the motion of the synchronous particle, describing perturbed motions on initial data (5). Equation (3) is an equation of change of density distribution of particles $\rho = \rho(t) = \rho(t, y(t))$ along trajectories (2) with given at initial moment law $\rho_0(y_0)$ of distribution of density of particles in set M_0.

This introduced mathematical model of control accounts for interaction of particles in the beam. Here vector-function F_1 determines influence of external fields on a particle, and vector-function F_2 — interaction of particles.

Let us consider the following functionals:

$$I_1(u) = \int_0^T \varphi_1(t, x(t), u(t)) dt + g_1(x(T)), \tag{7}$$

$$I_2(u) = \int_0^T \Phi(w_1(t)) dt + G(w_2), \tag{8}$$

where

$$w_1(t) = \int_{M_{t,u}} \varphi_2(t, x(t), y_t, \rho(t, y_t), u(t)) dy_t, \tag{9}$$

$$w_2 = \int_{M_{T,u}} g_2(y_T, \rho(t, y_T)) dy_T. \tag{10}$$

Here Φ, G, φ_1, φ_2, g_1, g_2 are non-negative, continuously differentiable functions.

Functional (7) characterizes program motion dynamics or dynamics of the synchronous particle, and functional (8) estimates behavior of the beam trajectories.

The following functional

$$I(u) = I_1(u) + I_2(u) \tag{11}$$

simultaneously assessing dynamics of program motion and particle beam dynamics with account their density of distribution and their interactions was introduced.

A new approach to the solving of optimization problem for charged particles dynamics in accelerators was suggested on the basis of this mathematical model. It includes: construction of mathematical model of controlled dynamical process; choice of control functions or parameters of optimization; construction of quality functionals, which allow efficient evaluation of various characteristics of examined controlled

motions; analytical representation of the functionals' variations, which allow to construct various methods of optimization for quality functionals; construction of methods and algorithms of optimization.

It should be noted that we used model that allows to separate longitudinal motion optimization from transverse motion one.

3. BDO-RFQ code

BDO-RFQ code is created in Matlab. BDO-RFQ code is the package of programs for modeling and optimization of dynamics of charged particles in RFQ structure. The complex of programs is supposed to allow the usage of various models of dynamics of charged particles in the process of choice of parameters of the RFQ structure and its optimization. Particular attention was paid to visualization of processes of modeling and optimization and to the creation of user-friendly interface.
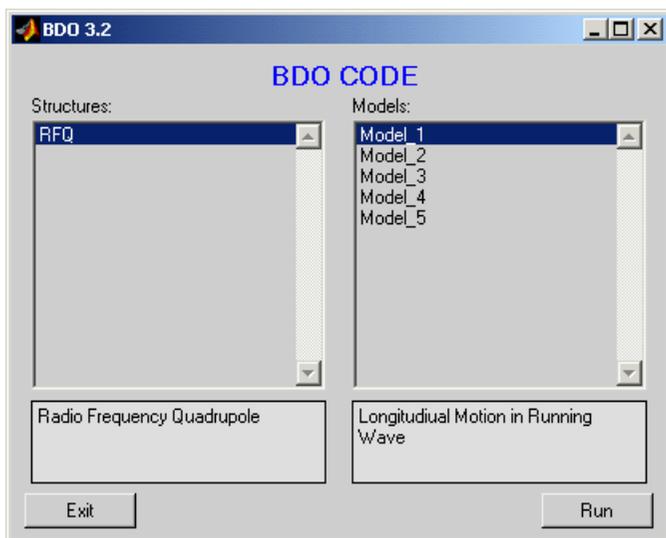
Figure 1. BDO-RFQ code start window.

The package of programs is developed with the consideration of demand for the extension of a range of examined models of dynamics and possibility of the enlargement of a number of criteria (quantity of functionals) of the estimation of particles dynamics. With this aim the complex of programs is structurally divided into command shell and blocks, which describe various models of particles' dynamics. With this goal, various models are described as special procedures containing all necessary information about a particular model such as: parameters of the model, equations of the motion, algorithms of calculation of functionals of quality of the dynamics for given model, the procedure of re-calculation control parameters of one model of dynamics into another, the range of values (characterizing the dynamics) put on the display, etc.

Given this approach to the construction of an integrated system of modeling, optimization and visualization of charged particles dynamics in RFQ structure in its nature is like object-oriented programming. In this case it is possible to consider models as objects which describe dynamics, while uniting command shell realizes standard (typical) operations over models, i.e. choice of model, calculation of dynamics,

optimization, visualization of characteristics of dynamic process and a transition from one model to another.

The representation of models of control as various separate blocks, the possibility of translation of information between them and the unified system of model's control allow a realization of step-by-step optimization of such complicated object as the RFQ structure. With this, it is natural to use simple models of dynamics on initial steps of optimization and on follows steps of it more developed models are necessary for verification of results and for further optimization.

3.1. *Longitudinal motion*

Under consideration of longitudinal motion optimization criteria are the following: obtaining of maximal capture of particles into acceleration mode; obtaining of required or maximal possible output energy; minimization of defocusing factor effect; obtaining of monotonicity of particles' grouping.
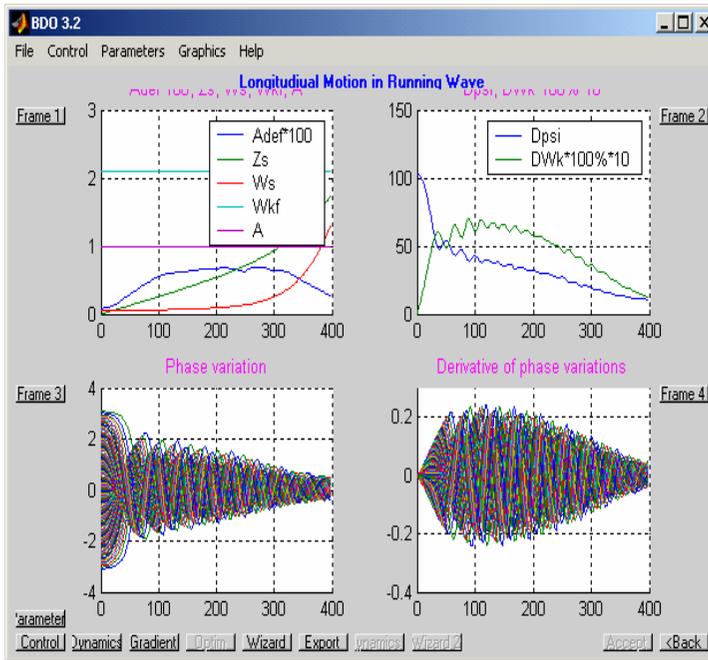
Figure 2. Main working screen of BDO-RFQ code.

The problem of deuteron beam dynamics optimization with following parameters was considered: frequency - 433MHz, initial energy - 60KeV, intervene voltage - 100kV, current - 25mA. Results of the optimization are shown at figures 2-4. At output energy 1MeV, the length of the calculated structure was 1585 mm (356 of cells). Variations of synchronous particle phase and intensity of acceleration were considered as control functions.
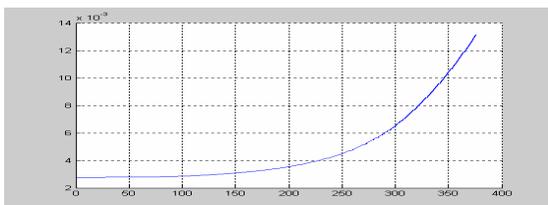
Figure 3. Cell length variation in dependence of cell number.
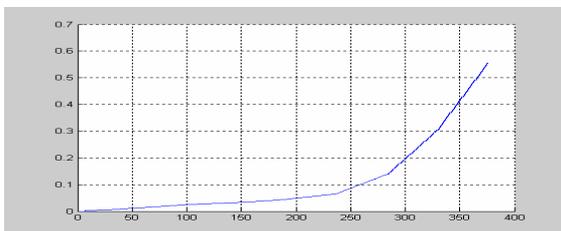
Figure 4. Acceleration effectiveness variation in dependence of cell number.

Also, the shorter variant (with length 1295 mm) of the RFQ structure was found. Characteristics of the synchronous particle and the beam are shown at figures 5 and 6.
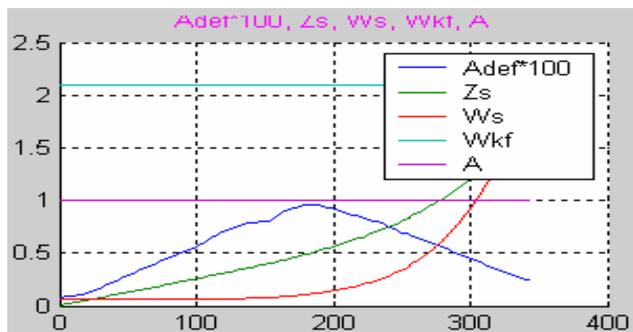
Figure 5. Variations of defocusing factor, energy and longitudinal coordinate of synchronous particle along structure.
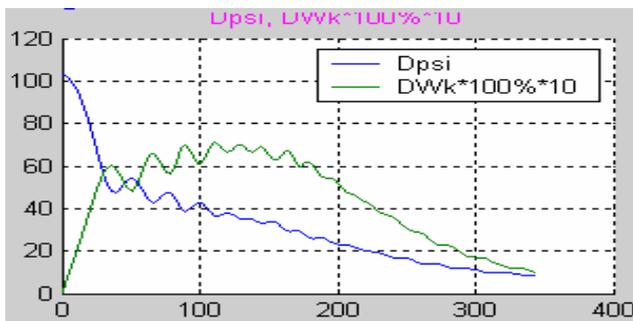
Figure 6. Variations of rms beam phase and energy width along structure.

278

3.2. *Transverse motion*

Under consideration transverse motion we used different equations describing transverse motion and applied corresponding optimization methods for finding optimal parameters [1-3,6-9]. In particular, we consider the case when particles were uniformly distributed along the beam cross-section. This allows us to use Kapchinsky-Vladimirov self-consistent distributions and to design according mathematical models of optimization. Minimal radius and intervene voltage are used as controls while already found longitudinal motion controls are preserved.

Figures 7-9 correspond to the first variant of the defined longitudinal dynamics.
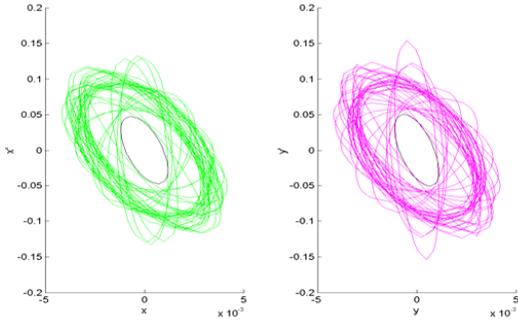
Figure 7. Acceptance at the beginning of optimized matching structure.
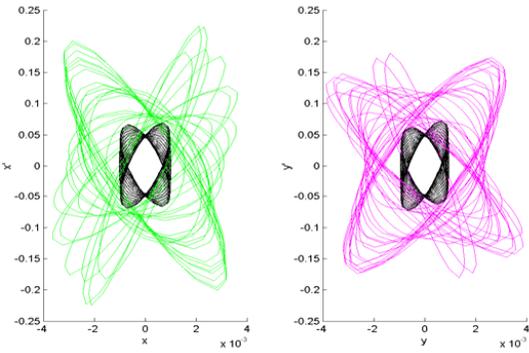
Figure 8. Beam matching at the beginning of the accelerating structure.
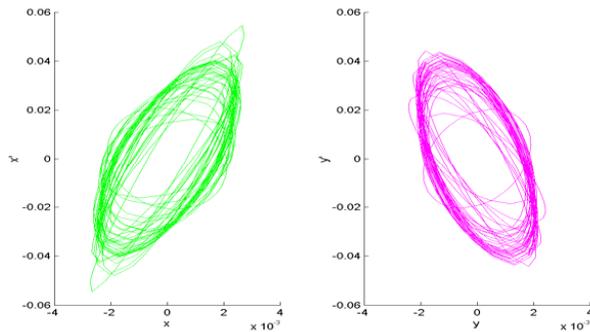
Figure 9. Emittance of beam at the end of accelerator.

4. Conclusion

Conducted calculations have shown the effectiveness of step-by-step optimization on various stages of the process of optimization. Thus, the minimization of the defocusing factor on the stage of the optimization of longitudinal motion allows the focusing of the beam with the conservation of chosen dynamics of longitudinal motion. Suggested mathematical models can be also applied to other types of accelerators: traveling wave accelerator, accelerator with drift tubes and linear undulator.

The author thanks Ovsyannikov D A, Durkin A P, Rubtsova I D, Svistunov Yu A, Snopok P V, Antropov I V, Kozynchenko V A, Merkuriev S V for taking part in code development and discussing of optimization results.

References

[1] Ovsyannikov D, "Modeling and Optimization of Charged Particle Beam Dynamics", Leningrad State University, Leningrag, 1990 (in Russian).
[2] Drivotin O I, Loukianova A E, Ovsyannikov D A, Garvish Yu N, Svistunov Yu A, Vorogushin M F, "The Choice of Acceleration Structure for PET-System", EPAC'96, Proceedings, Barselona, Spain, Vol.1, p.783-785, 1996.
[3] Ovsyannikov D, "Modeling and Optimization Problems of Charged Particle Beams Dynamics", Proceedings of the 4th European Control Conference, Brussels, 1997, pp. 390-394.
[4] Bondarev B I, Durkin A P, Ovsyannikov A D, "New Mathematical Optimization Models for RFQ Structures", Proceedings of the 1999 Particle Accelerator Conference, New York, 1999, pp. 2808-2810.
[5] Ovsyannikov A D, "New Approach to Beam Dynamics Optimization Problem", Proceedings of the 6th International Computational Accelerator Physics Conference ICAP 2000.
[6] Bondarev B, Durkin A, Vinogradov S, Ovsyannikov D, Ovsiannikov A, "RFQ Optimization: Methods&Codes", Proceedings of the 6th International Computational Accelerator Physics Conference ICAP 2000.
[7] Bondarev B I, Durkin A P, Ivanov Yu D, Shumakov I V, Vinogradov S, Ovsyannikov A D, Ovsiannikov D A, "The LIDOS.RFQ.Designer Development", Proceedings of the 2001 Particle Accelerator Conference.
[8] Ovsyannikov D A, Ovsyannikov A D, Bondarev B I, Durkin A P, "Modeling and Optimization Code for RFQ Structures", The 9th International Workshop on Beam Dynamics and Optimization, St.Petersburg, Russia, 2002.

[9]   Ovsyannikov A D, Snopok P V, "Matcher Optimization for RFQ Structures", The 9th International Workshop on Beam Dynamics and Optimization, St.Petersburg, Russia, 2002.

A multigrid based 3D space-charge routine in the tracking code GPT

Gisela Pöplau†‖ ¶**, Ursula van Rienen**†**, Marieke de Loos**‡ **and Bas van der Geer** §

† Institute of General Electrical Engineering, Rostock University, Germany
‡ Eindhoven University of Technology, The Netherlands
§ Pulsar Physics, Soest, The Netherlands

Abstract. Fast calculation of 3D non-linear space-charge fields is essential for the simulation of high-brightness charged particle beams. We report on our development of a new 3D space-charge routine in the General Particle Tracer (GPT) code. The model is based on a non-equidistant multigrid Poisson solver that is used to solve the electrostatic fields in the rest frame of the bunch. Since the multigrid Poisson solver depends only linearly on the number of mesh points for the discretized electrostatic problem the space-charge routine scales linearly with the number of particles in terms of CPU time. This performance allows over a million particles to be tracked on a normal PC. The choice of the routine parameters for an optimal performance will be discussed with the model of a spherical bunch.

PACS numbers: 02.60.Cb, 02.70.Bf, 03.50.De

1. Introduction

Numerical prediction of charged particle dynamics in accelerators is essential for the design and understanding of these machines. Applications such as colliders and SASE–FEL's demand very high quality electron bunches, where any anomaly severely degrades the final performance.

A powerful tool widely used for the study of the behaviour of charged beams is the tracking code GPT (General Particle Tracer) [2]. It calculates the trajectories of a large number of sample–particles through the combined external and self–induced fields generated by the charged particles (the so–called space–charge forces). Depending on charge density and energy, a direct point–to–point model can not be used to calculate space–charge forces because of granularity problems and the inherent $\mathcal{O}(N^2)$ scaling between the number of sample particles and CPU time [11].

In this paper we introduce a 3D model for the fast calculation of space–charge following the ideas in [10]. The space–charge fields are computed in the rest frame by a non–equidistant multigrid scheme. Hence, the numerical effort scales linearly with the number of particles in

terms of CPU time. The new model is well suited for a variety of applications, including the calculation of space–charge fields in a high–brightness photo–injector which are presented in [12].

In this paper we discuss the influence of various parameters on the efficiency of the 3D space–charge routine using the model of a spherical bunch with uniform or Gaussian particle distribution. Related numerical results are given in section 3. For further simulation tests regarding the calculation of important beam parameters such as emittance we refer to [8, 12].

2. The 3D space charge model

The particle tracking is performed by solving the relativistic equations of motion for a set of macro particles (sample particles) representing the distribution of the particles of a bunch. In the GPT code a 5th order embedded Runge–Kutta scheme with adaptive step size control is implemented for the numerical integration of these equations [11]. In each time step of the numerical integration the space–charge fields have to be taken into account. The space–charge calculation with the 3D model is performed as follows:

(i) Laboratory frame → rest frame: Transformation of the bunch from the laboratory frame to a rest frame with an average velocity by Lorentz transformation.

(ii) Rest frame: Determination of a non–equidistant 3D Cartesian grid in correspondence to the charge density of the bunch (see subsection 2.1).

(iii) Rest frame: Approximation of the charge density at the grid points.

(iv) Rest frame: Calculation of the electrostatic potential at the grid points via Poisson's equation applying a multigrid algorithm. The finite difference scheme (7–point stencil) is used for the discretization of Poisson's equation (see subsection 2.2).

(v) Rest frame: Derivation of the electric field and trilinear interpolation of the field values to the particle positions.

(vi) Rest frame → laboratory frame: Transformation of the field to the laboratory frame by Lorentz transformation.

The efficiency and accuracy of the space–charge calculation mainly depends on the determination of the 3D mesh and the applied multigrid scheme to solve Poisson's equation. Both we describe in the next two subsections.

2.1. The generation of the mesh

The electromagnetic potential is calculated on a 3D Cartesian mesh with an approximation of the charge density at the grid points in the rest frame. The 3D mesh is generated in a box around the bunch. To reduce the number of mesh lines needed, and thus to reduce CPU time, the density of the mesh lines is increased only if the charge–density increases. The actual positioning of the mesh lines is an iterative process. The mesh lines are distributed such that they are spaced according to the distribution of the beam charge density.

Strictly following this rule the resulting mesh spacing can lead to conflicts with the Poisson solver (see subsection 2.2). That is in such cases the multigrid Poisson solver converges only slowly or does not converge at all. Thus the parameter fn is introduced to maintain a maximum difference in spacing between neighboring mesh lines, to avoid the creation of a non–optimal mesh line distribution for the Poisson solver. If, e. g. $fn = 0.25$, then the difference in spacing between neighboring mesh lines can not vary by more than 25%. The effect of fn is shown in Fig. 1. When $fn = 0$, the spacing between all neighboring mesh lines is allowed to vary by 0%, creating an equidistant mesh. Such a mesh is most stable for the multigrid Poisson solver, but it will create many empty mesh boxes. On the other side, setting $fn = 0.2$ results in a dense sampling of the electron bunch and sparse sampling of the surrounding area.
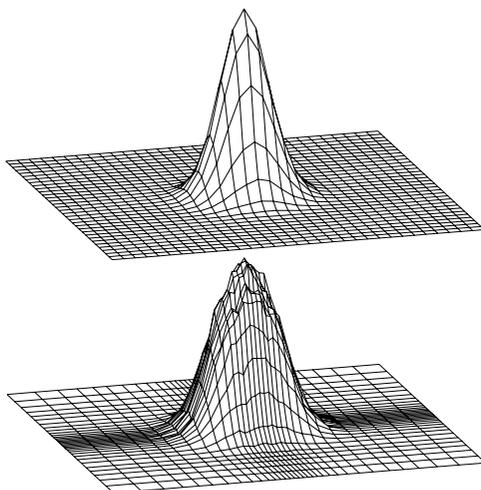
Figure 1. Mesh line positions ((x, y)–plane) for a Gaussian charge density with $fn = 0$ (top) and $fn = 0.2$ (bottom). The vertical axis shows the total charge in each mesh box, where the height of the top has been normalized in both plots.

2.2. The multigrid Poisson solver

The space–charge forces will be calculated in the rest frame of the bunch by means of Poisson's equation given by

$$-\Delta \varphi \;\; = \;\; \frac{\varrho}{\varepsilon_0} \qquad \text{in } \Omega \subset \mathbb{R}^3.$$

Here, φ denotes the potential, ϱ the charge density and ε_0 the dielectric constant. The domain Ω is a box around the particle bunch. On the boundary we consider three cases: a beam pipe, a cathode and free space. The beam pipe has ideal conducting walls transversely, what results in Dirichlet boundary conditions, and open boundary conditions longitudinally. The cathode has the same assumptions as the beam pipe accept the cathode surface, here Dirichlet boundary

conditions are assumed. The calculation in free space leads to open boundary conditions and can be used when the bounding box is located far enough from the boundaries of the structure. While a bounding cube is necessary with Dirichlet boundary conditions (see [8]) the open boundary conditions allow to put a bounding box relatively close around the bunch. Thus, a lot of computing time can be saved especially for very short or very long bunches.

Poisson's equation is discretized by finite differences on the non–equidistant mesh described in the previous subsection. An approximation of the charge density $\frac{\varrho}{\varepsilon_0}$ is computed on the mesh points from the charge of the macro particles.

The solution of the resulting system of equations requires a fast and robust solver. State-of-the-art is the application of a multigrid method as Poisson solver. In model cases the numerical effort scales with the number of mesh points. Here, we give only the general idea of a geometrical multigrid algorithm. Details can be found in [3, 1]. The multigrid algorithm operates on a certain number of grids starting with the mesh given by the discretization of Poisson's equation. This mesh is referred to as the fine grid or the fine level. Then a sequence of coarser grids is generated by removing mesh lines. On an equidistant mesh every second mesh line is removed. Now iteratively, a raw approximation of the solution of the systems of equations is obtained by the application of a few steps of a relaxation scheme (e. g. Gauss–Seidel iteration) which is called pre–smoothing. This approximation is then improved by a correction vector obtained on the coarser grids (the so–called coarse grid correction) where restriction and interpolation work as grid transfer operators. After applying interpolation another few steps of relaxation are necessary (post–smoothing). For the space charge calculations a multigrid V–cycle is realized. This scheme goes strictly down from the fine to the coarsest grid and then up again to the fine level.

As shown in [6, 7] the coarsening strategy is crucial for the convergence of the multigrid algorithm on non–equidistant grids. The generation of coarse grids with every second grid line removed as suggested in [1] is not reasonable with the dicretizations for bunches. It would lead to coarser grids with increasing aspect ratio of the mesh spacing. Hence the convergence of a multigrid scheme on such grids would considerably slow down. Here, the removal of mesh lines follows the rule: Two neighboring steps h_1 and h_2 remain also in the next coarser grid as long as either $h_1 \geq sh_{min}$ or $h_2 \geq sh_{min}$, where h_{min} denotes the overall minimal step size of the corresponding fine level. The factor s is chosen as $s = 1.6$ or $s = 1.7$ with the objective to obtain a decreasing aspect ratio of the mesh spacing.

Furthermore, the choice of the multigrid parameters such as the number of pre- and post–smoothing steps, the application of full or half restriction considerably influence the performance of the multigrid scheme. If the convergence of the multigrid algorithm turns out to be not sufficient (e. g. if the coarsening does not come out with decreasing aspect ratio on all levels), multigrid can be applied as a preconditioner for the conjugate gradient algorithm. This method leads to a better convergence at least in cases where a plain multigrid scheme converge too slow [5, 4].

3. Numerical test cases

The new 3D space–charge routine has been tested with a sphere filled with electrons with both uniform and Gaussian distribution. The space–charge forces has been computed for only one time step. For numerical tests with other bunch shapes and tracking examples we refer to [8, 12]. For an optimal performance of the space–charge routine several questions have to be considered:

(i) What is an optimal distribution of mesh lines both for the approximation of the particle distribution of the bunch and for the multigrid performance? (Choice of parameter fn)

(ii) Which number of mesh lines is optimal? (As few as necessary.)

(iii) How does the choice of multigrid parameters influence the performance of the algorithm?

Three possible versions of the multigrid scheme has been investigated. Two of these schemes have been performed with the following components: the Gauss–Seidel red–black iteration has been taken as smoothing operator with 2 pre– and 2 post–smoothing steps MG(2,2), full restriction has been tested versus half restriction. The third algorithm has been the application of multigrid as preconditioner for the conjugate gradient method (MG-PCG). It requires that the multigrid scheme is a symmetric and positive operator [4, 5]. Thus the components have been chosen as follows: two pre–smoothing steps with red–black Gauss–Seidel relaxation, two post–smoothing steps with black–red Gauss-Seidel relaxation and full restriction. Two V–cycles have been performed per CG–iteration step (MG–PCG(2,2)(2)).

3.1. Sphere with uniform particle distribution
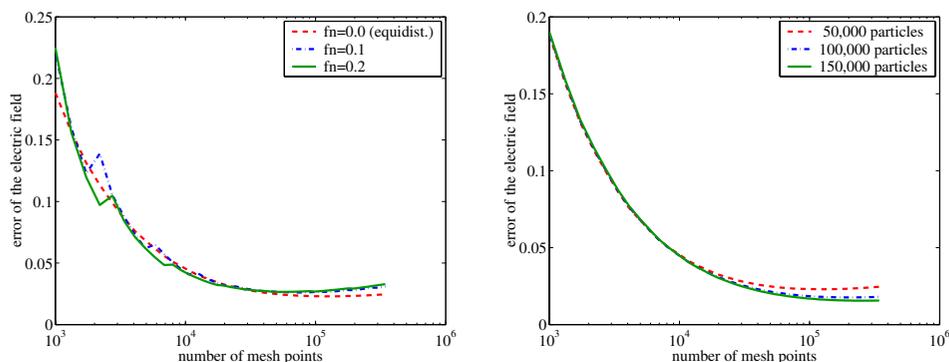
Figure 2. Error of the electric field for uniformly distributed particles in a sphere: Comparison of different meshes depending on parameter fn performed with 50,000 particles (left), comparison of different numbers of particles on an equidistant mesh with $fn = 0.0$.

For the investigation of the first two questions we tested the model of uniformly distributed particles in a sphere with the known analytical electrical field of a uniformly charged sphere. The error for the electrical field has been measured after interpolation at the
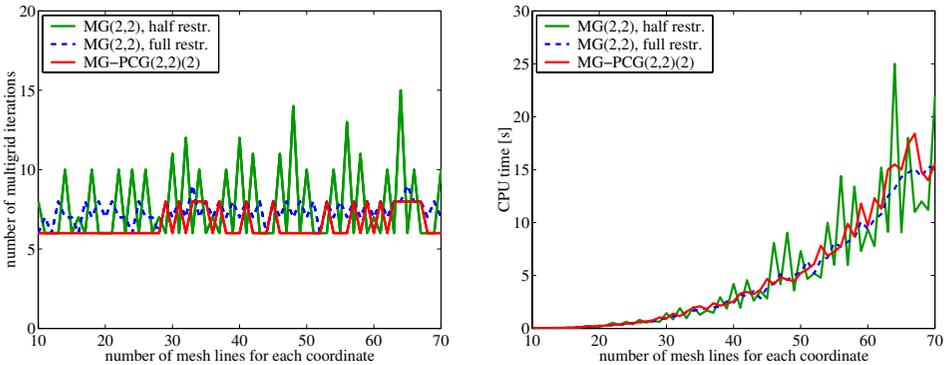
Figure 3. Influence of the choice of multigrid parameters to the performance of the space–charge routine (MG: multigrid, MG–PCG: multigrid preconditioned conjugate gradient): the number of multigrid iterations until the residual is less than 10^{-6} in the maximum norm (left), the related CPU times measured on a 800 MHz Pentium PC (right). The calculations have been performed on an equidistant grid with $fn = 0.0$ with a spherical bunch of 50,000 particles, where the particles have a uniform distribution.

position of the particles. Fig. 2 shows the best convergence for the equidistant grid ($fn = 0.0$) what is not surprising for a uniform particle distribution. Since the multigrid scheme has the best convergence and stability on equidistant meshes, these meshes should be preferred in the case of uniformly distributed particles. Furthermore, it can be concluded from Fig. 2 that neither a large number of mesh points nor a large number of particles improve the final result. Consequently, for a high efficiency of the 3D space–charge routine no more mesh lines or particles than reasonable should be taken for simulations.

Fig. 3 shows the performance of the above introduced 3 multigrid algorithms. As expected the multigrid preconditioned conjugate gradient method has the most stable performance. The multigrid scheme with half restriction as grid transfer operator turns out to be very sensitive to non–optimal meshes. In this case these are meshes with an even number of mesh lines for each coordinate direction. Here, the coarsening can not generate good results concerning the mesh spacing (see subsection 2.2).

3.2. Sphere with Gaussian particle distribution

Tests with the model of a Gaussian particle distribution in a sphere show the advantage of the application of non–equidistant meshes. Fig. 4 shows that meshes with larger aspect ratios in mesh spacing, i. e. larger fn result in a better convergence of the error of the electric field. It can be concluded that meshes with $fn > 0.2$ would not improve the results for the field error. Similar as for the uniformly distributed particles an increasing number of particles (greater than 100,000) in the sphere leads to no smaller field errors.

The multigrid schemes have been performed on a non-equidistant mesh with $fn = 0.2$ (see Fig. 5). Again the multigrid preconditioned conjugate gradient method turns out to be
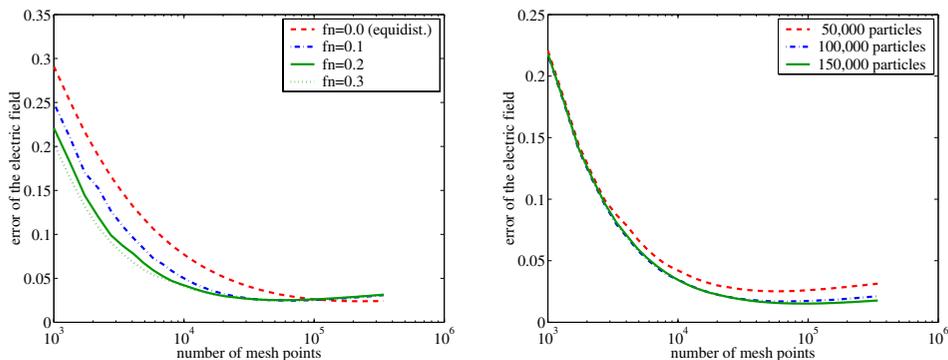
Figure 4. Error of the electric field for Gaussian distributed particles in a sphere: Comparison of different meshes depending on parameter fn performed with 50,000 particles (left), comparison of different number of particles on a non–equidistant mesh with $fn = 0.2$.
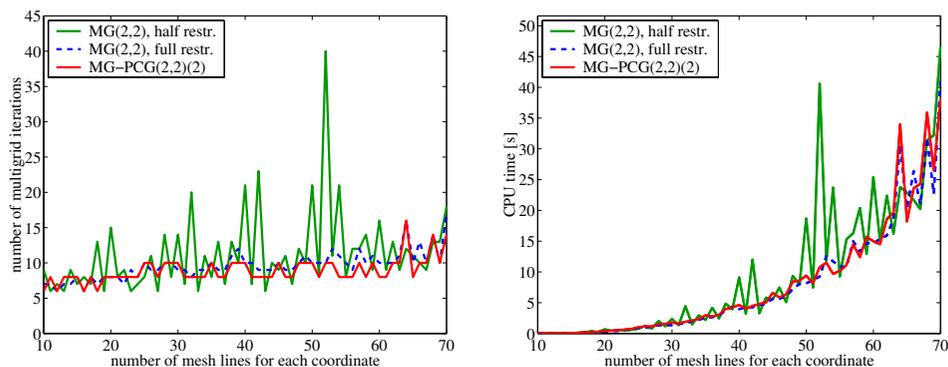
Figure 5. Influence of the choice of multigrid parameters to the performance of the space–charge routine: the number of multigrid iterations until the residual is less than 10^{-6} in the maximum norm (left), the related CPU times measured on a 800 MHz Pentium PC (right). The calculations have been performed with a spherical bunch of 50,000 particles on a non–equidistant grid with $fn = 0.2$, where the particles have a Gaussian distribution.

the most stable algorithm. Multigrid performed with half restriction is as in the previous test case very sensitiv to the distribution of mesh lines.

4. Conclusion

A new 3D space-charge routine implemented in the GPT code has been described in this paper. The new method allowing 3D simulations with a large number of particles on a common PC is based on a multigrid Poisson solver for the calculation of the electrostatic potential in the rest frame.

Various parameters of the routine have been tested for model bunches in order to find out

a stable and fast performance of the space-charge routine. Various applications like colliders or SASE-FEL's require very high quality bunches. Related to such applications we refer to [12], where the space-charge effects of very short and very long bunches with the new 3D space-charge routine in GPT are studied.

References

[1] Briggs W L, van Emden H and McCormick S 2000 *A Multigrid Tutorial. 2nd edition* (SIAM, Philadelphia)

[2] General Particle Tracer (GPT), release 2.60, Pulsar Physics, De Bongerd 23, Soest, The Netherlands

[3] Hackbusch W 1985 *Multi-Grid Methods and Applications* (Springer, Berlin)

[4] Jung M and Langer U 1991, Applications of multilevel methods to practical problems *Surv. Math. Ind.* **1** 217–257

[5] Jung M, Langer U, Meyer A, Queck W and Schneider M 1989, Multigrid preconditioners and their applications. *Proc. of the 3rd GDR Multigrid Seminar (Biesenthal)* 11–52

[6] Pöplau G, van Rienen U, Staats J and Weiland T 2000, Fast Algorithms for the Tracking of Electron Beams *Proc. of EPAC 2000 (Vienna)* 1387–1389

[7] Pöplau G, van Rienen U 2002, Multigrid Solvers for Poisson's Equation in Computational Electromagnetics *Proc. of the 3rd Conf. on Scientific Computing in Electrical Engineering (SCEE–2000)* U van Rienen, D Hecht and M Günther, eds, LNSCE **18**, (Springer, Berlin) 169–176

[8] Pöplau G, van Rienen U, de Loos M J and van der Geer, S B 2002, A Fast 3D Multigrid Based Space-Charge Routine in the GPT Code *Proceedings of EPAC 2002 (Paris)* 1658–1668

[9] ——2002 Fast Calculation of Space Charge in Beam Line Tracking by Multigrid Techniques *submitted to the Proc. of the 4th Conf. on Scientific Computing in Electrical Engineering, SCEE–2002 (Eindhoven)*

[10] Staats J, Weiland T, Kostial S and Richter A 1999, Tracking of Electron Beams with Numerically Determined Space Charge Forces *Proc. of the 1999 Particle Accelerator Conf. PAC'99 (New York)* 2740–2742

[11] Van der Geer S B and de Loos M J 2001 *The General Particle Tracer Code. Design, implementation and application* PhD thesis

[12] Van der Geer S B, de Loos M J and Luiten O J 2002, 3D Space-charge model for GPT simulations of high-brightness electron bunches *submitted to this volume*

On the applicability of the thin dielectric layer model for wakefield calculation

Sebastian Ratschow, Thomas Weiland and Igor Zagorodnov

Technische Universitaet Darmstadt, Fachbereich Elektrotechnik, Theorie
Elektromagnetischer Felder, Schlossgartenstrasse 8, D-64289 Darmstadt, Germany

E-mail: [`ratschow, weiland, zagor`]`@temf.tu-darmstadt.de`

Abstract. Some years ago A. Novokhatski, M. Timm and T. Weiland developed the thin dielectric layer model (TDLM) to predict and characterize the impact of surface roughness wakefields. Recently two articles referring to the TDLM have appeared raising new questions. This paper is intended to answer these questions.

In the first article the TDLM is used to interpret a measurement carried out at the Brookhaven ATF. We show analytically and by simulations with a newly developed program that the TDLM cannot be applied to the geometry used in this experiment.

The second article investigates analytically a geometry very similar to the geometry of the TDLM but in a square beam pipe and with only two sides of the beam pipe corrugated. The amplitude of the wake function found in that article differs significantly from the predictions of the TDLM. We compare both predictions to computer simulations made with MAFIA-T3 and investigate which of the models fits better to the results.

1. Introduction

It is expected that wakefields generated by unavoidable small (typically $< 1\mu$m in radial direction) geometrical imperfections of the inner wall of beam pipes, so called surface roughness wakefields, will no longer be negligible in the design of future linear colliders and future X-FELs, whose bunches will be very short. The planned bunch length e.g. of the TESLA X-FEL is 25μm RMS ([1] p. 288).

The different scales ranging from roughness depths on a sub micrometer level to beam pipe lengths of several meters up to kilometers exlude numerical analysis the effect unless gross simplifications are made.

In view of this the so called thin dielectric layer model (TDLM) ([2, 3, 4, 5] etc.) was developed as one of several competing approaches to make surface roughness wakefields calculable.

The TDLM is based on the observation that under certain conditions surface roughness wakefields look similar to wakefields induced by a thin dielectric coating on the inner surface of an otherwise perfectly conducting beam pipe [2].

The waveguide modes of a dielectrically coated beam pipe with circular cross section are well understood. The dielectric coating slows down the phase velocity of the modes. At one particular frequency, depending on the properties of the layer, the phase velocity of the fundamental TM waveguide mode (We will only consider monopole modes and longitudinal wakes in this paper.) is equal to the speed of the bunches running through the beam pipe. This synchronous fundamental TM mode is dominantly driven by the beam when the dielectric coating is thin, as in that case the frequencies of all other synchronous modes are much higher.

The TDLM predicts such a monofrequent behaviour also for the surface roughness wakefield. The synchronous fundamental TM mode is then called the "rough tube mode".

Analytic derivations of the rough tube mode exist for the azimuthally symmetric case, where the corrugations are irises, [6] and for a rectangular beam pipe with two opposite sides roughened by irises [7]. Provided the vacuum wavelength of the synchronous mode is large compared to the average distance between the corrugations the longitudinal electric field within each gap between adjacent irises can be assumed to be independent of the longitudinal coordinate [7] or equivalently the layer containing the corrugations can be replaced by an anisotropic dielectric [6]. The rough tube mode is then found by mode matching. Both approaches are not new. The first one was used to calculate wave propagation in corrugated waveguides. The latter averaging of material properties has been used e.g. for the analysis of waves incident upon laminated cores of transformers (see references given in [6, 7]).

3D numerical simulations show, that also for geometries which are not azimuthally symmetric but consist e.g. of individual bumps, monofrequent wakefields are to be expected provided the conditions for the applicability of the TDLM are fulfilled [4].

2. Discussion of the ATF experiment from the TDLM point of view

Recently an experiment was carried out at the Brookhaven Accelerator Test Facility (ATF) to decide between different models for surface roughness wakefield calculation [8]. One of these models was the TDLM. In this section we will show that the TDLM is not applicable to the geometry used in that experiment.

2.1. Analytical estimate

According to reference [6] the wavenumber k_0 of the rough tube mode satisfies

$$k_0^2 = \frac{2\varepsilon_z}{R\delta},\tag{1}$$

where $\varepsilon_z = (d_1 + d_2)/d_1$ is the relative dielectric constant in longitudinal direction, d_1 is the average length of the gaps between the irises, d_2 is the average length of the irises, i.e. $d_1 + d_2$ is the average longitudinal period length, R is the radius of the beam pipe and δ is the height of the irises. We point out that this expression for the wavenumber differs from corresponding expressions used in all earlier publications on the TDLM ([2, 3, 4, 5] etc.).

The key point to the TDLM is, that the wavelength $\lambda_0 = 2\pi k_0^{-1}$ must be much larger than the average longitudinal period in order to allow the averaging of the material properties. That is, the TDLM is only applicable when

$$d_1 + d_2 \ll \lambda_0\tag{2}$$

is satisfied.

The beam pipes used for the experiment at the ATF have a radius of $R = 3$mm, a length of $L = 970$mm and are artificially roughened by about 3000 presumably drop shaped bumps [8]. Artificial roughening is necessary because the bunch lengths available for the experiment, ranging from 0.3mm to 1.2mm RMS, are too long to be affected in a measurable way by the naturally occuring surface roughness.

We shall assume that the protrusions are brick shaped for simplicity. The wavelength of the rough tube mode tends to be shorter for non brick shaped surface roughnesses [3]. The area of the tube surface is $A_{tube} = 2\pi RL = 18284$mm^2.
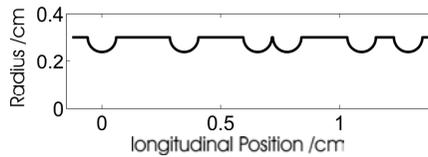
Figure 1. A small portion of the geometry with randomly spaced irises used for the simulation. The average distance between adjacent irises is 2.51mm.

In one representative beam pipe there are 2900 randomly distributed bumps. On average there is one bump per 6.305mm^2. It is not said, whether average transverse and longitudinal distance of the bumps are equal. We assume that they are. Then the average longitudinal distance between two bumps is $d_1 + d_2 = \sqrt{6.305\text{mm}^2} = 2.51\text{mm}$.

The width of the protrusions is $w = 1.2$mm and their height is $h = 0.6$mm [8]. With $\delta = h$ and $d_1 = d_1 + d_2 - w$ the wavelength of the rough tube mode can be estimated to be $\lambda_0 \approx 4.31$mm.

Obviously condition (2) is not fulfilled. This means that the TDLM is not applicable to the geometries studied in that experiment and that no conclusions can be drawn from this experiment on whether the TDLM gives a reasonable prediction of the rough tube wake potentials.

2.2. Simulations

Numerical simulations using a new conformal 2D code for geometries of revolution [9] also prove the non applicability of the TDLM.

Figure 1 shows part of the geometry with random iris spacing that was used for the simulation. The total length of the model was 1m. Height and base width of the protrusions are those given in reference [8]. The average distance between adjacent irises was taken to be 2.51mm. We also calculated the same geometry with equidistant iris spacing.

The wakepotentials of the two models, the one with random spacing and the one with equidistant spacing, are very different (see figures 2 and 3). Nevertheless the loss factors of both models are comparable namely 541V/pC in the equidistant and 499V/pc in the random distance case. These simulations are in contradiction to the experiment, where significantly less energy loss was measured in the random distance case.

The Fourier transform of the wakepotential of the equidistant model in figure 4 shows three different frequencies, two of them with about equal amplitudes. This contradicts the prediction of the TDLM and again shows that the TDLM is not applicable to these geometries.

2.3. Conclusion

We have shown in two independent ways, that the TDLM cannot be applied to the geometries used in the ATF experiment. Furthermore our simulations do not show the significant differences in energy loss reported in reference [8]. This point needs further clarification, as also the TDLM within the limits of its applicability does not distinguish between regular and random spacing of the protrusions.

Condition (2) is rather strict. In the present case the smallness of R makes λ_0 too short. In the other extreme the same happens when $\delta \to 0$ at constant period length.

292
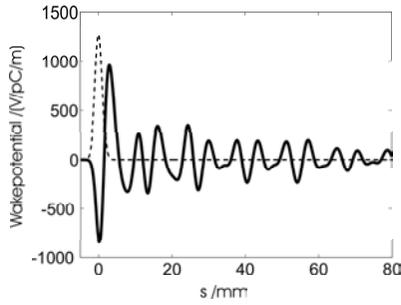
Figure 2. Wakepotential of the model with randomly spaced irises.
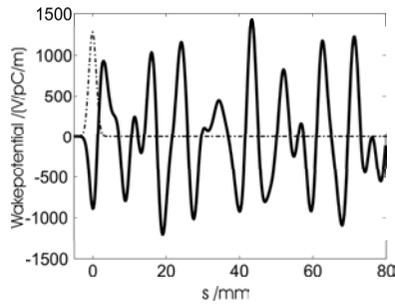
Figure 3. Wakepotential of the model with equidistant irises.
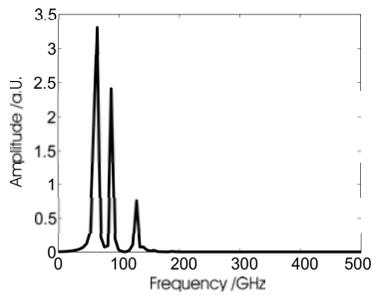
Figure 4. Fourier transform of the wakepotential shown in figure 3.

Specifically shallow and gradual protrusions cannot be treated within the framework of the TDLM.

3. Amplitude of the wake function

A derivation of the amplitude of the wake function, i.e. the wakepotential of a point charge, by calculating the coupling of the self field of the bunch to the synchronous mode, was not given in the context of the TDLM, because it was deduced from simulations that the amplitude agrees with that of the dielectric layer case [2].

In a recent paper [7] the amplitude of the wake function was derived using the reciprocity principle. To distinguish this model from the TDLM we will call it the "two side model", because the formula was derived for the situation where two opposite sides of a square beam pipe are corrugated. The amplitude of the two side model differs significantly from that of the TDLM.

As the derivations in both cases are rather involved, instead of trying to prove either formula, we will in this section calculate the expected wakepotentials for a Gaussian bunch in both models and compare them to MAFIA-T3 [10] simulations.

3.1. Wake functions

In all papers on the TDLM the wake function per unit length is given by

$$w(s) := w_0^{\parallel}(s) = \Theta(s)\hat{W}\cos(k_0 s) \tag{3}$$

with

$$\hat{W}_{\text{TDLM}} = \frac{Z_0 c}{\pi R^2}. \tag{4}$$

$s := ct - z$ is the longitudinal coordinate of the wake coordinate system, whose origin moves together with the bunch and which measures distances behind the bunch, $\Theta(s) = \begin{cases} 1 & \text{for } s > 0 \\ 0 & \text{for } s \leq 0 \end{cases}$ is the Heaviside step function, $Z_0 \approx 377\Omega$ is the impedance of free space and c is the velocity of light in vacuum.

The wake function per unit length of the two side model is structurally identical to that of the TDLM, but $\hat{W}$ is now ([7] p. 5)

$$\hat{W}_{\text{tsm}} = 8\pi \frac{Z_0 c}{ab} \frac{\delta}{a} \left\{ \coth\left(\frac{\pi}{2}\frac{b}{a}\right) \left[\frac{\sinh(\pi b/a)}{\pi b/a} - 1\right] \right\}^{-1}. \tag{5}$$

a and b are width and height of the open cross section of the beam pipe.

3.2. Wake potential of Gaussian bunches

The MAFIA simulations were carried out for a Gaussian bunch with line charge density

$$\rho(s) = \frac{q}{\sqrt{2\pi}\sigma_z} \exp\left(-\frac{s^2}{2\sigma_z^2}\right). \tag{6}$$

q is the total charge of the bunch and σ_z is the standard deviation. The wake potential of a bunch is the convolution of its charge distribution with the wake function normalized

Table 1. Parameters of the MAFIA Simulations.

Parameter	Symbol	Unit	Value
Length	L	mm	60
Open transverse cross section	πR^2	mm^2	2×2
Length of irises	d_2	μm	10
Height of irises	δ	μm	100/100/150
Period length	$d_1 + d_2$	μm	100
Number of corrugated faces	#		4/2/2

with respect to the charge,

$$W(s) = \frac{1}{q} \int_{-\infty}^{\infty} \rho(s - s') w(s') \mathrm{d}s' \tag{7}$$

$$= \frac{\hat{W}}{\sqrt{2\pi}\sigma_z} \int_0^{\infty} \exp\left(-\frac{(s - s')^2}{2\sigma_z^2}\right) \cos(k_0 s') \mathrm{d}s'. \tag{8}$$

This integral is easiest evaluated in the complex number space.

$$W(s) = \frac{\hat{W}}{\sqrt{2\pi}\sigma_z} Re\left[\int_0^{\infty} \exp\left(-\frac{(s - s')^2}{2\sigma_z^2} + \mathrm{i}k_0 s'\right) \mathrm{d}s'\right]. \tag{9}$$

$Re[\ldots]$ denotes the real part of the term in brackets.

Upon substituting $t := s'(\sqrt{2}\sigma_z)^{-1}$ and after some adding and substracting of terms within the exponential we find

$$W(s) = \frac{\hat{W}}{\sqrt{\pi}} \exp\left(-\frac{k_0^2 \sigma_z^2}{2}\right) Re\left[\exp(\mathrm{i}k_0 s) \int_0^{\infty} \exp\left(-(t - z_1)^2\right) \mathrm{d}t\right], \tag{10}$$

where we have set $z_1 = z_1(s) := (\sqrt{2})^{-1}(s\sigma_z^{-1} + \mathrm{i}k_0\sigma_z)$. Using the complex probability integral defined by ([11] p. 16f)

$$\Phi(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp(-t^2) \mathrm{d}t \tag{11}$$

we finally get

$$W(s) = \frac{\hat{W}}{2} \exp\left(-\frac{k_0^2 \sigma_z^2}{2}\right) Re\left[\exp(\mathrm{i}k_0 s)\left(1 - \Phi(-z_1)\right)\right]. \tag{12}$$

$\Phi(z)$ is proportional to the error function. For $s \to \infty$ quickly $\Phi(-z_1) \to -1$, so the cosine behaviour of the wake function is restored for large s. $\Phi(z)$ must be evaluated numerically.

3.3. Mafia simulations

The geometry amenable to MAFIA simulations is restricted to parameters which are not very near to what would be needed for real situations. Still these simulations can be compared to the model predictions. We simulated a rectangular beam pipe with small irises. The relevant parameters are given in table 1. Previous simulations have shown, that for longer beam pipes the wake potential does not change significantly from what is found for 6cm [5]. We report on three simulations distinguished by the number of corrugated sides and respective corrugation depth. The simulations with corrugations on two opposite faces were made for easier comparison with the two side model. For the
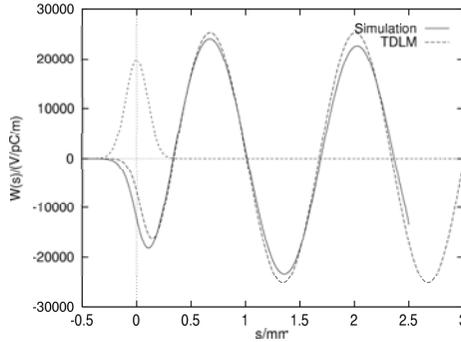
Figure 5. Longitudinal wake potential of a Gaussian bunch ($\sigma_z = 100\mu m$) in a sqare beam pipe of $2 \times 2mm^2$ open cross section + $100\mu m$ irises on all four sides. Shown are the result of a MAFIA simulation for a 6cm long beam pipe and a plot of equation (13).

same reason the length of the irises was chosen to be as small as possible. All surfaces are perfectly conducting. For the simulations we used a cubic grid with $10\mu m$ edge length. The beam was Gaussian with standard deviation $\sigma_z = 100\mu m$.

3.4. Comparison of the models with MAFIA simulations

First we compare the prediction of the TDLM to the simulation with irises on all four sides. After that we compare the prediction of the two side model to the other two simulations.

3.4.1. Comparison with the TDLM Using the Parameters of table 1 we can evaluate the factors of equation (12). In $\hat{W}_{\mathrm{TDLM}}$ we replace the transverse cross section of the round beam pipe πR^2 by the open transverse area of the rectangular pipe $4mm^2$ and find $\hat{W}_{\mathrm{TDLM}}/2 = 1.4138 \cdot 10^4 \mathrm{V/pC/m}$. ¿From equation (1) the wavenumber can be estimated to be $k_0 = 4.714 mm^{-1}$. Here we have used $R = 1mm$, because the part of the iris nearest to the beam will have the biggest influence. Next we find $\exp(-k_0^2\sigma_z^2/2) = \exp(-1/9) = 0.8948$. So for the simulated problem equation (12) specifies to

$$W(s) = 1.262 \cdot 10^4 \frac{\mathrm{V}}{\mathrm{pC\ m}} \cdot Re\left[\exp(ik_0 s)(1 - \Phi(-z_1))\right]. \tag{13}$$

In figure 5 the result of the simulation, the wake potential per 6cm, multiplied by 100/6, is compared to equation (13).

The slight differences in amplitude as well as in wavelength can be explained by the fact, that the simulation was carried out on a square beam pipe, whereas the theory is valid for a beam pipe with a circular cross sections. The overall agreement between both curves however is surprisingly good.

3.4.2. Comparison with the two side model We assume that the frequency is the same as that of the TDLM and calculate k_0 again from equation (1).
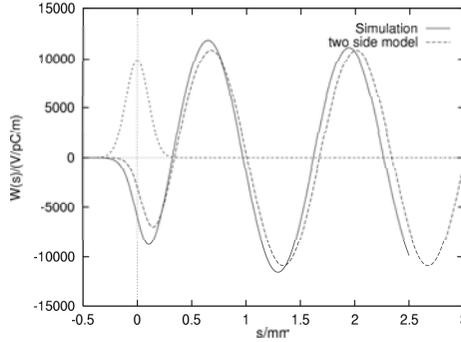
Figure 6. Longitudinal wake potential of a Gaussian bunch ($\sigma_z = 100\mu$m) in a sqare beam pipe of 2×2mm^2 open cross section $+ 100\mu$m irises on two opposite sides. Shown are the result of a MAFIA simulation for a 6cm long beam pipe and a plot of equation (15).

Comparison of equation (4) and equation (5) shows that both predictions differ by the factor

$$\frac{\hat{W}_{\text{tsm}}}{\hat{W}_{\text{TDLM}}} = 8\pi \left\{ \coth\left(\frac{\pi}{2}\frac{b}{a}\right) \left[\frac{\sinh(\pi b/a)}{\pi b/a} - 1\right] \right\}^{-1} \frac{\delta}{a} = 8.61356 \; \frac{\delta}{a}. \quad (14)$$

The second equality holds because in the present case $a = b = 2$mm. The linear dependence on δ would lead for small δ to a significant reduction of the wake potential.

We test this dependency on δ by calculating the geometry described in table 1 with two opposite sides roughened by irises once with $\delta = 100\mu$m and once with $\delta = 150\mu$m. The predicted wake potentials calculated from equation (12) are

$$W(s) = 5435\frac{\text{V}}{\text{pC m}} \cdot Re\left[\exp(ik_0 s)\left(1 - \Phi(-z_1)\right)\right] \quad \text{for } \delta = 100\mu\text{m} \quad (15)$$

and

$$W(s) = 8460\frac{\text{V}}{\text{pC m}} \cdot Re\left[\exp(ik_0 s)\left(1 - \Phi(-z_1)\right)\right] \quad \text{for } \delta = 150\mu\text{m} \quad (16)$$

respectively. Note that in the second case k_0 and the value of the expontential factor $\exp(-k_0^2\sigma_z^2/2)$ are different.

These wake potentials are compared to the respective simulations in figures 6 and 7. This time the geometry of the simulation exactly matches the one of the two side model (apart from the minor fact, that the irises are not infinitely thin). The wake potential in figure 6 is larger than the prediction of the two side model, whereas the amplitude in figure 7 is smaller than the two side model predicts. The wake potential in both simulations is almost equal and approximately half the size of the amplitude found in the simulation with all four sides corrugated. An amplitude independent of δ is predicted by the TDLM. The linear dependence on δ as predicted by the two side model cannot be found.

We conclude, that the TDLM describes these wakes, whereas the observed agreement of the two side model in figure 6 must be coincidential, because the linear dependence on δ predicted by that model is not observed.
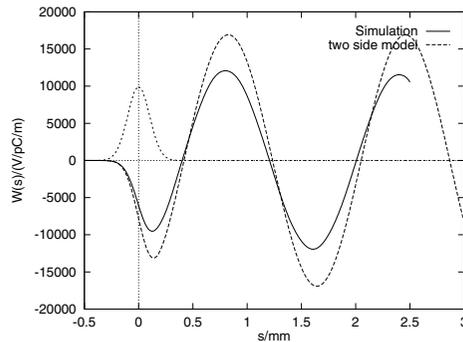
Figure 7. Longitudinal wake potential of a Gaussian bunch ($\sigma_z = 100\mu m$) in a sqare beam pipe of $2 \times 2\text{mm}^2$ open cross section + $150\mu m$ irises on two opposite sides. Shown are the result of a MAFIA simulation for a 6cm long beam pipe and a plot of equation (16).

3.5. Conclusion

Most parameters of the simulation were dictated by what is feasible within MAFIA. From the point of view of the models the parameter setting was arbitrary. The amplitude predicted by the TDLM does not contradict the simulations. Exact comparison is difficult in this case, because the theory was derived for round beam pipes. The linear dependence of the amplitude of the wake potential on δ predicted by the two side model is not observed in the simulations.

Acknowledgments

References

[1] TESLA Technical Design Report, Part II, DESY 2001-11, ECFA 2001-209, TESLA Report 2001-23, TESLA-FEL 2001-05.
[2] Novokhatski A and Mosnier A: Wakefields of short bunches in the canal covered with thin dielectric layer. *Proc. of the PAC 1997*, Vancouver, 1661–63
[3] Novokhatski A, Timm M and Weiland T: The surface roughness wake field effect. *Proc. of the ICAP 1998*, Monterrey.
[4] Novokhatski A, Timm M and Weiland T: A proposal for the surface roughness wake field measurement at the TESLA Test Facility. *Proc. of the PAC 1999*, New York, 2879–81
[5] Timm M 2000 *Wake Fields of Short Ultra-Relativistic Electron Bunches, Thesis.* Darmstaedter Dissertation D 17 (Osnabrueck: Der ander Verlag)
[6] Ratschow S, Weiland T and Timm M: On the mechanism of surface roughness wake field excitation. *Proc. of the PAC 2001*, Chicago, 167–69
[7] Mostacci A *et al* : Wakefields due to surface waves in a beam pipe with a periodic rough surface. PRST-AB, Vol. **5**, 044401 (2002)
[8] Zhou F *et al* : Surface roughness wakefield measurement at the Brookhaven accelerator test facility. *Proc. of the EPAC 2002*, Paris, 1598–1600
[9] Zagorodnov I and Weiland T: A Conformal Scheme for Wake Field Calculation. *Proc. of the EPAC 2002*, Paris, 1682–84

[10] MAFIA V4.107. CST Gmbh, Bad Nauheimer Strasse 19, D-64289 Darmstadt, Germany. `www.cst.de`

[11] Lebedev N N 1965 *Special Functions an their Applications* Prentice-Hall Inc. N.J.

COSY INFINITY's EXPO symplectic tracking for LHC

M.L. Shashikant†, Martin Berz† and Bela Erdélyi‡

†Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA
‡Fermi National Accelerator Laboratory, P. O. Box 500, Batavia, IL 60510, USA

E-mail: manikond@msu.edu, berz@msu.edu, erdelyi@fnal.gov

Abstract. The use of symplectified one-turn maps has been shown to decrease the CPU computational time when compared to the use of conventional symplectic integrators. In this paper we investigate the performance of COSY INFINITY's EXPO (The EXtended POincare generating function type) symplectic tracking tool by studying predictions for the dynamic aperture of the LHC lattice Version 6.0.

PACS numbers: 02.60.Cb, 29.27.Bd, 41.85.-p, 29.20.Lq

1. Introduction

The use of symplectic tracking for the study of the long-term behavior of hadron storage rings usually allows a more accurate prediction of dynamic apertures and stability. One way to preserve the symplectic structure is by the use of symplectic integrators. However, this method is usually computationally expensive, and cannot easily treat the dynamics in fields that vary along the reference orbit like in the fringe fields of particle optical elements. On the other hand, high order transfer maps for one turn or for individual subsection of the lattice allow a very accurate description of the dynamics[1], but in general do not directly lead to a symplectic representation of the dynamics.

Hence it is of interest to symplectify such one-turn maps, and several such symplectic schemes have been developed over the years. They can be classified into methods based on factorization in exactly integrable maps[2] on the one hand, and methods based on the generating functions on the other[1, 3]. The recent advances in the methods based on generating functions have provided us with the theory to not only symplectify the truncated maps but also to do this in an optimal way[4, 5]. In the next section a brief overview of the theoretical background and its implementation in EXPO tracking tool of COSY INFINITY[6] will be given. In section 3 we present the results of the study conducted on the realistic model of the LHC accelerator.

2. Theoretical background and implementation

The main step in tracking symplectically with the maps is the symplectification of the truncated Taylor map that is obtained with differential algebraic methods[1, 7, 8]. Various schemes have been developed for the purpose of symplectic one-turn-map tracking, many of which have not been tested for practical cases. Some of these are the generating function method, the Jolt factorization, the monomial factorization, the integrable-polynomial factorization, the fitted map methods, and the dynamic rescaling methods. However, the symplectified tracking obtained from the method depends on the specifics of the tool used.

2.1. Theory

In this paper we refrain from extensive details about the background behind the method and refer to other sources [9, 4]. One of the central results for the practical use of the method is the following theorem, which provides a connection between a symplectic map and a representation through a generalized generating function.

Theorem 1 *Let $\mathcal{M}$ be a symplectic map and let M represent its Jacobian. Then, for every point z there is a neighborhood of z such that $\mathcal{M}$ can be represented by a function F via the relation*

$$(\nabla F)^T = \left(\alpha_1 \circ \begin{pmatrix} \mathcal{M} \\ \mathcal{I} \end{pmatrix}\right) \circ \left(\alpha_2 \circ \begin{pmatrix} \mathcal{M} \\ \mathcal{I} \end{pmatrix}\right)^{-1}, \tag{1}$$

where α is a conformal symplectic map such that if

$$\alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} \text{ and } Jac(\alpha) = \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \text{ then} \tag{2}$$

$$\det(C(\mathcal{M}(z),z) \cdot Mz + D(\mathcal{M}(z),z)) \neq 0.$$

F is called the generating function of type α of $\mathcal{M}$, and denoted by $F_{\alpha,\mathcal{M}}$.

Conversely, let F be a twice continuously differentiable function with gradient $\mathcal{N}$, where $\mathcal{N} = Jac(\nabla F)^T$. Then, the map $\mathcal{M}$ defined by

$$M = (NC - A)^{-1}(B - ND)$$

is symplectic.

The theorem says that, once the generator type is fixed, locally there is a one-to-one correspondence between the symplectic map and a scalar function, which is unique up to an additive constant. Due to the fact that there exist infinitely many maps which satisfy the conformal symplectic condition, we conclude that for each symplectic map one can construct infinitely many generating function types.

For accelerator physics applications the maps of interest are usually weakly nonlinear around the equilibrium points. Hence, it is sufficient to constrain oneself to the equivalence class of types of generating functions associated with the subgroup of linear conformal symplectic maps. Further, the transformation properties of the generating function can be used to reduce the number of equivalence classes. It can be shown that every generator type belongs to an equivalence class $[S]$ associated with

$$\alpha = \begin{pmatrix} -JM^{-1} & J \\ \frac{1}{2}(I+JS)M^{-1} & \frac{1}{2}(I-JS) \end{pmatrix}$$

and represented by a symmetric matrix S. Thus, the determination of a generator merely requires the choice of a symmetric matrix S.

2.1.1. Optimal symplectification While the choice of S may be influenced by the map under consideration, there is a way to determine one particular matrix S, which when used for a large class of symplectic maps, on an average has optimal performance. Such a symplectic map satisfying the optimal condition has the following properties; (1) it works well for every particle in the given Poincaré section, (2) the outcome of the symplectification is independent of the specific Poincaré section used, (3) the symplectification works for any number of turns, (4) based on the previous three conditions the assessment of the optimality of the symplectification is unambiguous. The metric satisfying these conditions should also be coordinate independent. It can be shown that Hofer's metric satisfies all the above conditions, and it can be used to define distance between two arbitrary maps in the space of compactly supported Hamiltonian symplectomorphism $\text{Ham}^c(\mathbb{R}^{2n})$. Further, Hofer's metric can be used to show that optimal symplectification is achieved by class of generators $[S]$ obeying $S = 0$, and associated with

$$\alpha_{opt} = \begin{pmatrix} -JM^{-1} & J \\ \frac{1}{2}M^{-1} & \frac{1}{2}I \end{pmatrix}.$$

The details can be found in [4, 5].

2.1.2. Implementation The method starts with a given map $\mathcal{M}_n$ truncated at order n, and some initial conditions z. Utilizing (1) and α given by (2), the truncated α-generating function F_{n+1} is obtained. The arbitrary symmetric function S must be specified. For optimal symplectification, $S = 0$.

Table 1. The summary of the algorithm of the EXPO symplectic tracking tool.

Compute order n truncation of the map $\mathcal{M}$
$\downarrow$
By choosing $S = 0$, compute the generator to order $n + 1$
$\downarrow$
For each of the N turns
Push particle through the map
$\downarrow$
Using the generator compute exactly symplectic map by an iterative process
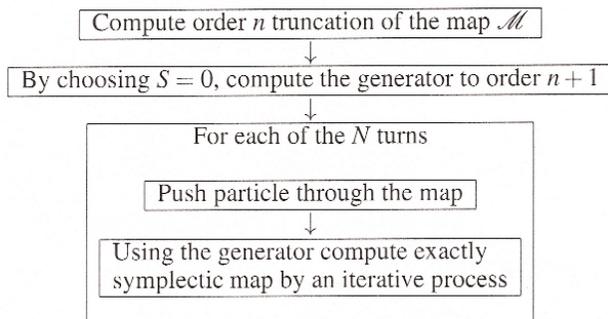
All necessary operations of map composition, map inversion, differentiation and integration are already available in COSY INFINITY[6, 10]. The flow chart in Table 1 summarizes the steps involved in the EXPO tracking tool. For practical purposes it is useful to perform the operations of pushing particles through the map and using the generator to compute exactly symplectic maps by treating groups of particles simultaneously. This reduces bookkeeping overhead and can be achieved with vector data type in COSY INFINITY[10].

3. Results and analysis

To study the performance of the EXPO symplectification tool, we use the lattice of the LHC based on the optics database, version 6 [15] (see Table 2 for the parameters), which
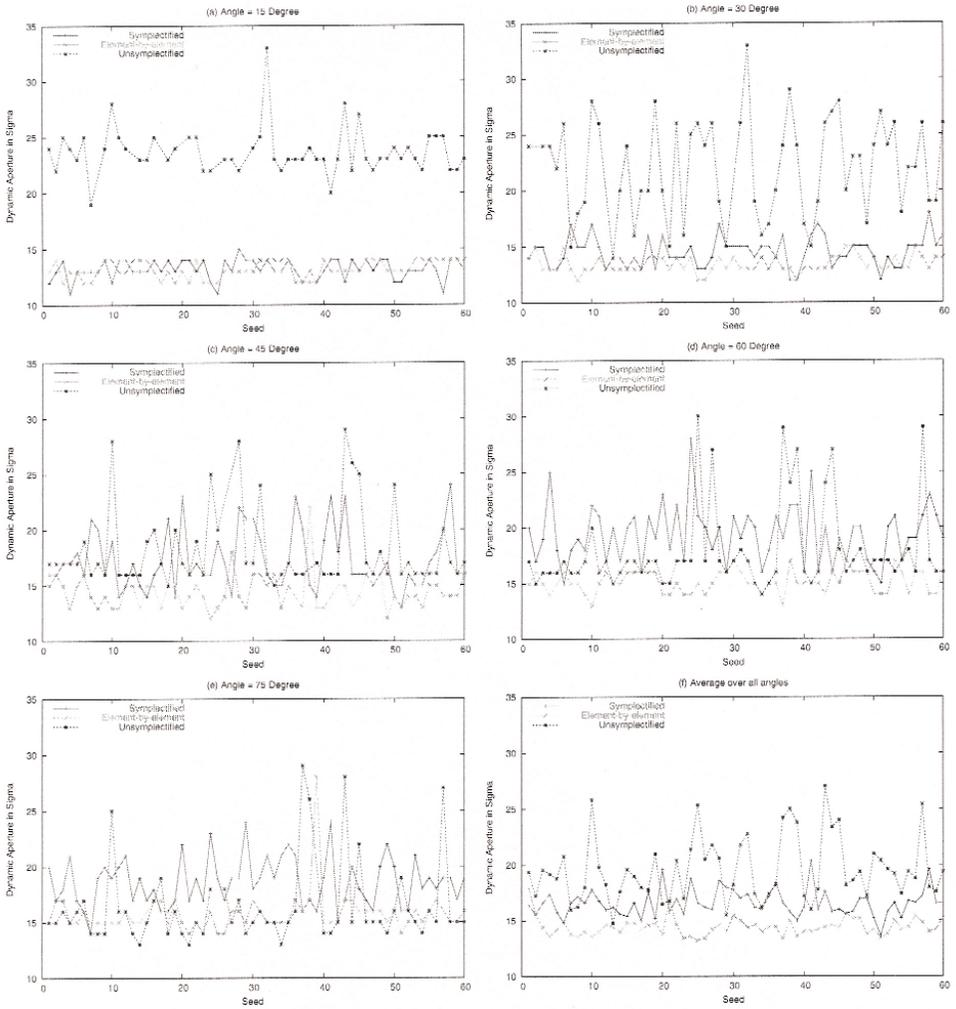
Figure 1. Dynamic aperture computed from tenth order tracking of realistic LHC lattice Version 6.0, with element-by-element, unsymplectified and EXPO symplectic tool. The particles were tracked for 10^5 turns, sixty error sets (seeds) and five different angles, (a)15°, (b)30°, (c) 45°, (d) 60° and (e) 75°. The picture (f) represents the average dynamic aperture over all angles.

comprises a design (ideal) lattice together with sixty different error sets representing the range of imperfections expected to exist in the real machine. The tenth order Taylor map for each error set, computed using the kick approximation-based SIXTRACK code [16] were obtained from CERN [17]. These Taylor maps are then used to compute the exactly symplectic Taylor map by using the EXPO symplectification tool of COSY INFINITY[6].

To study the dynamics, we first consider the difference in the dynamic aperture predicted by the element-by-element tracking, unsymplectified tracking and the EXPO symplectic tracking tool of COSY INFINITY[6]. We perform tracking using a tenth order map for the

Table 2. LHC Parameters (lattice version 6)

Relativistic gamma, γ	479.6
Beta at injection, β	18 m
Normalized emittance, ε_n	0.375×10^{-5} $m \cdot rad$
Beam Tube dimensions	$44mm \times 36mm$
Beam size, σ	0.375×10^{-3} m
Momentum deviation δ	7.5×10^{-4}

unsymplectified and EXPO symplectic methods, and predict the dynamic aperture for 60 error sets and five different launch angles, and the results are shown in Figure 1. As can be seen in Figure 1, the agreement between the optimal symplectic tracking and the element-by-element tracking decreases with an increase in angle.

Picture (f) in Figure 1 shows that symplectification significantly improves the accuracy of the prediction of the dynamic aperture. In fact, for some error sets and at certain angles the non-symplectic procedures predicts a rather large and unphysical dynamic aperture.

Since the beam consists of several particles launched at different angles and positions, it is useful to look at the average value of the dynamic aperture for all orders. Therefore, we next consider the difference in the dynamic aperture predicted by tenth order EXPO symplectic tracking as well as the average of tenth, sixth and fourth order EXPO symplectic tracking. The results are shown in Figure 2 for 60 error sets, 10^5 turns and five different launch angles. We also plot the results of element-by-element tracking for reference.

It can be observed that an increase in the order of calculation increases accuracy of the result; at order ten, the prediction of the dynamic aperture by the symplectification method approximately reaches the prediction of the element by element tracking.

3.1. CPU time

As mentioned in section 2.1.2, the first step in the process is obtaining the nth order DA map. These DA maps are obtained using the code SIXTRACK[16], which in turn uses the Differential Algebra package [13, 14] for performing the calculation. Using the data extracted from the Figure 1 of [2], the Figure 3 shows the execution time to obtain the DA map using SIXTRACK. As can be seen from Figure 3, considerable amount of time ($\approx 450\,sec$) is spent in generating the tenth order DA maps.

The second step is to find the generating function, which can then be used to obtain the exactly symplectic map after each turn. For a tenth order map it approximately takes 25 seconds to compute the generating function, which is only a small fraction of the time needed to compute the map.

Table 3. Time for tracking particles (a) and (b) for 10^5 turns.

	Particle (a)	Particle (b)
Radius	10σ	24σ
Angle = $\tan^{-1}(y/x)$	$45°$	$75°$
Average Iterations	9	42
Time for Tracking	$70\,min\,14\,sec$	$194\,min\,30\,sec$

The final step in the process is to track the particle for N turns. The time taken for this process is dictated by the average number of iterations needed to obtain the exactly symplectic

304
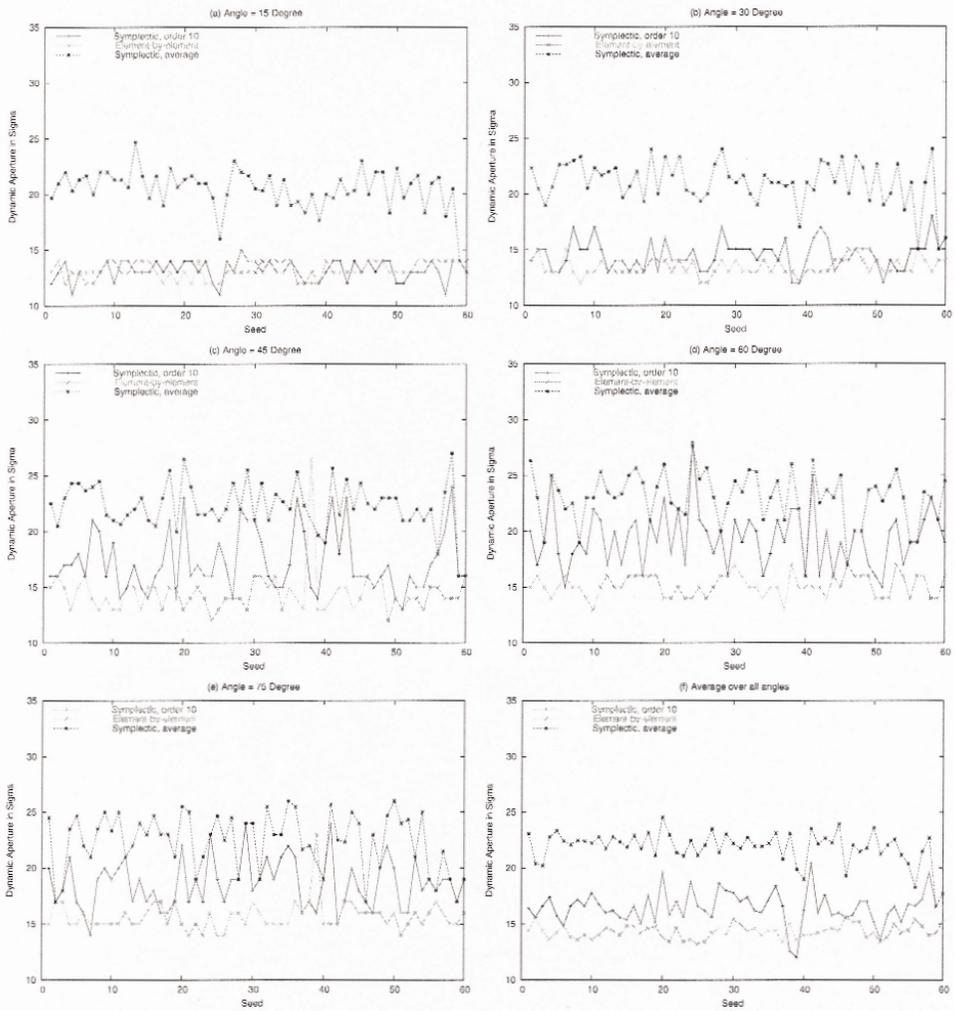
Figure 2. Dynamic aperture predicted by the element-by-element, tenth order EXPO symplectic and average of tenth, sixth and fourth order EXPO symplectic tracking, performed for 60 error sets, 10^5 turns and launch at angles, (a) 15°, (b) 30°, (c) 45°, (d) 60°, (e) 75°. The picture (f) represents the average dynamic aperture over all angles.

map after each turn, which in turn depends on the initial phase space coordinates of the particle. To understand this better we consider two cases (particles), (a) a particle launched within the dynamic aperture (close to design orbit), (b) a particle launched from outside the dynamic aperture. For both cases, Table 3 summarizes the parameters and time taken for tracking on a Pentium IV, 2 GHz, 512 MB Ram, Linux machine with COSY INFINITY[6, 10] compiled using the GNU Fortran compiler.

From our study we already know that the particle (a), launched at 10σ, is very stable. So it is expected to take the minimum time for tracking and the least number of average
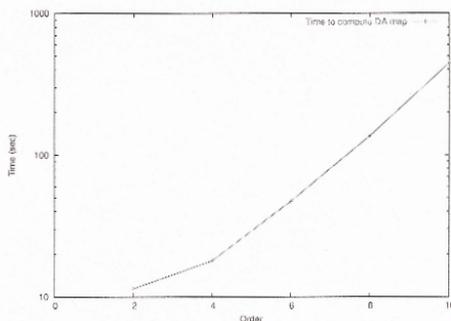
Figure 3. Execution time for creating a differential algebra map (DA map) for various orders.

iterations. Whereas the particle (b), launched at 24σ, is one of the worst cases possible. Tracking particle (b) for 10^5 turns will then give us the upper limit on the time taken for tracking any particle in a beam. By calculating the total time taken for tracking this particle, we conclude that the time taken for tracking using EXPO symplectic tool is of the order 10^4 sec. These computations were done in single-particle mode and not in the faster way of grouping a variety of particles together for one run, which is known to usually increase computational efficiency significantly.

For 10^5 turns the element by element tracking method takes time in the order of 10^5 sec. Hence, we gain at least a factor of 10 by performing the tracking using the EXPO symplectification tool of COSY INFINITY[6].

Acknowledgments

References

[1] M. Berz. *Modern Map Methods in Particle Beam Physics.* Academic Press, San Diego, 1999. Also available at http://bt.pa.msu.edu/pub.

[2] D. T. Abell, E. Macintosh and F. Schimdt. Fast Symplectic Map Tracking for the LHC. *LHC Project Report.* See also http://wwwinfo.cern.ch/pdp/pa/Ctrack/jolt/jolt.html.

[3] M. Berz. Symplectic tracking in circular accelerators with high order maps. In *Nonlinear Problems in Future Particle Accelerators*, page 288. World Scientific, 1991.

[4] B. Erdélyi. *Symplectic Approximation of Hamiltonian Flows and Accurate Simulation of Fringe Field Effects.* PhD thesis, Michigan State University, East Lansing, Michigan, USA, 2001.

[5] B. Erdélyi and M. Berz. Optimal symplectic approximation of Hamiltonian flows. *Physical Review Letters*, 87,11:114302, 2001.

[6] M. Berz and K. Makino. COSY INFINITY Version 8.1 - user's guide and reference manual. Technical Report MSUHEP-20704, Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, 2001. See also http://cosy.pa.msu.edu.

[7] M. Berz. Differential algebraic description of beam dynamics to very high orders. *Particle Accelerators*, 24:109, 1989.

[8] M. Berz. Arbitrary order description of arbitrary particle optical systems. *Nuclear Instruments and Methods*, A298:426, 1990.

[9] B. Erdelyi and M. Berz. Local theory and applications of extended generating functions. *Submitted*.

[10] M. Berz, J. Hoefkens, and K. Makino. COSY INFINITY Version 8.1 - programming manual. Technical Report MSUHEP-20703, Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, 2001. See also http://cosy.pa.msu.edu.

[11] M. Boge, A. Faus-Golfe, J. Gareyte, H. Grote, J.P. Koutchouck, J. Miles, Q. Qing, T. Risselada, F. Schmidt, S. Weisz, Overview of the LHC Dynamic Aperture Studies CERN, Geneva, Switzerland

[12] Y. T. Yan, P. J. Channell, M. Li and M. J. Syphers Long-Term Tracking with Symplectic Implicit One-Turn Maps PAC 1993 P38-40

[13] M. Berz The Differential Algebra Precompiler Version 3 Reference Manual, MSUCL-755, Michigan State University, East Lansing, MI 48824.

[14] LBL diffential algebra package and LieLib routines courtesy of . Forest.

[15] CERN LHC Optic site http://proj-lhc-optics-web.web.cern.ch/proj-lhc-optics-web/ .

[16] Home page of SIXTRACK http://frs.home.cern.ch/frs/sixtrack.html .

[17] LHC maps http://frs.home.cern.ch/frs/LHC_maps/ .

ORBIT: Parallel implementation of beam dynamics calculations

A. Shishlo, V. Danilov, J. Holmes, S. Cousineau, J. Galambos, S. Henderson

SNS-project, ORNL, Oak Ridge, TN, USA

Abstract. The parallel implementation of the ORBIT code is discussed. This implementation includes algorithms for two and three-dimensional (3DSC) space charge calculations, transverse and longitudinal impedances, a parallel diagnostics module to calculate beam characteristics, and a transverse feedback module. The 3DSC parallel algorithm, and its timing and scaling with problem size and number of processors are described in detail.

1. Introduction

Collective beam dynamics will play a major role in determining beam losses and intensity instability thresholds in high intensity rings such as the PSR at Los Alamos, the Booster at Fermilab, the AGS Booster, the Spallation Neutron Source (SNS), and future proton drivers. The details of these processes are complicated, and a good understanding of the underlying physics will require careful computer modeling. In order to study the dynamics of high intensity rings, a task essential to the SNS project [1], the macro-particle tracking computer code ORBIT [2,3] has been developed. The simulation of collective processes, including space charge and wake forces, requires three-dimensional modeling of the beam self and wall interactions. In many cases, the resulting simulations may require tracking millions of interacting particles for thousands of turns, which constitutes a legitimate high-performance computing problem. In order to meet the need for credible simulations of collective processes in high intensity rings, we are developing and implementing a parallel version of the ORBIT code.

The main goals of parallel computer simulations are to shorten the tracking time and to provide for the treatment of larger problems. There are two possible situations for tracking large numbers of particles with macro-particle tracking codes such as ORBIT. In the first case, particles are propagated through the accelerator structure independently without taking into account direct or indirect interactions among them, so there is no necessity for parallel programming. It is possible to run independent calculations using the same program with different macro-particles on different CPUs and to carry out the post-processing data analysis independently. In the case of interacting particles, there are collective processes, and we must provide communication between the CPUs where programs are running. Unfortunately, there is no universal efficient parallel algorithm that can provide communication for every type of collective process. The best parallel flow logic will be defined by the mathematical approach describing the particular process

and the ratio between computational and communication bandwidth. Therefore, our solutions for parallel algorithms cannot be optimal for every computational system.

Our implementation of parallel algorithms utilizes the Message-Passing Interface (MPI) library. The timing analysis has been carried out on the SNS Linux workstation cluster including eight dual Athlon XP 1600+ CPUs with each having 1024 MB RAM and the Gigabit Ethernet switch for communication. The communication library MPICH version 1.2.4, a portable implementation of MPI, has been installed under the Red Hat 7.2 Linux operating system.

All work was carried out under contract DE-AC05-00OR22725 for the U.S. Department of Energy. The research on the Spallation Neutron Source is managed by UT-Battelle, LLC.

2. ORBIT modules describing collective effects

At present, there are six ORBIT dynamics modules that describe interactions between macro-particles, and one diagnostics module that requires communication between CPUs in the case of a parallel run. There are two different transverse impedance modules, based on the wake field and Fourier decomposition, a longitudinal impedance module, two and three-dimensional space charge (2D and 3D SC) modules, and a fast feedback module. All of these modules can work with an arbitrary distribution of macro-particles between CPUs and use different parallel algorithms. The ORBIT's parallel implementation of the two-dimensional space charge module inherits the UAL 1.0 approach [4]. All others except the 3D SC module have very simple original algorithms based on summation of the resulting collective forces across all CPUs.

A typical example of a SNS ring simulation deals with the ring lattice with about one thousand nodes through which macro-particles are propagated in series. Half of these are 3D SC nodes, because it is more accurate to apply the space charge kick to the momentum of the macro-particles after each real lattice element. The lattice typically includes at most a few nodes related to the other collective interaction modules. Therefore, the efficiency of the parallel implementation of the whole code is defined basically by the effectiveness of the 3D SC module, and it will be at the focus of our attention.

3. Three-dimensional space charge model

The force in our three-dimensional space charge model is calculated as the derivative of a potential, both for longitudinal and transverse components. The potential is solved as a sequence of two-dimensional transverse problems, one for each fixed longitudinal coordinate. These separate solutions are tied together in the longitudinal direction by a conducting wall boundary condition on the inner surface of the beam pipe, thus resulting in a three-dimensional potential. This method, sometimes called a slice model, depends for its legitimacy, especially in the calculation of the longitudinal force, on the assumptions that the bunch length is much greater than the transverse beam pipe size and that the beam pipe shields out the forces from longitudinally distant particles. According to this approach we ignore some longitudinal effects, primarily longitudinal variation in the space charge fields on the distances less than longitudinal size of the slice. Although our model is applicable only to long bunches, and not to the spherical bunches that are the point of interest in many linac calculations, the three-dimensional space charge model adopted here is adequate to most calculations in rings.

The three-dimensional model implemented in ORBIT closely follows a method discussed by Hockney and Eastwood [5]. A three-dimensional rectangular grid, uniform in each direction, in the two transverse dimensions and in the longitudinal coordinate is

used. The actual charge distribution is approximated on the grid by distributing the particles over the grid points according to a second order algorithm, called "triangular shaped cloud (TSC)" in [5]. Then, the potential is calculated independently on each transverse grid slice, corresponding to fixed longitudinal coordinate value, as a solution of a two-dimensional Poisson's equation. The charge distribution is taken from the distribution procedure and, for the two-dimensional equation, is treated as a line charge distribution. The two-dimensional Poisson equation for the potential is then solved using fast Fourier transforms and a Green's function formulation with periodic boundary conditions [6]. The periodic boundary conditions are used only to obtain an interim solution, and this solution is then adjusted to obey the desired conducting wall boundary conditions. These are imposed on a specified circular, elliptical, or rectangular beam pipe through a least squares minimization of the difference on the surface of the beam pipe between the periodic Poisson equation solution and a superposed homogeneous solution. The homogeneous solution is represented as a series constructed from a complete set of Laplace equation solutions with variable coefficients, as described in [7]. In addition to accounting for image forces from the beam pipe, the boundary conditions serve to tie together the independently solved potentials from the various transverse slices, resulting in a self-consistent three-dimensional potential.

Finally, with the potentials determined over the three-dimensional grid, the forces on each macro-particle are obtained by differentiating the potential at the location of the macro-particle using a second order interpolation scheme. The resulting forces include both the transverse and longitudinal components. The interpolating function for the potential is the same TSC function used to distribute the charge. The detailed description of the three-dimensional space charge algorithm can be found in [8].

4. Parallel implementation of the 3D SC slice model

The approach to parallelization of the three-dimensional space charge algorithm is obvious. We distribute the two-dimensional space charge problems to different CPUs for solution. If the number of transverse slices is greater than the number of CPUs, then we must group the slices. To implement this scheme it is necessary to distribute the macro-particles among the CPUs before the solving two-dimensional problems. Then, after accomplishing the two-dimensional problems, we must provide for the exchange of neighboring transverse grids (with potentials) between CPUs to carry out the second order interpolation scheme in the longitudinal coordinate necessary for calculating and applying the space charge force kick to the macro-particles.

The 3D SC module includes four classes: 3D Space Charge class, the Boundary class, the Load Manager class, and the Macro-Particle Distributor class. The 3D Space Charge class is the main class in the module and uses all other classes. The Boundary class is just a 2D Poisson's equation solver and does not have any inter-CPU communication routines inside. The Load Manager defines how many transverse 2D slices will be assigned to each CPU. Initially, there is an even distribution of 2D slices among CPUs. The Macro-Particle Distributor distributes macro-particles among CPUs according their longitudinal positions and information received from the Load Manager class instance. This operation is the first operation of the 3D SC module.

4.1. The Macro-Particle Distributor class

The Macro-Particle Distributor class analyses the longitudinal coordinates of macro-particles currently residing on the local CPU, determines which macro-particles don't belong to this particular CPU, and sends them to the right CPU. This means that the class describing the macro-particle bunch should be a resizable container including 6D

coordinates of the macro-particle and an additional index indicating what CPU this macro-particle belongs to. This container should also have extra space to avoid changing the container size frequently.

The logic flow for the macro-particle distributor class is shown in Table 1. During the two first steps we define maximum and minimum longitudinal coordinates among all macro-particles in all CPUs. To eliminate the necessity of frequent changes in the longitudinal grid we add an additional 5% to each limit and save the result. During subsequent calls of the Macro-Particles Distributor class instance we don't change the longitudinal limits unless necessary.

After defining the longitudinal grid, we sort macro-particles according to the distance to the nearest grid point. Particles that no longer belong to the appropriate CPU are stored in an intermediate buffer together with additional information about where they belong. At the step 4 we define the exchange table $N_{ex}(i,j)$ where "i" is the index of the current CPU, "j" is the index of destination CPU, and the value is the number of macro-particles that should be sent from "i" to "j". After step 5 all CPUs know the number of macro-particles they will receive. The exchange table defines the parameters of the sending and receiving procedures used in step 7; therefore we avoid a deadlock. Finally, once all macro-particles are located in the correct CPUs, we can start to solve the two-dimensional space charge problems on all CPUs. The passage through the exchange table is organized in a special manner to provide communication between non-overlapping pairs of CPUs first. In this case the number of independent communication cycles will not exceed six for slow longitudinal motion. This approach guaranties that the time of communication during the step 7 will be the same for different numbers of CPUs.

Table 1. The flow logic of the Macro-Particle Distributor class. The "Communication" column indicates data exchanging between CPUs

N stage	Actions	Communication
1	Determine the extrema of longitudinal macro-particle coordinates	-
2	Find the global longitudinal limits throughout all CPUs	+
3	Get information from the Load Manager about number of slices assigned to this CPU	-
4	Analyse macro-particle longitudinal coordinates to determine on which CPU they belong. Store the 6D macro-particle coordinates to be exchanged in an intermediate buffer and mark these macro-particles as "dead". Define an exchange table $N_{ex}(i,j)$ (see text for the explanation)	-
5	Sum the exchange table throughout all CPUs by using the MPI_Allreduce MPI function with the MPI_SUM operation parameter	+
6	Check the free space in the bunch container and resize it if necessary	-
7	Distribute the 6D macro-particle coordinates in the intermediate buffer to the correct CPUs according the exchange table. Store the received coordinates in the bunch container in the available places	+

4.2.   The 3D Space Charge class

In the parallel version of the three-dimensional space charge algorithm, each CPU performs the same calculation of the potential on the transverse grids as in the non-parallel version. There is no need for communication between CPUs, because the macro-particles have already been distributed between CPUs by the Macro-Particles Distributor class instance and each CPU uses its own information to solve its own segment of the longitudinal grid. There is only one difference between parallel and non-parallel versions: in the parallel version there are two additional transverse slices beyond the ends of the CPU's own segment. Therefore the number of transverse slices for one CPU is $N_{slices}/N_{CPU}+2$ instead of N_{slices}/N_{CPU}, where N_{slices} is the total number of the transverse slices and N_{CPU} is the number of CPUs. The two additional slices are necessary because of the second order interpolation scheme. After the solution of the two-dimensional problem, the potential values from the two transverse grids on the ends of the segment are sent to the CPU that is the neighbor according to its index. In same fashion, the local CPU obtains the potential values from its neighbors and adds these potentials to its own. In this case the results of the parallel and non-parallel calculations are the same.

4.3.   The Load Manager class

The Load Manager class solves two tasks. First, it gathers timing information inside and outside the 3D SC module. Second, it redistributes the transverse slices between CPUs to provide the minimum time of execution of the whole code. It measures the wall clock time required to process unsynchronised parts of the code for each CPU and increases the number of slices for CPUs with smallest execution times. Thus it provides even loading of the cluster CPUs dynamically. Uneven loading can arise because of a non-uniform distribution of the macro-particles along the longitudinal coordinate or due to the presence of another parallel task on one or on several CPUs. The Load Manager tries to fix the situation and achieve the maximum parallel efficiency of the code.

5. Timing of the 3D SC module

Timings of the parallel algorithms were performed to elucidate the contributions of different stages to the total time of calculation and the parallel efficiency of their implementation. To avoid the effects of other jobs running on the same machine and other random factors, we performed the timings on the Linux cluster with no other users and computed the average time for a number of iterations. We used only eight CPUs of our cluster to eliminate competition of CPUs for network card resources on dual CPU nodes.

5.1.   The Macro-Particle Distributor class timing

The timing of the bunch distributor module was carried out without including additional MPI functions in the code of the module. We measured the time needed to distribute macro-particles between CPUs according to their longitudinal positions when we have N_{part} previously distributed and N_{rand} undistributed macro-particles.

Figure 1 shows the required time vs. N_{part} for 2 CPUs and $N_{rand} = 20000$ and 10000. As we expected, this time consists of three parts. The first part is the time of communication between all CPUs to sum and to spread the exchange table and to define global longitudinal limits through all of them. This is the execution time of the communication during steps 2 and 5 in the Table 1. The second part is proportional to the number of previously distributed particles. This is the time require for carrying out steps 1 and 4 in Table 1. The third part is proportional to the number of undistributed macro-particles that are distributed among CPUs during step 7. The step 6 is normally carried
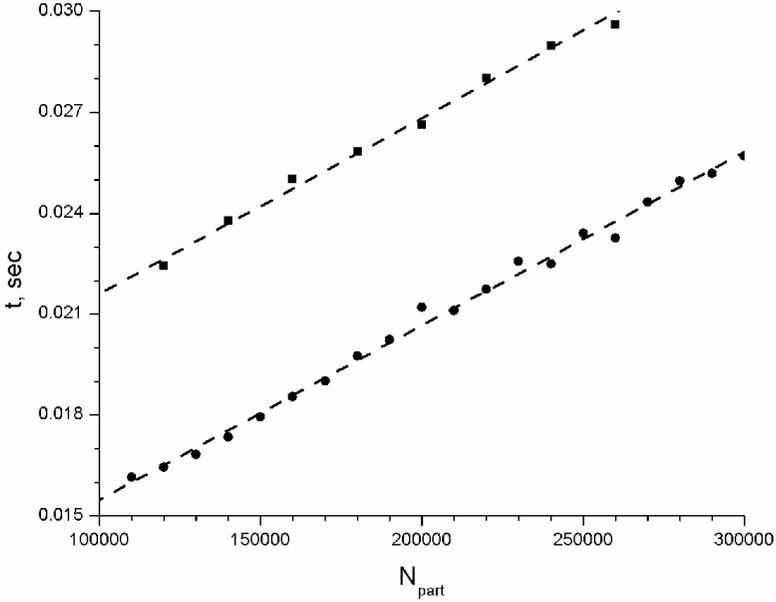
Figure 1. The time required by the Macro-Particles Distributor class to distribute N_{rand} between 2 CPUs in addition to N_{part} already distributed. The points are results of measurements, and the lines are linear approximations. The squares and circles denote $N_{rand} = 20000$ and 10000 macro-particles, respectively.

out only once. The total communication time of steps 2 and 5 in Table 1 depends on the number of CPUs and may vary from 0.5 ms to 4 ms for our case with $N_{CPU} < 9$. The analysis of graphs for several numbers of CPUs gives us the following three components approximation for the distribution time

$$t_{dist} = \tau_{comm}^{dist} + \tau_1 \cdot N_{part} / N_{CPU} + \tau_2 \cdot \alpha \cdot N_{part} \cdot (N_{CPU} - 1) / (N_{CPU} \cdot N_{CPU}) \qquad (1)$$

where the parameters τ_1 and τ_2 are equal to 1.0E-7 and 3.6E-6 sec, respectively. The parameter α in the equation (1) is the fraction of macro-particles that have to be distributed. In our simulations α is between 0 and 1E-4. Equation (1) demonstrates limited scalability of this parallel algorithm. For instance, if N_{part} = 300000 the distribution time will decrease with increasing number of CPUs up to 8 CPUs. It means that the parallel implementation of the Macro-Particle Distributor class has low parallel efficiency by itself, but we should consider the efficiency of the whole 3D SC module and ORBIT code.

5.2. The 3D Space Charge class timing

For timing the parallel implementation of the 3D Space Charge class, we used a procedure analogous to that described in the previous part of this report. The calculation times were measured as a function of the number of macro-particles, number of CPUs, and 3D grid size. Fitting the measurements, we obtained the following formula for the time of calculation with the (N_x x N_y) transverse grid size and N_z transverse slices

$$t_{3DSC} = \tau_3 \cdot N_{part} / N_{CPU} + \tau_4 \cdot (N_x \cdot N_y \cdot N_z) / N_{CPU} + \tau_{comm} \cdot (N_x \cdot N_y) \qquad (2)$$

where the parameters τ_3, τ_4, and τ_{comm} are 7.5E-7, 1.7E-7, and 7.9E-7 sec, respectively. The first term in the formula (2) describes the time spent binning the macro-particles, applying the space charge kick, etc. The second term is the time required to solve the set of two-dimensional space charge problems, and the last is the time for communication to exchange potential grids between CPUs and is proportional to the amount of exchanged data.

Equation (2) was obtained for a uniform distribution of macro-particles along the longitudinal axis. If the macro-particles are not distributed uniformly in the longitudinal direction, we should use the maximum number of macro-particles on one CPU instead of N_{part}/N_{CPU}.

The formula (2) does not predict the maximum useful number of CPUs, but the combination with the formula (1) gives the minimum time of calculation at N_{CPU} about 40 for $N_{part} = 300000$ and N_{CPU} more than 200 for one million macro-particles.

5.3. The parallel efficiency

Using Eqs. (1) and (2) we can define the parallel efficiency of the whole algorithm as follows

$$\eta = 100\% \cdot (t_{dist}(N_{CPU}=1) + t_{3DSC}(N_{CPU}=1))/(N_{CPU} \cdot (t_{dist} + t_{3DSC})) \qquad (3)$$

For the cases of 64x64x64 grid, 300000 macro-particles, and 2,4, and 8 CPUs we obtained 97, 92, and 85 %, respectively. These results are for a uniform distribution of the macro-particles along the longitudinal direction. Of course, in the real calculation scenario we are far away from this case. The figure 2 demonstrates the timing of a real scenario. It is a case with an accelerator lattice including 400 elements, and after each of them we perform 3D SC calculations. A 64x64x64 grid was used in the 3D SC module. The time is the function of the number of turns. At the beginning of every turn, 500
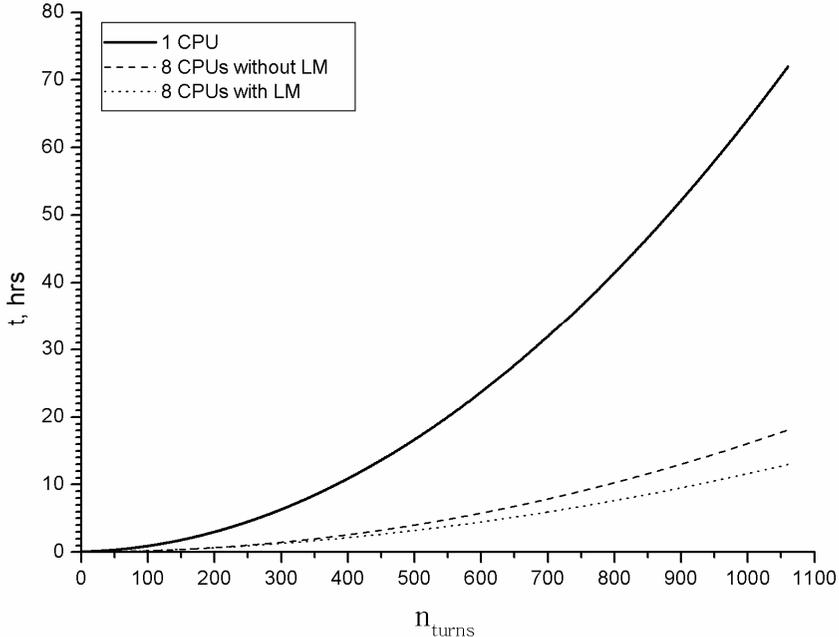
Figure 2. The timing for the real calculation scenario. See text for explanations.

macro-particles are injected into the lattice, so we have 500000 macro-particles at the end of the accumulation period. The final parallel efficiencies are 70% and 50% for calculation with and without the Load Manager, respectively. As one can see, using of the Load Manager improves the performance significantly, but we cannot perform detailed investigation of timing and parallel efficiency because of the dynamic nature of the Load Manager's actions.

In addition to the SNS Linux cluster, the ORBIT code was installed onto ORNL's IBM RS6000 "Eagle" supercomputer. The tests demonstrated parallel efficiencies 91% and 78% for 8 and 16 CPUs for the same scenario as above. Higher efficiencies for "Eagle" are defined by a better ratio between the network bandwidth and CPU performance.

6. Conclusions

Parallel algorithms of the ORBIT Code's modules describing collective effects and diagnostics have been developed and implemented. The 3D SC module, most critical for performance of the whole code, attains about 90% parallel efficiency for uniform longitudinal distributions of macro-particles and 70-80% for real case scenarios.

7. Acknowledgments

References

[1]   National Spallation Neutron Source Conceptual Design Report, 1997 NSNS/CDR-2/V 1, 2

[2]   Galambos J, Holmes J, Olsen D, Luccio A, and Beebe-Wang J, 1999 ORBIT Users Manual, http://www.sns.gov/APGroup/Codes/Codes.htm

[3]   Danilov V, Galambos J, and Holmes J, 2001 Proceedings of the 2001 Particle Accelerator Conference TPPH040 (Chicago); also Holmes J, Danilov V, Galambos J, Shishlo A, Wei J, Chou W, Michelotti L, Ostiguy F, and Cousineau C 2002 Proceedings of the 2002 European Particle Accelerator Conference THPLE022 (Paris)

[4]   Malitsky N and Shishlo A 2001 Proceedings of the 2001 Particle Accelerator Conference RPAH311 (Chicago)

[5]   Hockney RW and Eastwood JW 1988 Computer Simulation Using Particles (Bristol: Institute of Physics Publishing)

[6]   Holmes JA, Galambos JD, Jeon D, Olsen DK, Cobb JW, Blaskiewicz M, Luccio, AU and Beebe-Wang J 1998 Proceedings of International Computational Accelerator Physics Conference (Monterey, CA) p 179

[7]   Jones FW 2000 Proceedings of the 2000 European Particle Accelerator Conference (Vienna) p 1381

[8]   Holmes J and Danilov V "Beam dynamics with transverse impedances: a comparison of analytic and simulated calculations", SNS Technical Memo. 82, 2002

Vlasov simulation of beams

Eric Sonnendrücker† and Francis Filbet‡

IRMA,Université Louis Pasteur, 7 rue René Descartes, 67084 Strasbourg cedex, France

Abstract. In this paper we give an overview of numerical methods for beam simulation based on the direct resolution of the Vlasov equation on a grid of phase space and focus on recent developments of these methods, in particular their implementation on unstructured meshes and the first step towards a self adaptive method being able to adaptively refine the discretization mesh based on wavelet techniques.

1. Introduction

Numerical simulation has become a major tool for modeling and theoretical understanding of beam propagation as well as for accelerator design. Most simulations today are being performed using the Particle-In-Cell (PIC) approach. This method combines Lagrangian particle motion with grid based computation of the mean self fields, a deposition and interpolation step making the link between the two parts. The PIC method enables to perform cost effective simulations and becomes especially efficient compared to grid based Vlasov methods when dimensionality increases. The drawback of PIC methods is their inherent numerical noise which decreases only slowly when the number of particles is increased. Moreover, in the PIC method, the phase space is populated with particles according to the value of the distribution function, i.e. more particles are put in regions of phase space where the distribution function is larger. This means that low density regions of phase space are very sparsely populated with particles, which makes the resolution very poor in these areas. This aspect of PIC methods makes them inefficient for simulations where what happens in those regions is of major importance, like beam halo formation. Note that populating phase space uniformly with particles of different weights does not help and makes things even worse for highly non linear problems, as particles of different weights mix, making light weight particles useless.

For such problems it seems attractive to use grid based methods solving the Vlasov equation. They have the advantage of being completely devoid of numerical noise and of having the same resolution everywhere in phase space independently of the value of the distribution functions. Unfortunately, they are more expensive than PIC methods especially when dimension increases. But even so, computers are now powerful enough that these methods can be used for many relevant problems, and hence provide an essential complement to PIC solvers in one's beam simulation toolbox, as in addition to their other features, they provide an alternative to PIC methods which is useful for benchmarking.

The aim of this paper is to give an overview of current Vlasov solvers and pinpoint their strong points and limitations, as well as problems for which they should help. We shall also give some hints on ongoing and future research which will most likely broaden their range of applicability.

† sonnen@math.u-strasbg.fr
‡ Present address: MAPMO, Université d'Orléans, B.P. 6759, 45067 Orlans cedex 2, France

The paper is organized as follows. First, we shall recall the most widely used Vlasov solvers and their features. At this point we shall give some examples of simulations using these methods, after which we shall pinpoint their limitations. Then we shall explain how the methods can be extended to unstructured meshes. And finally, we shall introduce a novel adaptive Vlasov solver which is meant to overcome some of these limitations.

2. Vlasov solvers on uniform structured meshes

2.1. Description of the numerical methods

High intensity particle beams can be described by their distribution function $f(x, v, t)$ which satisfies the Vlasov equation

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + \frac{q}{m} \left(E + v \times B\right) \cdot \nabla_v f = 0, \tag{1}$$

which is generally coupled with Poisson's equations

$$-\Delta \phi = q \int f \, dv.$$

The electric field is the sum of an applied field and the self consistent field $-\nabla \phi$ and we assume that the magnetic field is only an applied field. In some cases of course the self magnetic field also matters and it might be necessary to use the full Maxwell equations.

The first Vlasov method we are going to describe is the backward semi-Lagrangian Method. Let us introduce the characteristics of the Vlasov equation (1), which are the particle trajectories and are solution of the following dynamical system

$$\frac{dX}{dt} = V, \tag{2}$$

$$\frac{dE}{dt} = \frac{q}{m} \left(E(X(t), t) + V(t) \times B(X(t), t)\right). \tag{3}$$

The distribution function f solution of (1) has the property that it is conserved along the characteristics, i.e. the particle trajectories, obtained by solving (2)-(3). This means that

$$f(t, x, v) = f(s, X(s; x, v, t), V(s; x, v, t)),$$

where we denote $(X(s; x, v, t), V(s; x, v, t))$ the unique solution of (2)-(3) at time s with initial conditions $X(t) = x, V(t) = v$ at time t, or in other words $(X(s; x, v, t), V(s; x, v, t))$ is the position in phase space at time s of the particle which was at position (x, v) at time t. It is this property of the Vlasov equation which builds the backbone of the semi-Lagrangian method. We can now write down the algorithm for this method: the distribution function f^n being known at time step n at the grid points, its value f^{n+1} at time step $n + 1$ is computed in the following way:

(i) Find the origin of the characteristics ending at the grid points. That is for each grid point (x_i, v_j) compute $(X(t_n; x_i, v_j, t_{n+1}), V(t_n; x_i, v_j, t_{n+1}))$, the position in phase space at time t_n of the particle that ends on the grid point (x_i, v_j) at time t_{n+1}.

(ii) Interpolate old value at origin of characteristics from known grid values.

A sketch of the principle of this method is given in Figure 1 on the left-hand side.

Remark 2.1 *The interpolation part introduces diffusion which can be limited by using high order interpolation methods, at least cubic in practice. However using more than first order interpolation, local minima and maxima are not preserved, this can in particular introduce negative values of the distribution function which are clearly unphysical. A remedy against this phenomena is to use an appropriate linear combination of high order and first order interpolation see [2, 3] for details.*

The first version of the semi-Lagrangian method that was used for the Vlasov equation is the split cubic spline solver of Cheng and Knorr [6]. This was generalized to more general Vlasov equations by Sonnendrücker et al. [10], and its properties compared to those of other solvers were studied [8]. The CIP method of Nakamura and Yabe [13] also belongs to this kind of methods. In this case a cubic Hermite interpolation is used, and the derivatives, necessary for the Hermite interpolation, are also advected using the split Vlasov equation.
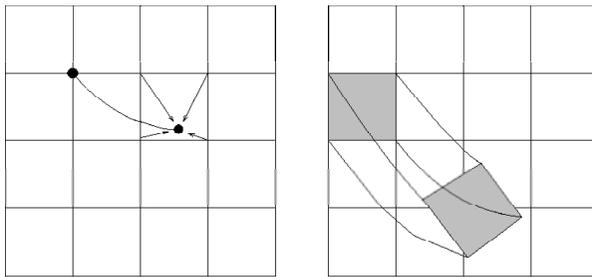
Figure 1. Sketch of the update method for the semi-Lagrangian case (left) and the finite volume case (right).

Let us now introduce what we call the Finite Volume method, as the unknown quantities are the averages of the distribution function over the cells (or the finite volumes) of the mesh $\frac{1}{V}\int_V f\,dx\,dv$. This method uses the property that $\int_V f\,dx\,dv$ is conserved along the characteristics solution of (2)-(3).

Given the discrete distribution function, i.e. its averages over the cells of the mesh known at time t_n. The algorithm for updating theses values consists of three steps:

(i) Reconstruct a high order polynomial (usually second or third order) whose averages over the cells match the given averages of the distribution function at time t_n.

(ii) Resolution: compute origin of cell downstream of the characteristics.

(iii) Compute average value over origin of cell.

A sketch of the principle of this method is given in Figure 1 on the right-hand side.

Remark 2.2 *As for the semi-Lagrangian method, one needs to use a high order reconstruction to control numerical diffusion, but this process introduces oscillations which destroy the conservation of local extrema and in particular the positivity of the distribution function. However, these local extrema can be preserved even when using high order interpolation by introducing slope limiters as described in [9].*

The first method of this kind used for the Vlasov method is the method of Boris and Book [5]. It was then further used and described by Fijalkow [7] and generalized to higher order in [9] where slope limiters to enforce positivity where also introduced. An overview of this method and comparison to others can be found in [8, 1].

2.2. Numerical simulation of a beam in a uniform focusing channel

In order to illustrate the possibilities offered by the methods we described in the previous section, we present here simulation results of a RMS matched semi-Gaussian beam in a continuous and periodic focusing channel. The simulation were performed using the VADOR code [11], using the finite volume method, and are representative for Vlasov simulations on a uniform phase-space grid.

The initial value of the distribution function for a semi-Gaussian beam is

$$f_0(x, y, v_x, v_y) = \frac{n_0}{(2\,\pi\,v_{th}^2)\,(\pi\,a^2)} e^{-\frac{v_x^2+v_y^2}{2\,v_{th}^2}}, \quad \text{if } x^2 + y^2 \leq a^2,$$

and $f_0(x, y, v_x, v_y) = 0$, if $x^2 + y^2 > a^2$. We consider a beam of potassium ions with current I=0.2 A, energy $80keV$ and radius a=0.02 m. The other parameters are computed so that the beam is RMS matched. The results are given in Figure 2. We observe in particular that the solution is completely devoid of numerical noise. In this case we get very good results and the code is fairly fast. Indeed eulerian methods based on a uniform mesh have some very positive features: first interpolation is very efficient on such a mesh and second they are easy to parallelize and show perfect scalability with the number of processors as the grid is fixed. In particular there is no load balance problem as is the case for PIC methods.
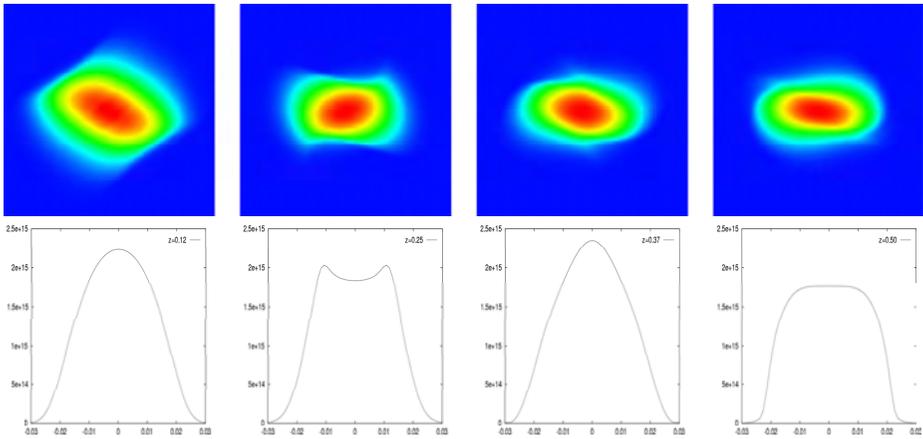
Figure 2. Time evolution of the x-v_x projection (top), slice of charge density (bottom)for a semi-Gaussian beam in a uniform focusing channel.

2.3. Numerical simulation of a beam in a periodic focusing channel

In this case we still consider a semi-Gaussian beam of potassium ions with $I = 0.1A$, energy $80keV$ and emittance $\epsilon = 0.005\pi\,m\,rad$ now in a periodic focusing channel of length $1m$ and focusing length $0.2m$, the focusing is done with a constant magnetic field of $2T$ such that $\omega_0 = 2.47 \times 10^6 s^{-1}$. An envelope code is used to match the beam. The initial beam radius is $R(0) = 0.1m$ and $R'(0) = 0$ as the beam is initialized in the center of the focusing section. The results are represented in Figure 3. Because of the important displacement of the beam during the time evolution, a large portion of phase space needs to be meshed. Hence unless
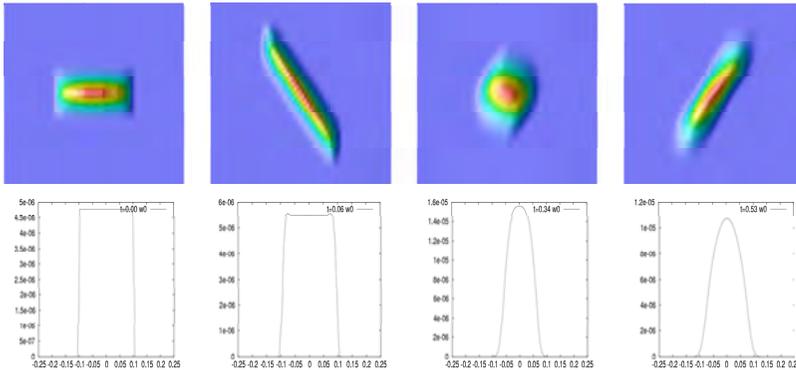
Figure 3. Time evolution of the $x - v_x$ projection of f (top), slices of charge density (bottom).

we use huge computational resources it is hard to get a fine enough meshing of the beam at all position which accounts for the ripples we can see on the x-v_x plots.

This simulation in a periodic focusing field already emphasizes one of the limitations of Vlasov methods using a uniform fixed grid. As we can see on figure 3, at any given time a big part of the phase space where the distribution function is computed corresponds to vanishing values. One easy way to improve this specific simulation would be to use the axisymmetric version of the VADOR code [12]. We have actually done it which yields faster and better results but the fundamental problem of meshing a lot of empty phase space parts remains. In order to overcome this limitation, we are currently investigating using different grids during the time evolution as well as adaptive mesh refinement.

3. The semi-Lagrangian method on unstructured grids

The semi-Lagrangian method can be generalized to an unstructured triangular finite element like mesh. The advantage is that we get more flexibility in where we need to put the grid points, allowing for large cells where the distribution function is smooth an finer cells where the variations are more important.

The method was implemented (see [3]) using operator splitting used between x and v advections so that both advections yield an explicit solution for the characteristics. In the case of an unstructured mesh, there is no line of x or v values, so that an interpolation in full phase space is necessary at each split step. The interpolation can be done using Finite Element basis functions. Many of those are available in the literature, which gives us a large choice. In practice, it occurs that high order Lagrange finite elements seem unstable due to oscillations on the edge of elements. The most interesting elements for the Vlasov equation seem to be Hermite type elements. A CIP type method can then be used to advance the derivatives. As for the case of structured grids positiveness and conservativeness can be ensured by linear combination of high order and first order interpolation. Another specificity of unstructured grids and additional cost factor is that the computation of $\rho(x)$ needs interpolations.

To sum up, we can say that the semi-Lagrangian method can be implemented on unstructured meshes. However, it has two major drawbacks at the moment: especially when there are relatively flat triangles which one sometimes gets with a mesh generator, it is more diffusive than on structured meshes; moreover the additional complexity of handling the unstructured mesh makes it hard to code and computationaly a lot less efficient.
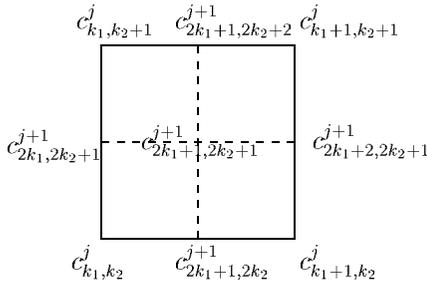
Figure 4. Mesh refinement in 2D.

4. An adaptive semi-Lagrangian method

The unstructured mesh enables to choose where one wants to put the grid points, but does not allow for evolution of the grid points in time, as seems to be needed for periodic focusing.

Our aim now is to optimize the number of grid points for a given numerical error and this for every time step. Following ideas from fluid dynamics, we found that multi resolution techniques using interpolating wavelets are well suited to determine where refinement is needed [4].

The principle of the method is to use several different levels of meshes. Starting from the coarsest level, the finer levels are obtained by adding the midpoints in each direction as shown in Figure 4. At one given level, the grid-function defined on this level is decomposed into the grid-function at the coarser level plus the missing piece. More precisely, denote by G_j the grid at level j, the grid becoming finer when j is increased. Then a more precise sample, i.e. the values of f at grid points of G_{j+1} (denoted by c_{i+1}) can be decomposed into a smaller sample i.e. values of f at grid points of G_j (denoted by c_i) plus the details (denoted by d_i). The details contain the difference between the exact value and the value predicted using an interpolation operator. Hence we have

$$c_{2k}^{j+1} = c_k^j \quad \text{same value at coarse mesh points}$$

$$d_k^j = c_{2k+1}^{j+1} - P_{2N+1}(x_{2k+1}^{j+1}),$$

where P is an interpolation operator, say for example a cubic Lagrange interpolation operator which we used in practice. This decomposition is a special kind of wavelet decomposition.

Using this idea, the semi-Lagrangian method can be easily adapted to this setting. Indeed the semi-Lagrangian method is based on polynomial interpolation, and the main idea for adaptivity is that details are small where the polynomial interpolation does a good job. So in the adaptive method, we use a wavelet decomposition to eliminate grid points corresponding to small details. There is no loss of information due to the wavelet decomposition. We only loose information once we have discarded wavelets corresponding to grid points where the details are small. Moreover the error committed when doing this can be estimated, and thus one can compare the error committed on the adaptive grid to this one would commit on the finest grid used.

Let us finally describe the algorithm for the adaptive Vlasov method:

- **Initialization**: decomposition and compression of f_0.
- **Prediction in** x of the grid $\tilde{G}$ (for important details) at the next split time step following the characteristics forward. Retain points at level just finer.

- **Construction of** $\hat{G}$: grid where we have to compute values of f^* in order to compute its wavelet transform.
- **Advection-interpolation in** x: follow the characteristics backwards in x and interpolate using wavelet decomposition $f^*(x, v) = f^n(x - v\,\Delta t, v)$
- **Wavelet transform of** f^*: compute the c_k and d_k coefficients at the points of $\tilde{G}$.
- **Computation** of electric field from Poisson.
- Same procedure for the velocity advance.

In order to validate the adaptive method we perform a simulation of a semi-Gaussian beam of potassium ions with energy 80 keV. The simulation is only 2D in phase space, (r, v_r), as we take particles with a vanishing canonical angular momentum which is conserved. The beam is submitted to a uniform focusing field and the tune depression is 0.25. The results are shown in Figure 5. The points where f is actually being computed on the adaptive grid are displayed below the corresponding phase space snapshot of the distribution function. One can here very well observe that the grid points automatically go to zones where the variation of the distribution function is large as we wanted.

This method does a good job optimizing the number of mesh points where the distribution function is actually computed. However there is an important overhead consisting in determining the needed mesh points. Moreover, as the number and position of grid points evolves in time, the implementation is a lot more complex and in particular harder to parallelize. As for PIC methods it becomes again a real problem to achieve a good load balance. Therefore in order to be useful the reduction in the number of points needs to be fairly important, say roughly of the order of 90% which can not be achieved for all problems. We yet need to investigate the method in higher dimensions to really assert its realm of application.

5. Conclusions

As of today, we can say that direct Vlasov methods are more a complement to PIC methods than a competitor: PIC method are fast and give accurate results for the macroscopic quantities over long times, however due to their noise, it is difficult to obtain a good precision on the microscopic quantities. Grid based Vlasov method need important computational resources but give a very good precision on microscopic quantities on short time scales. Numerical diffusion makes them less accurate over longer time scales, however even then they seem to reproduce accurately the macroscopic quantities.

Therefore, we can say that Vlasov methods are particularly useful as a second method in addition to PIC for specific situations, e.g. investigation of low density phase-space regions over short times and also as their numerical features are very different from those of PIC they are important for cross-checking and benchmarking PIC codes and verifying whether simulation results are linked to numerical problems.

In order to enhance their range of applicability one can implement them on unstructured meshes. The method has been implemented and works. However it is a lot more complicated and fairly slower, but might be useful in some specific situations. Adaptive methods look promising in 2D phase space as they allow to optimize the number of grid points used at each time step. However the implementation is more complex due to the adaptive mesh structure and due to the overhead of the mesh prediction the method only brings a gain when the number of mesh points is considerably reduced. But this is the case in many beam problems. More work is needed to assess for which kind of problems it will be really useful. We are currently working on the extension of the method to higher dimensions.

322
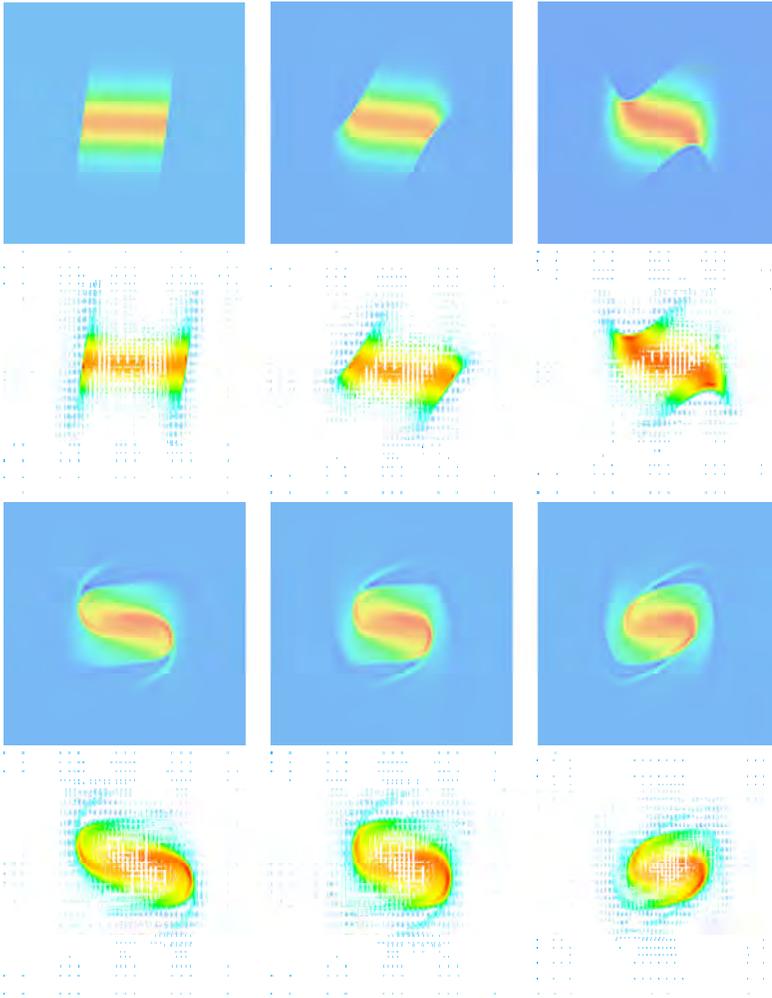
Figure 5. Semi-Gaussian beam evolution in applied and self field

References

[1] T.D. Arber, R.G.L. Vann "A Critical Comparison of Eulerian-Grid-Based Vlasov Solvers" 2002 *J. Comput. Phys.* **180** pp. 339-357.
[2] R. Bermejo "Analysis of a class of quasi-monotone and conservative semi-Lagrangian advection schemes" 2001 *Numer. Math.* **87** no. 4, 597–623.
[3] N. Besse "Semi-Lagrangian schemes for the Vlasov equation on an unstructured mesh of phase space" IRMA preprint 028 (2002)
http://www-irma.u-strasbg.fr/irma/publications/2002/02028.shtml
[4] N. Besse, F. Filbet, M. Gutnic, I. Paun, E. Sonnendrücker, ENUMATH 2001, Ischia proceedings.
[5] J. P. Boris, D. L. Book "Solution of continuity equations by the Method of Flux-Corrected Transport" 1976 *J. Comput.. Phys.* **20** pp. 397-431
[6] C.Z. Cheng, G. Knorr 1976 *J. Comput. Phys.* **22** pp. 330-348
[7] E. Fijalkow "A numerical solution to the Vlasov equation" 1999 *Comput. Phys. Communications* **116** pp.

319-328

[8] F. Filbet, E. Sonnendrücker "Comparison of Eulerian Vlasov Solvers", to appear in *Computer Physics Communications*

[9] F. Filbet, E. Sonnendrücker, "A Positive and Flux Conservative scheme for the numerical resolution of the Vlasov equation" *submitted*.

[10] E. Sonnendrücker, J. Roche, P. Bertrand, A. Ghizzo "The Semi-Lagrangian Method for the Numerical Resolution of Vlasov Equations" 1998 *J. Comput. Phys.* **149** pp. 201-220

[11] The VADOR code for Direct Vlasov Simulations
     http://www-irma.u-strasbg.fr/~filbet/index_vad.html

[12] F. Filbet, E. Sonnendrücker and J.-L. Lemaire "Direct Axisymmetric Vlasov Simulations of space charged dominated beams. Lecture Notes in Computer Sciences" ICCS 2002, part 3, Springer-Verlag, 2002, pp. 305–314.

[13] T. Nakamura, T. Yabe "Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov-Poisson equation in phase space" 1999 *Comput. Phys. Communications* **120** pp. 122-154

Space charge studies and comparison with simulations using the FNAL Booster

Panagiotis Spentzouris† § **James Amundson**† **James Lackey**†
Linda Spentzouris‡ **and Raymond Tomlin**†

† Fermi National Accelerator Laboratory, PO Box 500, Batavia, IL 60510, USA
‡ Illinois Institute of Technology, Physics Division, 3101 South Dearborn St., Chicago, IL 60616, USA

Abstract. We present measurements of transverse and longitudinal beam phase space evolution during the first thirty turns of the FNAL Booster. We discuss the experimental techinque, which allowed us to obtain turn-by-turn measurements of the beam profile. We then compare our results with the prediction of the Synergia 3D space charge simulation code.

1. Introduction

The Fermilab Booster is a rapid-cycling, 15 Hz, alternating gradient synchrotron with a radius of 75.47 meters. The lattice consists of 96 combined function magnets in 24 periods, with nominal horizontal and vertical tunes of 6.7 and 6.8 respectively. The Booster accelerates protons from a kinetic energy of 400 MeV to 8 GeV, at a harmonic number h=84, using 17 rf cavities with frequency which slews between 37.7 MHz (at injection) and 52.8 MHz (at extraction). The revolution time at injection is 2.2 μ s. A comprehensive technical description of the Booster as built can be found in reference [3]. The injection system utilizes the H$^-$ charge-exchange injection technique [2]. The typical linac peak-current is 45 mA; usually up to ten turns of H$^-$ beam are injected in the booster. The injected beam is a stream of bunches equally spaced at the linac rf frequency of 201.2 MHz. During injection, a pulsed orbit bump magnet system (ORBUMP) is used to superimpose the trajectories of the circulating (protons) and injected (H$^-$) beams.

There are many factors affecting the behavior of the Booster beam, including the energy and emittance of the incoming beam, nonlinear field errors and space charge effects, which is believed to be responsible for a significant fraction of the observed losses in the Booster [4], during the first 2 ms of the cycle (injection, capture, and bunching phase). In general, space charge effects are recognized as one of the most important problems which limit the total number of particles in a low energy proton synchrotron. Since the performance of the Booster is what makes or breaks the current (MiniBooNE experiment) and future (MINOS experiment) FNAL neutrino programs, and its stable operation is required for the current FNAL collider program, it is essential to study and quantify these effects. In order to achieve this goal, we have developed a full three dimensional (3D), Particle In Cell (PIC) model of the booster, based on the package *Synergia* [1]. The *Synergia* package has been developed under the DOE SciDAC initiative for accelerator modeling *Advanced Computing for 21st Century Accelerator Science and Technology*. Synergia incorporates existing packages for modeling 3D space charge and computing transfer maps using Lie algebraic techniques. It utilizes a split operator technique for particle propagation, includes a parser of the *Methodical*

§ To whom correspondence should be addressed (spentz@fnal.gov)

Accelerator Design (MAD) language, and has multi-turn injection modeling capabilities. The code has the capability to compute higher order transfer maps, but linear maps were used for the simulations presented in this paper.

One of the most important tasks in the process of studying the performance of a real accelerator using a simulation is to establish the validity of the model, by comparing the prediction of the simulation to that of data from experiments with well defined initial conditions. This is a very complicated process, since it not only involves a large number of different parameters that should be kept under control, but also requires very good understanding of the instrumentation used to perform these measurements. In the following sections, we will describe the first steps of such a validation process for the *Synergia* package, using data taken with the FNAL Booster.

2. Experimental data

The objective of the experiment was to study the beam evolution in the first few turns after injection, by comparing beam widths (both transverse and longitudinal) to the simulation as a function of time, with single turn resolution, and for different beam currents. The FNAL Booster has two measuring devices capable of measurements of beam widths with single turn resolution: the *Ion Profile Monitor* detector (IPM) [5], which utilizes the ions from ionization of the residual gas by the proton beam to measure transverse beam profiles, and the *Resistive Wall Monitor* (RWM) device, which utilizes the induced current on the beam pipe by the particle beam, to measure the longitudinal beam profile. Since the response of the IPM depends on the charge of the beam, and since the goal of the experimental program is to use this detector for a quantitative study of space charge effects, we installed a third measuring device, in order to check and calibrate the performance of the IPM. This device utilizes a single wire placed just outside the beam envelope of the displaced beam orbit at the injection region. To obtain single turn time resolution, we use a "flying beam" rather than a "flying wire" technique. At injection, the ORBUMP magnets keep the beam trajectory displaced by ~ 4 cm with respect to the nominal beam orbit. The wire is placed between the displaced and nominal orbits. As the ORBUMP current decays back to zero, the beam moves back to the nominal trajectory, sweeping through the wire, and thus providing information about the transverse beam profile (in the bending –horizontal– view only). By recording the ORBUMP current and the response of the wire we can reconstruct the horizontal profile (the ORBUMP current translates very accurately to beam position). In order to control for which turn number the beam goes through the wire, we change the timing of the injected beam with respect to the time that the ORBUMP current rises, thus controlling for which turn number the measurement is made. There are two drawbacks in this method: first, the range is limited by the amount of time that the ORBUMP current stays on (about 30 turns equivalent), and second, since the injection timing with respect to the ORBUMP current plateau has to change to observe a different turn number, the measurements do not utilize the same beam, i.e if the width of turn number N is measured during a given cycle of the machine, turn N+1 can only be measured using one of the next cycles.

2.1. DC beam studies

This data set was obtained with the Booster running DC (rf system off and no ramping of the magnet power supplies). This was done in order to simplify the running conditions and reduce the number of parameters in the comparisons. To further reduce complications in the initial conditions, only a single turn worth of Linac beam was injected in the machine. The current of the Linac beam was controlled by detuning one of the Linac quadrupoles. Under
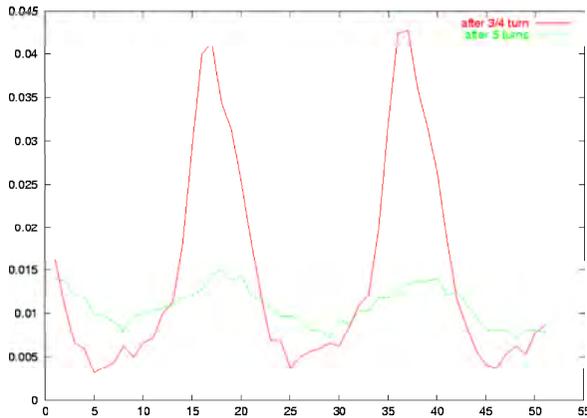
Figure 1. Beam current (arbitrary units) versus time/0.2×10^{-9} s from the RWM.

these conditions, we took data with injected beam of $11, 20$, and 42 mA. A snapshot of two of the injected 200 MHz Linac bunches is shown in figure 1 3/4 turn after injection (the location of the RWM) and 4 and 3/4 turns after injection. The picture is from the 42 mA data set. Already, after ~ 5 turns in the machine, the beam distribution in time is almost flat within the 200 MHz time slices. In figure 2 the RMS of the time distribution of the beam in a 200 MHz time slice is plotted versus the turn number. The data (points) is compared to the simulation (lines) for the different values of injected beam current. There is good qualitative agreement between data and simulation. The model predicts very small effects due to the different beam currents, well within the uncertainty of the measurement. Both the data and simulation extracted RMS widths tend to a constant value, since they are calculated within a 200 MHz time slice. Note that in the simulation we only model one 200 MHz bunch with periodic boundary conditions. This is an accurate representation of the main body of the beam (see discussion in [1]). The $\Delta p/p$ used in the simulation was 0.2%. This value compares very well with the value 0.12% extracted from the decay of the beam's peak current as function of time, measured with the RWM; see figure 3. For this measurement the RWM was ran at low time resolution, in order to allow the observation of the beam for a long period of time. Because of that, the data set is not very clean, with aliasing effects and noise. The data is filtered (second curve in figure 3) and the time constant of the exponential decay of the peak current is extracted. This time constant is equal to the debunching time, dt/t, which relates to the momentum spread by $dp/p = (dt/t)/\eta$. For this data set, we only obtained transverse profiles with the IPM. A sample comparison is shown in figures 4 for the horizontal, and 5 for the vertical beam profiles, for a beam current of 11 mA. The beam was injected with an offset in both the vertical ($\sim -2mm$) and the horizontal ($\sim 7mm$). In both cases, the width of the beam, represented by the size of the bars in the plots, is well modeled by the simulation. Also, in both cases the IPM measurement of the mean of the beam is suspect, since it does not exhibit the expected oscillatory behavior (or if it does, it shows a very small amplitude). The simulation clearly exhibits the expected behavior. The other data sets, 20 mA and 42 mA beam currents, have the same characteristics.

FNAL Booster space-chage modeling and experiment
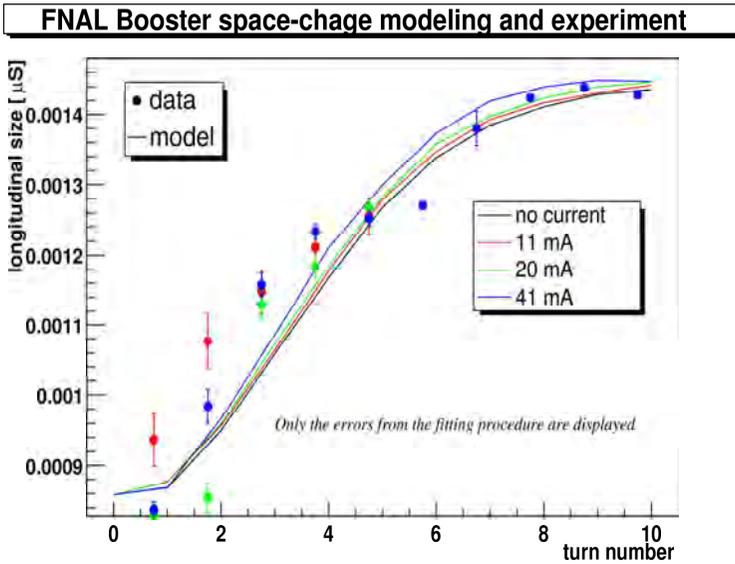
Figure 2. Longitudinal phase space evolution. Beam distribution RMS width in time as a function of turn number. The data from the RWM measurements (points with error bars) is compared to the model prediction from *Synergia* (lines).
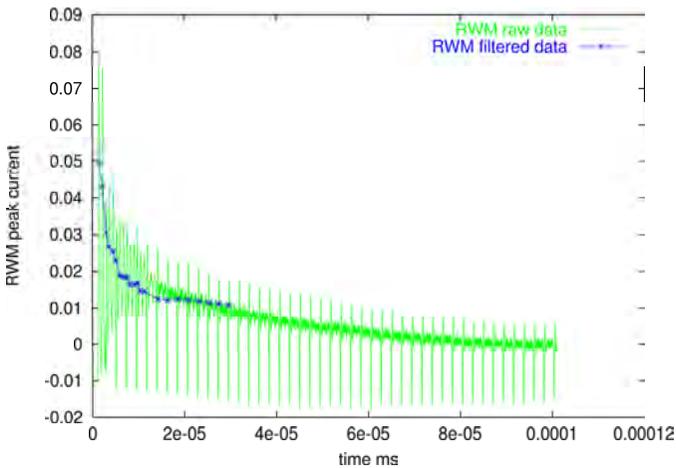
Figure 3. Peak beam current (arbitrary units) versus time/1×10^{-8} s from the RWM.

2.2. Multi-turn injection studies

This data set was obtained with the Booster cycling but with no net acceleration (the rf cavities were para-phased, i.e ran in pairs with equal and opposite phases). The beam from the Linac
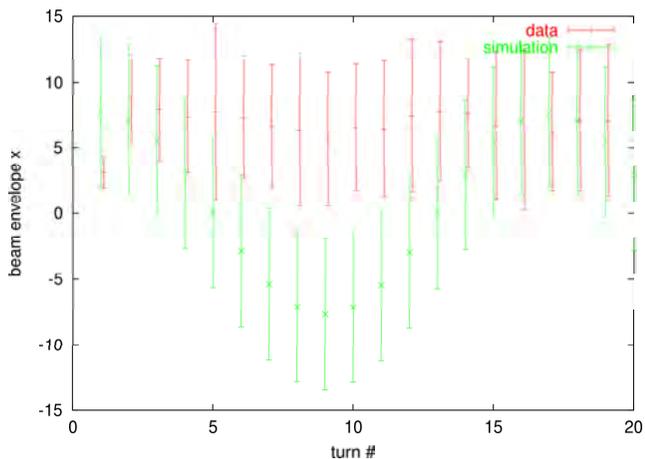
Figure 4. Horizontal beam width evolution as a function of turn number for beam current of 11 mA. The points represent the location of the mean of the beam while the size of the bars the width of the beam at this location. The IPM data are shown in red (points represented by a dash) and the model prediction in green (points represented by a star).
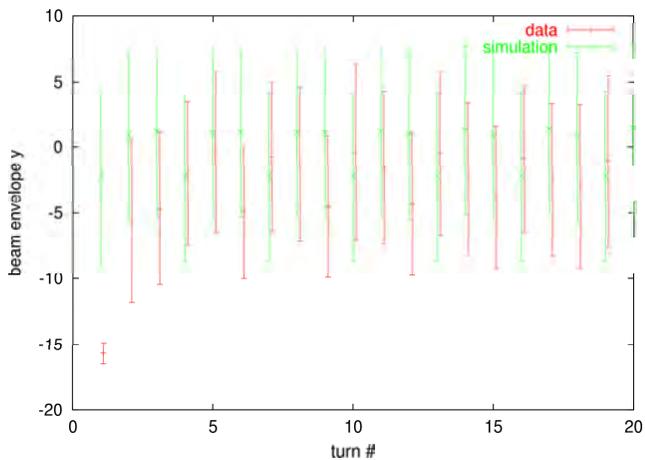
Figure 5. Vertical beam width evolution as a function of turn number for beam current of 11 mA. The points represent the location of the mean of the beam while the size of the bars the width of the beam at this location. The IPM data are shown in red (points represented by a dash) and the model prediction in green (points represented by a star).
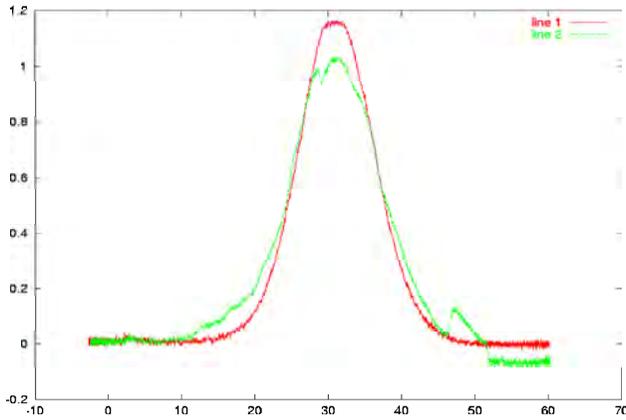
Figure 6. Sample of "flying beam" wire profile measurements. The horizontal axis is in mm. The red curve shows an example of a clean profile measurement, the green a case of a noisy one.

was injected in multiple turns. We collected data with one, five, and thirteen turn injection (13×40 mA beam current). No IPM data were taken, only "flying beam" wire data. An example of wire profiles is shown in figure 6. Most of the measurements were clean, like the one shown with the red curve in the figure. In some cases the data had additional "structure", such as the one shown with the green curve in the above figure. For all the data sets collected,
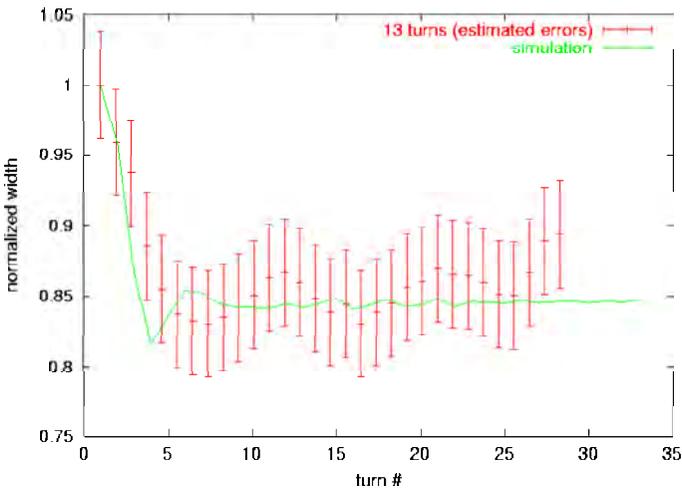
Figure 7. The normalized horizontal beam width (width divided by the width of the injected beam) versus turn number as measured with the wire (points) and extracted from the simulation (line). The error bars represent the estimated experimental error.

the beam was injected with a 20% mismatch in x'. A comparison of data and simulation is

shown in figure 7. The simulation qualitatively describes the data within the experimental uncertainty. The other data sets (different beam currents) have the same characteristics.

3. Summary

We have used *Synergia*, a beam dynamics package with 3D space charge capabilities, to model the FNAL Booster. We have collected data with the machine and compared the measured beam phase space evolution to the model prediction for the few first turns after injection. The model does a reasonable job describing the characteristics of the data in both the transverse and longitudinal coordinates. More work is needed (and is underway) to understand and cross-calibrate the instrumentation. The next step in this program will involve modeling and measurements of the beam parameters further into the machine cycle, where space charge effects are expected to be large.

4. References

[1] Amundson J and Spentzouris P 2002 *this proceedings*
[2] Ankenbrandt C *et al* 1980 *Proceedings of the 11th International Conference on High-Energy Accelerators* p 260
[3] Booster Staff 1973 *Booster Synchrotron* ed E L Hubbard *Fermi National Accelerator Laboratory Technical Memo TM-405*
[4] Popovic P and Ankenbrandt C 1998 *Workshop on Space Charge Physics in High intensity Hadron Rings* ed A U Luccio and W T Weng (Woodbury, New York: AIP Conference Proceedings) p 128
[5] Zagel J, Chen D, and Crisp J 1994 *Beam Instrumentation Workshop* (AIP Conference Proceedings 333) p 384

Progress in the study of mesh refinement for particle-in-cell plasma simulations and its application to heavy ion fusion

J.-L. Vay†

† Lawrence Berkeley National Laboratory, Berkeley, CA, USA

A. Friedman, D. P. Grote‡

‡ Lawrence Livermore National Laboratory, Livermore, CA, USA

Abstract. The numerical simulation of the driving beams in a heavy ion fusion power plant is a challenging task, and, despite rapid progress in computer power, one must consider the use of the most advanced numerical techniques. One of the difficulties of these simulations resides in the disparity of scales in time and in space which must be resolved. When these disparities are in distinctive zones of the simulation region, a method which has proven to be effective in other areas (e.g. fluid dynamics simulations) is the Adaptive-Mesh-Refinement (AMR) technique. We follow in this article the progress accomplished in the last few months in the merging of the AMR technique with Particle-In-Cell (PIC) method. This includes a detailed modeling of the Lampel-Tiefenback solution for the one-dimensional diode using novel techniques to suppress undesirable numerical oscillations and an AMR patch to follow the head of the particle distribution. We also report new results concerning the modeling of ion sources using the axisymmetric WARPRZ-AMR prototype showing the utility of an AMR patch resolving the emitter vicinity and the beam edge.

1. Introduction

The numerical simulations of the beam ions transport in a Heavy Ion Fusion [1] accelerator and reaction chamber currrently model different stages of the process separately. A completely self-consistent simulation, which is ultimately needed, requires an end-to-end simulation from the ion source to the fusion target. This represents a real challenge even extrapolating near-future computer power from current state and past progress and we must consider the use of the most advanced numerical techniques. One of the difficulties of these simulations resides in the disparity of scales in time and in space which must be resolved. When these disparities are in distinctive zones of the simulation region, a method which has proven to be effective in other areas (e.g. fluid dynamics simulations) is the Adaptive-Mesh-Refinement (AMR) technique. We have begun at LBNL the exploration of introducing this technique into the method that we use the most (i.e. Particle-In-Cell or PIC) and started a collaboration with NERSC researchers to develop an AMR library of subroutines targeted at providing AMR capabilities for existing plasma PIC simulation codes[5]. In [5], we have exposed the main issues associated with the coupling of the AMR and PIC techniques, for both electrostatic and magnetostatic simulations. In this article, we present additional results obtained with a one-dimensional and a two-dimensional axisymmetric prototypes of Particles-In-Cell+Adaptive-Mesh-Refinement code.

2. Time-dependent modeling of transient effects in diode with fast rise-time

The control of the beam head by using appropriate shapes and rise-time of the applied voltage controlling the beam extraction and acceleration in the source is of great importance if one needs to avoid ion loss to the wall and its adverse effects. Fig.1 displays snapshots of the beam head in the matching section of the High-Current Experiment (HCX[2]) conducted at LBNL for two different rise-time (using the same waveform obtained from experimental data). Using a rise-time of 800ns, the simulation predicts a significant mismatch of the head resulting with loss of a small fraction of particles at the wall. According to the simulations, shortening the rise-time to 400ns provides a better match of the beam head with outer particle trajectories away from any structure.

Short pulse (200ns flat top) with very short rise-time (¡50ns) are envisioned for the next possible HIF experiment, the Integrated Beam Experiment (IBX[2]). The modeling in this range of parameter is quite challenging and the simulation tools must first be proven to be efficient on similar problems for which the solution is known.

For a one-dimensional diode, Lampel and Tiefenback [4] demonstrated that an analytical solution for a waveform of the applied voltage producing a Heaviside step for the current profile existed and is given by

$$V(t) = \frac{1}{3} \frac{t}{t_{transit}} \left[4 - \left(\frac{t}{t_{transit}} \right)^3 \right] V_{max} \tag{1}$$

where V_{max} is the voltage that is applied at steady state and $t_{transit}$ is the transit time of a particle from the emitter to the collector. The latter quantity is given by

$$t_{transit} = 3 \cdot d \cdot \sqrt{\frac{m}{2qV_{max}}} \tag{2}$$

where q and m represent the charge and mass of particles and d is the distance between the emitter and the collector.

In WARP[3], the standard algorithm for injecting particles uses a virtual surface located at distance d_i from the emitting surface. After a field solve, the potential drop V_i computed between the emitter and the virtual surface is used to evaluate a current $I = \chi V_i^{3/2}/d_i^2$ assuming a Child-Langmuir emission between the two surfaces ($\chi = \frac{4}{9}\varepsilon_0\sqrt{2q/m}$). The obtained current is used to launch N new macroparticles using the formula $N = I\delta t/q_m$ where q_m is the charge of a macroparticle and δt is the time step.

Using this algorithm on a uniform grid of 160 cells, a diode length of 0.4 meters, a time step of 1ns and a steady-state current of 30A with the Lampel-Tiefenback waveform as given in (1), we have obtained the current history given in Fig.2,a). We observe that we get large amplitude low-frequency oscillations slowly damping in time. The history of the number of particles injected at each time step that is given in Fig.3 also features high amplitude low-frequency oscillations which may be the cause of the oscillations observed on the current history.

2.1. Application of the Lampel-Tiefenback method at the discrete level

The only parameter controling the number of particles injected at each time step is the voltage drop V_i computed between the emitter and the virtual surface. Getting a Heaviside current profile implies to emit a fix number of particles per time step from time zero, implying to start with a non-zero applied voltage, in contradiction with the Lampel-Tiefenback profile. In order to circumvent this contradiction which was due to the application of a solution obtained

using infinitesimal calculus to discretized simulations, we decided to apply the method that Lampel and Tiefenback used to derive their solution at the discretized level directly. Knowing the current I and the distance d_i, the quantity V_i is uniquely determined by $V_i = \left(Id_i^2/\chi\right)^{2/3}$ and must be a constant to ensure a constant number of particles emitted per time step. Using the linearity of the Poisson equation, we can separate the overall solution into one component V_0 resulting from the emitter-collector system with applied voltage but no charge between the electrodes and another component V_g resulting from the system with charge but grounded electrodes. We name V_{0i} and V_{gi} the potentials corresponding to these two components at the virtual surface location. At any given step, we want $V_i = V_{0i} + V_{gi} = Constant$. The charge density profile is known from the particle distribution at all time steps. Thus V_{gi} is always defined and we can compute $V_{0i} = V_i - V_{gi}$. Remarking that V_0 is a scaling of the solution at maximum voltage without charge, we get that the voltage that is to be applied between the electrodes at a given time step is given by $V_{applied} = V_{max} \cdot V_{0i}/V_{imax} = V_{max}\left(V_i - V_{gi}\right)/V_{imax}$ where V_{imax} is the voltage on the virtual surface at maximum voltage with no charge. The history of the voltage thus obtained is displayed in Fig.4,a) and compared to the infinitesimal Lampel-Tiefenback solution. As expected, the voltage does not start at zero but at almost a quarter of the maximum voltage. It is quite remarkable to notice that for times greater from the transit time of particles from the emitter to the virtual surface, the computed voltage history catch-up with the infinitesimal profile and become indistinguishable from it. As can be seen on Fig.2,b), starting at a non-zero value for the voltage in order to get a constant number of particles emitted at each time step provides a current history profile which is closer to a Heaviside step, except for a peak at the front.

2.2. Subgridding of the emitter to virtual surface region

As informative and useful the application of the Lampel-Tiefenback technique at the discrete level to obtain an ideal current profile is, it is of little help for modeling the beam response to profiles that differ from the ideal profile, as may be imposed by experimental constraints. However, if we can get to lower the initial voltage value as obtained by this technique to a negligible value, then applying this technique or applying the infinitesimal voltage solution will be equivalent.

If we set the maximum voltage to start the simulation with to be one per-cent of V_{max}, then we get from the previous relations and a little algebra that we would have to augment the resolution by a factor of 10000, meaning a grid of 1.6 million meshes for a 1-D simulation. It is clear that the use of an irregular mesh imposes itself at this point. For easy extension to 2-D and 3-D (which description is differed to another article), we have opted for a subgrid patch which extends from the emitter to the virtual surface. In order to be able to get to a very fine resolution close to the emitter, we set the mesh spacing so as to obtain a uniform charge density in the patch at steady-state (assuming Child-Langmuir flow). At each time step, the potential is first obtained on the main regular grid and the boundary values of the patch are interpolated from the main grid solution. The field is then solved inside the patch (note that charge density deposition is performed inside both the main grid and the patch). Using 200 cells in the patch allows for an initial applied voltage to be less than one per-cent of V_{max} and we can check on Fig.4,b) that the voltage history profile obtained from the simulation is now undistinguishable from the infinitesimal Lampel-Tiefenback solution from the start. This has been obtained without deterioring the current profile, as can been checked in Fig.2,c), which is almost identical to the one of Fig.2,b). The peak, however, that we may have assumed to be caused by the voltage starting at a non-zero value is still present. Our assumption was at this point that the peak is due to an underresolved front of the particle distribution.

2.3. Application of an Adaptive-Mesh-Refinement patch following the particle distribution front

In order to check this assumption, we have added an Adaptative-Mesh-Refinement patch following the front of the particle distribution. The patch had a resolution sixteen times higher than the main grid and its position was reset at each time step according to the front position, computed by locating the maximum absolute value of the current profile space derivative. The result is displayed in Fig.2,d) and shows that the peak has now disappeared, giving a result remarkably close to the infinitesimal solution. Another calculation was made at four times the resolution used so far (mesh size x4, nb particles x4, time step/4) and snapshots of the results are displayed in Fig.5. It shows that all the computed quantities are very close to the infinitesimal solution.

3. Progress on the simulation of ion source using WARP axisymmetric AMR prototype

We have also pursued the simulation of the configuration of the source using the axisymmetric PIC-AMR model prototype implemented in WARP, as described in [5]. We have showed in [5] that setting a mesh refinement patch around the emitter region was helpful in getting result almost as accurate as a run at higher resolution, at a fourth of the computational cost. The edge of the beam, however, did not seem to be as well modeled as the core and we have done the same simulation using this time an Adaptive-Mesh-Refinement patch which covers the vicinity of the emitter but also follows the beam edge (see Fig.6, top). Fig.6,bottom) displays a comparison of the emittance profiles obtained once the beam has reached steady-state for three different resolutions without using AMR (ngf: factor of number of grid meshes for each dimension, npf: factor number of particles) and one case using AMR. The phase-space projection in the space R-R' and the charge density profiles taken at the exit of the source are also given in Fig.7. Using four times less particles, the run using AMR uses a fourth of the memory and a fourth of the run time necessary to achieve essentially the same result as obtained from a run on a unique regular grid at the highest resolution used in this test.

4. Conclusion

We have extended the work presented in [5] by performing a 1-D time-dependent and a 2-D axisymmetric converged to steady-state tests using both the AMR method coupled to the Particle-In-Cell technique. For time-dependent simulation, the use of the AMR patch as proved to be, together with the use of an irregular gridded patch, indispensable for reaching an accurate modeling of the beam front. For steady-state simulations, AMR is essential for modeling the beam edge with accuracy while maintaining a good level of statistic in the core of the beam. The obtained results demonstrate that the Adaptive-Mesh-Refinement technique can be successfully used with the Particle-In-Cell technique and open the way to simulations that would otherwise be out of reach on current and near-future hardwares.

References

[1] http://hif.lbl.gov
[2] http://hif.lbl.gov/VNLresearch
[3] D. P. Grote, A. Friedman, I. Haber, "Three-Dimensional Simulations of High Current Beams in Induction Accelerators with WARP3d". Fus. Eng. & Des. **32-33**, 193 (1996)
[4] M. Lampel and M. Tiefenback, "An Applied Voltage to Eliminate Current Transients in a One-dimensional Diode", *Appl. Phys. Lett.* **43**, No. 1, 1 (1983)
[5] J.-L. Vay et al, "Mesh Refinement for Particle-In-Cell Plasmas Simulation: application - and benefits for - Heavy Ion Fusion", *Laser and Particle Beams*, to be published
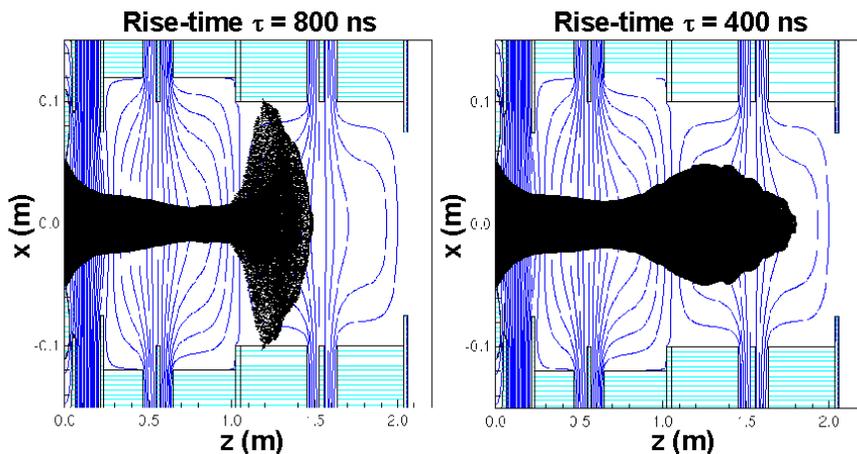
Figure 1. Snapshot of the beam head for simulations of the HCX experiment for two different rise-time of the applied voltage. On the left, the rise-time was of 800 ns (wavefrom and rise-time were given from actual experimental data) and the simulation predicts an explosion of the beam head, leading to loss of particles at the wall and raising concerns for potential breakdowns and electron effects. The same simulation using a shorter rise-time (400ns)
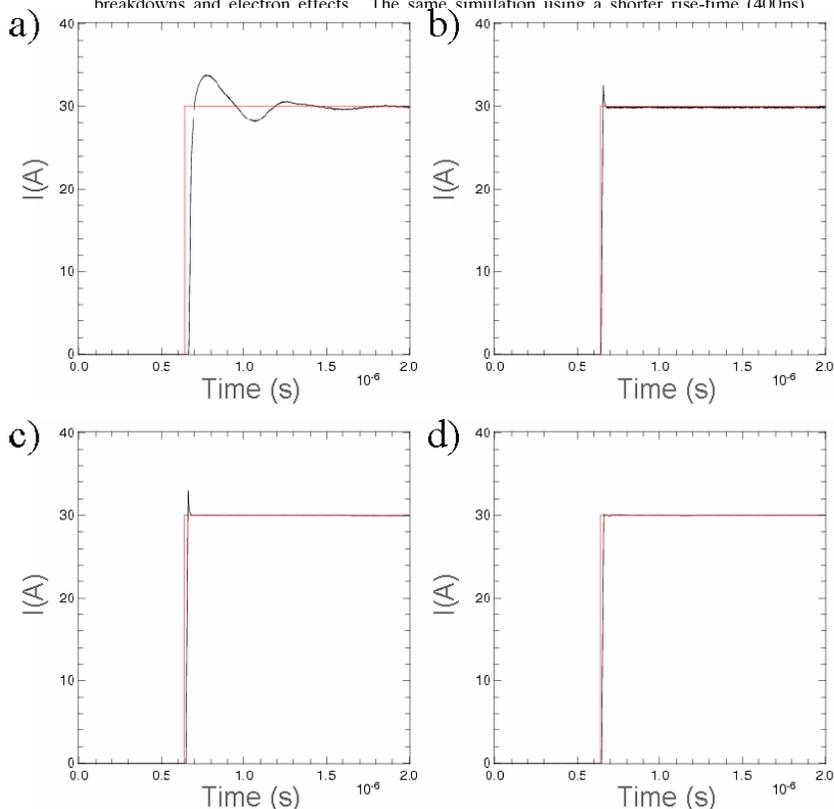
Figure 2. The current history is plotted versus time for four simulations using different numerical techniques. The black curve represents the history profile obtained from the simulation (averaged over ten time steps to damp high-frequency statistical noise). The red curve is the analytic solution. Details of algorithms are given in the text.
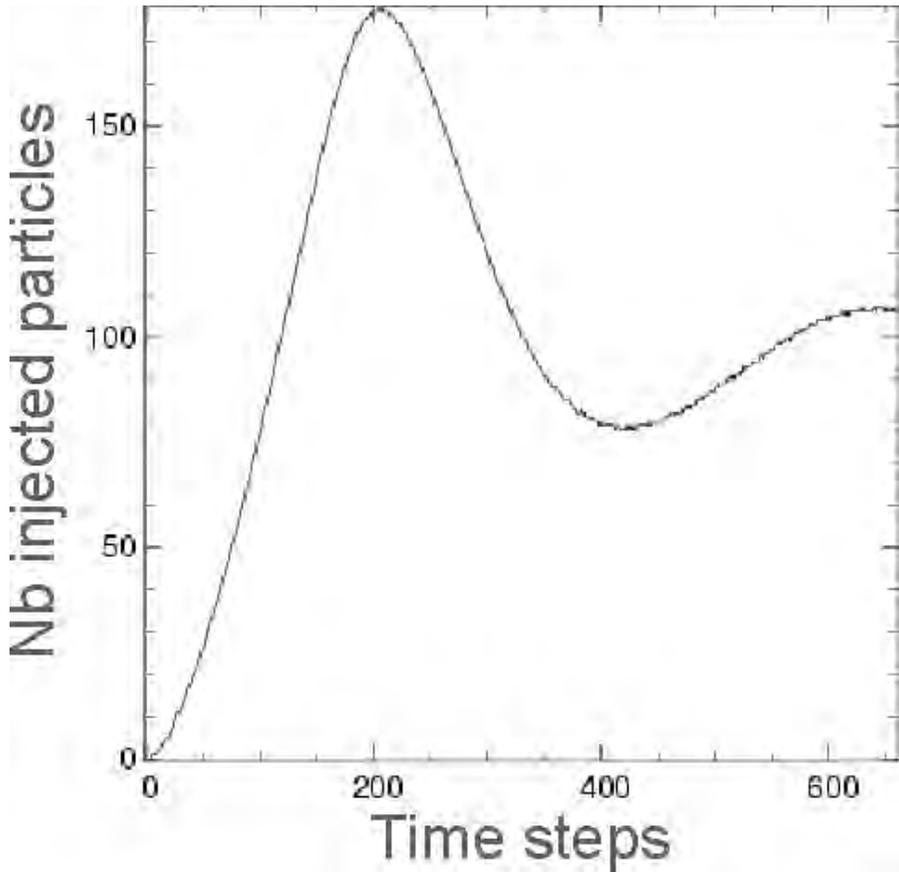
Figure 3. History of the number of particles injected per time step for the first 660 time steps corresponding to the case of Fig.2,a.
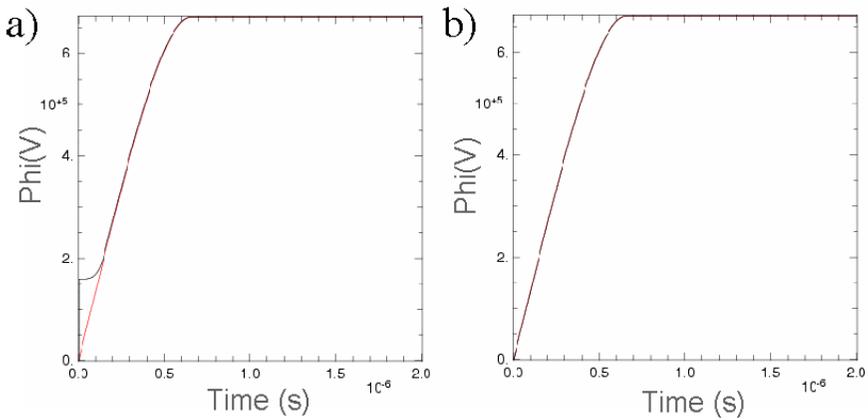
Figure 4. Voltage history used in results displayed in Fig.2,b) and Fig.2,c) respectively (black curves). The red curves is the infinitesimal Lampel-Tiefenback solution.
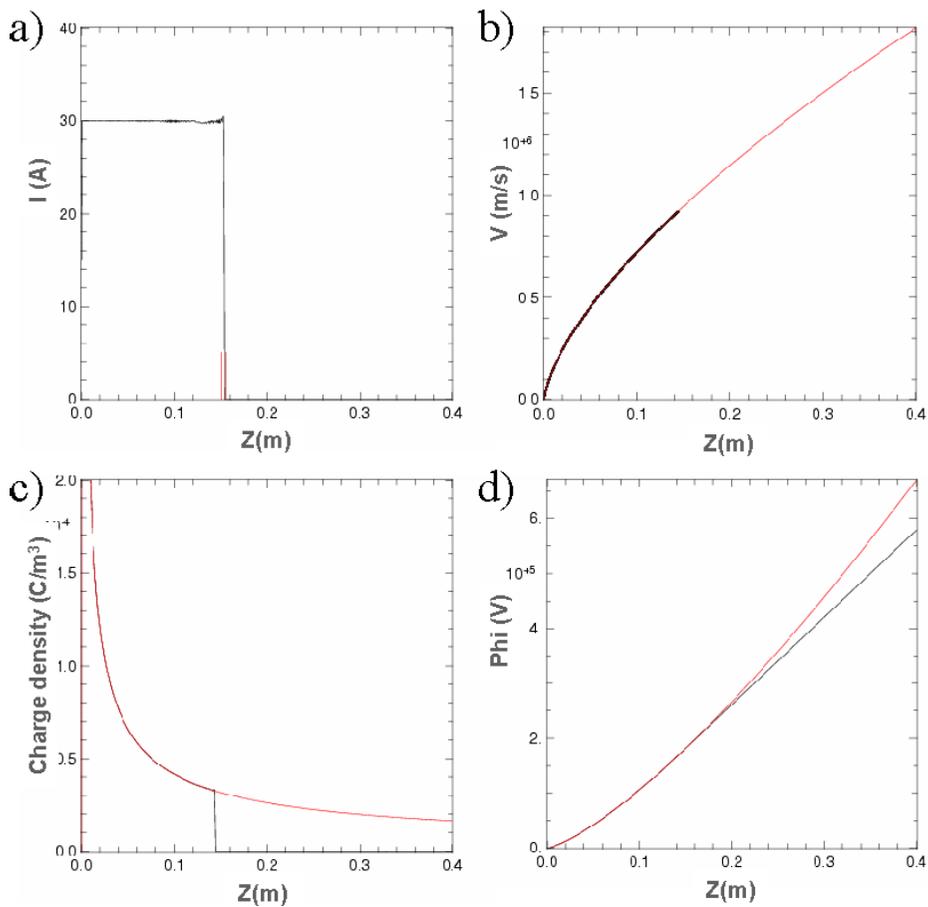
Figure 5. Snapshots of a) current profile, b) z-Vz phase-space, c) charge density, d) applied voltage versus z (black curves). The red curves represent the infinitesimal steady-state solutions except on a) where the steady-state current solution is not displayed and the two red marks show the limits of the AMR patch as located at the time step of measurement (the snapshots are from a calculation at four times the resolution of results displayed on previous figures).
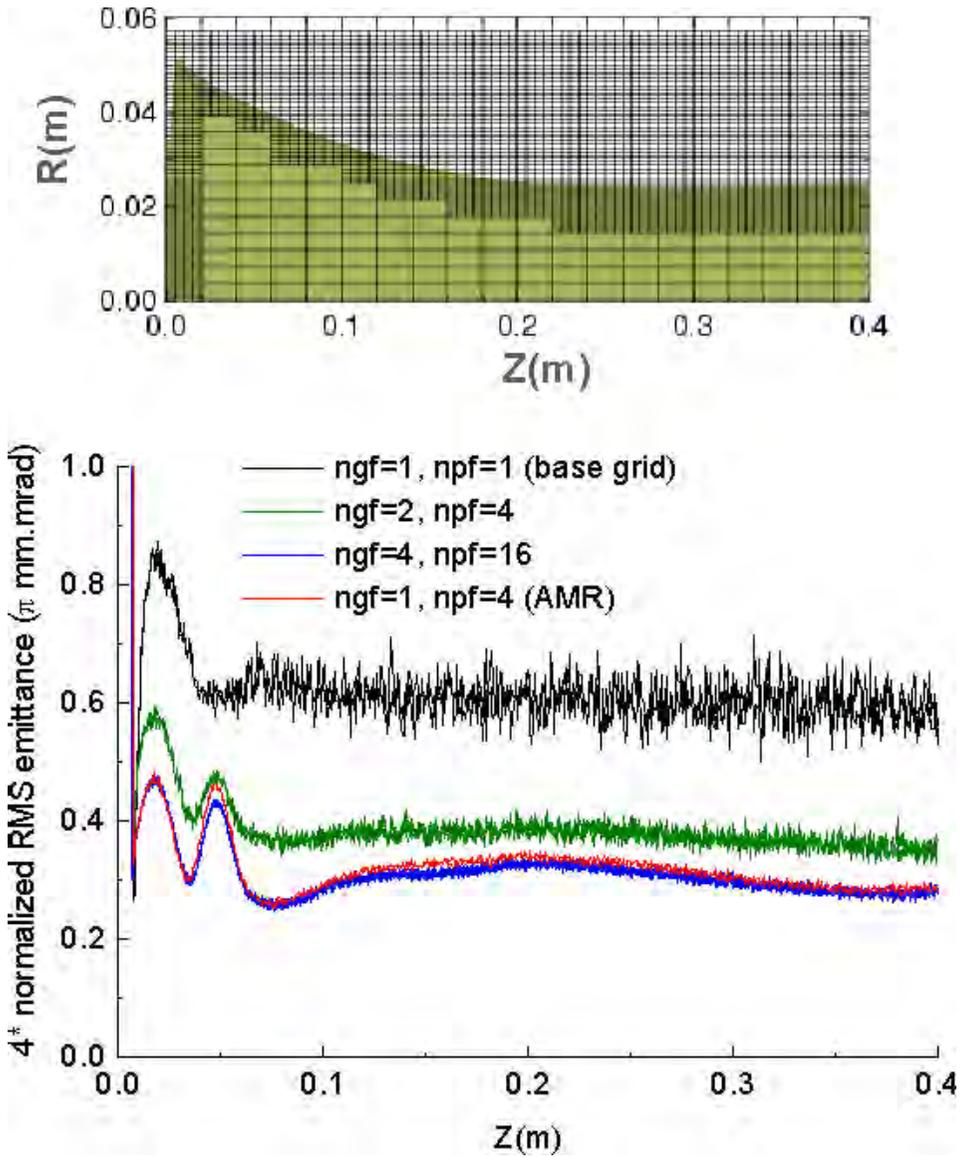
Figure 6. Top: prototype of AMR meshing used to follow the beam edge. The fine gridded area is remapped at each time step to cover the emitter region and the edge of the beam. Bottom: comparison of emittance results for three different resolutions without AMR and for one run using AMR.
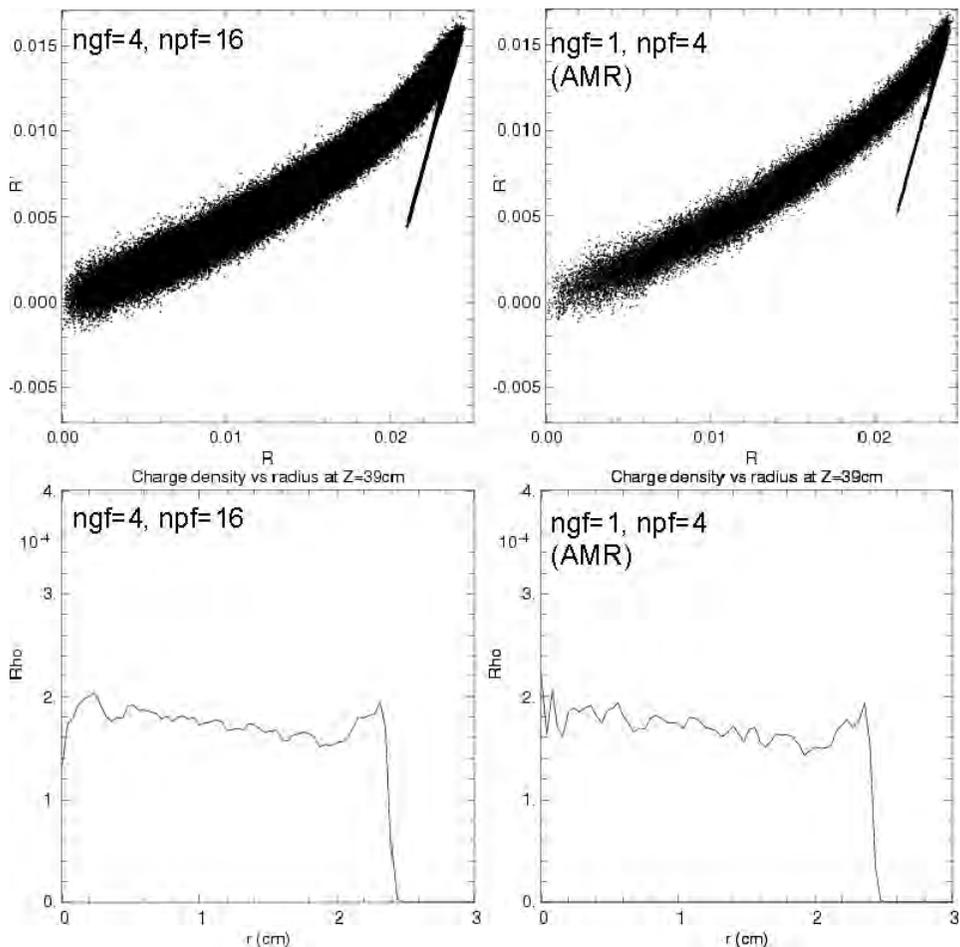
Figure 7. Left: phase-space projection and charge density at grid exit for run on one grid at highest resolution. Right: same data for run at coarser resolution using AMR patch to compensate. The run using AMR takes about a fourth of the run at highest resolution both in computer memory and run time, for an almost equivalent result.

Calculation of transversal wake potential for short bunches

Igor Zagorodnov and Thomas Weiland
Fachbereich 18, Computational Electromagnetics Lab.,
Technische Universitaet Darmstadt, Germany

Abstract. We present a new implicit scheme for calculation of transversal wake potential of short bunches in very long smooth structures. The scheme is able to model curved boundaries and does not suffer from dispersion in longitudinal direction. It is based on splitting the discrete curl operator in transversal and longitudinal parts. Unlike previous conformal approaches the scheme has a second order convergence without the need to reduce the maximal stable time step of conventional staircase method. This feature allows usage of a moving mesh easily. Several numerical examples are presented and algorithm is compared to other approaches.

1. Introduction

The beam dynamics of high-current linear accelerators is dominated by the effect of wake potentials of the highly conducting metallic structures surrounding the beam trajectory. In most cases the finite difference methods can be applied successfully for calculation of wake fields in accelerators [1], [2]. However the existing computer codes experience severe problems in short range wake field calculation for ultra short bunches [3]. Two main sources of the problems are the grid dispersion and the staircase geometry approximation.

To develop the scheme without dispersion in longitudinal direction we split the curl operator in the transversal and the longitudinal parts and use an implicit scheme based on the transversal part. The new scheme is conditionally stable and allows using the "magic" time step, equal to the space step in the longitudinal direction divided by the speed of light.

To overcome the staircase problem a conformal scheme described in the [4] is used. Unlike other conformal approaches this scheme is second order convergent without the need to reduce the maximal stable time step of the conventional staircase method. This feature allows to use a moving mesh and the "magic" time step without a need for interpolation.

In the second section we state the problem. In the third part the new implicit scheme is introduced. In the fourth part realization of the scheme in staircase approximation for rotationally symmetric case is described and then in the next section a conformal second order convergent scheme is developed. In the sixth part a new algorithm for indirect integration of wake potential is introduced. Finally we show several numerical examples and compare the scheme with other approaches.

2. Wave equation for vector potential

We consider a perfectly conducting structure S and assume that the bunch is moving in

domain Ω with the velocity of light $\mathbf{c}$ and is characterized by a charge distribution ρ. The bunch introduces an electric current $\mathbf{j} = \mathbf{c}\rho$ and thus we have to solve for

$$\nabla \times \mathbf{H} = \frac{\partial}{\partial t}\mathbf{D} + \mathbf{j}, \qquad \nabla \times \mathbf{E} = -\frac{\partial}{\partial t}\mathbf{B}, \tag{1}$$

$$\nabla \cdot \mathbf{D} = \rho, \qquad \nabla \cdot \mathbf{B} = 0$$

$$\mathbf{H} = \mu^{-1}\mathbf{B}, \qquad \mathbf{D} = \varepsilon\mathbf{E}, \qquad x \in \Omega,$$

$$\mathbf{n} \times \mathbf{E} = 0, \qquad x \in S$$

The full field $\mathbf{D}, \mathbf{H}$ can be decomposed into the field of the bunch in free space $\mathbf{D}^0, \mathbf{H}^0$ and a scattered field

$$\mathbf{D}^s = \mathbf{D} - \mathbf{D}^0, \qquad \mathbf{H}^s = \mathbf{H} - \mathbf{H}^0. \tag{2}$$

The scattered field can be presented by vector potential $\mathbf{A}$:

$$\mathbf{D}^s = \nabla \times \mathbf{A}, \qquad \mathbf{H}^s = \frac{\partial}{\partial t}\mathbf{A}. \tag{3}$$

Substitution of the presentation (3) in the system (1) gives the problem for the vector potential $\mathbf{A}$

$$\nabla \times \varepsilon^{-1}\nabla \times \mathbf{A} = -\frac{\partial^2}{\partial t^2}\mu\mathbf{A} - \mathbf{j}^s, \qquad \mathbf{j}^s = \nabla \times \varepsilon^{-1}\mathbf{D}^0 + \frac{\partial}{\partial t}\mu\mathbf{H}^0, \tag{4}$$

$$\nabla \cdot \mu(\frac{\partial}{\partial t}\mathbf{A} + \mathbf{H}^0) = 0, \qquad x \in \Omega,$$

$$\mathbf{n} \cdot \mathbf{A} = -\int_{-\infty}^{t}\mathbf{n} \cdot \mathbf{H}^0 d\tau, \quad \mathbf{n} \times \nabla \times \mathbf{A} = -\int_{-\infty}^{t}\mathbf{n} \times \nabla \times \mathbf{H}^0 d\tau, \qquad x \in S.$$

3. Implicit scheme

The new scheme will be introduced in context of Finite Integration Technique [5]. Starting from Maxwell's equation in integral form and introducing decomposition of the computation domain into a collection of cells, we obtain a set of discrete equations on a grid doublet:

$$\mathbf{C}\mathbf{e} = -\frac{d}{dt}\mathbf{b}, \quad \mathbf{C}\mathbf{h} = \frac{d}{dt}\mathbf{d} + \mathbf{j}, \quad \mathbf{S}\mathbf{b} = \mathbf{0}, \quad \mathbf{S}\mathbf{d} = \mathbf{q}. \tag{5}$$

They are completed by the discrete form of the material relations $\mathbf{e} = \mathbf{M}_{\varepsilon^{-1}}\mathbf{d}$, $\mathbf{h} = \mathbf{M}_{\mu^{-1}}\mathbf{b}$. To establish a time-stepping algorithm we can approximate the time derivatives in (5) by central difference expressions with $t_n = t_0 + n\Delta t$:

$$\mathbf{h}^{n+1} = \mathbf{h}^n - \Delta t \mathbf{M}_{\mu^{-1}}\mathbf{C}\mathbf{e}^{n+1/2}, \quad \mathbf{e}^{n+1/2} = \mathbf{e}^{n-1/2} + \Delta t \mathbf{M}_{\varepsilon^{-1}}(\mathbf{C}_1^T\overline{\mathbf{h}}^n + \mathbf{C}_2^T\mathbf{h}^n - \mathbf{j}^n), \tag{6}$$

$$\overline{\mathbf{h}}^n \equiv \theta\mathbf{h}^{n+1} + (1-2\theta)\mathbf{h}^n + \theta\mathbf{h}^{n-1},$$

where we have split the operator $\mathbf{C} = \mathbf{C}^T$ into the transversal operator $\mathbf{C}_1^T$ and the longitudinal operator $\mathbf{C}_2^T$ and θ is a numerical parameter to be defined. If we note the longitudinal coordinate by z and the transversal coordinates by r, φ, the operators have the form

$$\mathbf{C}_1^T = \begin{pmatrix} 0 & 0 & -\mathbf{P}_\varphi^T \\ 0 & 0 & \mathbf{P}_r^T \\ \mathbf{P}_\varphi^T & -\mathbf{P}_r^T & 0 \end{pmatrix}, \qquad \mathbf{C}_2^T = \begin{pmatrix} 0 & \mathbf{P}_z^T & 0 \\ -\mathbf{P}_z^T & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

From the system (6) we receive a numerical scheme for the vector potential $\mathbf{A}$:

$$(\mathbf{I} + \theta\mathbf{T})\mathbf{a}^{n+1} = 2\mathbf{a}^n - \mathbf{a}^{n-1} - \mathbf{T}\left((1-2\theta)\mathbf{a}^n + \theta\mathbf{a}^{n-1}\right) - \mathbf{L}\mathbf{a}^n + \mathbf{F}^n, \qquad (7)$$

$$\mathbf{T} = \Delta t^2 \mathbf{M}_{\mu^{-1}} \mathbf{C} \mathbf{M}_{\varepsilon^{-1}} \mathbf{C}_1^T, \quad \mathbf{L} = \Delta t^2 \mathbf{M}_{\mu^{-1}} \mathbf{C} \mathbf{M}_{\varepsilon^{-1}} \mathbf{C}_2^T,$$

$$\mathbf{F}^n = -\left(\mathbf{I} + \theta\mathbf{T}\right)\mathbf{a}_0^{n+1} + 2\mathbf{a}_0^n - \mathbf{a}_0^{n-1} - \mathbf{T}\left((1-2\theta)\mathbf{a}_0^n + \theta\mathbf{a}_0^{n-1}\right) - \mathbf{L}\mathbf{a}_0^n,$$

$$\mathbf{a}^n = \int_{-\infty}^{t_n} \mathbf{h}_s d\tau, \qquad \mathbf{a}_0^n = \int_{-\infty}^{t_n} \mathbf{h}_0 d\tau,$$

where the vectors $\mathbf{h}_0$, $\mathbf{h}_s$ correspond to the fields $\mathbf{H}^0$, $\mathbf{H}^s$ in the representation (2). This scheme approximates the problem (4). The vector $\mathbf{F}^n$ approximates the boundary conditions.

It is easy to show [4] that all eigensolutions of the spatial discretization correspond to non-dissipative and non-growing oscillations with a real-valued circular frequency $\omega_j = \sqrt{\lambda_j}$ and the time-dependency $\mathbf{h}(t) \propto \text{Re}\{e^{i\omega_j t}\}$.

The next step in the stability analysis is the stability of the time-stepping scheme. A sufficient spectral stability condition in free space is

$$c\Delta t \le \Delta z, \quad 0.25 \le \theta. \qquad (8)$$

With the time step $c\Delta t = \Delta z$ allowed by condition (8) the scheme has no dispersion in the longitudinal direction and a moving mesh can be employed easily. The results in this case are fully equivalent to a stationary mesh, as no interpolation is necessary.

To reduce dispersion in the transversal direction we should use minimal value of θ.

4. Realization of the scheme for rotationally symmetric geometry with staircase approximation of the boundary

In this section we describe the realization of the above scheme for the case of a rotationally symmetric geometry with diagonal material matrices $\mathbf{M}_{\mu^{-1}}$, $\mathbf{M}_{\varepsilon^{-1}}$.

For a bunch moving offset a from and at speed of light c parallel to the axis of a rotationally symmetric structure, the source current $\mathbf{j}$ can be presented as

$$\mathbf{j} = \frac{c\lambda(z/c - t)\delta(r-a)}{\pi a} \sum_{m=0}^{\infty} \frac{\cos m\varphi}{1 + \delta_{m0}} \mathbf{z}$$

where $\lambda(s)$ is the longitudinal charge distribution and m is the azimuthal mode number.

The numerical scheme (7) for mode m has the form

$$-\mathbf{M}_{\mu_r^{-1}}^{-1}\Delta t^{-2}\left(\mathbf{a}_r^{n+1} - 2\mathbf{a}_r^n + \mathbf{a}_r^{n-1}\right) = \mathbf{P}_z\mathbf{M}_{\varepsilon_\varphi^{-1}}\mathbf{P}_z^T\mathbf{a}_r^n + \left(m^2\mathbf{M}_{\varepsilon_z^{-1}}\bar{\mathbf{a}}_r^n - \mathbf{P}_z\mathbf{M}_{\varepsilon_\varphi^{-1}}\mathbf{P}_r^T\bar{\mathbf{a}}_z^n + m\mathbf{M}_{\varepsilon_z^{-1}}\mathbf{P}_r^T\bar{\mathbf{a}}_\varphi^n\right) - \mathbf{F}_r^n, \quad (9)$$

$$-\mathbf{M}_{\mu_\varphi^{-1}}^{-1}\Delta t^{-2}\left(\mathbf{a}_\varphi^{n+1} - 2\mathbf{a}_\varphi^n + \mathbf{a}_\varphi^{n-1}\right) = \mathbf{P}_z\mathbf{M}_{\varepsilon_r^{-1}}\mathbf{P}_z^T\mathbf{a}_\varphi^n + \left(\mathbf{P}_r\mathbf{M}_{\varepsilon_z^{-1}}\mathbf{P}_r^T\bar{\mathbf{a}}_\varphi^n + m\mathbf{P}_z\mathbf{M}_{\varepsilon_z^{-1}}\bar{\mathbf{a}}_z^n + m\mathbf{P}_r\mathbf{M}_{\varepsilon_z^{-1}}\bar{\mathbf{a}}_r^n\right) - \mathbf{F}_\varphi^n,$$

$$-\mathbf{M}_{\mu_z^{-1}}^{-1}\Delta t^{-2}\left(\mathbf{a}_z^{n+1} - 2\mathbf{a}_z^n + \mathbf{a}_z^{n-1}\right) = m\mathbf{M}_{\varepsilon_r^{-1}}\mathbf{P}_z^T\mathbf{a}_\varphi^n - \mathbf{P}_r\mathbf{M}_{\varepsilon_\varphi^{-1}}\mathbf{P}_z^T\mathbf{a}_r^n + \left(\mathbf{P}_r\mathbf{M}_{\varepsilon_\varphi^{-1}}\mathbf{P}_r^T + m^2\mathbf{M}_{\varepsilon_r^{-1}}\right)\bar{\mathbf{a}}_z^n - \mathbf{F}_z^n,$$

where $\bar{\mathbf{a}}_p^n \equiv \theta\mathbf{a}_p^{n+1} + (1-2\theta)\mathbf{a}_p^n + \theta\mathbf{a}_p^{n-1}$, $p = r, \varphi, z$.

When a bunch moves along the axis, only the $\mathbf{A}_\varphi$ component of vector potential is different from zero and our scheme with $\theta = 0.5$ is reduced to the staircase scheme of the paper [6].

The material matrices $\mathbf{M}_{\mu^{-1}}$, $\mathbf{M}_{\varepsilon^{-1}}$ in staircase approximation of geometry are diagonal and it is possible to use the divergence relation

$$\mathbf{a}_\varphi = m^{-1}\mathbf{M}_{\mu_\varphi^{-1}}\left(\mathbf{P}_r\mathbf{M}_{\mu_r^{-1}}^{-1}(\mathbf{a}_r + \mathbf{a}_r^0) + \mathbf{P}_z\mathbf{M}_{\mu_z^{-1}}^{-1}(\mathbf{a}_z + \mathbf{a}_z^0)\right) - \mathbf{a}_\varphi^0 \qquad (10)$$

to eliminate the $\mathbf{a}_\varphi$ component from the equations for the $\mathbf{a}_r$, $\mathbf{a}_z$ component:

$$-\mathbf{M}_{\mu_r^{-1}}^{-1}\Delta t^{-2}\left(\mathbf{a}_r^{n+1} - 2\mathbf{a}_r^n + \mathbf{a}_r^{n-1}\right) = \mathbf{P}_z\mathbf{M}_{\varepsilon_\varphi^{-1}}\mathbf{P}_z^T\mathbf{a}_r^n +$$
$$+\left((\mathbf{M}_{\varepsilon_z^{-1}}\mathbf{P}_r^T\mathbf{M}_{\mu_\varphi^{-1}}\mathbf{P}_r\mathbf{M}_{\mu_r^{-1}}^{-1} + m^2\mathbf{M}_{\varepsilon^{-1}})\overline{\mathbf{a}}_r^n + (\mathbf{M}_{\varepsilon_z^{-1}}\mathbf{P}_r^T\mathbf{M}_{\mu_\varphi^{-1}}\mathbf{P}_z\mathbf{M}_{\mu_z^{-1}}^{-1} - \mathbf{P}_z\mathbf{M}_{\varepsilon_\varphi^{-1}}\mathbf{P}_r^T)\overline{\mathbf{a}}_z^n\right) - \hat{\mathbf{F}}_r^n, \qquad (11)$$
$$-\mathbf{M}_{\mu_z^{-1}}^{-1}\Delta t^{-2}\left(\mathbf{a}_z^{n+1} - 2\mathbf{a}_z^n + \mathbf{a}_z^{n-1}\right) = \mathbf{M}_{\varepsilon_r^{-1}}\mathbf{P}_z^T\mathbf{M}_{\mu_\varphi^{-1}}\mathbf{P}_z\mathbf{M}_{\mu_z^{-1}}^{-1}\mathbf{a}_z^n + (\mathbf{M}_{\varepsilon_r^{-1}}\mathbf{P}_z^T\mathbf{M}_{\mu_{\varphi}^{-1}}\mathbf{P}_r\mathbf{M}_{\mu_r^{-1}}^{-1} - \mathbf{P}_r\mathbf{M}_{\varepsilon_\varphi^{-1}}\mathbf{P}_z^T)\mathbf{a}_r^n +$$
$$+\left(\mathbf{P}_r\mathbf{M}_{\varepsilon_\varphi^{-1}}\mathbf{P}_r^T + m^2\mathbf{M}_{\varepsilon_r^{-1}}\right)\overline{\mathbf{a}}_z^n - \hat{\mathbf{F}}_z^n.$$

The system (10) does not contain static solutions and its dimension is reduced by 1/3 in its rank. It can be solved easily. At the first step we calculate the vector $\mathbf{a}_z^{n+1}$ and have to solve the linear system with the matrix $\mathbf{I} + \Delta t^2\theta\mathbf{M}_{\mu_z^{-1}}\mathbf{P}_r\mathbf{M}_{e_\varphi^{-1}}\mathbf{P}_r^T + m^2\Delta t^2\theta\mathbf{M}_{\mu_z^{-1}}\mathbf{M}_{\varepsilon_r^{-1}}$. This matrix is a block diagonal one with N_z blocks. Each block is a three-band matrix of size N_r and can be resolved by $O(N_r)$ operation. Now we can use the component $\mathbf{a}_z^{n+1}$ in the equation for the component $\mathbf{a}_r^{n+1}$ and have to solve the linear system with the matrix $\mathbf{I} + \Delta t^2\theta\mathbf{M}_{\mu_r^{-1}}\mathbf{M}_{\varepsilon_z^{-1}}\mathbf{P}_r^T\mathbf{M}_{\mu_\varphi^{-1}}\mathbf{P}_r\mathbf{M}_{\mu_r^{-1}}^{-1} + m^2\Delta t^2\theta\mathbf{M}_{\mu_r^{-1}}\mathbf{M}_{\varepsilon_z^{-1}}$. This matrix is again a block diagonal one with N_z blocks. Each block is also a three-band matrix of size N_r and can be resolved by $O(N_r)$ operation.

The above consideration shows that resolving implicitly of the system (11) demands only $O(N_r N_z)$ operations and the algorithm requires the same order of operations as the explicit FI-TD/FDTD method used in such codes as TBCI, MAFIA [5].

5. Conformal scheme

With the standard staircase approximation of curved boundaries we obtain only a first order convergent scheme in the L_2^h grid norm. To develop a second order convergent scheme we use the approach of the paper [4].

We allow the cells of the grid to be only partially filled by a PEC material with an arbitrarily shaped interface. Since the area of the cells near the boundary is reduced, the time step in the conformal scheme has to be reduced, too. To overcome this problem and to receive a stable algorithm without reducing the time step we modify only the original material matrix $\mathbf{M}_{\mu^{-1}}$ which is a composition of diagonal matrices: $\mathbf{M}_{\mu^{-1}} = \mathbf{RM}$, $\mathbf{M} = \left\|\mu_{pijk}^{-1}\right\|$, $\mathbf{R} = \left\|r_{pijk}^{-1}\right\|$, $\mu_{pijk} = \mu s_{pijk}/S_{pijk}$, $r_{pijk} = S_{pijk}/L_{pijk}$, where s denotes a reduced cell area [4]. A new material matrix $\mathbf{M}_{\mu^{-1}}$ is composed by the relation $\mathbf{M}_{\mu^{-1}} = \mathbf{V}^T\mathbf{D}\mathbf{V}$, where $\mathbf{D} = \mathbf{RU} > 0$ is a diagonal matrix, responsible for the order of the approximation, and $\mathbf{V}$ is a matrix of weights. In [4] we described building

of the matrix $\mathbf{M}_{\mu^{-1}}$ for explicit algorithm. Since the scheme is implicit in transversal direction we use weights only in longitudinal direction and only for facets in rz and φz planes.

To simplify the notation, we consider only one rz-plane and omit the index j for the φ-direction. At first we will build an auxiliary matrix $\mathbf{V}^0$:

$v^0_{\varphi ik,\varphi ik} = 1$ (⊞);

$v^0_{\varphi ik,\varphi i+1k} = \max(0, a - \mu_{\varphi ik}\varepsilon_{ri+1k})$ (⊞); $v^0_{\varphi ik,\varphi i-1k} = \max(0, a - \mu_{\varphi ik}\varepsilon_{rik})$ (⊞);

Here, a is a constant parameter with $0.5 \le a \le 1$. In all numerical examples shown in the next section the parameter a is equal to 0.99. It follows from the relations above that only cells near the boundary will give a contribution to non-diagonal elements of the matrix $\mathbf{V}^0$.

From the matrix $\mathbf{V}^0$ we will build a matrix $\mathbf{V}$:

$$v_{\varphi ik,\varphi i_1 k_1} = v^0_{\varphi ik,\varphi i_1 k_1} \Big/ \sum_{i_2 k_2} v^0_{\varphi i_2 k_2,\varphi i_1 k_1} \tag{12}$$

where the summa is taken over all elements of corresponding column.

The diagonal matrix $\mathbf{U}$ has elements

$$u_{\varphi ik} = \left(\sum_{i_1 k_1} v_{\varphi ik,zi_1 k_1} \mu_{\varphi i_1 k_1} \right)^{-1}, \tag{13}$$

where the summa is taken over all elements of the corresponding row.

A geometric interpretation of the above procedure is considered in [4].

Relations (12), (13) allows us to show that in the boundary cells we have at least first order local approximation error in the material relations. We consider again the situation when only the weight $v_{\varphi ik,\varphi i+1k}$ for one adjacent cell is unequal to zero,

$$\mathbf{M}_{\mu^{-1}} = \mathbf{V}^* \mathbf{R} \mathbf{U} \mathbf{V}, \tag{14}$$

$$\left(\mathbf{R}\mathbf{U}\mathbf{V}b\right)_{\varphi ik} = \frac{b_{\varphi ik} v_{\varphi ik,\varphi ik} + b_{\varphi i+1k} v_{\varphi ik,\varphi i+1k}}{\not{a}_{\varphi ik} v_{\varphi ik,\varphi ik} + \not{a}_{\varphi i+1k} v_{\varphi ik,\varphi i+1k}} = h_{\varphi ik} + O(h^2),$$

$$\left(\mathbf{M}_{\mu^{-1}}b\right)_{\varphi ik} = h_{\varphi ik} + O(h^2).$$

The last relation follows from the relation (12), which means that the summa of all elements of any row of the matrix $\mathbf{V}^*$ is equal to one. Thus, we have at least a first order local approximation error in the material relations near the boundary, and globally a second order convergent scheme in L_2^h grid norm if it is stable.

For spatial stability, it follows directly from (14) that the material matrix $\mathbf{M}_{\mu^{-1}}$ is symmetric and matrix $\mathbf{A} = c^2 \mathbf{M}_{\mu^{-1}} \mathbf{C} \mathbf{M}_{\varepsilon^{-1}} \mathbf{C}^T$ is positively semidefinite.

In our conformal scheme the material matrix $\mathbf{M}_{\mu_r^{-1}}$ is non-diagonal one. It makes the usage of divergence relation (10) too expensive. For this reason we will describe next a different approach based directly on the system (9).

At the first step we calculate the vector $\mathbf{a}_z^{n+1}$ and have to solve linear system with the matrix $\mathbf{I} + \Delta t^2 \theta \mathbf{M}_{\mu_z^{-1}} \mathbf{P}_r \mathbf{M}_{e_j^{-1}} \mathbf{P}_r^T + m^2 \Delta t^2 \theta \mathbf{M}_{\mu_z^{-1}} \mathbf{M}_{\varepsilon_r^{-1}}$. The same as in staircase approximation this matrix in the conformal scheme is a block diagonal one with N_z

blocks. Each block is a three-band matrix of size N_r and can be resolved by $O(N_r)$ operation.

Now we can use the component $\mathbf{a}_z^{n+1}$ in the equations for components $\mathbf{a}_r^{n+1}$, $\mathbf{a}_\varphi^{n+1}$ which are coupled. We have to solve a system with the matrix

$$\begin{pmatrix} \mathbf{I}+\Delta t^2\theta m^2\mathbf{M}_{\mu_r^{-1}}\mathbf{M}_{\varepsilon_z^{-1}} & m\theta\mathbf{M}_{\mu_r^{-1}}\mathbf{M}_{\varepsilon_z^{-1}}\mathbf{P}_r^T \\ m\theta\mathbf{M}_{\mu_\varphi^{-1}}\mathbf{P}_r\mathbf{M}_{\varepsilon_z^{-1}} & \mathbf{I}+\Delta t^2\theta\mathbf{M}_{\mu_\varphi^{-1}}\mathbf{P}_r\mathbf{M}_{\varepsilon_z^{-1}}\mathbf{P}_r^T \end{pmatrix}, \tag{15}$$

which is a block diagonal one with N_z blocks.
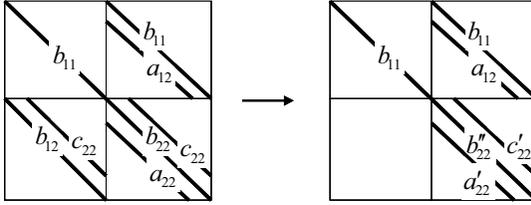
Figure 1. Reduction of the matrix

Now as the first step we consider an algorithm for the case of diagonal material matrices (as it has place in staircase approximation of geometry) and then we show how to use this algorithm for the conformal scheme with non-diagonal material matrix $\mathbf{M}_{\mu_r^{-1}}$. In the case of diagonal matrix $\mathbf{M}_{\mu_r^{-1}}$ each block of matrix (15) is of size $2N_r$ and has a seven-band structure as shown in Fig.1.

It can be reduced to a three diagonal type by $O(N_r)$ operation. This means its resolving takes only $O(N_rN_z)$ operations and the algorithm demands the same order of operations as the explicit FDTD method.

For the common case of non-diagonal matrix $\mathbf{M}_{\mu_r^{-1}}$ we use an iterative algorithm based on the splitting of the transversal operator $\mathbf{T}$. If we note by $\mathbf{M}_{\mu^{-1}}^0$ the diagonal part of the material matrix $\mathbf{M}_{\mu^{-1}}$, the iterative scheme reads

$$\left(\mathbf{I}+\theta\mathbf{T}^0\right)(\mathbf{a}^{n+1})^i = \mathbf{F}^n - \theta\mathbf{T}^1(\mathbf{a}^{n+1})^{i-1}, \; i=1,2,... \tag{16}$$

$$\mathbf{F}^n = -\mathbf{T}\left((1-2\theta)\mathbf{a}^n+\theta\mathbf{a}^{n-1}\right)+2\mathbf{a}^n - \mathbf{a}^{n-1} - \mathbf{L}\mathbf{a}^n + \mathbf{F}^n,$$

$$\mathbf{T}^0 = \Delta t^2\mathbf{M}_{\mu^{-1}}^0\mathbf{C}\mathbf{M}_{\varepsilon^{-1}}\mathbf{C}_1^T, \mathbf{T}^1 = \Delta t^2\left(\mathbf{M}_{\mu^{-1}}-\mathbf{M}_{\mu^{-1}}^0\right)\mathbf{C}\mathbf{M}_{\varepsilon^{-1}}\mathbf{C}_1^T,$$

where the operator $\mathbf{T}^1$ is of very low rang. As the start value of the unknown vector we use a solution of the system

$$\left(\mathbf{I}+\theta\mathbf{T}^0\right)(\mathbf{a}^{n+1})^0 = -\mathbf{T}^0\left((1-2\theta)\mathbf{a}^n+\theta\mathbf{a}^{n-1}\right)+2\mathbf{a}^n - \mathbf{a}^{n-1} - (\mathbf{T}^1+\mathbf{L})\mathbf{a}^n + \mathbf{F}^n. \tag{17}$$

The expression (17) the same as (7) is an approximation of the problem (4) of order $O(\Delta z^2 + \Delta t^2)$ but unstable one for $c\Delta t = \Delta z$. After solution of equation (17) it was enough to do only one iteration (16) to receive a stable algorithm for our numerical examples. For both equations (16), (17) we have to invert the matrix $\mathbf{I}+\theta\mathbf{T}^0$ and

consequently the described above algorithm for seven-band matrix can be applied.

In all examples shown in the next section we used the scheme (16), (17) with moving mesh.

6. Indirect method for calculating of wake potential

To calculate the wake potential W^m we used a modification of the indirect method described in [8].
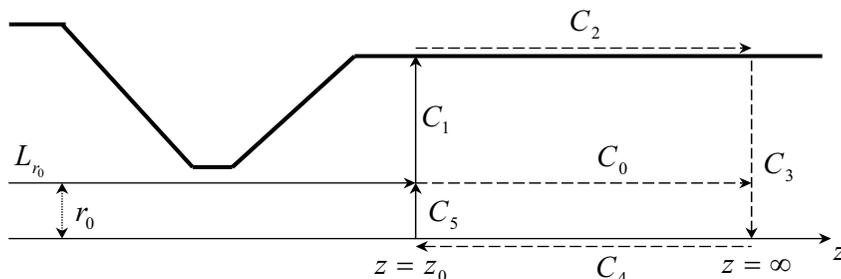
Figure 2. Contours for the indirect integration.

The main feature of our method is that (the same as in direct method) we integrate only $\overline{e}_z^s$ component of electromagnetic field along a straight line L_{r_0} at radius r_0 and use other field components only at the end of the structure. Note, that in original method [8] along the straight line L_{r_0} we have to integrate a linear combination of $\overline{e}_z^s$ and $\overline{b}_z^s$ components.

As it was shown in [8], for each mode m the differential forms

$$\omega_S = r^m [\overline{e}_r^s + c\overline{b}_\varphi^s - \overline{e}_\varphi^s + c\overline{b}_r^s] dr + r^m [\overline{e}_z^s + c\overline{b}_z^s] dz ,$$

$$\omega_D = r^{-m} [\overline{e}_r^s + c\overline{b}_\varphi^s + \overline{e}_\varphi^s - c\overline{b}_r^s] dr + r^{-m} [\overline{e}_z^s - c\overline{b}_z^s] dz ,$$

are closed. Hence, we can write

$$QW^m = \int_{-\infty}^{\infty} \overline{e}_z^s dz = \int_{L_{r_0}} \overline{e}_z^s dz + \int_{C_0} \overline{e}_z^s dz ,$$

$$\int_{C_0} \overline{e}_z^s dz = -\frac{1}{2} \left(\int_{C_0} \left(r_0^m \omega_D + r_0^{-m} \omega_S \right) - \frac{\beta}{a^m} \int_{C_{15}} \omega_S \right), \quad \beta = \left(\frac{a}{r^0} \right)^m - \left(\frac{a}{r^0} \right)^{-m}, \quad C_{15} = \sum_{i=1}^{5} C_i ,$$

where a is a radius of outgoing tube and other parameters are shown in Fig.2. For perfectly conducting geometry from the above relations it is easy to obtain

$$\int_{C_0} \overline{e}_z^s dz = -\frac{1}{2} \left(\int_{C_1} \left(r_0^m \omega_D + r_0^{-m} \omega_S - \frac{\beta}{a^m} \omega_S \right) - \frac{\beta}{a^m} \int_{C_5} \omega_S \right) =$$

$$= -\frac{1}{2} \int_0^{r_0} \beta (r/a)^m [\overline{e}_r^s + c\overline{b}_\varphi^s - \overline{e}_\varphi^s + c\overline{b}_r^s] dr - \frac{1}{2} \int_{r_0}^{a} \left((r_0/r)^m + (r_0/r)^{-m} - \beta (r/a)^m \right) [\overline{e}_r^s + c\overline{b}_\varphi^s] dr -$$

$$- \frac{1}{2} \int_{r_0}^{a} \left((r_0/r)^m - (r_0/r)^{-m} + \beta (r/a)^m \right) [\overline{e}_\varphi^s - c\overline{b}_r^s] dr.$$

As we see the infinite contour C_0 can be replaced by finite contours C_1, C_5. The transversal wake potential $W_\perp^m$ can be found from longitudinal one by applying the Panowsky-Wenzel theorem[9].

7. Numerical examples

The conformal scheme (16) is included in a newly developed computer code. Numerical examples for monopole wake $m = 0$ were presented in [10]. In this section we will show results for the dipole wake $m = 1$.
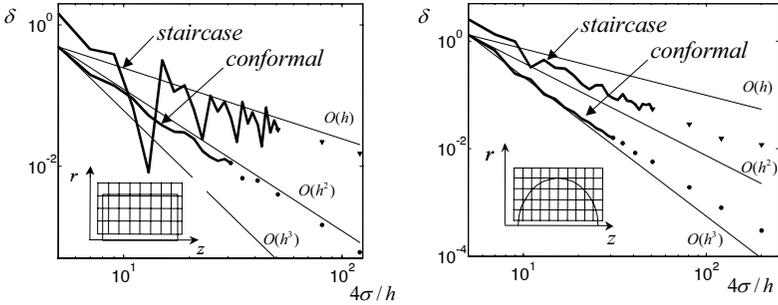
Figure 3. Realtive error of dipole loss factor L^1 for a pillbox and a sphere vs. number of mesh steps per RMS bunch length (Gaussian charge distribution)

Fig.3 shows the relative error of the loss factor $\delta = \left| L^1_{calc} - L^1 \right| / L^1$ for a Gaussian bunch with $\sigma = 0.25$ cm passing through a pillbox (Fig.3 left) and a spherical resonator (Fig.3 right). The pillbox has the length 1.8 cm and radius 0.9 cm. The sphere has the diameter 1.8 cm. Offset of the bunch from the axis is equal to 2σ for both geometries. The analytical loss factor L^1 is equal to 0.7589 V/pC for the pillbox. For the sphere we use an extrapolated loss factor equal to 0.3752 V/pC. The error for stationary mesh is demonstrated by lines, the results for a moving mesh are shown by triangles and circles.
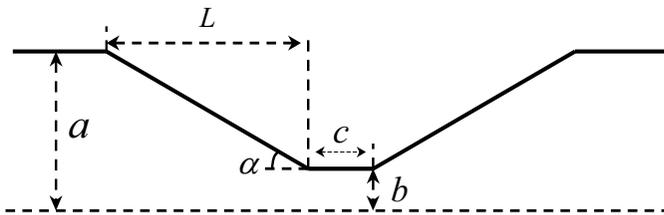
Figure 4. Geometry of the collimator.

Fig. 4 shows the geometry of the collimator. In our numerical examples we use $a = 35mm$, $b = c = 2mm$. In Fig. 5(left) the transversal dipole loss factor $L^1_\perp$ for the collimator with $L = 10cm$ is shown for different mesh resolutions σ/h, where $\sigma = 1mm$ for the Gaussian bunch and h is the mesh step. The error compared to reference value is shown on the figure too. The blue line shows results for the new scheme and the black one for ABCI 9.2.1 code [7]. In Fig. 5(right) the transversal dipole loss factor $L^1_\perp$ for the collimator with $L = 20cm$ is shown.
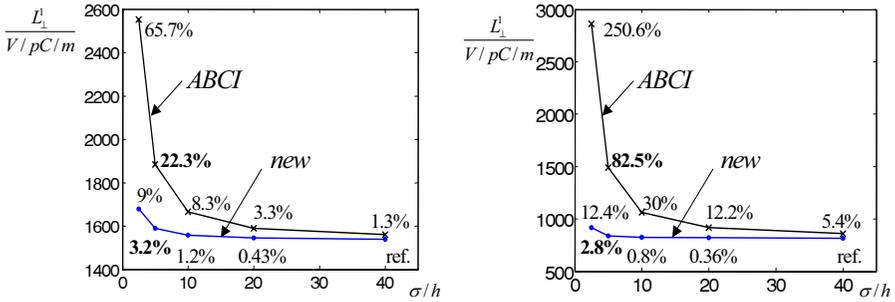
Figure 5. Transversal dipole loss factor $L_\perp^1$ for the collimators with
$L = 10cm$ (left) and $L = 20cm$ (right)
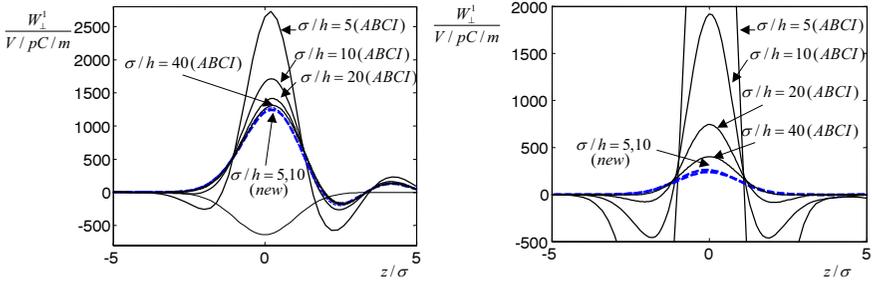
Figure 6. Transversal dipole wake function for the collimators with
$L = 20cm$ (left) and $L = 50cm$ (right)

Fig. 6 shows the transversal dipole wake potential $W_\perp^1(s)$ for collimators with $L = 20cm$ and $L = 50cm$. The black curves show results for ABCI code and the blue ones show results for the new scheme.
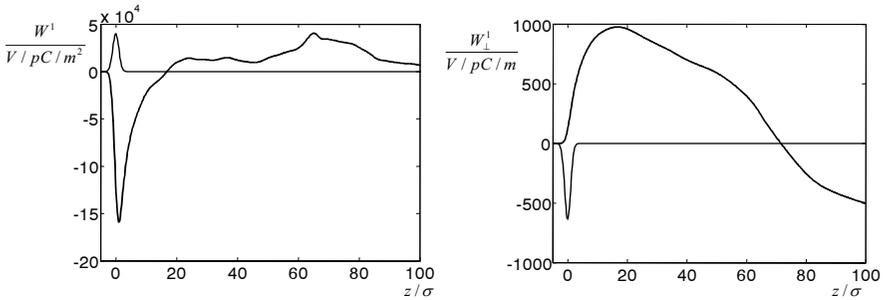
Figure 7. Dipole wakes for TESLA cryomodule.

From the above examples we see that the error for the new code remains on the same level (for example, it is about 3% for $\sigma / h = 5$) independently from the length of collimator. The reference code ABCI demands much more dense mesh for the same accuracy, which strongly depends on the collimator length. In the last example even with 40 points on σ the error is in excess of 100% for ABCI code.
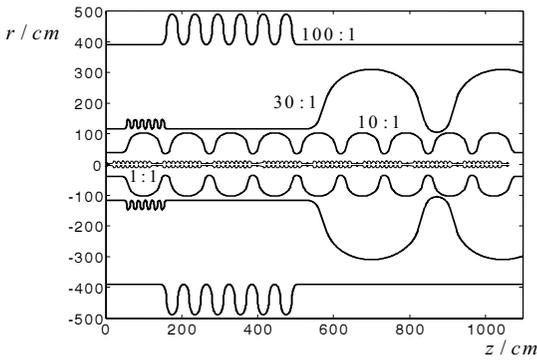
Figure 8. Geometry of TESLA cryomodule.

Finally, we show in Fig.7 the dipole wake potentials of Gaussian bunch with $\sigma = 1\,mm$ for TESLA cryomodule [11] containing eight cavities and nine bellows as shown in Fig. 8.

References

[1] Weiland T. "On the Numerical Solution of Maxwell's Equations and Applications in Accelerator Physics" 1984 *Particle Accelerators* **15**: 245-291.
[2] Balakin V, Novokhatski A. "VLEPP: Longitudinal Beam Dynamics" Proc. of the 12-th Int. Conference on High-Energy Accelerators, Fermilab, August 1983.
[3] Meincke O, Wagner A, Zotter B. "New Wake Field and Bunch Lengthening Codes" SL-Note-97-17(AP), CERN, 1997.
[4] Zagorodnov I, Schuhmann R, Weiland T. "A Uniformly Stable Conformal FDTD-Method on Cartesian Grids" *Int. J. Numer. Modelling*, to appear.
[5] Weiland T. "Time Domain Electromagnetic Field Computation with Finite Difference Methods" 1996 *Int. J. Numer. Modelling* **9:** 295-319.
[6] Novokhatski A, Timm M, Weiland T. "Transition Dynamics of the Wake Fields of Ultra Short Bunches" Proc. of the ICAP 1998, Monterey, California, USA.
[7] Chin YH. "User's guide for ABCI Version 8.7" CERN-SL-94-02-AP, 1994.
[8] Napoly O, Chin Y, Zotter B. "A Generalized Method for Calculating Wake Potentials" 1993 *Num. Instrum. Meth* **344**:255.
[9] Panofsky WKH, Wenzel WA. "Some Considerations Concerning the Transverse Deflection of Charged Particles in Radio-Frequency Fields" 1956 *Rev. Sci. Instrum* **27**:947.
[10] Zagorodnov I, Weiland T. "A Conformal Scheme for Wake Field Calculation" Proc. of the EPAC 2002, Paris, France.
[11] TESLA Technical Design Report. Part II, DESY 2001-011, Hamburg, Germany.

Author Index