

LIONS_LINAC : A new particle in cell code for LINACS

P. Bertrand

► **To cite this version:**

P. Bertrand. LIONS_LINAC : A new particle in cell code for LINACS. 2003 Particle Accelerator Conference (PAC 2003), May 2003, Portland, United States. pp.3488-3490, 10.1109/PAC.2003.1289957. in2p3-00437665

HAL Id: in2p3-00437665

<http://hal.in2p3.fr/in2p3-00437665>

Submitted on 1 Dec 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

LIONS_LINAC : A NEW PARTICLE IN CELL CODE FOR LINACS

P. Bertrand[#], GANIL, Caen, France

Abstract

In the frame of the SPIRAL II project at GANIL, a new particle-in-cell code called LIONS_LINAC has been developed. It is written in pure Fortran 95, and can work either on PCs and workstations or on vectorial/parallel computers. We present the main characteristics of the code, in particular the treatment of the space charge effects, using the multigrid method with conjugate gradient smoothing iterations.

INTRODUCTION

A detailed study on a new facility at GANIL for the production of high intensity exotic beams is underway. The SPIRAL II project, based on the LINAG I preliminary conceptual design [1], and using the Uranium target fission method, requires the acceleration of q/A=1/3 ions (1mA) and Deuterons (5mA) at 40Mev. It will consist of 2 dedicated ECR sources, a warm RFQ, and a linac composed of independently phased superconducting QWR and/or HWR cavities [2][3]. Accurate beam dynamics calculations using various codes are needed to optimise the linac and minimize the losses. In this context, we have developed a new code, LIONS_LINAC, using very precise 3D electromagnetic maps, able to solve huge space charge problems and study the halo dynamics.

LANGUAGE AND IMPLEMENTATION

We have chosen the Fortran 95 language and optional HPF directives, without any specific library, in order to ensure perfect portability and high efficiency on vectorial or parallel computers. In particular this allows one to use intensively the array notation, and the main object-oriented features (dynamic allocation, recursivity, derived type, etc), which makes the program easy to maintain and improve, and facilitates the multigrid method of coding.

INPUT FILES AND LINAC DESIGN

The code uses 2 main ASCII input-files : the first gives the ion characteristics (q, m, I, initial conditions...) and the algorithmic options (discretization scheme, time step, space charge solver...). The second file gives the list of elements (one per line) describing the MEBT and the linac (drifts, quadrupoles, solenoids, cavities...). The elements are included in boxes with extra-drifts so that it is easy to replace analytical fields by realistic field maps.

INITIAL CONDITIONS

The initial beam characteristics can be introduced in two ways: by giving the Twiss parameters in the different planes and generating a 6D uniform particle distribution, or by reading a portable binary file coming for example

from realistic output results given by an RFQ TOUTATIS computation [4]. This allows us to compare the behaviour of LIONS_LINAC with that of TRACEWIN [4].

DISCRETISATION IN TIME

In dealing with space charge effects, we have chosen the time as the evolution variable [5], and (x, P_x, y, P_y, z, P_z) as the phase space variable, where P is the generalised momentum. Leap-Frog and RK4 schemes are both implemented, which allows us to know the influence of the discretisation algorithm on the precision obtained.

ELECTROMAGNETIC FIELDS

Solenoids and incorporated steerers

Each solenoid can be chosen to be hardedge of analytical, with optional (x,y) steerers. In the hardedge case, the time step must be chosen small enough to avoid artificial emittance growth. In the analytical case, the solenoid is composed of a main coil and inverse correction coils on each side, following the suggestion in [6]. Using a 3rd order development of the 3 fields, we can chose the length, radius and B_z correction values with respect to the main coil, in order to minimize the resulting fringe field.

Cavities

The cavities can be QWR or HWR, and we use analytical or 3D (E,B) realistic field maps coming from SOPRANO [7]. In the case of QWR resonators, the vertical steering, due mainly to the B_x component, can be an issue and must be studied carefully. Knowing that the $B_x(z)$ function is generally close to that of $E_z(z)$, within a multiplicative constant, we introduce the parameter:

$$\delta = \frac{B_x}{E_z}$$

This also allows us to define useful "double-sinus" analytical fields including the magnetic steering component, and defined as follows:

$$V = \frac{9}{16} V_{stem} (\cos(kz) - \frac{1}{9} \cos(3kz) + \frac{8}{9}) \sin(\omega t + \varphi)$$

$$E_z = \frac{9}{16} k V_{stem} (\sin(kz) - \frac{1}{3} \sin(3kz)) \sin(\omega t + \varphi) + \dots$$

$$E_r = -r \frac{9}{32} k^2 V_{stem} (\cos(kz) - \cos(3kz)) \sin(\omega t + \varphi) + \dots$$

$$B_x = \delta E_z \cot g(\omega t + \varphi) \quad ; \quad \omega = 2\pi f_{rf} \quad ; \quad k = \frac{2\pi}{\beta_0 \lambda}$$

Integrating this formulae along the cavity, we can obtain analytical expressions of the Transit Time Factor, the energy gain, the magnetic steering angle, the emittance

[#]bertrand@ganil.fr

growth induced by such a steering and the negative cavity shift Δy needed for a good compensation:

$$u = \frac{\beta}{\beta_0} ; T(u) = \frac{16 \sin(\pi/u) u^4}{\pi(1-u^2)(1-9u^2)} ; T(1)=1$$

$$\Delta W = \frac{9\pi}{16} q V_{stem} T(u) \cos(\varphi)$$

$$\frac{\Delta p_y}{p_{mag_steering}} = \frac{9\pi}{16} \frac{q V_{stem}}{m \beta c} \delta T(u) \sin(\varphi)$$

$$\frac{\Delta p_y}{p_{cavity_shift}} = \frac{9\pi}{32} \frac{q V_{stem}}{m \beta^2 c^2} k \frac{T(u)}{u} \sin(\varphi) \Delta y$$

$$\frac{\mathcal{E}_{cavity_exit}}{\mathcal{E}_{cavity_entrance}} = 1 + \frac{1}{2} \left(\frac{9\pi}{16} \frac{q V_{stem}}{m \beta c} \delta T(u) \cos(\varphi) \frac{\Delta \varphi_{max}}{\Delta y_{max}} \right)^2$$

These calculations are very useful to compare the motion evolution in analytical or realistic fields, and understand the origin of emittance growths and anomalies.

SPACE CHARGE EFFECTS

The space charge force can be computed using analytical formulae or a 3D multigrid Poisson solver. The first method allows us to tune rapidly the various parameters using a limited number of particles; the second produces realistic results in the case of a mismatched initial beam and misalignments of elements.

Analytical method

In this case, we use an initial particle distribution which is *uniform in the 6D phase space*, and consider this property maintained along the acceleration. At a given time t , we calculate the second order space momenta and RMS values, which multiplied by $\sqrt{8}$ give a realistic evaluation of the marginal ellipsoidal envelopes. Then we apply to each particle the analytical fields corresponding to an ellipsoidal *3D uniform distribution* whose RMS parameters are the same: the internal and external analytical fields are of course necessary (fig (1)).

Multigrid solver

The multigrid method is a very powerful tool to accelerate the convergence of a 3D Poisson solver [8]. However some precautions have to be taken in order to obtain good results on a vectorial/parallel computer. We have chosen the Conjugate Gradient algorithm as the pre- and post-smoothers. It is highly parallel even if we put Dirichlet boundary conditions inside the 3D computing box (e.g. a tube). The restriction and prolongation use 27 points. The grids are nested in the natural way using the finite-difference scheme. The object-oriented data structures are defined in terms of user-defined types with chained lists, using pointers. The multigrid routine itself is recursive. All the parameters defining the nested meshes can be chosen in the initial data-file, and the different 3D arrays are dynamically

allocatable. In order to avoid CPU and memory consuming duplications when passing parameters to the specific calculation routines (conjugate gradient, restriction, prolongation), just the address of the huge arrays $(A(0,0,0) \dots)$ are transmitted.

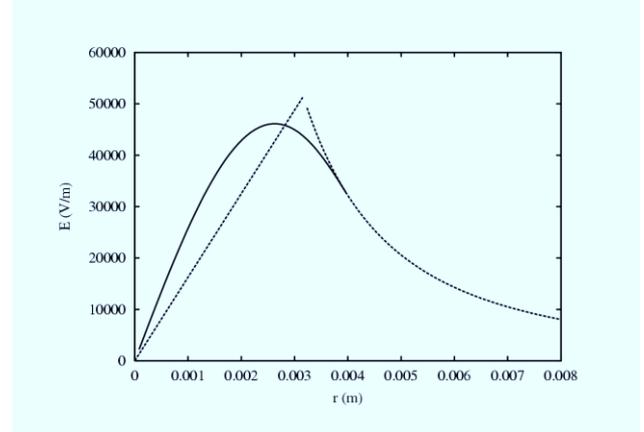


Figure 1: Theoretical electric fields for in a uniform 6D distribution and a uniform 3D one having the same RMS value, the space envelope being a sphere.

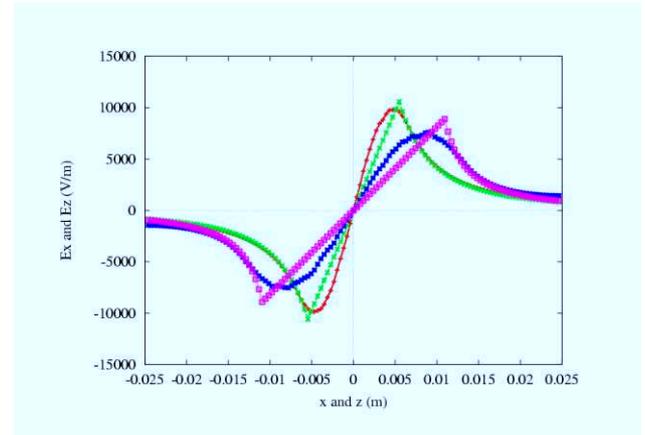


Figure 2: Analytical and Poisson electric fields deduced from a uniform 6D distribution of 200000 particles with a longitudinal length greater than the transverse size, using a 127x127x127 mesh.

For a bunch size of 1 cm in each direction, we choose a typical box of 3x3x3 cm, with about 2 million unknowns on the finer mesh, corresponding to a space step of 0.236 mm. With one CPU on a vectorial FUJITSU VPP-5000 computer, we obtain 0.3 second for 20 complete V-cycle multigrid iterations with a rate of 2.5 Gigaflops/s.

MISALIGNMENT TESTS

The misalignment of the cavities and the solenoids can be a problem in a linac. We can study these effects by applying to each element small misalignments in x and y and Euler angles in the case of solenoids. A misalignment-random generator for such misalignments will be added soon.

TUNING AND PHASE ADVANCE

An optional ASCII input-file can be read to activate the optimisation of the linac parameters. The linac is divided into successive sections. A Newton's method tunes the B_z value of each solenoid, according to a chosen envelope radius. A similar philosophy will be used for the optimisation of the cavity voltage and synchronous phase.

In order to evaluate the phase advances for each section we proceed as follows:

Once the *initial* bunch of n particles is randomly generated, and the beam matrix σ calculated with the 2nd momenta, we determine the subset of \tilde{n} particles corresponding to the beam core, having in mind the evaluation of the 1st order transfer matrix. The complete tracking being achieved for a given section, let P_0 and P the input and output $(6, \tilde{n})$ matrices giving the $(\Delta x, \Delta x', \Delta y, \Delta y', \Delta z, \Delta z')$ phase space positions. Then we calculate the $(6,6)$ transfer matrix T , which is numerically close to the symplecticity:

$$P = T P_0 \quad ; \quad \frac{P P_0^t}{\tilde{n}} = T \frac{P_0 P_0^t}{\tilde{n}} = T \tilde{\sigma}_0 \quad ; \quad T = \frac{P P_0^t}{\tilde{n}} \tilde{\sigma}_0^{-1}$$

The transverse and longitudinal planes being decoupled, the longitudinal phase advance is easy to deduce from T by finding the Twiss parameters. The situation is more complex in the transverse planes due to the coupling induced by the solenoids. Following [9], we decouple the (x, x') and (y, y') planes by a change of basis. This leads in fact to the determination of 2 *characteristic transverse phase advances*.

OUTPUT FILES

LIONS_LINAC generates a set of ASCII files compatible with the GNUPLOT package and giving various quantities as a function of the z value: particular (x, y, z) trajectory with electromagnetic applied components, beam envelopes and emittances, maximum spatial extensions, maximum space charge forces, etc. In order to compare the results with TRACEWIN calculations, LIONS_LINAC also generates binary files compatible with the PLOTWIN viewer (fig.3).

CONCLUSION

We have developed a new Particle In Cell code for linacs having in mind 2 main issues: the *emittance growth* and the *space charge* effects. The emittance growth may have various origins : longitudinal halo from the RFQ, random generation and number of particles, precision of the finite-element mesh in (E, B) field maps, time step, discretization method, non-linearities due to MEBT bunchers, $\Delta z \Delta x$ and $\Delta z \Delta y$ second-order effects through cavities, magnetic steering, misalignments, resonance coupling, space charge, etc. The space charge itself is very CPU and memory consuming when the halo

evolution has to be correctly simulated, which is the case in superconducting accelerators. The numerous options and the vectorial and parallel implementation of LIONS_LINAC will help to simulate all these effects in the best conditions.

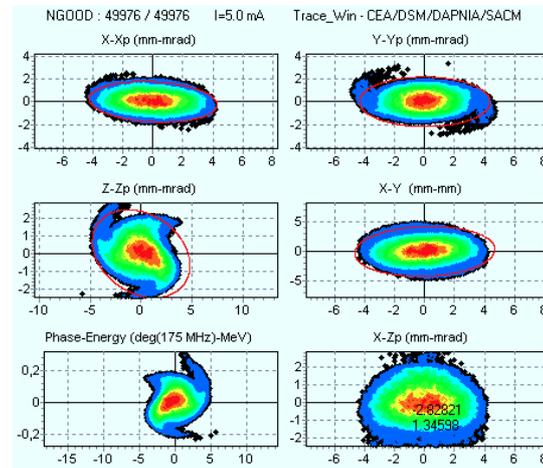


Figure 3 : Example of figures obtained at the linac exit.

ACKNOWLEDGMENTS

We would like to thank the student Yvan Gauthier, for its participation in the coding of the multigrid solver.

REFERENCES

- [1] http://www.ganil.fr/research/sp/reports/files/LINAG_PhaseI.pdf
- [2] A. Mosnier, SPIRAL II : "A High Intensity Deuteron and Ion Linear Accelerator for Exotic Beam Production", PAC2003, Portland, May 2003.
- [3] R. Duperrier et Al. , "Beam Dynamics Studies in SPIRAL II Linac" , PAC2003, Portland, May 2003.
- [4] R. Duperrier, N. Pichoff, D. Uriot, "CEA Saclay codes Review for High Intensity Linacs Computations" , ICCS2002, Amsterdam.
- [5] J. Qiang, R.D. Ryne, "Systematic comparison of position & time dependant macroparticle simulations in beam dynamics studies", Physical Review special topics, Accelerators and Beams, Vol .5. (2002).
- [6] P.N. Ostroumov, K.W. Shepard, S.H. Kim, E.S. Lessner, R. Laxdal, R. Wheatley, "A New Generation of Superconducting Solenoids for Heavy-Ion Linac Application", LINAC2003, Gyeongju.
- [7] OPERA-3D, Vector Fields. Ltd.
- [8] W. Hackbusch, "Multigrid Methods & Applications", Springer-Verlag, (1985).
- [9] D.A. Edwards and L.C. Teng, "Parametrization of linear coupled motion in periodic systems", IEEE Trans. Nucl. Sci 20, 3, pp 885-889 (1973).