Linac Code Benchmarking in Preparation of the UNILAC experiment

A. Franchi, W. Bayer, G. Franchetti, L. Groening, I. Hofmann, A. Orzhekhovskaya, S. Yaramyshev, X. Yin
GSI, Darmstadt, Germany

A. Sauer, R. Tiede, G. Clemente
IAP, Frankfurt am Main, Germany

R. Duperrier, D. Uriot
CEA, Saclay, France

G. Bellodi, F. Gerigk, A. Lombardi, T. Mütze
CERN, Geneva, Switzerland

Abstract

In the framework of the European network HIPPI (High Intensity Pulsed Proton Injectors) a linac code comparison and benchmarking program have been promoted. An intermediate goal is to compare different space-charge solvers and lattice modelling implemented in each code in preparation of experimental validations from future measurements to be carried out at the UNILAC of GSI. In the last two years a series of different tests and comparisons among several codes (DYNAMION, HALODYN, IMPACT, LORASR, PARMILA, PARTRAN, PATH and TOUTATIS) have been undertaken. The quality of Poisson solvers has been evaluated and a number of code adjustments has been carried out to obtain the best agreement in terms of RMS moments. In this note we report on the status of this program.

LINAC CODE BENCHMARKING IN PREPARATION OF THE UNILAC EXPERIMENT

A. Franchi, W. Bayer, G. Franchetti, L. Groening, I. Hofmann, A. Orzhekhovskaya, S. Yaramyshev, X. Yin
GSI, Darmstadt, Germany

A. Sauer, R. Tiede, G. Clemente
IAP, Frankfurt am Main, Germany

R. Duperrier, D. Uriot
CEA, Saclay, France

G. Bellodi, F. Gerigk, A. Lombardi, T. Mütze
CERN, Geneva, Switzerland

Abstract
In the framework of the European network HIPPI (High Intensity Pulsed Proton Injectors) a linac code comparison and benchmarking program have been promoted. An intermediate goal is to compare different space-charge solvers and lattice modelling implemented in each code in preparation of experimental validations from future measurements to be carried out at the UNILAC of GSI. In the last two years a series of different tests and comparisons among several codes (DYNAMION, HALODYN, IMPACT, LORASR, PARMILA, PARTRAN, PATH and TOUTATIS) have been undertaken. The quality of Poisson solvers has been evaluated and a number of code adjustments has been carried out to obtain the best agreement in terms of RMS moments. In this note we report on the status of this program.

1 Introduction

One of the main tasks of the beam dynamics working package of the European network "High Intensity Pulsed Proton Injector" (HIPPI) is the comparison and validation of 3D linac codes in the high current regime. Several codes are available and currently run for such simulations. The Alvarez DTL section of UNILAC (five tanks, $L \approx 55$ m) is used as reference lattice, as a dedicated machine experiment will be carried out in order to measure the three phase space projections (x-x'), (y-y') and ($\delta\varphi$-$\delta w/w$) at both ends of the section under various space-charge and mismatch conditions. The initial measured phase space projections will be used to generate the input particle distributions to be tracked using the codes. The final measured phase space projections will be then compared with the numerical predictions.
Different space-charge and lattice modelling may pose severe problems in understanding the source of discrepancies, when tracking simulations at high current and in presence of mismatch are run. For this reason, in preparation of the experimental validation, the code benchmarking has been divided in three steps.
The first is a static benchmarking of the space-charge routines: common ensembles of particles are given in input to the different space-charge solvers; the resulting space-charge electric fields are then compared with the analytical solutions against different numerical parameters and boundary conditions (for PIC codes). To investigate the effects of numerical errors on the single particle dynamics, the single particle depressed tune is inferred using the electric fields previously calculated, and is compared again with an analytical solution. Both tests require modifications in the source codes (that usually do not print out the space-charge electric field) and have been performed on codes with source code available only.

The second step consists of tracking simulations with a zero-current beam and a common input distribution. Scope of this test is twofold: first, the preparation of the input files for all the codes, checking carefully that they describe the same structure; second, the understanding of discrepancies arising from the different representation of physical elements implemented in the codes, especially for the RF.
In the last step, tracking simulations are run under the same conditions of the experiments planned for the end of 2006 and the results are compared among the codes. Here the scope is to investigate how space charge and nonlinear RF effects couple in the codes and to establish the most suitable numerical parameters to be used when simulating the experiment conditions.

This note is organized as follow. In Sec. 3 the codes that we have been running are briefly described and the main features are compared. In Sec. 4 the static benchmarking is described in detail together with the definitions of the quality factors chosen to quantify the goodness of the space-charge solvers. In Sec. 5 the comparison of tracking simulations is presented: the lattice modelling used for each code is discussed together with the main results and problems encountered when comparing the RMS emittances.
Several bugs (mainly related to the charge state $Z \neq 1$) have been found during the benchmarking. Despite the fact that they have been fixed, we report on them, as well as on the problems we encountered, believing that this can be of help when including other codes in the benchmarking. All the known problems that have been not yet fixed are also listed and discussed. Most of the contents here included, together with all the input files used to run the codes, is available in the HIPPI code benchmarking web page [1].

2 Organization

- **Coordinator**: I. Hofmann (GSI)
- **GSI**
    1. Tools for benchmarking the space-charge solvers (G. Franchetti, A. Orzhekhovskaya, A. Franchi)
    2. UNILAC modeling (L. Groening, W. Barth, W. Bayer, S. Yaramyshev)
    3. run IMPACT, HALODYN and PATH (A. Franchi)
    4. run DYNAMION (S. Yaramyshev, W. Bayer)
    5. run PARMILA and TRACE-3D (X. Yin)
    6. Alvarez DTL matching (L. Groening, W. Bayer, X. Yin)
    7. collecting material and web page editing (A. Franchi)
- **IAP**
    1. update and run LORASR (R. Tiede. G. Clemente, J. Dietrich)
    2. run SUPERFISH for RF UNILAC modeling of (A. Sauer)
    3. run PARMILA (A. Sauer)
    4. help running TRACE-3D (A. Sauer)
- **CEA**
    1. run TOUTATIS for Poisson solver test (R. Duperier)
    2. run PARTRAN (D. Uriot)
    3. improving the UNILAC modeling (D. Uriot)
- **CERN**
    1. update and support running PATH (T. Mütze, A. Lombardi, G. Bellodi)
    2. support running IMPACT (F. Gerigk)
- **External Support**
    1. J. Qiang (LBNL): support running IMPACT
    2. J. Billen (LANL), H. Takeda : update and support running PARMILA
    3. S. Rambaldi, G. Turchetti (Bologna Univ.): update HALODYN

3 The codes

In this section a review of the main features of the codes involved in the benchmarking is given. Particles are tracked in the 6D space, whereas the space-charge solver is 2D *r-z* or 3D depending on the code. Most of the solvers have PIC algorithm implemented, the charge distribution being deposited onto a grid and the Poisson equation solved on the grid. The space-charge electric field at any position is then computed via interpolation. A brief description of each code follows (in alphabetic order).

- **DYNAMION** [2] is a scalar code developed in ITEP Moskow and GSI Darmstadt. The space-charge routine is a 3D particle-particle integrator with a hard-sphere cut-off, introduced to avoid artificial short-range collisions. The RF description is based on the expansion of the RF voltage, whose coefficients are computed in pre-processing (solving the Laplace equation with boundary conditions defined by the 3D DTL geometry) [3].
- **HALODYN** [4] is a parallel code developed in the University of Bologna. The space-charge routine is based on a scalar 3D PIC spectral Poisson solver (FFT with inversion of a linear system) with closed boundary conditions defined on a rectangular pipe [5]. The RF is modelled using the thin lens approximation and an expansion in terms of Bessel functions.
- **IMPACT** [6] is a parallel code developed in Los Alamos (LANL) and Berkeley (LBNL). The space-charge routine is based on a parallel 3D PIC spectral Poisson solver (Green function with convolution) with several boundary conditions (open, closed and periodic) on both rectangular and elliptical pipes. The RF description is inferred from the on-axis electric field with either a linear or a nonlinear Lorentz integrator.
- **LORASR** [7] is a scalar code with GUI developed in IAP, J.W. Goethe University, Frankfurt am Main. The space-charge routine is based on a scalar 3D PIC Poisson spectral solver (FFT with inversion of a linear system) with closed boundary conditions defined on a rectangular pipe. The RF description is inferred from the radial (on- and off-axis) electric field.
- **PARMILA** [8] is a scalar code developed in Los Alamos (LANL). The user can choose either a 2D *r-z* (SCHEFF) or a 3D (PICNIC) PIC Poisson solver with open boundary conditions. The RF is modelled making use of either the transit-time-factor (TTF) table generated by SUPERFISH or a nonlinear thin kick.
- **PARTRAN** [9] is a scalar code developed in CEA, Saclay. The space-charge routine is a 3D PIC (PICNIC [10]) Poisson solver with open boundary conditions. The RF description is inferred by either importing an electro-magnetic field map (1D, 2D or 3D) or using a nonlinear thin kick.
- **PATH** [11] is a scalar code with GUI developed in CERN. The user can choose either a 2D *r-z* (SCHEFF) Poisson solver with open boundary conditions or a 3D particle-particle integrator. The RF description is inferred by either importing a 3D electro-magnetic field map or using a nonlinear thin kick.
- **TOUTATIS** [12] is a scalar code for RFQ developed in CEA, Saclay. The space-charge routine is a 3D PIC multi-grid Poisson solver with either open or periodic boundary conditions on an arbitrary geometry. The RF is modelled making use of a 3D electromagnetic field map.

In Tab.1 the main general features are listed together with an indication of the requested CPU time. The latter one has been obtained running the codes on the same Linux node and a hardware-equivalent Windows PC with the number of macro-particles indicated in the fifth column. The parallel codes have been run using one CPU only: while the CPU time for the fully parallel code IMPACT scales with the number of CPUs, the partially-parallel code HALODYN distributes over all the available nodes the tracking only, being the Poisson

solver serial and run by the master node only. The grid resolution of PIC codes is of 64^3 for IMPACT, 64^2x256 for HALODYN, 20x40 for the 2D solver SCHEFF (PARMILA and PATH) and 48^3 for the 3D solver PICNIC (PARMILA and PARTRAN). The number of space-charge calculations per DTL cell can be chosen by the user not in all the codes: it varies from 200 of DYNAMION, 80 for IMPACT, ≈60 for HALODYN, 40 for PATH, ≈15 for LORASR, 3 in PARTRAN. Due to the different algorithms it was not possible to fix the same integration step and grid resolutions for all the codes. In case of DYNAMION and IMPACT in fact the choice of a large number of steps is necessary in order to avoid artificial collision (DYNAMION) and numerical problems when using the Lorentz integrator (IMPACT). The number of step in PARTRAN has been chosen by taking the emittance curve obtained with a high-resolution simulation and by lowering the number of integration steps: the optimal choice is the one providing the same emittance curve with the minimum number of integration steps. The CPU time has been found dependent on the choice of boundary conditions in IMPACT: selecting the closed boundary condition on the rectangular pipe (the same of HALODYN) the requested CPU time is reduced of about 40%.

code (a.o.)	*platform*	*GUI*	*parallel*	*particles*	*s. c. solver*	*boundary conditions*	*CPU time*
DYNAMION	Windows	no	no	$5x10^3$	3D p-p		1.3 days
	(Li)Unix						2.5 days
HALODYN	(Li)Unix	post	yes	$1x10^6$	3D PIC	closed	1.0 day
IMPACT	(Li)Unix	no	yes	$1x10^6$	3D PIC	open	4.0 days
						closed	2.5 days
LORASR	Windows	yes	no	$1x10^6$	3D PIC	closed	N.A.
PARMILA	Windows	post	no	$1x10^5$	2D PIC	open	1.5 days
					3D PIC		7.0 days
PARTRAN	Windows	post	no	$1x10^5$	3D PIC	open	6.0 days
PATH	Windows	yes	no	$1x10^5$	2D PIC	open	1.5 days
				$2x10^4$	3D p-p		1.5 days

Table 1: Summary table with an indication of the requested CPU time for different choice of solvers and boundary conditions. See text for the discussion on the choice of the number of macro-particles, the integration step and grid resolution. All the codes having a post-processor for the graphical analysis are labeled with "post" in the GUI entry.

4 Static Benchmarking

In order to investigate the quality of the different space-charge routines, several tests without any tracking (to avoid coupling with different lattice modelling) have been run, after modifying the source code to output the electric field at the position of each particle and on the grid points of the mesh box (for PIC codes only). In order to compare the results as function of numerical parameters, such as the number of macro-particles *Np* and the mesh resolution *Dx*, we abolished any box resize and adaptive re-mesh in the PIC codes IMPACT and TOUTATIS.

The goal of these tests is to quantify the accuracy of the solver when computing the electric field and the sensitivity to numerical parameters (noise and discretization errors).

4.1 Benchmarking the Space-Charge Electric Field

We generated three Gaussian distributions of 10^4, 10^5 and $5\text{x}10^5$ macro-particles having $\sigma_{x,y}$ = 4 mm and σ_z = 8 mm and representing a $^{238}U^{+28}$ bunched beam of current I=1 mA at the energy of 1.4 MeV/u. The grid box of the PIC codes (IMPACT, TOUTATIS, LORASR and HALODYN) is fixed to $Lx = Ly = 6.4$ cm and $Lz = 18.4$ cm, whereas for DYNAMION no grid must be introduced, this code having a direct particle-particle solver. The electric field at the particle position obtained in output from the codes is compared with a semi-analytical solution computed with an algorithm described in [13]. The error we used as figure of merit is defined as follow: only a longitudinal slice of $2\,\sigma_z$ is taken into account (see Fig. 1); for each particle within a cylindrical shell $r \pm \delta r$, the error is defined as

$$\epsilon(r,n) = \frac{\|(E^C_{x_n} - E^A_{x_n}, E^C_{y_n} - E^A_{y_n})\|}{\|(E^A_{x_n}, E^A_{y_n})\|} \ ,$$

and averaged over the shell, providing

$$\epsilon(r)_{rms} = < \epsilon(r,n) > |_{\sqrt{x_n^2+y_n^2} \in r \pm \delta r} \equiv \frac{\delta E}{E}(r) \ .$$

$(E^C x_n\,,\ E^C y_n)$ is the transverse electric field at the position of the nth particle as computed by the code, whereas $(E^A x_n\,,\ E^A y_n)$ is the corresponding semi-analytical solution.

The results for DYNAMION and the PIC codes with a grid resolution of 64^3 (or 65^3 according to the algorithm) and 128^3 (129^3) are shown in Fig. 2 and Fig. 3 respectively. The relative error shows for all codes an exponential drop within the bunch core, whereas outside some differences appear: while the IMPACT (open boundary conditions) error keeps converging to zero, it remains on the 1% level for DYNAMION, TOUTATIS and IMPACT (closed b.c.) and it increase up to 10% in HALODYN and LORASR. Simulations with larger boundaries revealed a general improvement of this error: by running HALODYN with a box of Lx = 9 cm, for instance, the error remains at 1% level. Note that the semi-analytical solution assumes open boundary conditions. We interpret the 100% error at the bunch centre for all the codes as follow: going the electric field E linearly to zero as $r \rightarrow 0$, the same is true for the error δE.

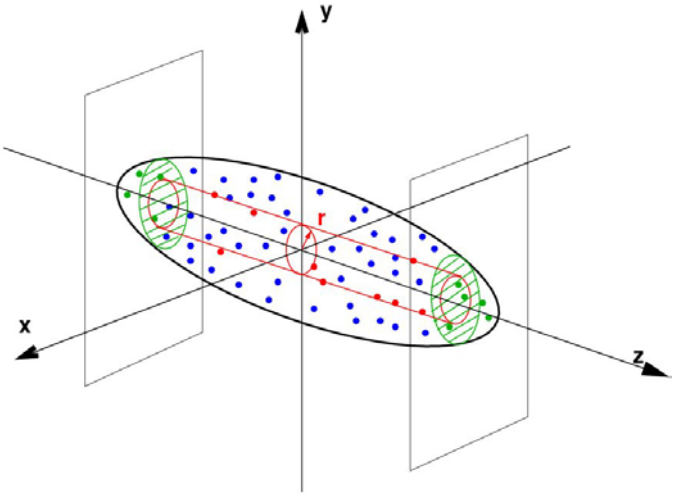


Figure 1 Error computation scheme: in green the particles outside the longitudinal slice, which are ignored; in blue the inner particles, in red the one having the same radius *r*.

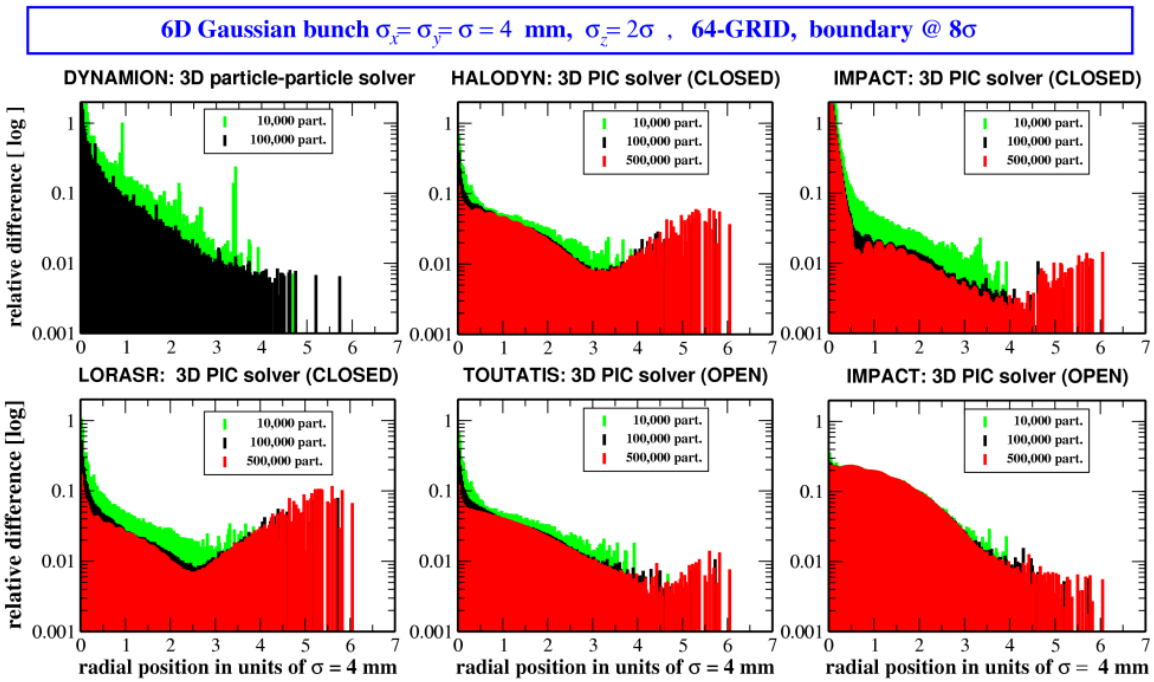


Figure 2 error δE/E for DYNAMION and PIC codes with a grid resolution of 64^3 (65^3).

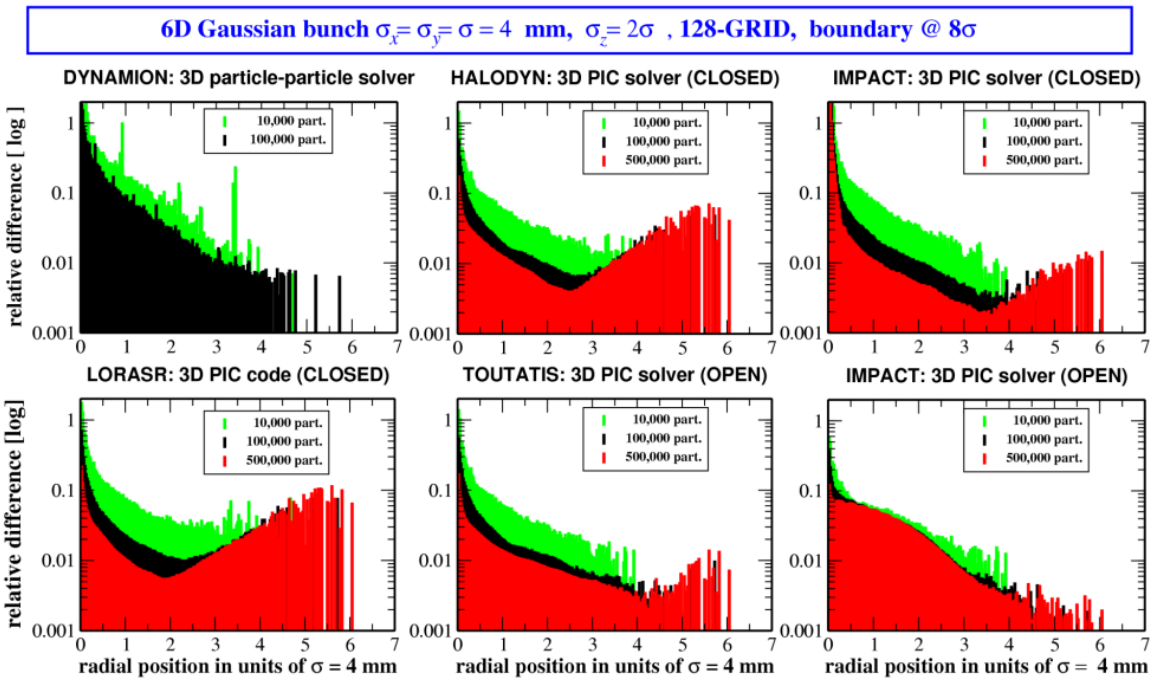


Figure 3 Field error δE/E for DYNAMION and PIC codes with a grid resolution of 128^3 (129^3)

In APPENDIX A technical details for the computation of the electric field error are given. The formats of the input and output files are also outlined, together with some typical conversion relations. Code developers who want to join the comparison should modify the sources in order to read from file a common particle distribution and produce a "standard" output. Modifications in the source are needed since tracking codes usually do not print out the space-charge electric field, whose "internal" units are usually code-dependent.

4.2 The Single Particle Tune Test

Even if the quality of the space-charge electric field is a clear figure of merit of a solver, its uncertainty does not provide an estimation of the induced error in the beam dynamics. Resonant halo and resonance trapping and de-trapping are both mechanisms of interest in high intensity regimes. A correct description of these phenomena passes through the correct representation of the single particle dynamics, which in turn is characterized by the single particle tune (SPT) and the crossing of a resonant condition. Space charge depresses the tune due to its intrinsic defocusing characteristic. Given an ideal lattice description, i.e. a correct tune diagram, error in the electric field computation results in wrong depressed SPT and inaccurate description of the resonance crossing.

In order to establish a common SPT test for the codes we set up the following procedure. A single test particle is made oscillate in a 1D constant focusing channel with a given bare tune q_0 using a simple single particle tracking routine (left plot of Fig. 4). The initial conditions for the test particle are $x \neq 0$, $y = z = 0$ and $p_x = p_y = p_z = 0$. Superimposing the frozen space-charge field, i.e. an electric field independent on both the test particle oscillations and the internal motion of the particle distribution, the test particle explores the entire space-charge field. Recording its oscillations, the depressed SPT is inferred via FFT (centre plot of Fig. 4). As the electric field is originated by a Gaussian distribution, analytical formulae are available to derive the correct depressed SPT, to be compared with the one experienced by the test particle. Being the electric field computed by the codes frozen, we speak about "instantaneous" SPT.

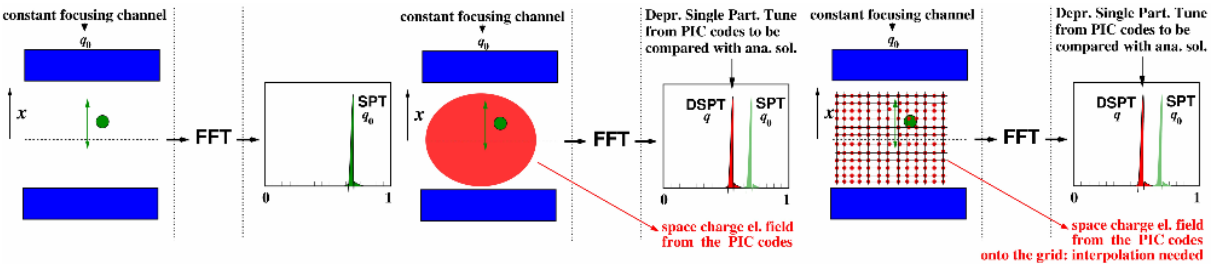


Figure 4 Schematic view of the SPT test: the 1D oscillation of a single particle are recorded and from the FFT the bare SPT is inferred (left); if the space-charge field is superimposed the resulting SPT is shifted towards left (center); in order to compute the electric field at any location, the field on the grid point needs to be used together with the interpolation routine of each code (right).

This test so far has been performed for PIC codes only. To apply the space-charge force to the test particle we indeed need the electric field on the grid points and an interpolation routine to infer the field at the particle position (right plot of Fig. 4). To take into account differences in the interpolating routines of each code, the latter ones were "exported" in the single particle tracking routine and used consistently with the code under investigation. This procedure is incompatible with the nature of a particle-particle solver like the one in DYNAMION, which was kept out of this test.

Errors in PIC solver are generally driven by two main factors: the finite number of macro-particle Np, because of the fluctuating number of particles within a grid cell, and the grid resolution Δx. It was observed that, averaging over the depressed SPT inferred from several random distributions, the discretization results in a numerical "*tune shift*" (bias, not physical), whereas the statistical fluctuations produce a numerical "*tune spread*". These two quantities can be used to compare different solvers and to investigate their dependence on numerical parameters. A typical SPT plot against the initial single particle position, for twenty random distributions, is shown in Fig. 5.

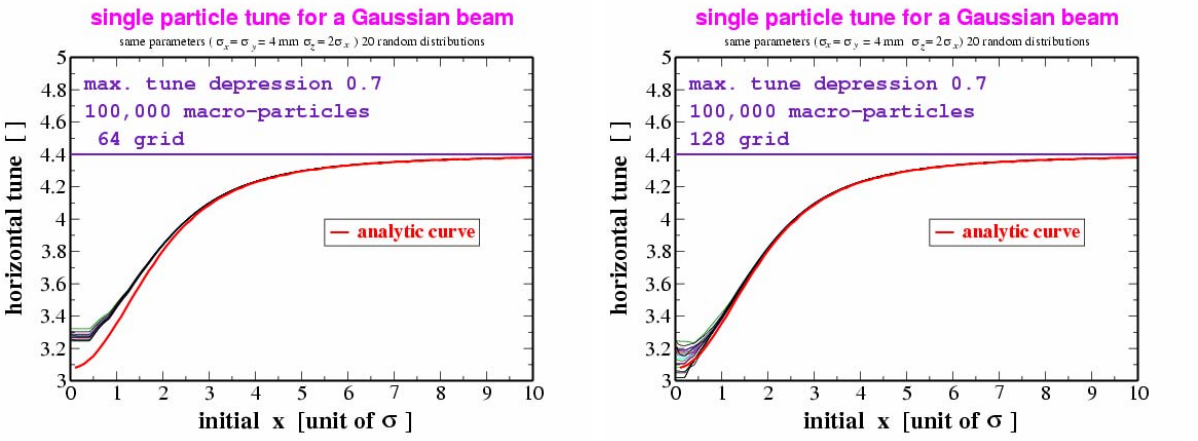


Figure 5 Single particle tune as function of the initial condition for twenty random distributions of 105 macro-particles for a 64^3 (left) and a 128^3 (right) grid, computed by IMPACT. The statistical "tune spread" is clearly visible at the bunch center together with the numerical "tune shift" (the average SPT does not coincide with the exact one).

In order to compare different codes we propose a scaling law, originally derived by Turchetti [14] that relates the error in the depressed SPT reconstruction to these two parameter $\delta q = f(Np;\Delta x)$. Fixing both the numerical parameters and the constant focusing channel, the PIC solvers can be compared looking at the coefficients of this law: the smaller they are, the more accurate is the solver. The scaling law for the depressed SPT at the bunch centre reads

$$\delta q = K_1 \left(\frac{K_2(f_n, Vol_b)}{\sqrt{\Delta x^3 N_p}} + K_3(Vol_G, Vol_b)\Delta x^{\alpha} \right) ,$$

where K_1 depends on the constant focusing channel only, K_2 depends on the particle distributions f_n and the bunch volume Vol_b , whereas K_3 depends on both Vol_b and the grid box volume Vol_G . The power is inferred via data fitting. The first term in the r.h.s. $\propto(\Delta x^3 Np)^{-1/2}$ depends basically on the number of particle per cell and introduces a numerical “tune spread" simulating several initial particle distributions. The second one $\propto\Delta x^{\alpha}$ has no statistical origin and introduces a numerical "tune shift". These two quantities have been calculated running static simulations using 20 random Gaussian distributions with the same RMS sizes and varying the grid resolution from 32^3 to 128^3. For each code and grid resolution, the SPT at the bunch centre was averaged (providing the numerical "tune shift" after subtracting the analytical value), computing also its standard deviation (the "tune spread").
In Fig. 6 these two quantities are shown together with the constants in the scaling law obtained from the fitting. The same quantities are also computed against the initial particle position x and are plotted in Fig. 7: the numerical spread drops to zero exponentially, whereas the numerical shift appears to have a power-law decay.
In APPENDIX B technical details of the SPT test are given. The code developer needs to modify the source code in order to print out the space-charge electric field on the grid points, and to extract two subroutines to be included in the routine for SPT test **`tune_error`** : the first shall make the routine read this file, the second shall perform the same interpolation implemented in the PIC code. After running **`tune_error`** a file with the depressed SPT, the numerical "tune spread" and "tune shift" versus the particle position will be generated.

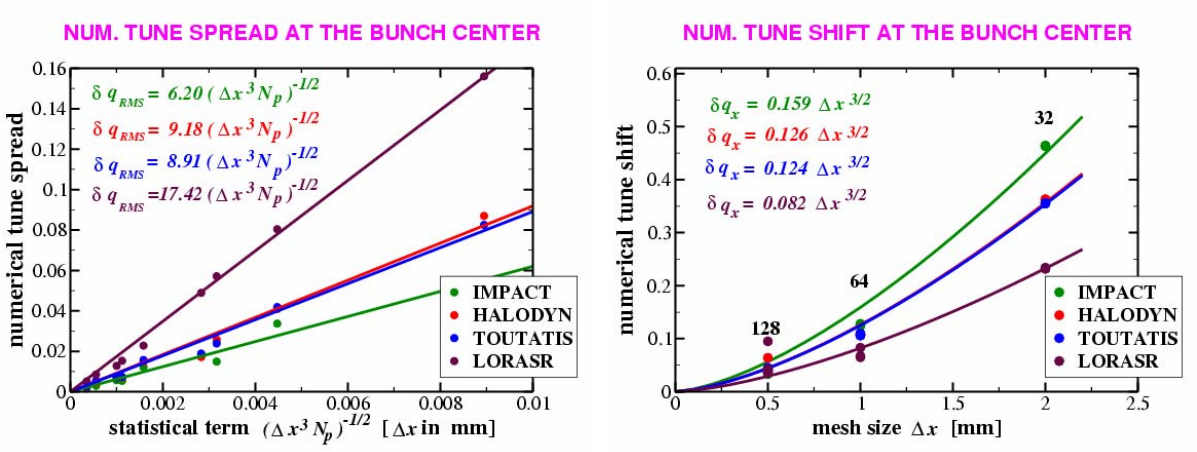


Figure 6 Numerical "tune spread" (left) and "tune shift" (right) at the bunch centre and corresponding constant K_2, K_3 .

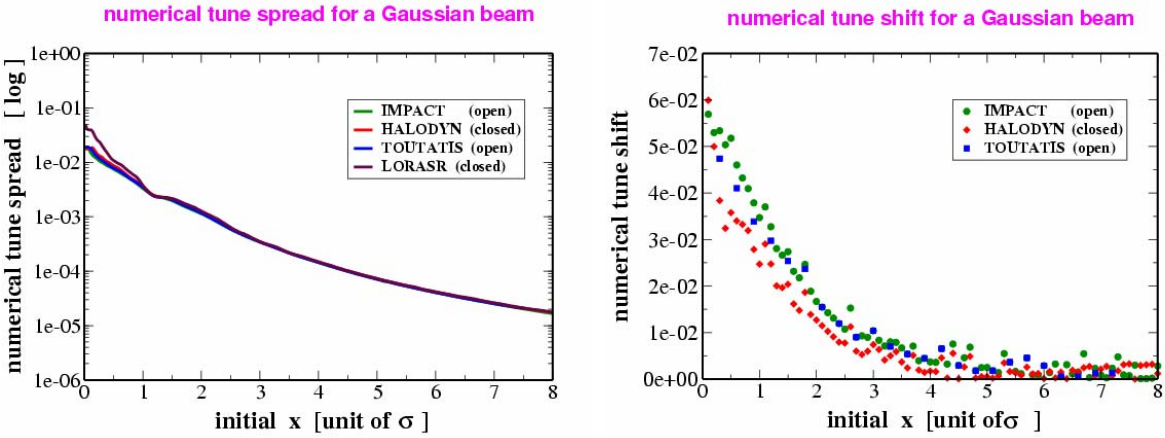


Figure 7 Numerical tune "spread" and "shift" versus the particle position.

5 UNILAC tracking

Crucial points in comparing and benchmarking several codes are the proper lattice representation (the five Alvarez DTL tanks in our case) in the input files, and the choice of proper numerical parameters. Each code has its own input file (or files). Units and conventions differ also from code to code as well as the definition of the element parameters. Some codes can describe macro-structures such DTL or SSCL (cell or tank), some others can track particle in a transport lines consisting of drift, quadrupole, solenoids and RF cavity only. Different RF modelling influences the definition of the synchrotron phase. Although in the transverse plane the particle loss definition is rather standard (beam pipe), this is not the case in the longitudinal plane. The integration step (and the consequent number of space-charge calculations) is another parameter highly code-dependent: a large number of integrations (≈100 per DTL cell), although redundant from the space-charge point of view, is sometimes necessary to guarantee the numerical stability of some tracking integrators. In some other codes such a refined calculation is not necessary and the same results are obtained with a lower resolution with few space-charge calls per DTL cell. Particular attention was therefore devoted to tracking simulations at zero current: if space charge is excluded from the

calculations and the different lattice representations are equivalent (and consistent with the real lattice), the same results from all the codes are expected. Different RF descriptions might introduce differences at level of single particle dynamics, that however should not affect the second order (RMS) moments.

After reaching a satisfactory agreement with the zero-current simulation, the next step is to include space charge. $^{238}U^{+28}$ of 12.5 mA (averaged over one RF period) is a reference beam for the future UNILAC operations. As the section upstream the DTL structure operates at 36 MHz and the DTL resonates at 108 MHz, only one bucket out of three is filled and the real bunch current is 37.5 mA. Neither IMPACT nor HALODYN can handle a bunch frequency different from the RF frequency. Empty buckets up- and downstream are therefore ignored by these codes and considered as filled by equivalent bunches. Two different kind of high-intensity simulations have been run.

In the first (CASE 1), the 37.5 mA are distributed over a 6D spherical Gaussian bunch, of RMS radii $\sigma_{x,y,z}$ = 1.75 mm and $\sigma_{x',y',z'}$ = 1.75 mrad, corresponding to longitudinal spreads in phase and energy of $\sigma_{\delta\varphi}$ = 4.15° and $\sigma_{\delta w/w}$ = 0.35% respectively. In the left plot of Fig. 8 the longitudinal phase space at injection is plotted together with the separatrix [1]: the beam is initially entirely within the stable bucket. The small (RMS normalized) emittances $\varepsilon_{x,y,z}$ = 0.167 mm-mrad result in severe space-charge tune depression (0.55 transversely and 0.35 longitudinally), making the longitudinal dynamics be dominated by space charge.

In the second simulation (CASE 2), the same current is distributed over a longer ellipsoidal Gaussian bunch with different transverse RMS sizes (in order to investigate possible difference between *r-z* and 3D space-charge solvers) and larger longitudinal dimensions $\sigma_{\delta\varphi}$ = 12.4° and $\sigma_{\delta w/w}$ = 1.05%. The beam occupies almost the entire stable region up to the separatrix (see right plot of Fig. 8). The larger longitudinal emittance ε_z = 1.5 mm-mrad results in lower space-charge tune depression (0.567 transversely and 0.88 longitudinally), making in this case the longitudinal dynamics be dominated by nonlinear RF fields.

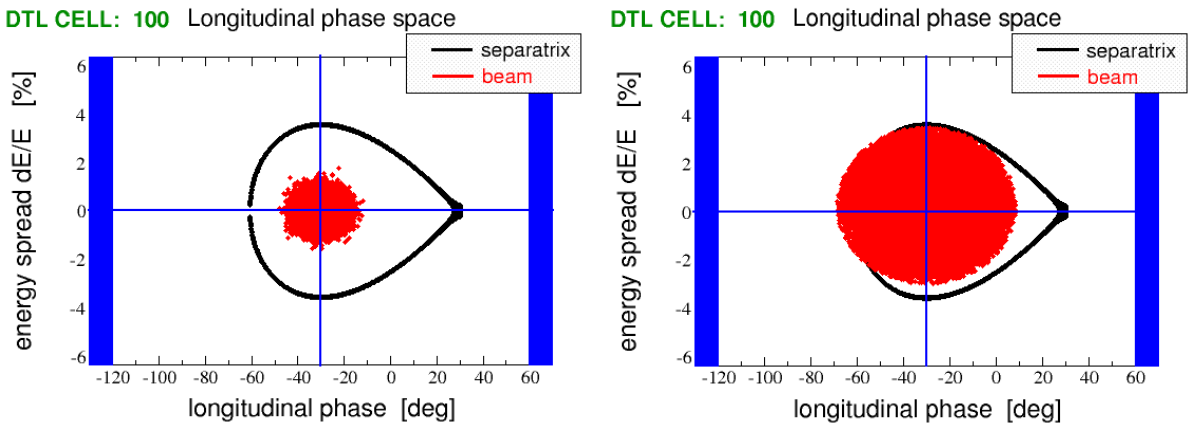


Figure 8 Longitudinal phase space portrait at the entrance of the first Alvarez DTL tank of UNILAC and corresponding separatrix for CASE 1 (left) and CASE 2 (right).

5.1 Lattice modelling

The Universal Linear Accelerator UNILAC is a part of the complex of accelerators at GSI, whose layout is sketched in Fig. 9. The ions (partially or fully stripped) are injected in the Alvarez DTL at energy of 1.4 MeV/u and delivered at 11.4 MeV/u either to the experiment

[1] The separatrix is inferred solving numerically Eq.(6.12) of T. Wangler, *Principles of RF Linear Accel- erators*, Wiley, New York, 1998 (ISBN 0-471-16814-9)

hall or to the heavy-ion synchrotron SIS18. The 54 meter-long DTL consists of five independent tanks (the second and the third are included in the same container). The RF of one or more tanks can be turned off for experiments at energy lower than 11.4 MeV/u (see Fig. 10). Two buncher cavities are also installed in two inter-tank sections, but they are not routinely in use and have been not included in the modelling. In total the DTL consists of 178 RF cells. The focusing scheme follows the FDDF structure and the dc quadrupoles are grouped in 13 families, each one sharing a common power supply. The inter-tank focusing is obtained by independent quadrupoles defining a triplet lattice FDF. The normalized acceptance in the transverse planes is of about 15 mm-mrad. The design synchronous phase is φ_s= -30° in the first three tanks and φ_s=-25° in the last two tanks.

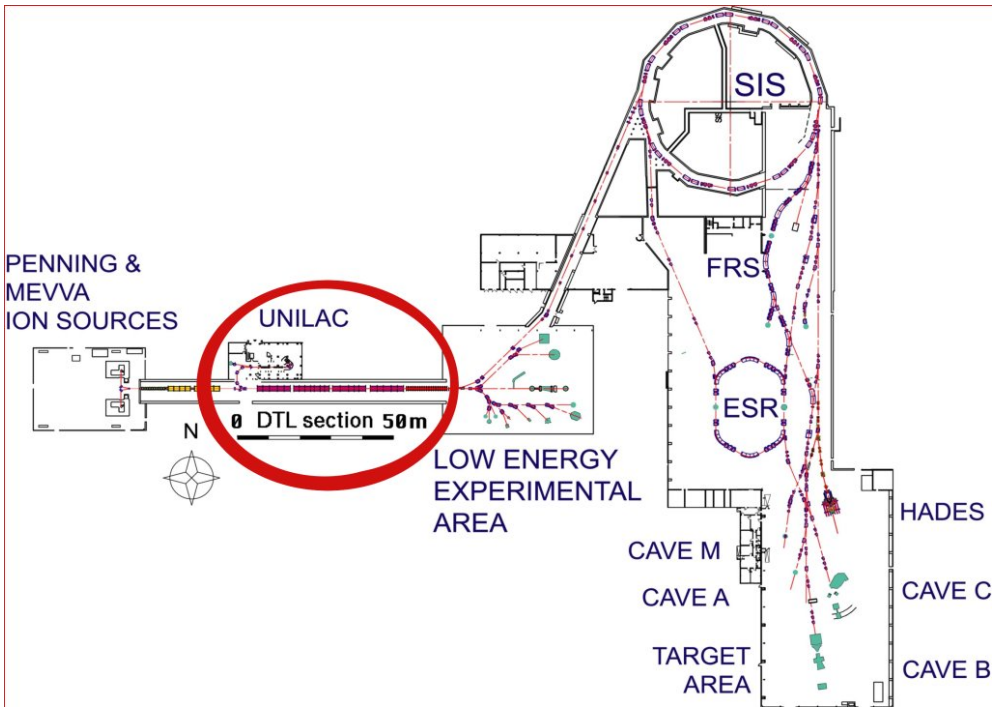


Figure 9 View of the actual complex of accelerators at GSI. The Alvarez DTL section is the one delimited by the red circle

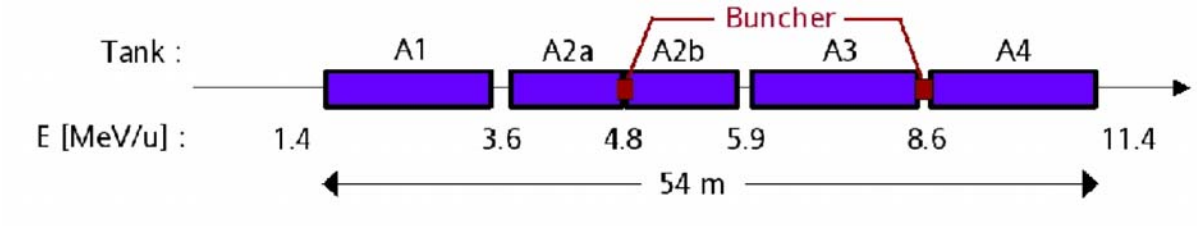


Figure 10 Sketch of the DTL section with the beam energy at the end of each tank.

The DTL has been modelled in different ways according to the different features of the codes. In some cases the choice of a certain representation has been dictated by some code limitations,
whereas in other cases by convenience.

- **PARMILA**: The structure has been defined by using the "DTL" command line. With this option, once the entrance energy and the number of cells are defined, the code builds automatically the DTL tank. The FDDF focusing structure is defined by using the "change" command. The RF modelling is defined importing the Transit-Time-Factor

(TTF) table generated by SUPERFISH. It was observed that letting PARMILA define the structure, tanks longer of several centimetres with respect to the real one were defined. To preserve the overall length and to compare the beam envelopes with other codes, it was decided to cut part of the inter-tank drift at the end of each tank. Notice that the choice of using the "DTL" command line might not be suitable for simulations with one or more DTL tanks with the RF turned off. These tanks should be better described using "transport" sections, consisting of drift and quadrupole only (in analogy with the modelling done in HALODYN).

- **IMPACT**: The structure has been defined by using the DTL-cell option, where the user defines the cell geometry and the quadrupole gradients. The accelerating RF on-axis field is read from an external file, which is inferred from the SUPERFISH output files (see Fig. 11). If the Lorentz integrator is used (and this is the case of the HIPPI simulations), an additional pre-processing is needed: an external routine written by Ji Qiang reads the above external file and produces the one suitable for this integrator. Frank Gerigk wrote a phyton script for the automatic conversion of all the 178 files.

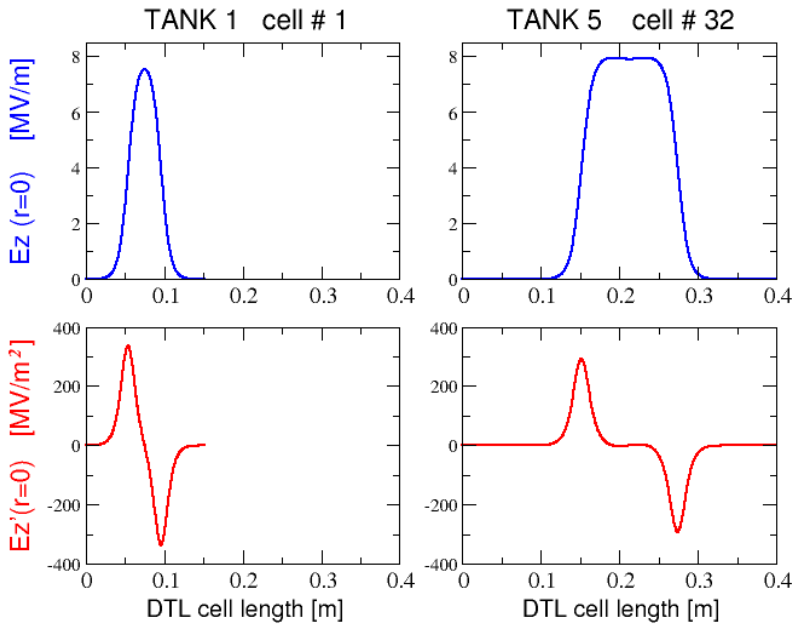


Figure 11 RF on-axis electric field and its derivative as computed by SUPERFISH and imported by IMPACT. Left: the first DTL cell in the first tank. Right: last DTL cell in the last tank.

- **DYNAMION**: each DTL cell is defined in way similar to the one of IMPACT. The RF electrical field is calculated in pre-processing for each cell. The real geometry of the drift tube is used, including inner/outer diameters and rounding of the edges. Assuming an azimuth asymmetry, the Laplace equation is solved by a finite-element code. The potential, within the gap as well as inside the drift tube, is approximated by series with 30 coefficients

$$U(r,z) = -V\left[\frac{z}{l} + \sum_{n=1}^{30} A_n \sin(2nkz) I_0(2nkr)\right], \qquad k = \frac{\pi}{l},$$

where $l = L/2$ is half of the cell length, V is the RF voltage, and I_0 is the modified Bessel function. The expansion coefficients A_n are stored in external files and read by DYNAMION while tracking. The electric field components are eventually computed from the derivative of the potential.

- **LORASR**: also in this code each DTL cell is defined in way similar to the one of IMPACT. The RF field of each cell, generated by MICROWAVE-STUDIOLAB, is read

from external file. The RF field is obtained interpolating the longitudinal on- and off-axis electric field. The latter ones is computed at four different values of the distance from the axis *r* (see Fig. 12). According to the particle transverse position, i.e. *r*, the RF field is assigned by linear interpolation of the stored fields at the adjacent radial zones.

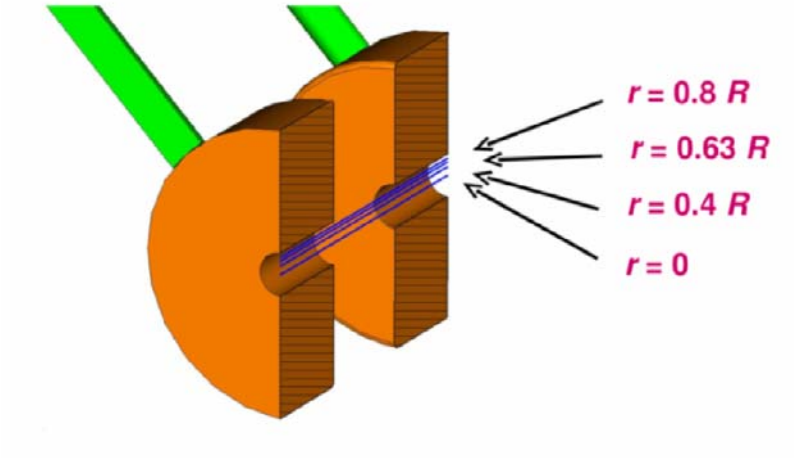


Figure 12 Sketch of the RF modeling for LORASR, with the four longitudinal electric field profile compute at different radial positions.

- **HALODYN**, **PATH** and **PARTRAN**: for these codes each DTL cell is modelled by a sequence of drift, quadrupoles and thin-lens RF cavities, whose accelerating voltage is inferred from the averaged accelerating electric fields (see Fig. 13). The RF kick to the particle momentum is given according to

$$F_{x,y}(r,z) = -K_0^{int}\frac{I_1(kr/\gamma)}{kr/\gamma}\sin(-kz+\phi_s)$$
$$F_z(r,z) = \frac{K_0^{int}}{kz}[I_0(kr/\gamma)\cos(-kz+\phi_s)-\cos\phi_s]$$

where

$$K_0^{int} = \frac{q2\pi\sin(-\phi_s)}{mc^2\lambda_{RF}\gamma^3\beta^3}(E_0TL), \qquad k=\frac{2\pi}{\beta\lambda_{RF}},$$

φ_s is the synchronous phase, $\lambda_{RF} = c = f_{RF} = 2.76$ m is the RF wavelength, I_0 and I_1 are the modified Bessel functions of order zero and one respectively, and $V = E_0TL$ is the effective accelerating voltage. For each cavity, the latter one is tuned in order to reach at the end of the cell a velocity satisfying the relation

$$\beta\lambda_{RF} = L_{cell}$$

where L_{cell} is the cell length. The choice of this modelling was mandatory for HALODYN, as this code does not include any other RF description. It was chosen for PATH and PARTRAN for convenience, as it was decided to avoid the time-consuming production of the 3D electromagnetic field map required for a more refined modelling. Notice that the HALODYN input le is compatible with PARMILA. For simulations with turned-off tanks using this code is therefore enough to replace the "DTL" command line and the following instruction with the corresponding segment of the HALODYN input file, after setting to zero all the accelerating voltages.

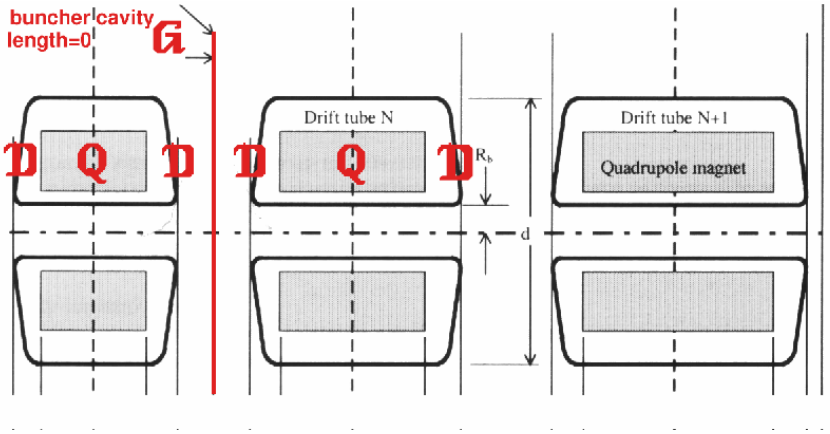


Figure 13 DTL modeling for HALODYN, PATH and PARTRAN.

The input files for SUPERFISH used to generate the TTF table for PARMILA and the "rfdata" files for IMPACT can be downloaded from the HIPPI code benchmarking web page [1]. Notice that so far no cross check between the RF fields used by DYNAMION, LORASR (MICROWAVE-STUDIOLAB) and the one generated by SUPERFISH has been performed. In simulations at zero current, all the codes using an RF modelling inferred from SUPERFISH calculations (IMPACT, PARMILA, HALODYN, PARTRAN and PATH) showed a good agreement in terms of the longitudinal RMS emittances. Since some evident discrepancies appeared between theses codes, DYNAMION and LORASR (see Sec. 5.2), such a cross check is desirable for the future. This should be accompanied by a re-tuning of the energy gain per cell, as the ones of IMPACT, HALODYN, PATH and PARTRAN presents errors in the level of few percent's in the last two tanks. Notice also that the RF modelling used in all the codes assume the azimuth symmetry. This approximation is legitimate for the purpose of this comparison, as the Alvarez DTL satisfies this condition. For other structures, such as spoke cavities, half-wave resonators, and slot-finger structures, the use of 3D electromagnetic field maps would be desirable.

5.2 Tracking at zero current

Preliminary tracking simulations were run using a (not matched) spherical 6D-Gaussian bunch of $\sigma_{x,y,z}$ = 2 mm ($\sigma_{\delta\varphi}$ = 4.7°), $\sigma_{x',y',z'}$ = 2 mrad ($\sigma_{\delta W}$ = 1.3 MeV) and 10^4 particles, representing a zero-current $^{238}U^{+28}$ beam of kinetic energy W = 1.4 MeV/u. The Twiss parameters of the injected beam were chosen for convenience $\beta_{x,y,z}$ = 1 m and $\alpha_{x,y,z}$ = 0, which resulted in equal RMS normalized emittances[2] in the three planes, $\varepsilon_{x,y,z}$ = 0.22 mm-mrad ($\varepsilon_{\delta\varphi\text{-}\delta W}$ = 0.22 deg-MeV). In Fig. 14 the density profile of each plane is plotted against the longitudinal position along the DTL.

The ensemble of particles was generated and written on file by a stand-alone routine. Its units and format were then translated in order to be readable by each code. In the HIPPI code benchmarking web page[3] the different units and conventions implemented in the codes are listed, together with the conversion relations. As mentioned in the introduction, the scope of this test is twofold. First goal was the preparation of the input files for all the codes, crosschecking the correctness of both the different lattice parameters (such as lengths, gradients, voltages, energy gain) and the unit conversion. Second goal was the estimation of the level of disagreement expected by different lattice and RF modelling.

[2] In the rest of this note, while speaking of emittance, we always refer to the normalized RMS emittance.
[3] http://www-linux.gsi.de/~franchi/HIPPI/web_tools.html

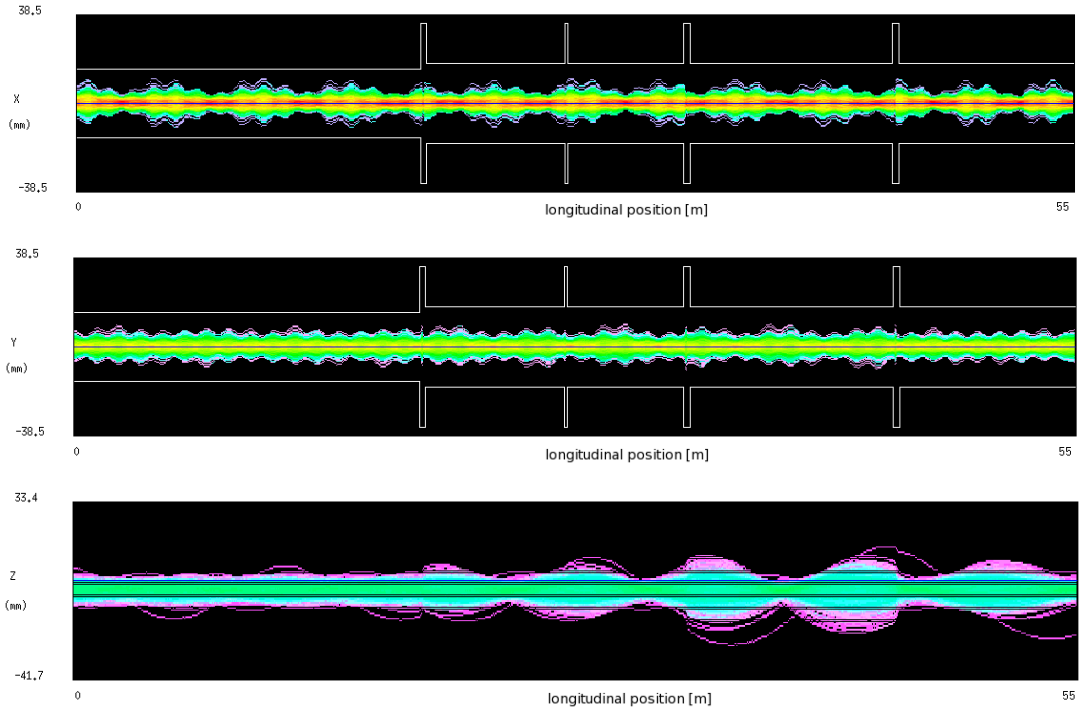


Figure 14 Beam density profile against the longitudinal position along the DTL (HALODYN). The canonical variable z is defined from the distance in phase with respect the synchronous particle according to $z = \beta\lambda_{RF}(\varphi - \varphi_s)/(2\pi)$.

In Fig. 15 the transverse RMS beam sizes as computed by the codes are plotted along the DTL, together with the beam envelope. The agreement among the codes in terms of the RMS values is excellent. The maximum values show an almost perfect agreement between HALODYN, IMPACT, PATH and PARTRAN. Small differences appear in the results of other codes. In case of PARMILA, this might be due to the tiny differences in the lengths of the tank (see discussion of Sec. 5.1). In case of LORASR and DYNAMION, the different RF modelling (and hence RF defocusing) might be drive a slightly different dynamics with respect to the other codes. Similar considerations apply to the longitudinal plane as shown in the right plot of Fig. 16: DYNAMION and LORASR predict a RMS emittance which differs of about 10% in some regions of the last two tanks.

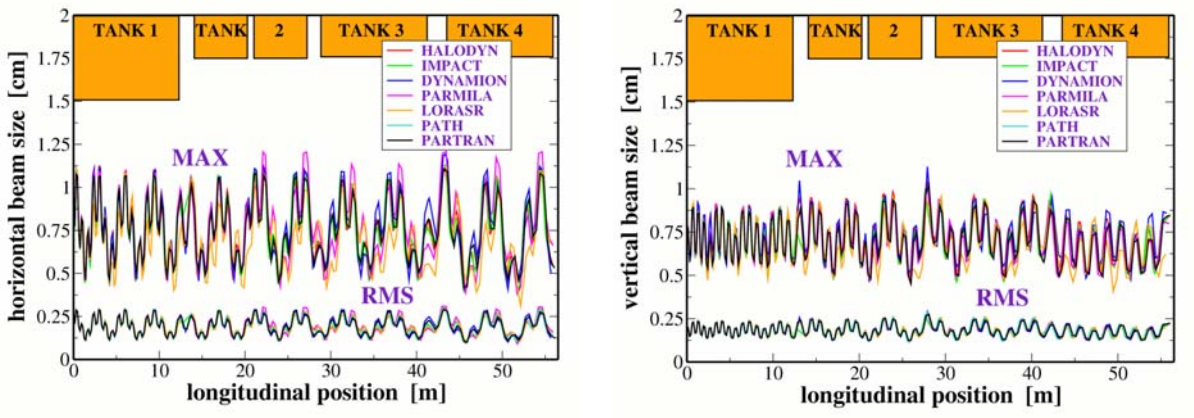


Figure 15 Horizontal (left) and vertical (right) RMS beam sizes and envelopes computed by all the codes and plotted along the DTL.

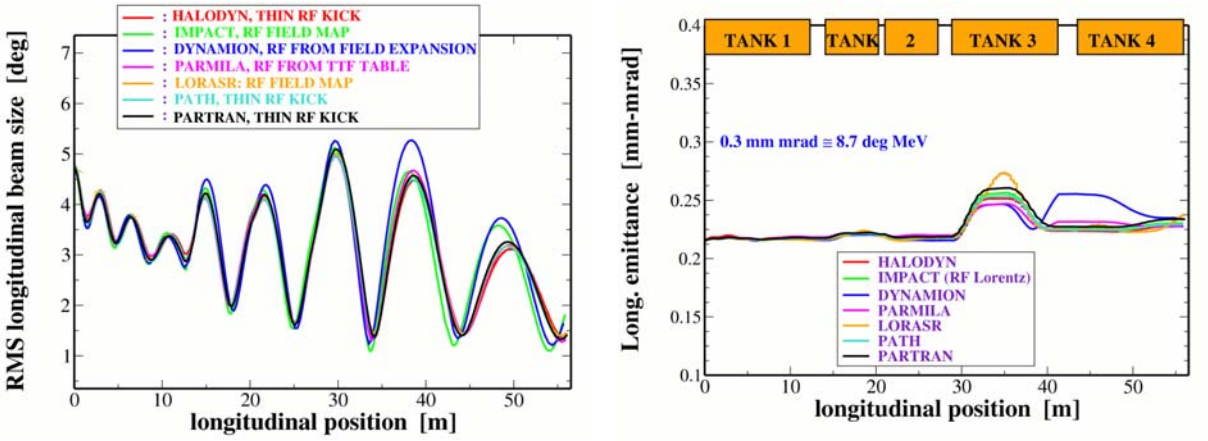


Figure 16 RMS longitudinal phase and emittance computed by all the codes and plotted along the DTL.

A more careful cross check of the RF field between these two codes and the one simulated by SUPERFISH would be of help in trying to better understand this discrepancy. The transverse emittances plotted in Fig. 17 show no emittance growth due to the RF defocusing (the one predicted by LORASR is lower than 3%).

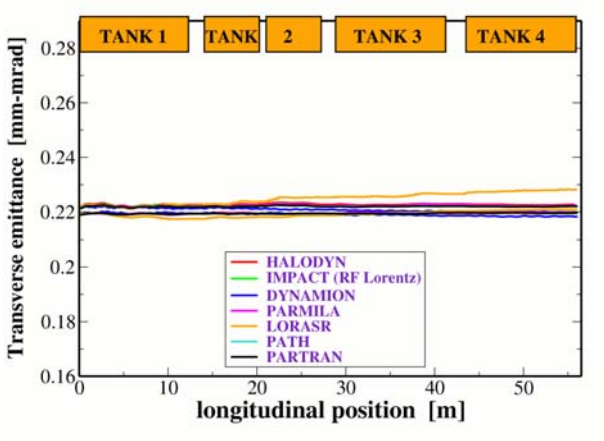


Figure 17 RMS transverse emittance computed by all the codes and plotted along the DTL.

Encountered problems and fixed bugs

In preliminary simulations using the IMPACT code, the linear map was used to describe the RF field. It was observed that with this choice it was not possible to reproduce the same longitudinal emittance predicted by other codes. For this reason this and all the following simulations were run using the Lorentz integrator. The latter one requires a more refined integration step with respect the linear map, resulting in a higher CPU time. The choice was anyway mandatory for the simulation of CASE 2, as the linear map does not describe properly the dynamics of the bunch tail when they are in proximity of the longitudinal separatrix. It was also found that DYNAMION has a convention for the polarity of the quadrupole gradients opposite with respect to the one of all the other codes. This problem was overcome by exchanging the transverse coordinates and momenta while creating the external file with the initial distribution.

Bugs have been found and fixed in three codes. In PARMILA the energy gain was not properly scaled for the charge state *Z*. The same factor was missing in the quadrupoles focusing strengths of the in HALODYN. In PATH eventually, the computation of some quantities, like the longitudinal RMS size, was not properly performed.

5.3 Tracking at 37.5 mA: CASE 1

The first tracking simulations including space-charge computations were run using a (not matched) spherical 6D-Gaussian bunch (truncated in each phase space at 3σ), representing a $^{238}U^{+28}$ beam of kinetic energy W = 1.4 MeV/u. Other beam and lattice parameters are listed in Tab.2.

	x-x′	*y-y′*	*z-z′*	*δφ-δw/w*
σ_q	1.75 mm	1.75 mm	1.75 mm	4.17°
$\sigma_{q'}$	1.75 mrad	1.75 mrad	1.75 mrad	0.35%
ε_q	0.167 mm-mrad	0.167 mm-mrad	0.167 mm-mrad	4.84 deg-MeV
Twiss β	1 m	1 m	1 m	
Twiss α	0	0	0	
bare phase advance	45°	45°	42°	
depressed phase advance	25°	25°	15°	
tune depression	0.55	0.55	0.35	

Table 2 Initial lattice and RMS beam parameters of CASE 1.The phase advances are computed over the first focusing period. The depressed phase advance is computed assuming a 6D uniform space-charge distribution. No coupling exists between the planes.

In Fig. 18 the evolution along the DTL of the transverse emittances as computed by the codes are plotted. All the codes predict the same behaviour, up to few percents, until the beginning of tank 3 (≈ 29 m). At the exit of the entire DTL the agreement remains on the same level for HALODYN, IMPACT, PATH and PARTRAN only. Despite the fact that all the codes predict the same overall behaviour, the final horizontal emittance presents a large spread of about ±20%, whereas in the vertical plane the discrepancies remains confined to ±5%. Different transverse boundary conditions do not appear to explain this spread, as codes with closed conditions like HALODYN and LORASR predict a final horizontal emittance with the largest disagreement, whereas the results from codes with open conditions such as IMPACT, PATH and PARTRAN are compatible with the ones of HALODYN at the level of few percents.

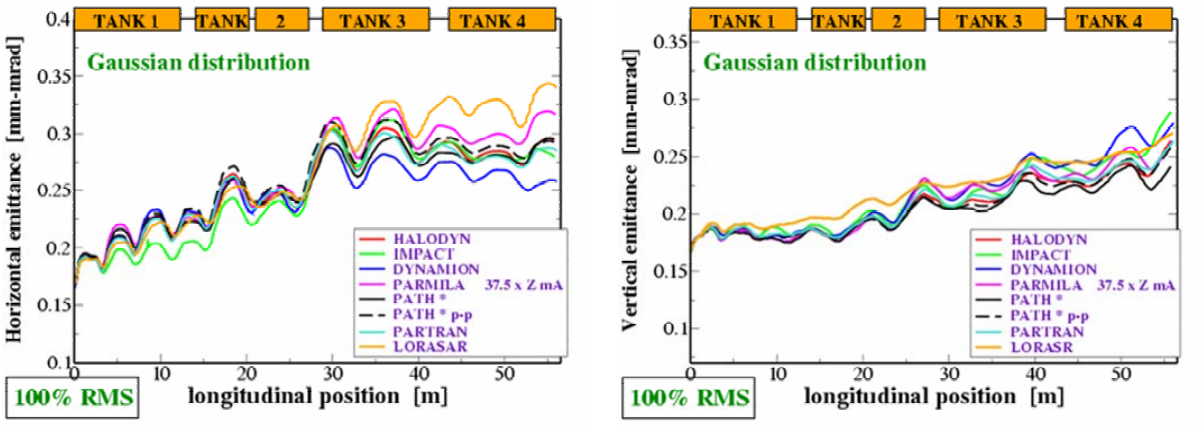


Figure 18 CASE 1: normalized transverse RMS emittance computed by all the codes along the DTL.

The small emittances $\varepsilon_{x,y,z}$ = 0.167 mm-mrad result in severe space-charge tune depression, making for instance the longitudinal dynamics be dominated by space charge, rather than by the RF nonlinearities. In the right plot of Fig. 19 the evolution along the DTL of the longitudinal RMS normalized emittances are plotted. After a series of code adjustments discussed below, the final agreement for almost of the codes is within few percents.

DYNAMION seems to predict a lower emittance growth with respect the results from the other codes.

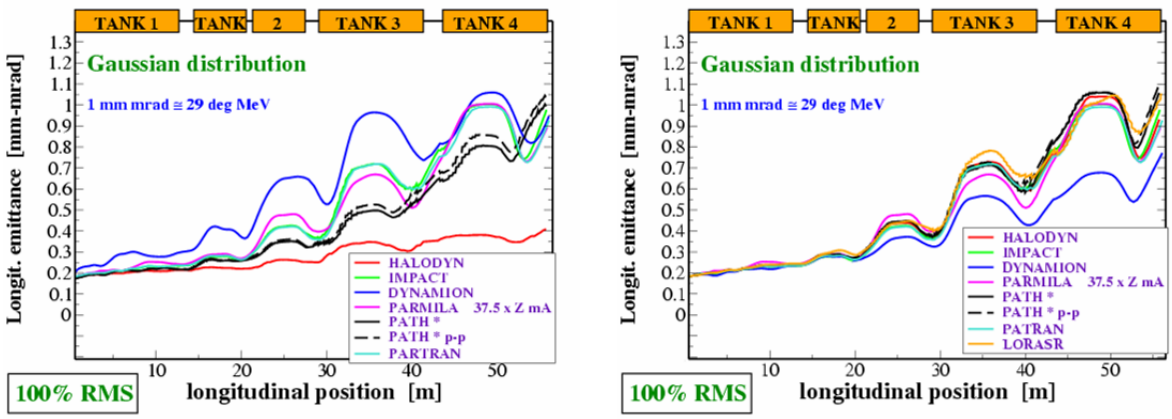


Figure 19 CASE 1: normalized longitudinal RMS emittance computed by all the codes along the DTL, before the code adjustments (left) and after (right).

Encountered problems and fixed bugs

As for the zero-current case; some bugs related to the charge state Z≠1 were found and fixed. When using the RF linear map in IMPACT, a scaling factor Z^2 was missing. Simulations with PARMILA at 37.5 mA showed no emittance growth, whereas multiplying the beam current by *Z*, curves compatible with the ones of other codes were obtained. It was not possible to receive a confirmation by the authors of PARMILA on this regard.

The different longitudinal emittances simulated by PATH (black lines in Fig. 19) are related to two separate problems, solved only recently. After the preliminary simulations (left plot), it was observed that the initial longitudinal phase space was generated according to an uniform distribution in the ($\delta\varphi$-δw) space. Due to technical problems making PATH reading an external file containing the particle distribution, an internally-generated RMS-equivalent Gaussian distribution was used. An internal bug made the longitudinal phase space be however uniform. This problem was overtaken by selecting a Binomial distribution among the different options, and by setting the distribution parameters $Nr = Nz = 500$. A second problem fixed by the authors was a wrong scaling in the space-charge calculations with the energy increase.

As far as DYNAMION is concerned (blue lines in Fig. 19), after the first simulations and the analysis of the longitudinal phase space evolution, it was observed that the initial RMS energy spread was a factor two larger. This was due to a wrong initial unit conversion from (z-z') to ($\delta\varphi$-$\delta w/w$). After this adjustment, the longitudinal emittance resulted appreciably lower than the one predicted by other codes. It was thought that a too large hard-sphere radius, introduced to avoid particle collisions, might underestimate the longitudinal space-charge forces for such a small and intense beam. It was observed that reducing this radius, the overall growth of the longitudinal emittance was effectively enhanced, although for radii a factor 100 smaller than the default value no further differences was observed.

The large differences between the emittances simulated by HALODYN (red lines in Fig. 19) are related to a boundary-conditions issue. In preliminary simulations (left plot) a particle loss of few percents, not predicted by any of the other codes, also occurred. HALODYN has implemented a space-charge solver with closed boundary conditions only. Until 2005 a non-adaptive mesh box was used to determine the particle loss: to increase the resolution of the Poisson solver, the mesh box length was always kept close to the separatrix (64^3 points over a 3^3 cm^3 box). This resulted in particle loss not observed by other codes and in an underestimated longitudinal space-charge field. In 2006 the simulations were run with a

longer box of 3x3x30 cm^3 and 64x64x512 grid points, producing the results shown in the right plot of Fig. 19. The CPU time increase was of about 20%.

5.4 Tracking at 37.5 mA: CASE 2

In the previous simulations the initial particle distribution was spherical and the energy spread was one order of magnitude lower than the measured one. The following simulations track a particle distribution with longitudinal emittance one order of magnitude larger than the transverse ones, resulting in a much lower longitudinal tune depression. Moreover, the initial transverse RMS sizes (or equivalently the β-functions) are not anymore equal. The latter choice is motivated by the fact that the space-charge routine SCHEFF implemented in PARMILA and PATH is a 2D *r-z* solver, and it is of interest to test how robust is this approximation for an asymmetric bunched beam. The input distribution is as usual a 6D-Gaussian (truncated in each phase space at 3σ), representing a $^{238}U^{+28}$ beam of kinetic energy W=1.4 MeV/u. Other beam and lattice parameters are listed in Tab. 3.

	x-x′	*y-y′*	*z-z′*	*δφ-δw/w*
σ_q	1.35 mm	1.75 mm	5.23 mm	12.4°
$\sigma_{q'}$	2.25 mrad	1.75 mrad	5.25 mrad	1.05%
ε_q	0.167 mm-mrad	0.167 mm-mrad	1.5 mm-mrad	43.5 deg-MeV
Twiss β	0.6 m	1 m	1 m	
Twiss α	0	0	0	
bare phase advance	45°	45°	42°	
depressed phase advance	30°	30°	37°	
tune depression	0.67	0.67	0.88	

Table 3 Initial lattice and RMS beam parameters of CASE 2.The phase advances are computed over the first focusing period. The depressed phase advance is computed assuming a 6D uniform space-charge distribution. No coupling exists between the planes.

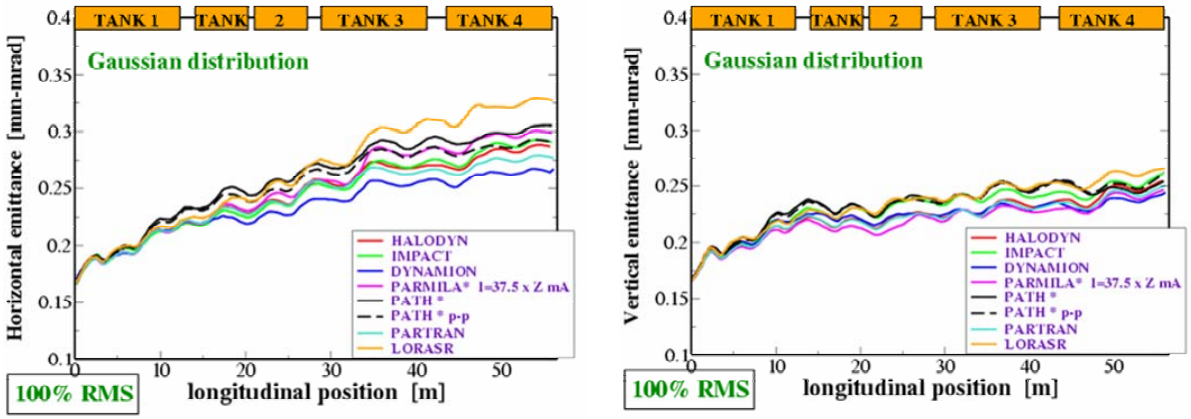


Figure 20 CASE 2: normalized transverse RMS emittance computed by all the codes along the DTL.

In Fig. 20 the evolution along the DTL of the transverse emittances as computed by the codes are plotted. In the horizontal plane (left plot) the codes begin to diverge already inside the first tank (≈ 10 m). As for CASE 1, the final horizontal emittance presents a large spread of about ±10%, whereas in the vertical plane the discrepancies remains confined to ±4%. Codes with the 2D Poisson solver SCHEFF are well within these values, confirming the robustness of the *r-z* approximations for the beam parameters under consideration.

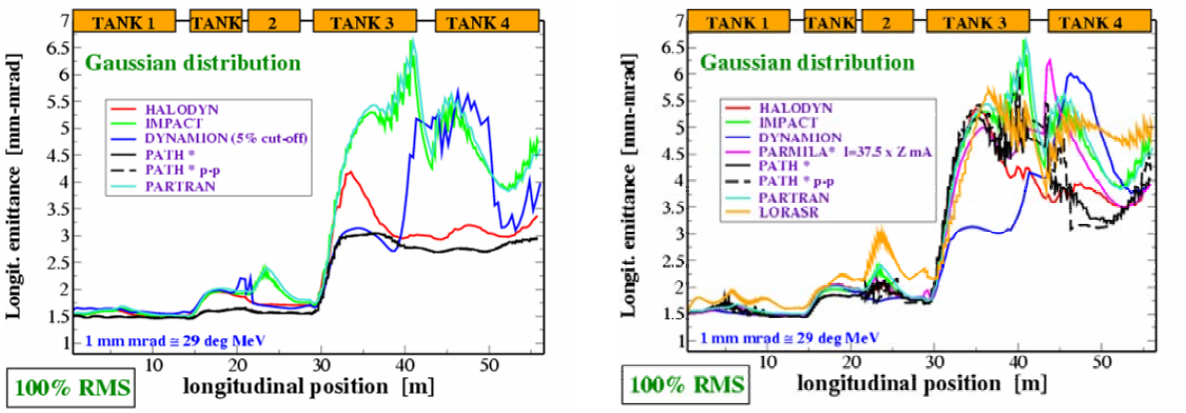


Figure 21 CASE 2: normalized longitudinal RMS emittance computed by all the codes along the DTL, before the code adjustments (left) and after (right).

More peculiar is the behaviour of the longitudinal emittance, shown in Fig. 21. After the preliminary simulations (left plot) and the same code adjustments carried out as in the previous case, it has been observed that results are highly sensitive to the definition of particle loss in the longitudinal plane. Few particles far from the RF bucket area, if included in the computation of the RMS quantities, might drive to an unrealistic overestimation.

Encountered problems and bugs

Besides the problems already described in Sec. 5.3, the results of CASE 2 shown in the left plot of Fig. 21 were driven mainly by different definitions, rather than by some bugs. By comparing the evolution of the longitudinal phase space predicted by all the codes, it was possible to conclude that they describe the same longitudinal beam dynamics. It was thought that the reason of the large discrepancies in the emittance curves, as well as of the different predictions in term of beam loss, might be in the different definition of particle loss in the longitudinal plane. While this definition in the transverse coordinates is rather straightforward (namely the beam pipe), this is not the case in the z-z' or $\delta\varphi$-$\delta w/w$ plane. This aspect was not of concern in CASE 1, as the small longitudinal emittance prevented the particle to reach the separatrix and no particle loss was predicted. It was observed that, keeping the default options, the definition of longitudinal particle loss was rather code dependent:

- **HALODYN** (version of 2005): a particle is lost if it is outside the (fixed) grid mesh box, i.e. if the longitudinal phase is such that the distance from the bunch centre is larger than the mesh box depth
- **IMPACT**: a particle is lost if the difference between the particle phase and the synchronous phase is larger (or smaller) than 180^o (-180^o)
- **DYNAMION**: a particle is lost if the relative momentum deviation $\delta p/p$ with respect to the synchronous particle is larger of Δp, where Δp is a free input parameter of the post-processor; while the code runs, all the particles are anyway tracked
- **PATH**: no internal check is performed
- **PARMILA**: a particle is lost if its energy is lower than E_{limit} , whose the default value is E_{limit}=1 GeV. If the phase deviation is larger than 180^o the particle at the head (or tail) of the bunch is automatically set at the tail (head) of the same bunch, simulating the jump in the neighbour bucket [to be confirmed]
- **PARTRAN**: a flag selects two options, either the same "check-and-jump" scheme of PARMILA or the same cut-off of IMPACT
- **LORASR**: particles with phase deviation larger than 180^o are mapped into the next bucket and thus they are not considered lost

In order to have a definition as much standard as possible the following measures in 2006 have been taken, taking the definition implemented in the IMPACT code as reference:

- **HALODYN** was updated to include the cut-off at ±180°
- **PATH** input file was updated introducing at the end of each DTL cell a "filter" (in phase) at ±180°
- **DYNAMION** post-processor was modified to remove the default cut-off in Δp and to implement the one at ±180°
- **PARTRAN** was run with the same option of IMPACT (the RMS emittances did not change, although a higher loss rate was obtained)
- **LORASR** was updated to include the cut-off at ±180°
- **PARMILA** input file was modified to include a cut-off in energy at the end of each tank, by means of the command E_{Limit}. The used cut-off energy is such that particles whose energy spread relative to the reference energy is larger than 9% are rejected

After this refined comparison, also the agreement in terms of final beam loss improved, as almost all the code predicted a loss of about 2% (See Tab. 4).

PARMILA *	IMPACT	DYNAMION	PATH *	HALODYN	LORASR	PARTRAN
1.9%	1.9%	2.1%	1.3%	2.2%	1.86%	2.05%

Table 3 CASE 2: beam loss at the end of the DTL as predicted by the codes. *: internally-generated iput distribution.

In Fig. 22 it is shown how different the RMS emittances can be, depending on the particle loss definition. In the left plot a cut-off at ±180° in the longitudinal phase is introduced, providing a particle loss of 0.2% and a longitudinal emittance of 1.67 mm-mrad. In the right plot no cut-off is introduced and the few particles far from the RF bucket, despite their limited number, drive an artificial emittance blow up (15 mm-mrad).

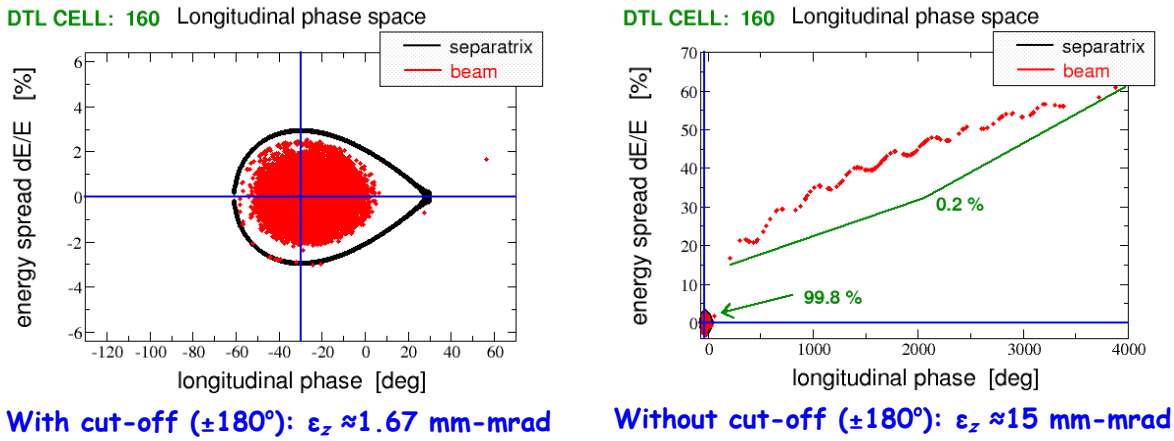


Figure 22 CASE 2: Longitudinal phase space at the exit of the first DTL tank as simulated by PATH. In te left plot a cut-off at ±180° is introduced by applying a "filter" at the end of each cell, whereas in the right plot no cut-off is introduced. This resulted in few particles (0.2%) leaving the RF bucket and driving an artificial emittance blow up.

Oulook: upon the longitudinal emittance blow up in tank 3

The right plot of Fig. 21 clearly shows an emittance blow up predicted by all the codes starting from tank 3 (≈ 30 m). A large dilution of the longitudinal phase space is accompanied to the emittance increase. It is believed that the peculiar and code-dependent behaviour after the first half of tank 3 is mainly driven by few particles close to the edge $|\Delta\varphi|\approx 180^{o}$.

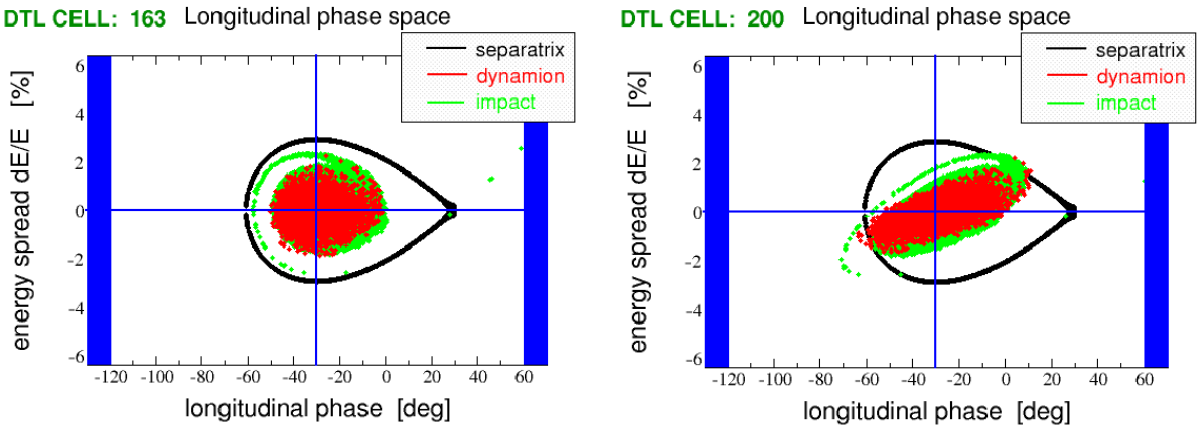


Figure 23 CASE 2: Longitudinal phase space at the exit of tank 1 (left) and at the entrance of tank 2A (right), as simulated by IMPACT (10^6 macro-particles) and DYNAMION ($5x10^3$ particles).

In tank 3 the synchronous phase jumps from -30 ^{o} to -25 ^{o}. However, this change alone cannot justify the large emittance growth (a factor three in less than five meters). A dedicated study to better understand the origin of this beam degradation was therefore undertaken. By looking at the evolution of the longitudinal phase space, it was observed that the beam enters the second tank (≈ 14 m) longitudinally mismatched, as shown in Fig. 23. The reason of this mismatch is the absence of any longitudinal focusing in the (relatively long) first inter-tank section, whose length of about 1.5 m corresponds to 8 βλ-periods (See Fig. 26). This initial mismatch drives, through space charge, a beam dilution. The longitudinal emittance experiences a weak growth in this region thanks to the large bucket area. When entering in tank 3, the phase jump and, more important, the shrinking of the bucket area, makes the bunch tails cross the separatrix, leading eventually to a even larger phase space dilution, as shown in Fig. 25.

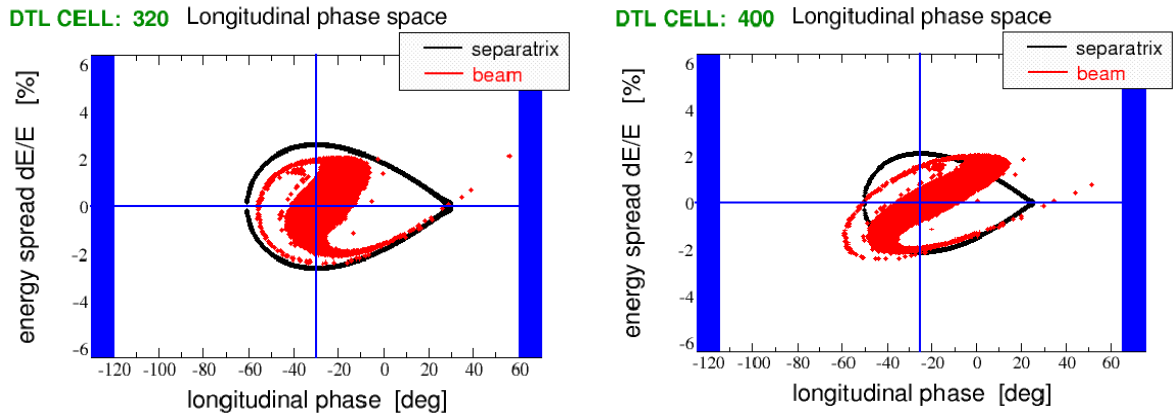


Figure 24 CASE 2: Longitudinal phase space at the exit of tank 2B with synchronous phase -30^{o} (left) and at the entrance of tank 3 with synchronous phase -25^{o} (left), as simulated by IMPACT (10^6 macro-particles).

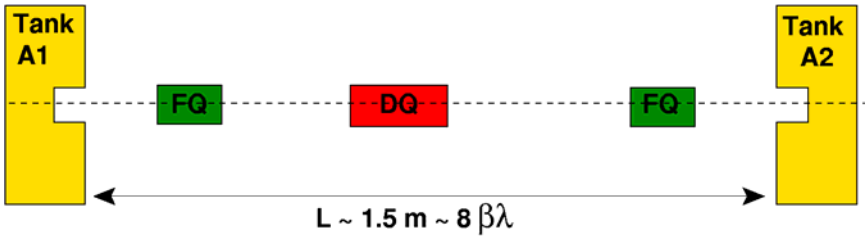
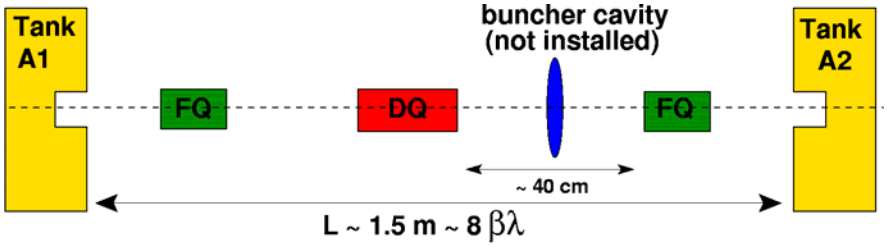


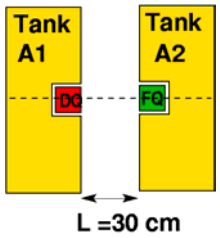
Figure 25 Sketch of the first inter-tank

Two possible countermeasures were numerically investigated in order improve the longitudinal matching and to limit the increase of the energy spread.

- Insertion of a new buncher cavity in the first inter-tank section, removing part of the beam diagnostics installed between the focusing triplet.



- Shortening of the first inter-tank section, making use of the end-quadrupoles for a transverse matching and two existing buncher cavities in the second and fourth inter-tank section (BB5 and BB6 respectively)



It was observed that both solutions are equivalent in suppressing the emittance growth, although the second one is preferable from the feasibility point of view. The second solution is also more suitable for a better transverse matching, as the present focusing triplet drives an unavoidable jump in the relative phase advance between the two tanks, which renders difficult the matching with space charge.
First it was studied the dependence of the emittance growth on the length of the first inter-tank section. In the left plot of Fig. 26 the results of this scan (performed by running HALODYN) are shown, clearly indicating how strong is the relation. For technical reasons a zero-length inter-tank section is not a practicable solution. For a scan over the gradients of the two existing buncher cavities, a length of 30 cm was used.
In the right plot of Fig. 26 the nominal emittance curve (red line) obtained with the existing lattice is compared with the one resulting from the shortening of the first inter-tank section up to 30 cm (black line), and with the curves obtained powering the two buncher cavities BB5 and BB6 at different voltages. The latter ones appear to improve the momentum spread, although for voltages higher than 500 kV no additional gain is observed. It was eventually observed that neither the shorter inter-tank section, nor the two buncher cavities affect the transverse emittances, if the end-quadrupoles of tanks 1 (G_1=-2844 Gauss/cm) and 2A (G_1=-

2243 Gauss/cm) are used. The gain in the momentum spread with this configuration is of the 30%, as the final value for the nominal lattice is $\delta w/w \approx 0.39\%$, whereas with the shorter inter-tank section and the two buncher cavities $\delta w/w \approx 0.26\%$.

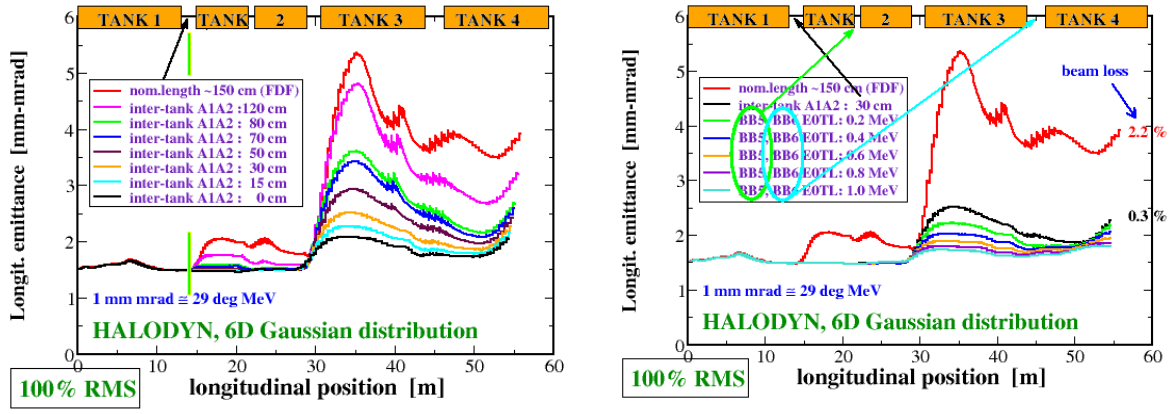


Figure 26 CASE 2 with modified lattice (HALODYN, 10^6 macro-particles). Left: scan of the longitudinal emittance against the length of the first inter-tank section. Right: scan over the voltages of the two buncher cavities BB5 and BB6 for a given length of 30 cm between tank 1 and tank 2A.

Conclusion and outlook

A series of different tests and comparisons among several codes (DYNAMION, HALODYN, IMPACT, LORASR, PARMILA, PARTRAN, PATH and TOUTATIS) have been undertaken. The benchmarking has been focused on the space-charge routines and the different lattice modelling. For this reason, in preparation of the experimental validation, the code benchmarking has been divided in three steps.

After the first static benchmarking of the space-charge routines it has been observed that

- all the space-charge solvers implemented in the above codes have the same qualitative performances in computing the space-charge electric field (see Figs. 2 and 3)
- by comparing the depressed single particle tune with analytical predictions, the PIC solvers appear to follow the same scaling law for the error propagation as function of the number of macro-particles and discretization step; the coefficients of this law are of the same order of magnitude for all the PIC codes (see Fig. 6)

After the second step consisting of tracking simulations with a zero-current beam, it has been observed that

- for Alvarez DTL structures with cylindrical symmetry, different RF descriptions do not cause differences in terms of RMS moments and beam envelopes (see Figs. 15-17)

After tracking simulations under different regimes of space charge, it has been observed that

- in terms of RMS emittances, the agreements among the codes varies from a few percent's up to 10-20% in the transverse planes (see Figs. 18 and 20)
- the evolution of the RMS emittances, for the beam and lattice parameter here investigated, is not sensitive to choice of different boundary conditions implemented in the Poisson solvers
- in the longitudinal plane, after a series of code adjustments, for short bunches the agreement is in most of the cases well confined (lower than few percent's, see Fig. 19), whereas for long bunches the longitudinal emittance is highly sensitive to the definition of particle loss (see Fig. 21)
- when simulating the reference beam of 37.5 mA with typical UNILAC beam sizes, final beam loss of about 2% are predicted by all codes. The predictions vary from

1.3% (PATH) and 2.2% (HALODYN). The regions with predicted higher losses are the same for all the codes

- The evolution of the phase space, despite the different algorithms and RF modellings, has shown a good agreement among the codes (see the HIPPI we page [1])

So far the code comparison has been carried out simulating mismatched beam, as the present lattice the matching between tanks with space charge prevented any possible global matching. Studies for a new inter-tank lattice aiming to an improved matching have been undertaken. It is expected that the agreement among the codes in terms of RMS moments increases for a matched beam.

Several bugs, mostly of them were related to the charge state Z≠1, have been found and fixed in many codes.

Acknowledgements

We acknowledge the support of the European Community-Research Infrastructure Activity under the FP6 "Structuring the European Research Area" programme (CARE, contract number RII3-CT-2003-506395)

Bibliography

[1] http://www-linux.gsi.de/_franchi/HIPPI/code benchmarking.html
[2] A. Kolomiets et al. , *DYNAMION - The Code for Beam Dynamics Simulation in High Current Ion Linac*, Proc. EPAC-98, pp. 1201-03.
[3] S. Yaramishev et al, *Development of the versatile multi-particle code DYNAMION*, Nucl. Intr.& Meth. A, Vol 558/1 pp 90-94, (2005).
[4] A. Franchi et al. , *A 3D Poisson-Vlasov Code to Simulate the Space-Charge Effects in the High Intensity TRASCO Linac*, Proc. LINAC-02, pp. 653-655
[5] G. Turchetti et al., *Accuracy analysis of a spectral Poisson solver*, Nucl. Intr.& Meth. A, vol. 561 Issue 2 (2006), pp. 223-229.
[6] J. Qiang et al., *An Object Oriented Parallel Particle- In-Cell Code for Beam Dynamics Simulation in Linear Accelerator*, Jo. of Comp. Phys., 163, 2000, pp. 434{45.
[7] R. Tiede et al., *LORASR code developement*, Proc. EPAC06.
[8] J. H. Billen, *PARMILA*, LA-UR-98-4478, 2001.
[9] R. Duperrier, N. Pichoff, and D. Uriot, in *Proceedings of the International Conference Computational Science: ICCS 2002, Amsterdam, the Netherlands, 2002* (Springer-Verlag, Berlin, 2002).
[10] N. Pichoff *et al.*, in *Proceedings of the International LINAC Conference, Chicago, IL, 1998*(Argonne National Lab., Chicago, 1998), p. 141.
[11] http://tmuetze.home.cern.ch/tmuetze/
[12] R. Duperrier, *TOUTATIS: A radio frequency quadrupole code*, Phys. Rev. STAB, 3, 2000, p.124201-06.
[13] A. Orzhekhovskaya et al., *A space-charge algorithm for ellipsoidal bunches with arbitrary beam size and particle distribution*, Proc. EPAC-04, pp. 1972-74
[14] G. Turchetti et al., Proc. ICAP-02, (15-18 October 2002, East Lansing, USA)

APPENDIX A

Electric Field benchmarking, technical details

The code-independent routine `electric_error.f` (see link in the HIPPI code-benchmarking web page[4] "used error field routine") computes $\varepsilon(r)_{rms}$, once the files containing the analytic field (`file_name1`) and the one from the code (`file_name2`) is provided. The output file is called `err.dat` and consists of three columns: r in [m], r in units of σ and $\varepsilon(r)_{rms}$. Some numerical parameters such as σ, the longitudinal depth σ_z and the cylindrical shell thickness can be change by hand in the source code.

Format of the Input and Output Files

The input particle distribution file contains N_particles rows and has the following (standard IO) format and units:

```
....
....{follow N_particles lines like ..}
....
x[m] p_x[rad] y[m] p_y[rad] z[m] p_z[rad]
....
```

where all the coordinates are relative to the bunch centre. In the HIPPI code-benchmarking web page three files containing Gaussian distributions, of 10^4, 10^5 and $5x10^5$ particles respectively, can be found.

The output file containing the electric space-charge field must contain an header line (character string) and six columns: x [m], y [m], z [m], Ex [N/C], Ey [N/C] and Ez [N/C]. (x; y; z) are the particle spatial coordinates, ($Ex;Ey;Ez$) the corresponding space-charge electric field. An example is shown below:

```
# x[m]     y[m]     z[m]    Ex[N/C]   Ey[N/C]  Ez[N/C]
....
....{follow N_particles lines like ..}
....
0.129E-2 0.678E-2 -0.309E-2 0.119E+3 0.633E+3 -0.138E+3
....
```

Units and Conversions

The input distribution files available on the HIPPI code-benchmarking web page contain the 6D particles coordinates, in units of [m] and [mrad], relative to the bunch centre. Different codes require different units in their own input files and different conversion factors between the "internal" electric field and the one expressed in units of [N/C]. For example

- **IMPACT** uses a charge distribution normalized to

$$\int \rho dv = \frac{IZ}{f_{RF}}$$

where I is the current in A averaged on a RF period, Z is the charge state and f_{RF} in Hz is the RF frequency. The transverse coordinates are dimensionless after the normalization

$$\tilde{x} = \frac{x}{x_l} = \frac{2\pi f_{RF}}{c} x$$

[4] http://www-linux.gsi.de/~franchi/HIPPI/web_code_benchmarking.html

It wasfound that the conversion factor between the "internal" transverse electric field and the one expressed in [N/C] is

$$E_{x,y}[N/C] = c^2 \times 10^7 E_{x,y}^{IMPACT}$$

Selecting the option "6" in the fourth "test.in" command line, IMPACT reads the particle distribution from the external file "partcl.data" whose format[5] is the following

```
N_particles T_s[eV] •_s[deg]
....
....{follow N_particles lines like ..}
....
x[cm] p_x[rad] y[cm] p_s[rad] •[rad] T[Mev]
....
```

where

- T_s is the kinetic energy of the synchronous particle;
- $\bullet_s$ is the initial phase of the synchronous particle;
- $\bullet$ is the "absolute" phase of the n-th particle;
- T is the "absolute" kinetic energy of the n-th particle;

"absolute" means here "in the lab frame": IMPACT routine "regen_Dist" reads the "partcl.data" and provides the "relative" longitudinal coordinates with respect to the synchronous particle.
For *x* and *y* the conversion is trivial, whereas for the longitudinal ones a more subtle conversion is required. According to Rob Ryne's notes "*The Linear Map for an RF Gap Including Acceleration*" and to the following relations

$$z = \frac{\beta c}{2\pi f_{RF}}(\varphi - \varphi_s) \qquad z' = -\frac{1}{Mc^2\gamma^3\beta^2}(T - T_s)$$

the following conversion equations from [m][rad] to [rad][MeV] can be derived

$$\varphi[rad] = \frac{2\pi f_{RF}[Hz]}{\beta c[m/s]} z[m] + \varphi_s[\deg]\frac{\pi}{180}$$

$$T[MeV] = -Mc^2[MeV]\gamma^3\beta^2 z'[rad] + T_s[eV]10^{-6}$$

Note that open boundary conditions are used, the code adjusts the mesh box around the beam, neglecting the dimensions given in the input file. In order to compare the different codes with the same grid resolution, we modified the source code and skipped the re-mesh.

- **HALODYN** uses a charge distribution normalized to $\int \rho dv = 1$ and solves the Poisson equation in the CGS system. The transverse coordinates are in mm, providing the following conversion

$$E_{x,y}[N/C] = \frac{Q_b}{4\pi\varepsilon_0} \times 10^6 E_{x,y}^{HALODYN}$$

where $Q_b = I/f_{RF}$ is the total charge of the bunch (assuming a CW mode, i.e. a bunch each RF period). Setting in the HALODYN input file the second option of the ";halodyn trk" command line to "4" , the code reads the particle distribution from the external file "INPUT/halo partcl" whose format does not require any conversion.

[5] Probably for heritage and/or compatibility with PARMILA

APPENDIX B

SPT test, technical details

On the HIPPI code-benchmarking web page the following files can be downloaded:

1. ANALYTIC_file: contains the analytical solution for the depressed SPT from a Gaussian distribution;
2. distribution_generator (f77) : is the f77 source code we used to generate 20 Gaussian distributions; it must be linked to the CERN libraries, included in the file together with a short shell script for the correct compilation;
3. tune_error.f f77 routine : is the source code used for the SPT computations;
4. IMPACT_EL.F._file_example and HALODYN_EL.F._file_example are .zip files containing examples of electric field on the grid from two codes (first lines contains parameters needed for the interpolation).

The code developer who wants to join the SPT comparison, should follow the following procedure:

1. generate three sets of 20 Gaussian distributions, respectively of 10^5, $5x10^5$ and 10^6 particles;
2. run the Poisson solver and store on files the corresponding electric fields on the grid (20x3 = 60 files, possibly with consecutive nomenclature, such as name_01, name_02 . . .);
3. create a subroutine that reads the numerical parameters needed for the interpolation (to be put in a common block shared with the interpolating routine) and the electric field (double precision -real*8- array `E(0:NPX,0:NPY,0:NPZ)`, where `NPX, NPY` and `NPZ` are the grid points). This routine must be structured like the subroutines impact_grid_read and halody_grid_read, one can find at the end of the tune_error.f source;
4. create a subroutine with the same interpolation used in the code[6] that has the same call structure of interpol_E_impact and interpol_E_halody one can find at line 133 of the tune_error.f source code ;
5. include the two subroutines in the tune_error.f source code and run it.

[6] This routine is needed since the test particle tracking needs to compute the electric field at an arbitrary position.