# AN EFFECTIVE SPACE CHARGE SOLVER FOR DYNAMION CODE

A. Orzhekhovskaya, W. Barth, S. Yaramyshev, GSI, Darmstadt, Germany

### *Abstract*

An effective analytical and semi-analytical method for internal electrical field calculations was proposed for ellipsoidal shaped beam as well as for a beam of arbitrary longitudinal shape with an elliptical transverse cross section. This method combines acceptable accuracy with a high speed of computation. The existing version of the DYNAMION code uses the particle-particle method to calculate the electrical field, which needs a significant time for computation. A Semi-Analytical Solver (SAS) for the ellipsoidal bunch was introduced into DYNAMION code. It allows much faster beam dynamics simulations than the old. The DYNAMION parameter "macroparticle size" was investigated in combination with the new space charge algorithm. The beam dynamics simulations were performed through the 1st Alvarez tank of the GSI linac UNILAC using the standard and the new methods. The RMS emittance growth as a benchmark parameter shows sufficient agreement between both solvers.

## INTRODUCTION

Fast and precise space charge solvers are especially important in the beam dynamics simulations for high current linear and circular accelerators, where space charge effects may dominate and lead to the emittance growth and beam losses. Space charge effects can be calculated using different analytical and numerical methods. Recently various modifications of the PIC solver are mainly used for the simulations. The advanced multiparticle code DYNAMION [1], dedicated to beam dynamics simulations in linacs, was created in 1985 in the Institute of Theoretical and Experimental Physics (ITEP, Moscow) and was developed in collaboration of ITEP and GSI Helmholtzzentrum fuer Schwerionenforschung (Darmstadt)

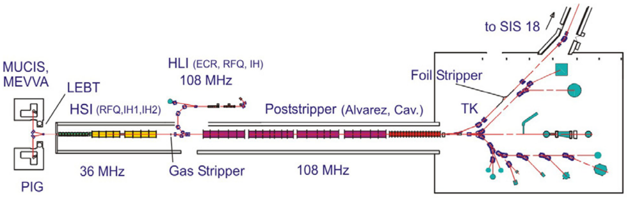

Figure 1: GSI – UNILAC.

Since 1991 the code DYNAMION is used for study, optimization and upgrade of the heavy ion high current GSI linac UNILAC (Fig. 1), serving as a high current injector for FAIR - International Facility for Antiproton and Ion Research at Darmstadt together with the synchrotron SIS 18 (Fig. 2). The UNILAC comprises high current injector (HSI; 2.2 keV/u - 1.4 MeV/u), stripper section and poststripper accelerator (5 Alvarez type tanks; up to 11.4 MeV/u) [2,3].

For electrical field calculation the code DYNAMION uses recently two methods: the particle-particle interaction and the PIC solver. An analytical and Semi-Analytical space charge Solver (SAS) was originally created for beam dynamics simulations in the GSI synchrotron SIS18 and in FAIR rings SIS100, SIS300 [4-6]. This algorithm being implemented into the DYNAMION code allows also for fast and reliable beam dynamics simulations for linacs.

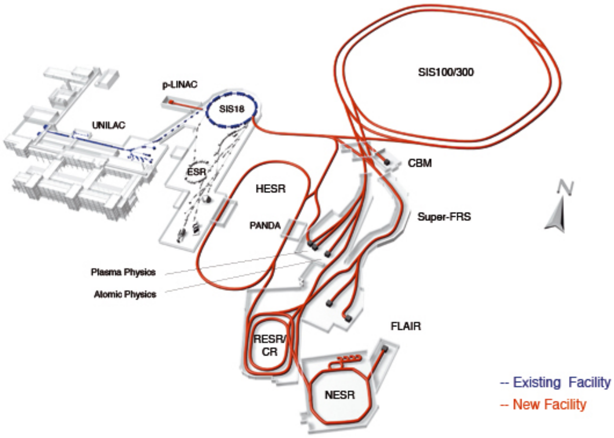

Figure 2: Scheme of GSI Helmholtzzentrum für Schwerionenforschung (Darmstadt, Germany) with existing and future facilities.

## SEMI-ANALYTICAL SOLVER (SAS) FOR THE INTERNAL ELECTRIC FIELD CALCULATIONS

### *Kellogg's Formulae*

A 3D ellipsoidal frozen bunch is considered; the charge density is given by

$$\rho(x,y,z)=\frac{Q}{4\pi abc}n(t),$$

where $Q$ - total charge of the bunch, $a,b,c$ – horizontal, vertical and longitudinal axis of the ellipsoid, $n(t)$ – analytical function, representing particle distribution, $t$ – isodensity parameter

$$t=x^2/a^2+y^2/b^2+z^2/c^2,\ 0\le t\le 1.$$

The general formulae for the electrical field of such a bunch were derived by Kellogg [7]:

$$E_x=\frac{Q}{2}x\int_0^\infty\frac{n(T)ds}{(a^2+s)^{3/2}(b^2+s)^{1/2}(c^2+s)^{1/2}},$$

$$E_y=\frac{Q}{2}y\int_0^\infty\frac{n(T)ds}{(a^2+s)^{1/2}(b^2+s)^{3/2}(c^2+s)^{1/2}},\qquad(1)$$

$$E_z = \frac{Q}{2} z \int_0^\infty \frac{n(T)ds}{(a^2+s)^{1/2}(b^2+s)^{1/2}(c^2+s)^{3/2}},$$

$$T = \frac{x^2}{a^2+s} + \frac{y^2}{b^2+s} + \frac{z^2}{c^2+s}.$$

In common case, the integrals (1) are calculated numerically for each particle. During beam dynamics simulations this process should be repeated many times on each integration step of the particle motion equation, resulting in time consumption simulations.

### Interpolation of the Particle Distribution

In the proposed method a particle distribution *n(t)*, given by an analytical formula, is interpolated as a polynom:

$$n(t) = \sum_{n=0}^{N} c_n t^n$$

using Chebyshev nodes

$$t_k = \frac{1}{2} + \frac{1}{2}\cos\frac{(2k+1)\pi}{2N+2},$$

$$k=0,1,..., N.$$

For the polynom of certain order $N$ these nodes provide the minimum absolute error of interpolation. The optimum order of the interpolating polynom was investigated and found as $N \approx 20$ [4].

### Analytical Solution for an Axisymmetric Bunch

For an axisymmetric ellipsoidal bunch (*a=b, a<c*) with a polynomial representation of the particle distribution the following transformation of standard Kellogg's formulae was proposed:

$$E_x = \frac{Q}{2} x \sum_{l=0}^{N} c_l \sum_{i+j=l} \frac{l!}{i!\,j!} r^{2i} z^{2j} I_{i+1,j}$$

$$E_y = \frac{Q}{2} y \sum_{l=0}^{N} c_l \sum_{i+j=l} \frac{l!}{i!\,j!} r^{2i} z^{2j} I_{i+1,j}$$

$$E_z = \frac{Q}{2} z \sum_{l=0}^{N} c_l \sum_{i+j=l} \frac{l!}{i!\,j!} r^{2i} z^{2j} I_{i,j+1}$$

$$I_{i,j} = \int_0^\infty \frac{1}{(a^2+\xi)^{1+i}(c^2+\xi)^{1/2+j}} d\xi .$$

Then integrals $I_{i,j}$ can be calculated analytically by using the hypergeometrical function:

$$I_{i,j} = \frac{{}_2F_1(1+i, \frac{1}{2}+i+j, \frac{3}{2}+i+j, \frac{c^2-a^2}{c^2})}{(\frac{1}{2}+i+j)c^{1+2i+2j}} .$$

Indexes *i,j* here depend only on the order of the interpolating polynom, but not on the number of particles. Thus the integrals $I_{i,j}$ can be calculated once for the whole bunch. It significantly reduces the computation time. The more particles are in the bunch, the higher is the advantage of the analytical method compare to the "standard" solvers [4].

### Numerical Solution for an Arbitrary Ellipsoidal Bunch

Using the polynomial representation of the particle distribution, the formulae (1) can be transformed to the following series:

$$E_x = \frac{Q}{2} x \sum_{l=0}^{N} c_l \sum_{i+j+k=l} \frac{l!}{i!\,j!k!} x^{2i} y^{2j} z^{2k} I_{i+1,j,k}$$

$$E_y = \frac{Q}{2} y \sum_{l=0}^{N} c_l \sum_{i+j+k=l} \frac{l!}{i!\,j!k!} x^{2i} y^{2j} z^{2k} I_{i,j+1,k}$$

$$E_z = \frac{Q}{2} z \sum_{l=0}^{N} c_l \sum_{i+j+k=l} \frac{l!}{i!\,j!k!} x^{2i} y^{2j} z^{2k} I_{i,j,k+1}$$

where

$$I_{i,j,k} = \int_0^\infty \frac{1}{(a^2+\xi)^{1/2+i}(b^2+\xi)^{1/2+j}(c^2+\xi)^{1/2+k}} d\xi .$$

The integrals $I_{i,j,k}$ are calculated numerically by Gauss quadrature with high accuracy of $10^{-5}$. It limits the error in field calculations to less than 0.1%. As before, the integrals $I_{i,j,k}$ can be computed once per integration step for the whole bunch.

### Previous Results and Implementations

A comparison of the described solvers with analytical solutions known in some particular cases, shows high accuracy of the proposed methods (error is always less than 0.1%) [4,5].

Benchmarking of the analytical method with different linac codes (*DYNAMION, IMPACT*, *LORASR*, *PARMILA*, *PARTRAN*, *PATH*, *TOUTATIS)* was performed for an axisymmetric ellipsoidal bunch ($10^5$ particles, static case) in frame of the High Intensity Pulsed Proton Injector project (HIPPI) [8]. This comparison showed good coincidence of results and demonstrated high speed of computation by the proposed analytical method (up to 15 times less CPU time).

The development of the described algorithms allows calculations of space charge forces for a beam of arbitrary longitudinal shape with elliptical transverse cross section as well [5].

All these methods were introduced into the MICROMAP library [9] and were used for beam dynamics simulations and beam loss estimations for the GSI synchrotron SIS18 and for the FAIR project.

## SAS IMPLEMENTATION INTO DYNAMION CODE

### Application of the Method on One Integration Step (Static Case)

The integration scheme in the multiparticle code DYNAMION has usually 100-200 steps per characteristic length $\beta\lambda$, where $\beta$ is the relative velocity of the particle, $\lambda$

- the wave length of the operating frequency. Space charge effects are calculated at each step of integration using particle-particle method, which needs significant computational time.

In previous simulations by SAS the particle distribution *n(t)* (given analytically) was polynomial interpolated. The code DYNAMION operates with a set of macroparticles represented by their coordinates and velocities, but not with the analytical function of the particle distribution. For the implementation of the new solver the continuous space charge density *n(t)* was the reconstructed from the discrete particles coordinates (*x,y,z*). To solve this problem we propose the following procedure on each step of integration.

All $N_p$ particles of the bunch are assumed inside the ellipsoid with the axis *a, b, c*, i.e.

$$t=x^2/a^2+y^2/b^2+z^2/c^2,\ 0 \le t \le 1.$$

The average distance between particles in terms of parameter *t* can be defined as $d=1/N_p$.

The value of function *n(t)* for a certain argument $t_0$ is defined as a number of particles with the parameter $t=x^2/a^2+y^2/b^2+z^2/c^2$ inside the interval $[t_0\text{-}kd,\ t_0+kd]$. Here *k* is a parameter which characterizes the vicinity of point $t_0$. Obviously, for different *k* the value of the function *n(t)* *(*and consequently the value of electrical field) at this point will be different.

In order to define the value of parameter *k*, the field calculation for the static case by SAS for different *k* was analyzed and compared with the p-p field calculations. The field calculation in *DYNAMION* code depends on the macroparticle size $r_{mp}$, dedicated to avoid the artificial particle collisions. Usually this parameter changes inside the *DYNAMION* code automatically in accordance with the beam size (but also can be fixed). Thus the comparison of the field calculations was done in 2 steps.

## *Step 1*

The value of function *n(t)* in the interpolation node *t* is defined inside the interval *[t-kd, t+kd]*. The electrical field $E_x^k(x,y,z)$ was calculated for each of $10^5$ test random particles *(x,y,z)* for the different value of the parameter *k* (2,3,4,5 and 6). The average and the maximum values of the relative error

$$M = \frac{| E_x^{\ k}(x,y,z) - E_x^{\ k+1}(x,y,z) |}{E_x^{\ k}(x,y,z)}$$

for all particles of the bunch in dependence on parameter *k* are presented in Fig. 3. The average error $M_{aver}$ is 2 - 3% for all values of parameter *k* and does not indicate the optimum value of *k*. The maximum relative error $M_{max}$ obviously appears for particles with coordinates close to zero while the field in the bunch center is also close to zero. Nevertheless, the value of $M_{max}$ changes systematically for different *k.* Obviously the smallest $M_{max}$ is seen for *k* = 5.

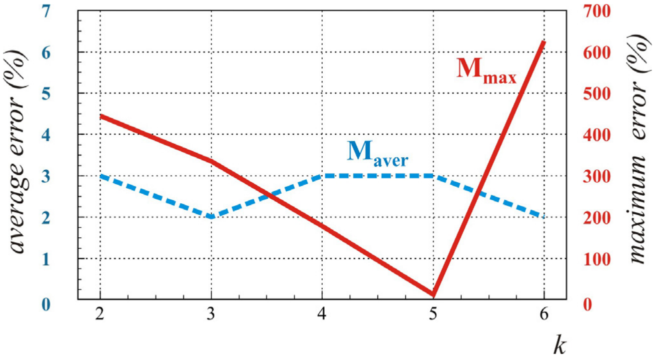

Figure 3: Average (dashed line, left scale) and maximum (solid line, right scale) values of the error (M) as a function of the parameter k.

## *Step 2*

The electrical field is calculated for all $10^5$ test particles by the *DYNAMION* code with a fixed macroparticle size $r_{mp}$ of 1 mm, 0.5 mm, 0.1 mm, 0.05 mm, 0.001 mm. Then SAS with a fixed *k* = 5 is applied to the same set of particles. We compare the results for SAS with DYNAMION computation for different $r_{mp}$. An average relative error of field calculation is about 1% for $r_{mp}$ = 0.1mm and about 3% for all others values of $r_{mp}$.

From the other side the value $r_{mp} \approx$ 0.1mm is automatically calculated for this bunch in the DYNAMION code. This comparison additionally verified coincidence of both methods and right definition of $r_{mp}$ in the DYNAMION code.

## *CPU Time Comparison for SAS and P-P Method (Static Case)*

The comparison of computational speed in the static case for the proposed algorithm and for particle-particle field calculation is presented in Fig. 4. Obviously SAS has a significant advantage for particle number above $5{\cdot}10^3$.

## *Beam Dynamics Simulations with SAS in the GSI Poststripper DTL*

The particle motion through the 1st Alvarez tank was simulated by the DYNAMION code using p-p and SAS methods. The computation scheme and integration of the particle motion equation was in both cases the same. Therefore difference in CPU time (with the same number of particle) occurs due to the different space charge solvers only.

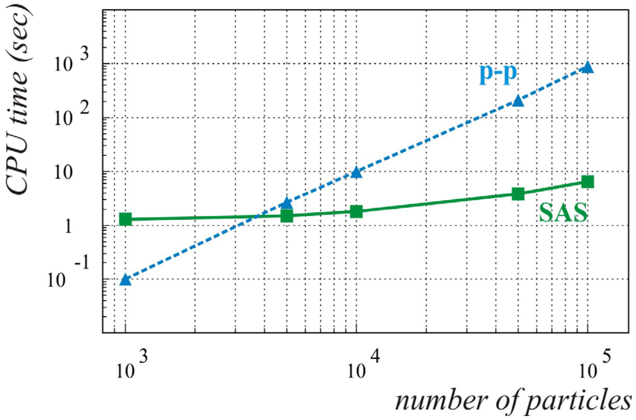

Figure 4: CPU time for p-p (dashed line) and SAS (solid line) calculations with different number of particles.

The distributions with the particle number of $10^3$, $10^4$ and $10^5$ (gaussian, truncated at $2\sigma$) were generated. These distributions with $\sigma_{x,y} = 4$mm, $\sigma_z = 8$mm represent a $U^{28+}$ bunched beam at an energy of 1.4 MeV/u. The beam current was varied from 0 up to 50 mA, while the design value is about 20 mA. Tab. 1 shows the speed of simulations for different numbers of particle using both methods. Obviously for high particle number SAS allows calculation of beam dynamics much faster than the p-p method. It allows for simulations with $10^5$-$10^6$ and even more particles with a reasonable CPU time, while standard DYNAMION simulations use $10^3$-$10^4$ particles.

Table 1: CPU Time of Simulation by DYNAMION Code for Different Number of Particles Using P-P and SAS Method

| particles | $10^3$ | $10^4$ | $10^5$ |
|---|---|---|---|
| p-p | 10 min | 48 hours | - |
| SAS | 1 hour | 2 hours | 10 hours |

The RMS emittance growth was chosen as a characteristic parameter of the calculations accuracy. More detailed investigations are recently under investigation. Fig. 5 shows the RMS transverse emittances behind the 1st Alvarez tank as a function of beam current for both space charge solvers. These tests were done for $2\cdot10^3$ particles with the p-p space charge solver and for $10^4$ particles with the SAS. These two models are comparable in accuracy and require similar CPU time.

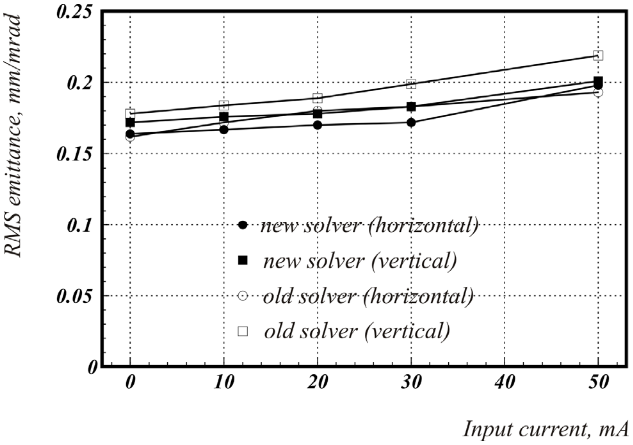

Figure 5: RMS emittance behind the 1st Alvarez tank as function of the beam current.

Just small difference might be explained by the simplification of models and by the choice of the parameters in the solvers:

- Slightly different macro-characteristics of the generated input distributions with $2\cdot10^3$ and $10^4$ particles.
- The parameter $k = 5$ is chosen on the base of the previous investigation in the static case.
- For this test the macroparticle size is fixed: $r_{mp} = 0.1$mm. Previous investigations showed a weak dependence of the results on the $r_{mp}$ parameter [1].
- For this test an integration scheme for the computing of particle motion equation has 100 steps per $\beta\lambda$.

Last issue was studied additionally. Fig. 6 shows the dependence of the RMS emittance on the number of integration steps in the DYNAMION code with p-p solver and with the SAS.

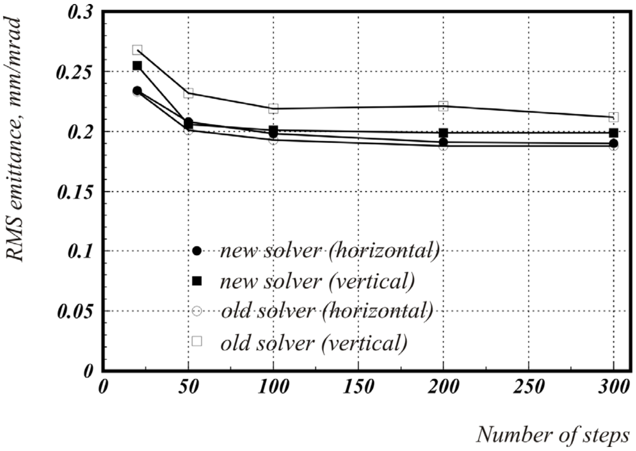

Figure 6: An RMS emittance behind the 1st Alvarez tank as a function of number of integration steps.

As already shown [1], for the reliability of simulation using p-p model the number of steps plays a more important role than the number of particles. The RMS emittance plato, also calculated with the semi-analytical solver, confirms this fact.

## CONCLUSION

A fast and precise Semi-Analytical space charge Solver (SAS) for the ellipsoidal bunch is added to DYNAMION code. A continuous space charge density function is reconstructed from the discrete particles coordinates by a polynomial interpolation (using Chebyshev nodes for higher accuracy).

For particle number above $5 \cdot 10^3$ SAS allows beam dynamics simulations much faster than the p-p method. New solver allows for the calculations with particle number up to $10^6$.

The analysis of the results shows a good coincidence between SAS and the existing DYNAMION solver, which is already proved by numerous benchmarking tests and by a comparison with measured data.

An advanced "2-step" scheme for beam dynamics simulation with DYNAMION code is already proposed. Initial investigations can be done by using the fast and reliable SAS method. Finally the results should be proved by high precision calculations with more time consuming space charge solvers.

## OUTLOOK

- A set of various tests is recently under investigation in order to optimize internal parameters of the algorithms.

- Different bunch shapes will be considered.

- A benchmarking of the beam dynamics codes DYNAMION (new solver) and LORASR is foreseen.

## REFERENCES

[1] S. Yaramishev et al., *Development of the versatile multi-particle code DYNAMION* // Nuclear Inst. and Methods in Physics Research A, Vol 558/1 pp 90-94, (2005)

[2] W. Barth, *The Injector Systems of the FAIR Project* // LINAC 08, 29 Sept. – 3 Oct. 2008, Victoria, Canada

[3] FAIR C.D.R. http://www.gsi.de/GSI Future/cdr/

[4] A. Orzhekhovskaya, G. Franchetti *A Space Charge Algorithm for Ellipsoidal Bunches with Arbitrary Beam Sizes and Particle Distribution* // EPAC 04, 5-9 July 2004, Lucerne, Switzerland

[5] A. Orzhekhovskaya, G. Franchetti, *A Space Charge Algorithm for the Bunches of Elliptical Cross Section* // ICAP 06, 2-6 October 2006, Chamonix, France

[6] G. Franchetti, I. Hofmann, P. Spiller, A. Orzhekhovskaya *Intensity and Bunch-Shape Dependent Beam Loss Simulation for the SIS100* // PAC 05, 16-20 May 2005, Knoxville, USA

[7] O.D. Kellogg, *Foundation of Potentional Theory*. - Dover Publications, New York, USA, 1953, p.192

[8] A. Franchi, et al., *Linac Codes Benchmarking for the UNILAC experiment* // LINAC 06, 21-25 August, Knoxville, USA

[9] MICROMAP library http://www-linux.gsi.de/giuliano/