SIMULATIONS OF BEAM INJECTION AND EXTRACTION INTO ION SOURCES

M. Cavenago*, INFN-LNL, Legnaro, Italy

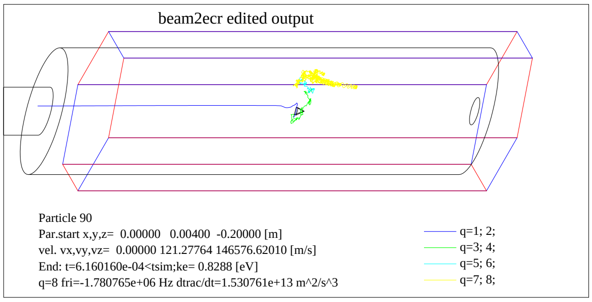
Abstract

Charge breeding, the injection of singly charged ions into Electron Cyclotron Resonance Ion Sources (ECRIS) to extract a highly charged ion beam, is a promising technique for producing beam of rare ions. Efficiency and extracted beam temperature are dominated by the strong collisional diffusion of ions. A program, named 'beam2ecr', simulating details of the injection, ionization, collision and extraction processes is described. A model of the injection plasma sheath and of source fringe fields is now included and neutral injection is also supported. Results, clearly favouring near axis injection for most cases, are described. Code is written in C-language and with possibility of parallel execution over a Linux cluster.

INTRODUCTION

Motion of ions inside an ECRIS (Electron Cyclotron Resonance Ion Source [1]) is dominated not only by the externally applied magnetic field (producing an ion cyclotron frequency $\Omega_i = ie|B|/(Am_u)$ where i and A are the charge and nucleon number) and by a small electric potential ϕ_a, named ambipolar [2] (that must be postulated to account for ion confinement and ion losses, see later), but also from ion-ion collisions, with frequency $\nu_i \propto i^2 E^{-3/2}$ with E the ion kinetic energy. For thermalized ions with $E \cong 1.5 T_b$ where T_b is the plasma ion temperature (in energy units), typically $\nu_i > \Omega_i$ for $i > i_1 \cong 15$. The need of a 3D collision and tracking code (called microscopic, because it avoids any drift center approximation) to check estimates of confinement times τ_i is evident. Moreover emission of a highly charged ion from ECRIS is most likely to follow a collision producing a suprathermal ion $E \gg 1.5T_b$; so a full understanding of extracted beam emittance also requires a 3D collision code. As an illustration of moderate complexity, a particle (here ^{107}Ag) track is shown in Fig 1.

The injection of ions inside an ECRIS [3, 4] is considered as a possible step (named charge breeder) of a RIB (radioactive ion beam) facility: ions are produced with charge $i_i = 1$ (or 2) in a simple source, mass separated and only interesting isotopes are injected into an ECRIS plasma, made mostly from a buffer gas B, where they are ionised up to charge state i_n reasonable for rapid reacceleration (to fix ideas, $i_n = 17$ for silver and $i_n = 22$ for lead). Beam trapping also depends on collision. The case $i_i = 0$ is also important in several ECRIS: neutrals are expelled with considerable kinetic energy (order of 100 eV) from Mevva ion sources, and ECRIS trapping requires high density plasma [5, 6]; slow neutral beams are emitted from ovens.



```
Particle 90
Par.start x,y,z=  0.00000   0.00400  -0.20000 [m]
vel. vx,vy,vz=  0.00000 121.27764 146576.62010 [m/s]
End: t=6.160160e-04<tsim;ke= 0.8288 [eV]
q=8 fri=-1.780765e+06 Hz dtrac/dt=1.530761e+13 m^2/s^3
```

Figure 1: Track of a ^{107}Ag ion, starting with charge $i_i = 1$, $E = 12003$ eV and stopping with $i = 8$ at walls; source potential $V_2 = 12$ kV; hexapole inner faces shown

.

Some preliminary results of 3D collisional montecarlo codes were reported elsewhere [7]. Further progresses are described later: precision of ionization routines to treat neutral injection and of tracking routines (requested by the deceleration sheath model for injection of charged particle) was implemented in version 2.9 of 'traj00'. A new version including parallel execution options is named 'beam2ecr' v3.0.

BASIC EQUATIONS

The injected beam current I_1 is typically small (100 nA); the condition of the overall ECR plasma, named the background plasma, [ion density $n_i(Z_b, A_b)$ for each charge i and ion atomic number Z_b and nucleon number A_b, external potential ϕ_e, ambipolar potential ϕ_a] is thus slightly perturbed and it will be assumed known, from measurement or as a hypothesis.

Collisions produce an average acceleration (or a deceleration) $\mathbf{A}$ and some randomly distributed kicks, that can be represented (approximately) by a Fokker-Planck equation with diffusion tensor D or by a set of finite increments of velocity (named kicks) $\mathbf{K}_j$ at some times $t = t_j$ that satisfies the statistical property:

$$< S^\alpha \, S^\beta >= t_d \, D^{\alpha\beta} \quad , \quad \mathbf{S}(t_d) = \sum_{t<t_j<t+t_d} \mathbf{K}_j \tag{1}$$

with $<>$ the statistical average and t_d an arbitrary time period [7, 8]. This gives a motion equation

$$\mathrm{d}_t \mathbf{v}_a = \frac{ie}{Am_u}(\mathbf{E}+\mathbf{v}_a \times \mathbf{B}) + \frac{\mathbf{f}}{Am_u} + \mathbf{A} + \sum_j \mathbf{K}_j \delta(t-t_j) \tag{2}$$

where $\mathbf{v}_a$ is the test particle velocity, $\mathbf{E}$ and $\mathbf{B}$ are the stationary electric and magnetic field, $\mathbf{f}$ represent pondero-

[*] cavenago@lnl.infn.it

0-7803-8859-3/05/$20.00 ©2005 IEEE

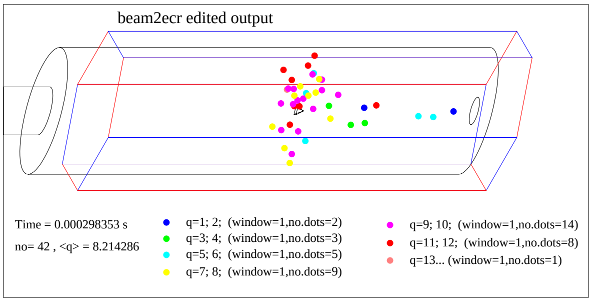


Figure 2: Trapped ^{107}Ag ions, starting with $i_i = 1$, $N = 160$, $V_b = 3$ V, $V_s = 12$ kV, $t_o = 0.298$ ms

motive effects of high frequency fields ($\omega > \nu_{i_n}$) and of plasma fluctuation and instability (yet to be included).

The applied magnetic field (adjustable for source optimisation) is $\mathbf{B} = w_1\mathbf{B}_{s1} + w_2\mathbf{B}_{s2} + w_3\mathbf{B}_h$ where w_j are weights, $\mathbf{B}_h$ a given hexapole field (or a multipole), and $\mathbf{B}_{s1}(r,z)$, $\mathbf{B}_{s2}(r,z)$ are azimuthally symmetric fields; typically $w_3 = 1$ because the hexapole is made of permanent magnets, $w_1 = 1$ because $\mathbf{B}_{s1} \cong 1$ T is the source optimal condition and $w_2 = 0$. Let $2R_h$ and L_h be the hexapole aperture and length, L_s the solenoid length, $z = 0$ be the solenoid center and $z = z_s$ the hexapole center.

Code Physics Improvements

Eq. 2 is simulated with a 4th order Runge-Kutta with variable time step dt, small enough so that: a) $dt\,\Omega_i < 0.05$ and similarly for ionization frequencies; b) $dt\,\nu_i < 0.05$; c) field $\mathbf{E}$ and $\mathbf{B}$ changes smoothly; d) kicks satisfies $||Am_u\mathbf{v}\cdot\mathbf{S}_j(\,dt\,)|| \not\gg T_b$, that is average energy gain in a dt is not larger than thermal particle energy; e) $dt > 2$ ps, to avoid program stall. Interpolation of field $\mathbf{E}$ from a grid [in cylindrical coordinates (ϑ, r, z) with corners $(0,0,z_l)$ and (ϑ_h, r_h, z_h)] of stored data must be taken with care, due to condition c: the nearest node to a particle (or the interpolation origin node) changes by discrete steps when a particle moves, and this lead to discontinuity of $\mathbf{E}$ or $\mathbf{B}$. The code locks to a given node for each Runge-Kutta step. While in 'traj00v1.1' the field $\mathbf{B}_h$ was computed from the permanent magnet four times each step dt, now also the hexapole field is interpolated from data, stored in a initialisation phase, for the sake of integration speed.

For $\mathbf{B}_s$ a fringe field model was added, since $\mathbf{B}_s$ may be still significant for $|z| \cong L_s$, due to partial saturation of iron yoke (while hexapole field region is more limited, say $-z_l = z_h = R_h + \frac{1}{2}L_h$ and $\vartheta_h = \pi/3$ exploiting symmetry). Only values near the z-axis (say for $r < r_h/2$) are necessary for tracking beam injection. A satisfying fit formula found after many tries is

$$A_\vartheta(r,z) = c_0 M(r,z;m,z_0) + \sum_{i=1}^{2} c_i R(r,z;R_i,z_i) \quad (3)$$

for $r < r_h/2$, where $M = \frac{1}{2}r(r^2 + (z-z_0)^2)^{-1-(m/2)}$ is a multipole and $R = \frac{1}{2}r(R_i^2 + (z-z_i)^2)^{-3/2}$ represents a current ring; $B_r = -\partial_z A_\vartheta$ and $B_z = r^{-1}\partial_r(rA_\vartheta)$. In practice R_1 and R_2 and m must be guessed manually (for example 70 mm, 100 mm and $m = 3.5$), while for the other six parameters c_i, z_i a good fitting routine usually converges. Model for ϕ_a are based on reasonable guesses with some parameters, of which the more physically significant are: $2Z_p$ length of ambipolar potential with depth Φ_1, $L_s \cong 10$ mm presheath length, Φ_{11} radial potential drop, R_p its radius; implementation of ever new options is in progress and a sheath model was added. Near the injection tube there is indeed a deceleration sheath; let $V_2 > 0$ be the potential of the ECRIS plasma with respect to this tube (and V_1 the potential of the first source plasma; the bias voltage is $V_b = V_1 - V_2$). Sheath has a width $2h_s$ and an end surface (where the quiet plasma begins) intersecting the z-axis at a point z_p with a curvature C_p. Present code assumes a flat sheath with a field E_z continuous up to its first derivative:

$$E_z = \frac{15V_2}{16h_s}(1-\xi^2)^2 \qquad \xi = \frac{z - z_p - h_s}{h_s} \qquad (4)$$

for $z_p - 2h_s < z < z_p$ and $E_z = 0$ elsewhere. It seems reasonable to keep $h_s \geq 2$ mm for numerical precision.

Table 1: Trapped Particles N_t (With Error as $2\sqrt{N_t}$) After 10 ms vs Starting Position x_i and y_i , With $i_i = 1$. Other Parameters as in Fig. 3; Total CPU Time 16 h (Plus I/O) Distributed on 25 Processors in 1 h Run Time

$y_i \backslash x_i$	0	3 mm	6 mm	9 mm	12 mm
0	176	171	143	133	82
3 mm	155	151	155	134	96
6 mm	138	146	121	127	82
9 mm	119	115	107	113	69
12 mm	31	68	58	70	60

SIMULATIONS

A number N of particles (non interacting) is started (in other words, injected) and concurrently evolved by the code. Outputs are grouped in three categories: verification (of fields, of random numbers, etc.), illustration (the track of some particles) and physically meaning average quantities (including the observed confinement time τ_i, time evolution of average energy, position, velocity, their correlations, phase plot of the extracted beam and exit plot of lost particles, scatter plots of trapped particles at given observation times t_o, see Fig 2). Output comes in two forms: evolution reports in form of tables, with several tables for each text formatted files, and graphic files, with pictures, graphs, histogram encoded by internal routines into the well-known Fig3.2 format [9], that is a text file.

Source code is written in C-language (ANSI), it is self contained and was tested under Linux (RedHat 7.3). Graphical output can be displayed when ready, invoking the well known program Xfig[9], or off-line (a re-play command file is available), according to user needs.

User input consists of three compulsory files plus some command line options (for example, switch -f references a

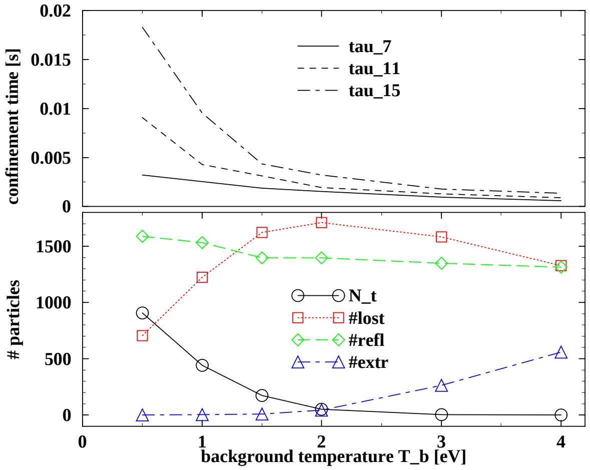


Figure 3: Number of trapped particles N_t after $t_s = 10$ ms (and in the upper panel, confinement times τ_i for charge states i) vs T_b. Here $N = 3200$, $\Phi_1 = 0.89$ V, $i_i = 1$, initial kinetic energy $E_i = eV_s + 2$ eV with source potential $V_s = 12$ kV, position $\mathbf{x}_i = (1, 4, -190)$mm, ECR plasma in $|z| < 138$ mm with ion temperature $T_b = 1.5$ eV.

file with the filenames of the other three, switch -m modifies some parameters for a long batch). One file contains the $\mathbf{B}_{s1}$ data (two tables). Another file gives data for the the hexapole field. Third file is structured in namelists, giving the geometry, the particle start points, the plasma background, directives for the integration time and the observation times, and the output options (colors, viewpoints, etc.). Namelists were implemented by defining routines on globally visible structures.

The most CPU consuming tasks are parametric studies of some output quantity, say τ_i with $i = 15$, as a function of some key inputs, say the background ion temperature T_b and the potential depth Φ_1. A continuation run option (-c) is thus provided, and an additional output file, the summary, gives a table of changed parameters and results, as selected by the user in another input file.

It is possible to execute the continuation runs on different processors, for example on a cluster of Linux systems [10], as installed at LNL. The main code spreads the computation between several processors (option -p), each one taking care of one simulation with a given set of parameters and with proper syncrhonization of the summary. If no cluster is available, still a continuation run may be performed by concurrent processes on the same machine, or by a single sequential process (the traditional way). Some simple point must be emphasised. Avoiding I/O conflicts proved difficult, but feasible. Most important, the results of a parallel run should be consistent with the rules of input priority for the sequential run, which are: first apply the program defaults, then apply the user input for the generic case (initial position, angular spread, source magnetic field maps and hexapole data); then apply the precision modification requests (option -m), finally apply the parameters to be plotted (last change is the higher priority change).

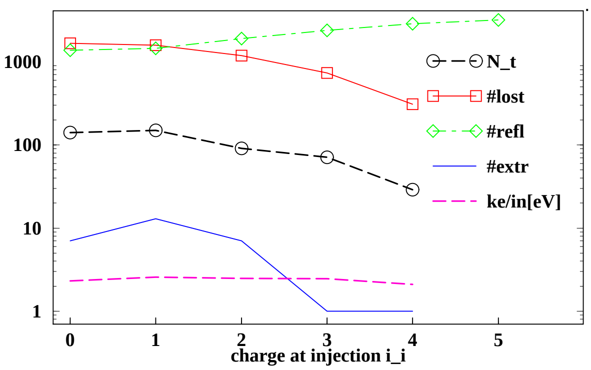


Figure 4: Number of trapped particles after $t_s = 10$ ms vs i_i (fast neutral or ions) with initial kinetic energy $E_i = i_i e(V_b + V_s) + 1$ eV with bias $V_b = 1$ V; other condition as in Fig. 3. With $i_i = 5$ all the 3200 particles are reflected.

The confinement times τ_j of ions with charge j depends on the trap potential ϕ_1, the background temperature T_b as verified by Fig. 3 simulations, each using $N = 3200$ particles; run time was about 1.8 hours, for a total CPU time of 4.4 hours distributed on 6 processors. Note that an higher temperature favors extraction, even if it depresses τ_j, and overall trapping is poor due to the the strong hexapole (1.5 T), off-axis injection and small trap ϕ_1.

In Fig. 4 we see that an higher i_i is not beneficial for off-axis injection, even if the plasma stopping power increase, since magnetic field may cause beam reflection for small bias potential. Neutrals and single charged ions are still trapped with some efficiency, and $< E >$ well fits the $\frac{3}{2}T_b = 2.25$ eV equilibrium value.

In table 1 a 2D scan of injection position is given, avoiding off axis displacement greater than 15 mm, with negligible trapping as known from preliminary simulation ($N = 160$). The good parellization efficiency is due to the fact that each simulation required comparable times. In general, to obtain this favourable condition, N should be larger for simulations with larger particles losses.

REFERENCES

[1] R. Geller, *Electron Cyclotron Resonance Ion Sources and ECR Plasmas*, Institute of Physics Publ., Bristol, 1996.

[2] H. I. West, UCRL-53391, (1982), NTIS Springfield, VA.

[3] U. Koster, O. Kester , D. Habs, Rev. Sci. Instr., **69**, 1316 (1998).

[4] P. Sortais et al., Rev. Sci. Instrum., **71** (2000), 617.

[5] M. Cavenago et al., Rev. Sci. Instrum., **69** (1998), 795.

[6] A. V. Vodopyanov et al., Rev. Sci. Instrum., **75** (2004), 1888.

[7] M. Cavenago et al., Rev. Sci. Instrum., **73** (2002), 537.

[8] F. L. Hinton, in *Handbook of Plasma Physics* (eds. M. N. Rosenbluth, R. Z. Sagdeev), vol. I, p. 147-197.

[9] S. Sutanthavibul et al., http://www.xfig.org/userman

[10] S. Badoer et al., Annual Report 2003 LNL, INFN-LNL(REP) 202/2004, 181.

0-7803-8859-3/05/$20.00 ©2005 IEEE