

Real-time beam profiles using SIMION and gnuplot

Jimmy RANGAMA

Centre de recherche sur les Ions, les MATériaux et la Photonique - CIMAP

rangama@ganil.fr

Simion user meeting, GANIL, 11/06/2015

Introduction

In the following example, SIMION 3D was used to design and characterize optical elements for the transport and shaping of singly charged ion beams in an energy range up to 20 keV. Ions are produced by an ECR (Electron Cyclotron Resonance) source built in 2002 by the GANIL source team and named MONO1000. This ECR source is able to provide high intensity (about 1.5 mA for He^+ at 18 keV) ion beams which are mandatory for our experimental set-up called PIBALE (*Plateforme d'Irradiation de Biomolécules et d'Agrégats Libres et Environnés*). With this set-up we are doing cross-beam collision experiments between ions and complex molecular ions. In order to obtain the maximum transport efficiency, the highest beam intensity and the best overlap between the two beams in the interaction region, we need to monitor the projectile ion beam at different positions in the beam line. Due to some mechanical constraints, particularly in the interaction region, it is more convenient to monitor the beam in simulations than in the experimental set-up. In the simulation the beam is monitored by using two virtual profilers and the obtained profiles can be visualized in real-time by using the graphing freeware gnuplot.

Beam line description

The beam line, from the ECR source exit to the interaction region, has been designed from SIMION simulations (see Fig. 1). The beam line is composed by an extraction triode (a), a commercial Wien filter (c), two einzel lenses (b and c) and a quadrupole triplet (e). Ions data (the coordinates ion_py_mm and ion_pz_mm , depending on the user's requirements) are recorded at the interaction region (ZI profiler) and at the Faraday cup (FC profiler) positions and are printed into two different ASCII files (`resultsZI.dat` and `resultsFC.dat`). An example of the file format is given in the LUA code. First, a file has to be created in writing mode, then the X position where we want to record the ion data is defined and finally, the 3-columns data are written in the file. In order to read this file with gnuplot, one should be aware to define the same data separator in the LUA code.

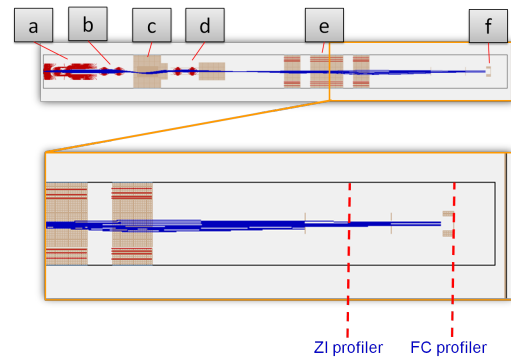


Figure 1: Simulation of MONO1000 Beam line.

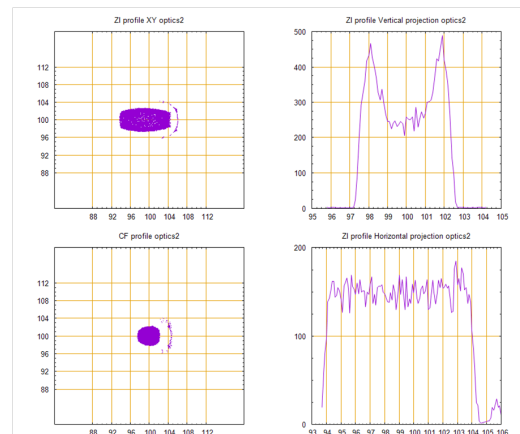


Figure 2: Typical plots obtained with gnuplot.

How to plot data with gnuplot

Gnuplot (<http://www.gnuplot.info/>) is a portable command-line driven graphing utility which works on multiplatform. The program is freely distributed even though it is not distributed under GNU General Public Licence. You do not have to pay for it. It is designed primarily for the visual display of scientific data. It can generate 2 and 3 dimensional plots of function and data. Initial release was in 1986. The last stable release is gnuplot 5.0.0 available since January 2015.

Here, we use a gnuplot script to read and plot the SIMION recorded data. The gnuplot script is an ASCII file loadable directly with gnuplot (by using `load 'live-plot.plt'`, for example). The first part of the script corresponds to the settings of the plot (tics, grid,...). In the second part, the function `bin(x,width)` is defined to generate an histogram. Then, by using `plot`,

one can plot a 2-D profile of the beam or/and a 1-D histogram profile (Fig. 2). Finally, the *reread* function allows to make an infinite loop of this script and then gnuplot re-reads the data files in real-time. With *pause 1*, we define the display frequency (here, every second). By using *pause -1*, gnuplot will open a message box which will allow the user to stop the infinite loop. To conclude, all recordable data (kinetic energy, TOF, angles, etc...) provided by SIMION could be plotted in real-time by using this process and more than 2 profilers could be used.

LUA Code

```
simion.workbench_program()

adjustable extract_plasma = 7000
function segment.init_p_values()
    if ion_instance == 1 then
        adj_elect01 = extract_plasma
        adj_elect02 = extract_e1
    end
end

local fileZI = assert(io.open("resultsZI.dat", "w")) -- write mode

function segment.other_actions()
    num_particles = ion_number

    if ion_px_mm > 2435 then
        if ion_px_mm < (2435+ion_time_step*ion_vx_mm) then
            TOF_ZI = ion_time_of_flight
            print('TOF_ZI', ion_time_of_flight)
            print('diffTOF(ZI-CHOPPER)', TOF_ZI-TOF_CHOPPER)
            fileZI:write(ion_px_mm.. " " ..ion_py_mm.. " " ..ion_pz_mm.."\n" )
            end
        end
    end
end

function segment.terminate_run()
    fileZI:close()
end
```

gnuplot script : liveplot.plt

```
reset

set multiplot layout 2,2
set xrange [80:120]
set yrange [80:120]
set size square
set xtics 88,4,112
set ytics 88,4,112
set mxtics 4
set mytics 4
set grid xtics lt 0 ls 100
set grid ytics lt 0 ls 100

unset key
bin(x,width)=width*floor(x/width)
set title "ZI profile XY"
plot 'resultsZI.dat' u 3:2 w d #points pointtype 7 pointsize .1

unset key
set grid
#set xrange [80:120]
#set yrange [80:120]
set autoscale
set ytics 100
set xtics 1
set title "ZI profile Vertical projection"
plot 'resultsZI.dat' u (bin($2,0.1)):(1.0) smooth freq w l
pause 1
reread
```