Simulation Efforts for the Gamma Factory Proof-of-Principle Experiment at the SPS

Alexey Petrenko (Budker INP)

Gamma Factory meeting in Krakow, 28 Jan. 2019

Gamma Factory Theory and Simulation efforts:

Evgeny Bessonov's Note:
Latest MS Word version is available at our OneNote Page at CERN.
Latex version (in slow progress… to be greatly expanded soon):
https://www.overleaf.com/read/gpjksbkrxqcx

Simulations:

1) My python-based code: https://anaconda.org/petrenko/psi_beam_vs_laser.
2) Camilla Curatolo's code.
3) Wiesław Płaczek's modification of CAIN code.
4) RH code.
5) Semi-Analytical calculations by Aurelien Martens.

Mattermost channel on GF simulations:
https://mattermost.web.cern.ch/gammafactory/channels/simulation-tools

At this stage we've made some preliminary benchmarking and have an agreement within a factor of two or better between several simulations. The idea is to do more accurate benchmarking once we define the realistic laser beam.

My python-based simulations:

Earlier work:

Longitudinal dynamics of a single ion and an ion bunch in the LHC/SPS + laser:
https://indico.cern.ch/event/668097/contributions/2796070/
http://www.inp.nsk.su/~petrenko/misc/ion_cooling/animations/
https://apetrenko.blob.core.windows.net/misc/Li_like_Pb_cooling.html

GF PoP Experiment (work in progress):

Laser intensity estimates and Monte Carlo scheme for laser-ion bunch interaction region:
https://anaconda.org/petrenko/psi_beam_vs_laser

Recent progress: photon output, saturation effect.

Next steps:

+ **realistic laser parameters**,
+ parallelization
+ release code as a python library
+ collective effects via longitudinal and transverse impedance
+ intra-beam scattering/stripping
+ …?

Monte Carlo scheme:

1) Define the ion beam

```
X = np.matrix([
  x ,
  xp,
  y ,
  yp,
  z ,
  dp
])
```

2) Define the 1-turn transformation (matrix + RF-cavity as a function)

```
M = np.matrix([
    [ R11,  R12, 0  , 0  ,  0  ,  R16],
    [ R21,  R22, 0  , 0  ,  0  ,  R26],
    [ 0  ,  0  , R33, R34,  0  ,  0  ],
    [ 0  ,  0  , R43, R44,  0  ,  0  ],
    [-R51, -R52, 0  , 0  ,  1  , -R56+L/(gamma_0*gamma_0)],
    [ 0  ,  0  , 0  , 0  ,  0  ,  1  ]
])
```

3) Define laser beam

Then turn-by turn simulation is defined in a simple loop:

```
for turn in range(0,100000):
    Excited = ExciteIons(X)           # ion excitations
    X = EmitPhotons(X, Excited)       # photon emissions
    X = RFcavity(X, h, eVrf, phi0)    # beam goes through RF-cavity
    X = M*X                           # 1-turn matrix
```

Maybe it already makes sense to publish these functions as a python library.

Saturation effect

$$N_{exc} = \frac{dN_{\hbar\omega}}{dS}\bar{\sigma} = \frac{2N_{\hbar\omega}}{\pi w^2}\exp\left(-\frac{2(\mathbf{r}-\mathbf{r}_l)^2}{w^2}\right)\bar{\sigma},$$

where $\bar{\sigma}$ is the excitation cross-section averaged over the laser frequency distribution. And as we've seen already $\bar{\sigma} \ll \sigma_0$.

In our model here we assumed that if $N_{exc} \ll 1$, then N_{exc} is the probability of ion excitation. For $N_{exc} \sim 0.1$ we should take into account the saturation effects. This can be done by solving the rate equation

$$\frac{dP_2}{dt} = m_2 P_1 - m_2 B P_2,$$

where P_2, P_1 are the probabilities of ion to be in the excited and non-excited state ($P_1 + P_2 = 1$). The rate of excitation events $m_2 P_1$ is proportional to the population of lower level P_1, photon density, and the absorption cross-section, $m_2 B P_2$ is the rate of stimulated emission events ($B = g_1/g_2$). Before ion enters into the laser pulse $P_1 = 1$ and $P_2 = 0$.

To solve the rate equation we need to separate the variables:

$$\frac{dP_2}{1-(1+B)P_2} = m_2 dt.$$

m_2 depends on time during the passage of the ion through the laser pulse, but we already know the answer in the case of small $P_2 = N_{exc} \ll 1$, hence

$$\int_t m_2(t)dt = \frac{dN_{\hbar\omega}}{dS}\bar{\sigma}.$$

(In the general case of a long laser pulse maybe we would need to keep the P_2 number during the slice-by-slice integration of ion motion through the laser pulse).

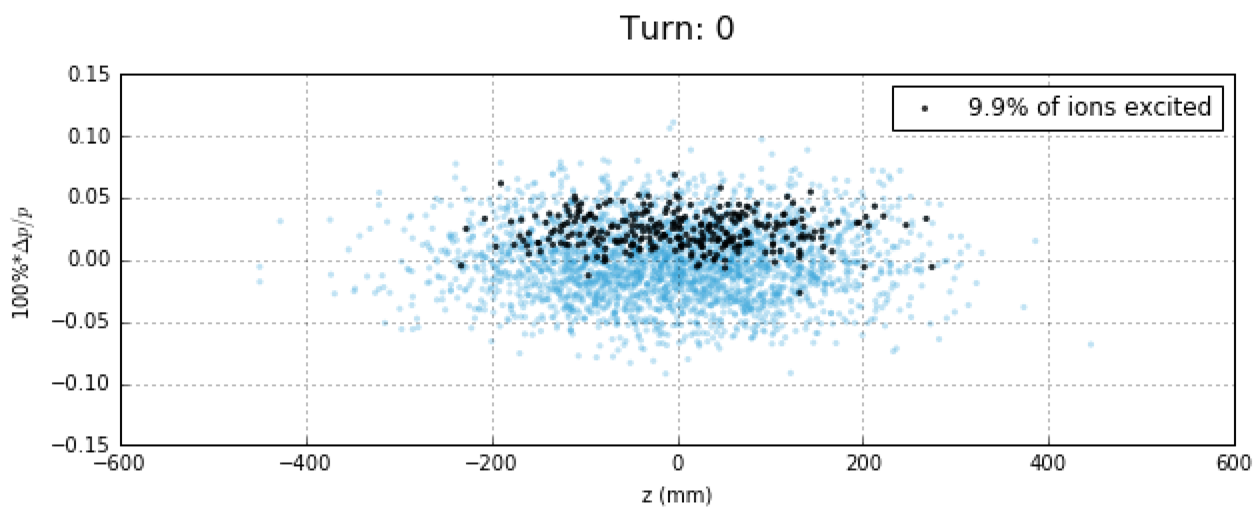
Therefore the result of integration

$$N_{exc} = P_2 = \frac{1-\exp\left[-(1+B)\int_t m_2(t)dt\right]}{1+B} = \frac{1-\exp\left[-(1+B)\frac{dN_{\hbar\omega}}{dS}\bar{\sigma}\right]}{1+B}.$$

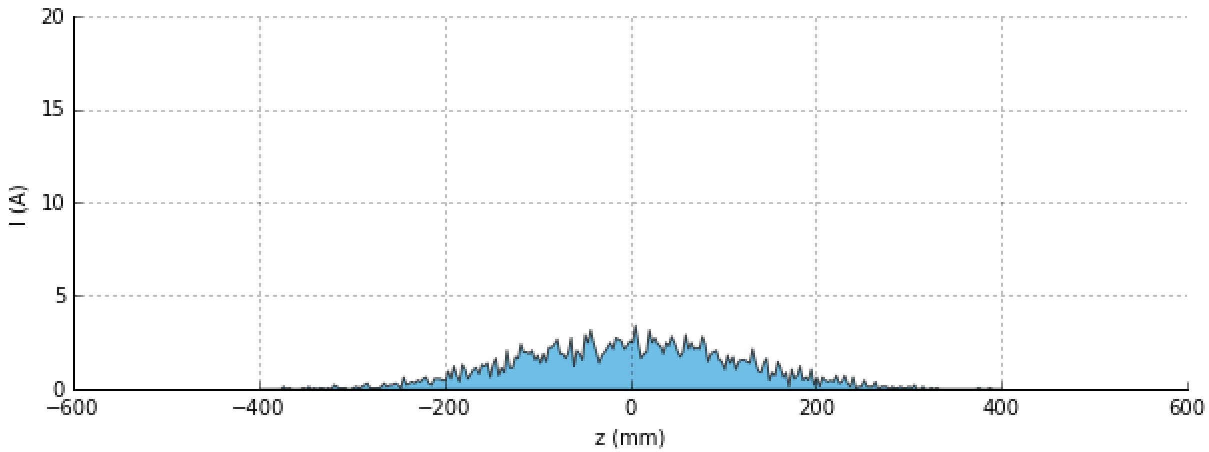
As we can see while $\frac{dN_{\hbar\omega}}{dS}\bar{\sigma}$ is small it equals N_{exc}, while for large $\frac{dN_{\hbar\omega}}{dS}\bar{\sigma}$ the result is limited by $1/(1+B)$.

Cooling over 90 sec (2 degrees crossing angle):

A

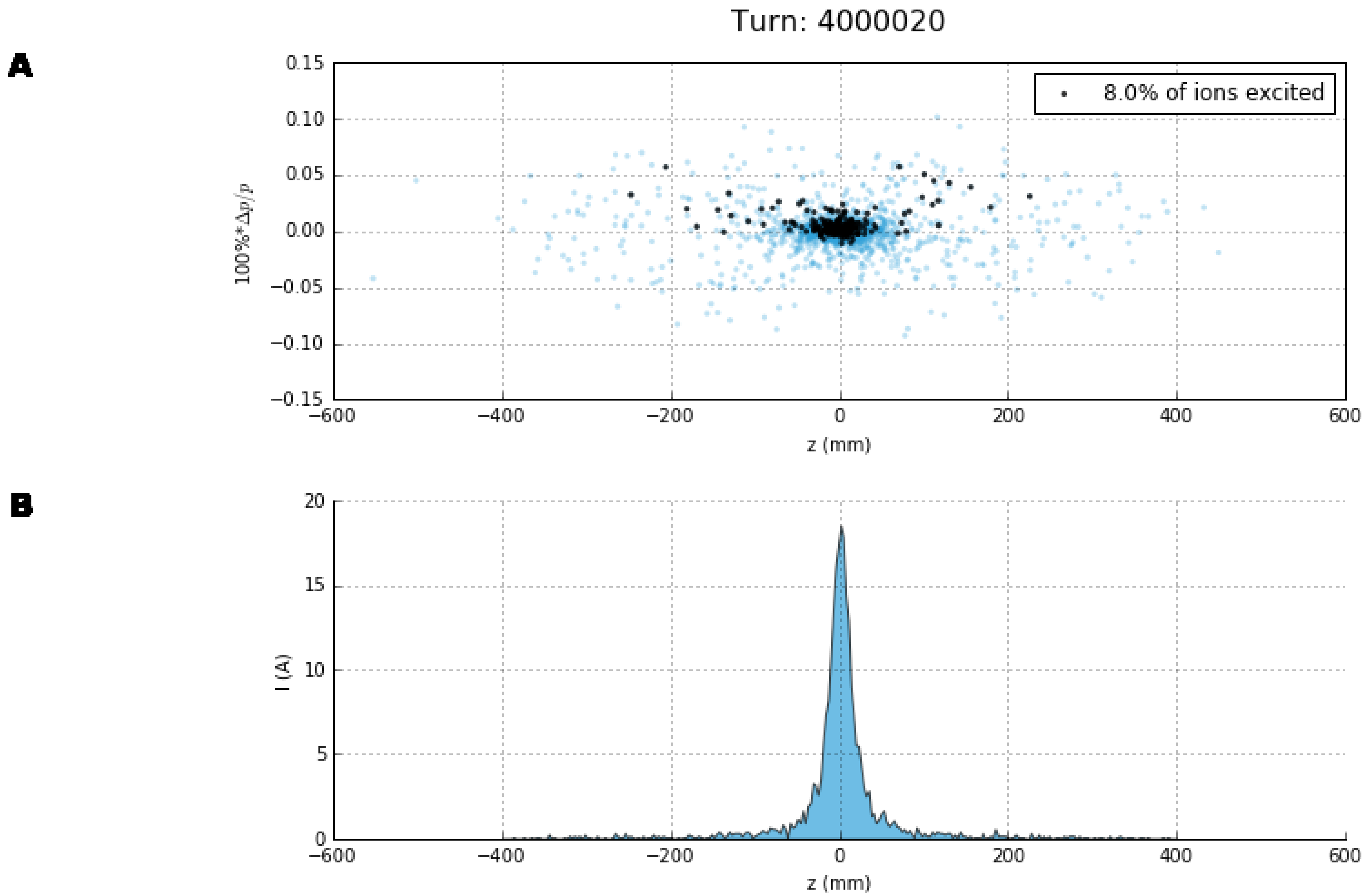


B



With 6 degrees crossing angle the number of excited ions is 2x less.

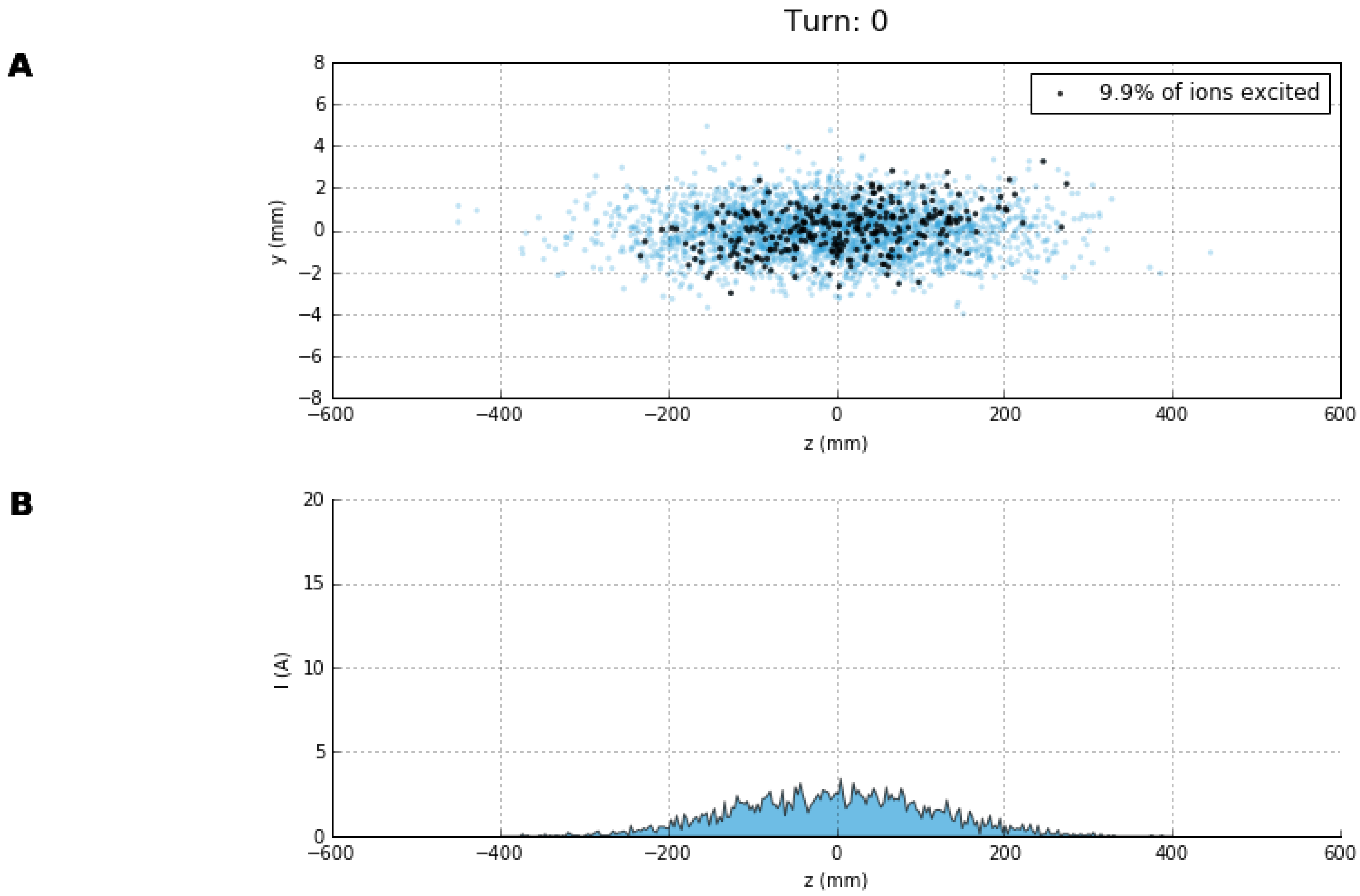
Cooling over 90 sec (2 degrees crossing angle):



Calculation time: 1.5 hours on my laptop.

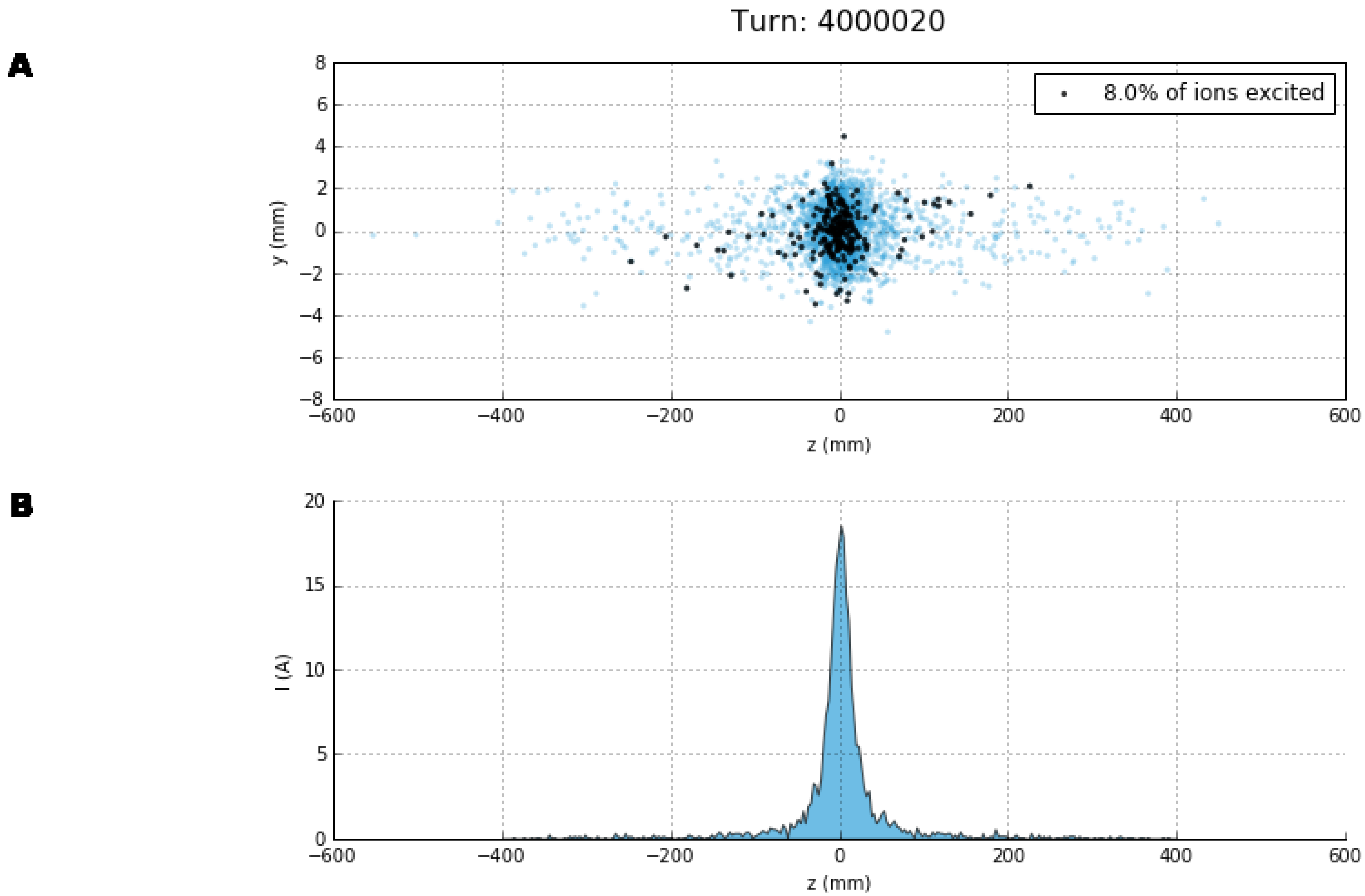
Cooling over 90 sec (2 degrees crossing angle):

In z-y plane:



Cooling over 90 sec (2 degrees crossing angle):

In z-y plane:



What defines a realistic laser beam?

What are the important parameters?

1) Energy
2) Angle of propagation, angular spread
3) Rayleigh length (size at the focal point)
4) Frequency spectrum
5) Duration, temporal shape
6) Frequency chirp
7) Polarization?
8) …?

9) How do you normally define the laser beam in simulations?
   Can we use photon macro-particles (like for ions)?
10) Any example of laser-propagation code?