SIRE

Software for **I**BS and **R**adiation **E**ffects

A Monte Carlo multi-particle simulation code
developed by A. Vivoli and M. Martini

Contents

- **Code implementation**

-Programming language(s)
-Programming style (object oriented, procedural, …)
-Prerequisites to run the code (OS, compilers, libraries, other codes)
-Parallelization technology (if any)

- **Description of the physics and the modeling of the code**

-What effects are included and what are not?
-What is the impact of the simulation tool for CERN studies?

- **Example of a typical application (use case)**

-How many simulations for one study
-Where the code is run (lxplus, lxbatch, other clusters)
-Computing time

- **Performance, available documentation and licensing policy**

-Is the performance in general adequate to the present needs?

- **Future plans and needs**

-Maintenance, extension and further development
-Include more physics to better model cases of interest for CERN?
-Performance improvement?
-Resource estimation for maintenance/development over next years
-What type of hardware resources would be best suited for the physics case?

Code Implementation

- Written in C
- Compile and run (locally or batch) the code:

  1. Compile the code:
  ```
  g++ lhcsire.c -o code
  ```

  2. To run the code at least 3 arguments are required, if less the code gives an error. Run the code:
  ```
  ./code arg1 arg2 arg3 arg4
  ```
  where arg1=twiss file, arg2=parameters file, arg3=the name of the output files. If arg4 is not defined it means that the distribution is generated in the code (default: Gaussian), if arg4 is defined then we give as an input the distribution.

- Other prerequisites: None
- Parallelization strategy: None

Description of the physics and the modeling of the code

Inputs

-The optics along a lattice (MADX twiss file).
-The parameters file.
-The particles distribution (default: Gaussian distribution).

The analytical models that describe the IBS effect assume Gaussian beam distributions. In the case of non-Gaussian beam distributions no theoretical models exist. **SIRE calculates IBS for any distribution.**

Description of the physics and the modeling of the code

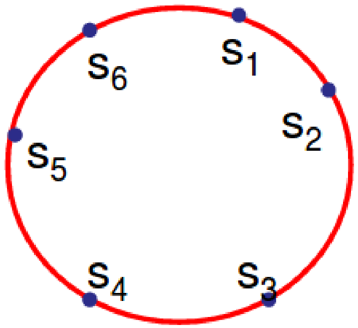
Inputs

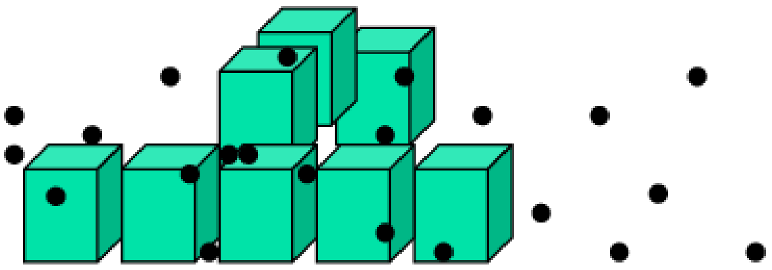
-The optics along a lattice (MADX twiss file).
-The parameters file.
-The particles distribution (default: Gaussian distribution).

The analytical models that describe the IBS effect assume Gaussian beam distributions. In the case of non-Gaussian beam distributions no theoretical models exist. **SIRE calculates IBS for any distribution.**

Computing IBS and Radiation Effects

-Particles are tracked from point to point in the lattice by their invariants.
-At each point of the lattice the scattering routine is called.
-6-dim coordinates of particles are calculated.
-Particles of the beam are grouped in cells.
-The intrabeam collisions between pairs of macro-particles are iteratively computed, the momentum of particles is changed because of scattering. Use of the classical Rutherford cross section.
-Invariants of particles are recalculated.
-Radiation damping and excitation effects are evaluated at the end of every loop.





Description of the physics and the modeling of the code

Inputs

-The optics along a lattice (MADX twiss file).
-The parameters file.
-The particles distribution (default: Gaussian distribution).

The analytical models that describe the IBS effect assume Gaussian beam distributions. In the case of non-Gaussian beam distributions no theoretical models exist. **SIRE calculates IBS for any distribution.**

Computing IBS and Radiation Effects

-Particles are tracked from point to point in the lattice by their invariants.
-At each point of the lattice the scattering routine is called.
-6-dim coordinates of particles are calculated.
-Particles of the beam are grouped in cells.
-The intrabeam collisions between pairs of macro-particles are iteratively computed, the momentum of particles is changed because of scattering. Use of the classical Rutherford cross section.
-Invariants of particles are recalculated.
-Radiation damping and excitation effects are evaluated at the end of every loop.

Outputs

-The beam distribution is updated and the rms beam emittances are recomputed, giving finally **the evolution of the emittance and particle distribution in time.**

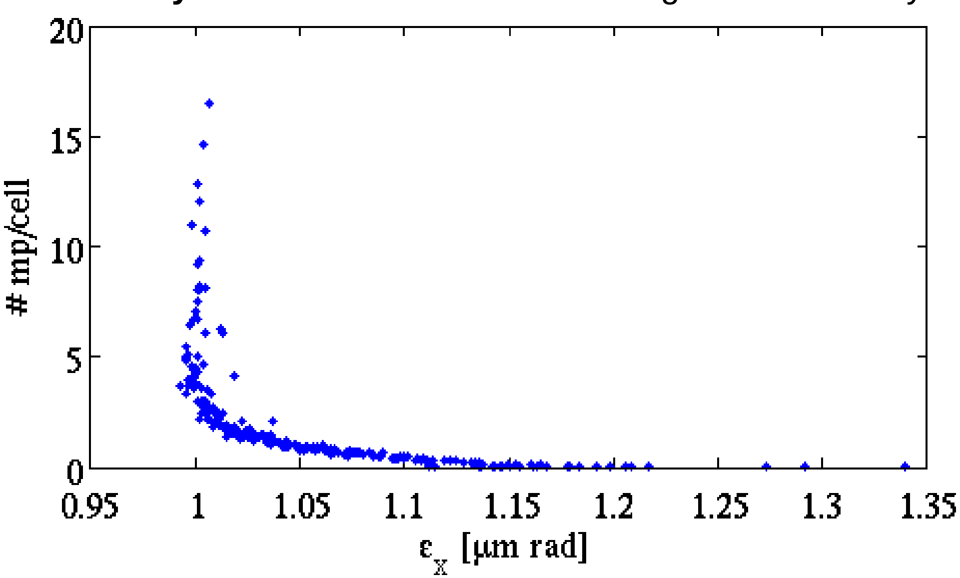
Parameters file

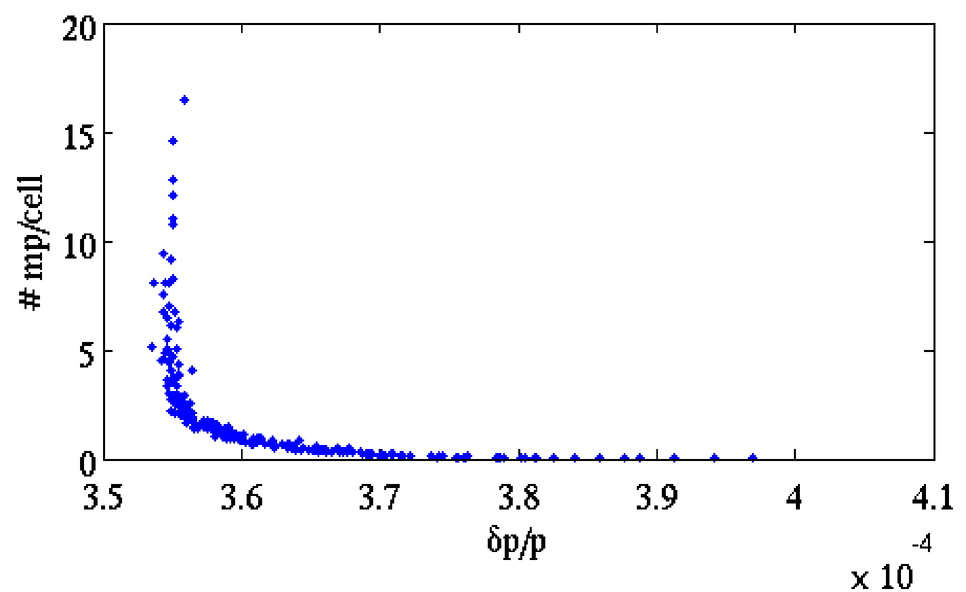
TEMPO: the full time of simulation [sec]
nturnstostudy: total number of turns
checktime: 0 if you want to see the TEMPO, 1 if it sees the turns instead of the TEMPO
oneturn: 1 for 1-turn calculation, 0 for multi-turn
NIBSruns: number of timesteps you devide the TEMPO to calculate IBS
fastrun: 1 if interpolation is used every TIMEINJ, NIBSruns determines how many turns are skipped for the IBS calculation. So be careful to have a sufficient NIBSruns for a specific time.
continuation: 1 so that if a job is killed it continues from where it stopped
epsx and epsz: initial horizontal and vertical emittance (geometrical)
delta and deltas: initial energy spread and bunch length (1sigma) in [m]
dtimex, dtimez and dtimes: horizontal,vertical and longitudinal damping times [sec]
eqx and eqz: equilibrium horizontal and vertical emittance
eqdelta and eqdeltas: equilibrium energy spread and bunch length
flag_rec: 0 if full lattice, 1 if recurrences; elements of the full lattice with twiss functions differing less than a specific prec. are considered equal.
damping: 0 if the radiation damping is not taken into account, 1 if SR
IBSflag: 0 if no IBS calculations, 1 if IBS
q_ex: 0 if the quantum excitation not taken into account , 1 if QE
momentum: beam momentum (energy) [MeV]
numbunch: bunch population
numpart: number of macroparticles
ncellx, ncellz and ncells: # cells in the horizontal, vertical and longitudinal plane
ncollisions: # collisions per particle (close encounters)
convsteadystate: 1 if we track until convergeance to steady-state not in full TEMPO time

Parameters file

TEMPO: the full time of simulation [sec]
nturnstostudy: total number of turns
checktime: 0 if you want to see the TEMPO, 1 if it sees the turns instead of the TEMPO
oneturn: 1 for 1-turn calculation, 0 for multi-turn
NIBSruns: number of timesteps you devide the TEMPO to calculate IBS
fastrun: 1 if interpolation is used every TIMEINJ, NIBSruns determines how many turns are skipped for the IBS calculation. So be careful to have a sufficient NIBSruns for a specific time.
continuation: 1 so that if a job is killed it continues from where it stopped
epsx and epsz: initial horizontal and vertical emittance (geometrical)
delta and deltas: initial energy spread and bunch length (1sigma) in [m]
dtimex, dtimez and dtimes: horizontal,vertical and longitudinal damping times [sec]
eqx and eqz: equilibrium horizontal and vertical emittance
eqdelta and eqdeltas: equilibrium energy spread and bunch length
flag_rec: 0 if full lattice, 1 if recurrences; elements of the full lattice with twiss functions differing less than a specific prec. are considered equal.
damping: 0 if the radiation damping is not taken into account, 1 if SR
IBSflag: 0 if no IBS calculations, 1 if IBS
q_ex: 0 if the quantum excitation not taken into account , 1 if QE
momentum: beam momentum (energy) [MeV]
numbunch: bunch population
numpart: number of macroparticles
ncellx, ncellz and ncells: # cells in the horizontal, vertical and longitudinal plane
ncollisions: # collisions per particle (close encounters)
convsteadystate: 1 if we track until convergeance to steady-state not in full TEMPO time

In order to avoid combinations which give very small #macroparticles/cell, it was observed that 5macroparticles/cell is the optimal minimum number.





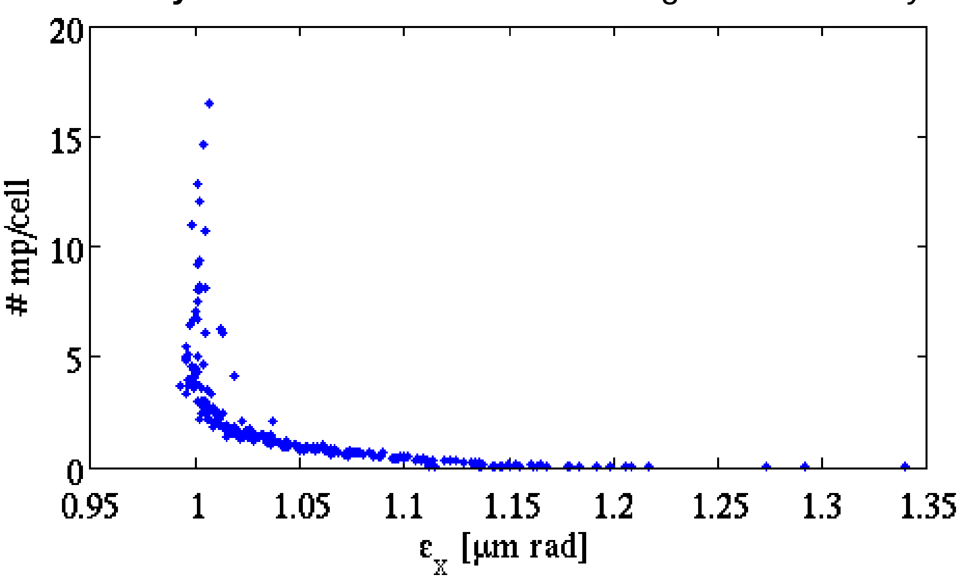
Parameters file

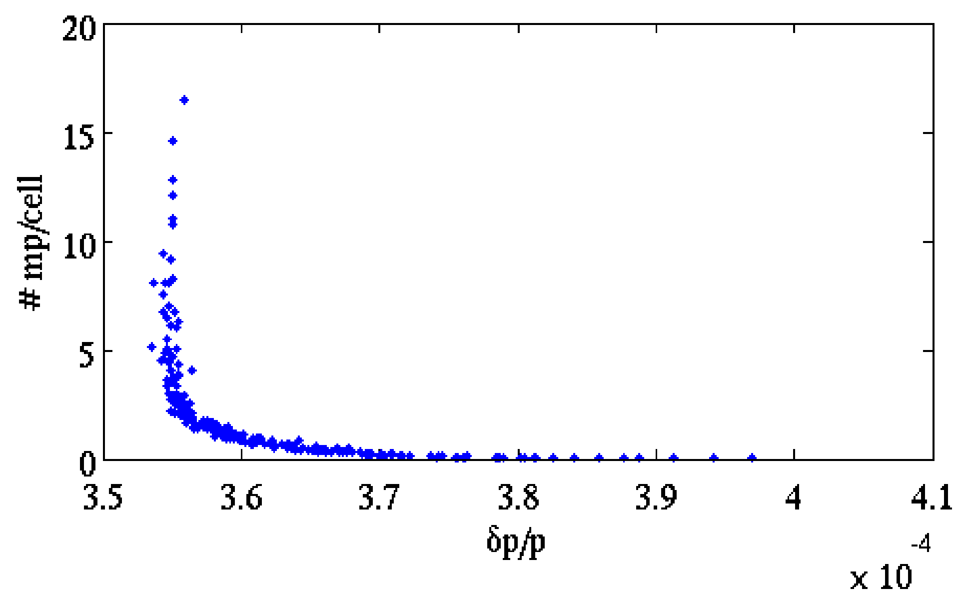
TEMPO: the full time of simulation [sec]
nturnstostudy: total number of turns
checktime: 0 if you want to see the TEMPO, 1 if it sees the turns instead of the TEMPO
oneturn: 1 for 1-turn calculation, 0 for multi-turn
NIBSruns: number of timesteps you devide the TEMPO to calculate IBS
fastrun: 1 if interpolation is used every TIMEINJ, NIBSruns determines how many turns are skipped for the IBS calculation. So be careful to have a sufficient NIBSruns for a specific time.
continuation: 1 so that if a job is killed it continues from where it stopped
epsx and epsz: initial horizontal and vertical emittance (geometrical)
delta and deltas: initial energy spread and bunch length (1sigma) in [m]
dtimex, dtimez and dtimes: horizontal,vertical and longitudinal damping times [sec]
eqx and eqz: equilibrium horizontal and vertical emittance
eqdelta and eqdeltas: equilibrium energy spread and bunch length
flag_rec: 0 if full lattice, 1 if recurrences; elements of the full lattice with twiss functions differing less than a specific prec. are considered equal.
damping: 0 if the radiation damping is not taken into account, 1 if SR
IBSflag: 0 if no IBS calculations, 1 if IBS
q_ex: 0 if the quantum excitation not taken into account , 1 if QE
momentum: beam momentum (energy) [MeV]
numbunch: bunch population
numpart: number of macroparticles
ncellx, ncellz and ncells: # cells in the horizontal, vertical and longitudinal plane
ncollisions: # collisions per particle (close encounters)
convsteadystate: 1 if we track until convergeance to steady-state not in full TEMPO time

scanning

In order to avoid combinations which give very small #macroparticles/cell, it was observed that 5macroparticles/cell is the optimal minimum number.

tested





The particles distribution

Input distribution

Default (if arg4 is not defined): Generates Gaussian distribution of macroparticles using a random number generator to give the action angle variables of all macroparticles in all planes.

or

Give the initial distribution as input (arg4 is defined): Create* the distribution you want in each plane and then give as an input file the action angle variables of all macroparticles for the chosen distributions.

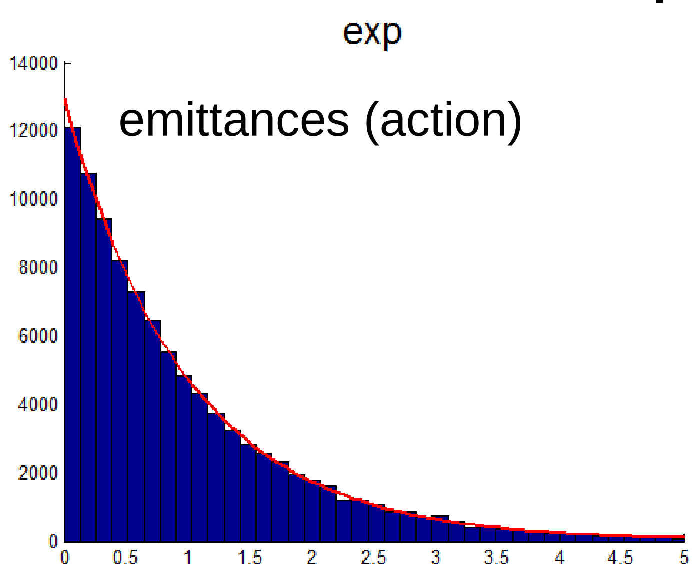
Output distribution

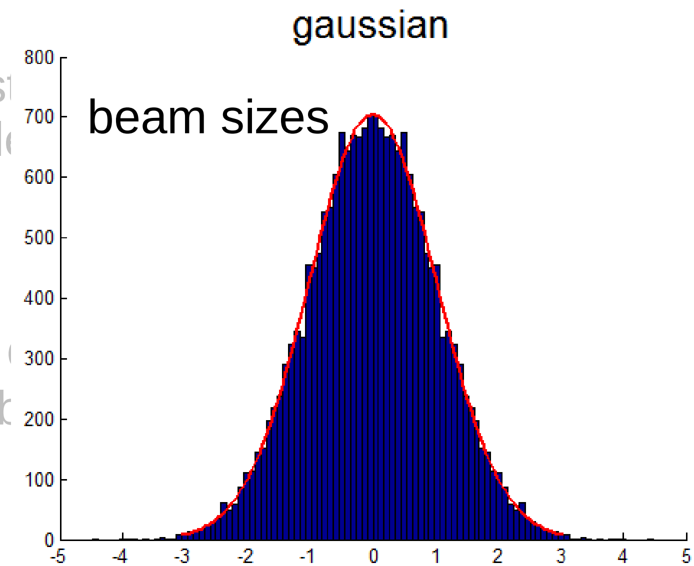
-Go from the emittances of the distribution file to the x, x' values.
-A histogram of x, x' shows the profile of the beam distribution.

*Example : use of the Abel inverse transform to go from beam sizes to emittances (action).
Acknowledgement: T. Argyropoulos

The particles distribution

Input distribution





Default (if Gaussian dist a
random nu angle variabl nes.

or

Give the i **is defined):** in each
plane and n angle variat chosen
distributior

Output distribution

-Go from the emittances of the distribution file to the x, x' values.
-A histogram of x, x' shows the profile of the beam distribution.

*Example : use of the Abel inverse transform to go from beam sizes to emittances (action).

Acknowledgement: T. Argyropoulos

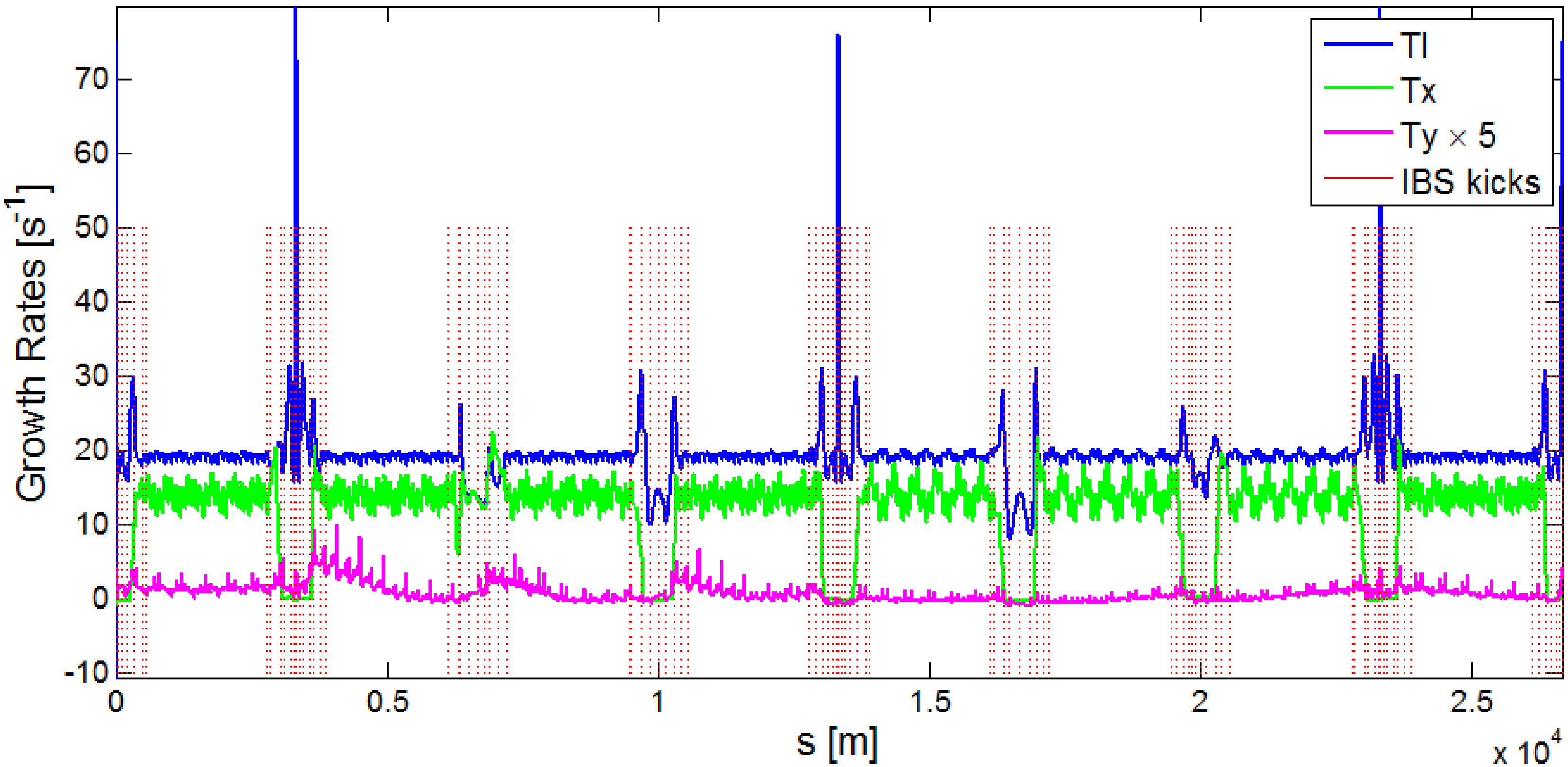
Examples of a typical application

Optics along the lattice: Reduced lattice

As the LHC has a large number of elements (more than 11000), the computational time that SIRE needs to track the distribution for all of them along the ring is very long.

- The IBS growth rates were calculated for the LHC's full optics, using the IBS module of the MADX, based on the B-M formalism.
- The optimal minimum number of IBS kick points around the lattice, without affecting the overall effect, are identified.

Finally, the reduced lattice used in SIRE has only 92 points (much lower computational time).



Examples of a typical application

A. SIRE benchmarking with the B-M formalism; at FB for 1h, for Gaussian distributions in all planes.

Parameters @ FB	Nominal (BCMS)	HiLumi
E [GeV]	450	450
$\varepsilon_{x,y}$ [μm]	1.5	2.0
4σ bunch length [ns]	1.0	1.2
Bunch population [10^{11}]	1.2	2.3
# of macroparticles	175000	175000

-It is assumed that the bunch profiles in all planes are Gaussian.

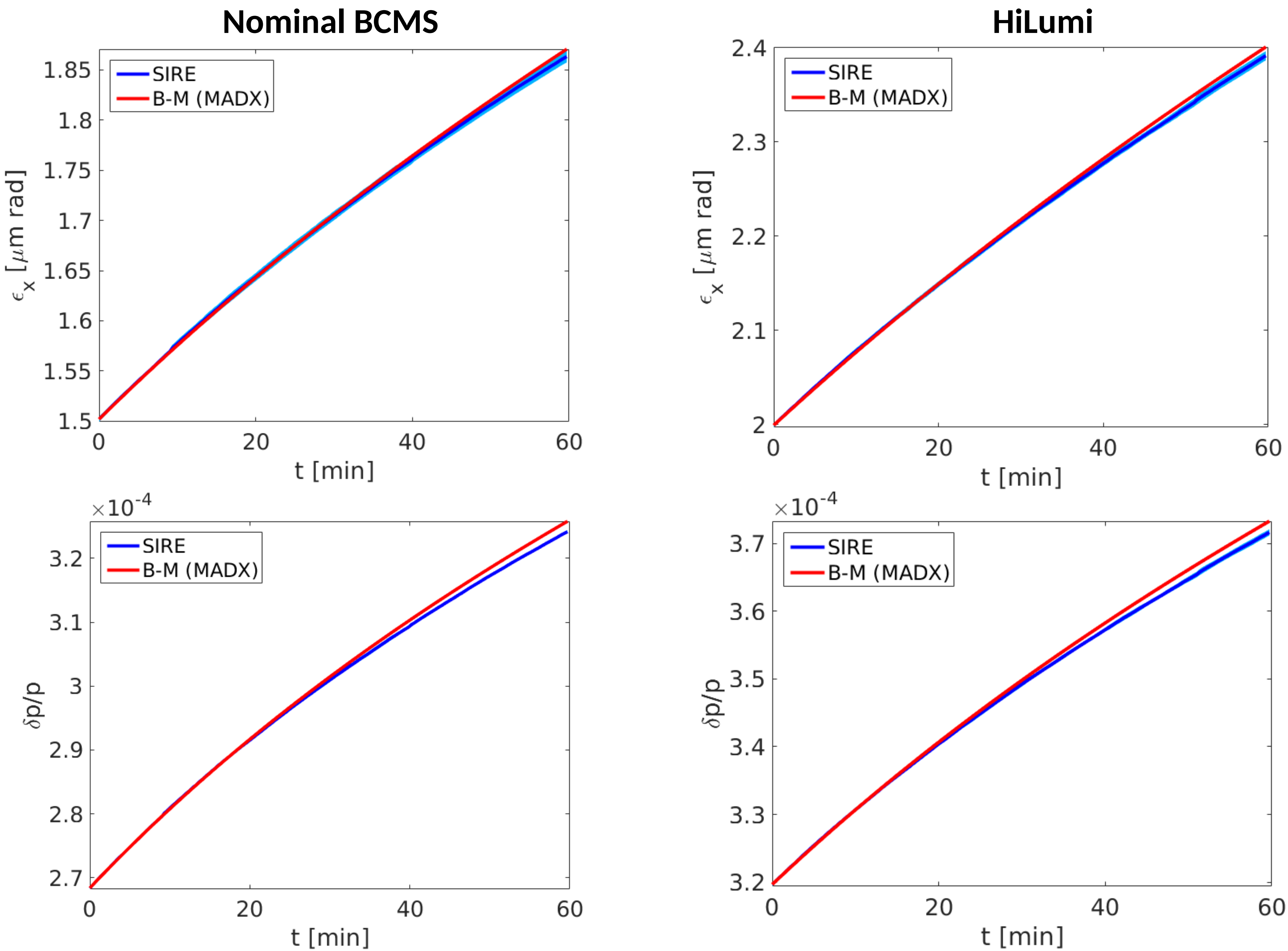
Simulation time: ~1h

B. SIRE comparison with experimental data; at FT for ~11.5h.

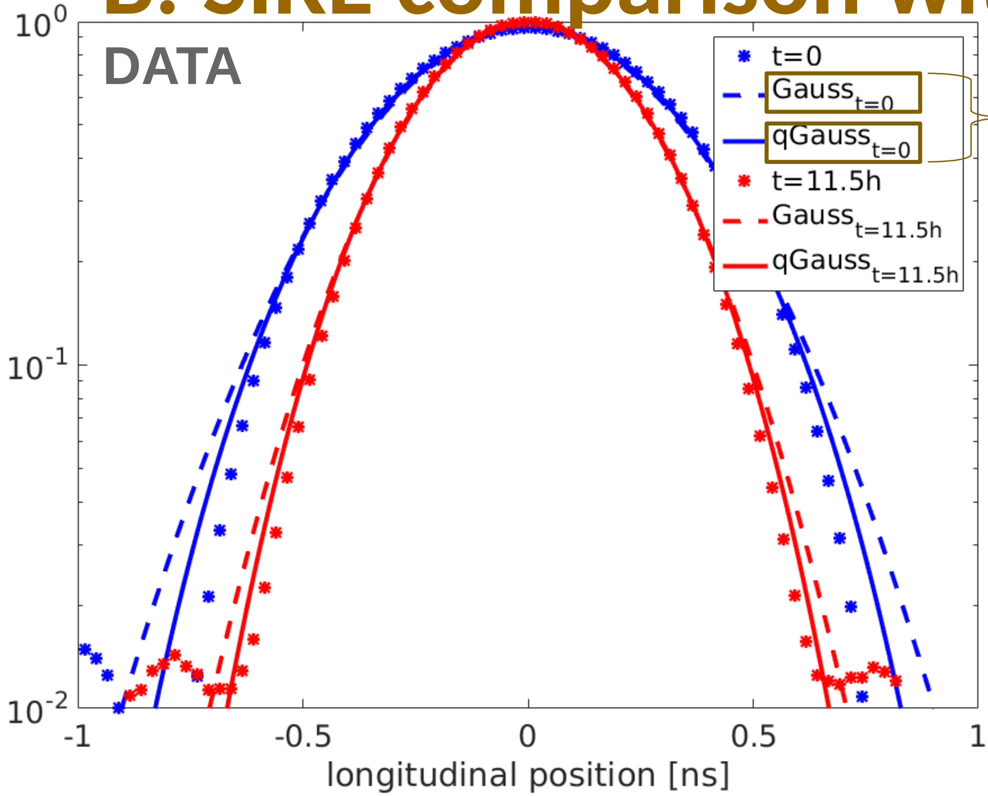
-The bunch population degradation and the extra (on top of IBS) transverse emittance blow up observed during stable beams in the LHC are also taken into account.
-For the simulations, it is assumed that the transverse bunch profiles are Gaussian and the longitudinal bunch profile is non-Gaussian (as observed from the LHC data).
-The distributions are saved every 1h such that to follow their evolution.

Simulation time: ~11h

A. SIRE benchmarking with the B-M formalism at FB

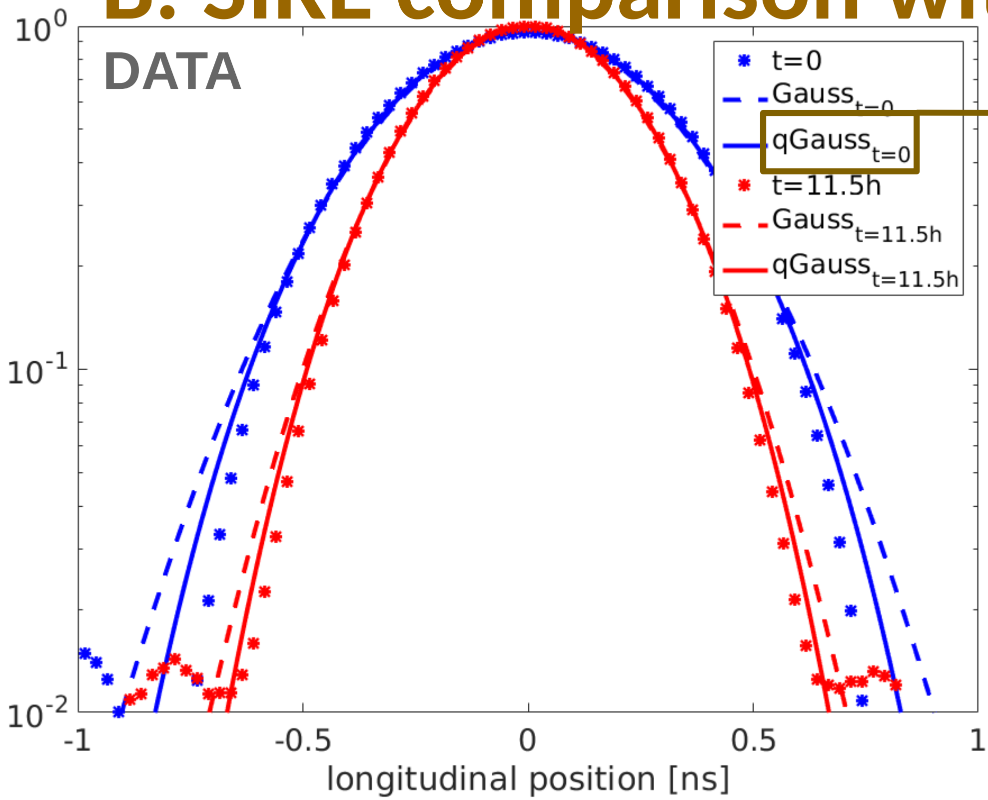


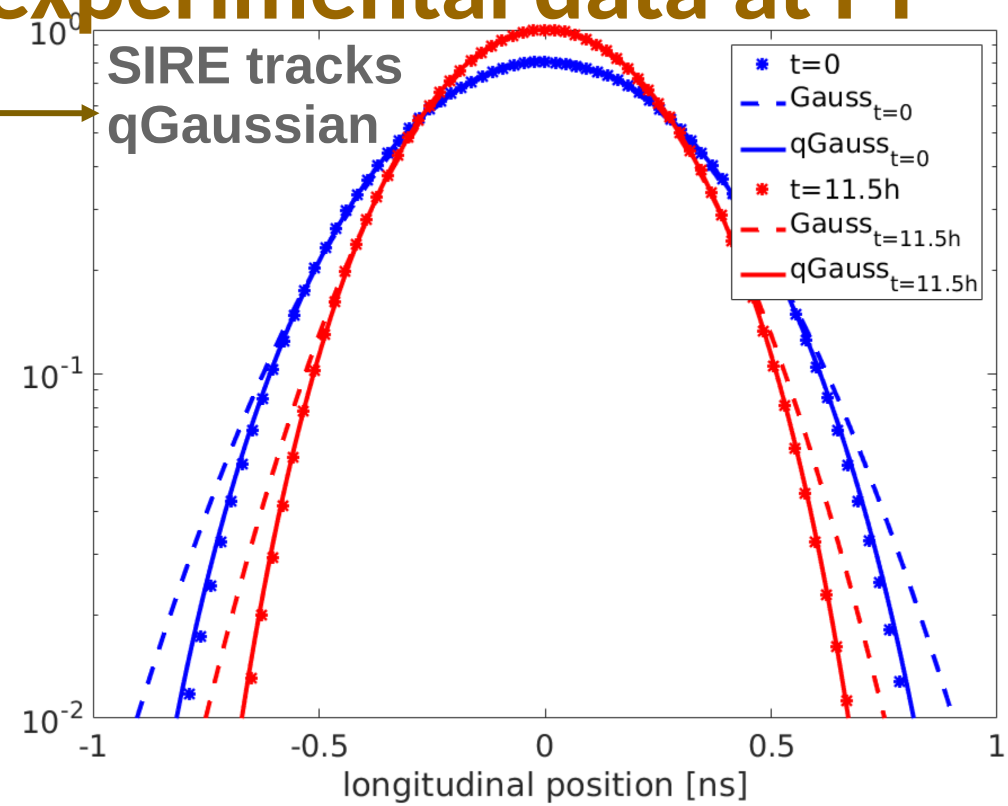
B. SIRE comparison with experimental data at FT



Track in SIRE the distributions coming from the initial data fits.

B. SIRE comparison with experimental data at FT

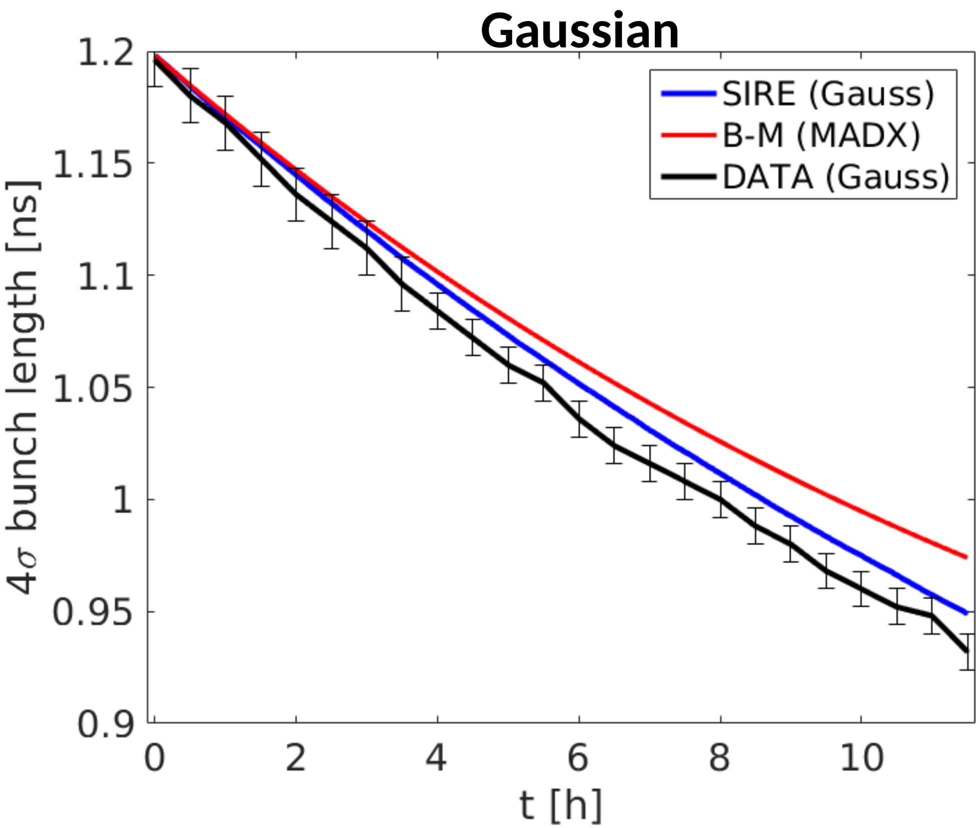


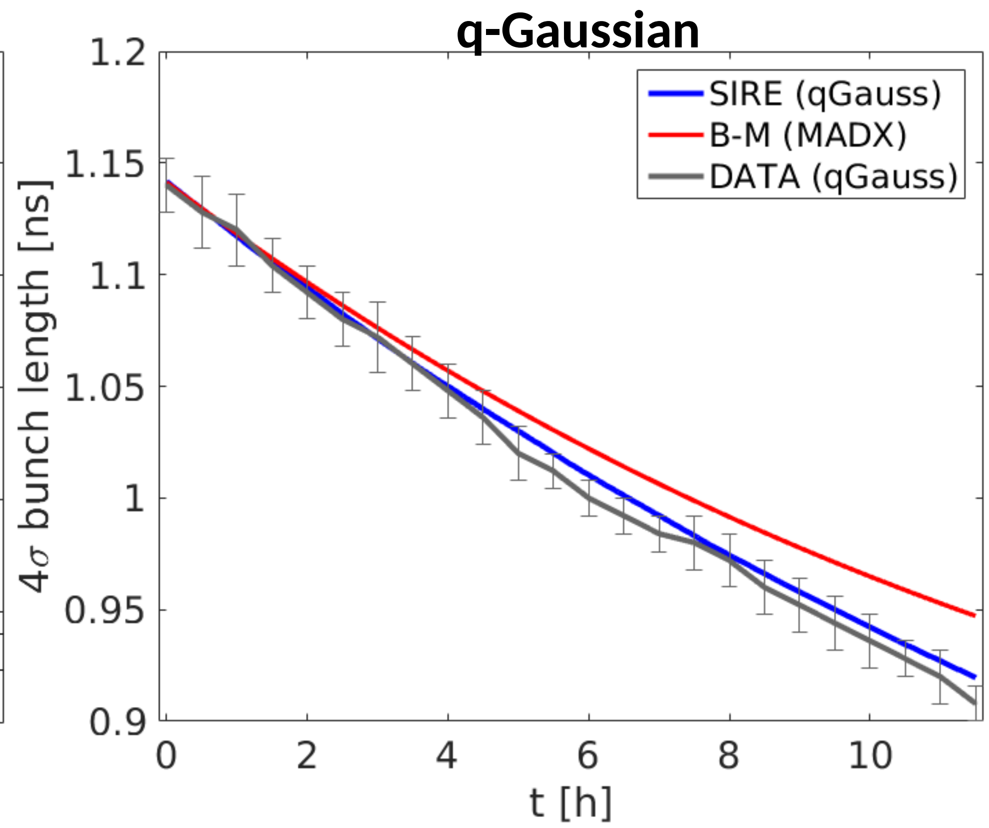


	q initial	sigma initial	q final	sigma final
DATA	0.88±0.03 (rmse=0.010)	0.286±0.004 (rmse=0.010)	0.93±0.03 (rmse=0.010)	0.227±0.002 (rmse=0.010)
SIRE qGaussian	0.85±0.01 (rmse=0.003)	0.290±0.001 (rmse=0.003)	0.86±0.01 (rmse=0.004)	0.235±0.001 (rmse=0.004)

For the q-Gaussian function, that is used to fit non-Gaussian distributions observed in the LHC, the rms depends on the q parameter that shows how heavy the tails are. For a “light” tailed distribution ($q<1$) the Gaussian overestimates the rms value, the opposite is true for a “heavy” tailed distribution ($q>1$).

B. SIRE comparison with experimental data at FT





-In SIRE the beam distribution is updated and the emittances are recomputed, while MADX, which is based on the analytical formulation of B-M, assumes always Gaussian distributions.
-Since the agreement between the experimental data and the SIRE code is very good, it can be used for distribution and emittance evolution predictions. Using a proper (determined by the rmse value) fitting function for the observed bunch profiles and then, based on these fits, generate distributions to be tracked in SIRE is the optimal way to accurately follow the real evolution of distributions and emittances.

Performance and available documentation

- The performance is in general adequate to the present needs
- Available documentation:

  https://twiki.cern.ch/twiki/bin/view/ABPComputing/SIRE

- Interesting links for SIRE:

  http://cds.cern.ch/record/1240834/files/sLHC-PROJECT-REPORT-0032.pdf?version=1

  https://agenda.linearcollider.org/event/4507/contributions/17682/attachments/14276/23411/CLIC_2010_IBS.pdf

  http://inspirehep.net/record/1507570/files/ICFA69_38-59.pdf

  http://accelconf.web.cern.ch/AccelConf/ipac2017/papers/tupva044.pdf

Future plans and needs

- Maintenance, extension and further development

  The code is presently maintained by F. Antoniou and S. Papadopoulou. One could try to add other sources of emittance growth and intensity lifetime. We may think to GPU or parallelize the code, but no resources available.

- Include more physics to better model cases of interest for CERN?

- Performance improvement?

- Resource estimation for maintenance/development over next years

  5% of a staff, 10% of a fellow for maintenance. For development, we need probably another 10% of a fellow, and a (software oriented) TS. Maybe combined with another software development activity in HSI.

- What type of hardware resources would be best suited for the physics case?

  Right now the code runs on lxplus locally. If parallelized or GPUd, we will have to think of the required HW.

Thank you!

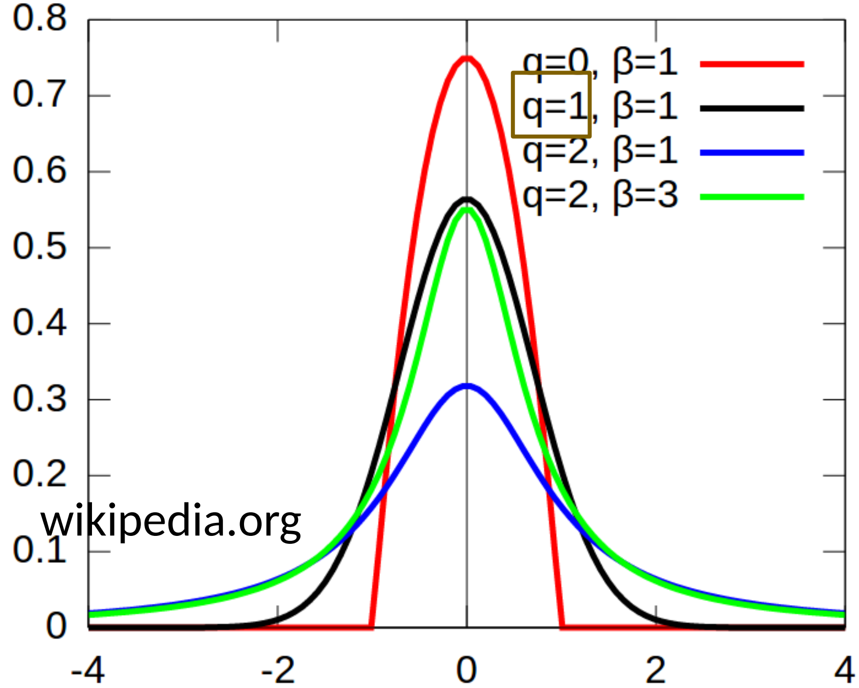
Back up

The q Gaussian function

In many cases, the bunch profiles in the LHC, appear to have **tails that differ from the ones of a normal distribution**. In order to describe them more accurately, the **q-Gaussian function**, is used. This distribution has a probability density function given by:

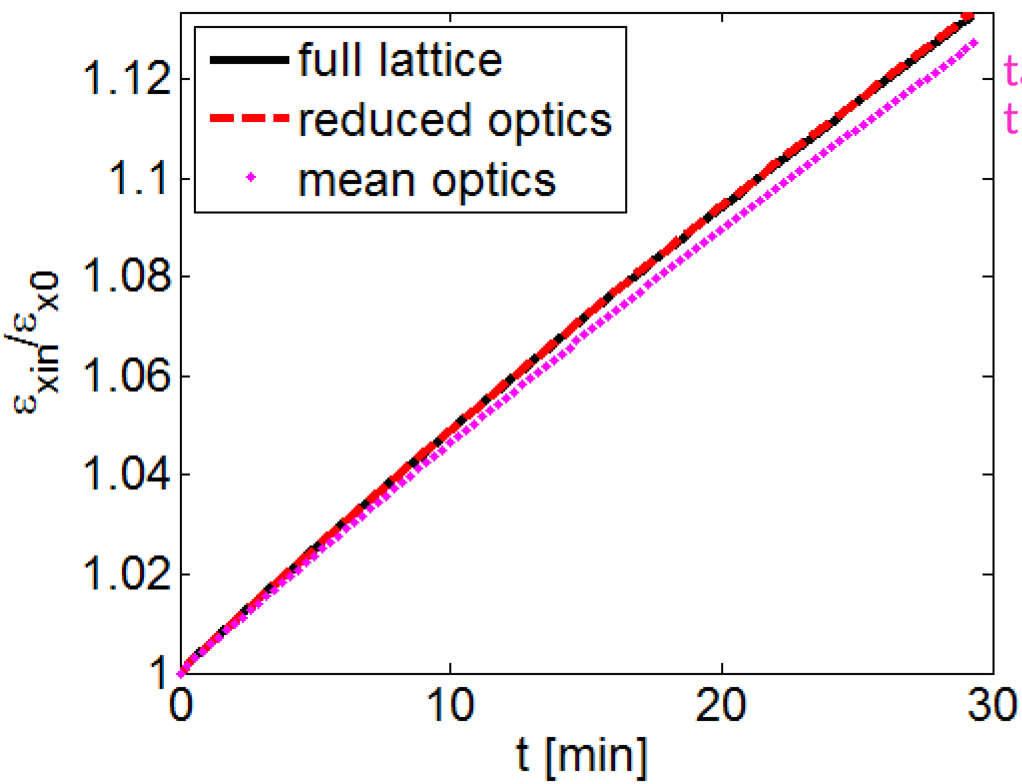
$$f(x) = \frac{\sqrt{\beta}}{C_q} e_q(-\beta x^2), \; e_q(x) = [1 + (1-q)x]^{\frac{1}{1-q}}$$

q<1→"light" tailed
q=1→Gaussian
q>1→ "heavy" tailed



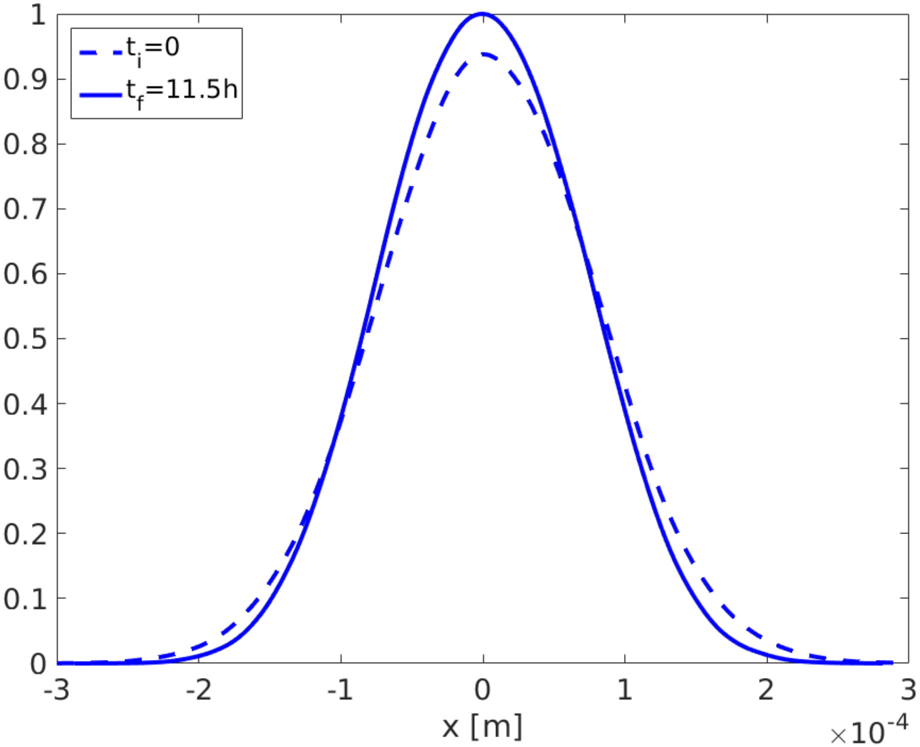
$$\sigma = \sqrt{\frac{1}{\beta(5-3q)}}$$

for q<5/3

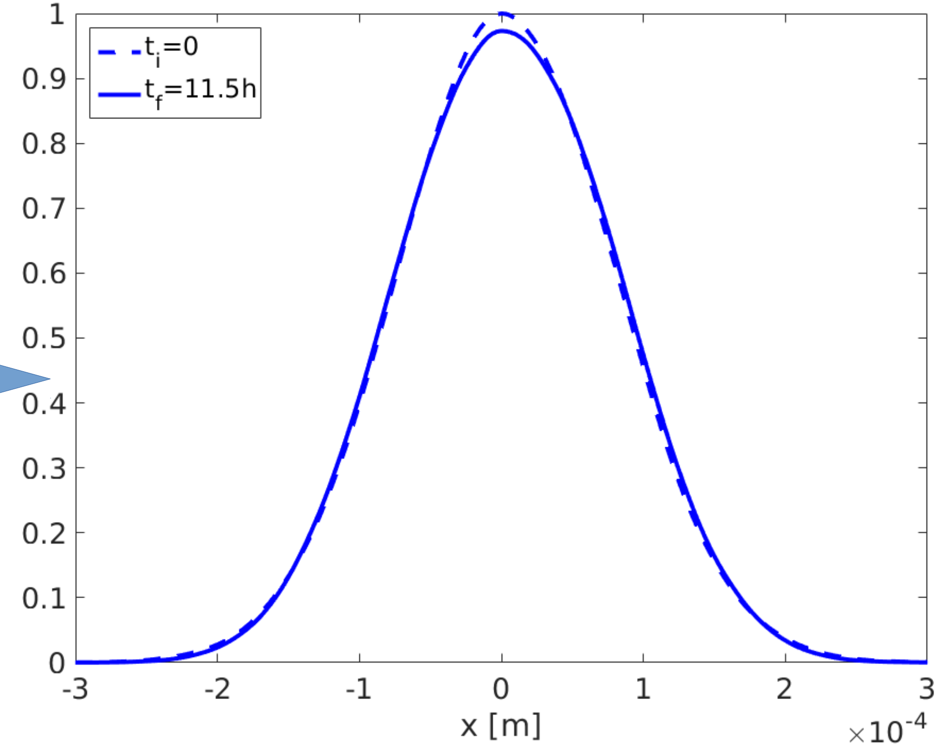


take <optics> and then a single element of the lattice that has similar optics, use its gr. rates.

Include extra trans. emit. blow up in SIRE



+extra
blow up



	horizontal profile	q initial	q final	
rmse=0.004	no extra blow up	1.±0.01	0.87±0.01	sigma=0.068
rmse=0.003	with extra blow up	1.±0.01	0.89±0.01	sigma=0.0756