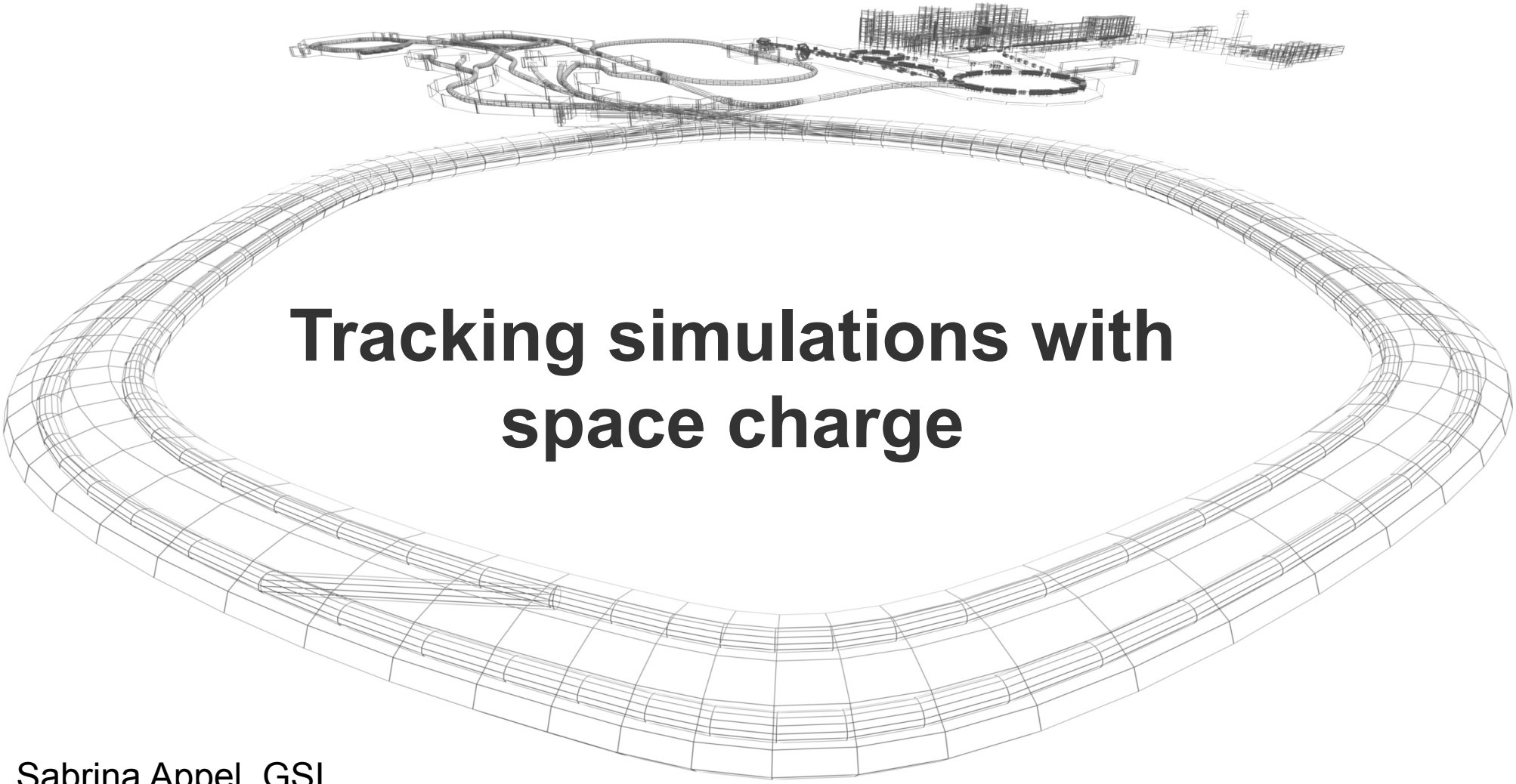
GSI Helmholtzzentrum für Schwerionenforschung GmbH



Tracking simulations with space charge

Sabrina Appel, GSI

Outline

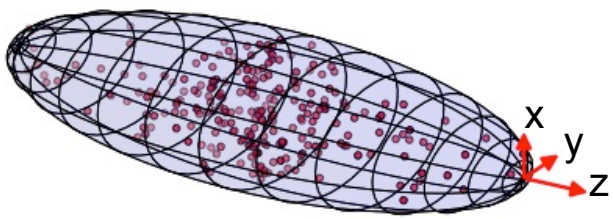
- Introduction
- Particle-In-Cell scheme
- Space charge solvers
  - Longitudinal space charge solver
  - Transversal space charge solver
  - 3D approaches
- Modern implementations
- Summary

Space charge (SC)

Concept:

- Space charge is the inter-particle Coulomb force.
- In the beam frame SC force be evaluated with the Poisson's equation.

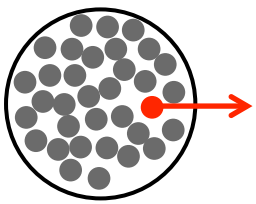
$$\nabla^2 \Phi = \frac{\rho}{\epsilon_0 \gamma^2}$$



Space charge effects:

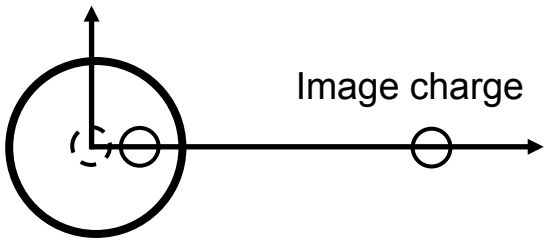
- SC limited or/and determine beam parameters and accelerator components (CERN LHC injector chain + FAIR)
  - Direct SC

Forces act directly from beam to particle.



  - Indirect SC

Beam interacted with its surrounding.



Modeling:

- One attempts to find the simplest model & fastest algorithms that contains the necessary physics.
  - Breakdown the problem to less dimensions (1D, 2D)

K. Ng: Physics of Intensity Dependent Beam Instabilities; H. Wiedemann: Particle Accelerator Physics; M. Reiser: Theory and Design of charged Particle Beams

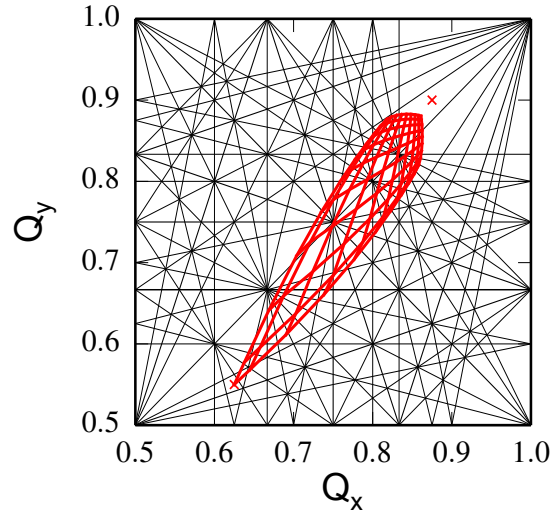
Frozen space charge model

Analytical model

- The solution of the Poisson equation for a 2D Gaussian and the space charge tune spread as a function of the particle amplitude can be calculated analytical.

$$\nabla^2\Phi = \frac{\rho}{\epsilon_0\gamma^2} \qquad \rho \sim e^{-\left(\frac{x^2}{2\sigma_x^2}+\frac{x^2}{2\sigma_x^2}\right)} \qquad \Delta Q_y^{sc} = -\frac{NZ^2 g_f}{2\pi A\beta_0^2\gamma_0^3 B_f \epsilon}$$

(Max. tune shift)



Tracking

- The kick acting on the particle is computed from the analytical electric field.
- During tracking simulations the electric filed is adapted on changed beam intensity and size.

Disadvantage

- This model is not self-consistent.
- Self-consistent means that the motion of the particles distribution changes the fields and the forces due to these fields change the particle distribution.

Codes

MadX, MICROMAP, …

A. Burov, et. al., Transverse instabilities of coasting beams with space charge, Phys. Rev. ST-AB (2009)
M. Bassetti, et. al., Closed expression for the electrical field of a two-dimensional Gaussian charge, CERN-ISR-TH/80-06

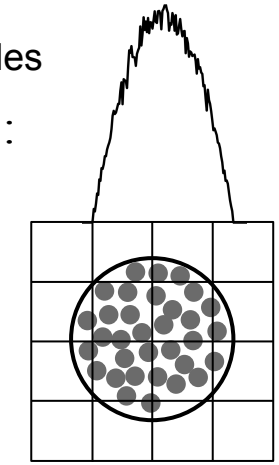
PIC (Particle In Cell) algorithms

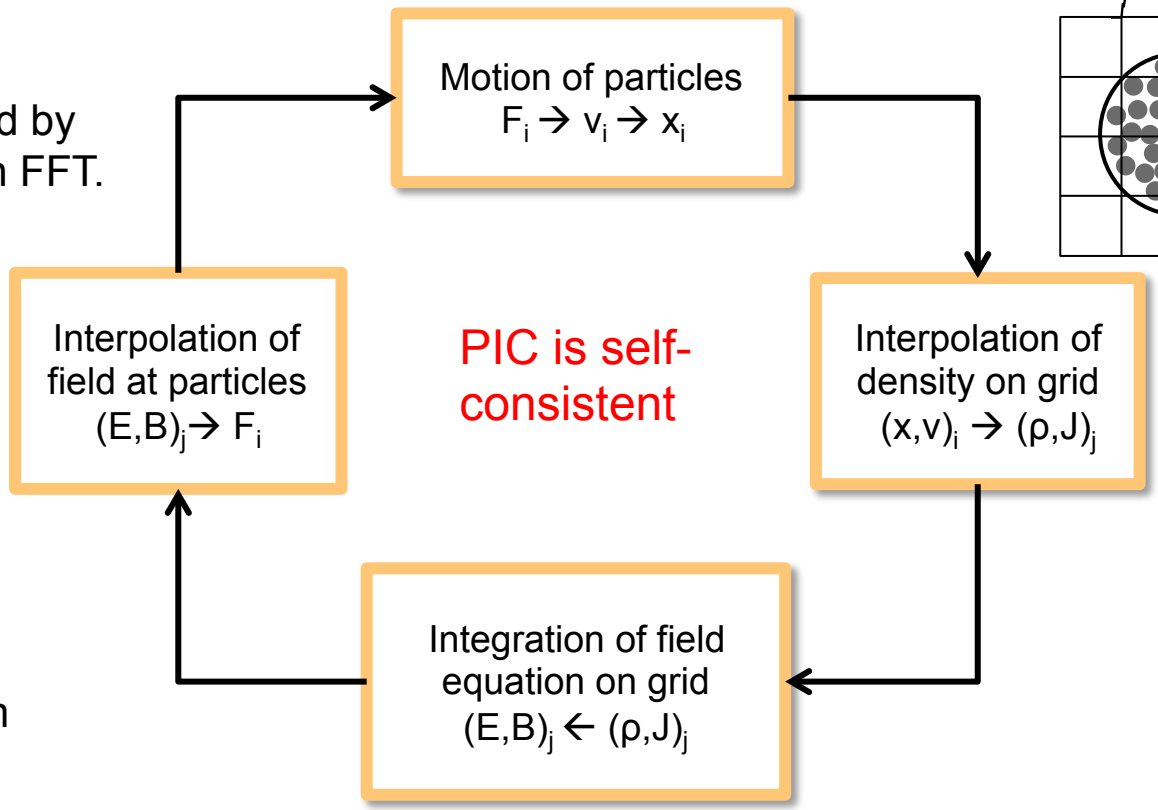
Particle In Cell

- Applied also in Astrophysics & Plasma physics.
- Space charge forces are obtained by solving the Poisson equation with FFT.
- Between the evaluation of SC forces, also other external forces can act on the beam.
- Between the "SC kicks" the beam oscillations have to be resolved

Macro-particles

$M \ll N$:



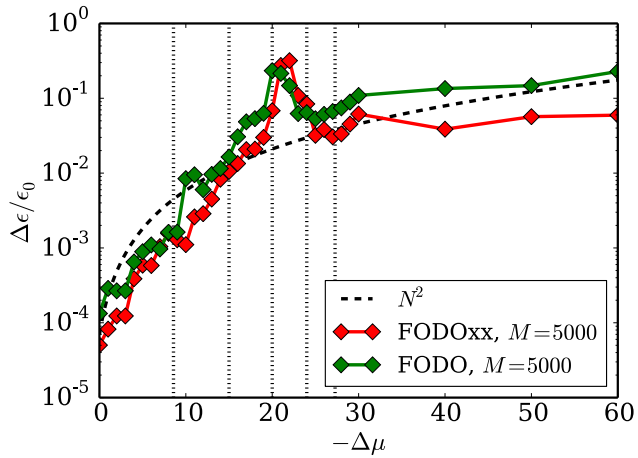


C. Bridsall & A B Langdon: Plasma Physics via computer simulation; R W Hockney & J W Eastwood: Computer simulation using particles

PIC (Particle In Cell) algorithms

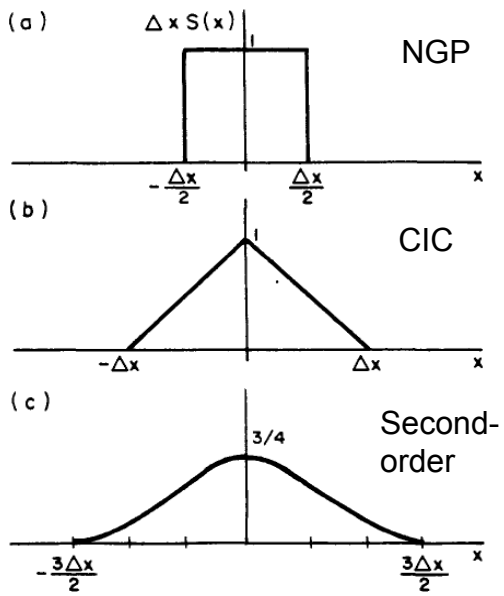
Noise

- Artificial collisions between macro-particles generate noise.
- The increase of emittance & entropy can be described analytically.
- The identification of the optimum number of macro-particles and the grid spacing is important.



Interpolation

- Nearest-Grid-Point (NGP) or Cloud-In-Cell (CIC) are widely.
- Higher-order interpolation reduces noise with the cost of more computation time.



Diagnostics

- PIC provides information of the particles in phase space & fields and should be frequently used.
- The user can
  - compare the electric field & potential against analytical expressions.
  - also verify, if the initial beam distribution is space charge matched.
  - study the artificial Schottky noise.

Struckmeier, Part. Acccel. 45 229 (1994); Boine-Frankenheim et al., Nucl. Instr. Meth A 770 (2015); Hofmann et al., IEEE Trans. Nucl. Sci. 26, 3526 (1979); Venturin et al, PRL 81, 96 (1998)

Longitudinal space charge field (1D)

Concept

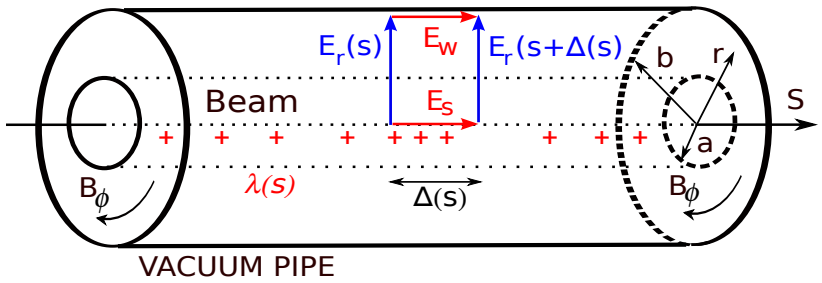
- The coupling impedance is introduced to relate the current modulations to the induced voltage along the beam path.

$$V = -Z(\omega)I(\omega)$$

1D Model

- Assuming a coasting beam with current modulations in a round beam pipe.

$$I = I_0 + I_n \exp(-inz/R)$$



- From Faraday's law follow the longitudinal electric to

$$\oint \vec{E}\cdot d\vec{l} = -\frac{\partial}{\partial t}\int \vec{B}\cdot d\vec{A}$$

$$E_s = -\frac{eg_0}{4\pi\gamma^2\varepsilon_0}\frac{\partial\lambda}{\partial s}$$

(depends on transversal geometry)

$$g_0 = 1 + 2\ln b/a$$

- Than space charge can be treated as impedance. $V_n = Z_n I_n$

$$\frac{Z_n^{sc}}{n} = -i\frac{g_0 Z_0}{2\beta_0\gamma^2}$$

Longitudinal space charge field (1D)

Implementation in a code

1. Interpolation to grid

$$I(X_i,t) = \beta_0 c \sum_{j=0}^{N} S(X_i - x_j)$$

2. FFT solver

$$\hat{I}_n(t^*) = FFT\left[I(Z_i,t^*)\right]$$

$$E(X_i,t) = FFT^{-1}\left[-Z_n \hat{I}_n(t^*)\right]$$

3. Interpolation to particles

$$E(x_j,t) = \Delta x \sum_i E(X_i,t) S(X_i - x_j)$$

(Also possible)

$$\left(V_i = FFT^{-1}\left[-Z_n \hat{I}_n(\omega_i)\right]\right)$$

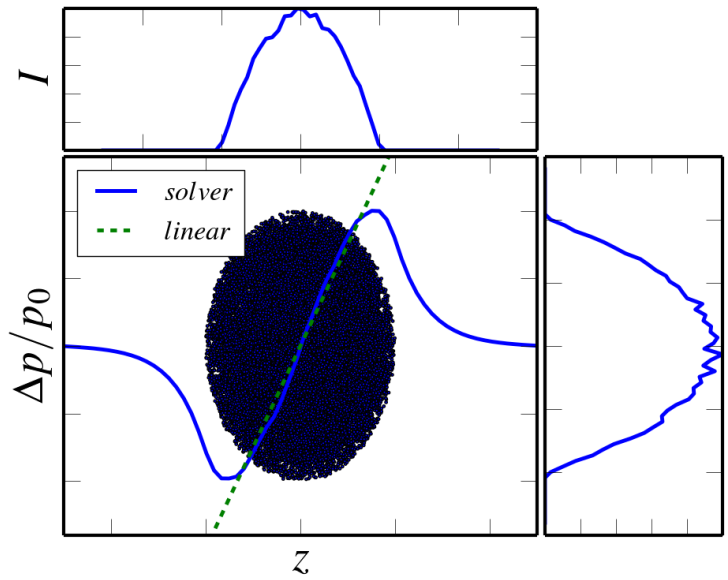
Outcome

- The long. electric field for a parabolic beam is linear (analytic relation)

$$\rho = \rho_0 \left(1 - \frac{z^2}{z_m^2}\right) \quad \Rightarrow \quad E_z \sim \rho_0 \frac{z}{z_m^3}$$

- Also other impedance sources can be included

$$Z_{\|} = Z_{\|}^{sc} + Z_{\|}^{bb} + \ldots$$



Codes

BLonD, Lobo, pyORBIT, …

2/3D space charge field

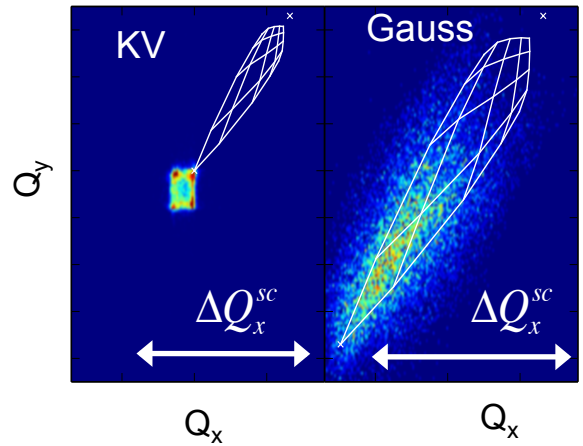
Concept

- The Poisson equation can be solved with the Green's function.
- The solution generated by a general source function $\rho(r)$ is simply the appropriately weighted sum of all of the Green's function solutions:

$$\nabla^2\Phi = \frac{\rho}{\epsilon_0\gamma^2} \quad \Rightarrow \quad \Phi(\vec{r}) = \frac{1}{4\pi\epsilon_0}\int_\Omega G(\vec{r},\vec{r}\,')\rho(\vec{r}\,')d^3\vec{r} \quad \Rightarrow \quad \hat{\phi} = \hat{G}\hat{\rho}$$

- Since $G(r)$ and $\rho(r)$ are periodic functions, the potential $\phi(r)$ can be computed efficiently using FFT (convolution theorem).
- The Fourier approach can be considered as direct Poisson solver
- Due to the clever selection of the Green's function, an accurate and efficient space charge calculation is possible.
- Depending on the problem boundary conditions for the potential and particles must be included.

SC tune shift from 2D Poisson solver



Transverse space charge field (2D)

2D Model

- The model is widely used for ring calculation to compute losses, emittance growth to SC
- Due to the 3D tracking also longitudinal effects (i.e. bunch factor) can be included.
- The 2D Green's function in free space is $G(x,y) = \frac{1}{2\pi} \ln \sqrt{(x^2 + y^2)}$

Implementation in a code

FFT potential solver

$$\hat{G}(k,l) = FFT\left[G(x,y)\right]$$
$$\hat{\rho}(k,l) = FFT\left[\rho(x,y)\right]$$
$$\phi(X,Y) = FFT^{-1}\left[\hat{\rho}\hat{G}\right]$$

Electric field

$$E_{x,y} = -\frac{\phi_{i+1}(x,y) - \phi_{i-1}(x,y)}{2dn_{x,y}}$$

+ interpolation of particles & fields

Codes

pyORBIT, PATRIC, Synergia, Simpsons,…

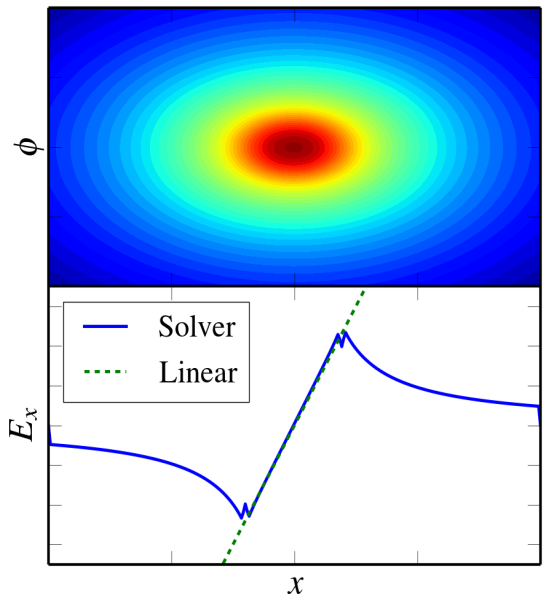
Outcome

- The transverse electric field for a KV beam is linear (analytic relation)

$$\rho(r) \sim \delta(r-a)$$

$$E_x \sim x$$

- Grid spacing has an influence on the determined el. field.

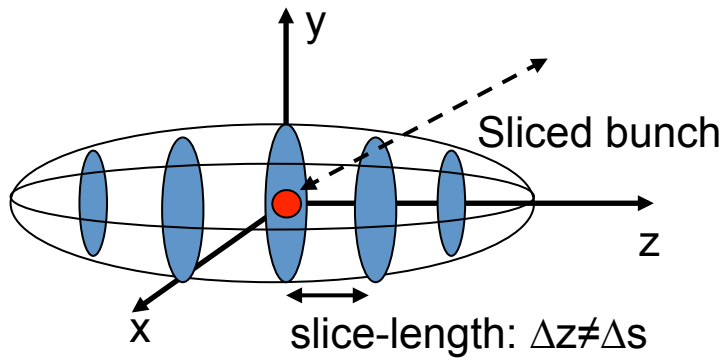


3D approaches

2.5D Model

- 2.5D SC is used if transverse properties vary fast or transverse impedances are of interest.
- The beam is slices n times along the longitudinal direction and in the slices SC is solved with the 2D model

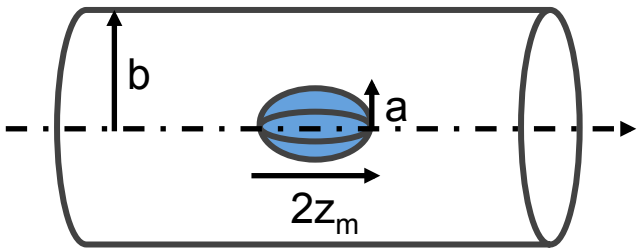
$$\frac{\partial E_x}{\partial x} + \frac{\partial E_y}{\partial y} = \frac{\rho(x,y,\{z,s_m\})}{\varepsilon_0}$$



- 3D Grid interpolation

3D model:

- 3D SC solver are used if the long. & trans. dimensions are comparable ($a \approx z_m$ or $b > z_m$)
- Widely used in linac & source studies but also important for the bunch compression in rings
- The convolution theorem in 3D is $\hat{\phi}_{l,m,n} = \hat{G}_{l,m,n}\hat{\rho}_{l,m,n}$
- The 3D Green's function in free space is $G(r) = \frac{-1}{2\pi}\frac{1}{\sqrt{x^2+y^2+z^2}}$



Codes

pyORBIT, PATRIC, Synergia, TRACEWIN, PARMILA, …

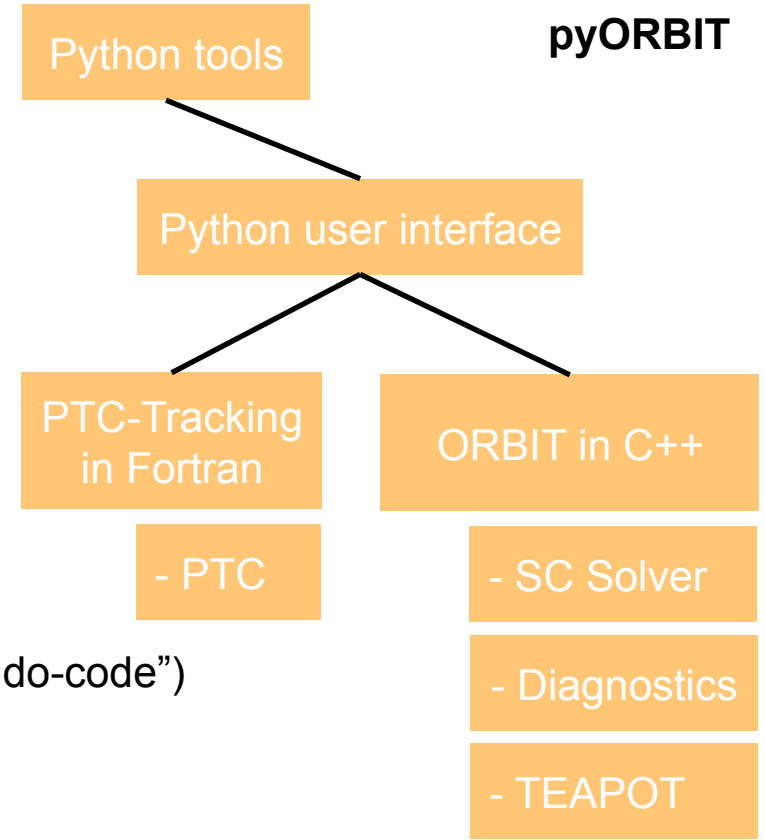
Modern PIC implementations

Advantage

- Use the simplicity and clarity of interpreted and high-level languages.
  - For scientific computing Python is very attractive (SciPy, NumPY, SymPy).

Code 'wrapping'

- Using Scripting languages and compiled code 'wrapping'
- The idea is to combine readability with fast language
- Cython created extension modules for Python for well-known & reliable codes in C,C++ & Fortran
- Examples are: pyORBIT, BlonD, ….

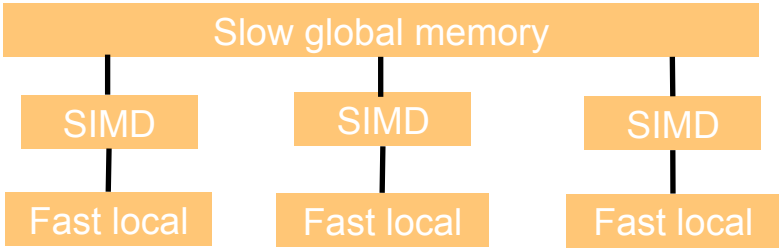


Rewrite codes (skeleton)

- The idea is to make the start of newcomers more simpler
- 2D Solver have only 30-40 lines in python ("executable pseudo-code")
- With modern tools one can reach a similar speedup
- Examples are: pyPATRIC

Millman et al, CISE 13 2011; Shishlo et al ICAP09

High-Performance Computing

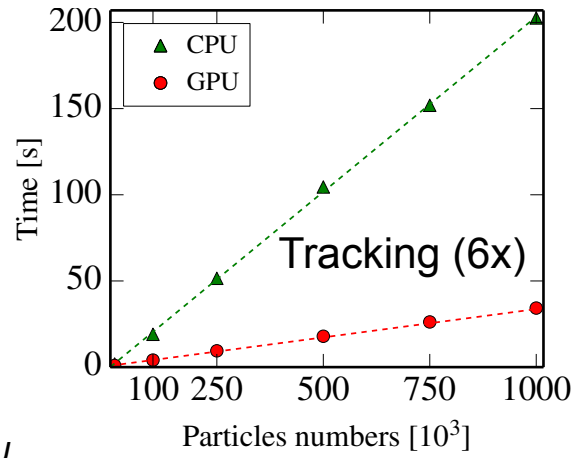


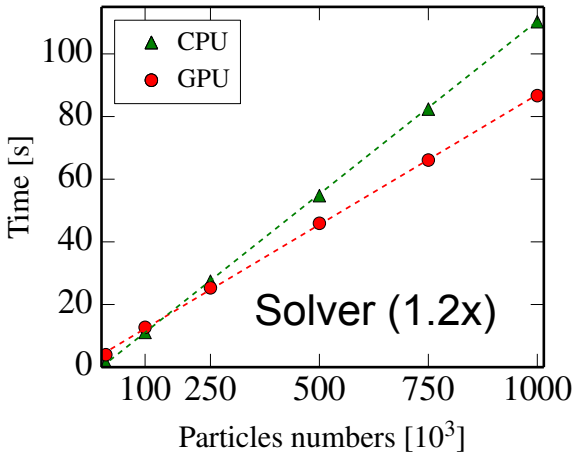
Parallelization of PIC with GPU-Programming

- Graphics Processing Unit (GPU)
- Particles are independent, good parallelization possible
- Collective effects are more difficult to accelerate
  - Load balance must be preserved
  - Communication should be low (different memories)

Parallelization of 1D SC solver

- Fast FFT algorithms exist also for GPUs
- Problem is the interpolation of particles
  - Many particles need to update the same grid point





V. Decyk, CISE 17, 2017; E. Carmona In: Concurrency: Practice and Experience 9 (1997);
J. Fitzek, GPU Technology Conference (2014); K. Amyx, GPU Technology Conference (2012)

Free available Codes (incomplete)

- **BLonD**:
  http://blond.web.cern.ch/
  - CERN
  - Acceleration, multiple RF systems
  - 1D space charge
  - Language: Python, C

- **MAD-X**:
  http://mad.web.cern.ch/mad/
  - CERN
  - Nonlinear and linear tracking
  - Frozen space charge
  - Language: Fortran, C

- **MICROMAP**:
  http://web-docs.gsi.de/~giuliano/
  - GSI
  - Nonlinear and linear tracking
  - Frozen + 2D SC solver
  - Language: Fortran

- **ORBIT**:
  http://web.ornl.gov/~jzh/JHolmes/ORBIT.html
  - SNS
  - Nonlinear and linear tracking, RF systems
  - 1D, 2D and 2.5D SC solver
  - Language: C++, SuperCode

- **pyORBIT***:
  https://code.google.com/p/py-orbit/
  - SNS
  - Script language: Python, C++
  - At this moment only few capabilities of the original ORBIT are implemented.
    - 1D, 2D and 2.5 SC solver are available

- **Synergia**:
  https://web.fnal.gov/sites/Synergia/SitePages/Synergia%20Home.aspx
  - Fermilab
  - Nonlinear and linear tracking
  - 2D and 3D SC solver
  - Language: Python, C++

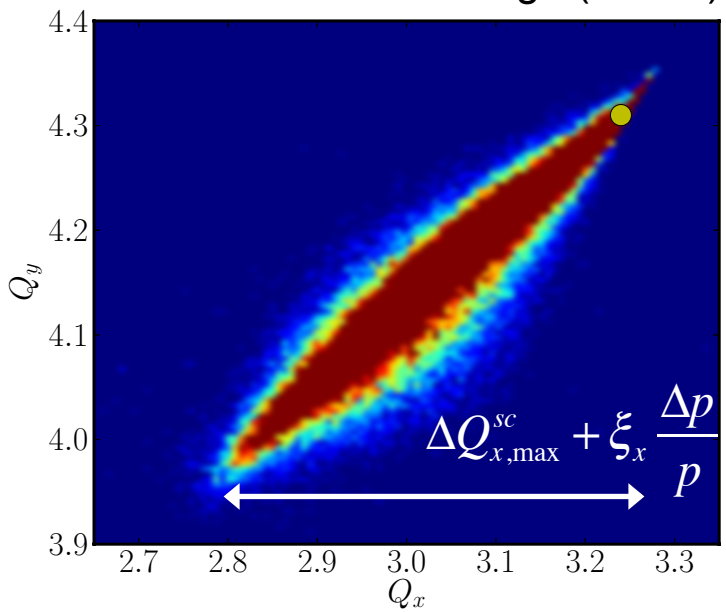
* Used by myself for space charge simulations (+ LOBO, PATRIC)

Summary

- Space charge effects determine beam parameters and accelerator components

- The PIC algorithm is very popular to simulate SC effects
  - 1D Solver: Longitudinal coupling impedance
  - 2/3D Solver: Poisson equation is solved with the Green's function

Tune footprint with space charge (SIS18)



- Modern PIC implementations
  - Code 'wrapping'
  - Parallelization of PIC with GPU-Programming

- Not addressed
  - Iterative solvers, direct Vlasov solvers, Δf-PIC solvers

Thank you for your attention