Bridging Elegant to Genesis1.3 simulations with Pseudo-one4one

- FLASH2020+ Start to End Simulation Workshop
- A presentation by Mihai Pop

Overview

- Describe the issue
  - Why we use Elegant
  - What is the current way of importing distribution from ELEGANT to genesis
  - Importing phase spaces with fine features
- Present a solution
  - Feature and pitfalls
- Discuss the solution

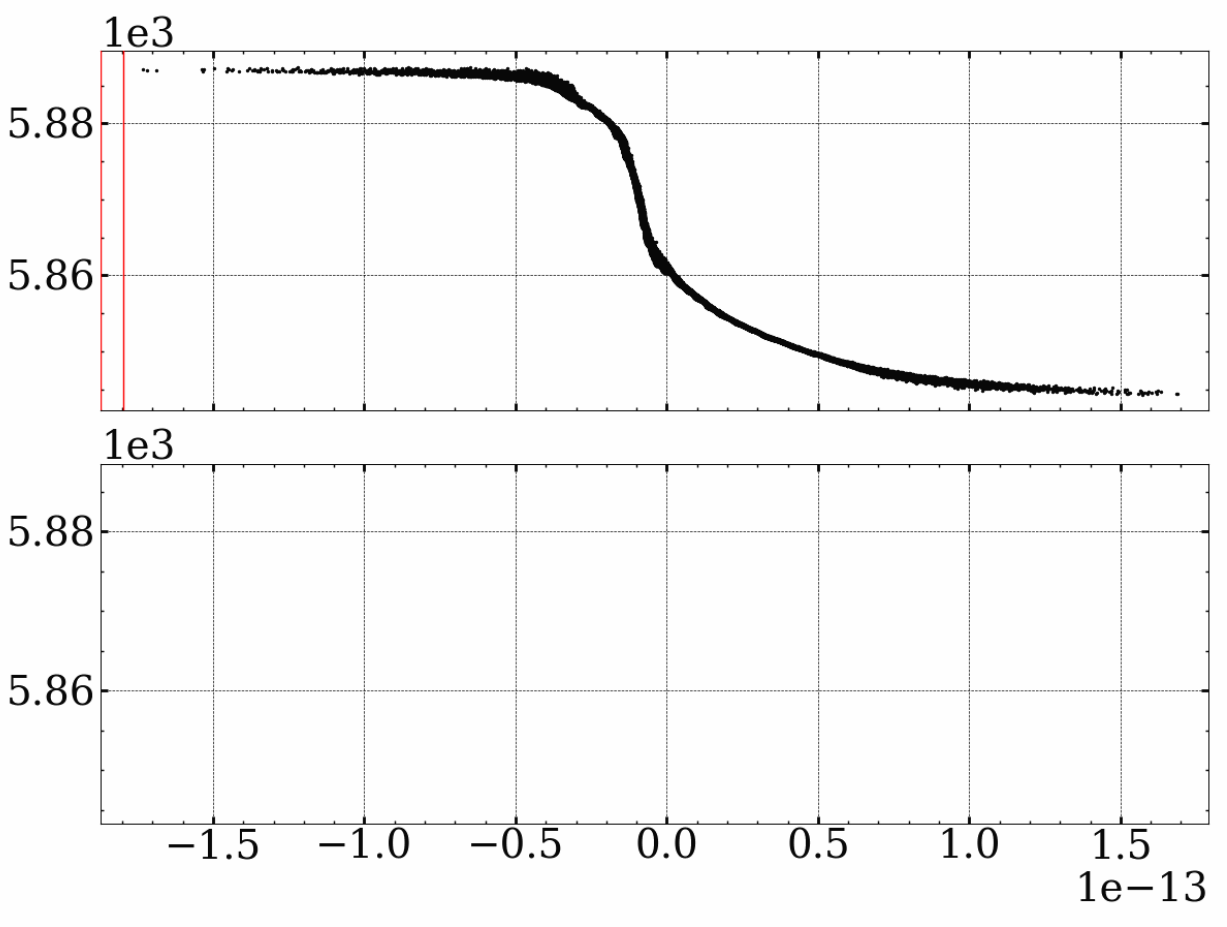
Current ways of importing from ELEGANT

- Elegant2genesis (sdds command)
  - Computes statistical slice analysis
  - Importing to genesis is done by importing each profile from the computed slices
  - Runs Genesis one4one
- Importdistribution (genesis command)
  - Difference w.r.t. elegant2genesis
  - Genesis reads the particle file and creates the profile
    - Uses all created profiles of all 6D :
    - Uses smoothing
    - Can match the twiss center distributions
    - Can run Genesis one4one

Why we use Elegant

- From Gun to undulators
- Nice matching routines
- Can simulate Collective effects
- Already some handshaking with Genesis 1.3
- A Legacy code at within our group
- Good scanning capabilities

The sliding window averaging in importing



Start2end beam for SXL

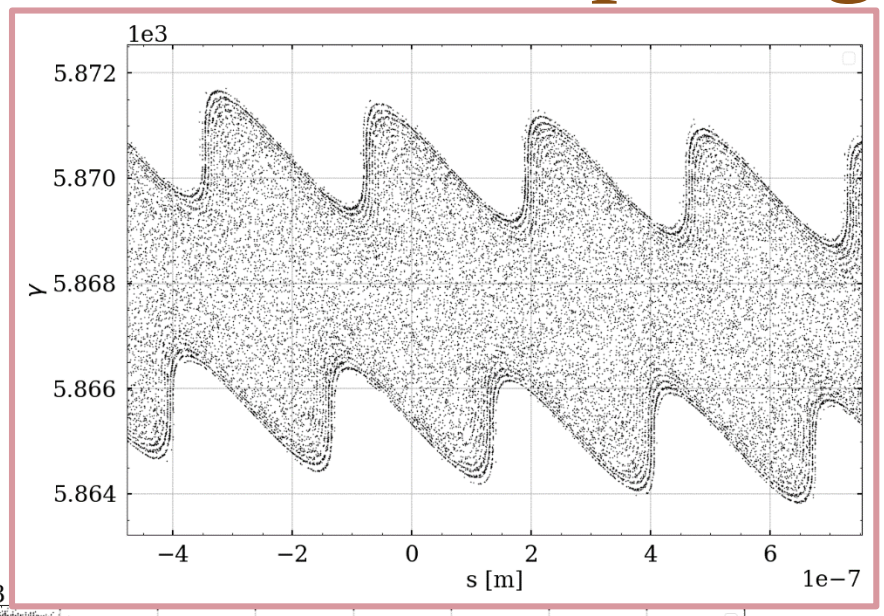
Imported with &importdistribution

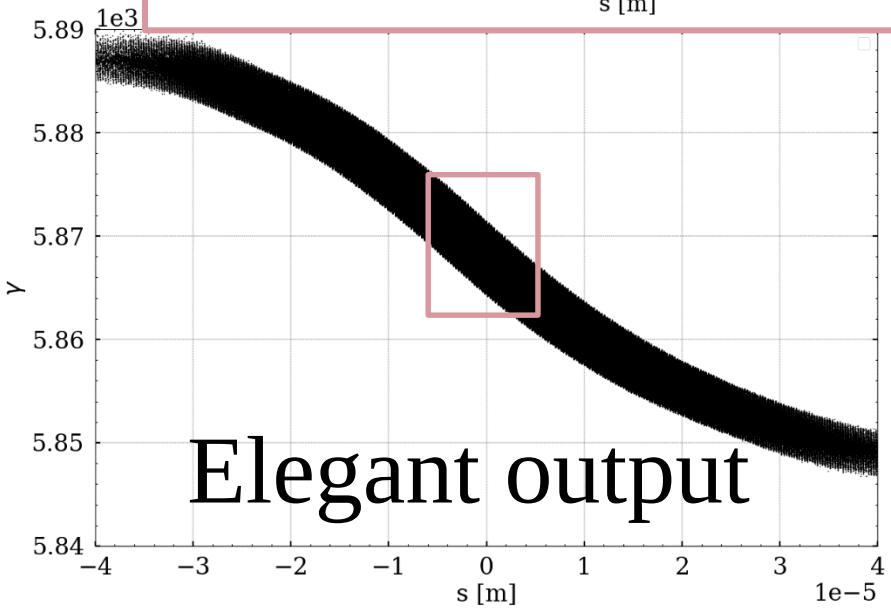
The interface between ELEGANT and Genesis

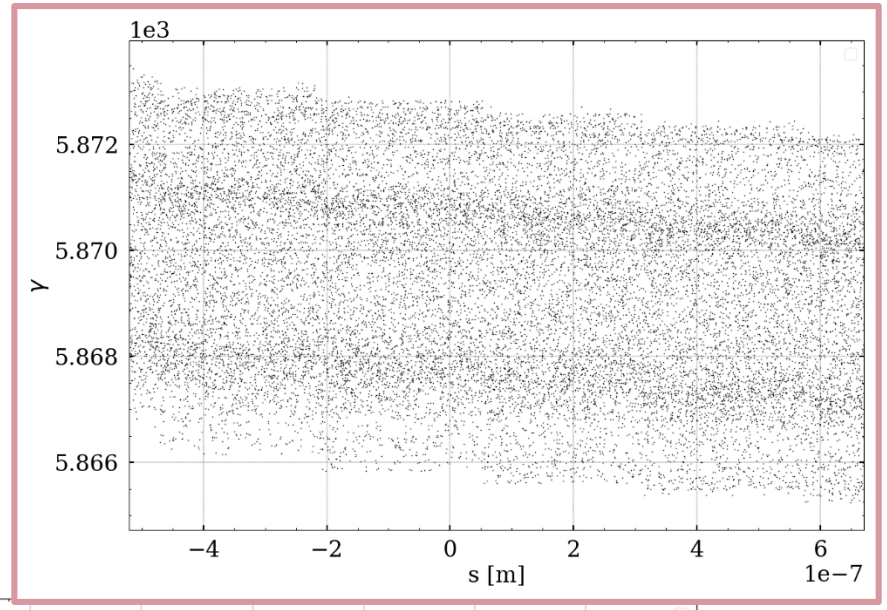
- ELEGANT is a widely used particle tracking code that can take collective effects into account. ELEGANT simulations are useful for:
  - Start2end FEL simulations (our group uses ASTRA, ELEGANT and Genesis1.3)
  - Seeding schemes that use phase space manipulations need to take into account collective effects such as CSR and intra-beam scattering.

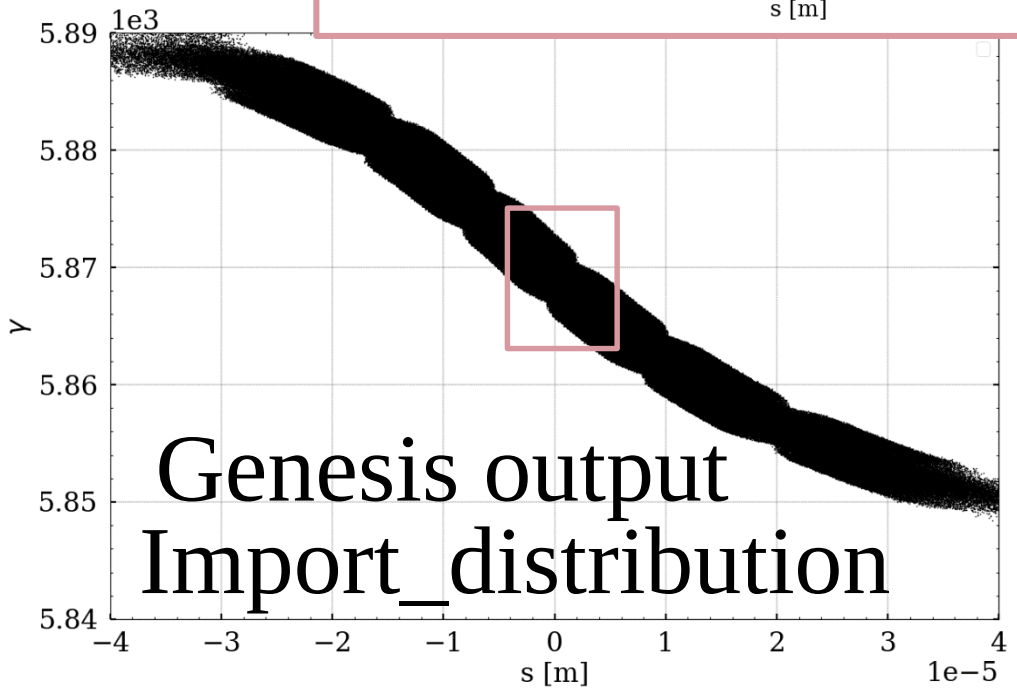
- The key features in the phase space distribution that are not handled well with standard importing:
  - The wavy stripes from seeding schemes such as EEHG
  - ~~Correlation between different coordinates (like chirp) that have sufficient variation inside one simulation slice in Genesis~~

Comparing phase space







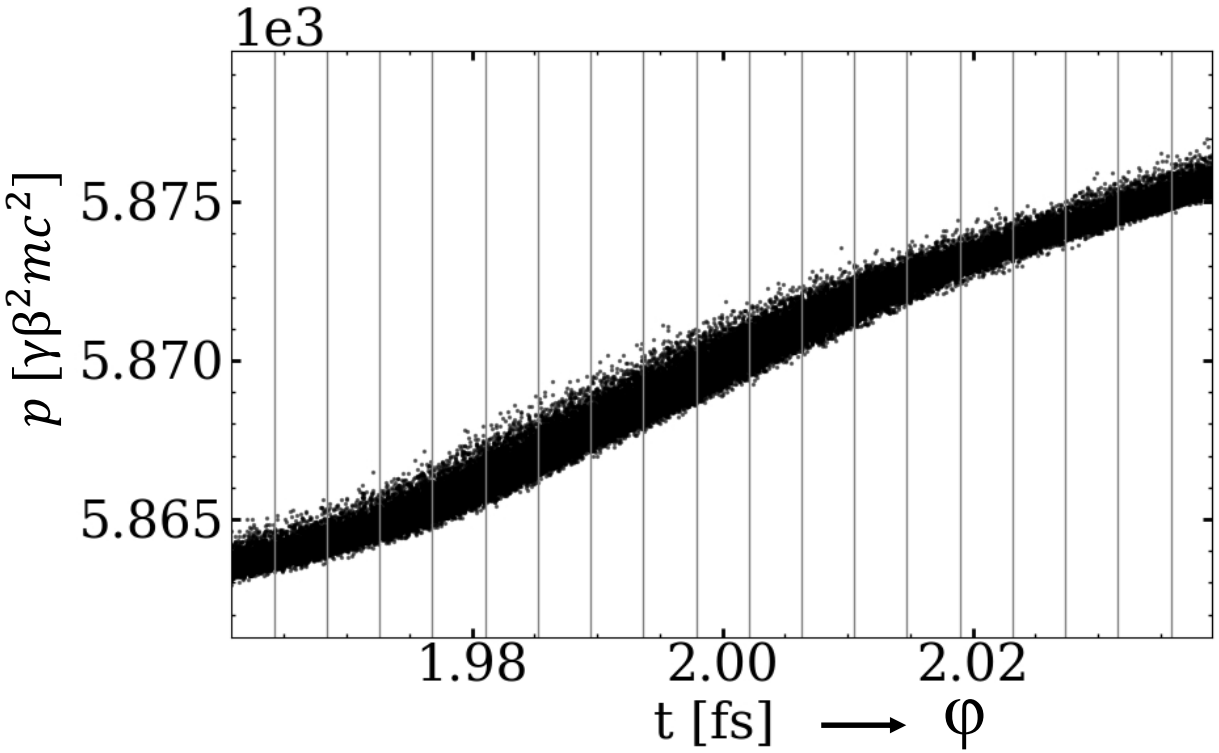


Another option

- Run Genesis one4one simulations with the actual particles used in ELEGANT
- Create my own *.par.h5 file for Genesis simulation from an ELEGANT output
- Assign to each slice a value for the current as if the macroparticles in Genesis simulations have the same charge as in ELEGANT (In Genesis all particles have the same charge as in ELEGANT).
- As a result Genesis runs with the same number of particles

Steps to create .par file

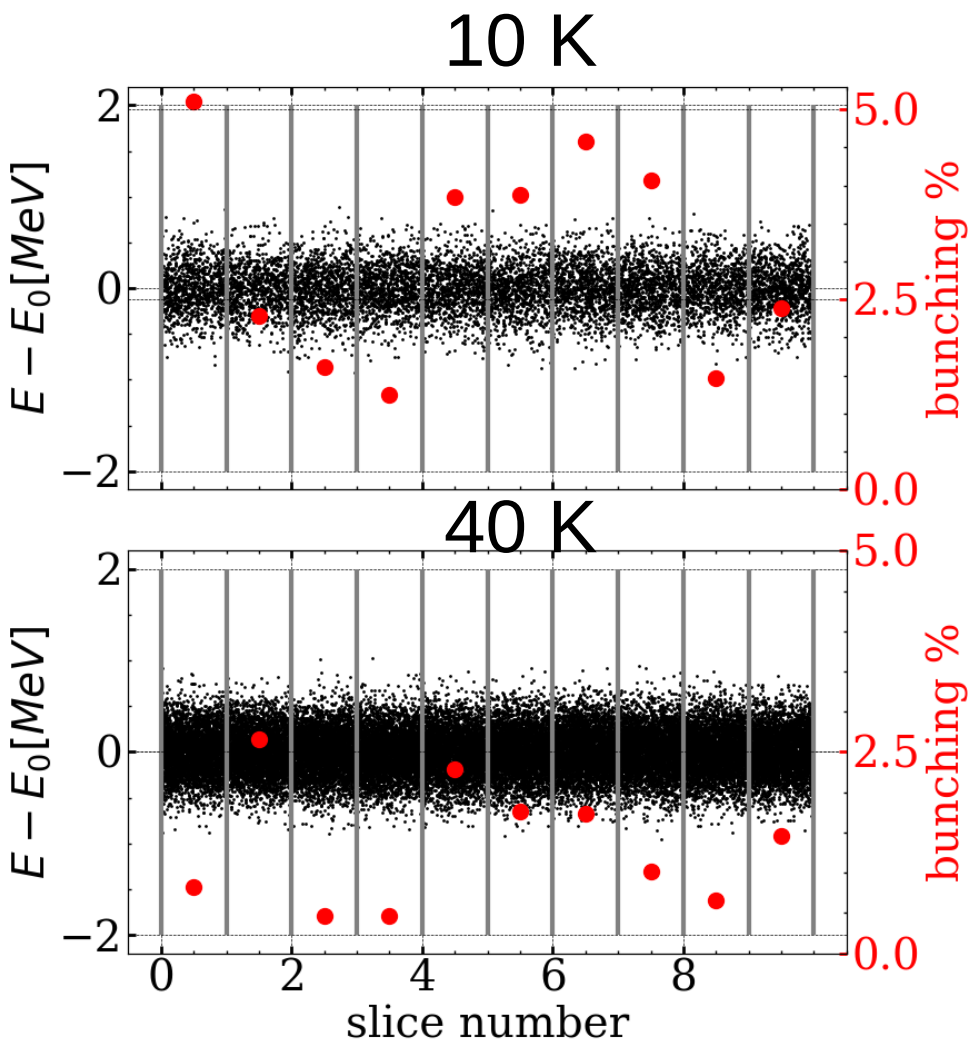
- Sort particles by time
- Put each particle in its corresponding slice
- Change t to $\varphi = mod(t * c, \lambda_{rad}) * 2\pi$
- Change px,py as px=xp*p
- Gamma=p+1 (as I saw in the import distribution function Genesis source code)
- Allow a few slices after particle distribution for radiation slippage
- Assign current for each slice



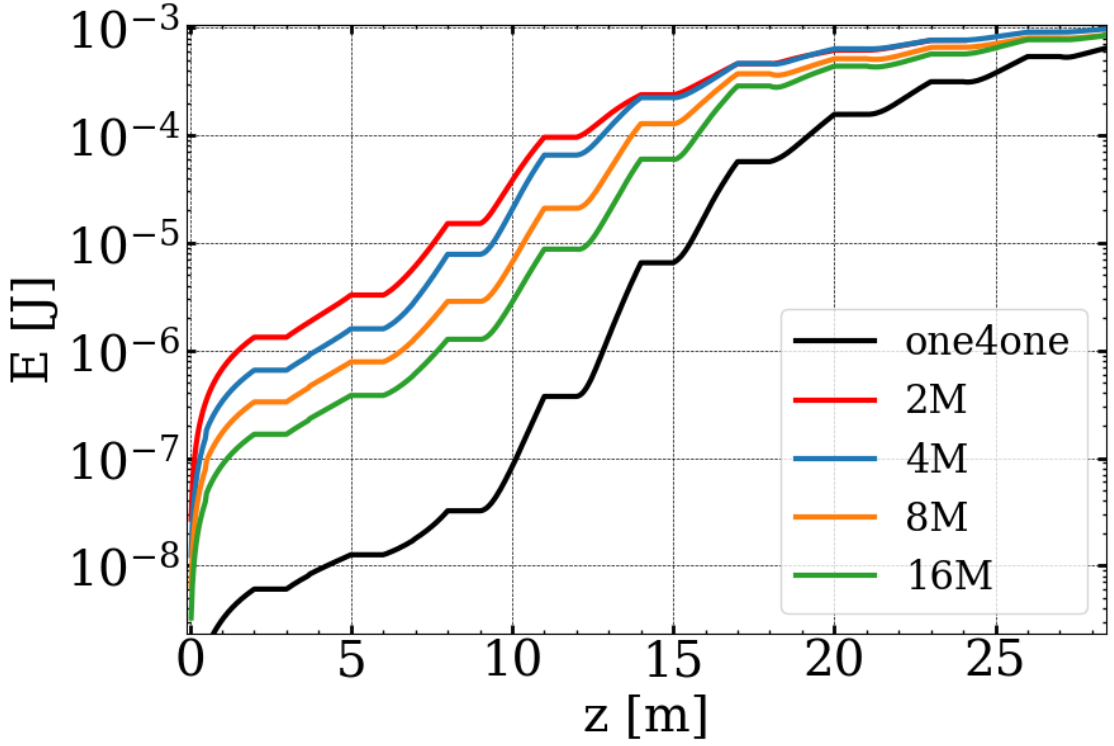
Elegant	Genesis
t 5.294e-14 [s]	phi 8.041e-01 [rad]
p 5.887e+03 [$\gamma\beta^2 mc^2$]	γ 5.888e+03 []
x -3.163e-05 [m]	x -3.163e-05 [m]
xp 3.783e-06 []	px 2.227e-02 []
y 2.206e-05 [m]	y 2.206e-05 [m]
yp -3.278e-06 []	py -1.930e-02 []

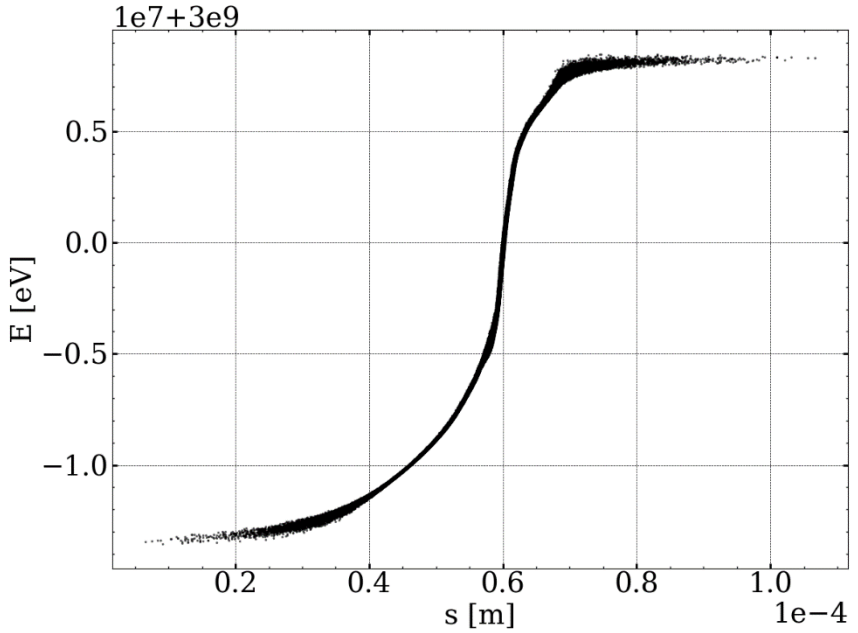
Shot noise with reduced number of particles

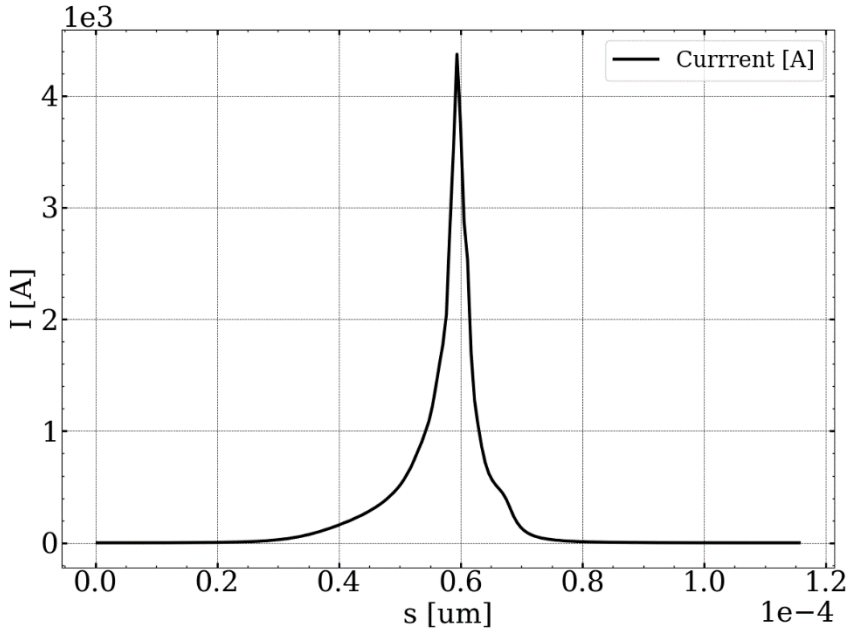
- Assuming we have different number of macroparticles distributed randomly
- The variation bunching scales with $\sqrt{N}$
- Therefore in the case of SASE initial shot-noise bunching is over-estimated



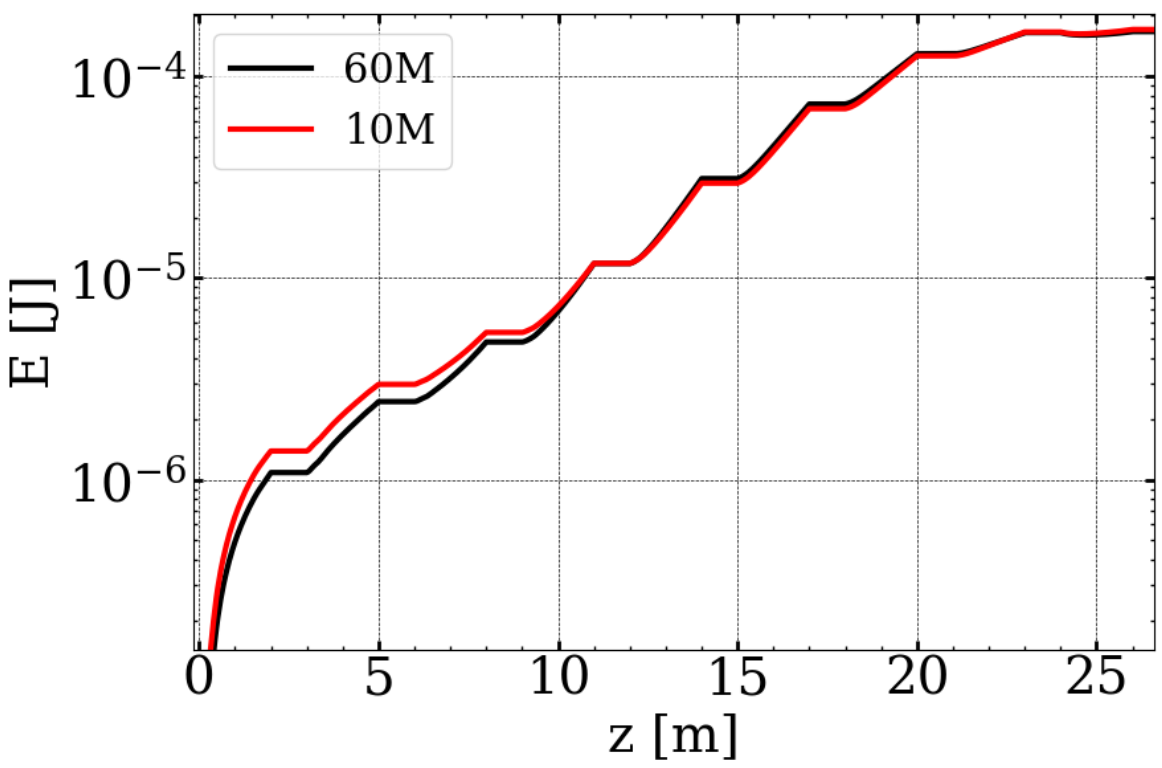
SASE simulation for different number of macroparticles

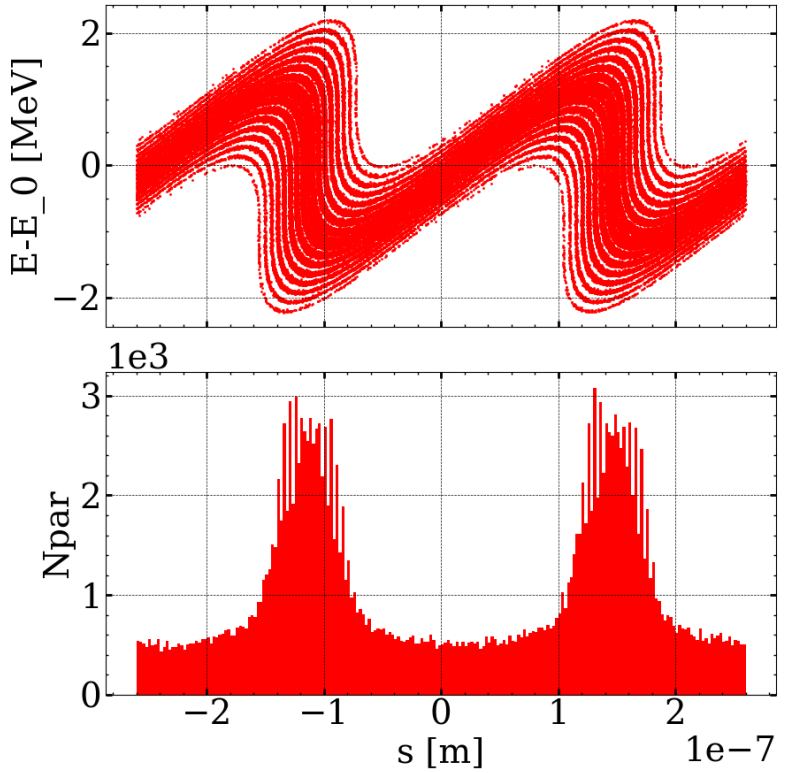






For a seeded EEHG case





No CSR, benchmark with one4one (60M)

Discussion

- Not suitable for reducing the numbers of macroparticles in SASE simulations
- Converges for seeded simulations between 1/8 and 1/6 number of particles
- Idea after yesterday's talks : multiple Elegant runs to add up to one4one
  - Might smear bunching created by EEHG scheme (needs investigating)
- Issues I am unsure of:
  - Is the energy conserved in my simulations (no indication that it isn't) solving the coupled field particle equations the mass of the particles is m_e but the charge is n*e
  - Planning to run Genesis with exactly one4one by duplicating the macro-particles until the correct number is reached. Will not help with the shot-noise but maybe with the energy conservation?