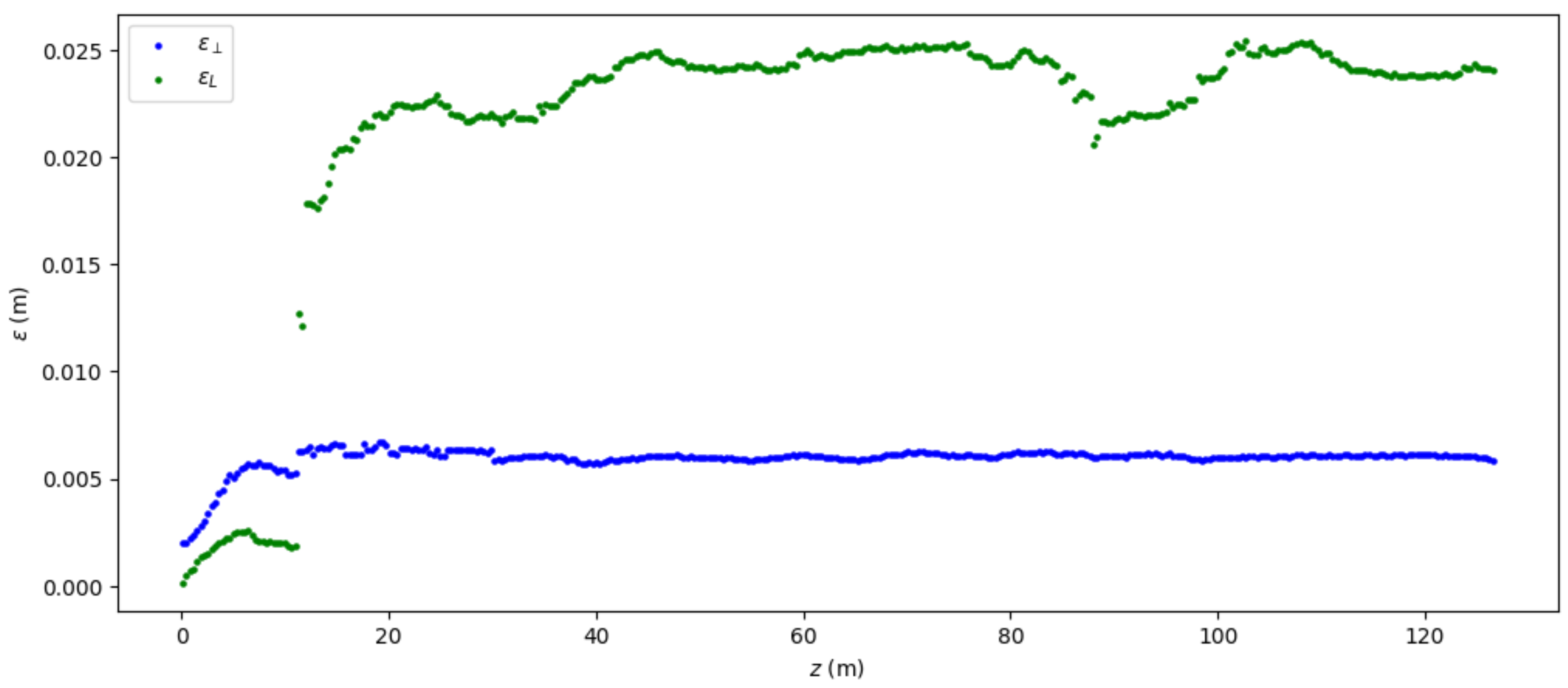
MUON COOLING PROJECT UPDATES

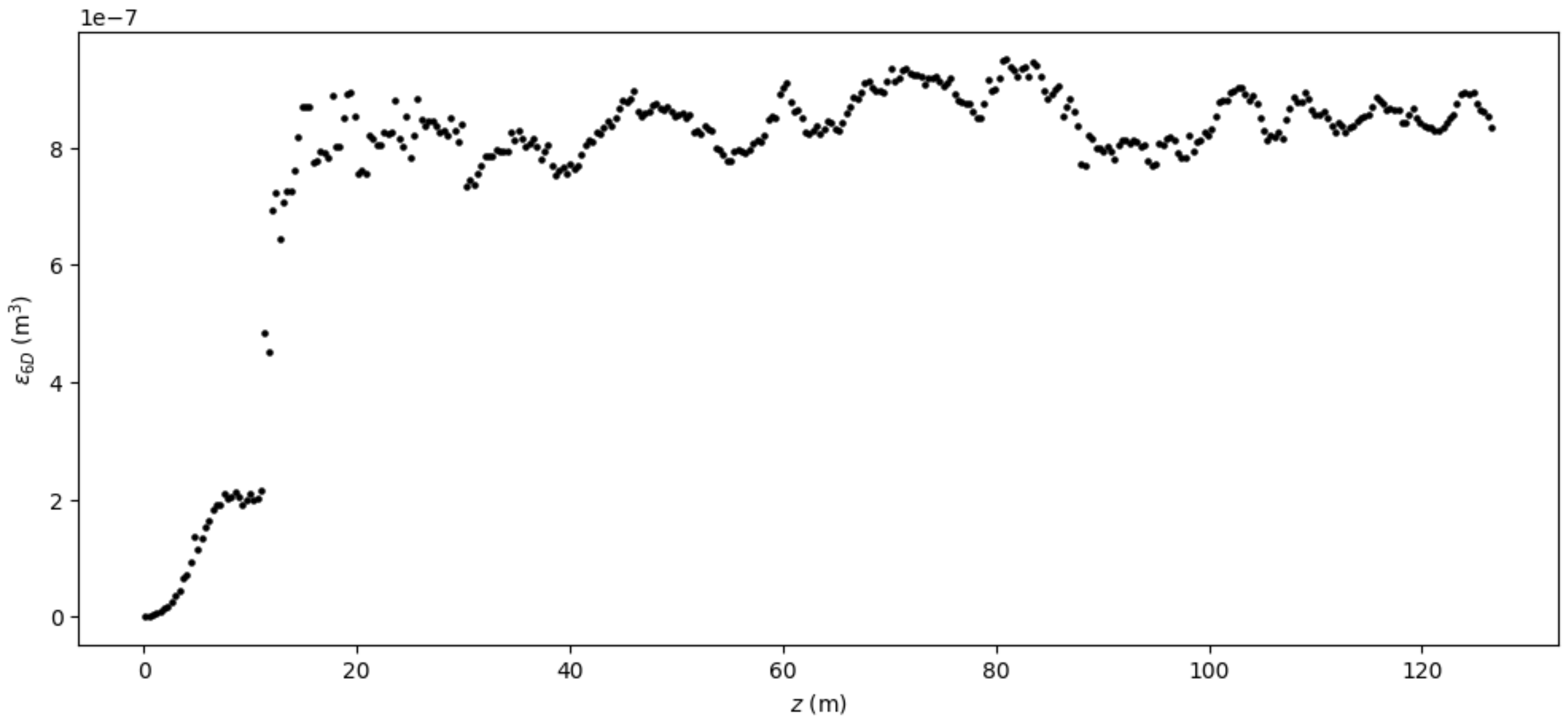
Week of June 30 - July 4, 2025

https://github.com/criggall/muon-cooling/tree/main

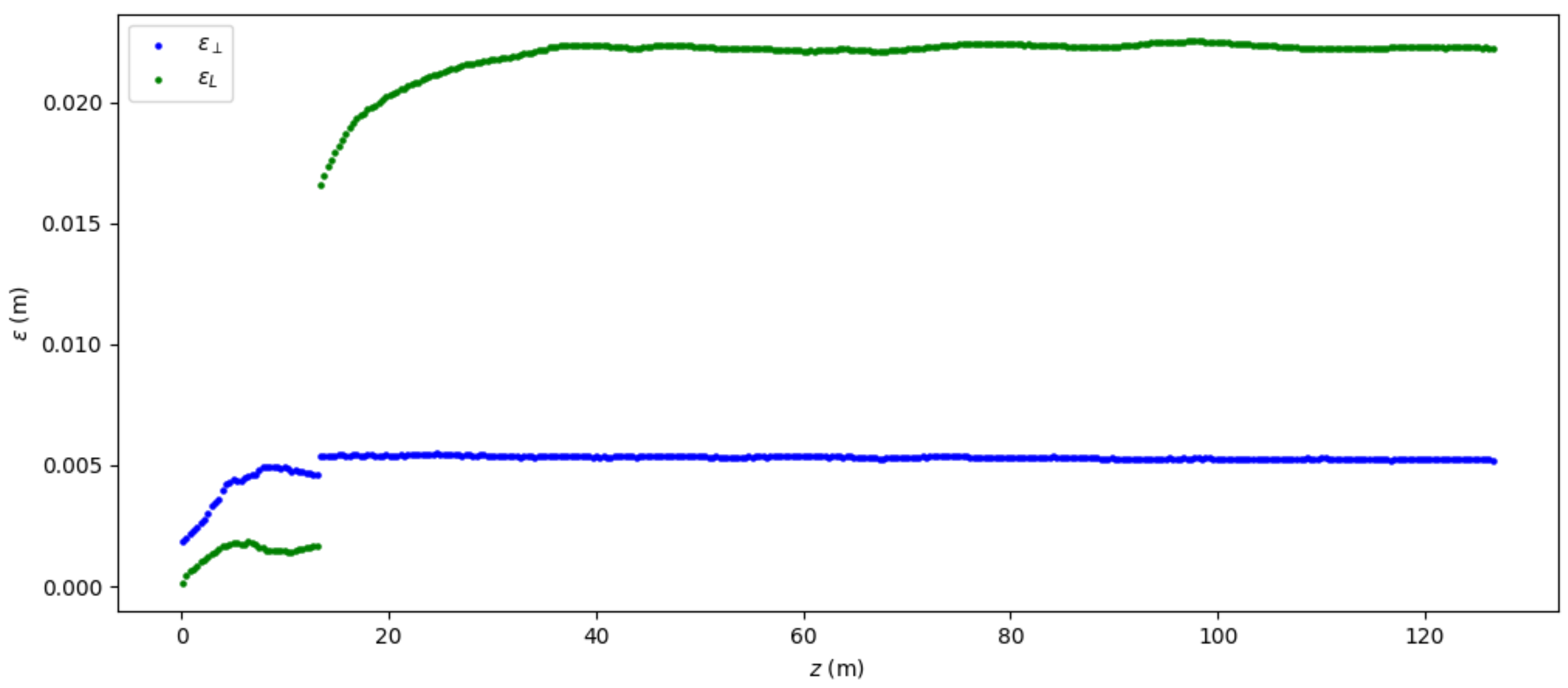
BRIEFLY REVISITING ECALC9F

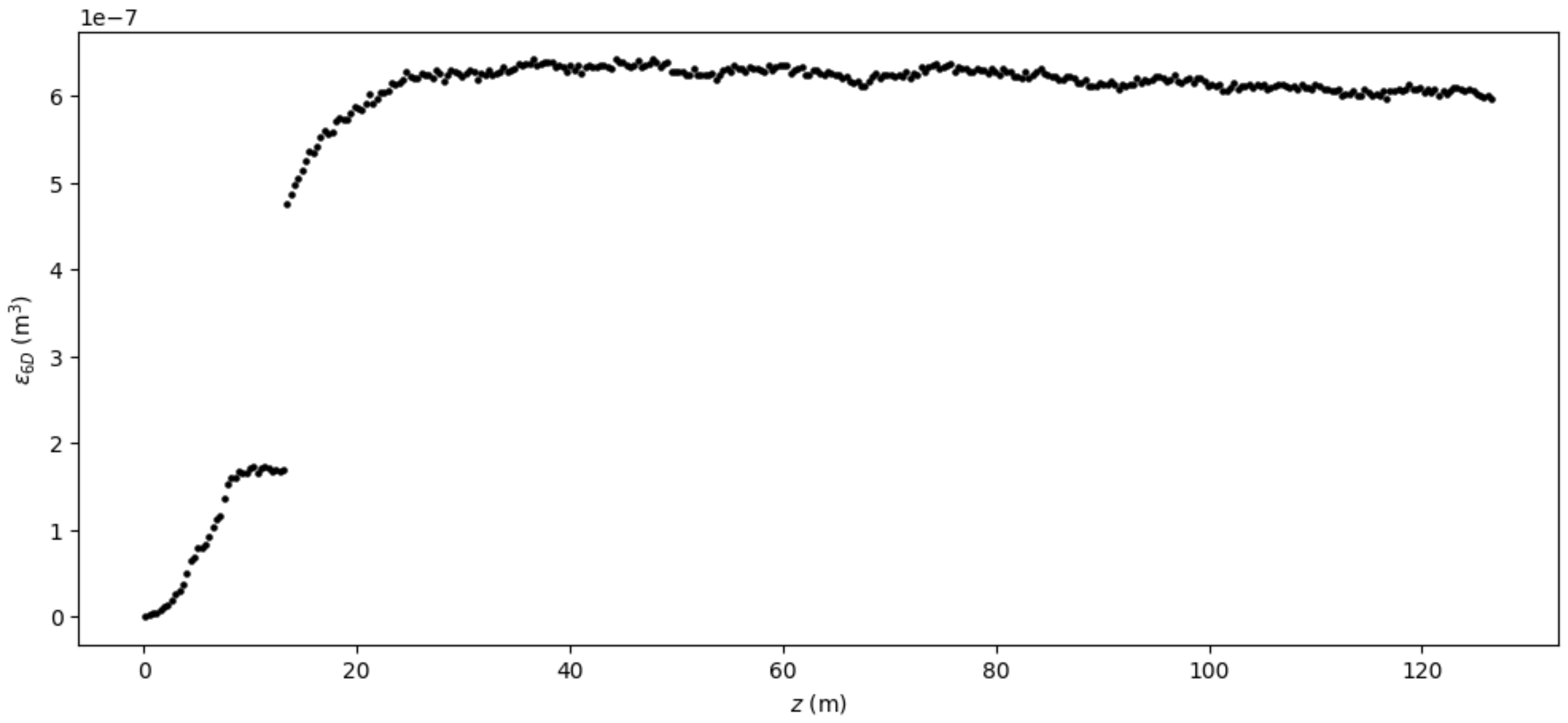
From last week, with 100 events:



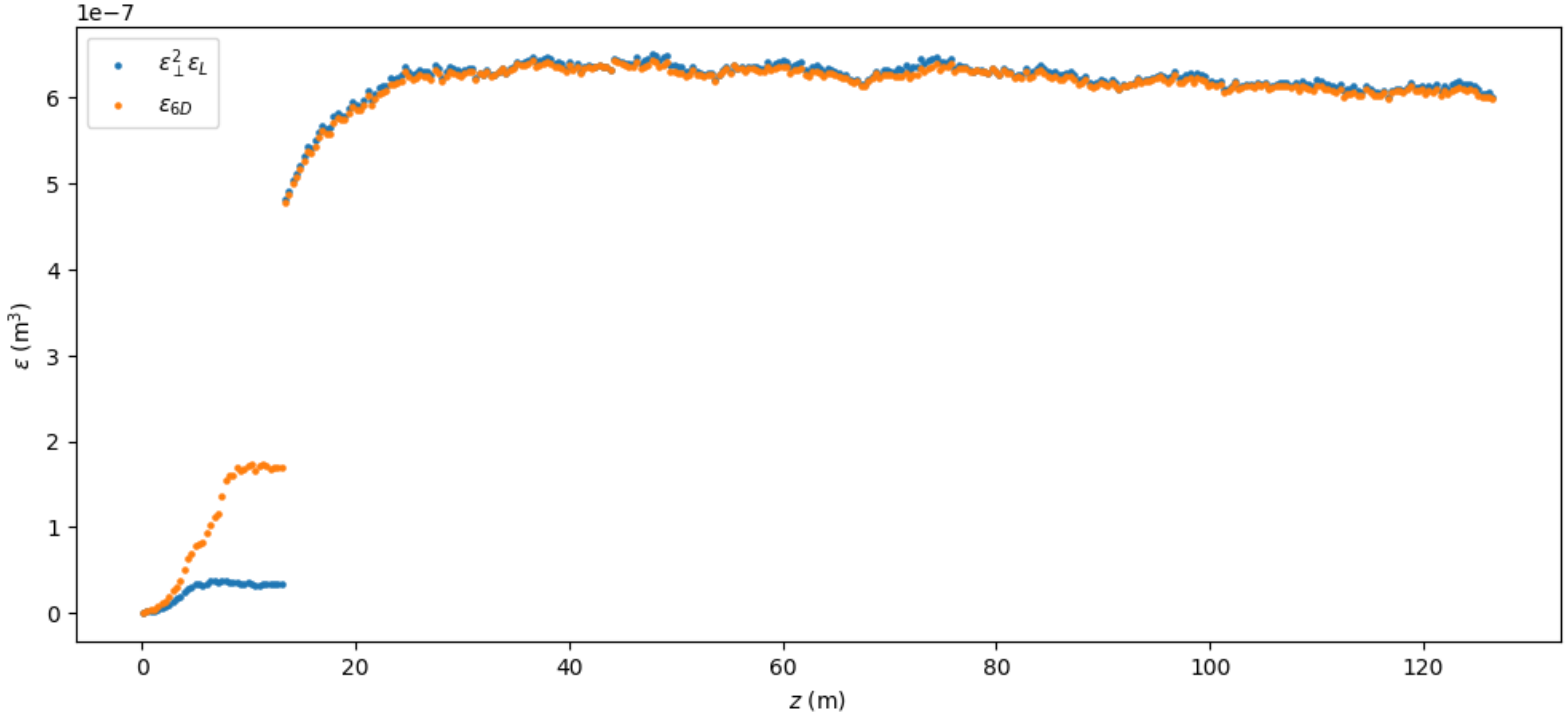


Now with 10,000 events:





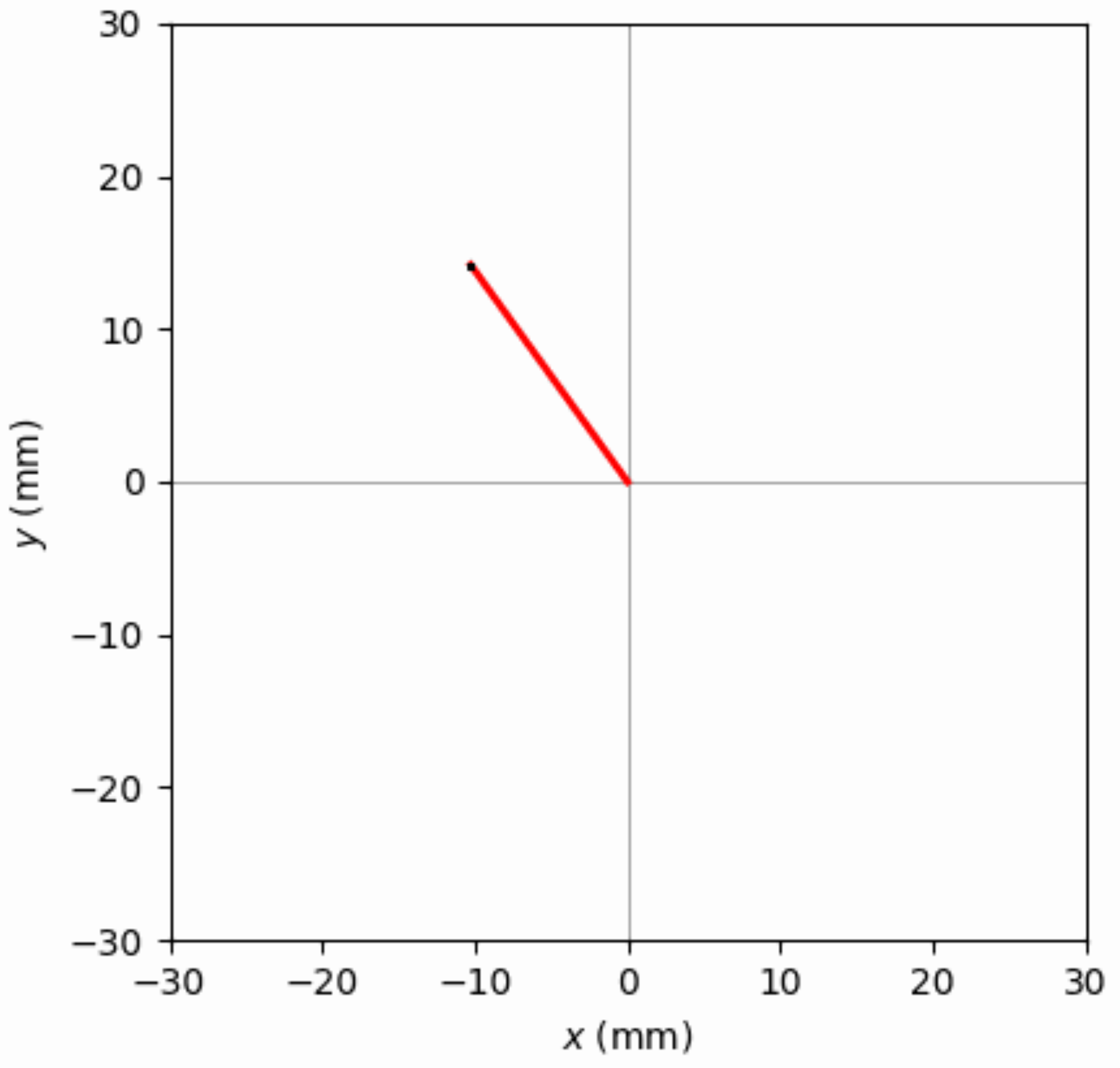
BRIEFLY REVISITING ECALC9F

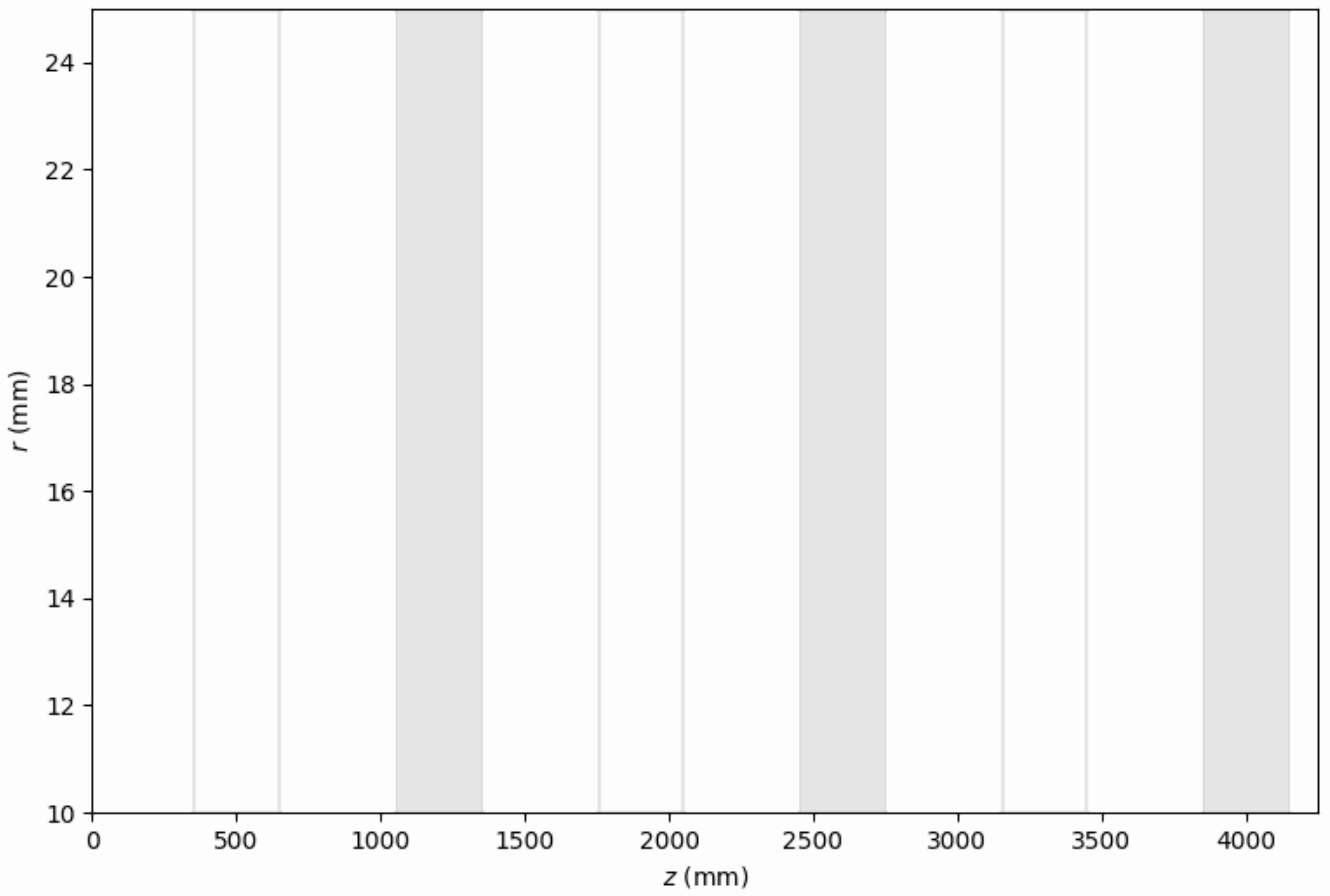


CONSIDERING DYNAMICS IN POLAR COORDINATES

$$r = \sqrt{x^2 + y^2}$$

(Animation)

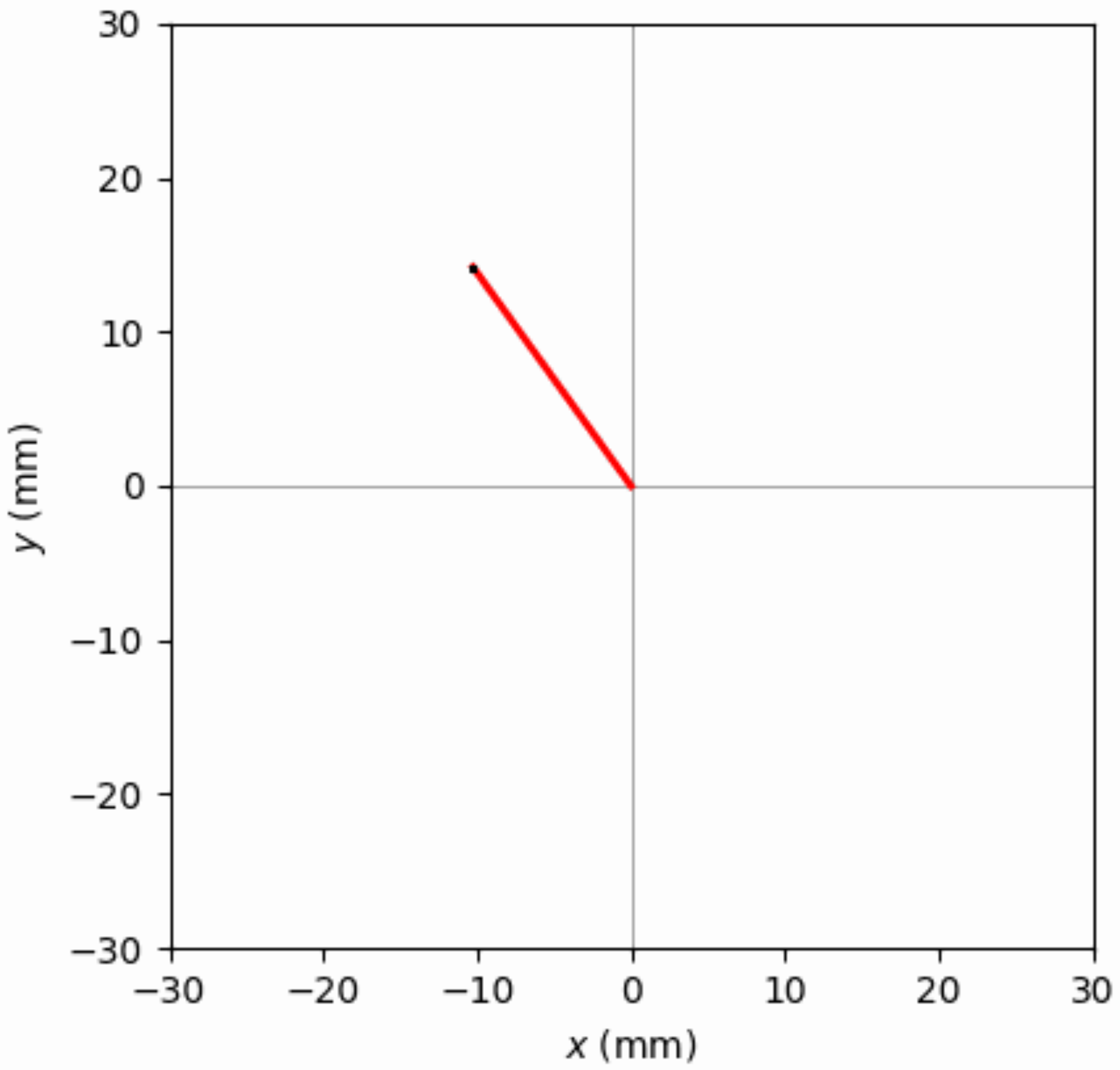


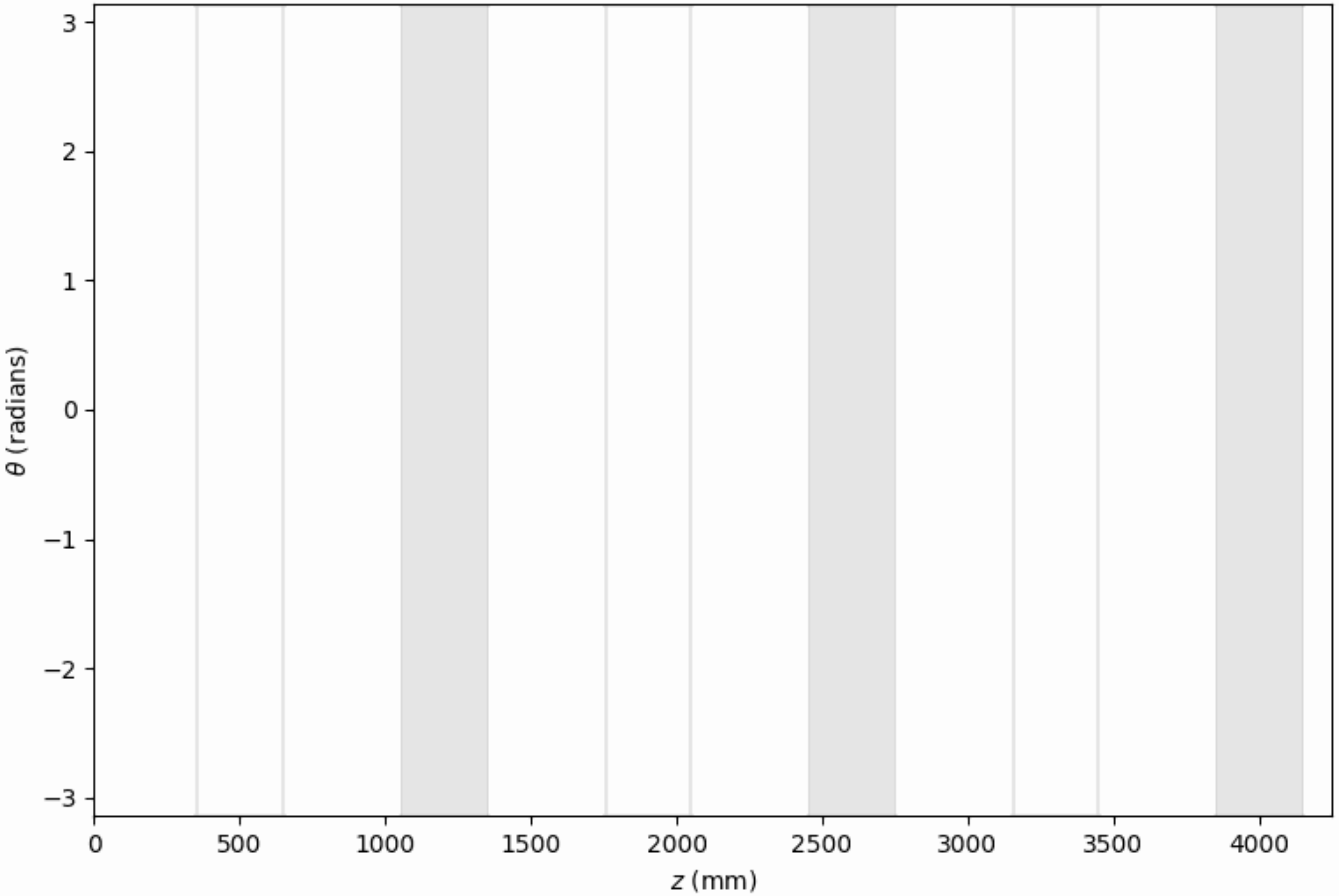


CONSIDERING DYNAMICS IN POLAR COORDINATES

$$\theta = \arctan\left(\frac{y}{x}\right)$$

(Animation)





ICOOL EMITTANCE ROUTINES

Conclusion from last week: we do not want to use *ecalc9* or *ecalc9f* due to the assumption of azimuthal symmetry which is not applicable to this system

ecalc9.for	**Gregg Penn's original postprocessor for calculating emittance with standard cuts.**
ecalc9f.*	**A modified version of Gregg Penn's postprocessor that reads a second file containing the parameter values for the analysis.**
emitcalc.*	**Gregg Penn's postprocessor for calculating emittance with standard cuts in all three phase space planes. The routine reads a second file containing the parameter values for the analysis.**
ecalcxy.*	**Dave Neuffer's postprocessor for calculating emittance in channels that are not azimuthally symmetric. This is more suitable for channels containing dipoles and quadrupoles than ECALC9F. The routine reads a second file containing the parameter values for the analysis.**
eigemit.*	**This routine computes eigenemittances along with other emittance and auxilliary quantities.**

Why do we want to use *emitcalc* over *eigemit*?

EMITCALC INPUTS

emitcalc.inp

```
HFOFO
for009.dat
emitcalc.out
2
0.100  0.350

9.75e-3  15e-3
false
false
0 3.25e8
4
```

emitcalc.man

```
Description of EMITCALC Program  (Gregg Penn)

This still needs to be checked that all of the optionswork properly;  the basic cases seem okay.Calcul

This program reads the input file 'emitcalc.inp'. An example is:
# sample    ! description
for009.dat  ! input file name
emitcalc.out    ! output file name
2       ! particle type (0 = use all)
0. 0.       ! pzmin, pzmax [GeV/c]; if pzmax<=0., no cut
0 2.0 vp.in ! vector pot. model, assumed B0 [T], input file
0.02  0.1   ! transverse, longitud. amplitude cutoffs [m] (if <=0, no cut)
.false.     ! subtract linear corr of transverse coords with E, t
.false.     ! subtract nonlinear corr of E, t with transverse amplitude
0  800.     ! time overlap model (no overlap if =0), rf frequency
6.      ! sigma cut to remove tails (no cut if <=0.)

The vector potential model (line 6) and time overlap model (line 10) may need some explaining.  The ve
0.  model 0 assumes a uniform solenoid field, which is given in the input file as the second term in l
1.  model 1 is roughly similar to what ECALC9F does, but with an extra correction that can be useful f
2.  model 2 also assumes axisymmetric fields, but calculates the vector potential by fitting the value
3.  model 3 allows one input a Taylor expansion of Ax and Ay from a file. The first line should be 2 n

The time overlap option is mostly as in ECALC9F, it overlaps particles in time modulo the given rf per
1.  model 1 uses the reference particle to define t=0, if there is no reference particle, model 2 is u
2. model 2 uses a weighting scheme (think of periodic time as the circumference of a circle) to find t

Other comments:The "asymmetry" output is a test to see how axisymmetric the beam is, it should be a nu
```

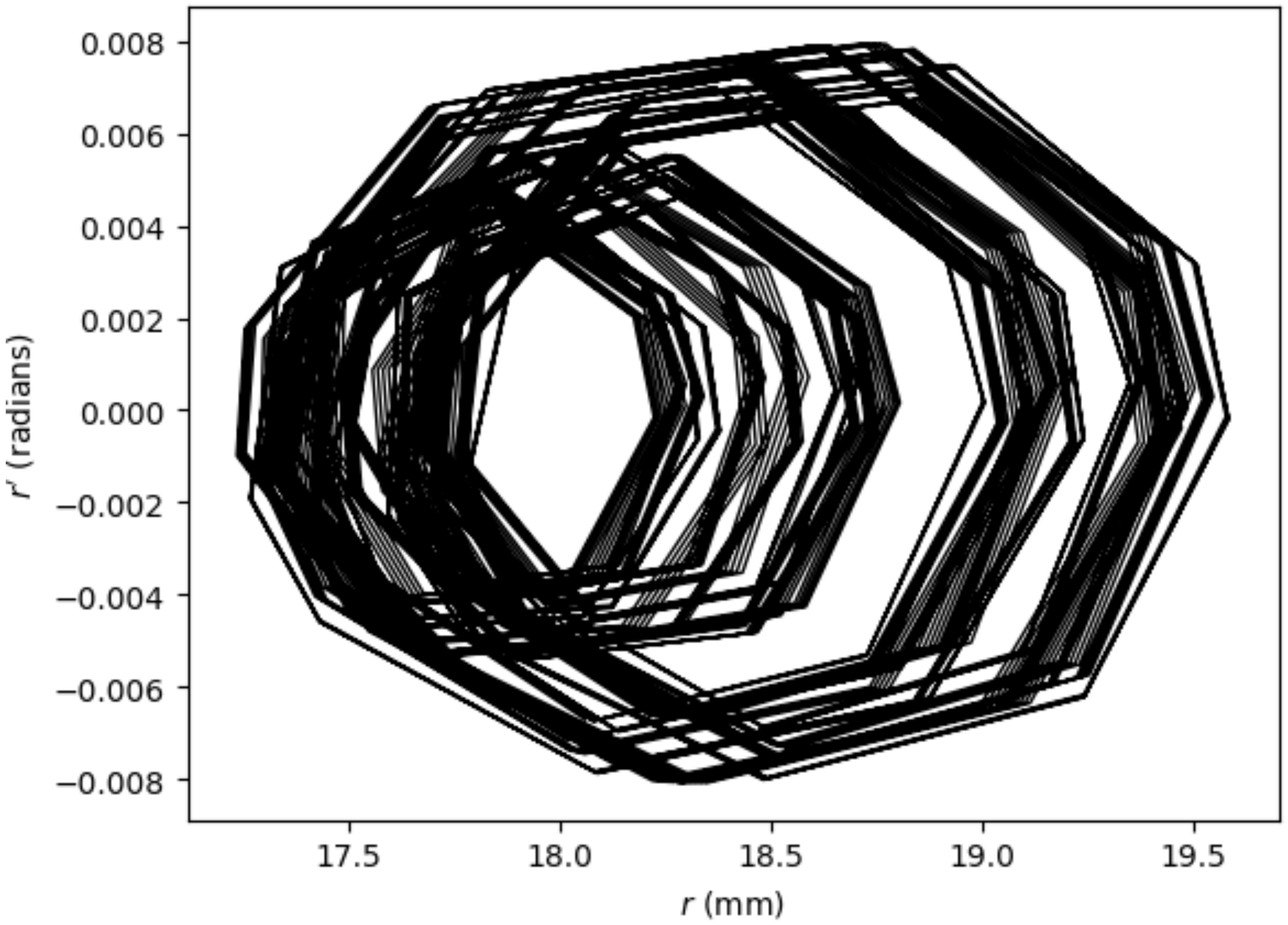
REVISITING POLAR PHASE SPACE

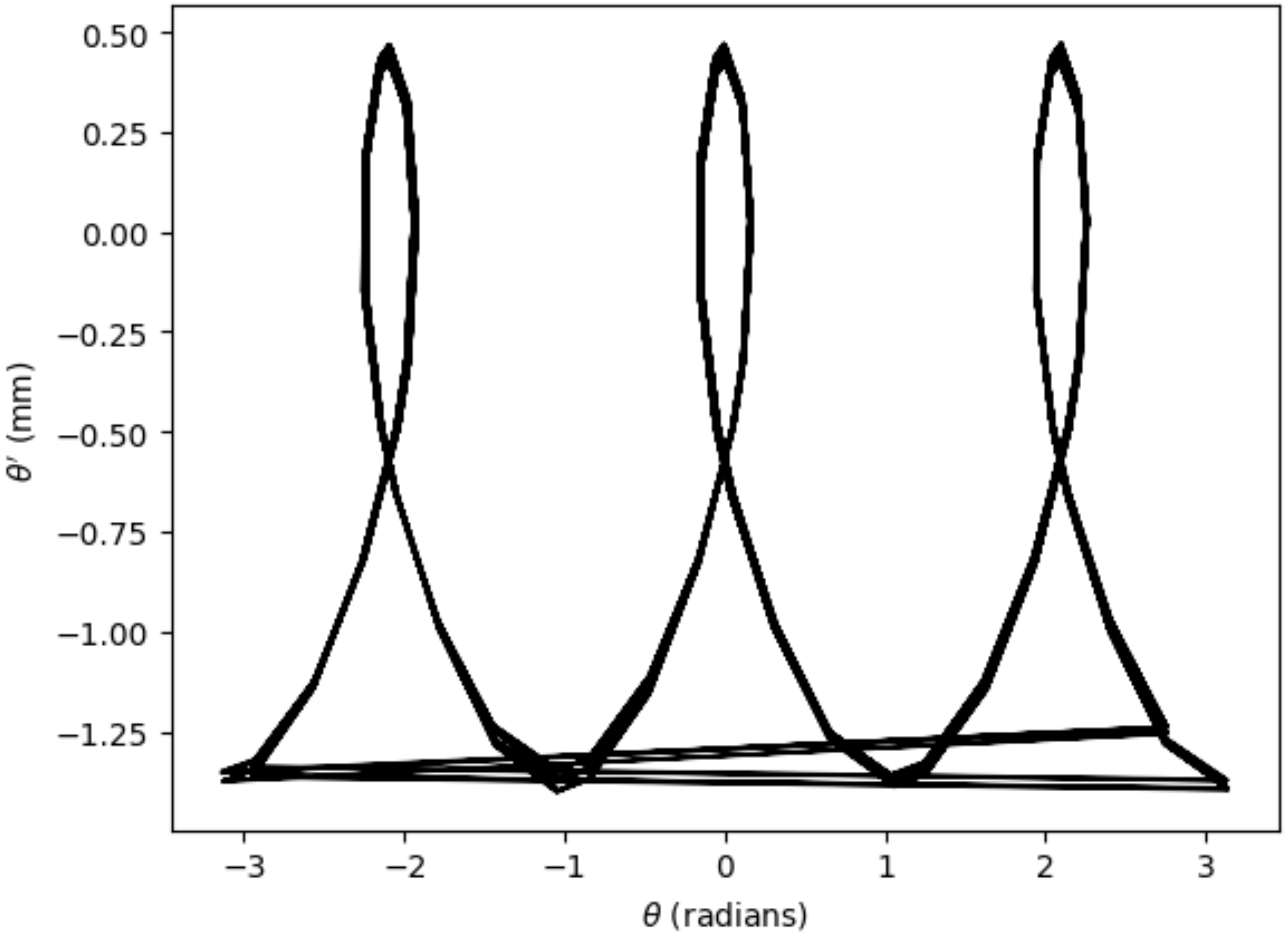
$$r = \sqrt{x^2 + y^2}$$

$$r' = \cos\theta x' + \sin\theta y'$$

$$\theta = \arctan\left(\frac{y}{x}\right)$$

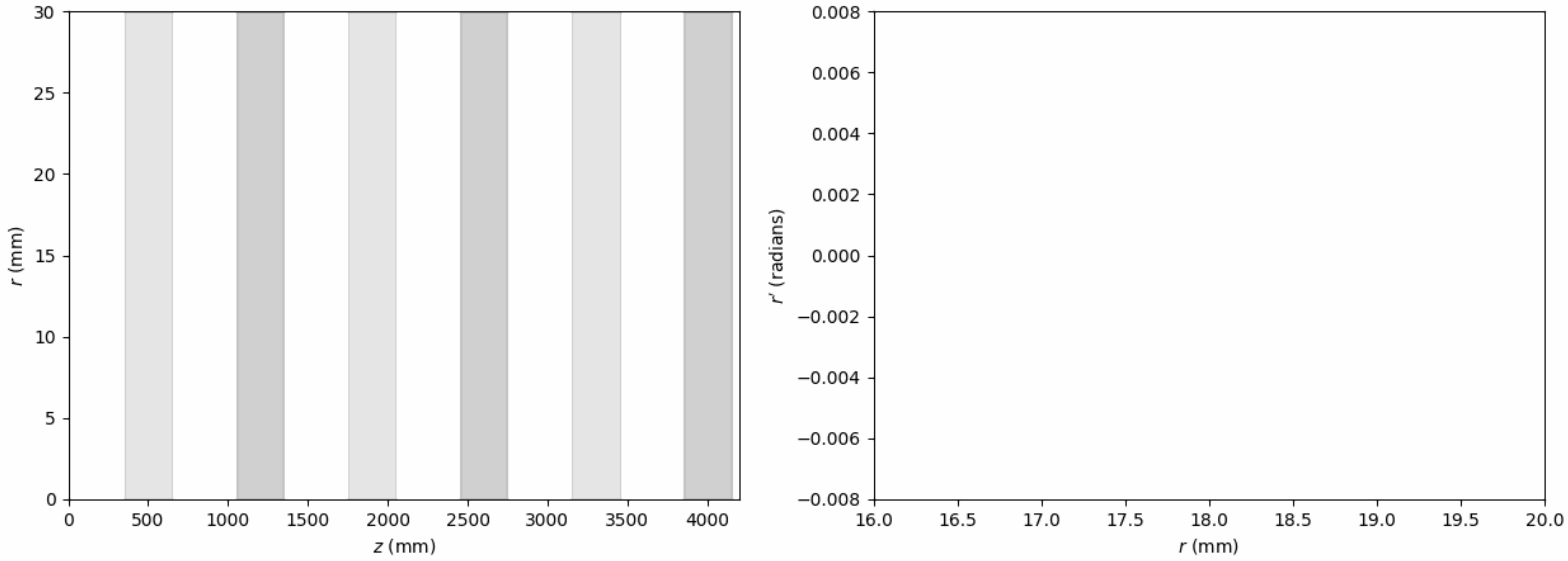
$$\theta' = -r\sin\theta x' + r\cos\theta y'$$





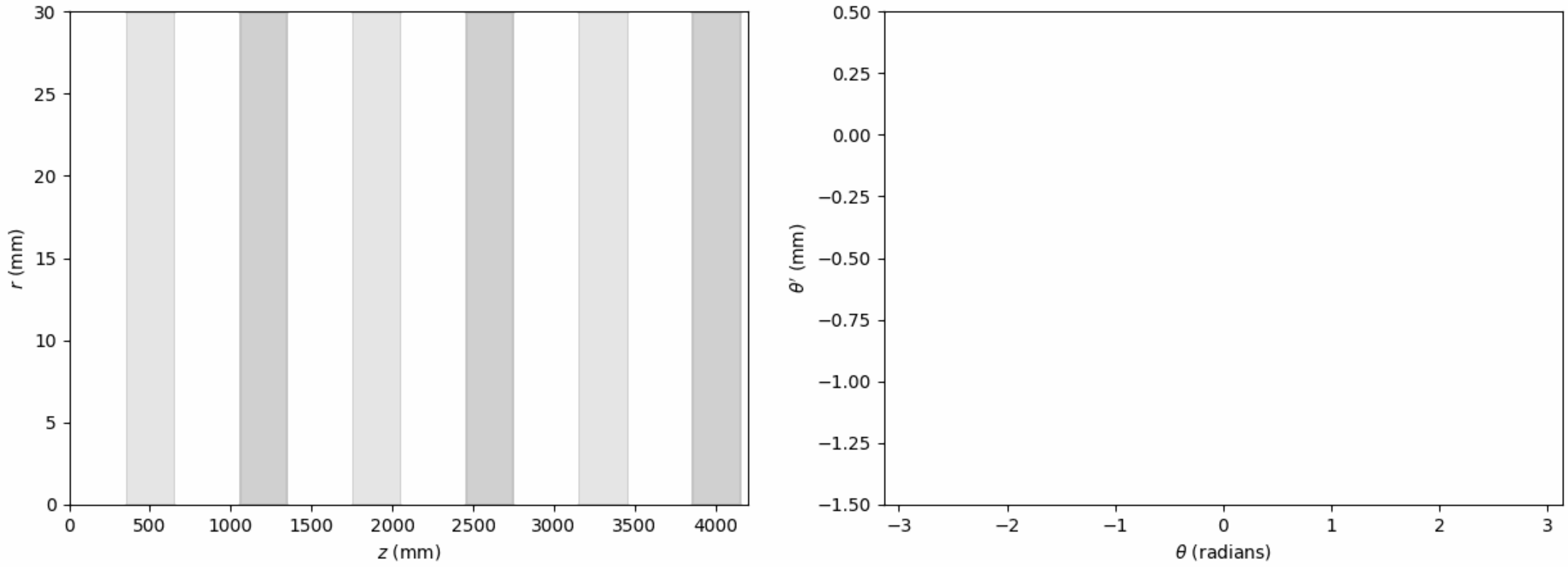
REVISITING POLAR PHASE SPACE

(Animation)



REVISITING POLAR PHASE SPACE

(Animation)



STUDYING SIMPLIFIED HFOFO WITH EMITCALC

Terminal output

```
#VirtualDetector/det
 Emitcalc version 1.0
 settings:
   input file: for009.dat
   output file: emitcalc.out
   particle type:          2
   pzmin/pzmax:   0.10000000000000001      /  0.34999999999999998
   vp model:              1
   trans/long cuts:    0.0000000000000000        0.0000000000000000
   do not remove linear dispersions
   subtract out amplitude correlation
   not periodic in time
   sigma cut:    0.0000000000000000
```

emitcalc.inp

```
HFOFO
for009.dat
emitcalc.out
2
0.100  0.350

9.75e-3  15e-3
false
false
1 3.25e8
4
```

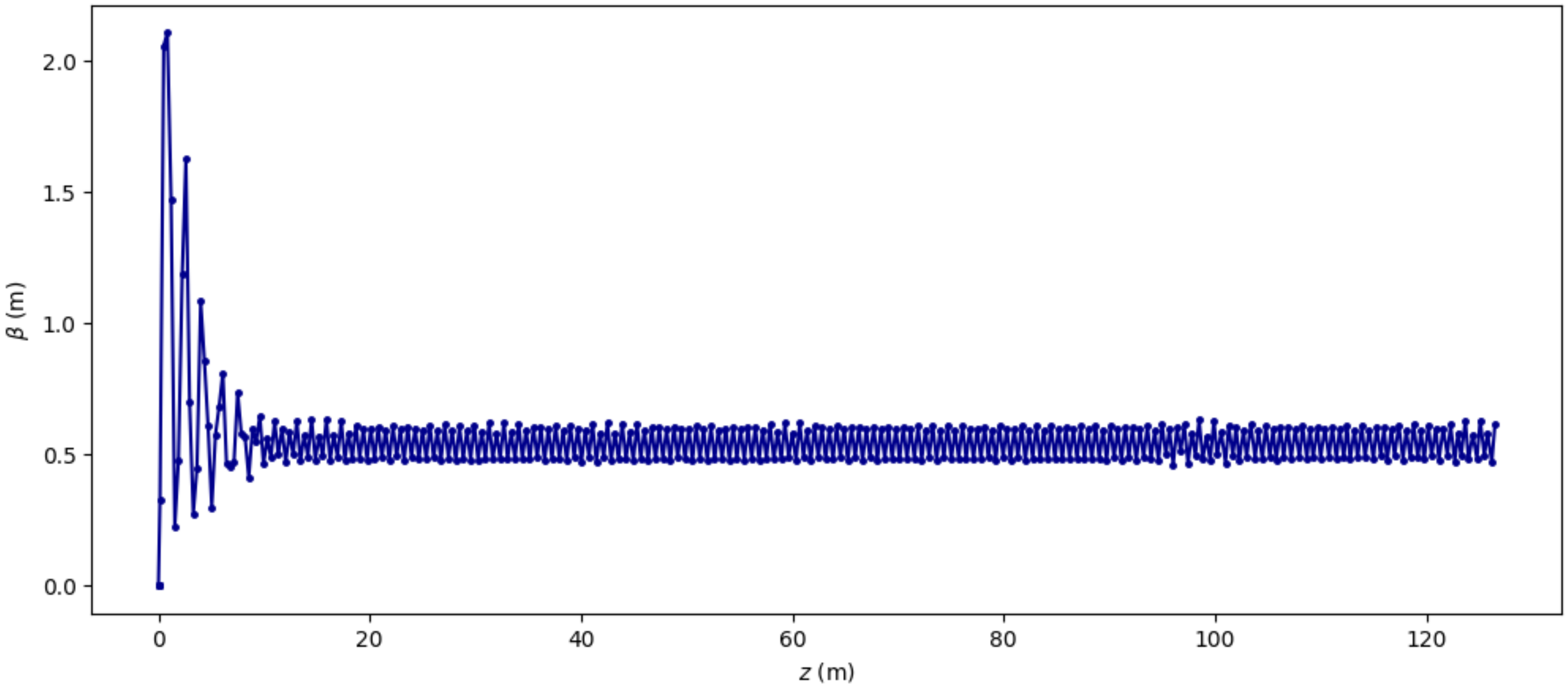
0. model 0 assumes a uniform solenoid field, which is given in the input file as the second term in line 6. For all other models, this second term is ignored.

1. model 1 is roughly similar to what ECALC9F does, but with an extra correction that can be useful for particles at large amplitudes. It uses Bz at the reference particle to determine the magnetic field on axis, but combines it with the local Bz to get a more accurate estimate of the vector potential. When there is no reference particle, it switches to model 2. Note that if the reference particle is not on axis this will lose accuracy.

2. model 2 also assumes axisymmetric fields, but calculates the vector potential by fitting the values of Bz at all particle locations to an expansion for Bz(r). So no reference particle is needed.

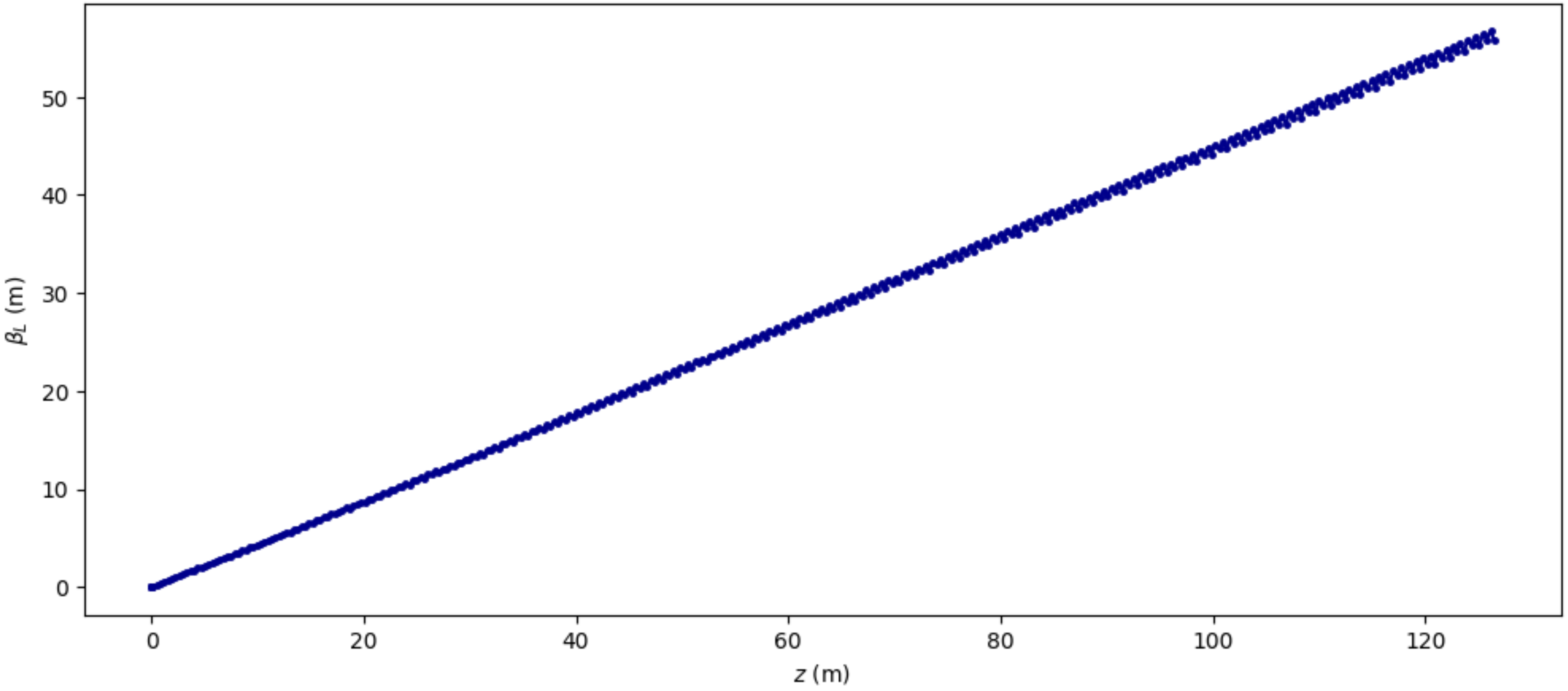
STUDYING SIMPLIFIED HFOFO WITH EMITCALC

With the same 10,000-event input file as before,



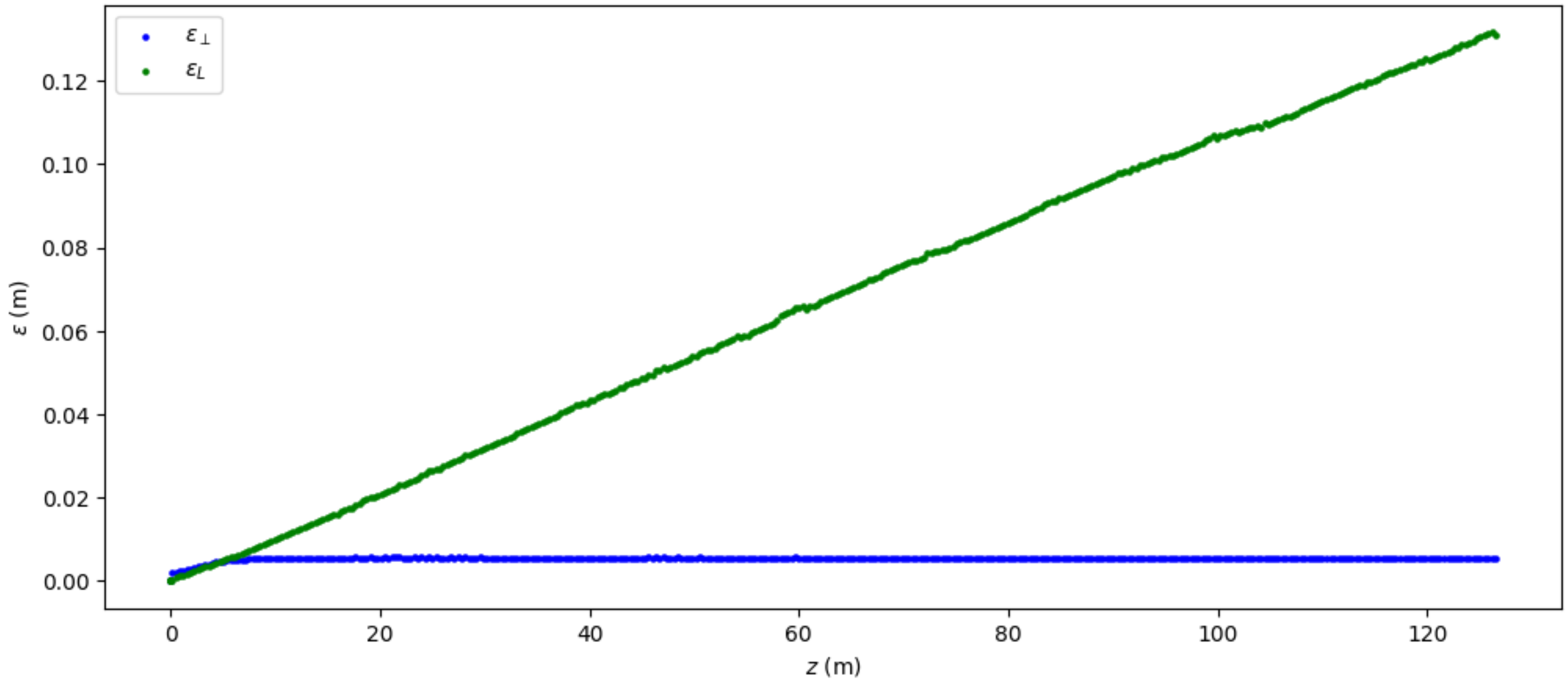
STUDYING SIMPLIFIED HFOFO WITH EMITCALC

With the same 10,000-event input file as before,



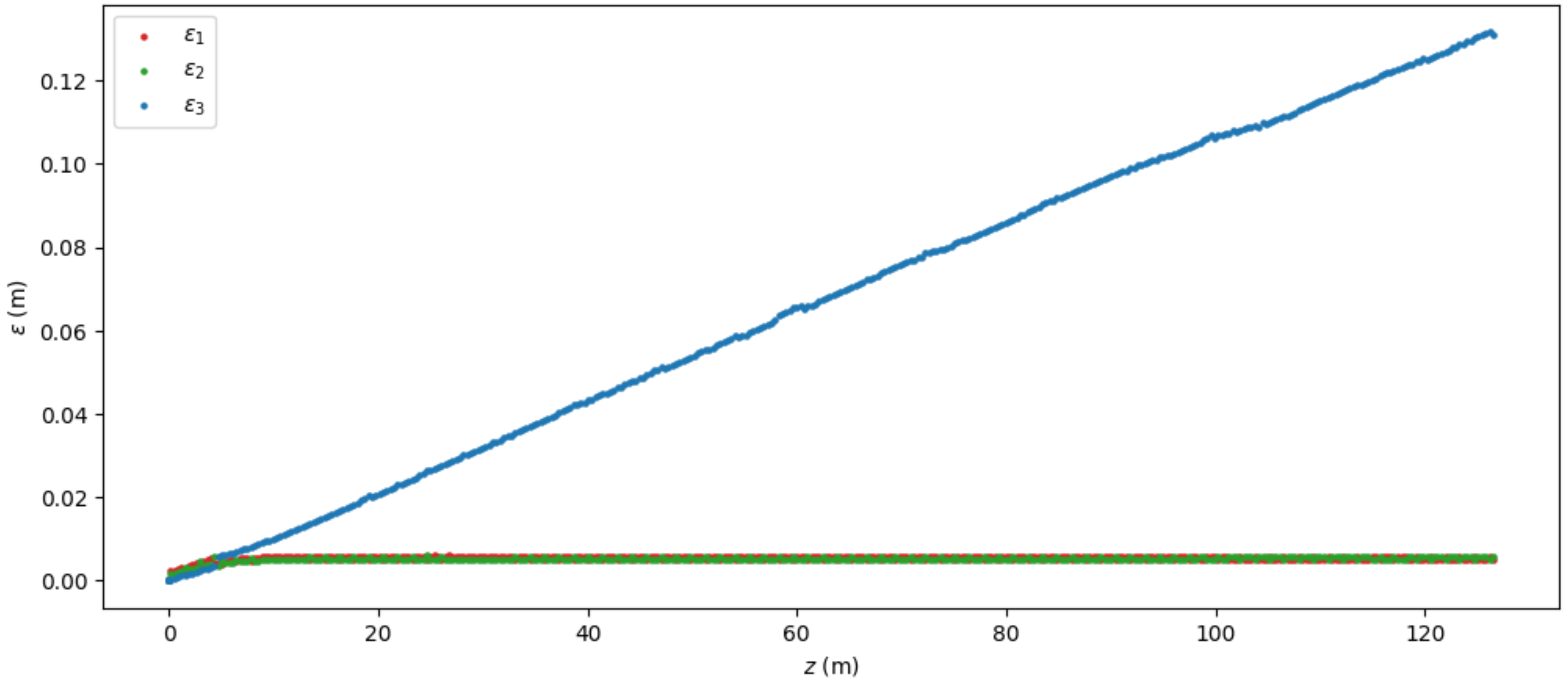
STUDYING SIMPLIFIED HFOFO WITH EMITCALC

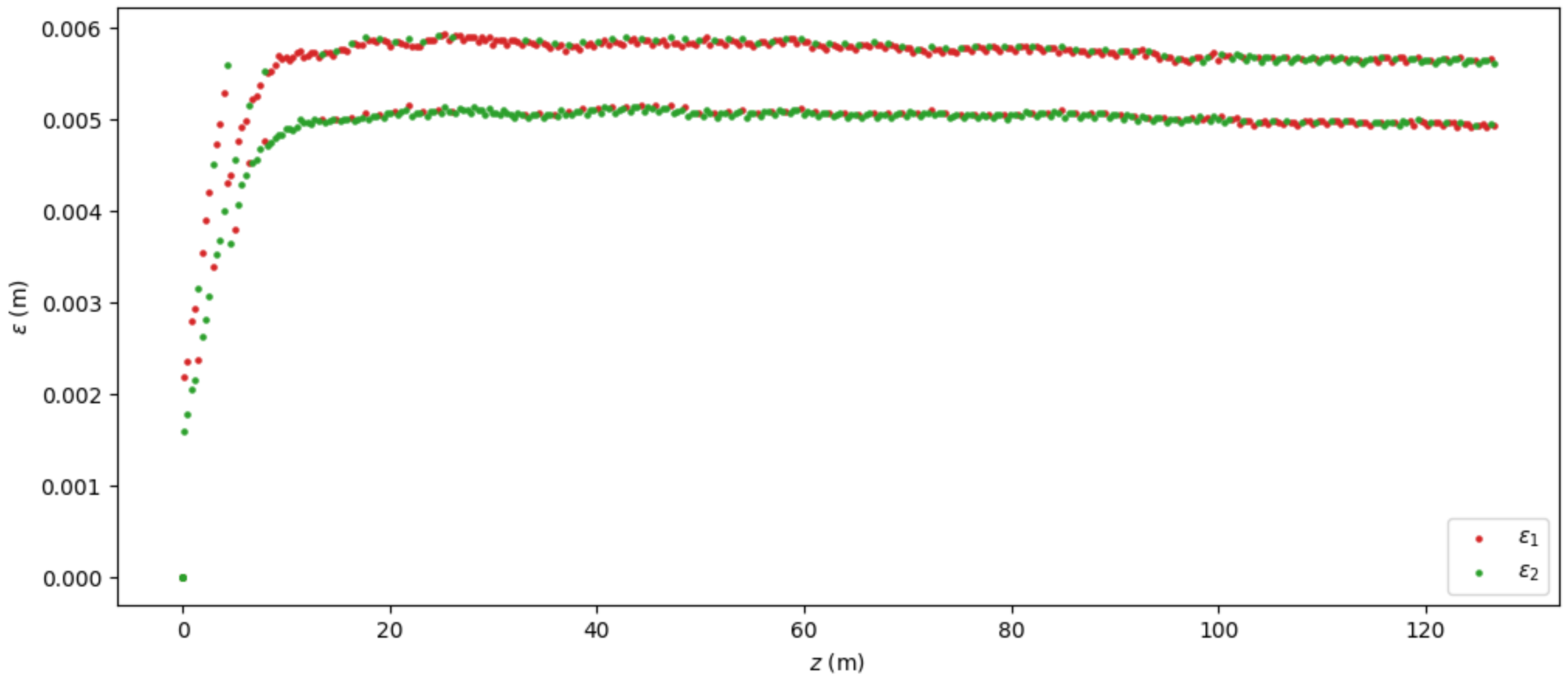
With the same 10,000-event input file as before,



STUDYING SIMPLIFIED HFOFO WITH EMITCALC

With the same 10,000-event input file as before,





NEXT STEPS

- Finish understanding and translating Yuri’s emittance calculation to python
- Study emittance in original HFOFO channel — then we have a control!

- Calculate B-field components by-hand for small tilt angles
- Then replace tilts with dipoles in simulation and try to replicate behavior