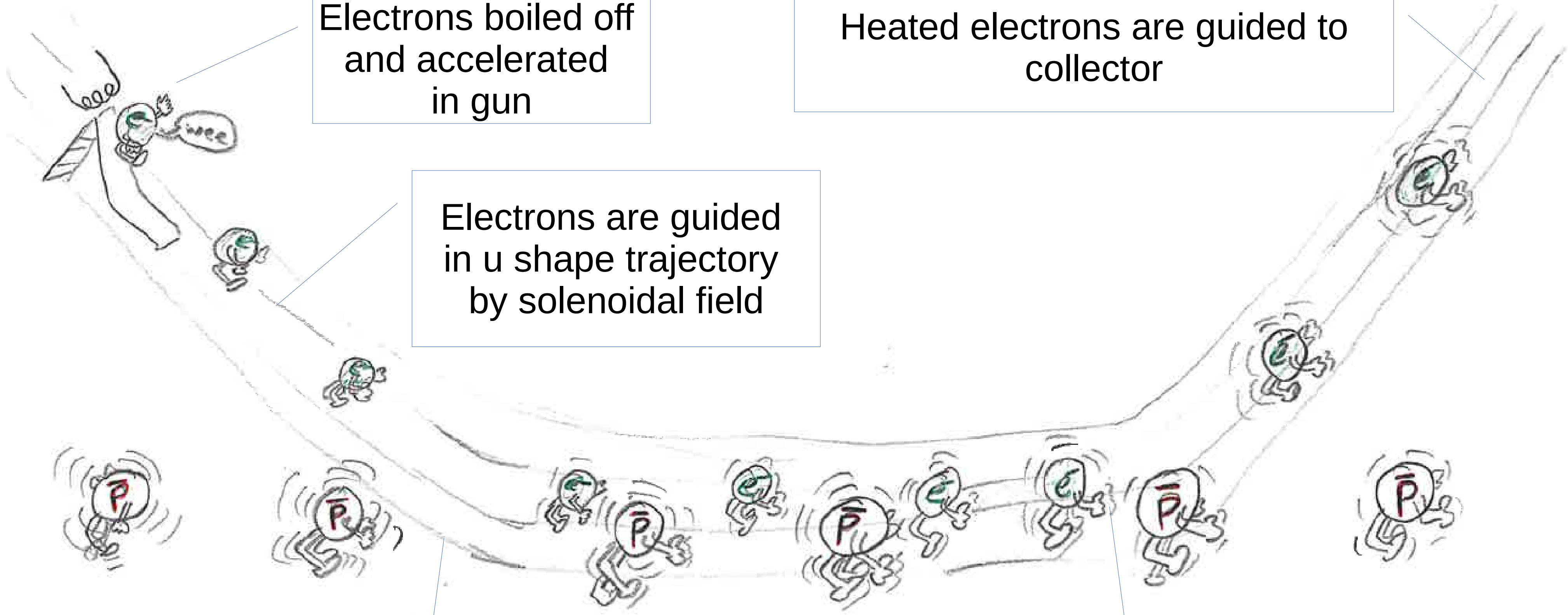
Particle tracking in Opera

24th November 2021

Luke von Freeden

CERN TE-MSC-NCM

Electron cooling



Electrons boiled off and accelerated in gun

Heated electrons are guided to collector

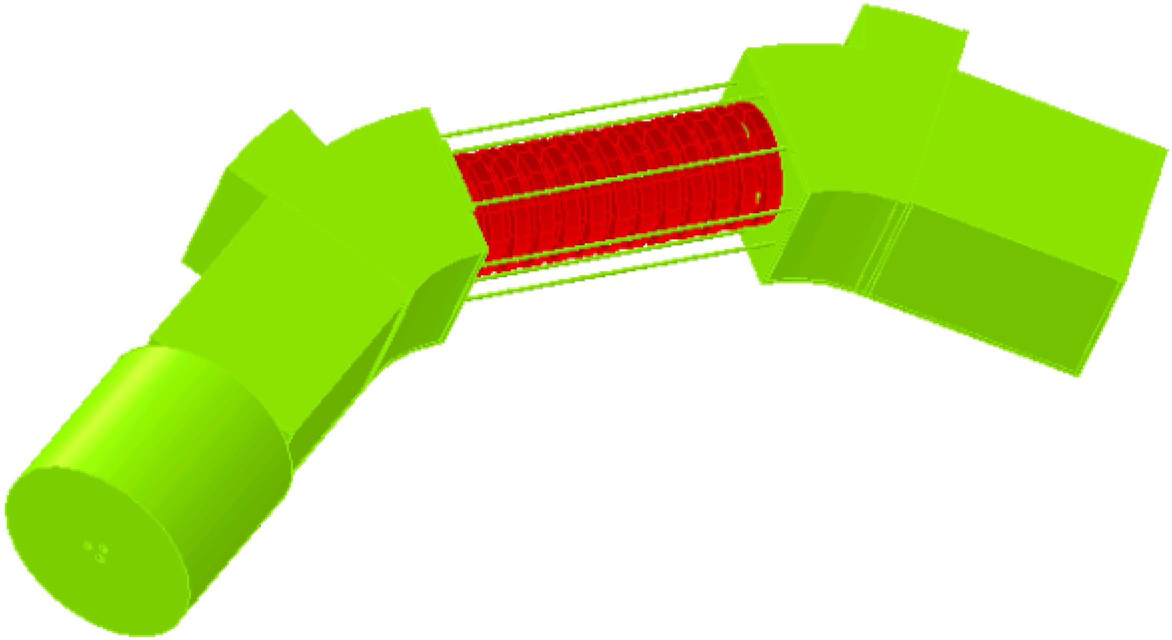
Electrons are guided in u shape trajectory by solenoidal field

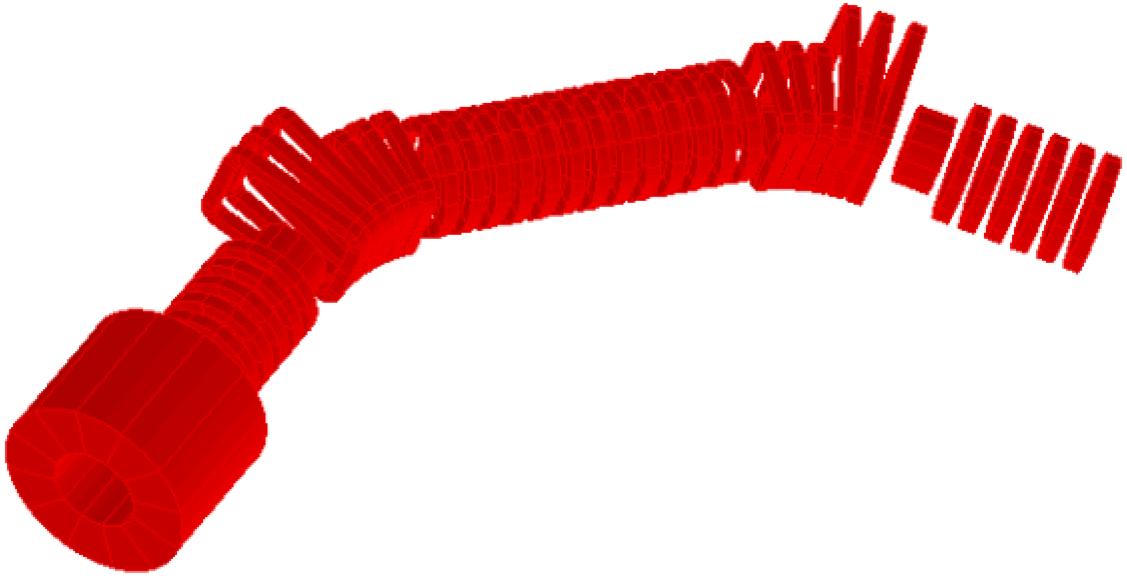
Anti-protons have higher beam rigidity so are relatively unaffected

Both particles have same charge Coulomb interactions heat transfer from anti-protons to electrons

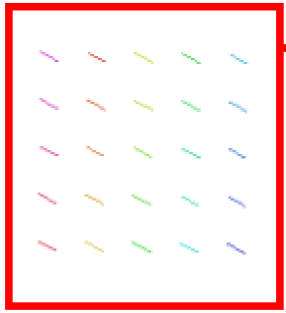
Cooled anti protons continue on there orbit of the machine

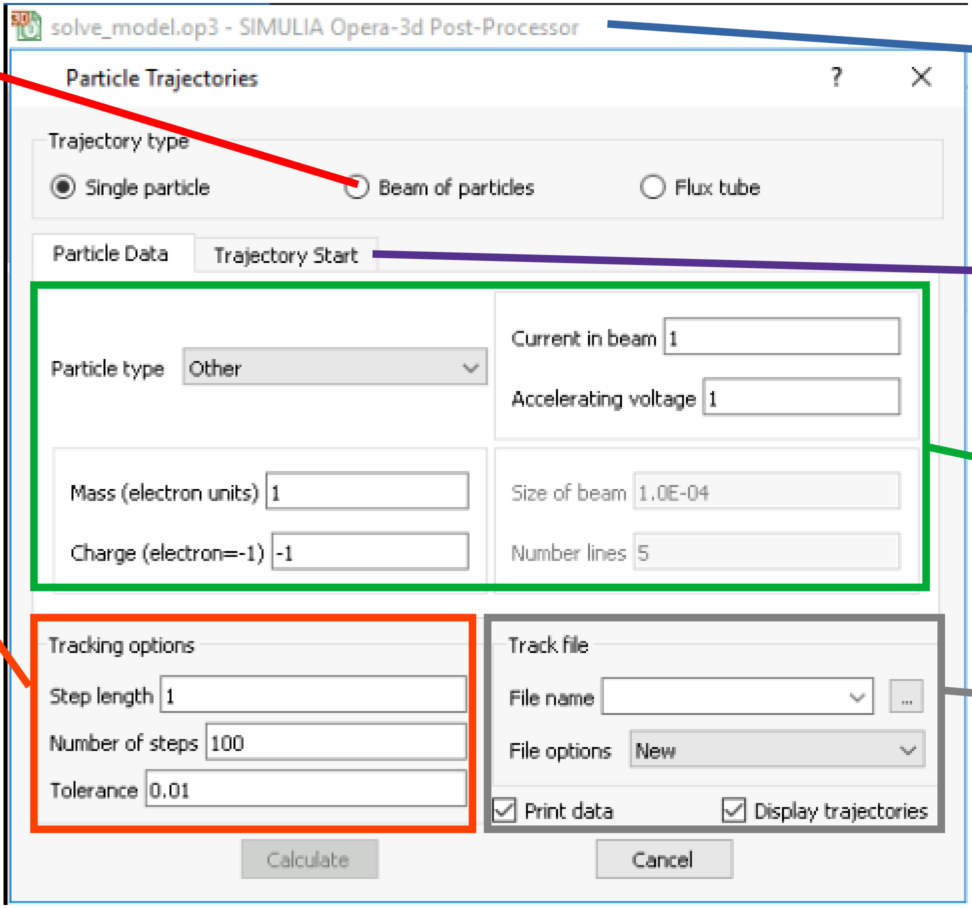
Electron cooler





Introduction to Opera feature





Tracked through a magneto-static solution

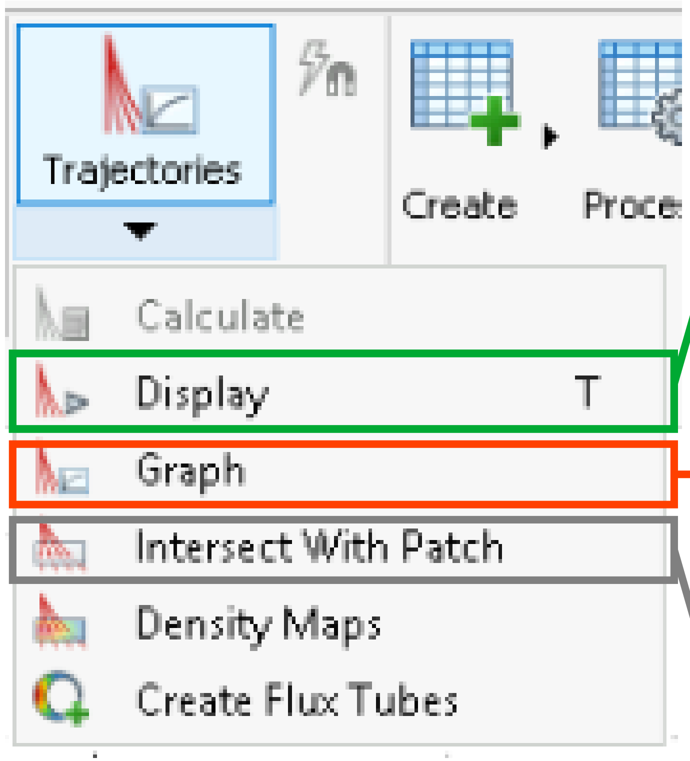
Released from arbitrary point and angle

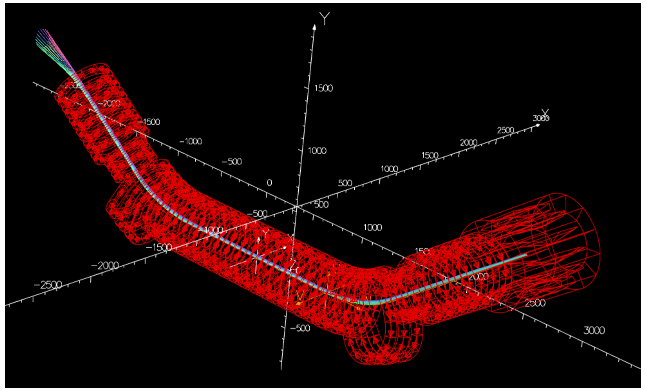
Energy, mass and charge can be defined

Solution saved to binary file
Can be reloaded later

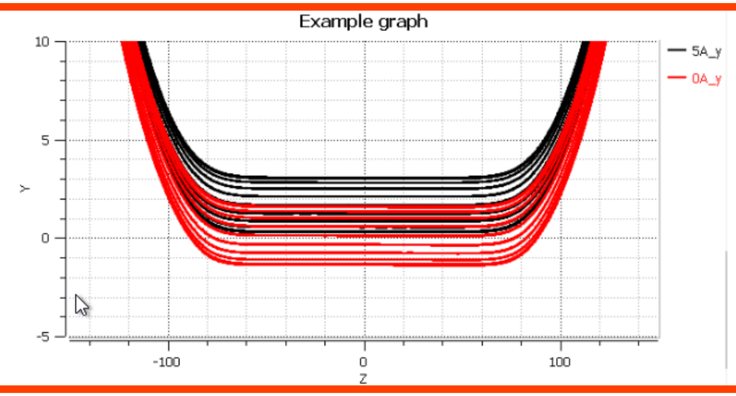
Solved in steps, computationally cheap

Post-processing and visualisation

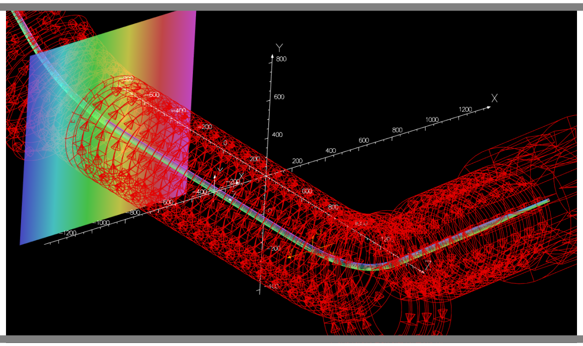




Qualitative analysis relative to nearby features. Pretty...



Good for qualitative comparison and analysis

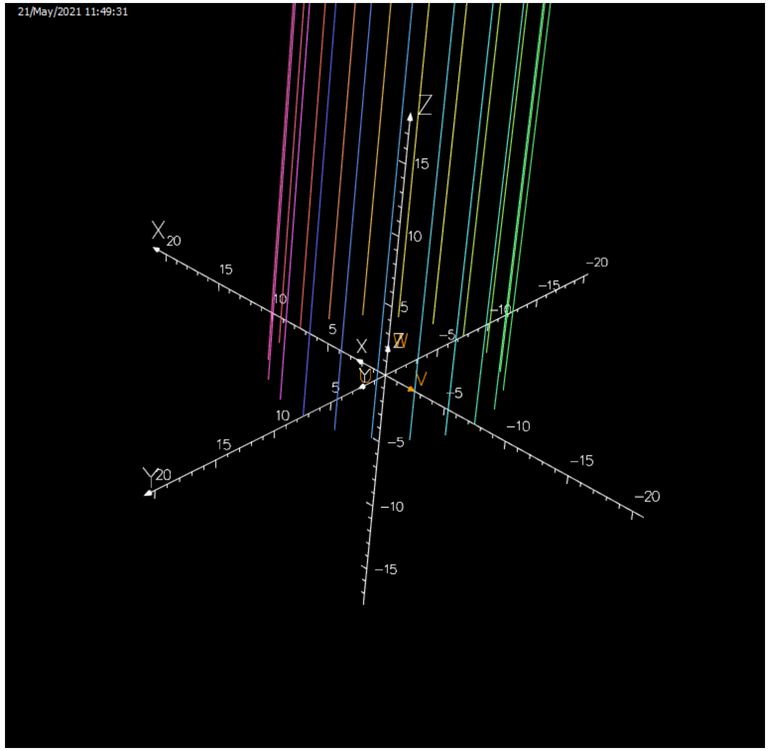


Can be exported for quantitative analysis. Strong in conjunction with other two.

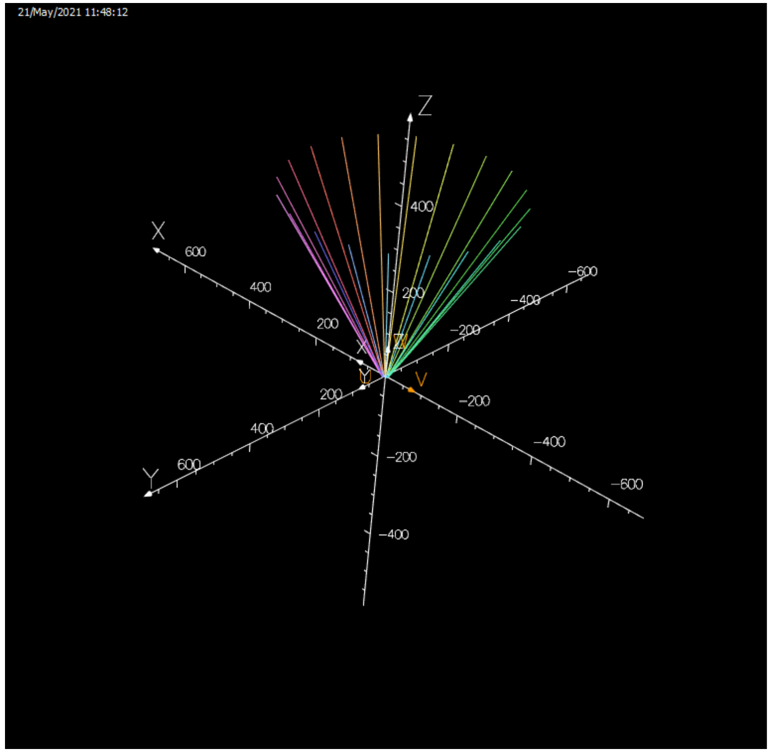
The approach

- Place a local co-ordinate system
  - Rotation angle and [x, y, z] as function of distance along reference trajectory, s
- Release a series of single particles
- Concatenate the resulting files
- Open the combined file and find intersections (re-use s $\rightarrow$ θ, x, y, z)
- Save to csv for post processing.

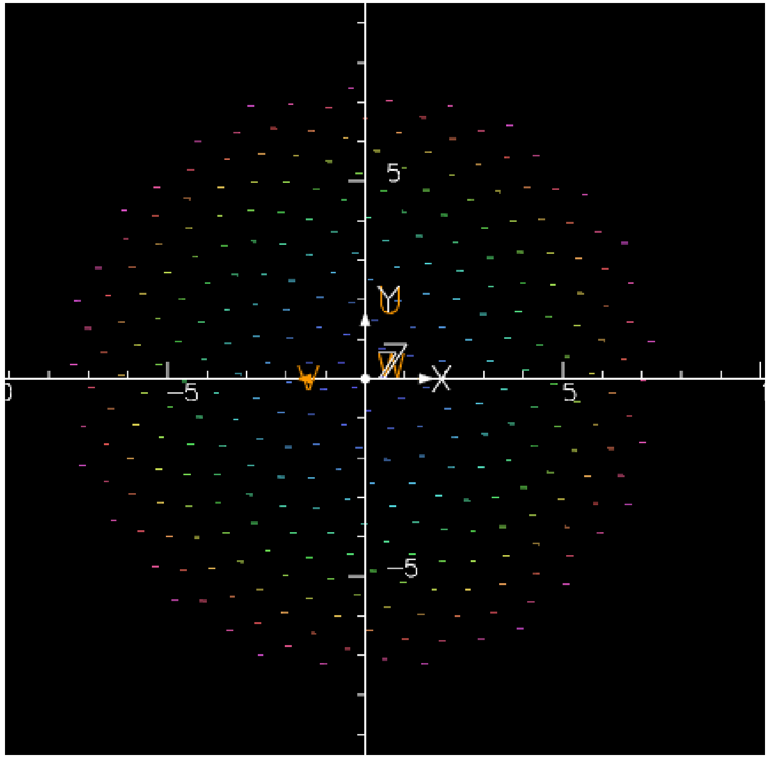
Release patterns



Ring: Probably the most useful



Divergent ring:



Sunflower spiral: circular beam with flat density function.

Others???

Implementation

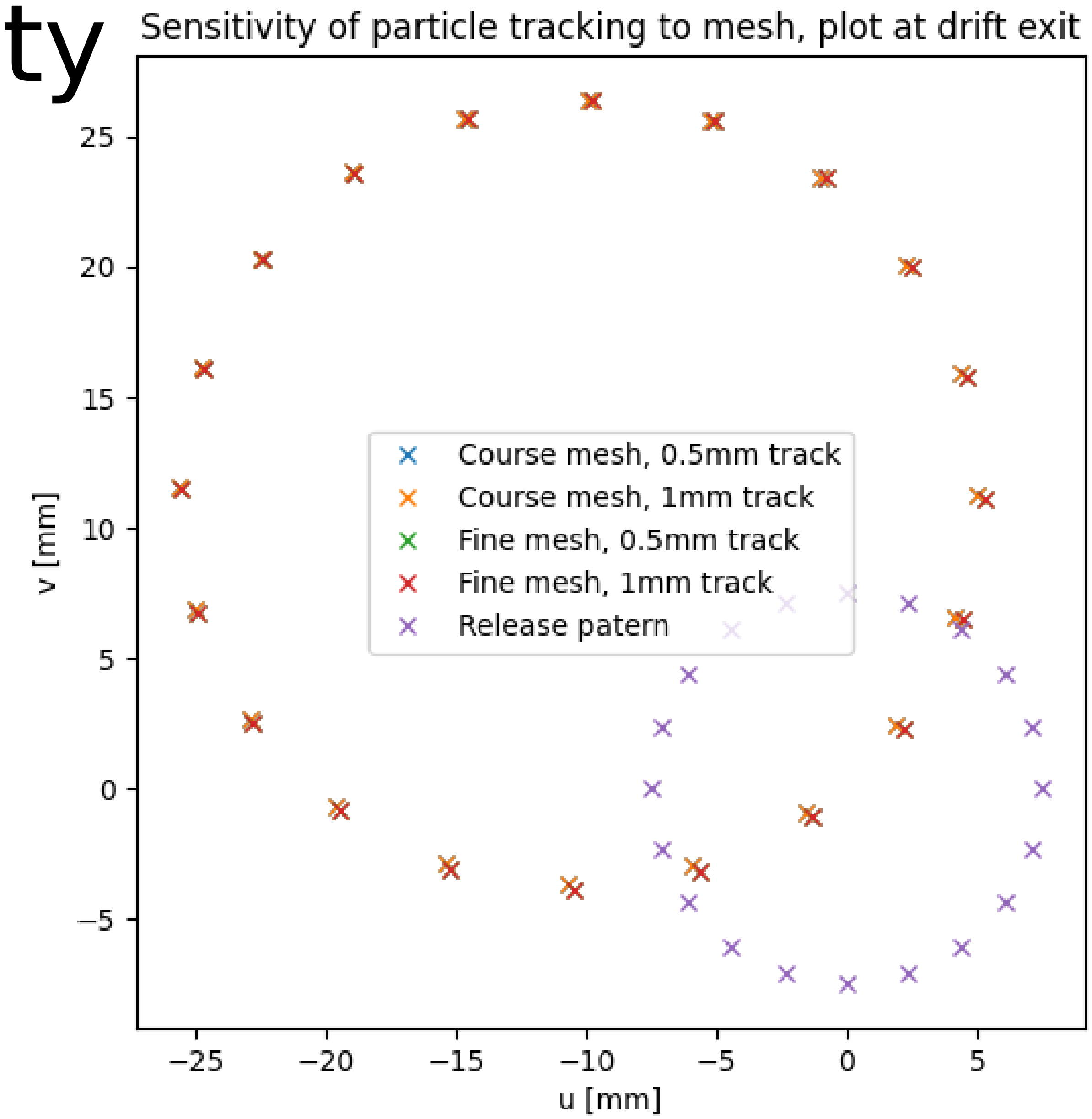
- Run python script (w/ numpy) from opera console.
- Do computation in python (simple language) and concatenate strings into opera commands.

```
angle = alpha(s)
coords = coordinates_xyz(s)
print("s="+str(s)+" angle="+str(angle)+" coords="+str(coords))
operafea.command("SET"
                 +" XLOCAL="+str(coords[0])
                 +" YLOCAL="+str(coords[1])
                 +" ZLOCAL="+str(coords[2])
                 # rotate by 90deg about z: direction is anti-clockwise
                 # looking down z i.e. from x to y. y' is now in old -x
                 +" PLOCAL="+str(90)
                 # rotate by alpha about y' / -x. *This is equivelent to
                 # rotating about global x by - alpha*
                 +" TLOCAL="+str(np.rad2deg(angle)) # rotate by -alpha(s) about x
                 +" SLOCAL="+str(0))
```

```
def alpha(s):
    ls = length_of_drift_solenoid/2
    if np.abs(s) < ls:
        alpha = 0
    elif ls <= np.abs(s) and np.abs(s) < ls + radius_of_transition*angle_of_transition:
        alpha = (np.abs(s) - ls)/radius_of_transition
    elif ls + radius_of_transition*angle_of_transition < np.abs(s):
        alpha = angle_of_transition
    return alpha*np.sign(s)
```

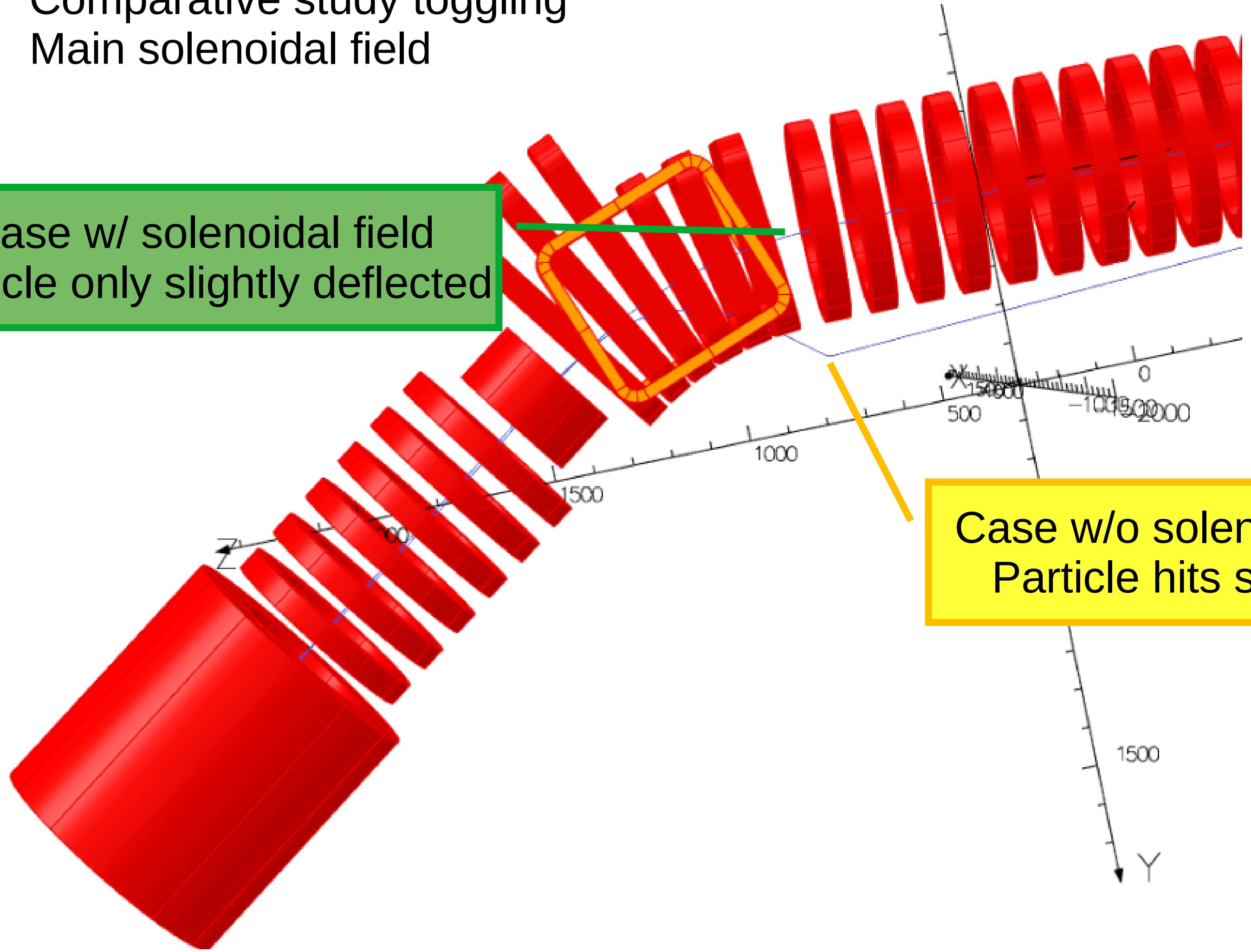
Numerical sensitivity

- Released at center of drift
  - 15mm Ø ring
  - Divergence = 0°
  - || to s
  - 70keV
- No difference between 0.5mm and 1mm discretisation of track
- 0.30 mm largest delta between meshes



Effect of solenoidal field on dipole bending

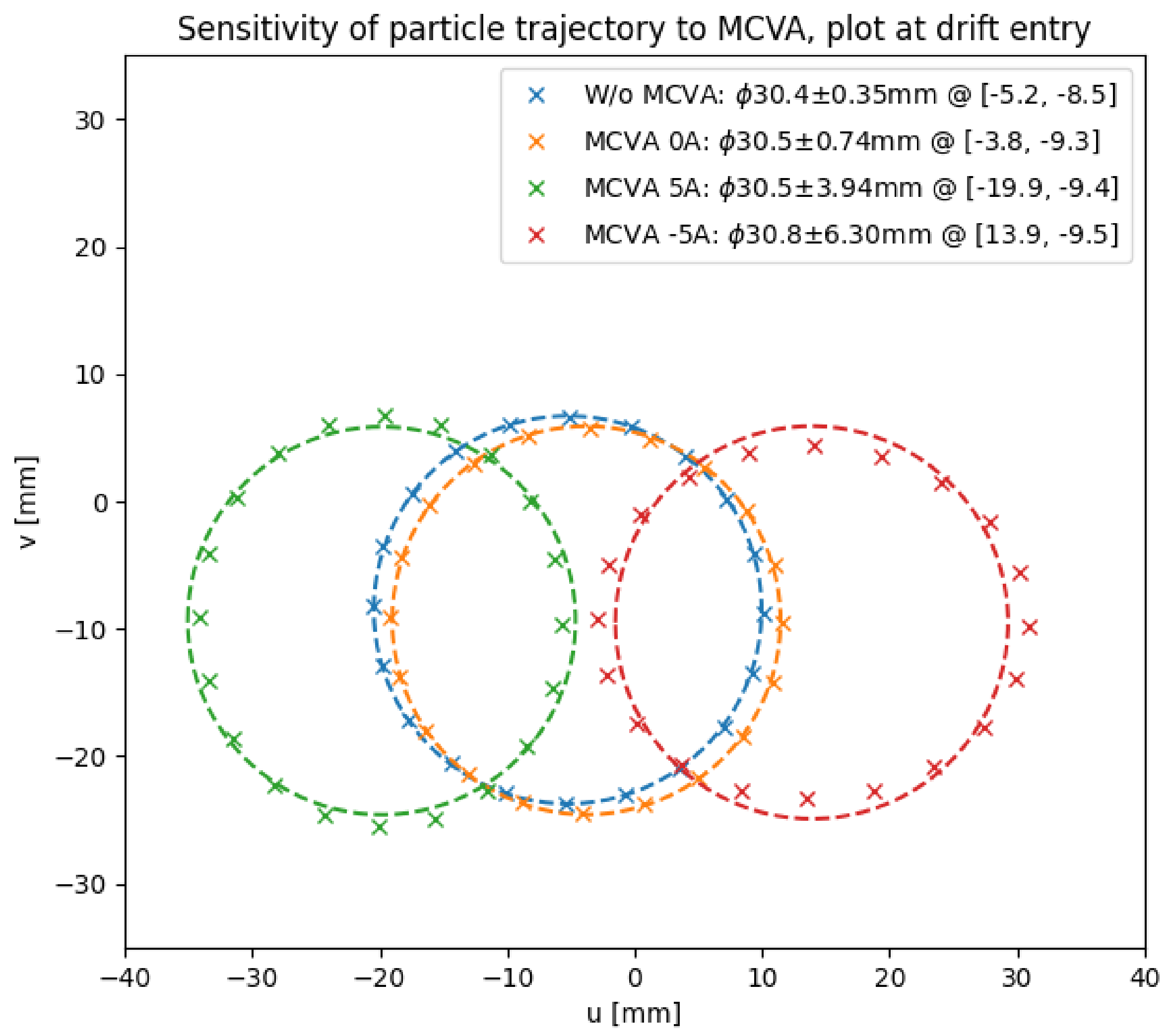
Comparative study toggling
Main solenoidal field



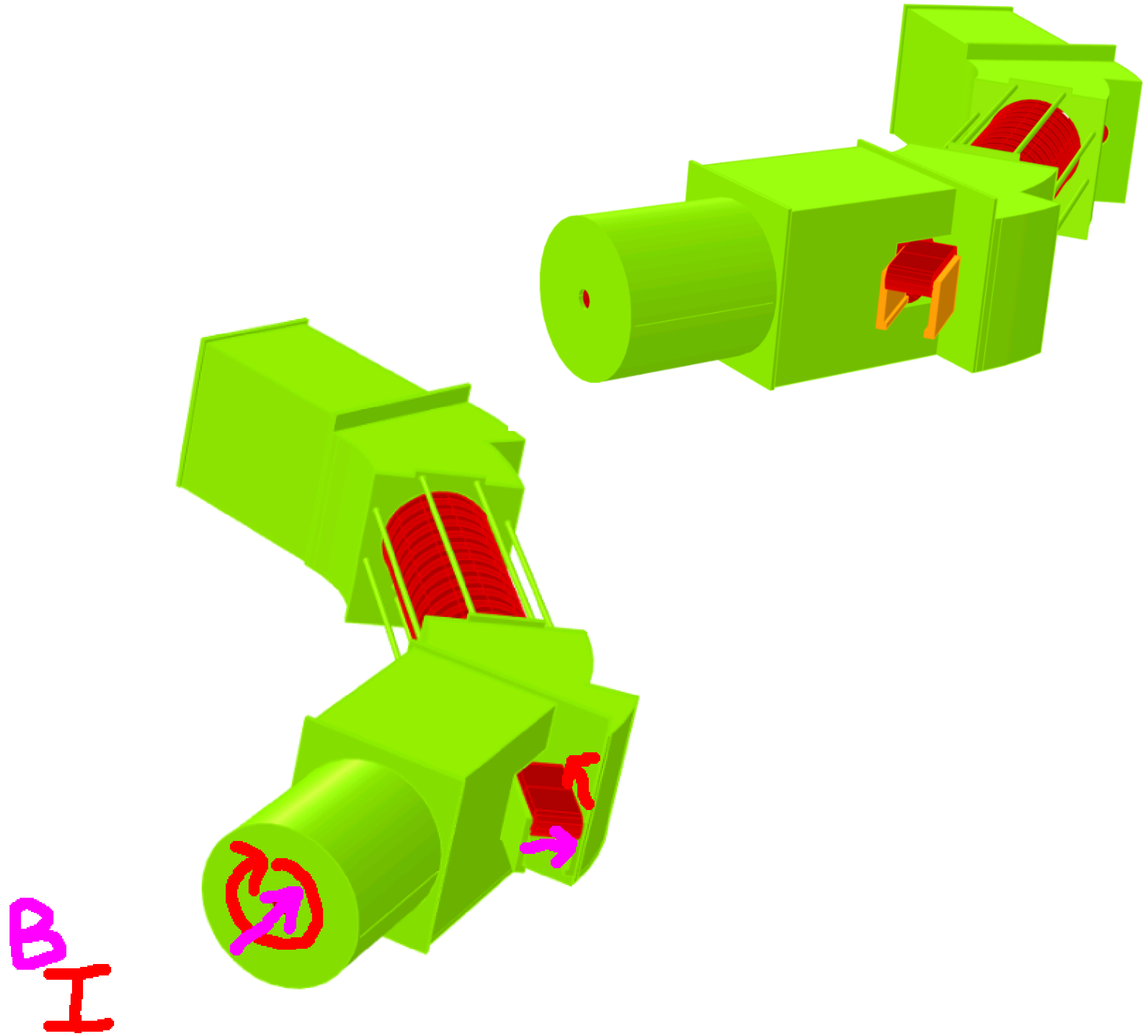
Case w/ solenoidal field
Particle only slightly deflected

Case w/o solenoidal field
Particle hits shielding

Example



Sensitivity of particle trajectory to MCVA, plot at drift entry
W/o MCVA: ϕ30.4±0.35mm @ [-5.2, -8.5]
MCVA 0A: ϕ30.5±0.74mm @ [-3.8, -9.3]
MCVA 5A: ϕ30.5±3.94mm @ [-19.9, -9.4]
MCVA -5A: ϕ30.8±6.30mm @ [13.9, -9.5]
v [mm]
u [mm]
30
20
10
0
−10
−20
−30
−40
−30
−20
−10
0
10
20
30
40



B
I

Conclusion

- Introduced the Opera particle tracking tools and outlined the extension to arbitrary beam patterns
- The numerical sensitivity of the particle tracking is driven by magneto-static result.
- In the presence of solenoidal field, harmonic decomposition is less useful.
- An example of the ring beam giving insight to the AD E-cooler design.