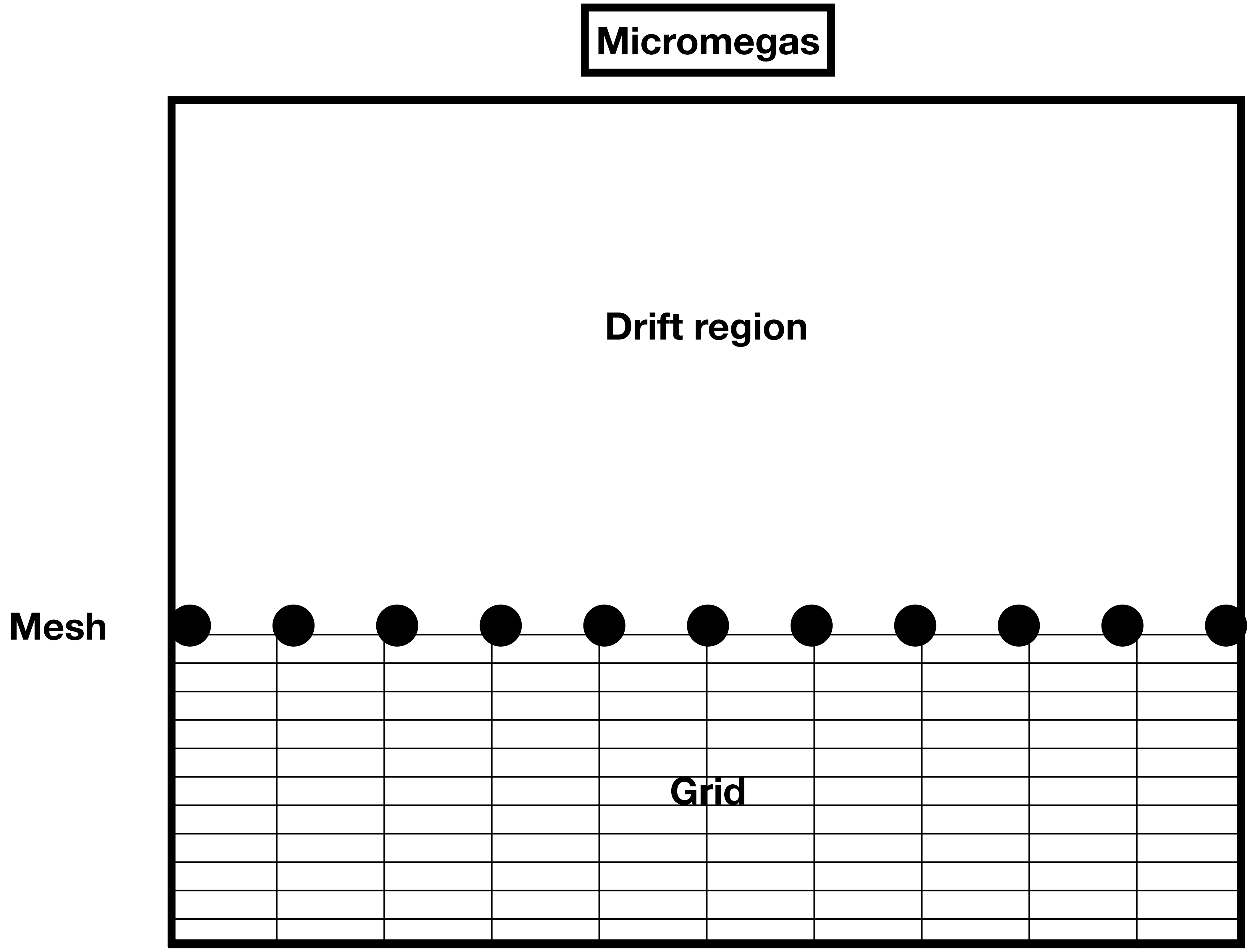
Space charge effects in simulations of large avalanche dynamics

Garfield++

Tom Szwarcer | University of Oxford | DRD1

The aim

- Use space charge effect in microscopic/MC avalanche simulations
- Allow the user to define regions where the effect should be included
- Include both electrons and ions
- No mirror charges in a general geometry - so use free space field as a first approximation

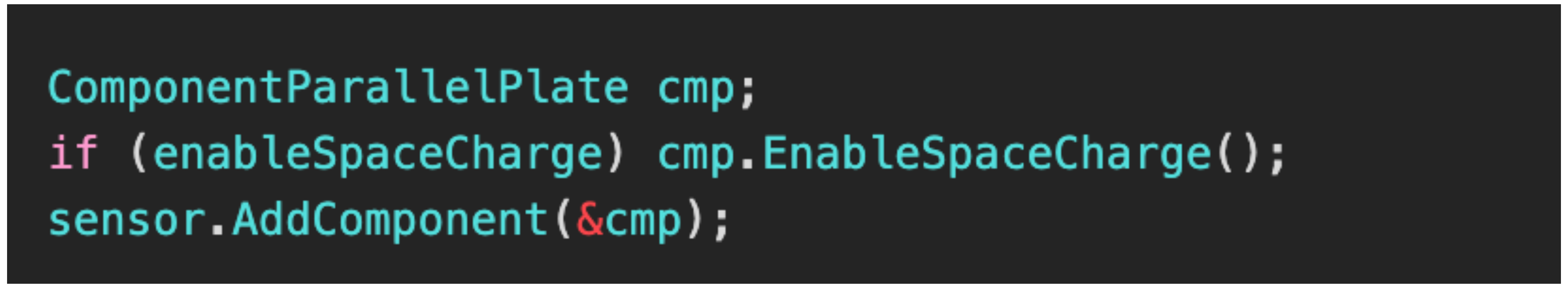


The grid

- Snap particles to a 2D (z, r) grid
- Compute the field due to axisymmetric rings of charge at a given (z, r)
- This prevents divergences that would be present in point charge fields.

Making use of the RPC code

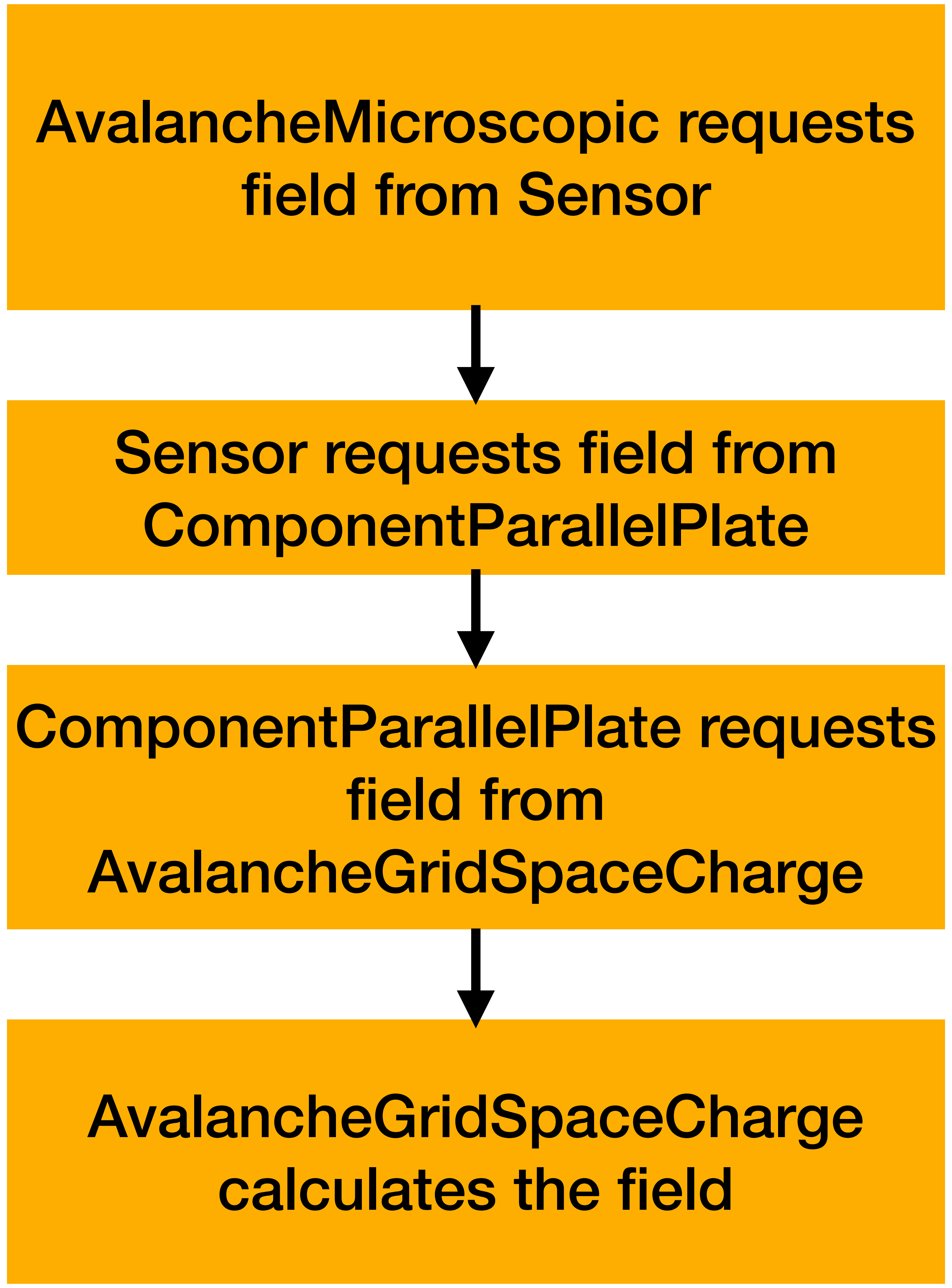
- We set up a ‘fake’ parallel plate system that exists at grid limits
- We set dV=0 so the only field present is the space charge field
- Tell Sensor that ComponentParallelPlate has its own field
- Sensor automatically adds the contribution to the field from all components
- This lets us reuse existing code for the space charge effect in RPCs



```
ComponentParallelPlate cmp;
if (enableSpaceCharge) cmp.EnableSpaceCharge();
sensor.AddComponent(&cmp);
```

Communicating the field

- In order to get the field into sensor we have to go through ComponentParallelPlate
- This allows a lot of code to be reused



AvalancheMicroscopic requests field from Sensor

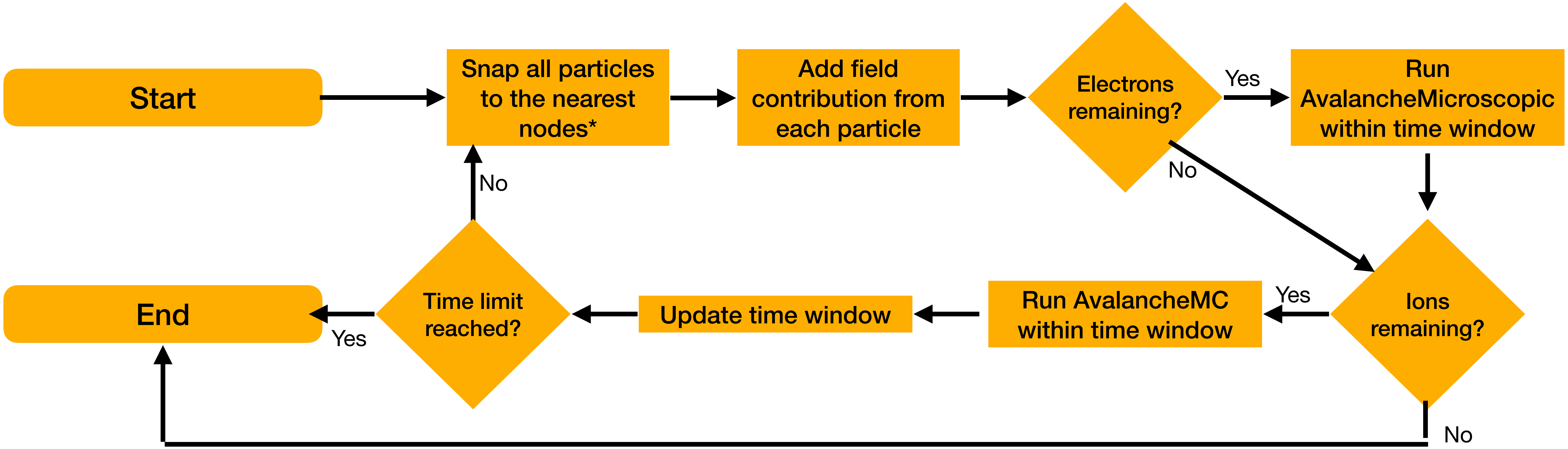
Sensor requests field from ComponentParallelPlate

ComponentParallelPlate requests field from AvalancheGridSpaceCharge

AvalancheGridSpaceCharge calculates the field

Electron & ion simulation

- As far as I know, it is not possible to simultaneously run AvalancheMC and AvalancheMicroscopic
- I do a leapfrog approach where ions and electrons are simulated alternately.


Start
Snap all particles to the nearest nodes*
Add field contribution from each particle
Electrons remaining?
Yes
Run AvalancheMicroscopic within time window
No
Ions remaining?
Yes
Run AvalancheMC within time window
Update time window
Time limit reached?
No
Yes
End
No

* the particles do not move in physical space - only on the grid, in order to calculate the field

Updating the field during simulation

- New charged particles may be produced during ionisations, penning transfer, or attachment
- These occur in real time during the simulation
- However the field is only updated once per timestep…
- How do we update the field mid-timestep?

User handles

- Functions that Garfield++ calls every time an ‘event’ happens.
- In practice the total charge at a given location doesn’t change in these events
- As new particles are produced at the same location, they will be snapped to the same grid point
- Therefore the field won’t change, and doesn’t need to be updated

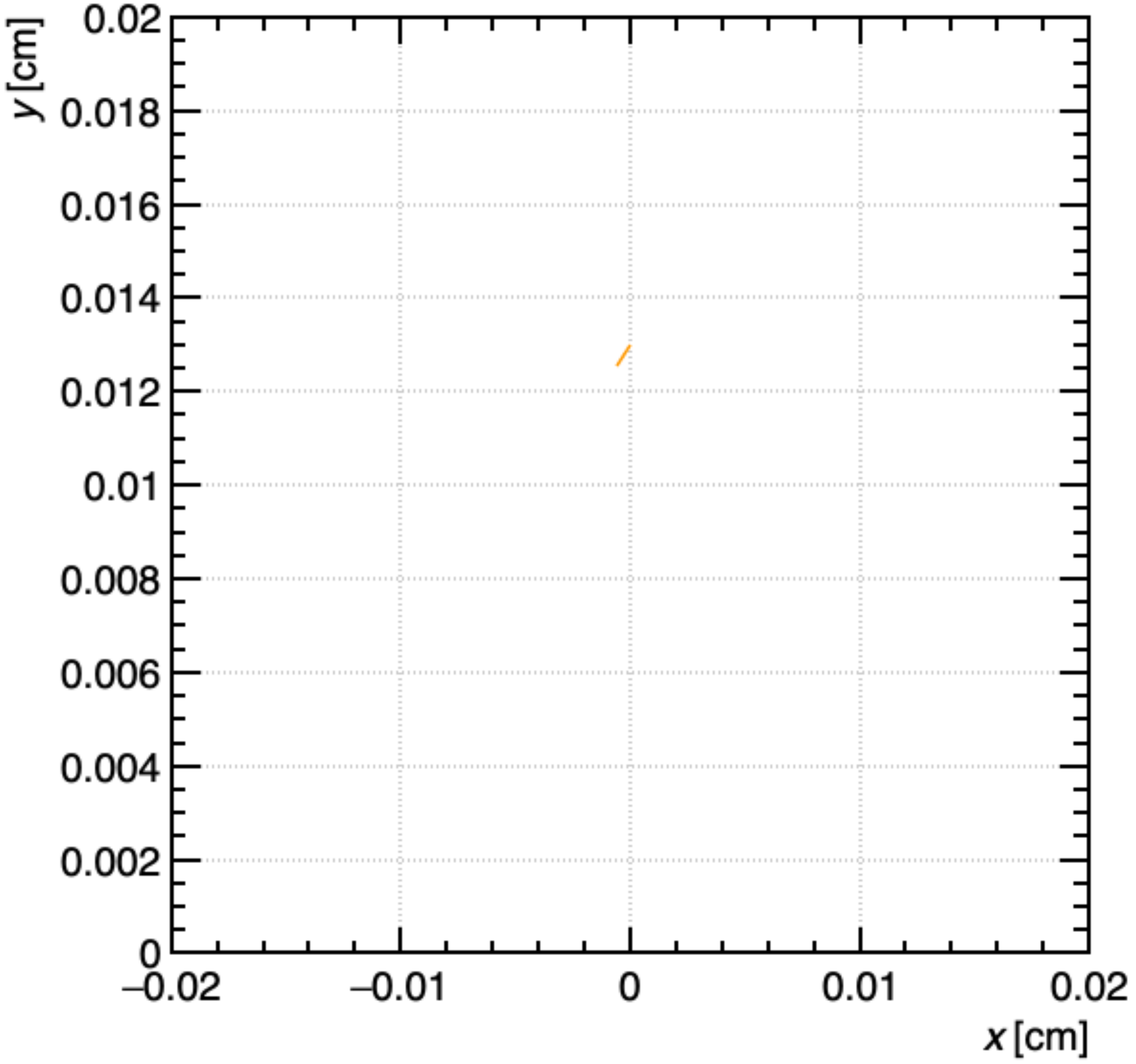
- However user handles provide a convenient way to pass on any new ions created to AvalancheMC
- New electrons are already accounted for in AvalancheMicroscopic

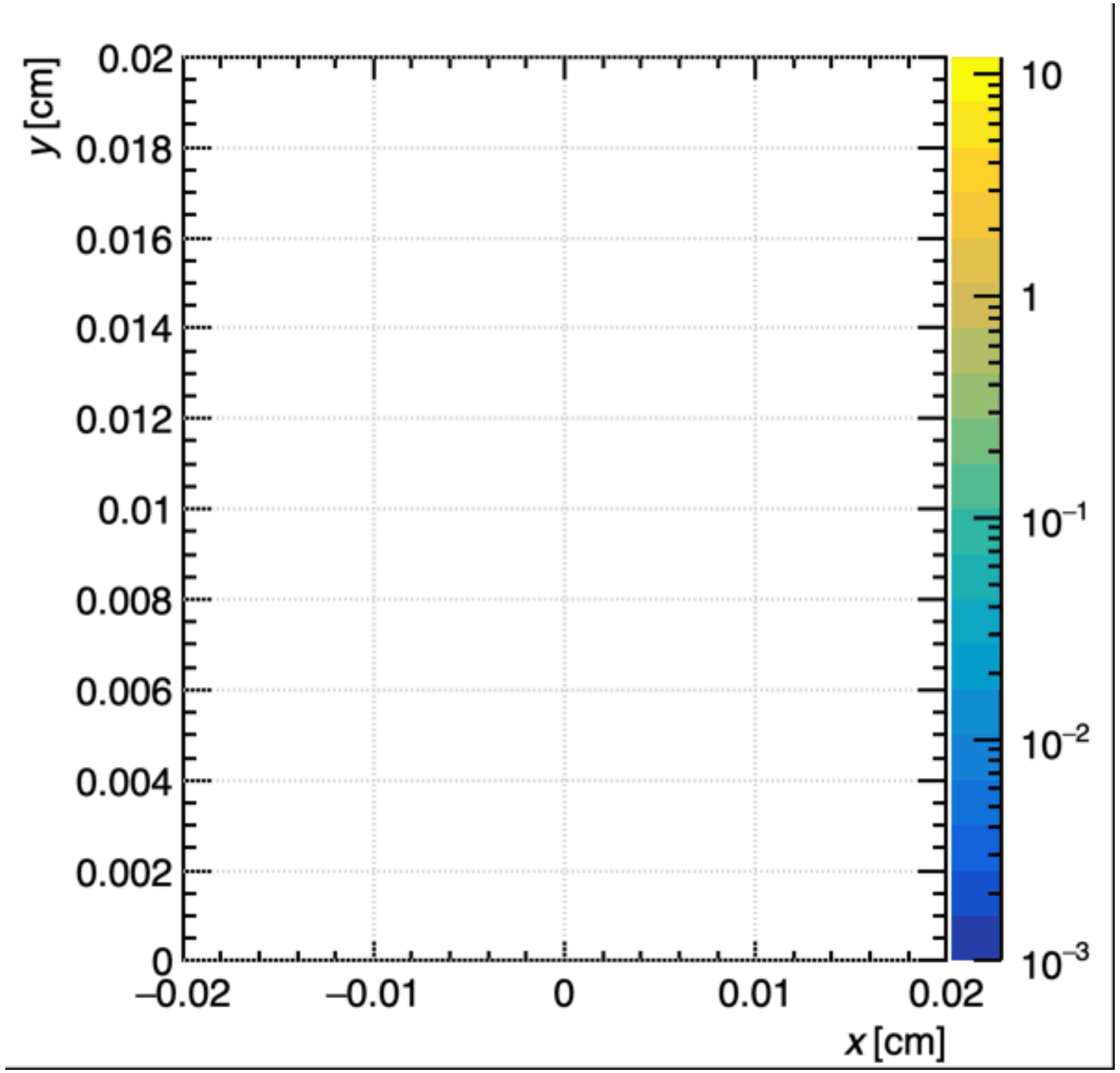
Dynamic mean position

- Avalanches will not always be centred on zero
- We need a way to centre the field on the mean position of the particles in the avalanche
- If there are particles in the avalanche, find the mean position and set as the zero of the radial coordinate

Visualising the field

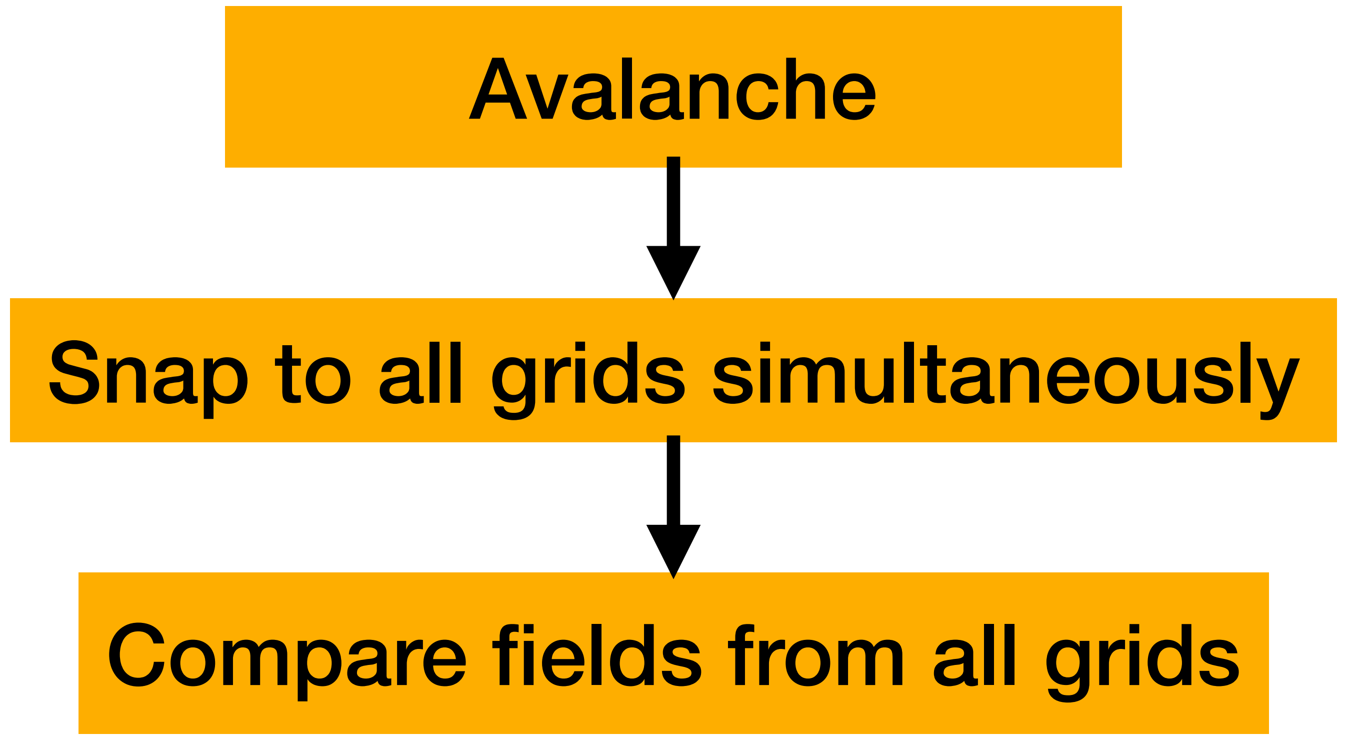
GIFs viewable at https://github.com/tomszwarcer/spacecharge-sim





Investigating optimal grid spacing

- Simulate an avalanche with multiple grids simultaneously
- Decide at what point it has converged ‘enough’ to justify a shorter computation time



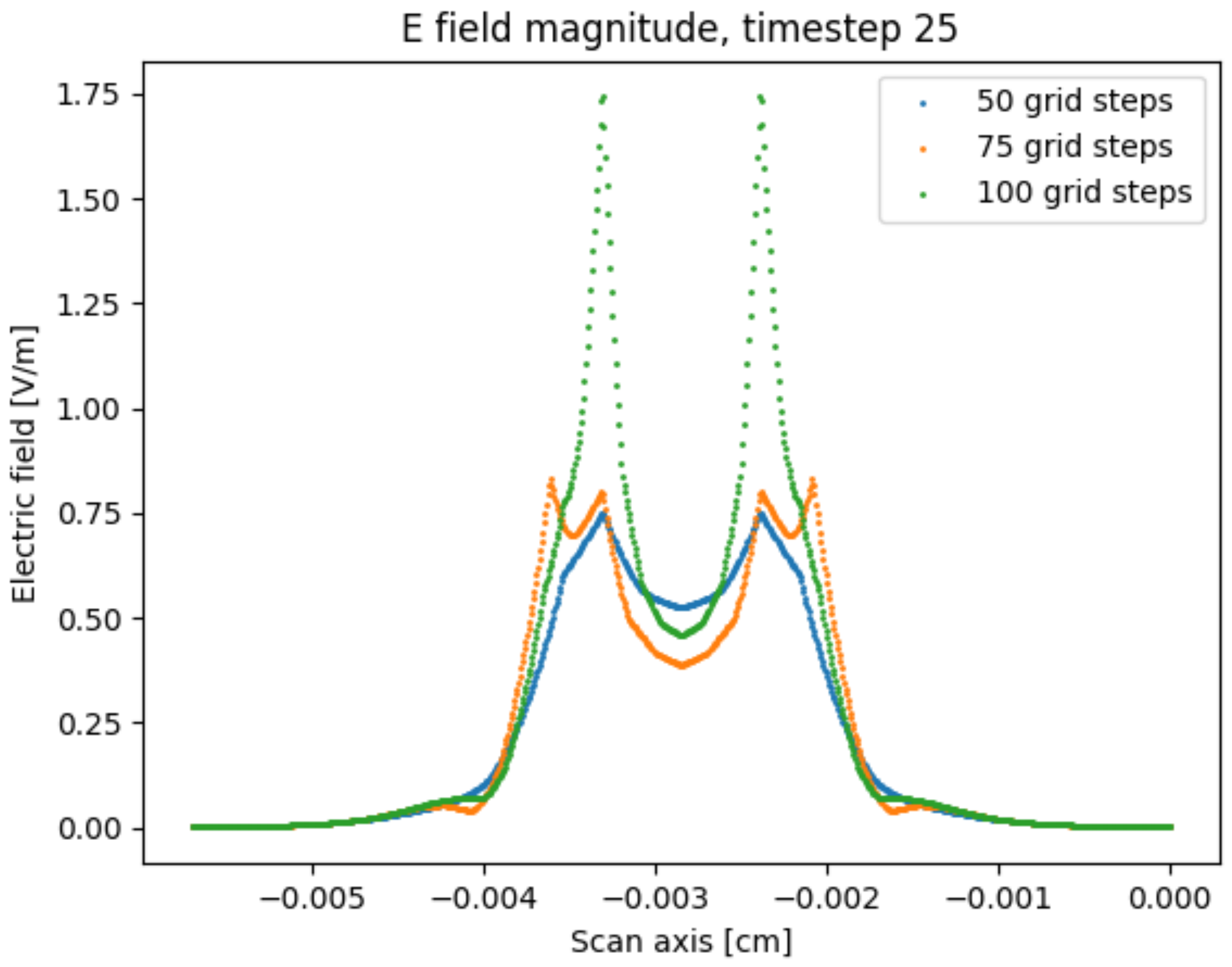
Avalanche

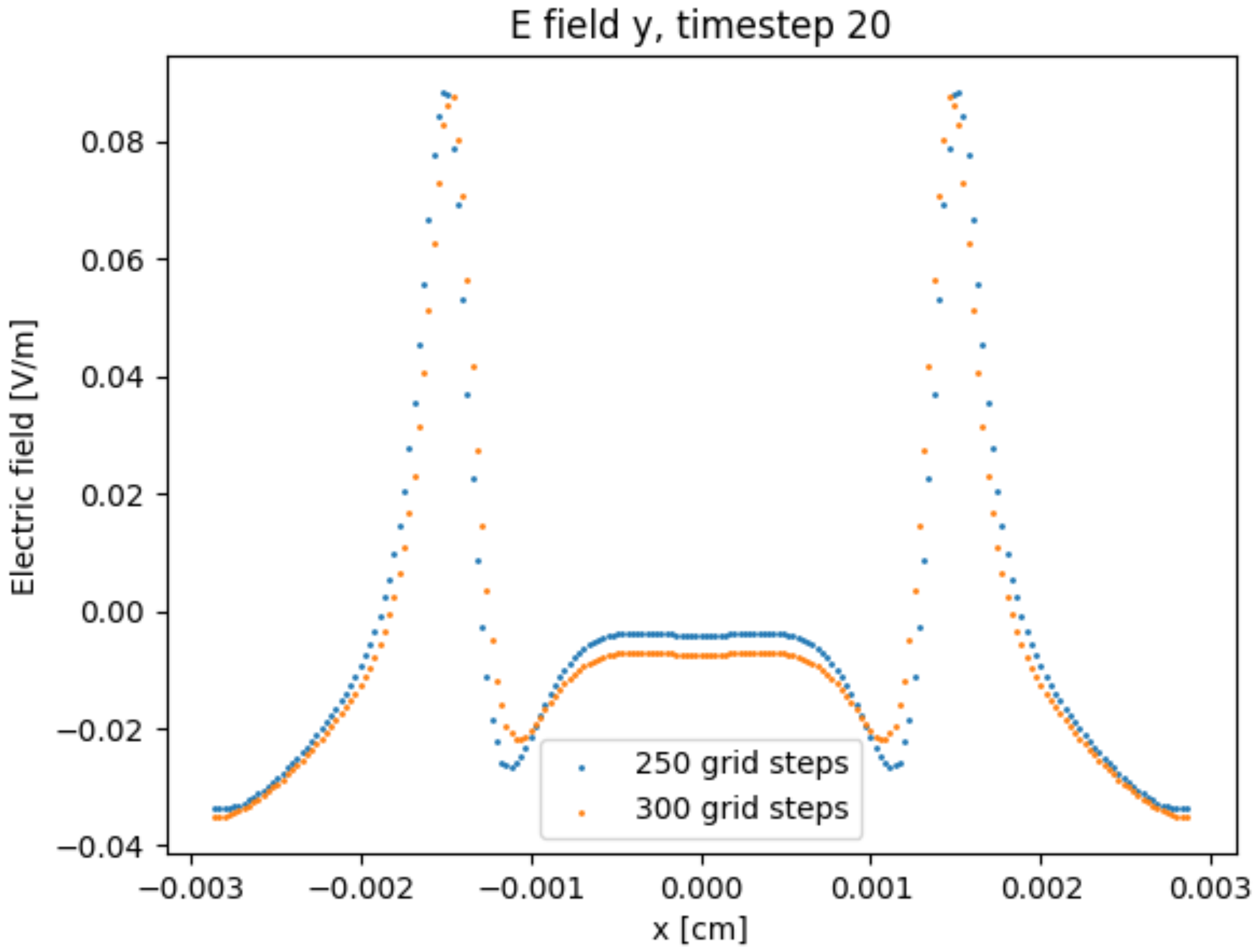
Snap to all grids simultaneously

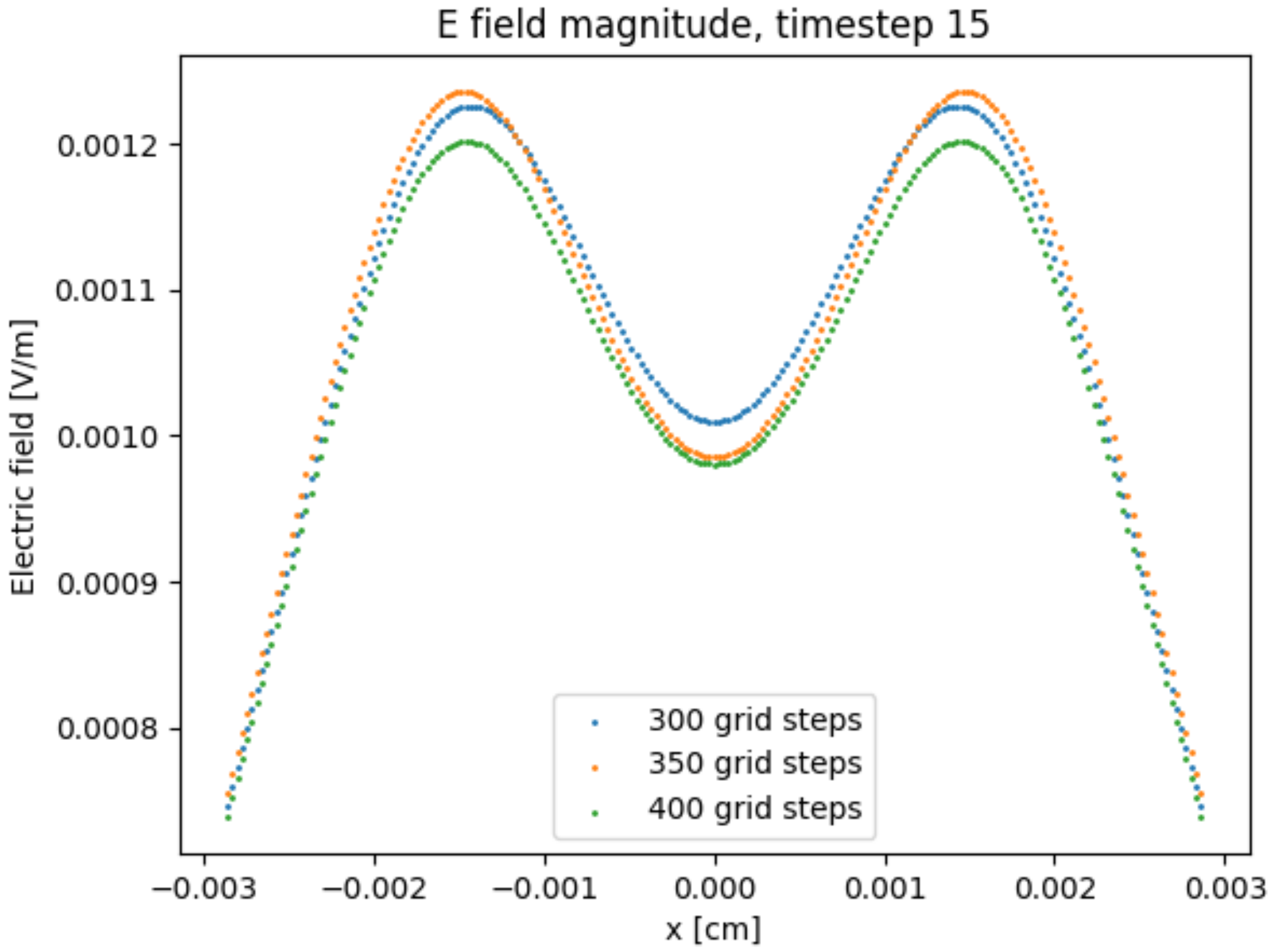
Compare fields from all grids

Investigating optimal grid spacing

- At low grid spacings, the field is a strong function of the grid spacing
- This is because electrons are being snapped a non-negligible proportion of the region of interest
- Above about 300, they start to converge





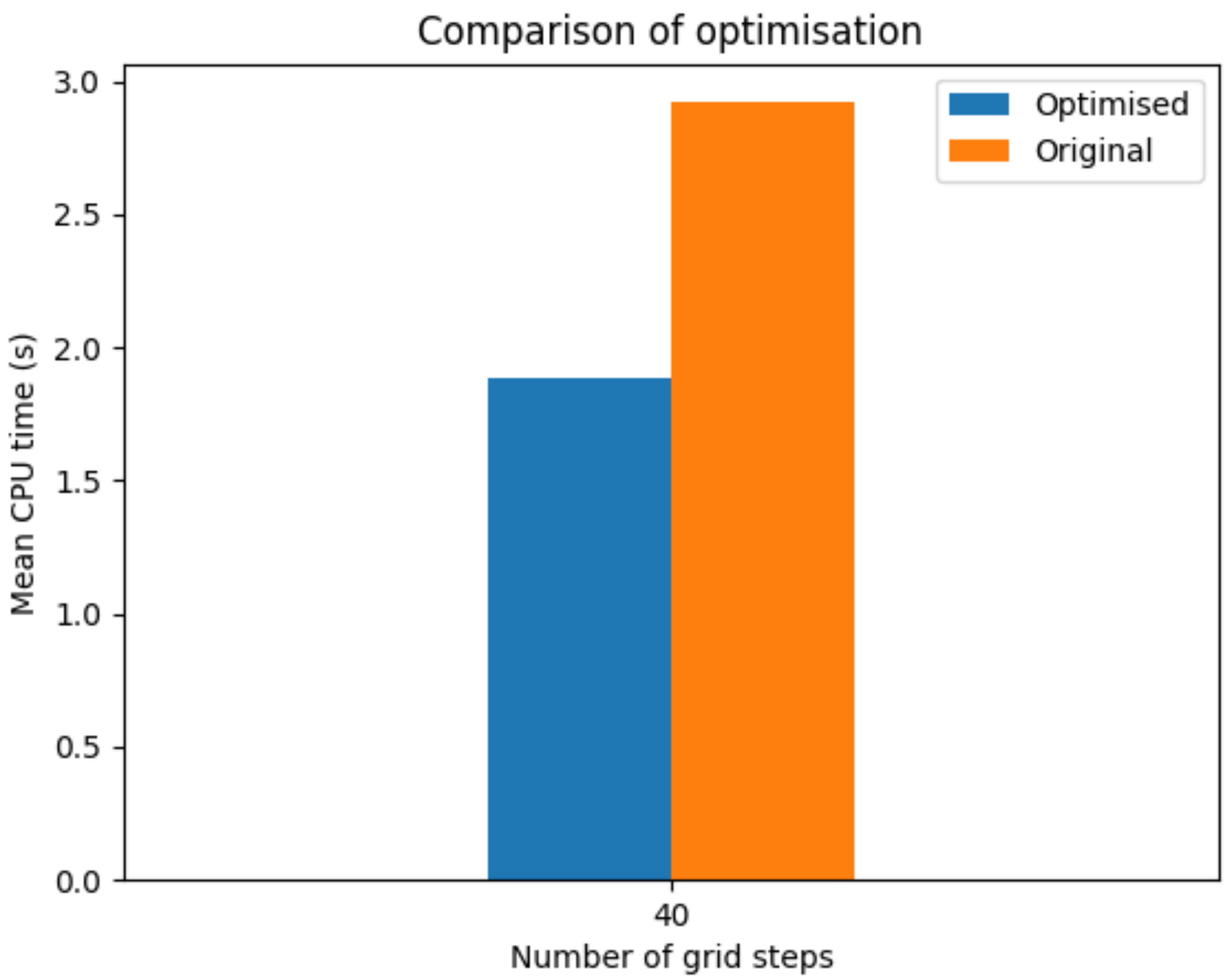


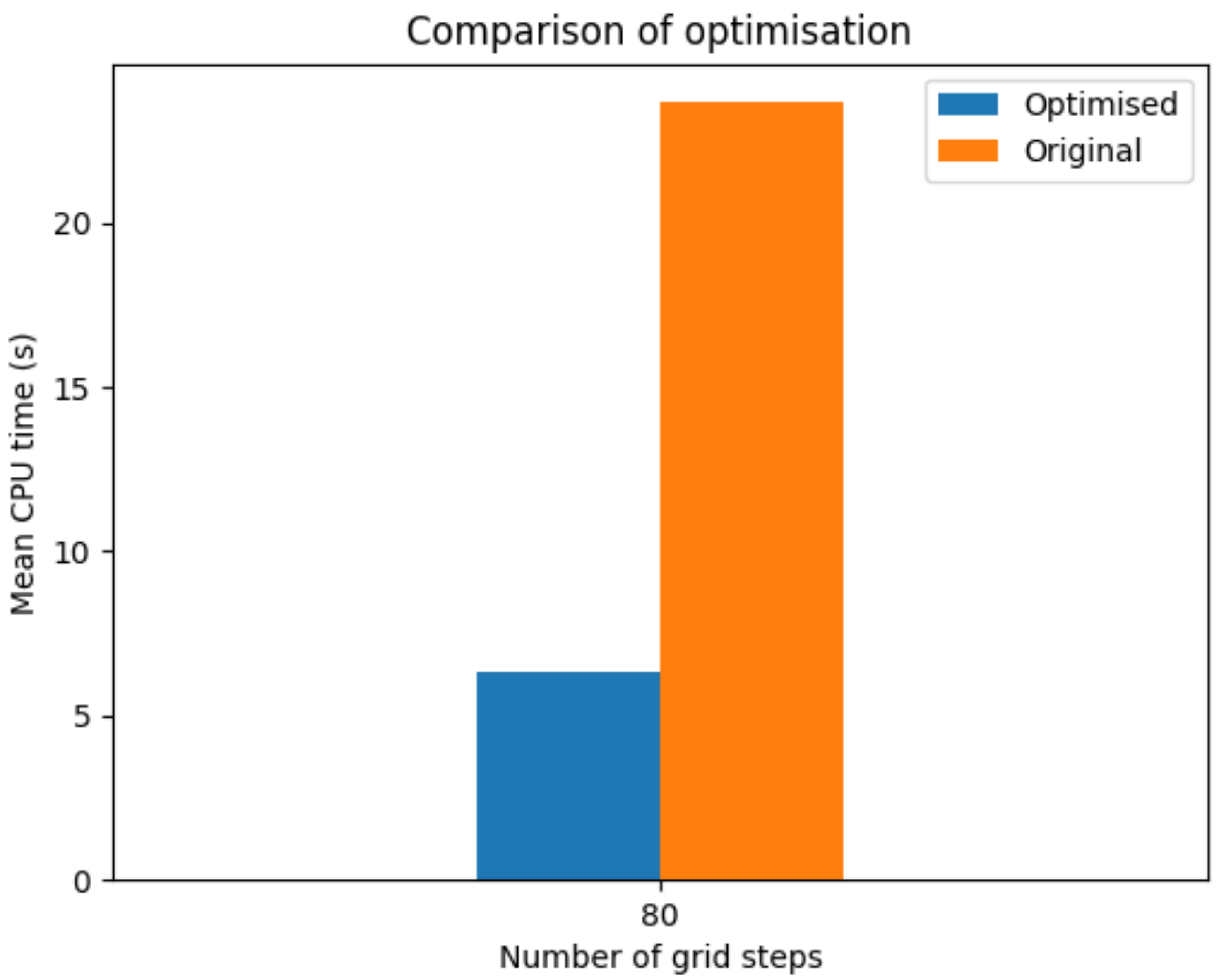
(These plots are from separate runs)

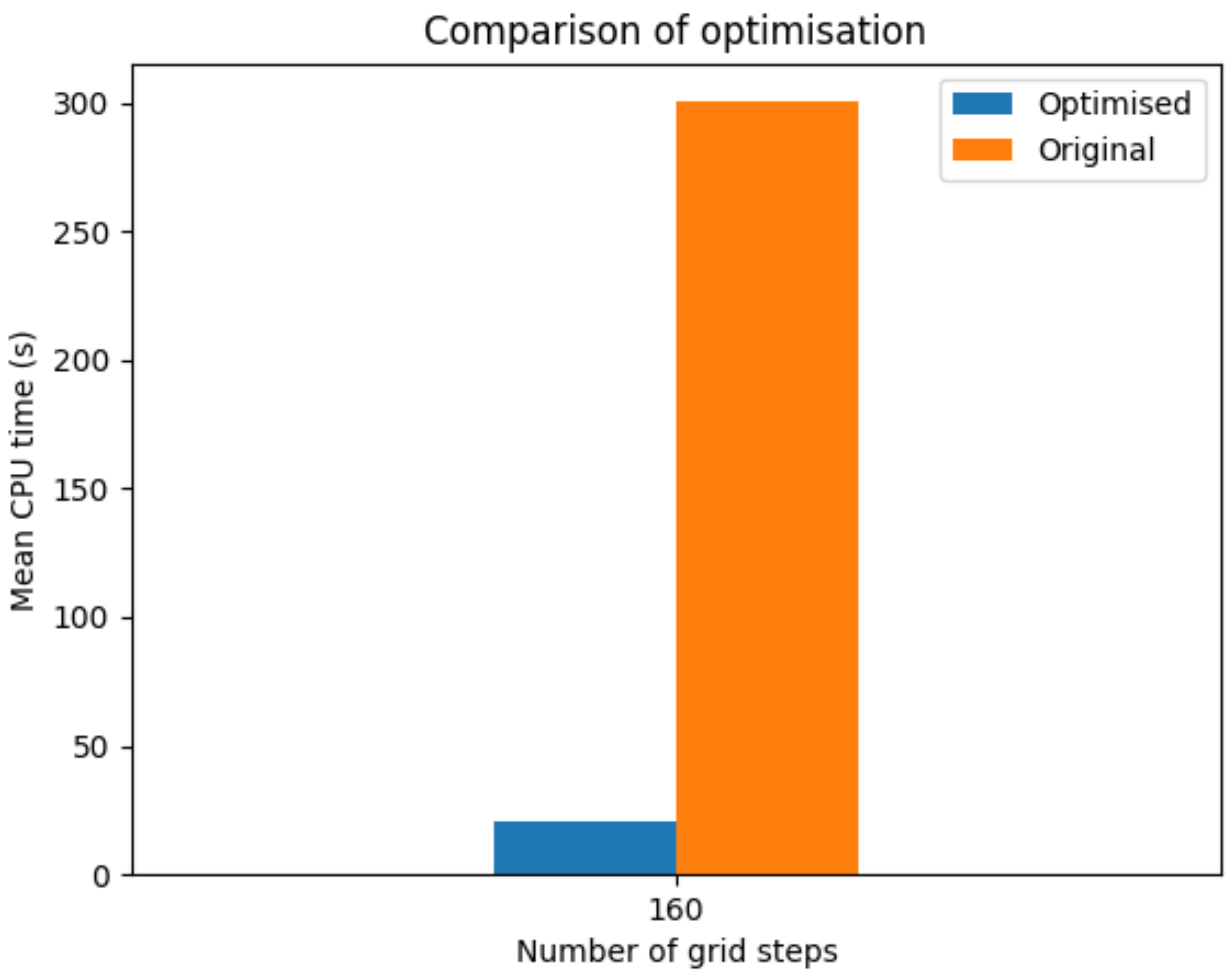
This plot took many hours to make

Optimisation

- Every time the field is requested, all grid nodes are looped over
- To avoid this we can just store a list of 'active' nodes and iterate over that
- For small gains (~10^2) this gives a major speedup and makes fine grids more accessible

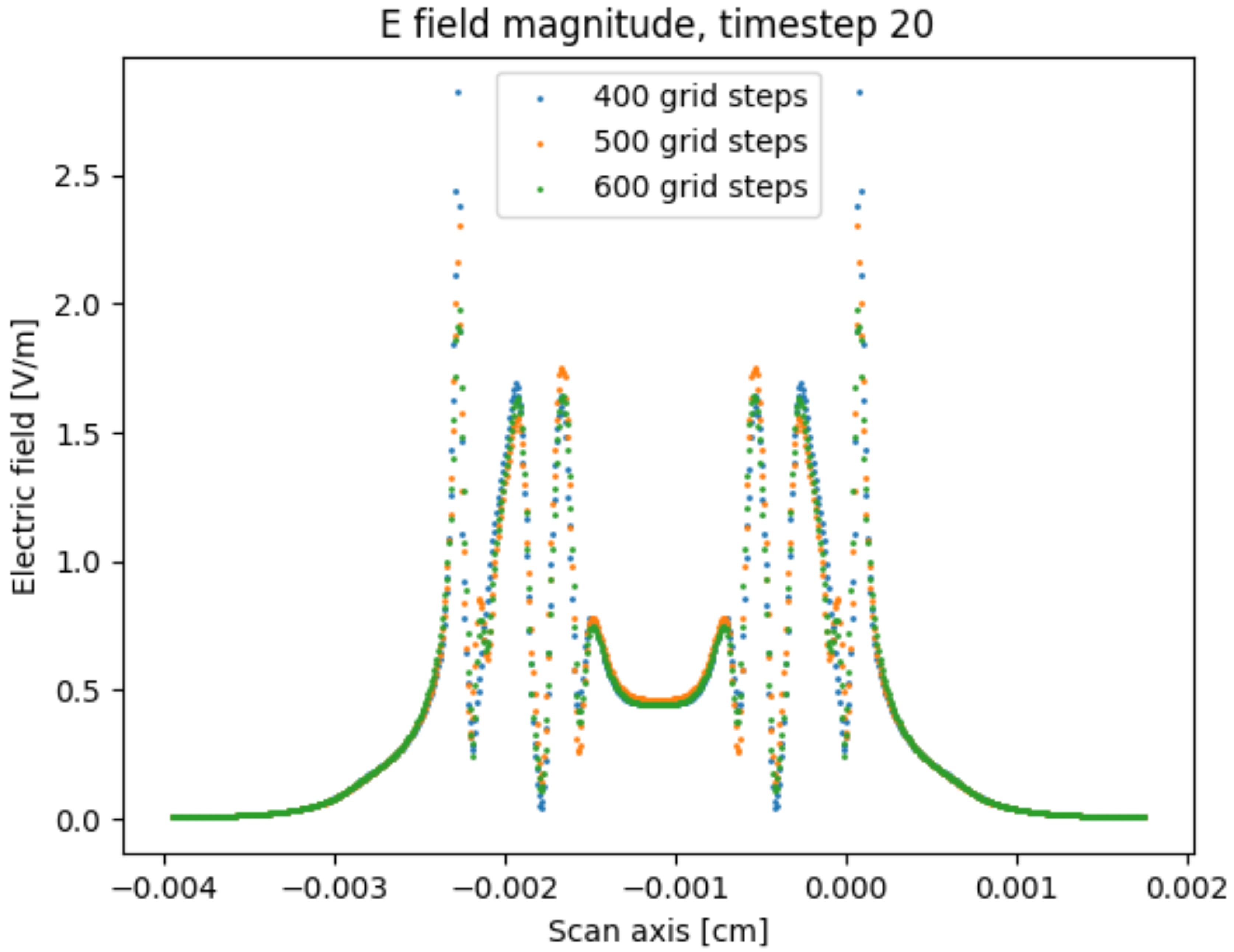


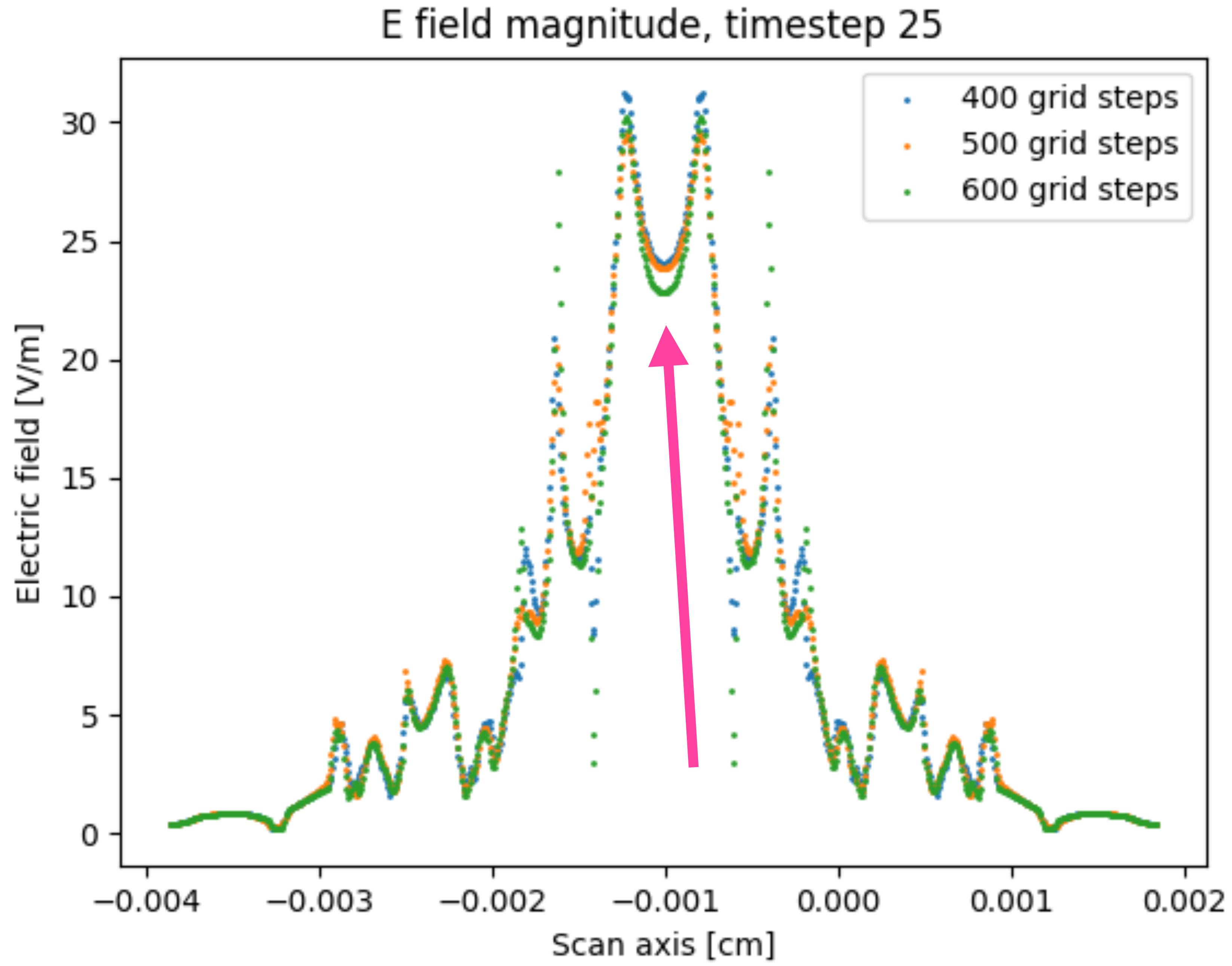


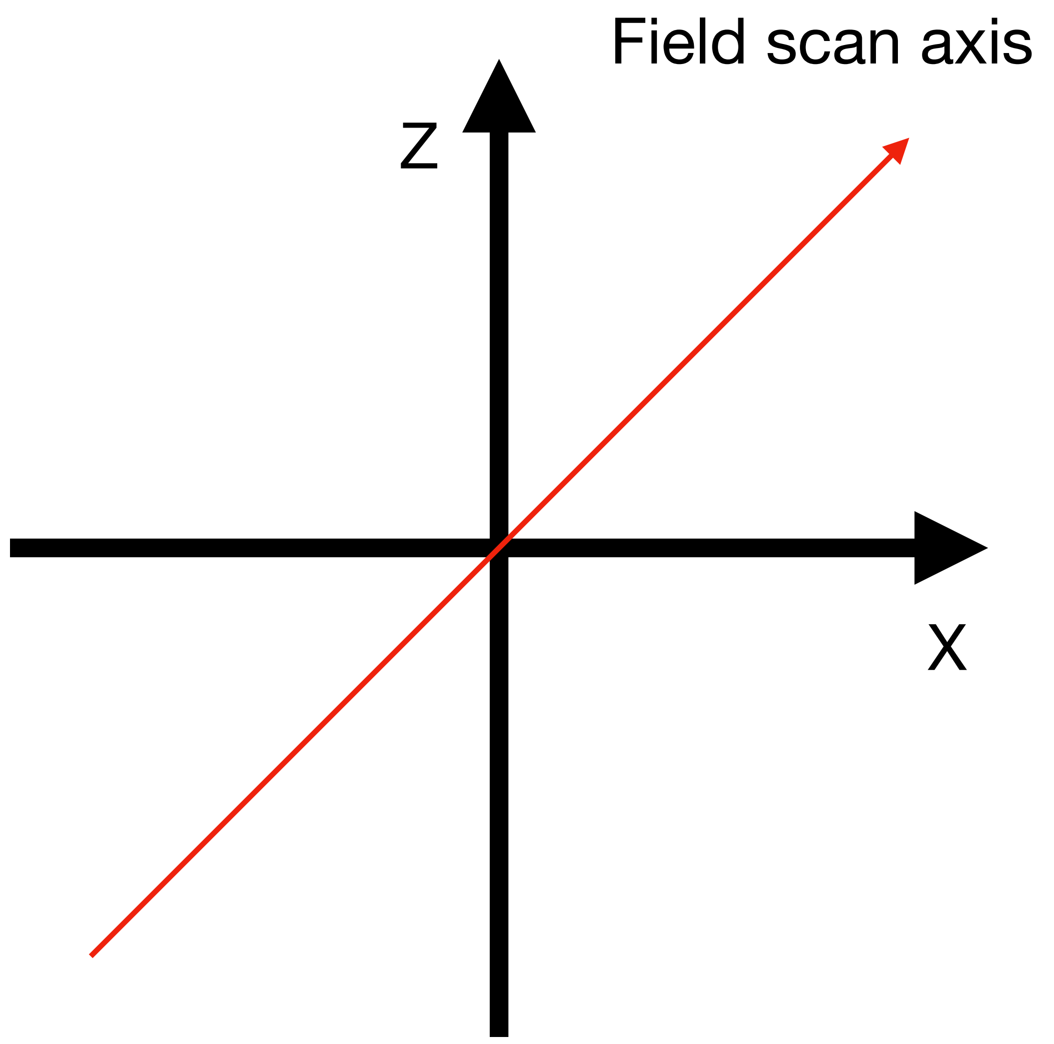


Convergence

- At higher grid spacings there is better convergence but still a small disagreement in certain regions







These plots are centred on the mean position of all particles

Future work

- Investigate using a point charge field instead of axisymmetric rings
- Run many avalanches with and without space charge effect and investigate the impact
- We are still working in a free space approximation: try adding a parallel plate boundary condition and see what the effect is
- Optimise further…