

Field Response Simulation (Garfield++) Update New Response File Summary

Jiangmei Yang

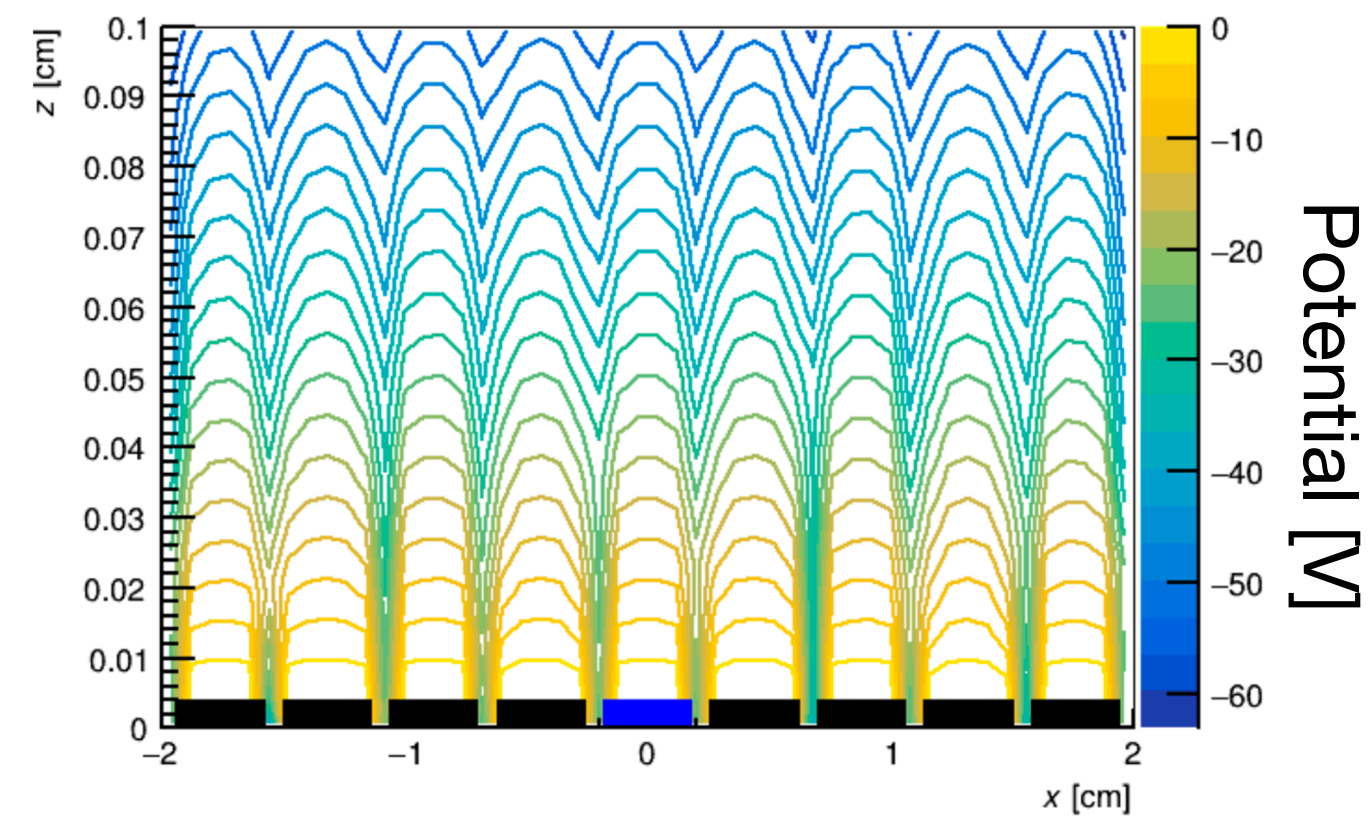
PhD Student, jyangdr@connect.ust.hk

May. 10th, 2024

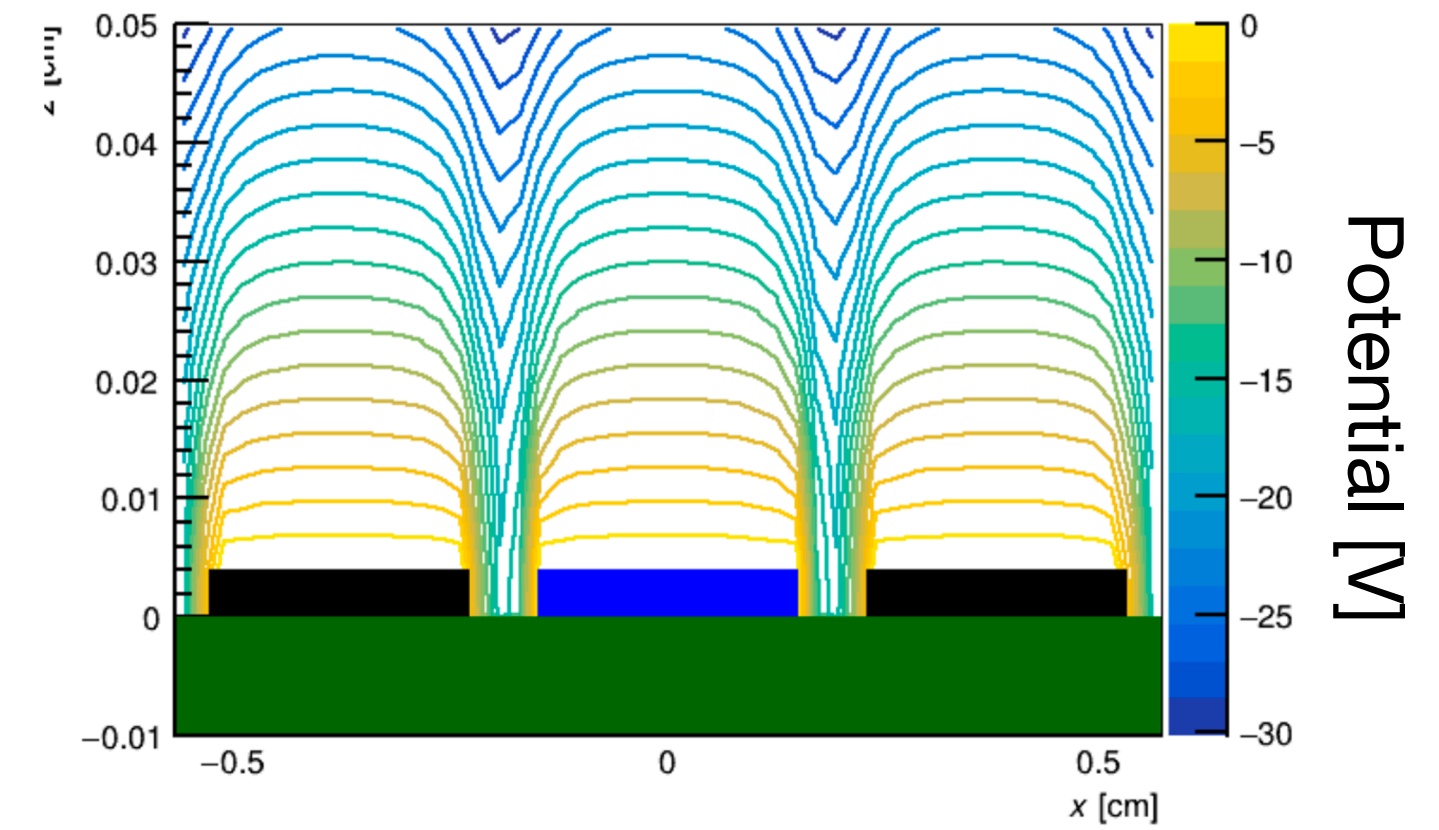
2x2 Sim_Cal_meeting

From last update (Mar. 26th)

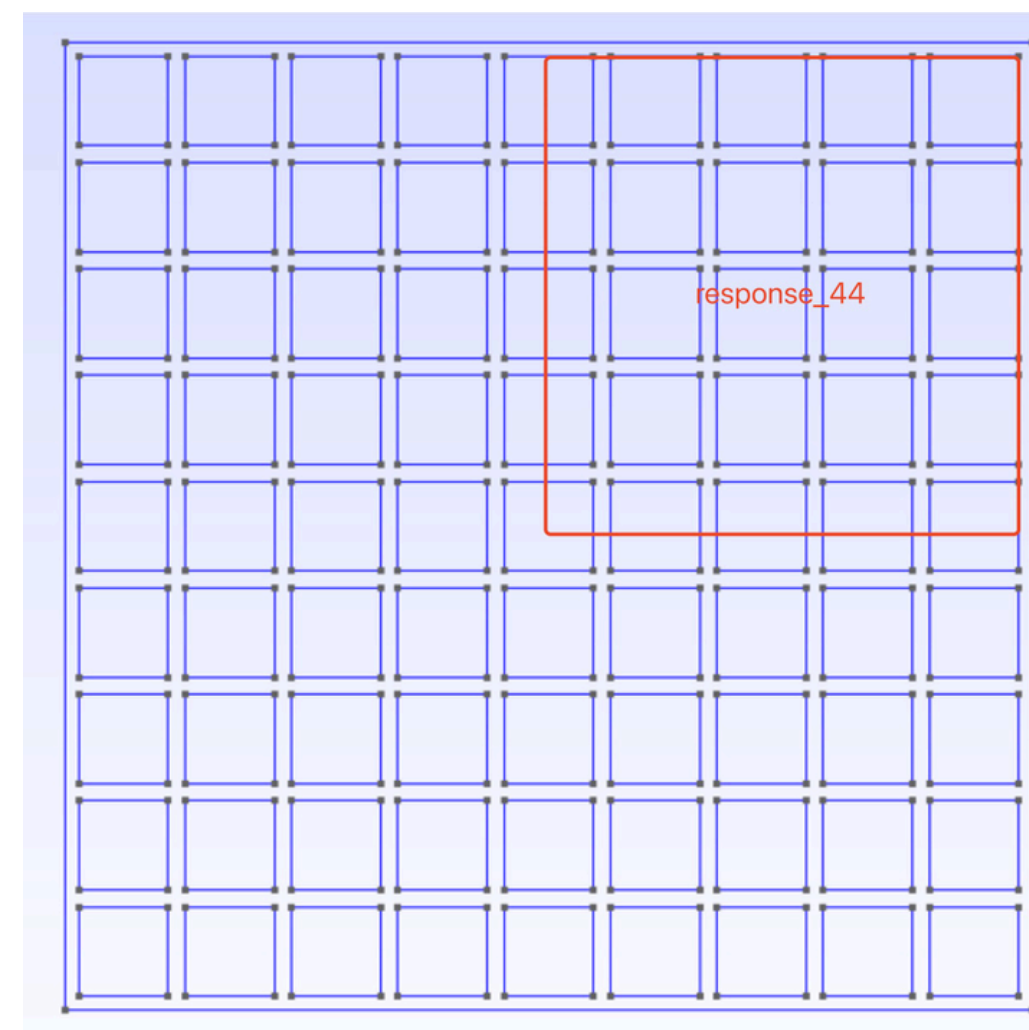
- In the previous simulation, we assigned a small offset voltage on the backing plane in order to drift electrons to the pixels
- -50V used in generating current response_38 and response_44
- Modify potential between pixels a lot
- New Geometry: Including a FR4 layer, with dielectric constant but not assigning voltage
- FR4: 0.1mm thickness
- Closer to reality
- Able to drift all electrons to the pixels



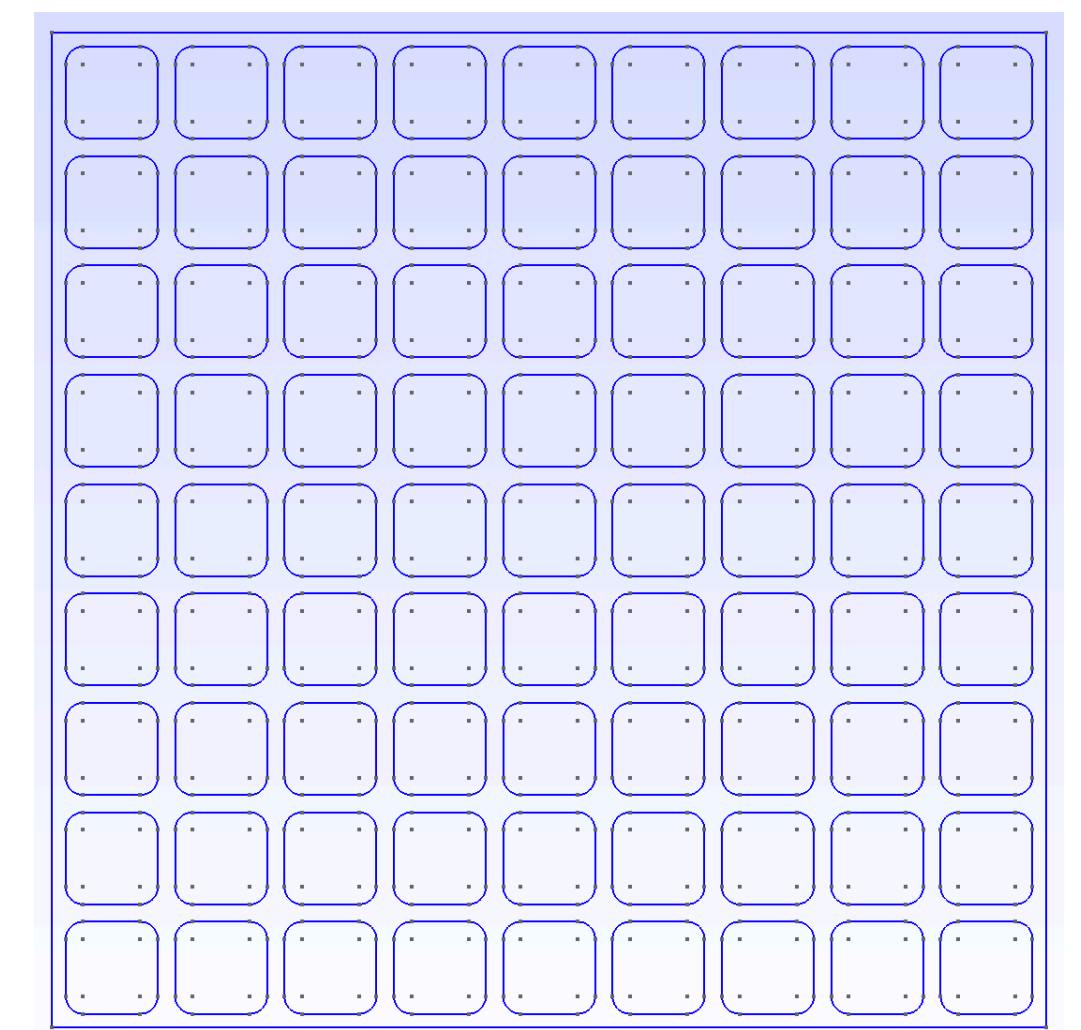
-50V as backing plane voltage



FR4 dielectric with no voltage assigned



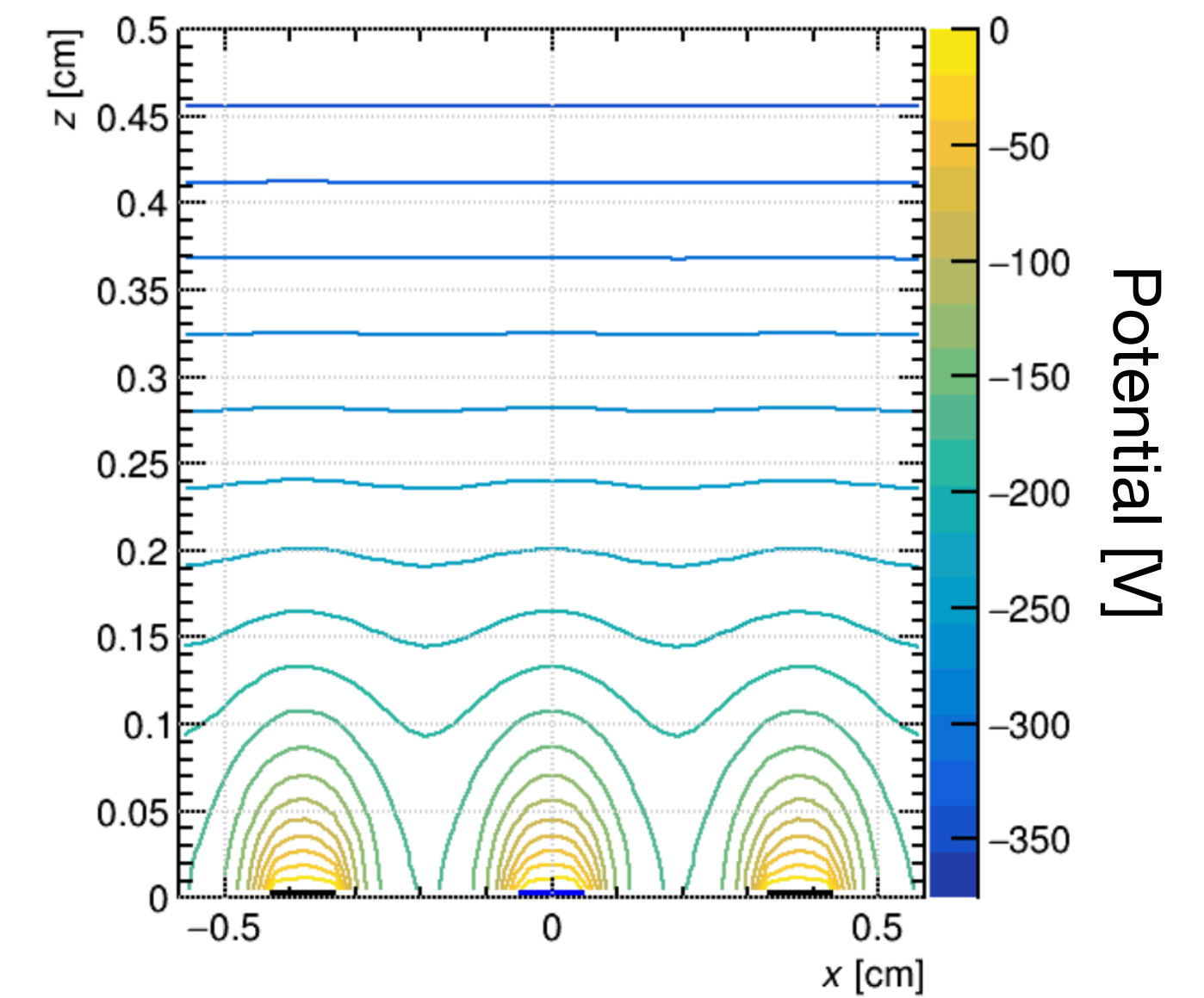
Square Corner



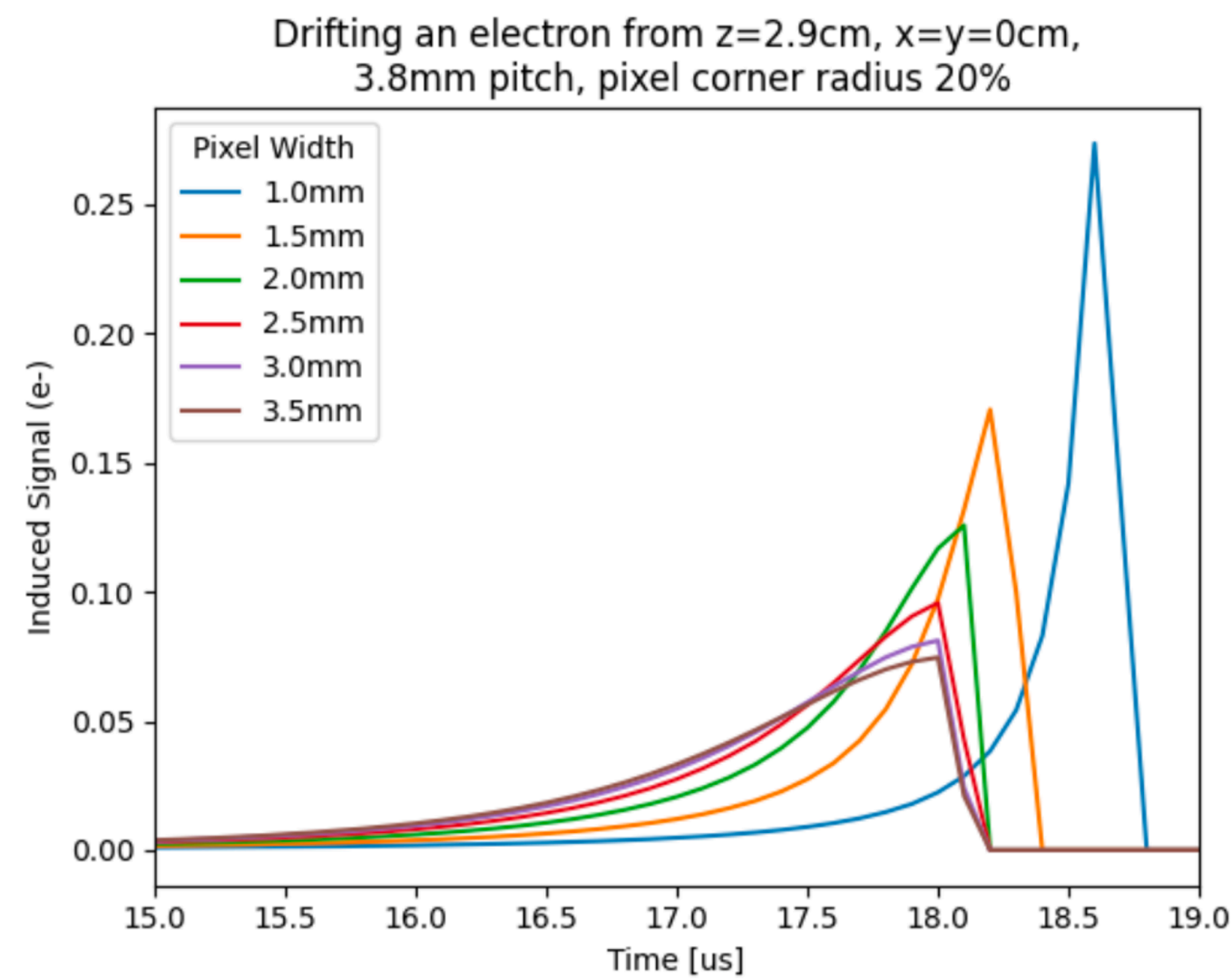
Round Corner

Induced Signal vs Pixel Size

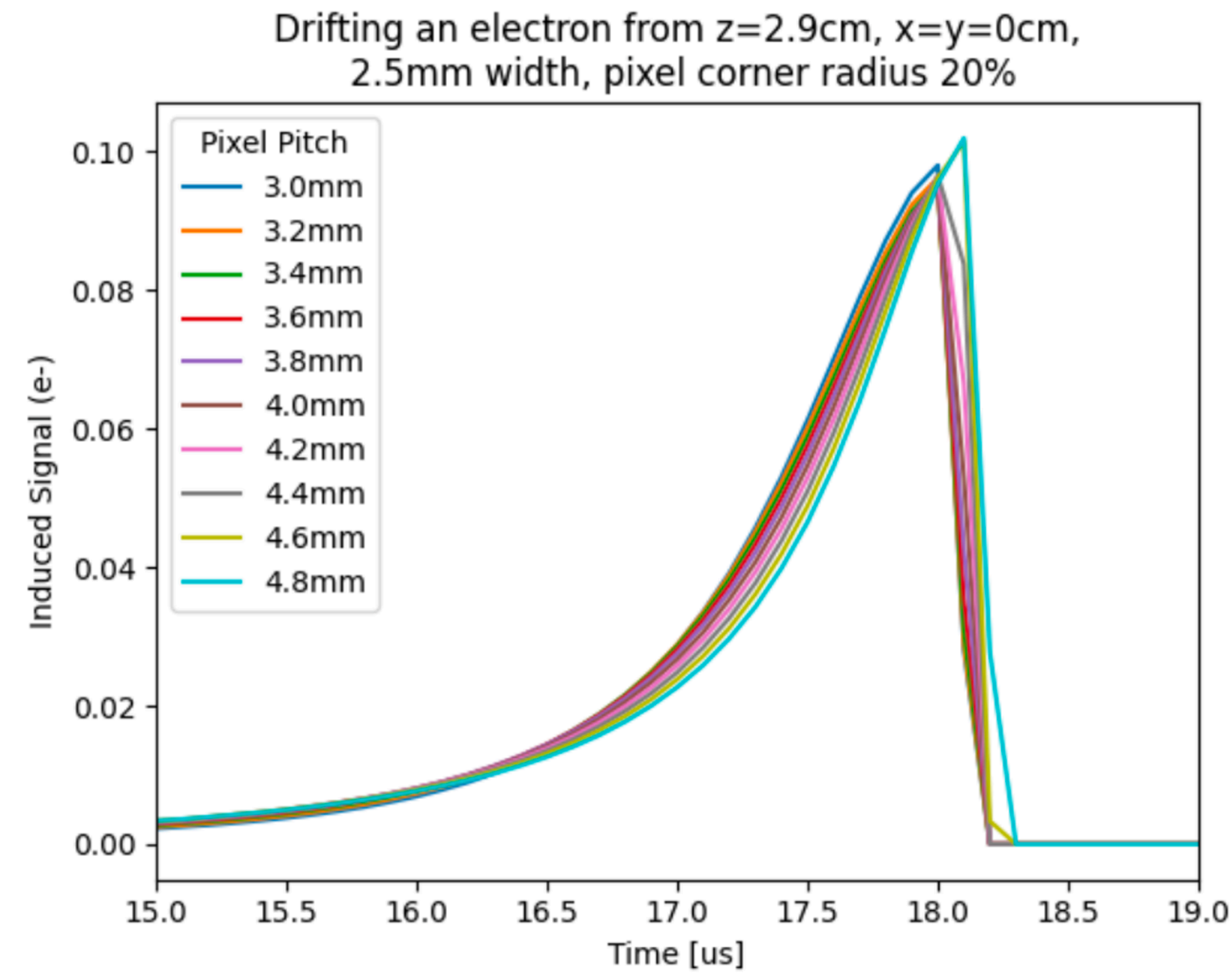
- Given the changes in fill factor, pixel size changes
- Induced signal shape and arrival time change
- Smaller Pixel Width, less uniform field, longer e- drift time



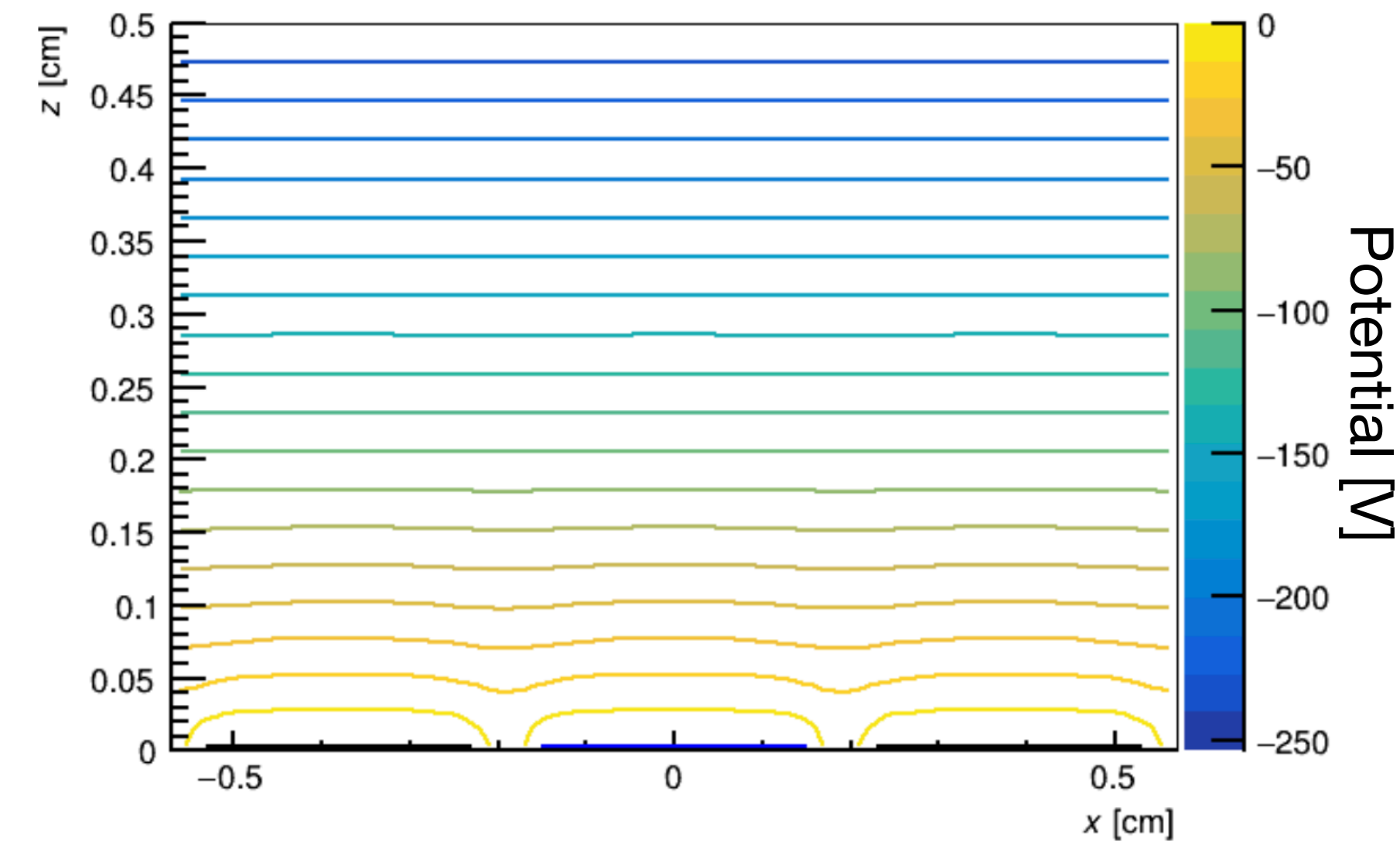
3.8mm pixel pitch, 1.0 mm width



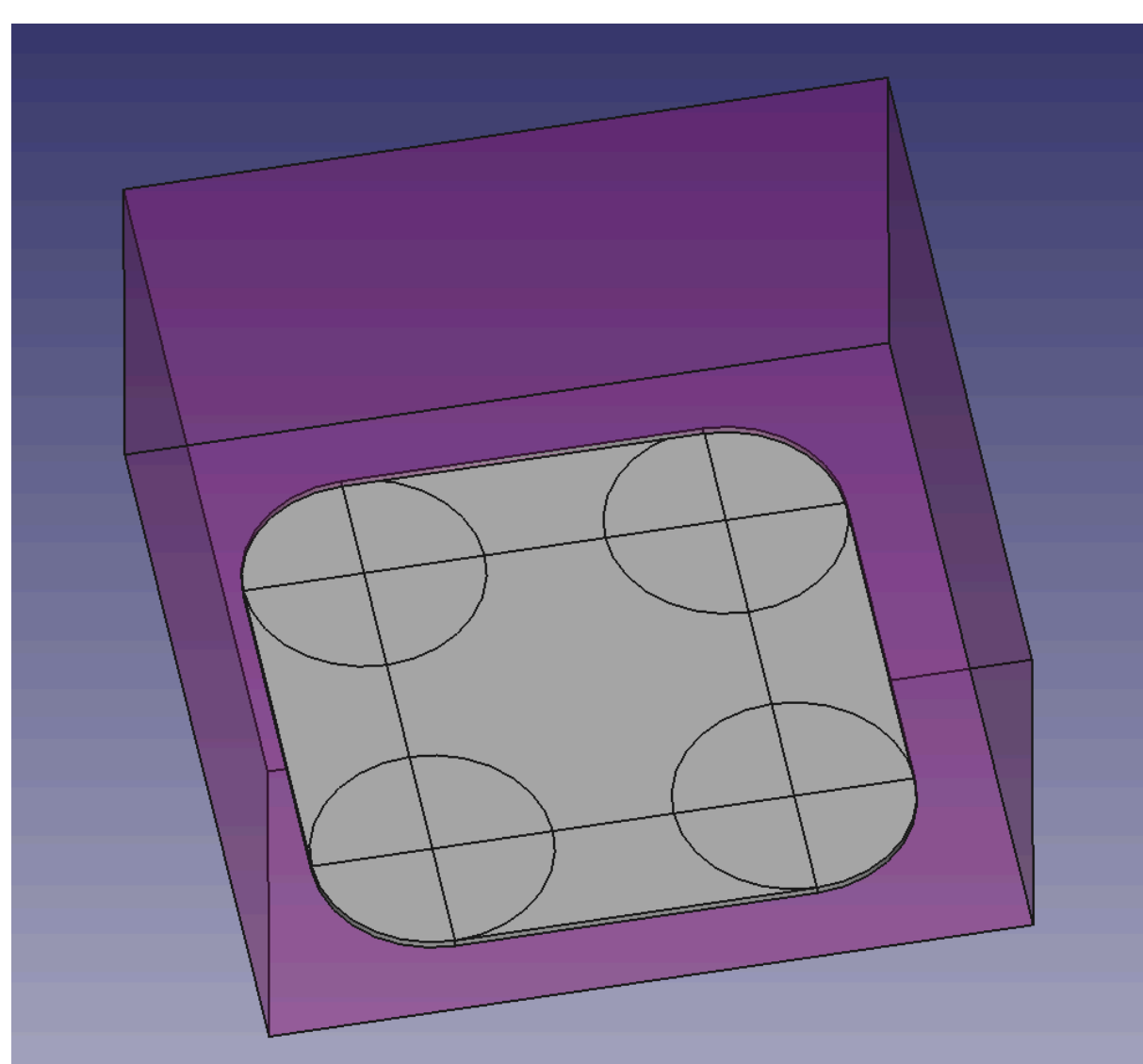
Induced Signal vs. Pixel Width



Induced Signal vs. Pixel Pitch



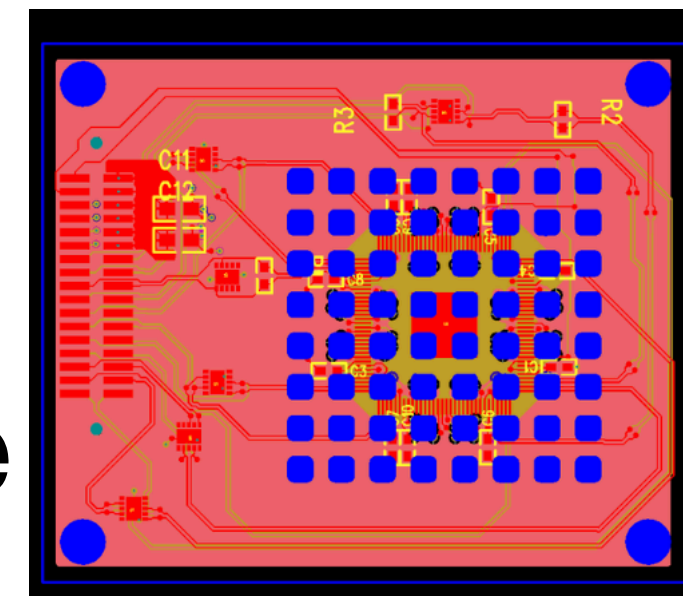
3.8mm pixel pitch, 3.0 mm width



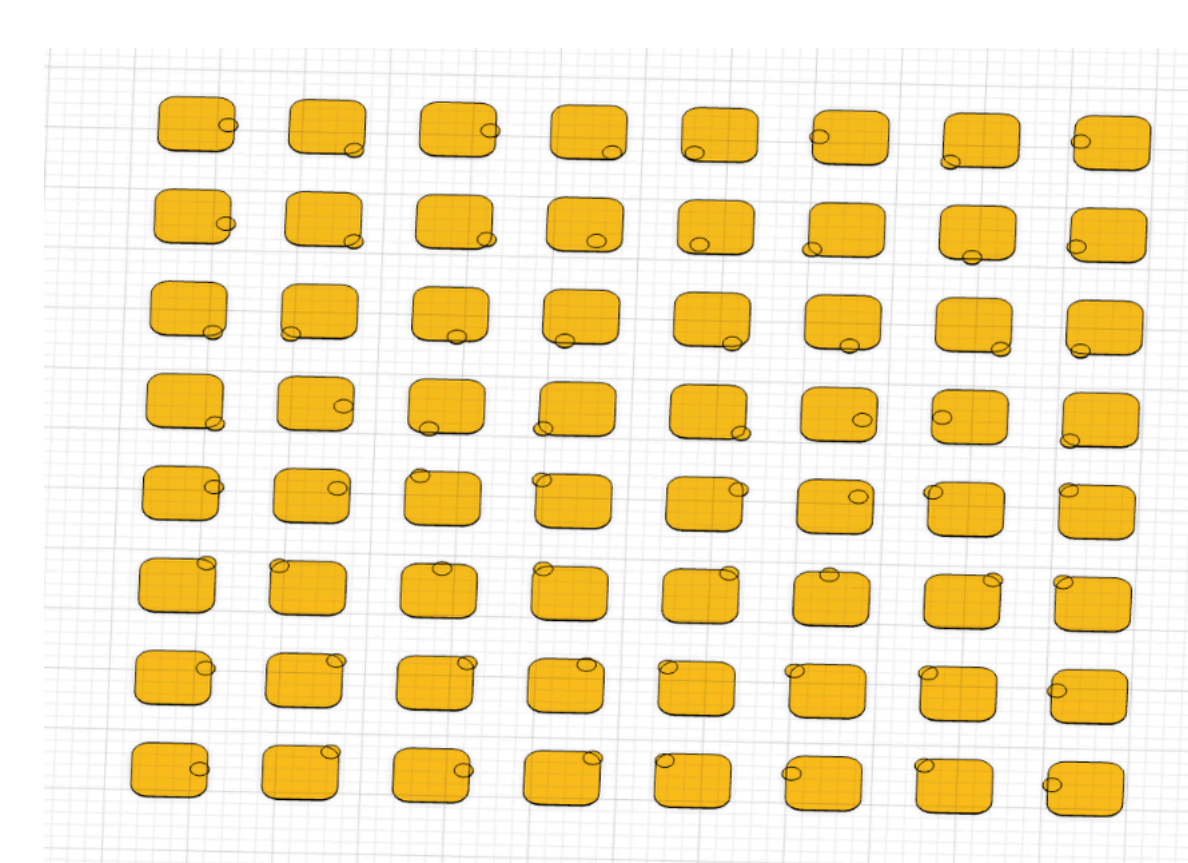
Current response_38

- In larnd-sim (module 1), 3.8mm-pitch pixel uses 3.0mm as pixel width (LArPix v2b/v3)
 - 0.6mm as corner circle radius
- In the latest development, field factor changed from 70% -> 40%
 - pitch size should be smaller

- Convert Single Chip (64 pixels) Gerber to CAD
- Since we are using single pixel induced signal as each pixel's response, remove the vias pattern.

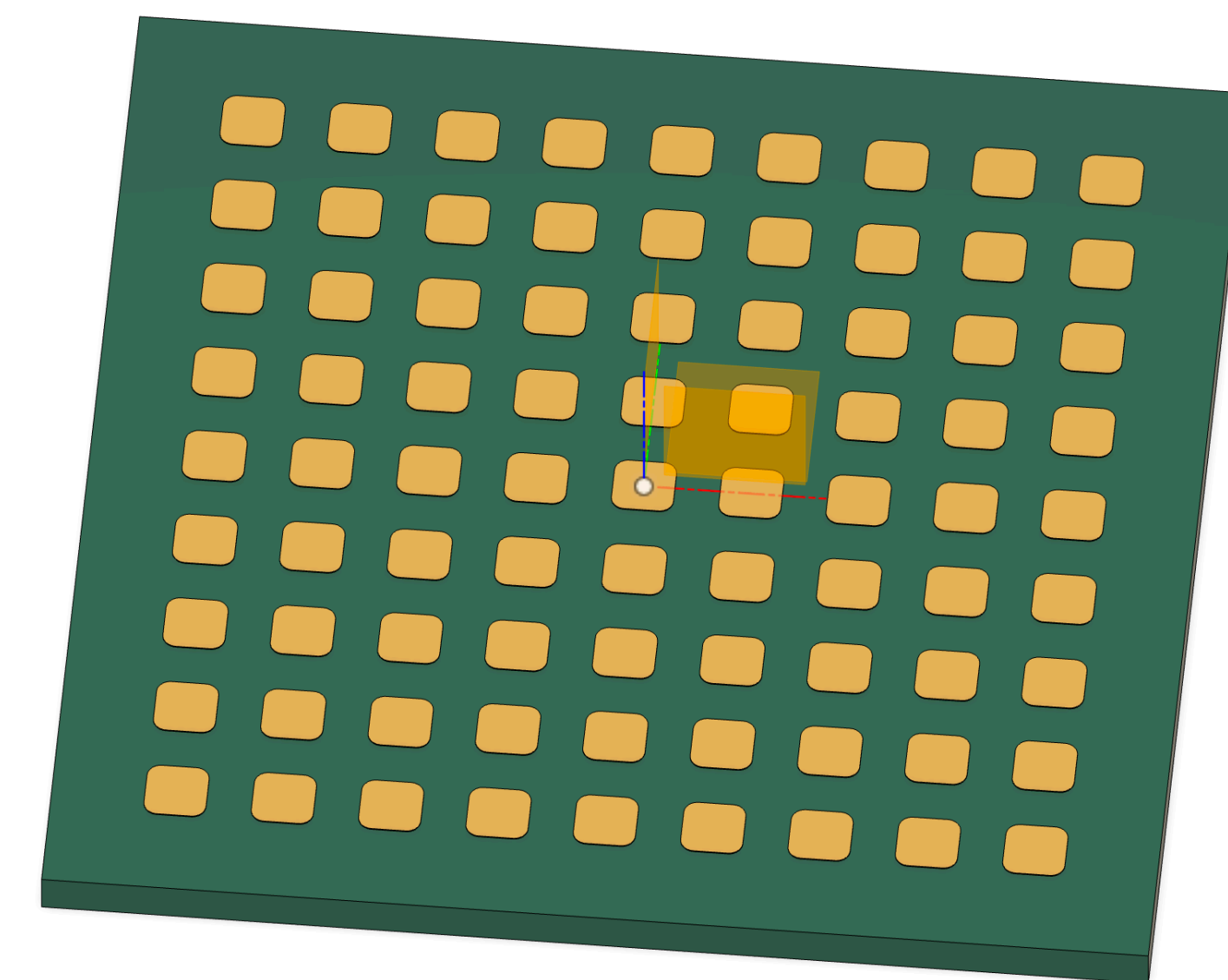


Single chip gerber



Single Chip 8x8 pixels with vias

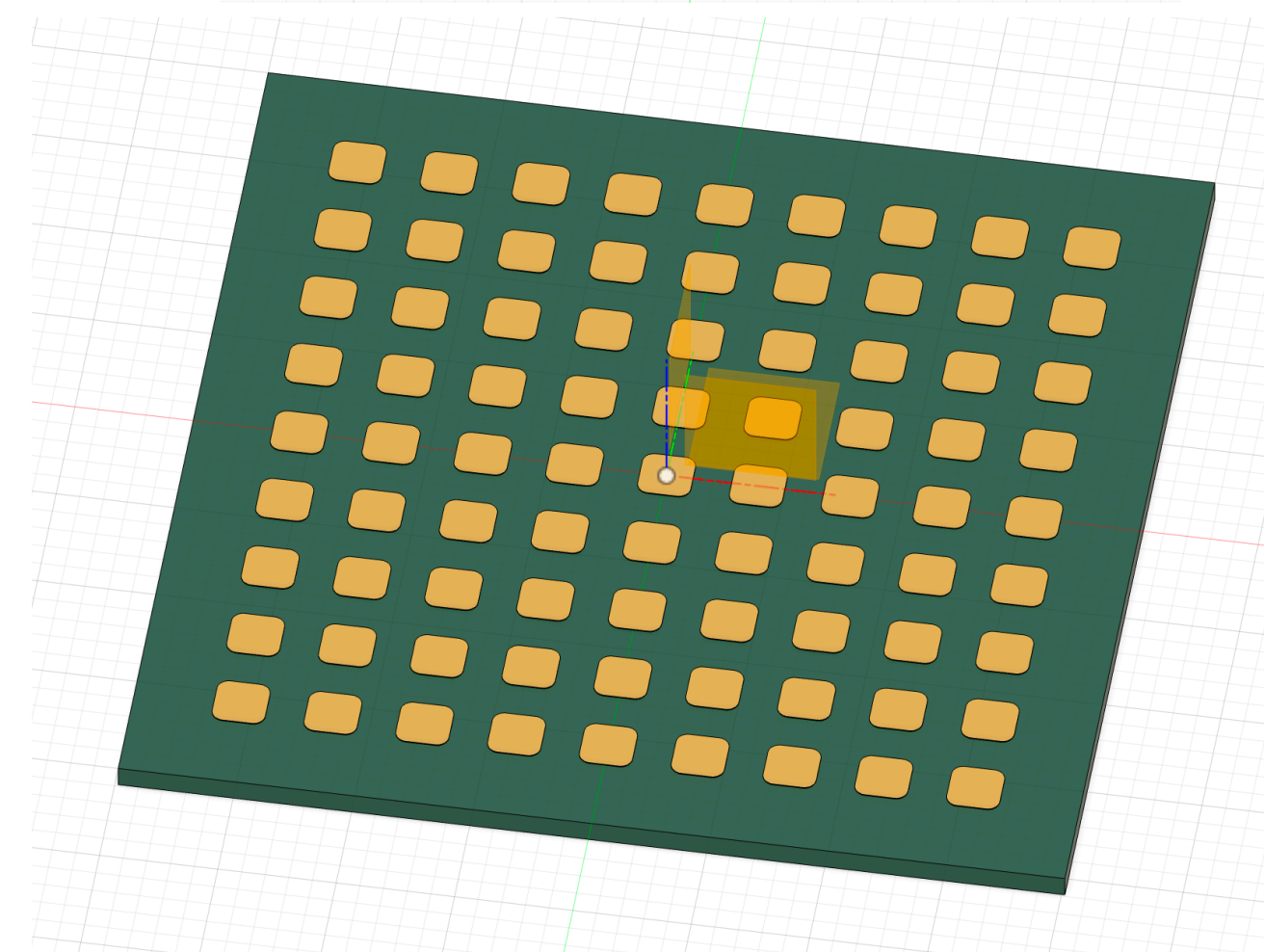
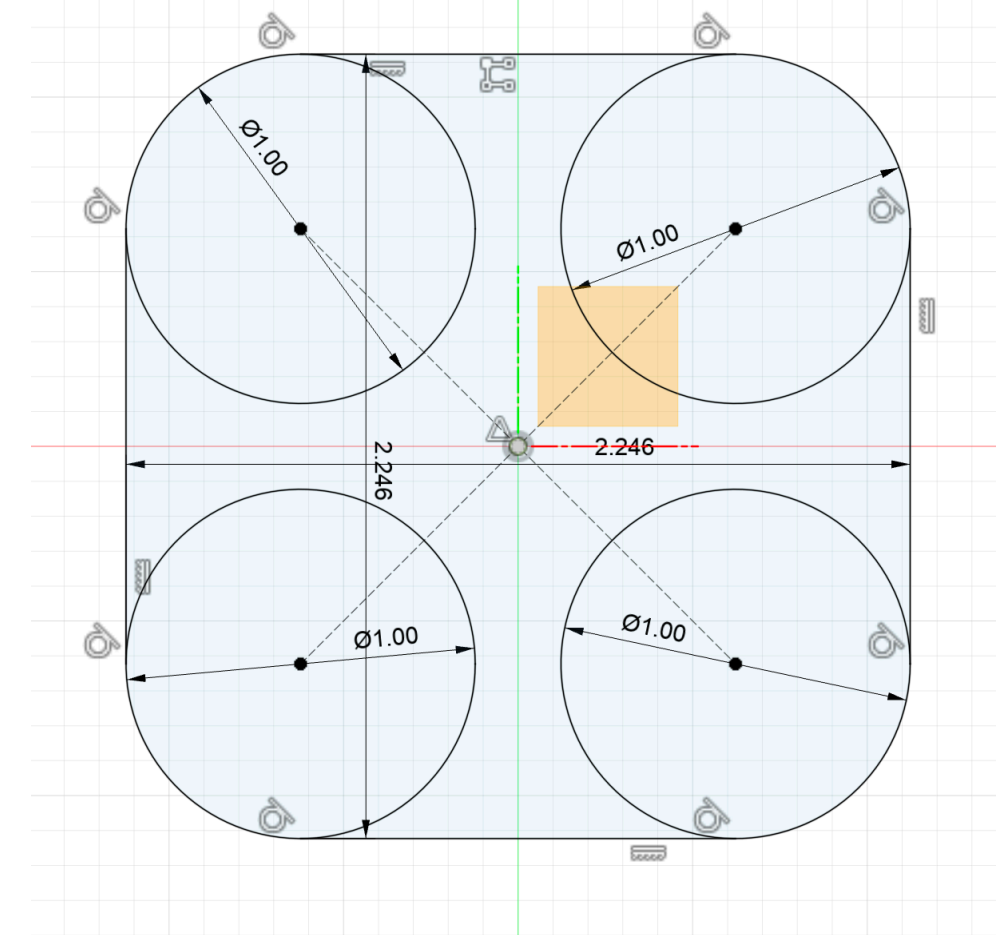
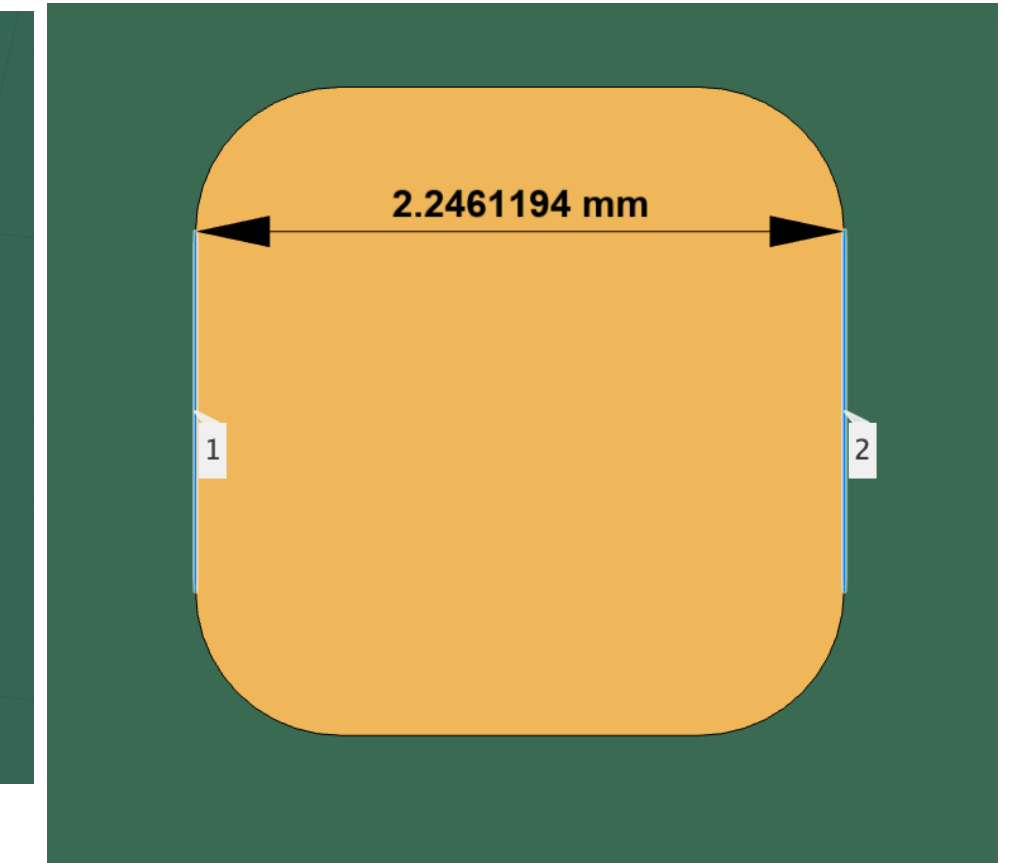
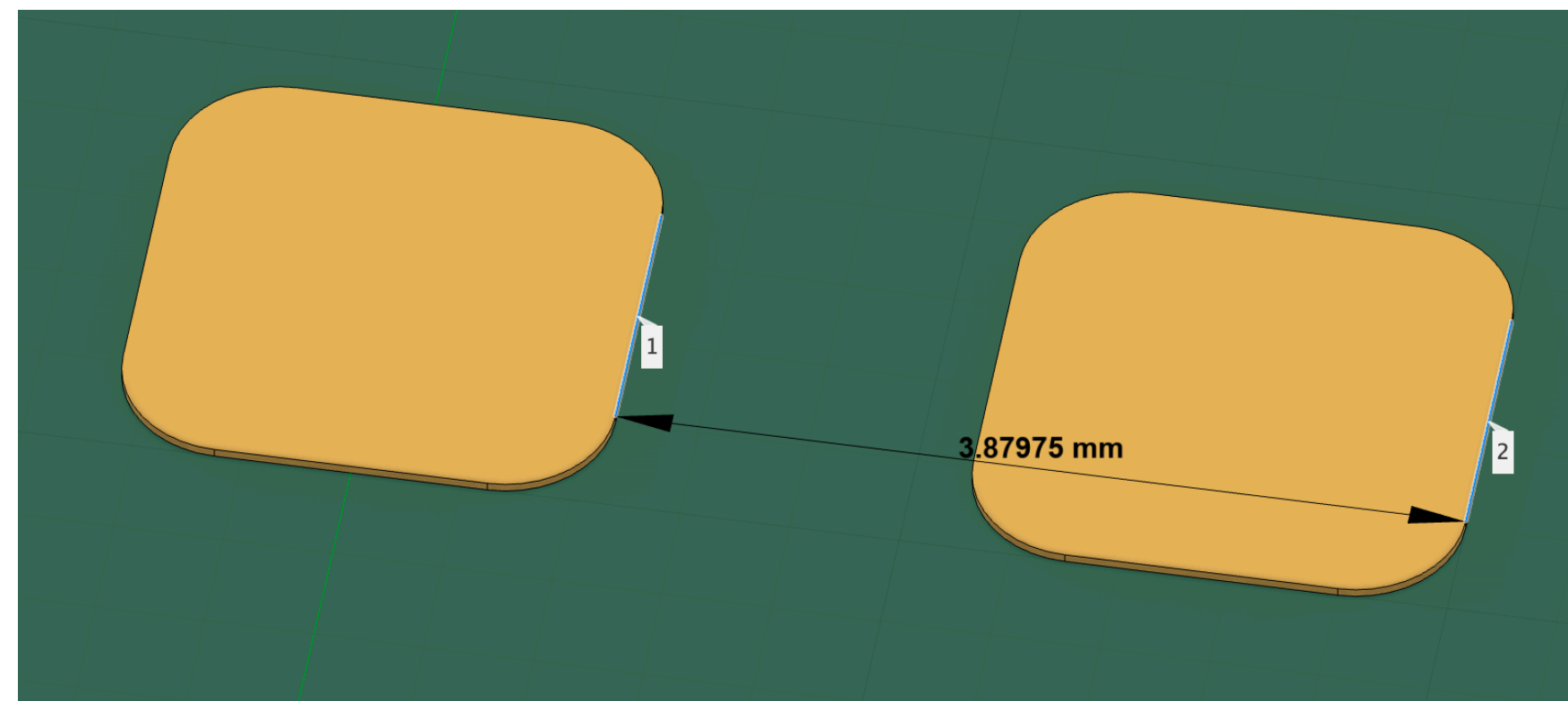
- Current response_38 and 44 was simulated with 9x9 pixels, so expand the single chip 64 pixels to 81 pixels and using central pixel as readout
- Modified the pixel pitch to be consistent with land-sim's



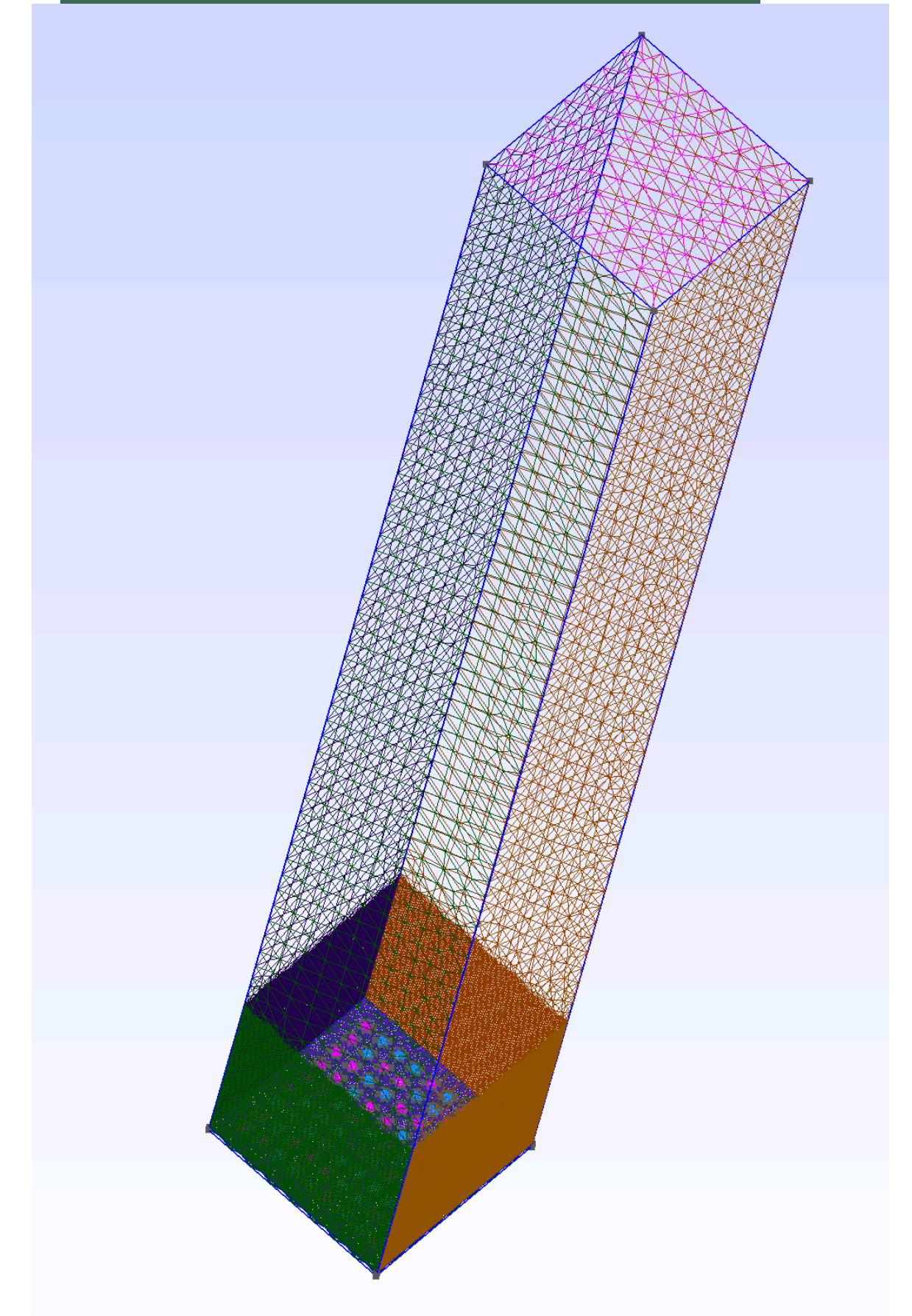
New response_38 (v2b) 9x9 pixels no vias

response_38_v2b dimension

- Pixel Pitch: 3.87975mm
- Pixel Width: 2.2461194mm
- Corner Circle radius: 0.5mm
- Pixel Thickness: 1oz copper, 43um (Measured by Armin)
- FR4 Thickness: 1mm
- Drift Volume: 4cm*4cm*31cm (Cathode position)
 - HV: -15500V
- Electrons drift from 30.27225cm
- Time interval [0,200) us
 - response_38 was [0,189.8) us



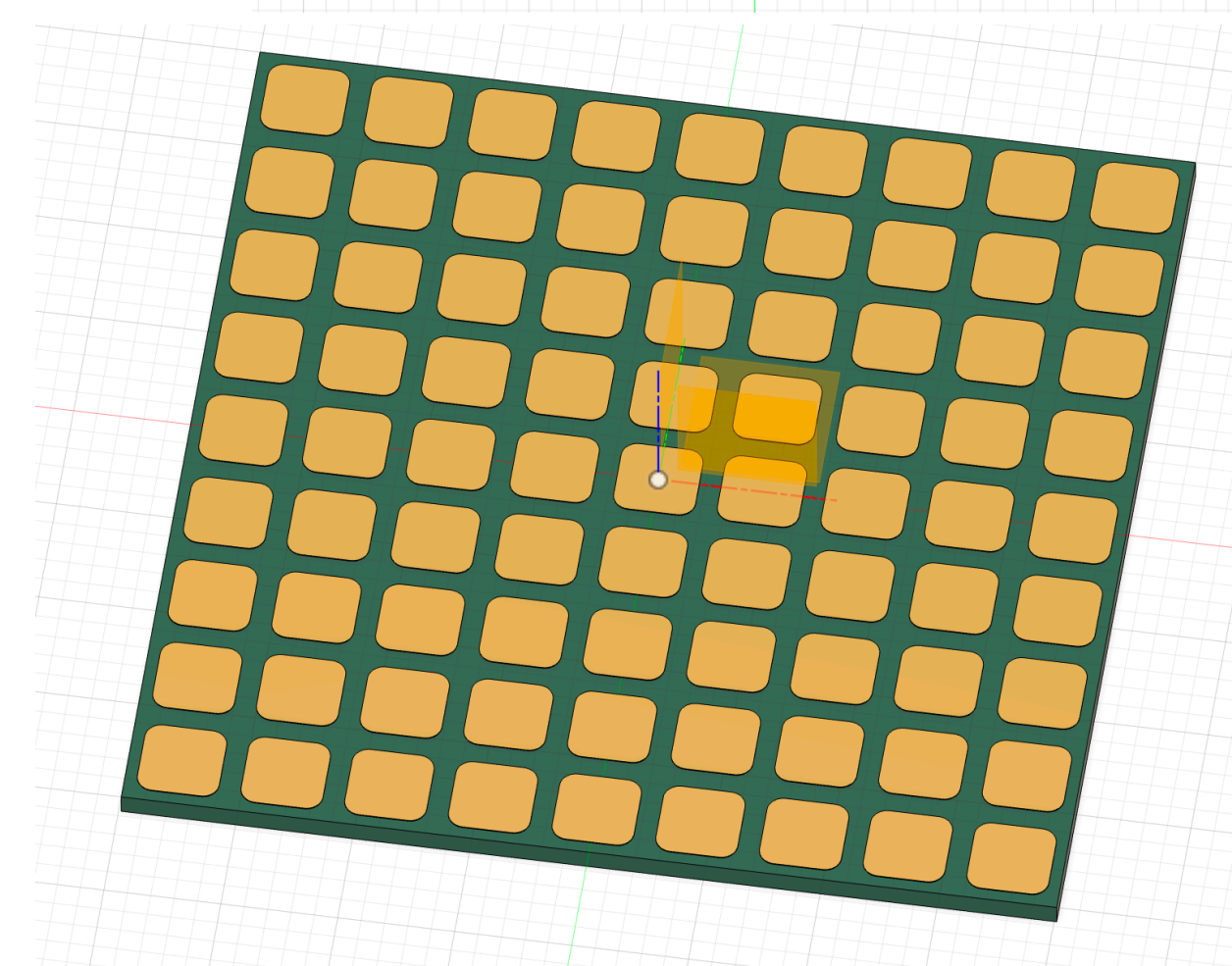
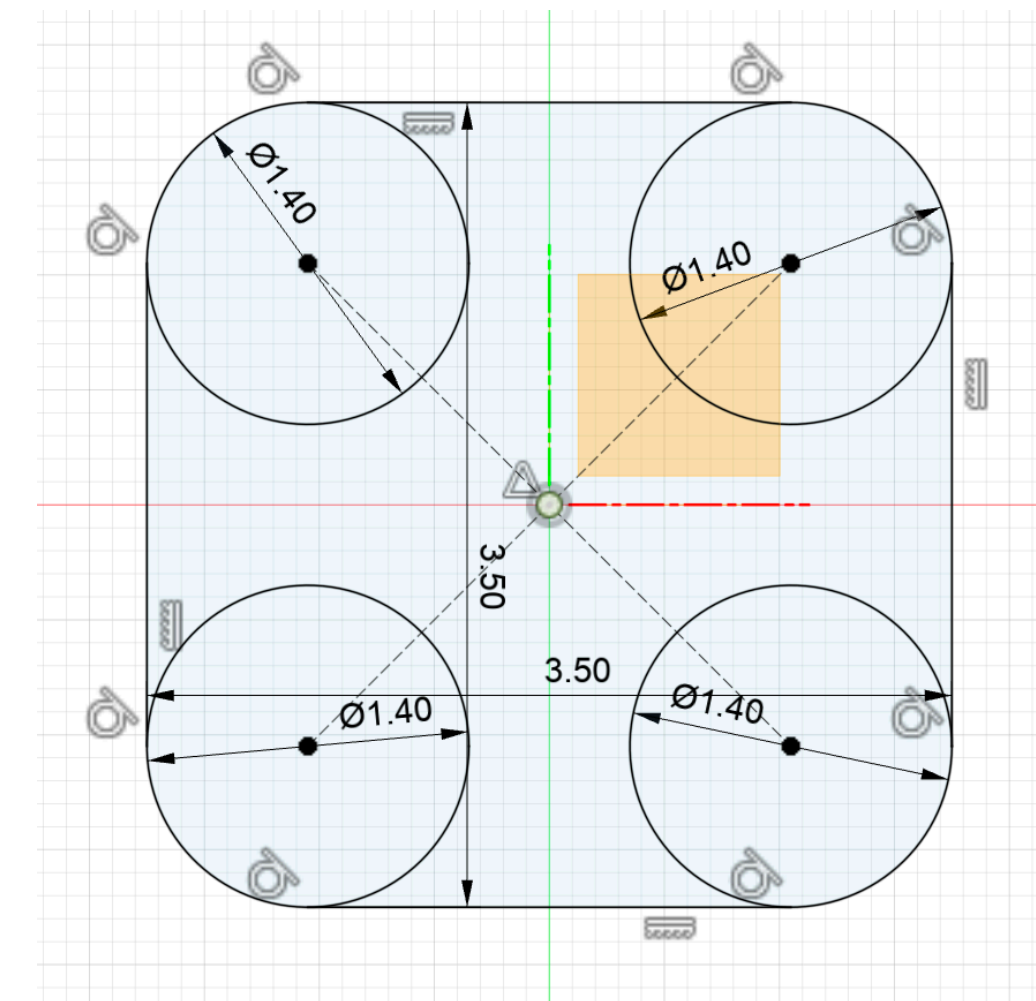
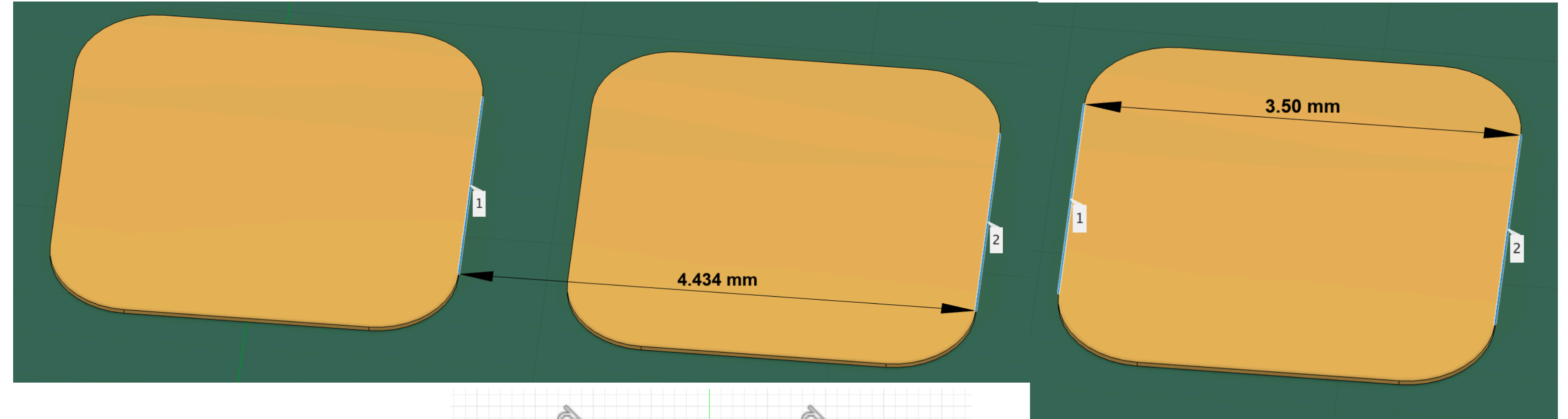
9x9 pixels no vias



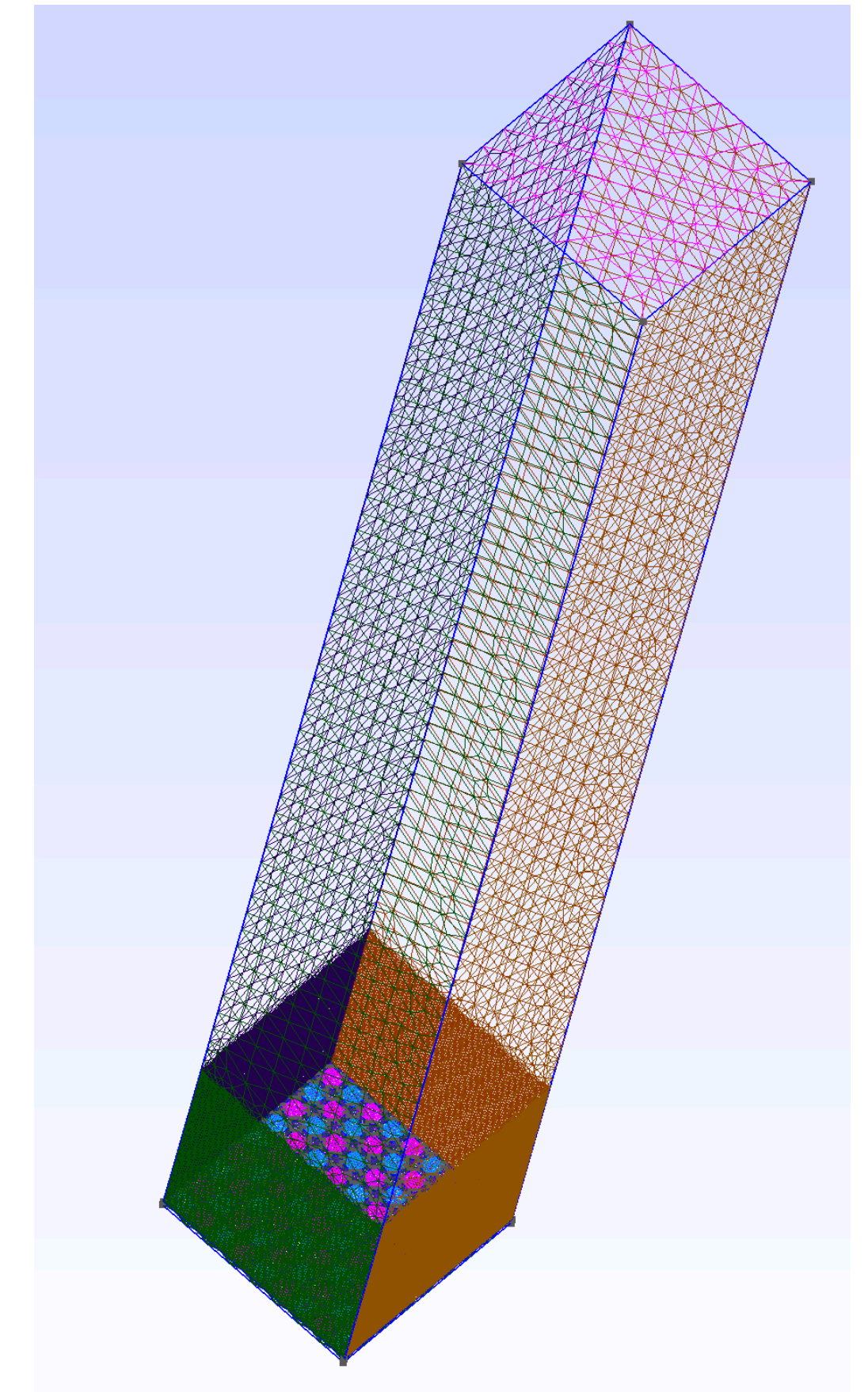
0.5mm Mesh for $z \leq 4\text{cm}$;
5mm Mesh for $z > 4\text{cm}$

response_44_v2a dimension

- Pixel Pitch: 4.434mm
- Pixel Width: 3.50mm
- Corner Circle radius: 0.7mm
- Pixel Thickness: 1oz copper, 43um (Measured by Armin)
- FR4 Thickness: 1mm
- Drift Volume: 4cm*4cm*31cm (Cathode position)
 - HV: -15500V
- Electrons drift from 30.27225cm
- Time interval [0,200) us
 - response_44 was [0,189.1) us



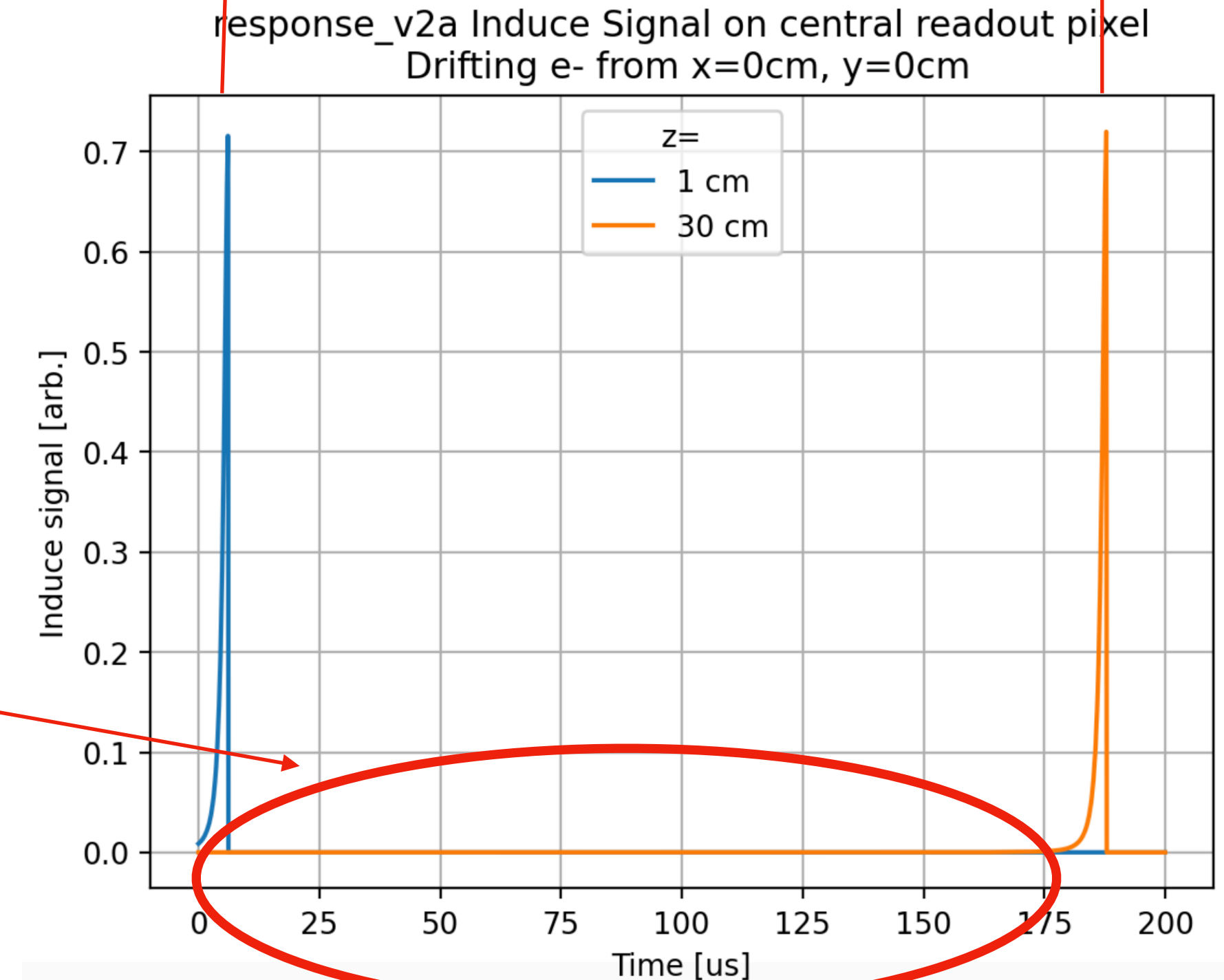
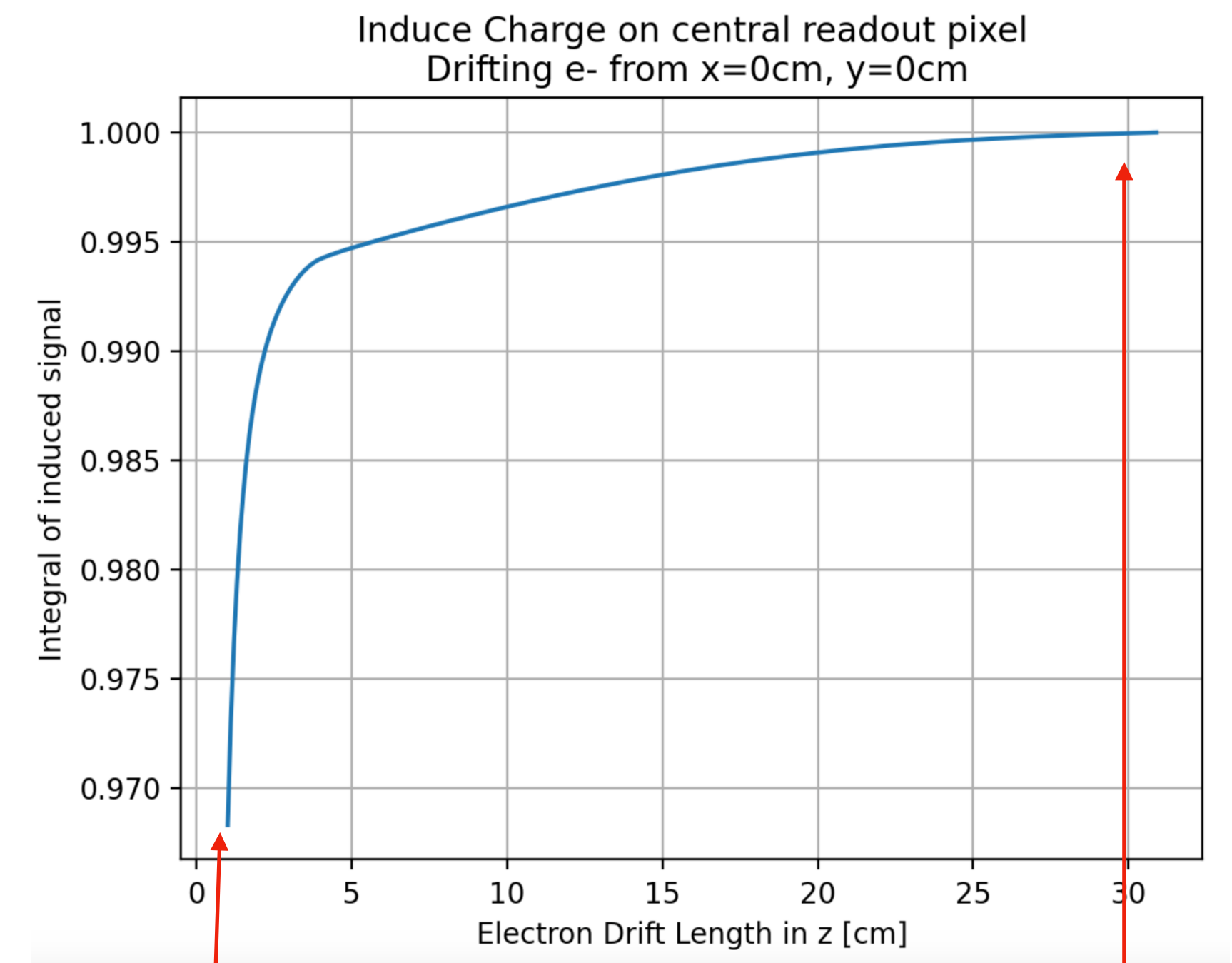
9x9 pixels no vias



0.5mm Mesh for $z \leq 4\text{cm}$;
5mm Mesh for $z > 4\text{cm}$

Normalization?

- Previous response signal was normalized to unity if e- arrives on readout pixel, to 0 if e- arrives on nearby pixels
- Thought:
 - The integration of induce signal may not be unity charge
 - But position dependent
 - As the e- formation position closer to the cathode, the induce signal integration closer to 1 e-
 - e.g. integrated charge from z=1cm and z=30 differ by ~3.5%, which given by the far-field induce signal tail
- So maybe normalization is not necessary?



- From Dan Douglas:

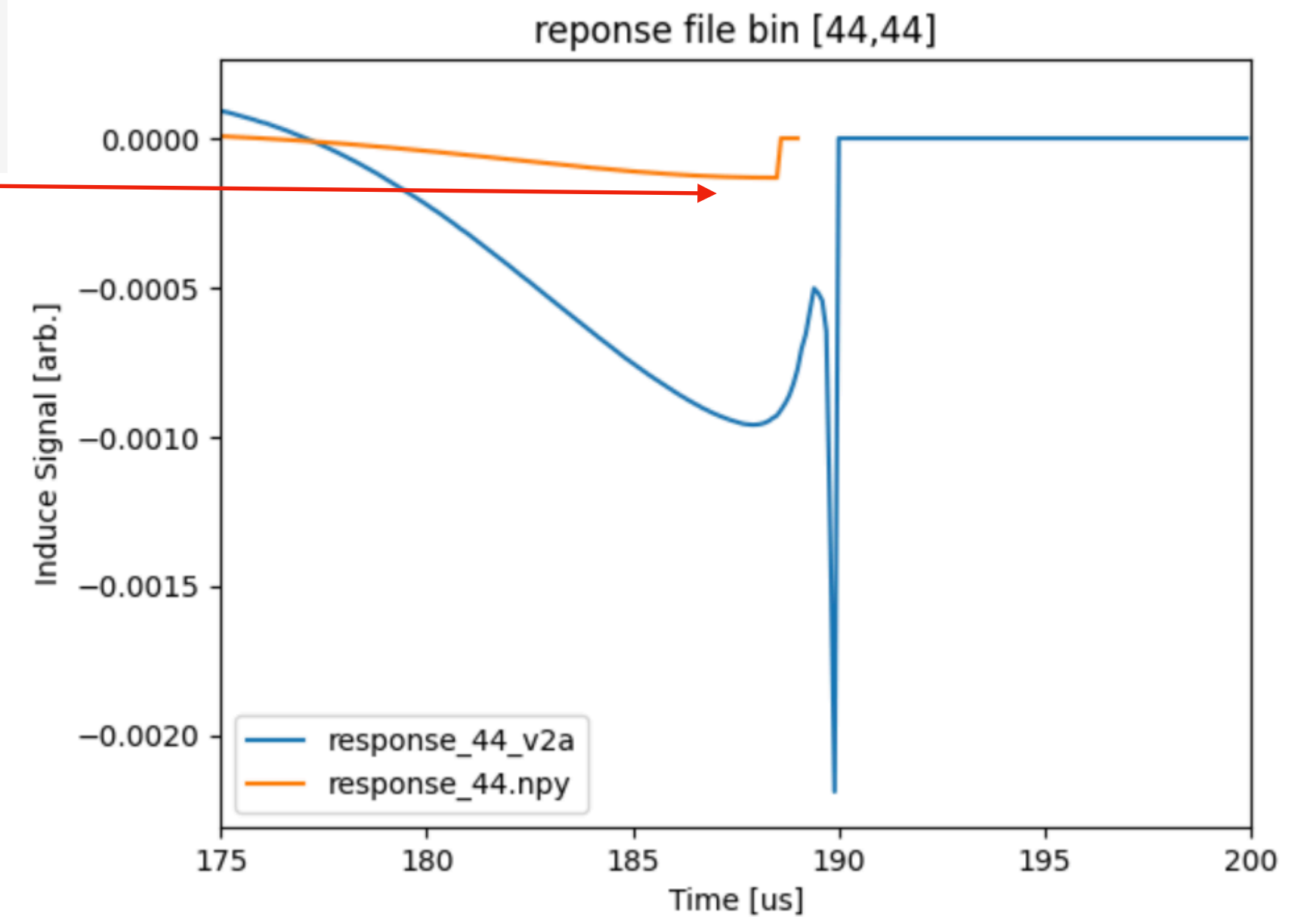
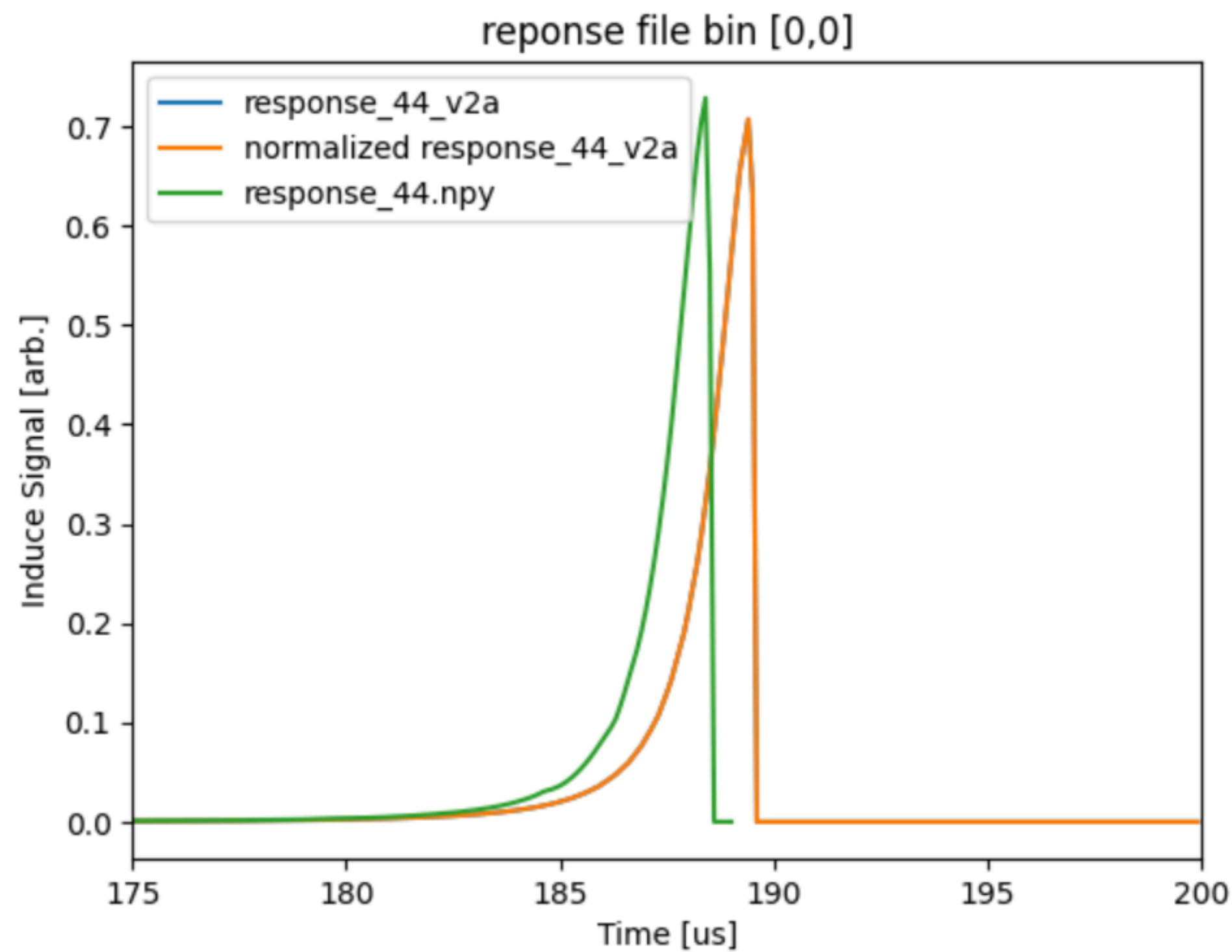
There is also some numerical error in the integration, and this is fixed post-hoc by calculating the integral and scaling or shifting the series accordingly. You should see that the first 5 rows and columns integrate to 1 (the current is calculated for a unit charge), while the rest integrate to 0. This should be very close to exact because of the correction. For collective paths (those that land on the central pixel), I integrate and divide the waveform by the integral, so if it originally has a 1% error and integrates to 0.99, we just divide by 0.99. This is not the ideal way to do it, but we prioritized getting the integral right to avoid compounding errors when you sum up many current signals, and this method leaves the shape intact. Likewise, we do an additive correction for the inductive signals (which land on non-central pixels), so that their integral is zero. This produces a weird leading edge sometimes, but these signals tend to be a few orders of magnitude smaller than the collective ones, so we ignore it for now.

0.003% diff before and after norm

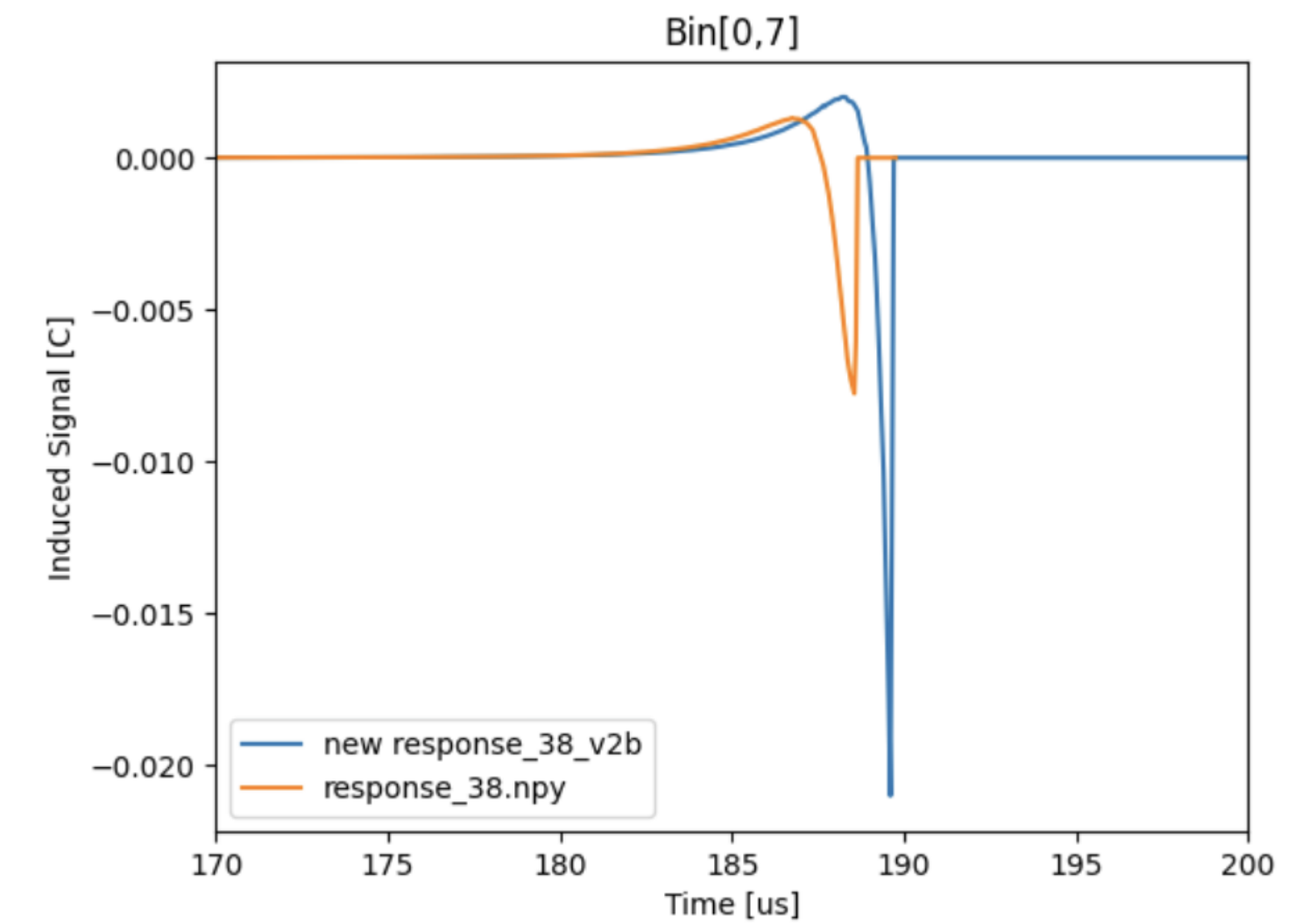
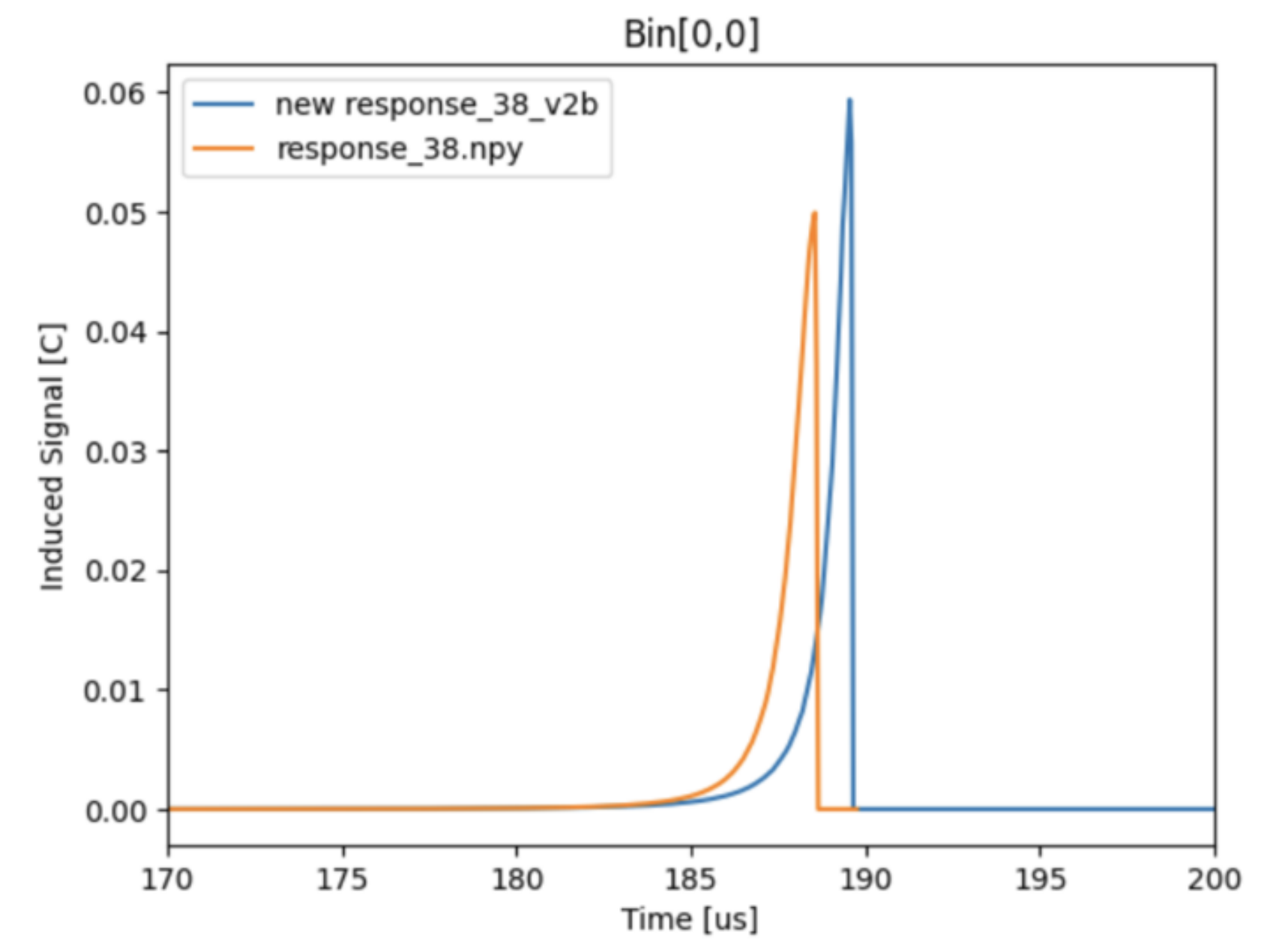
```

0 0
0.9999672940956169
After Norm: 0.9999999999999999
0 1
0.999967275455679
After Norm: 1.0
0 2
0.9999672919383521
After Norm: 1.0
0 3
0.9999673022306464
After Norm: 1.0
0 4
0.9999672858505781
After Norm: 1.0

```



- [response_44_v2b.npy](#) and [response_38_v2a.npy](#) are on land-sim GitHub
 - Not normalized
- To do
 - Compare larnd-sim/ndlar_flow with the old and new response files



Backup