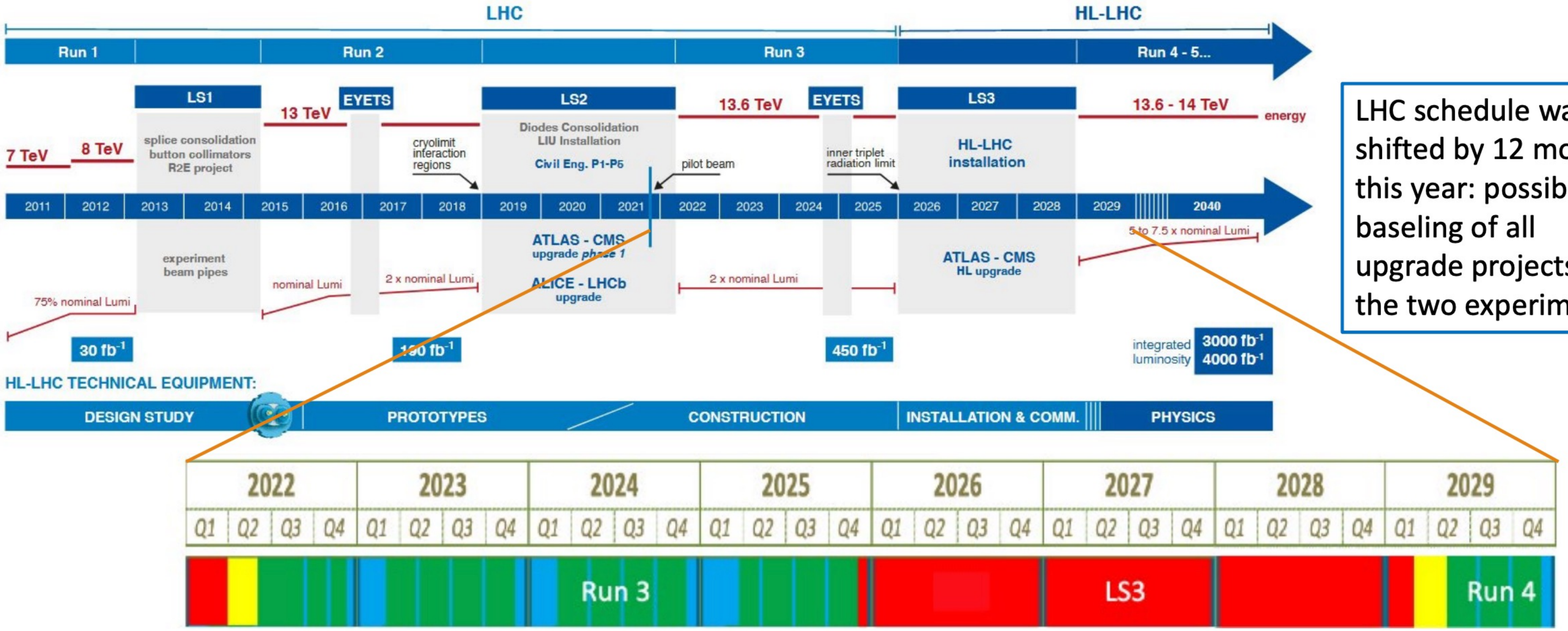
Simulated performance and calibration of CMS Phase-2 Upgrade Inner Tracker sensors

Tamas Almos VAMI [1], Morris Swartz [1]

[1] Johns Hopkins University

Pixel2022 conference

Intro about the Phase-2 upgrade



LHC schedule was shifted by 12 months this year: possible re-baseling of all upgrade projects in the two experiments

The HL-LHC will increase
* instantaneous luminosity 4X the current Run-2 value (2.0 to $7.5 \times 10^{34}\ s^{-1}cm^{-2}$)
* pile-up to 4X the current value (~55 -- > 140 to 200)
* thus the radiation damage to the detectors

Need for Phase-2 upgrade of CMS

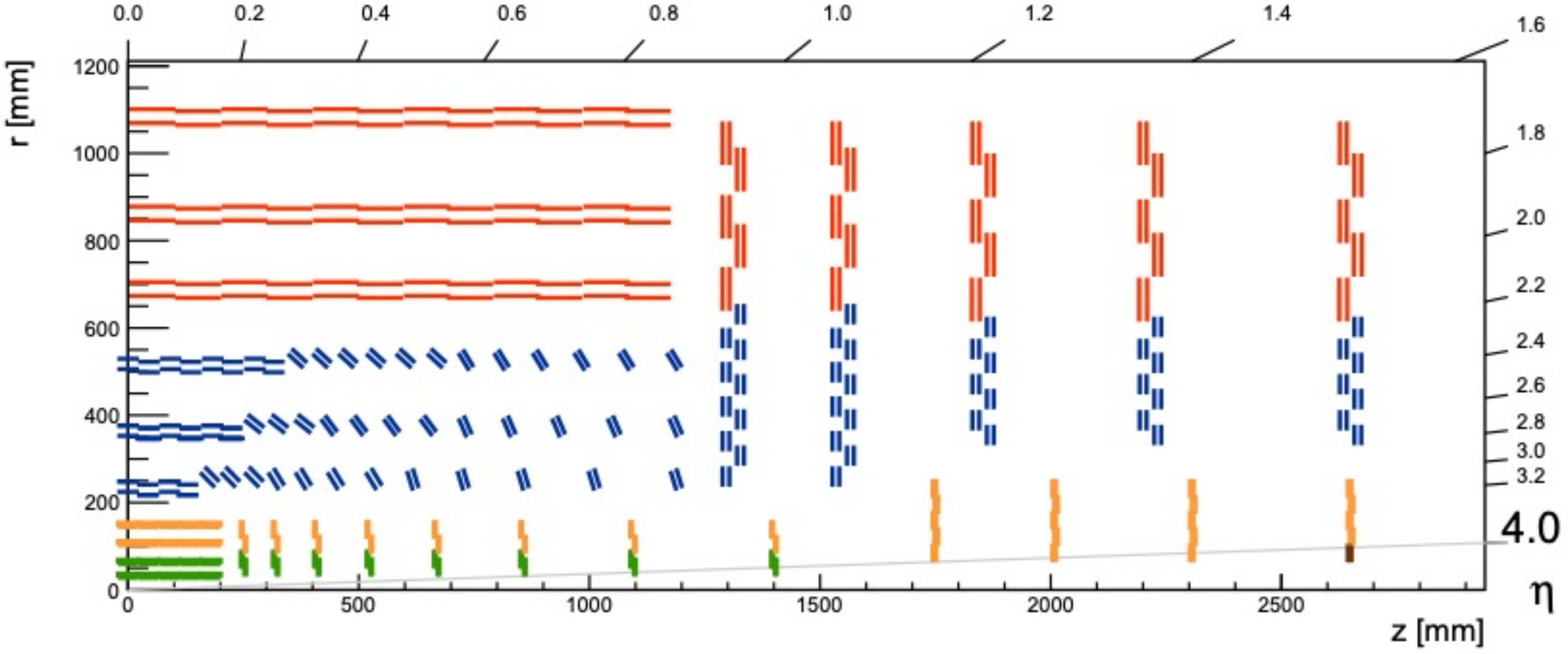
Intro about the Phase-2 upgrade

The new CMS tracker detector
* will have its acceptance increased to |eta| < 4
* low material budget (using carbon fiber mechanics, CO_2 cooling and serial power scheme)

Outer tracker (OT):
* 6 barrel layers
* 5 (x2) forward disk

Inner tracker (IT):
* 4 barrel layers
* 12 (x2) forward disk

This talk



Outline: Inner tracker sensor studies

- Sizes: 50x50 um^2 vs 25x100 um^2
- Technology: 3D vs planar sensors
- Simulation of avalanche gain effect

Using PixelAV

PixelAV is an external software to CMS Software, which can perform a more detailed simulation

Introduction to PixalAV software

Charge deposition based on Bichsel pion-Si cross-sections

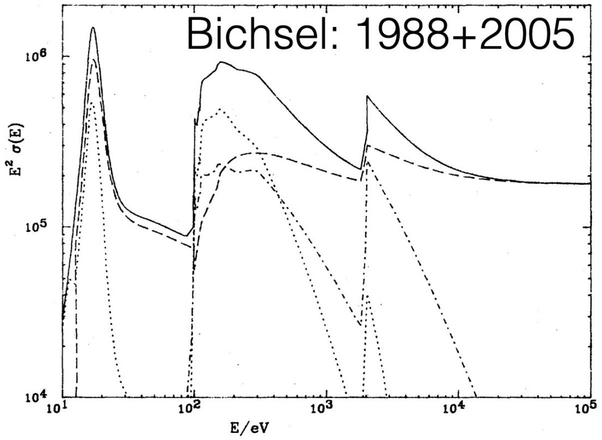
Delta-ray range using continuously slowing-down approach with NIST ESTAR dEdx data

Multiple scattering and magnetic curvature of delta-rays

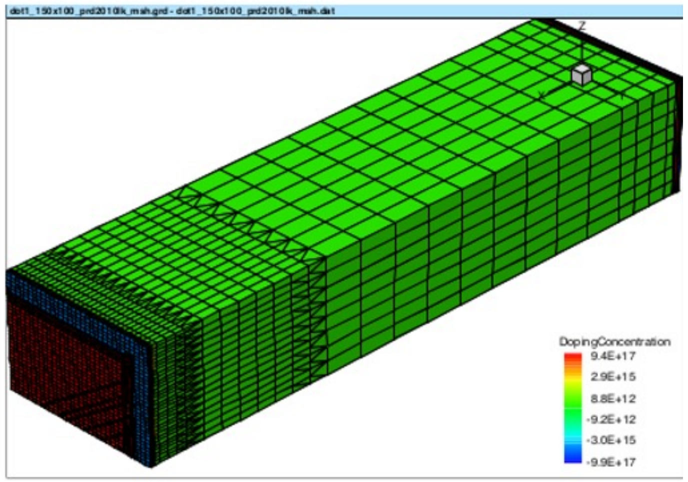
Carrier transport based on Runge-Kutta integration of saturated drift

- E-field is coming from ISE TCAD simulation of a pixel cell
- Includes charge trapping, diffusion, induction on implants

Electronics simulation: noise, linearity, thresholds, mis-calibration



$$\frac{d\vec{x}}{dt} = \vec{v} = \frac{\mu\left[q\vec{E} + \mu r_H \vec{E}\times\vec{B} + q\mu^2 r_H^2(\vec{E}\cdot\vec{B})\vec{B}\right]}{1+\mu^2 r_H^2|\vec{B}|^2}$$



Simulation of Phase-2 IT sensors

Using non-uniform E-fields (even for new sensors)

Carrier focusing at the n+ implant

Irradiation simulation based on models developed for 2018 Phase-1 detector (1e15 n_{eq}/cm^2), but scale the fluence to the expected numbers from the HL-LHC

Readout chip threshold is 1000 electrons for each cases

Cross talk with neighbors:
* 25x100 um^2 has a 0.1 crosstalk
* 50x50 um^2 has 0 crosstalk

Bias voltage:
* 25x100 um^2 start with 350 V
* 50x50 um^2 start with 100 V

Evaluation of simulations

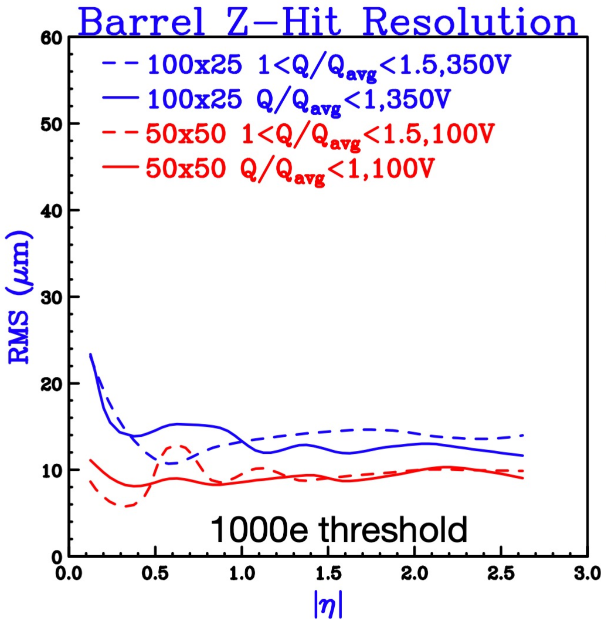
Simulation is evaluated by comparing detector resolution vs track angle

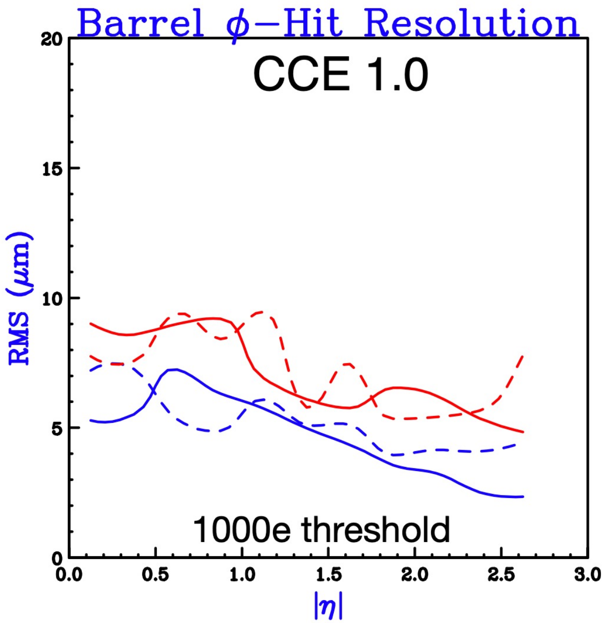
- Using the same reconstruction algorithm as the CMS Software
- Taking the RMS of (expected hit position - measured hit position)
  - The tails are important, so fitting a Gauss function is not appropriate
- This is performed in 2 charge bins:
  - $0 < Q/Q_{avg} < 1$
  - $1 < Q/Q_{avg} < 1.5$

Another important parameter is the charge collection efficiency (CCE) which is defined as the collected charge/all charges

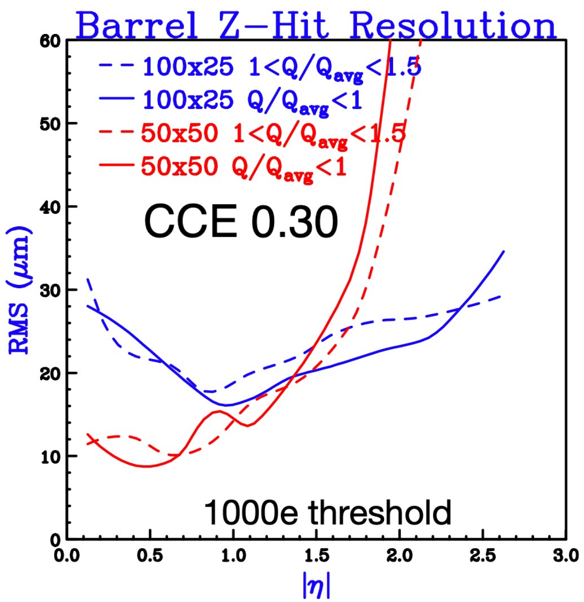
Size choice studies

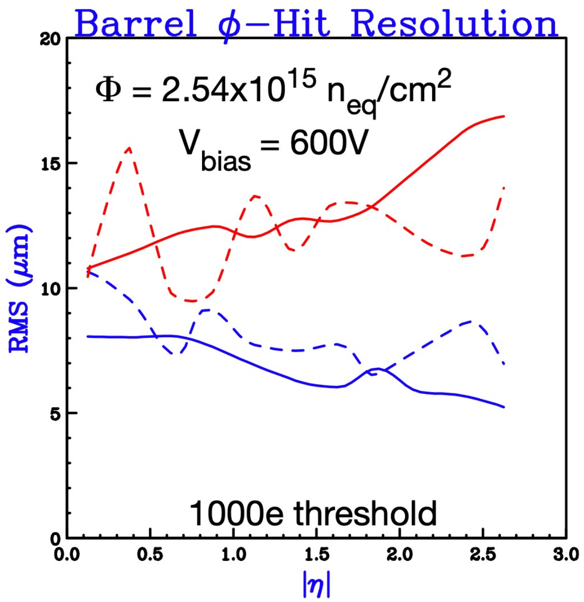
Unirradiated case





L2@3000 fb-1: 1000e threshold



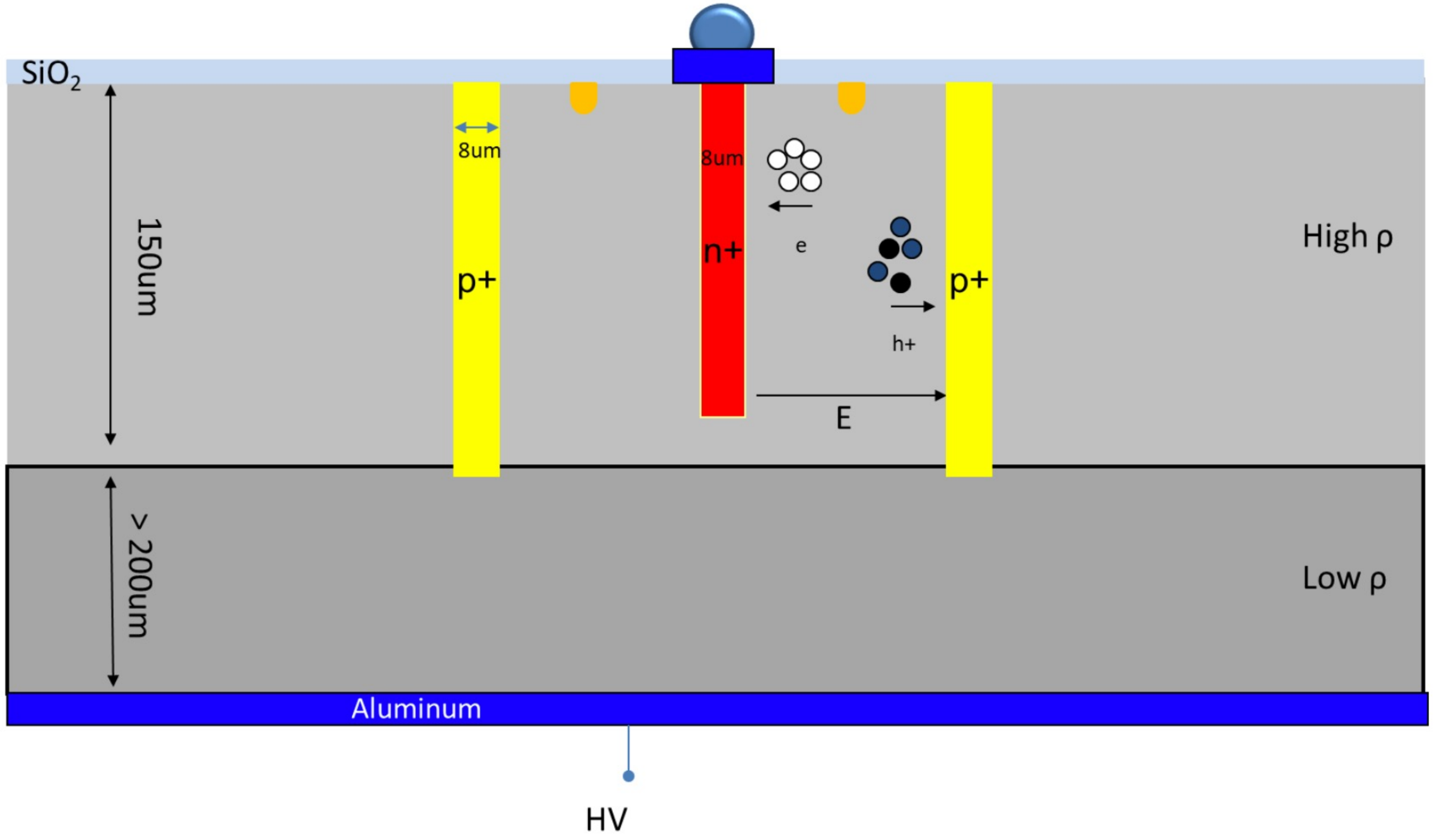


50x50 is better in z until radiation damage, after that 25x100 is better in both directions

25x100 performance on Layer-2 is similar in the end of HL-LHC as it was for the present detector in 2018

=> Decision: 25x100 um^2

Technology studies



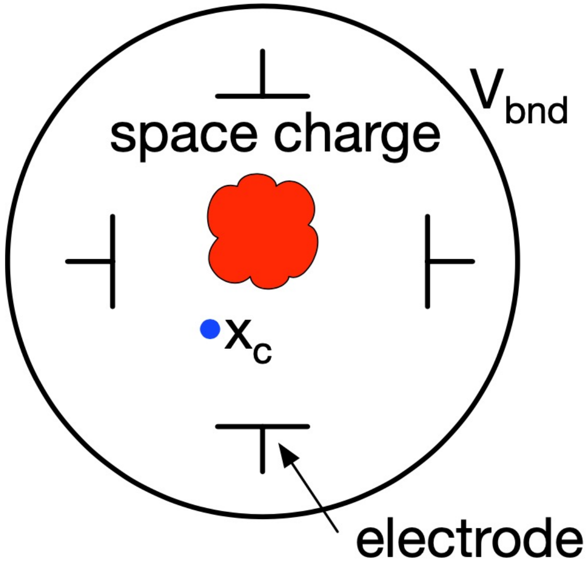
3D sensors collect charge on columnar implants that penetrate the substrate

Changes needed for PixelAV

PixelAV used a segmented parallel plate capacitor model to estimate trapped carrier induced signal --> uses symmetries that are not there in 3D sensors

Use Ramo - Shockley theorem instead

- Solve Laplace's eq for system with $V_j = V_0$ and $V_{bnd} = 0$ + all electrodes $V_i = 0$ $[i \neq j]$
- Charge on electrode j induced by carrier at x_c is $Q_j = q_c \varphi(x_c)/V_0$
  - where $\varphi(x_c)/V_0$ is the weighting potential



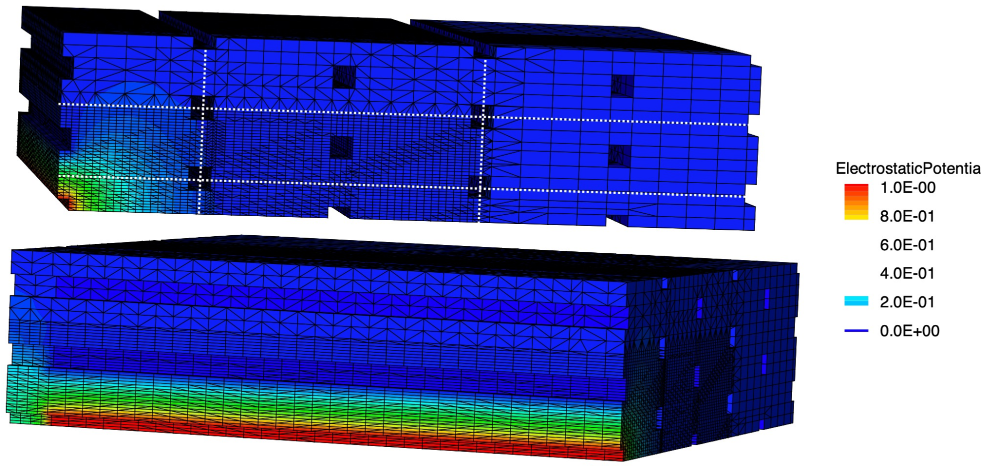
This is a general method that works for 3D sensors

Changes needed for TCAD

TCAD 9.0 does not support to mesh across region boundaries to calculate the weighting potential

- place equipotential conducting “contacts” on the inside surfaces of square voids to represent the implants in a 2.5x2.5 pixel array

Region definition for 3D case -->



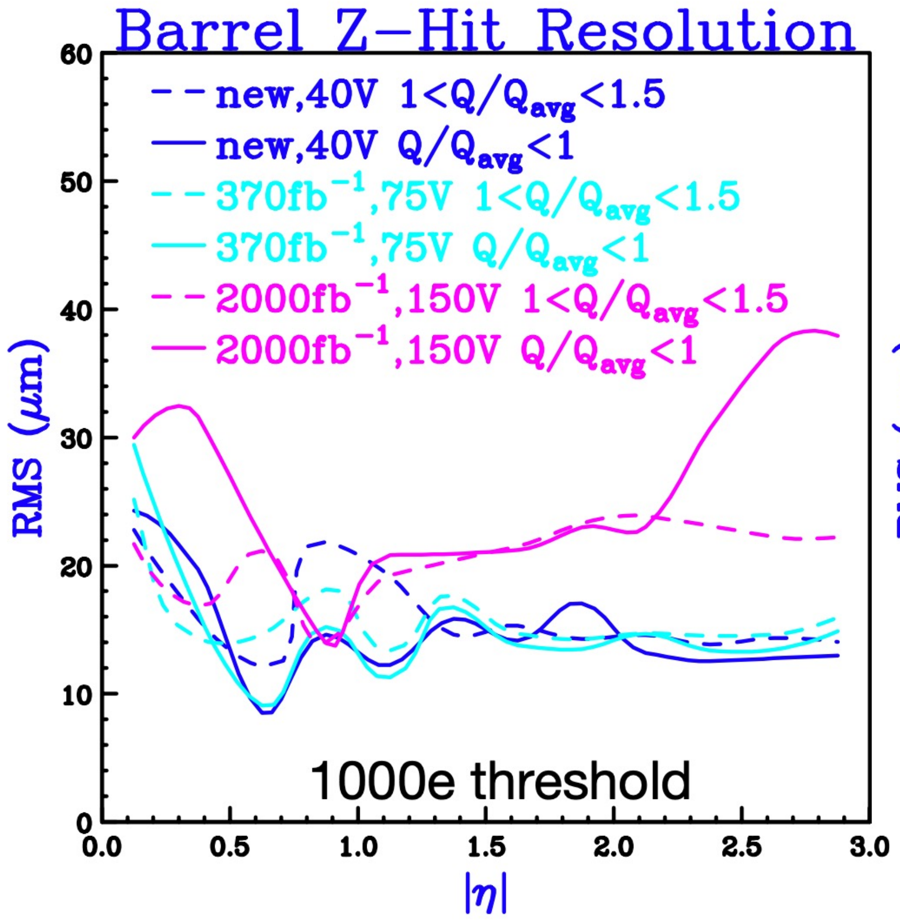
E-field 0.5x0.5 cell

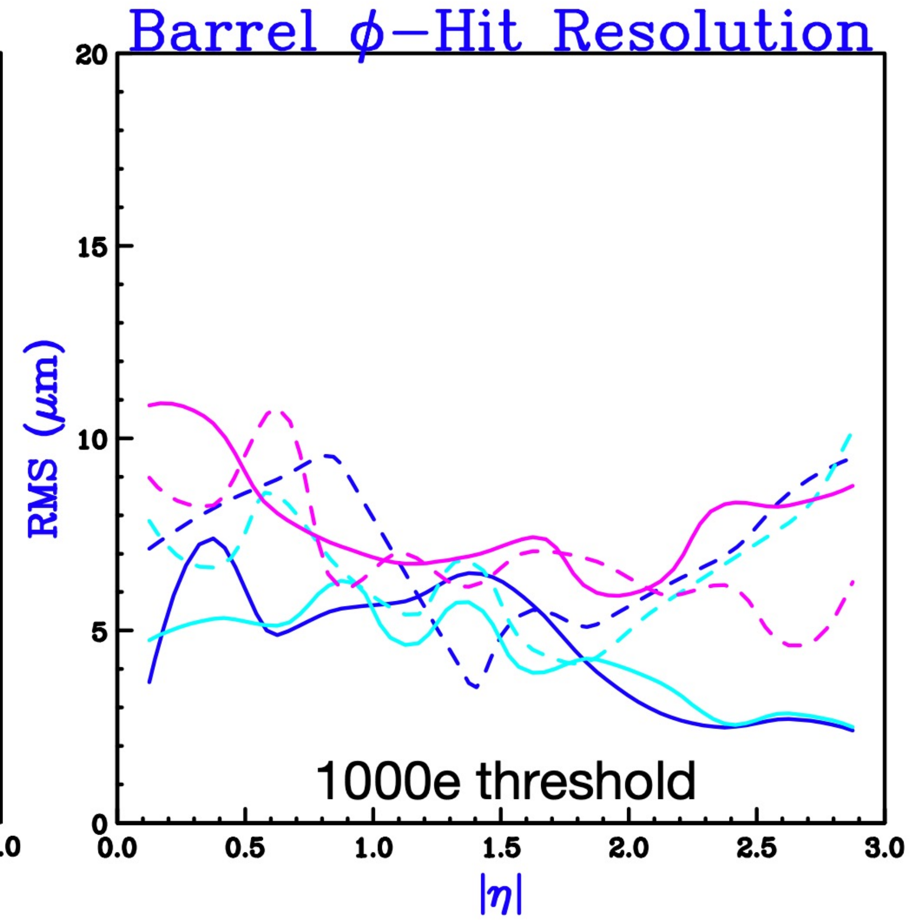
```
# Solid region 230um thick
Silicon "substrate" { cuboid [ (0,0,0) (230,50,12.5) ] }
```

Weighting potential 2.5x2.5 cell

```
# Solid region 230um thick
Oxide "substrate" { polyhedron { polygon [ (230.0,96.0,0.0) (230.0,96.0,4.0) (230.0,104.0,4.0) (230.0,104.0,0.0) (230.0,196.0,0.0) (230.0,196.0,4.0) (230.0,204.0,4.0)
(230.0,204.0,0.0) (230.0,250.0,0.0) (230.0,250.0,8.5) (230.0,246.0,8.5) (230.0,246.0,12.5) (230.0,154.0,12.5) (230.0,154.0,8.5) (230.0,146.0,8.5) (230.0,146.0,16.5)
(230.0,154.0,16.5) (230.0,154.0,12.5) (230.0,246.0,12.5) (230.0,246.0,16.5) (230.0,250.0,16.5) (230.0,250.0,25.0) (230.0,204.0,25.0) (230.0,204.0,21.0)
(230.0,196.0,21.0) (230.0,196.0,29.0) (230.0,204.0,29.0) (230.0,204.0,25.0) (230.0,250.0,25.0) (230.0,250.0,33.5) (230.0,246.0,33.5) (230.0,246.0,37.5)
(230.0,154.0,37.5) (230.0,154.0,33.5) (230.0,146.0,33.5) (230.0,146.0,41.5) (230.0,154.0,41.5) (230.0,154.0,37.5) (230.0,246.0,37.5) (230.0,246.0,41.5)
(230.0,250.0,41.5) (230.0,250.0,50.0) (230.0,204.0,50.0) (230.0,204.0,46.0) (230.0,196.0,46.0) (230.0,196.0,54.0) (230.0,204.0,54.0) (230.0,204.0,50.0)
(230.0,250.0,50.0) (230.0,250.0,58.5) (230.0,246.0,58.5) (230.0,246.0,62.5) (230.0,154.0,62.5) (230.0,154.0,58.5) (230.0,146.0,58.5) (230.0,146.0,62.5)
(230.0,54.0,62.5) (230.0,54.0,58.5) (230.0,46.0,58.5) (230.0,46.0,62.5) (230.0,0.0,62.5) (230.0,0.0,54.0) (230.0,4.0,54.0) (230.0,4.0,50.0) (230.0,96.0,50.0)
(230.0,96.0,54.0) (230.0,104.0,54.0) (230.0,104.0,46.0) (230.0,96.0,46.0) (230.0,96.0,50.0) (230.0,4.0,50.0) (230.0,4.0,46.0) (230.0,0.0,46.0) (230.0,0.0,37.5)
(230.0,46.0,37.5) (230.0,46.0,41.5) (230.0,54.0,41.5) (230.0,54.0,33.5) (230.0,46.0,33.5) (230.0,46.0,37.5) (230.0,0.0,37.5) (230.0,0.0,29.0) (230.0,4.0,29.0)
(230.0,4.0,25.0) (230.0,96.0,25.0) (230.0,96.0,29.0) (230.0,104.0,29.0) (230.0,104.0,21.0) (230.0,96.0,21.0) (230.0,96.0,25.0) (230.0,4.0,25.0) (230.0,4.0,21.0)
(230.0,0.0,21.0) (230.0,0.0,12.5) (230.0,46.0,12.5) (230.0,46.0,16.5) (230.0,54.0,16.5) (230.0,54.0,8.5) (230.0,46.0,8.5) (230.0,46.0,12.5) (230.0,0.0,12.5)
(230.0,0.0,4.0) (230.0,4.0,4.0) (230.0,4.0,0.0) ] rectangle[(0.0,96.0,0.0) (230.0,96.0,4.0)] rectangle[(0.0,96.0,4.0) (230.0,104.0,4.0)] rectangle[(0.0,104.0,4.0)
(230.0,104.0,0.0)] rectangle[(0.0,104.0,0.0) (230.0,196.0,0.0)] rectangle[(0.0,196.0,0.0) (230.0,196.0,4.0)] rectangle[(0.0,196.0,4.0) (230.0,204.0,4.0)]
rectangle[(0.0,204.0,4.0) (230.0,204.0,0.0)] rectangle[(0.0,204.0,0.0) (230.0,250.0,0.0)] rectangle[(0.0,250.0,0.0) (230.0,250.0,8.5)] rectangle[(0.0,250.0,8.5)
(230.0,246.0,8.5)] rectangle[(0.0,246.0,8.5) (230.0,246.0,12.5)] rectangle[(0.0,154.0,12.5) (230.0,154.0,8.5)] rectangle[(0.0,154.0,8.5) (230.0,146.0,8.5)]
rectangle[(0.0,146.0,8.5) (230.0,146.0,16.5)] rectangle[(0.0,146.0,16.5) (230.0,154.0,16.5)] rectangle[(0.0,154.0,16.5) (230.0,154.0,12.5)] rectangle[(0.0,246.0,12.5)
(230.0,246.0,16.5)] rectangle[(0.0,246.0,16.5) (230.0,250.0,16.5)] rectangle[(0.0,250.0,16.5) (230.0,250.0,25.0)] rectangle[(0.0,204.0,25.0) (230.0,204.0,21.0)]
rectangle[(0.0,204.0,21.0) (230.0,196.0,21.0)] rectangle[(0.0,196.0,21.0) (230.0,196.0,29.0)] rectangle[(0.0,196.0,29.0) (230.0,204.0,29.0)]
rectangle[(0.0,204.0,29.0) (230.0,204.0,25.0)] rectangle[(0.0,250.0,25.0) (230.0,250.0,33.5)] rectangle[(0.0,250.0,33.5) (230.0,246.0,33.5)]
rectangle[(0.0,246.0,33.5) (230.0,246.0,37.5)] rectangle[(0.0,154.0,37.5) (230.0,154.0,33.5)] rectangle[(0.0,154.0,33.5) (230.0,146.0,33.5)]
rectangle[(0.0,146.0,33.5) (230.0,146.0,41.5)] rectangle[(0.0,146.0,41.5) (230.0,154.0,41.5)] rectangle[(0.0,154.0,41.5) (230.0,154.0,37.5)]
rectangle[(0.0,246.0,37.5) (230.0,246.0,41.5)] rectangle[(0.0,246.0,41.5) (230.0,250.0,41.5)] rectangle[(0.0,250.0,41.5) (230.0,250.0,50.0)]
rectangle[(0.0,204.0,50.0) (230.0,204.0,46.0)] rectangle[(0.0,204.0,46.0) (230.0,196.0,46.0)] rectangle[(0.0,196.0,46.0) (230.0,196.0,54.0)]
rectangle[(0.0,196.0,54.0) (230.0,204.0,54.0)] rectangle[(0.0,204.0,54.0) (230.0,204.0,50.0)] rectangle[(0.0,250.0,50.0) (230.0,250.0,58.5)]
rectangle[(0.0,250.0,58.5) (230.0,246.0,58.5)] rectangle[(0.0,246.0,58.5) (230.0,246.0,62.5)] rectangle[(0.0,246.0,62.5) (230.0,154.0,62.5)]
rectangle[(0.0,154.0,62.5) (230.0,154.0,58.5)] rectangle[(0.0,154.0,58.5) (230.0,146.0,58.5)] rectangle[(0.0,146.0,58.5) (230.0,146.0,62.5)]
rectangle[(0.0,146.0,62.5) (230.0,54.0,62.5)] rectangle[(0.0,54.0,62.5) (230.0,54.0,58.5)] rectangle[(0.0,54.0,58.5) (230.0,46.0,58.5)] rectangle[(0.0,46.0,58.5)
(230.0,46.0,62.5)] rectangle[(0.0,46.0,62.5) (230.0,0.0,62.5)] rectangle[(0.0,0.0,62.5) (230.0,0.0,54.0)] rectangle[(0.0,0.0,54.0) (230.0,4.0,54.0)]
rectangle[(0.0,4.0,54.0) (230.0,4.0,50.0)] rectangle[(0.0,96.0,50.0) (230.0,96.0,54.0)] rectangle[(0.0,96.0,54.0) (230.0,104.0,54.0)] rectangle[(0.0,104.0,54.0)
(230.0,104.0,46.0)] rectangle[(0.0,104.0,46.0) (230.0,96.0,46.0)] rectangle[(0.0,96.0,46.0) (230.0,96.0,50.0)] rectangle[(0.0,4.0,50.0) (230.0,4.0,46.0)]
rectangle[(0.0,4.0,46.0) (230.0,0.0,46.0)] rectangle[(0.0,0.0,46.0) (230.0,0.0,37.5)] rectangle[(0.0,46.0,37.5) (230.0,46.0,41.5)] rectangle[(0.0,46.0,41.5)
(230.0,54.0,41.5)] rectangle[(0.0,54.0,41.5) (230.0,54.0,33.5)] rectangle[(0.0,54.0,33.5) (230.0,46.0,33.5)] rectangle[(0.0,46.0,33.5) (230.0,46.0,37.5)]
rectangle[(0.0,0.0,37.5) (230.0,0.0,29.0)] rectangle[(0.0,0.0,29.0) (230.0,4.0,29.0)] rectangle[(0.0,4.0,29.0) (230.0,4.0,25.0)] rectangle[(0.0,96.0,25.0)
(230.0,96.0,29.0)] rectangle[(0.0,96.0,29.0) (230.0,104.0,29.0)] rectangle[(0.0,104.0,29.0) (230.0,104.0,21.0)] rectangle[(0.0,104.0,21.0) (230.0,96.0,21.0)]
rectangle[(0.0,96.0,21.0) (230.0,96.0,25.0)] rectangle[(0.0,4.0,25.0) (230.0,4.0,21.0)] rectangle[(0.0,4.0,21.0) (230.0,0.0,21.0)] rectangle[(0.0,0.0,21.0)
(230.0,0.0,12.5)] rectangle[(0.0,46.0,12.5) (230.0,46.0,16.5)] rectangle[(0.0,46.0,16.5) (230.0,54.0,16.5)] rectangle[(0.0,54.0,16.5) (230.0,54.0,8.5)]
rectangle[(0.0,54.0,8.5) (230.0,46.0,8.5)] rectangle[(0.0,46.0,8.5) (230.0,46.0,12.5)] rectangle[(0.0,0.0,12.5) (230.0,0.0,4.0)] rectangle[(0.0,0.0,4.0) (230.0,4.0,4.0)]
rectangle[(0.0,4.0,4.0) (230.0,4.0,0.0)] rectangle[(0.0,4.0,0.0) (230.0,96.0,0.0)] polygon [ (0.0,4.0,0.0) (0.0,4.0,4.0) (0.0,0.0,4.0) (0.0,0.0,12.5) (0.0,46.0,12.5)
(0.0,46.0,8.5) (0.0,54.0,8.5) (0.0,54.0,16.5) (0.0,46.0,16.5) (0.0,46.0,12.5) (0.0,0.0,12.5) (0.0,0.0,21.0) (0.0,4.0,21.0) (0.0,4.0,25.0) (0.0,96.0,25.0) (0.0,96.0,21.0)
(0.0,104.0,21.0) (0.0,104.0,29.0) (0.0,96.0,29.0) (0.0,96.0,25.0) (0.0,4.0,25.0) (0.0,4.0,29.0) (0.0,0.0,29.0) (0.0,0.0,37.5) (0.0,46.0,37.5) (0.0,46.0,33.5) (0.0,54.0,33.5)
(0.0,54.0,41.5) (0.0,46.0,41.5) (0.0,46.0,37.5) (0.0,0.0,37.5) (0.0,0.0,46.0) (0.0,4.0,46.0) (0.0,4.0,50.0) (0.0,96.0,50.0) (0.0,96.0,46.0) (0.0,104.0,46.0) (0.0,104.0,54.0)
(0.0,96.0,54.0) (0.0,96.0,50.0) (0.0,4.0,50.0) (0.0,4.0,54.0) (0.0,0.0,54.0) (0.0,0.0,62.5) (0.0,46.0,62.5) (0.0,46.0,58.5) (0.0,54.0,58.5) (0.0,54.0,62.5) (0.0,146.0,62.5)
(0.0,146.0,58.5) (0.0,154.0,58.5) (0.0,154.0,62.5) (0.0,246.0,62.5) (0.0,246.0,58.5) (0.0,250.0,58.5) (0.0,250.0,50.0) (0.0,204.0,50.0) (0.0,204.0,54.0) (0.0,196.0,54.0)
(0.0,196.0,46.0) (0.0,204.0,46.0) (0.0,204.0,50.0) (0.0,250.0,50.0) (0.0,250.0,41.5) (0.0,246.0,41.5) (0.0,246.0,37.5) (0.0,154.0,37.5) (0.0,154.0,41.5) (0.0,146.0,41.5)
(0.0,146.0,33.5) (0.0,154.0,33.5) (0.0,154.0,37.5) (0.0,246.0,37.5) (0.0,246.0,33.5) (0.0,250.0,33.5) (0.0,250.0,25.0) (0.0,204.0,25.0) (0.0,204.0,29.0) (0.0,196.0,29.0)
(0.0,196.0,21.0) (0.0,204.0,21.0) (0.0,204.0,25.0) (0.0,250.0,25.0) (0.0,250.0,16.5) (0.0,246.0,16.5) (0.0,246.0,12.5) (0.0,154.0,12.5) (0.0,154.0,16.5) (0.0,146.0,16.5)
(0.0,146.0,8.5) (0.0,154.0,8.5) (0.0,154.0,12.5) (0.0,246.0,12.5) (0.0,246.0,8.5) (0.0,250.0,8.5) (0.0,250.0,0.0) (0.0,204.0,0.0) (0.0,204.0,4.0) (0.0,196.0,4.0)
(0.0,196.0,0.0) (0.0,104.0,0.0) (0.0,104.0,4.0) (0.0,96.0,4.0) (0.0,96.0,0.0) ] }
}
```

Technology studies





Sensors after 370 fb^{-1} perform similarly to new sensors

At 2000 fb^{-1} resolutions with 150 V are showing the effect of charge loss

Technology studies

Scenario	new	370/fb	2000/fb	2000/fb
Fluence	0 n_{eq}/cm^2	3e15 n_{eq}/cm^2	17e15 n_{eq}/cm^2	17e15 n_{eq}/cm^2
Bias	40 V	75 V	100 V	150 V
Resolution (x/y)	5.6 / 13.9 um	5.9 / 14.3 um	10.9 / 27.5 um	9.8 / 22.5 um
CCE	0.96	0.84	0.32	0.39

2000 fb^{-1} really needs 150V, otherwise significant cluster breakage is observed

3D sensors have great performance at high irradiation (comparable to current detector performance)

=> Decision: Use 3D in L1

Simulation of avalanche gain effect

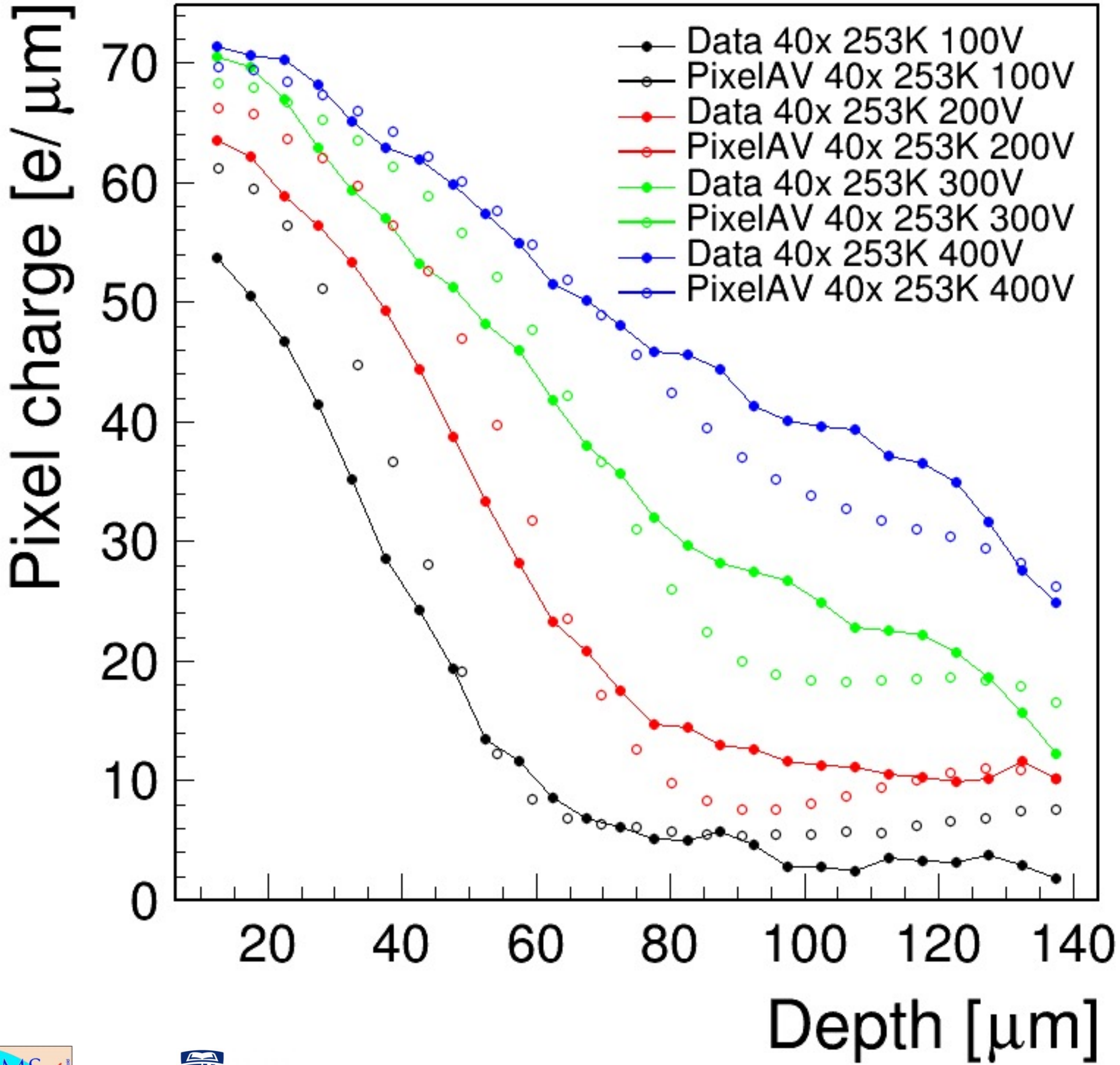
The avalanche gain effect is non-negligible for high HV values for the Phase-2 planar sensors, so we should simulate it correctly

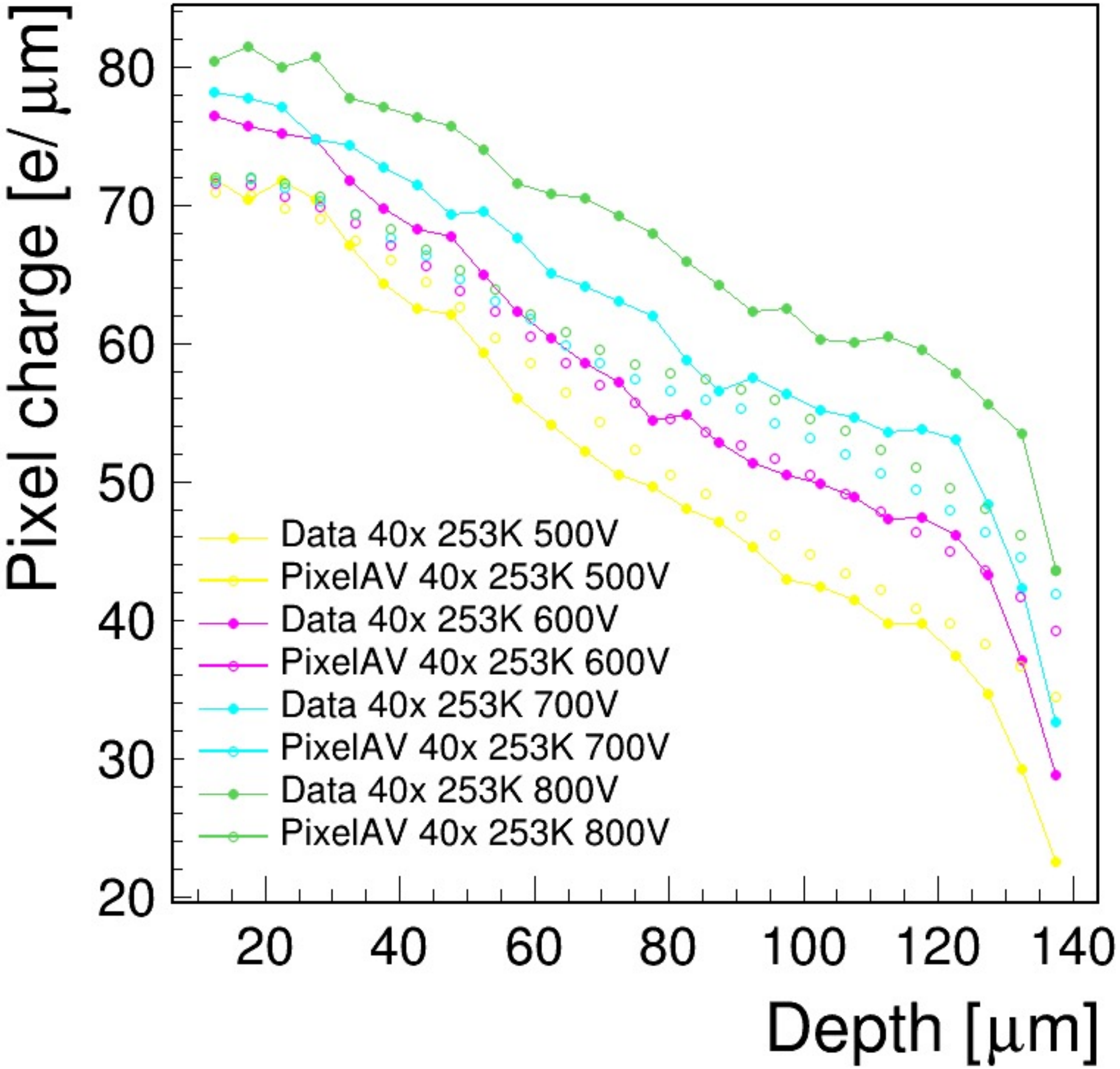
Compare test beam data from DESY to PixelAV simulations

- Test beam data with irradiation of 4.0E15 n_{eq}/cm^2 data (denoted as "Data 40x")
- PixelAV with the same procedure as earlier: by rescaling the 2018-based simulation (this will be denoted as "PixelAV 40x")

Data contains several HV setting, so for each point a new simulation was created with the appropriate HV value, but same temperature (253K), threshold, etc

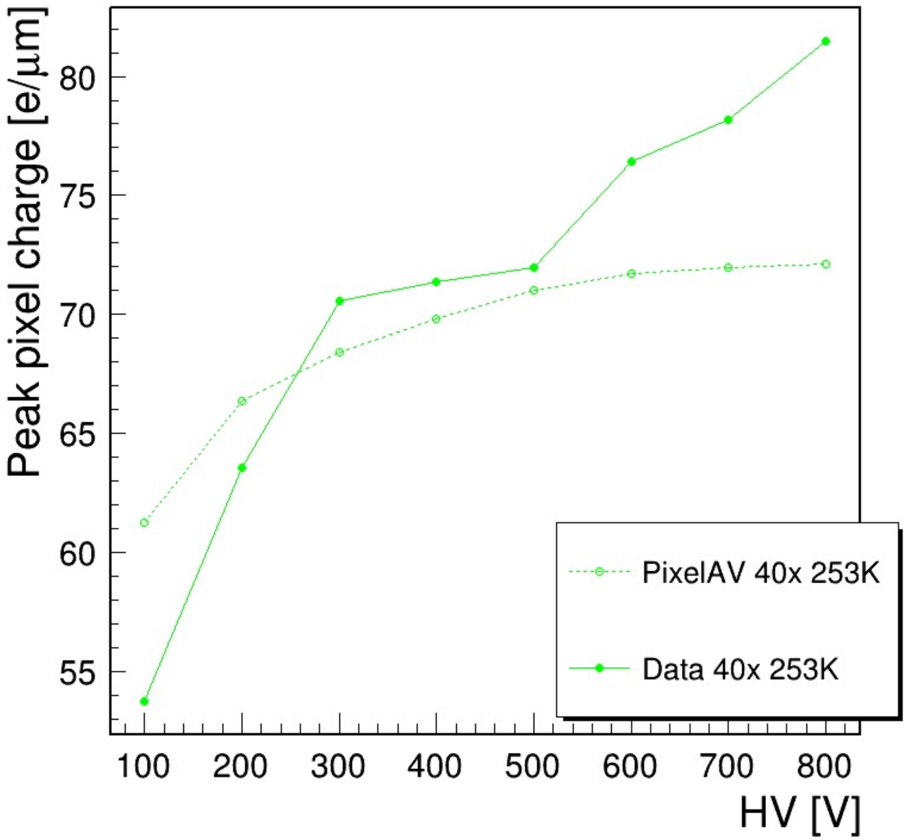
Charge profiles (data comparison)



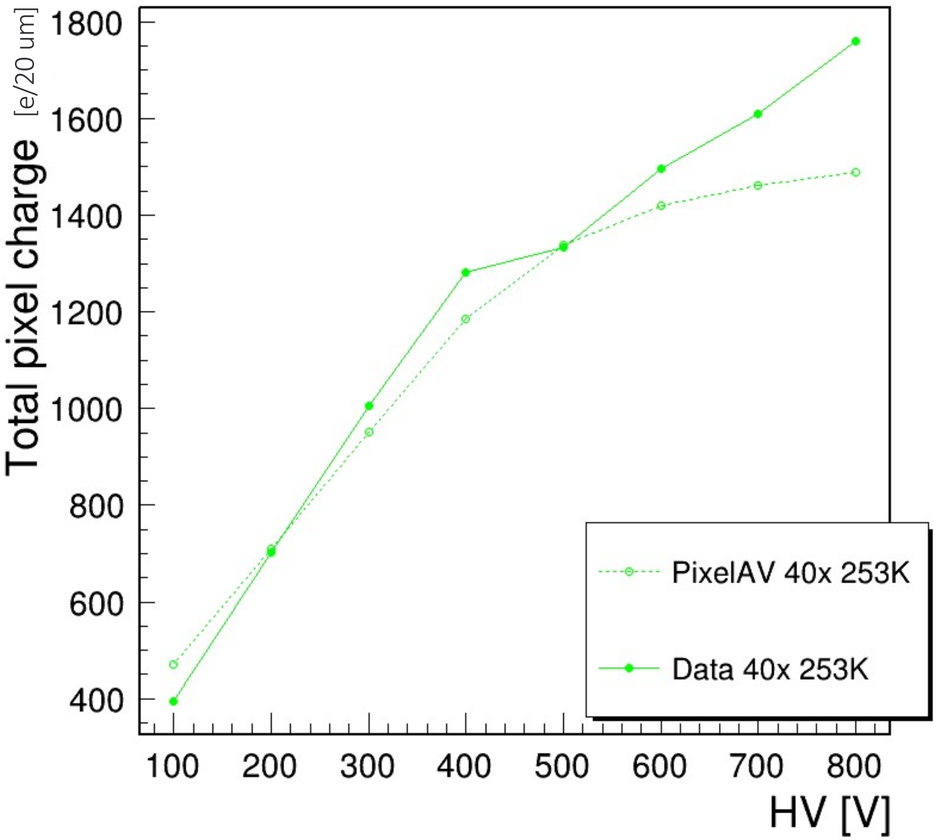


Peak and total charge comparisons

Max of the previous histograms



Integral of the previous histograms



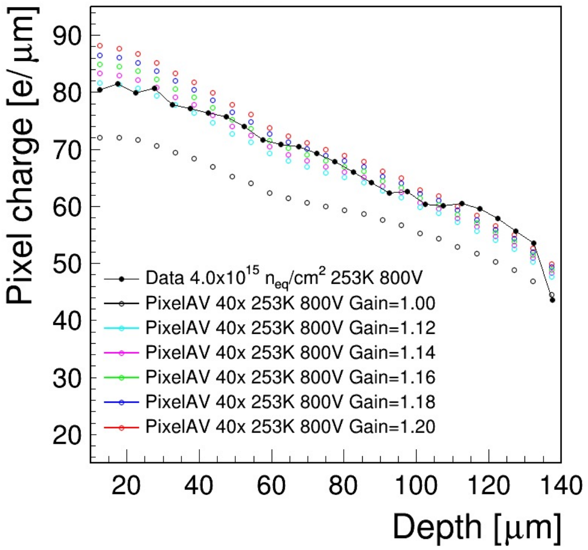
PixelAV is good in describing both quantities for low HV.
For high HV we have the avalanche effect -- to be simulated

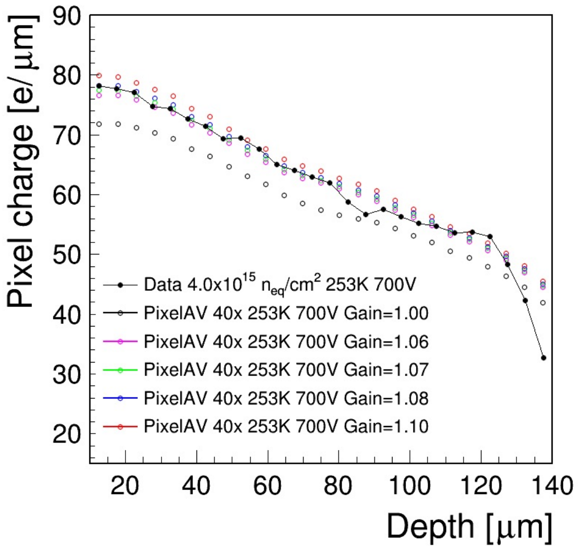
First simulations of the gain factor in PixelAV

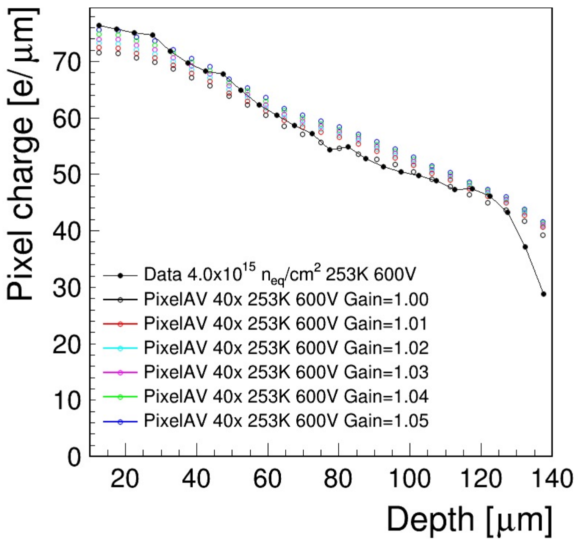
Change PixelAV to include a gain factor as an external parameter when collecting electrons only [the induced charge from trapped carriers would not experience any gain]

Testing the code with gain = 1 leads to identical results to vanilla pixelAV

Make a scan for each samples with a gain factor variation, then choose the one that describes the data the most







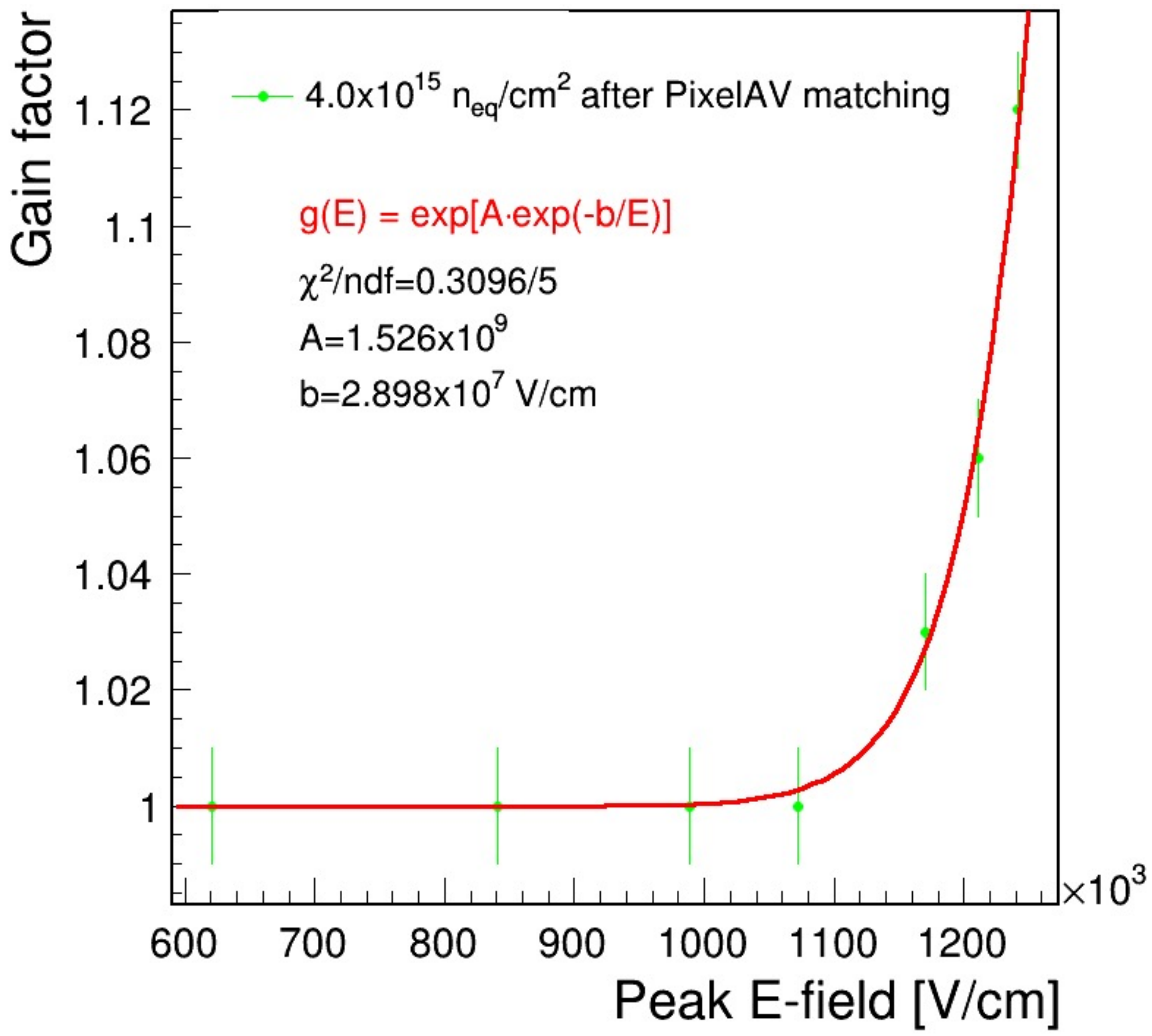
Looking at the factors vs peak E-field

For V<600 PixelAV seems to be ok with the default (gain=1.0)

I also gave all values an error of 0.01

Literature suggest the function g(E), where $A \cdot \exp\left(-\frac{b}{E}\right)$ is the coefficient of the impact ionization for electrons/holes, b is the parameter for the breakdown E-field

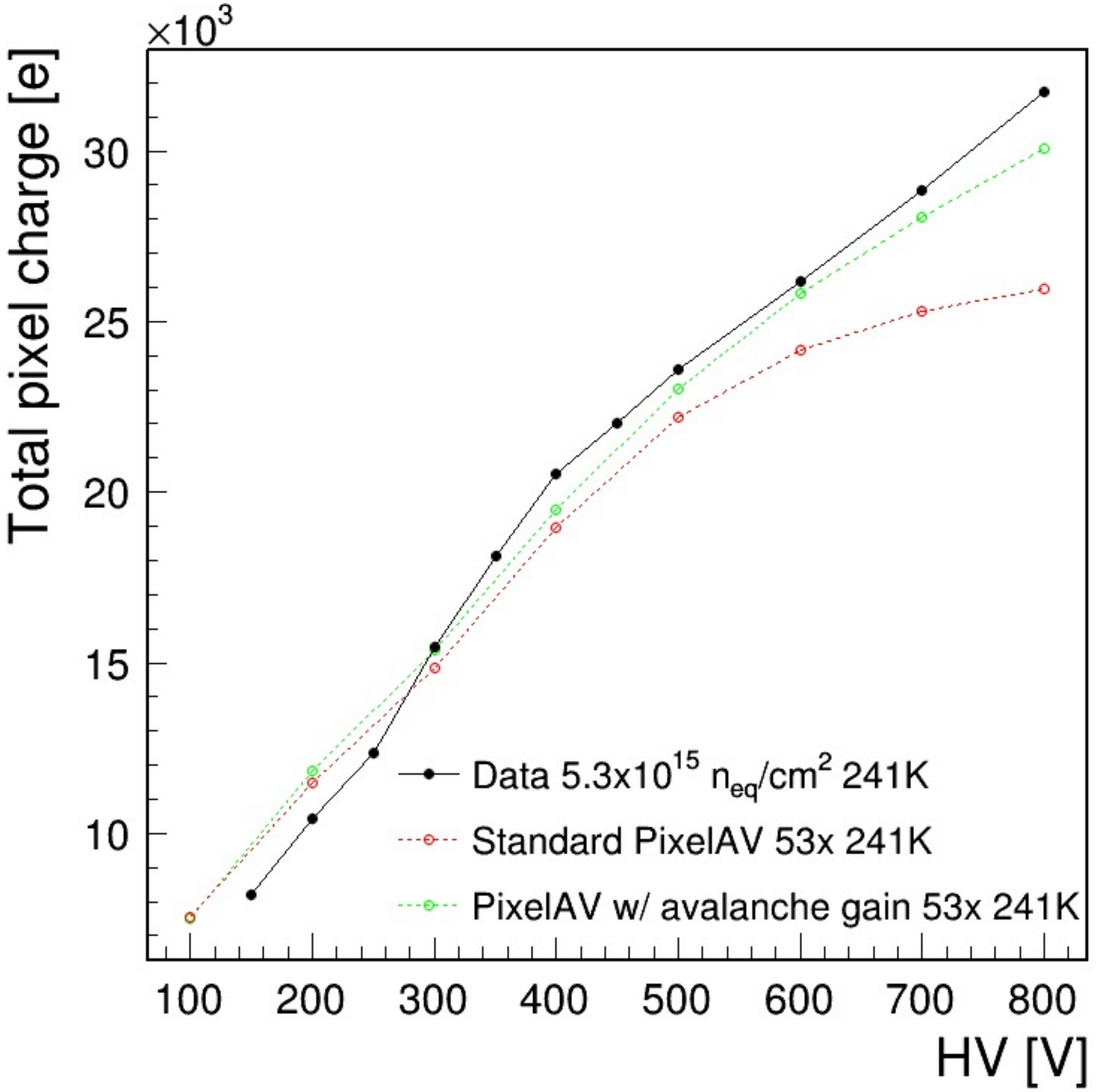
Serezhkin, Y.N., Shesterkina, A.A. Carrier multiplication in silicon *P-N* junctions. *Semiconductors* **37,** 1085–1089 (2003). https://doi.org/10.1134/1.1610124



Validation with 5.3E15 data

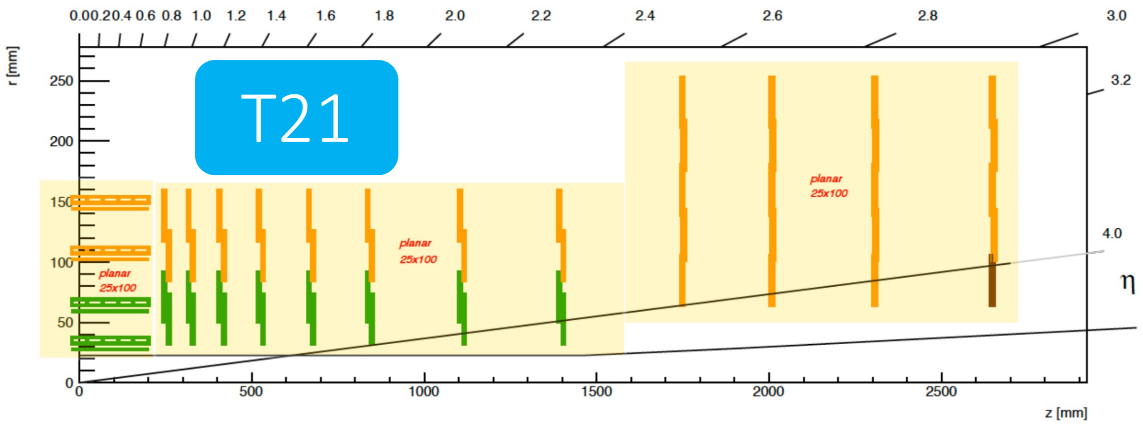
As a validation of the procedure, I used test beam data with irradiation of 5.3E15 n_{eq}/cm^2 and run simulations in which I included the avalanche gain effect.

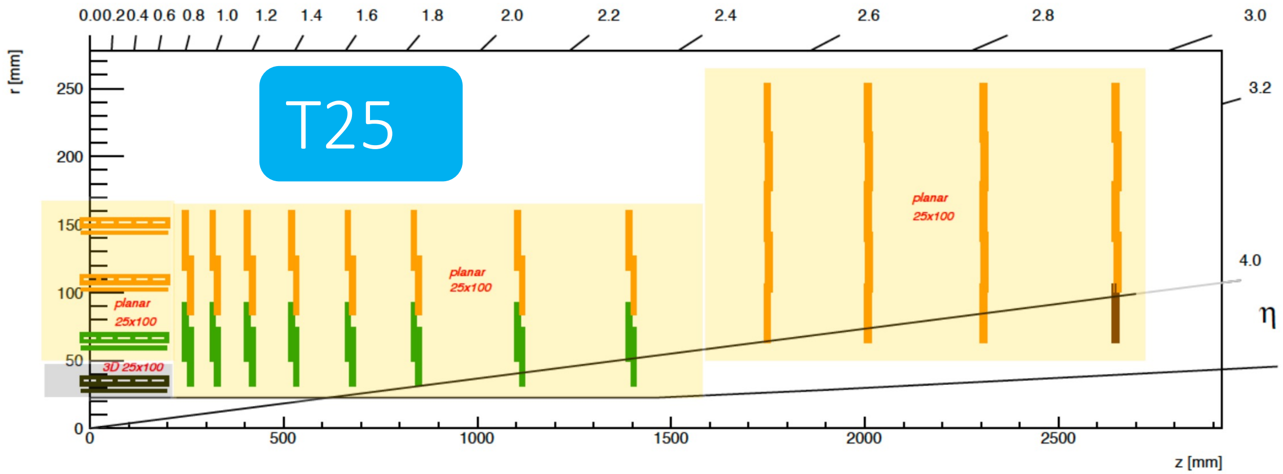
PixelAV with avalanche gain effect is better at describing the test beam data



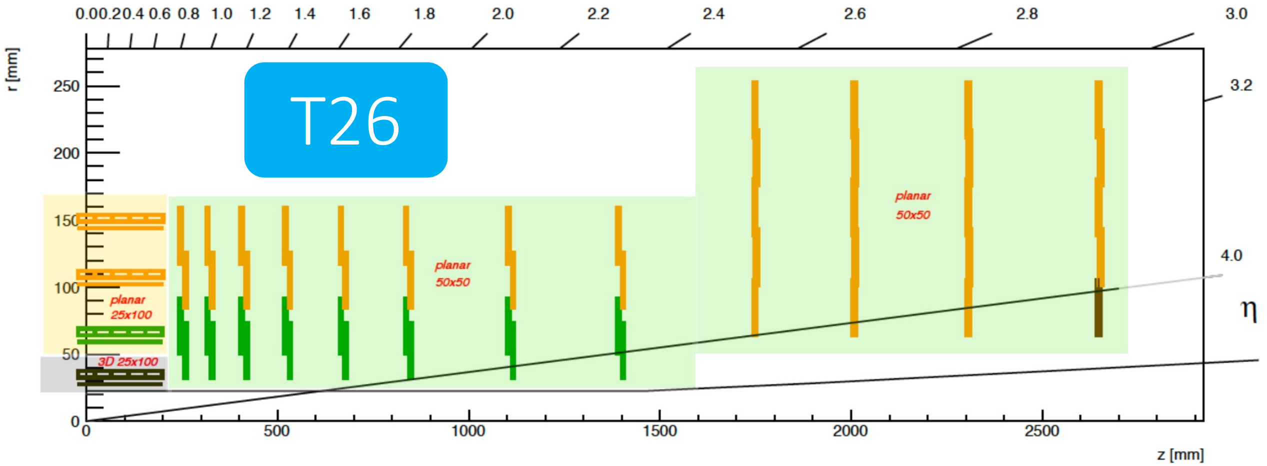
Let’s use CMS (CMSSW) full simulations!

Different layouts





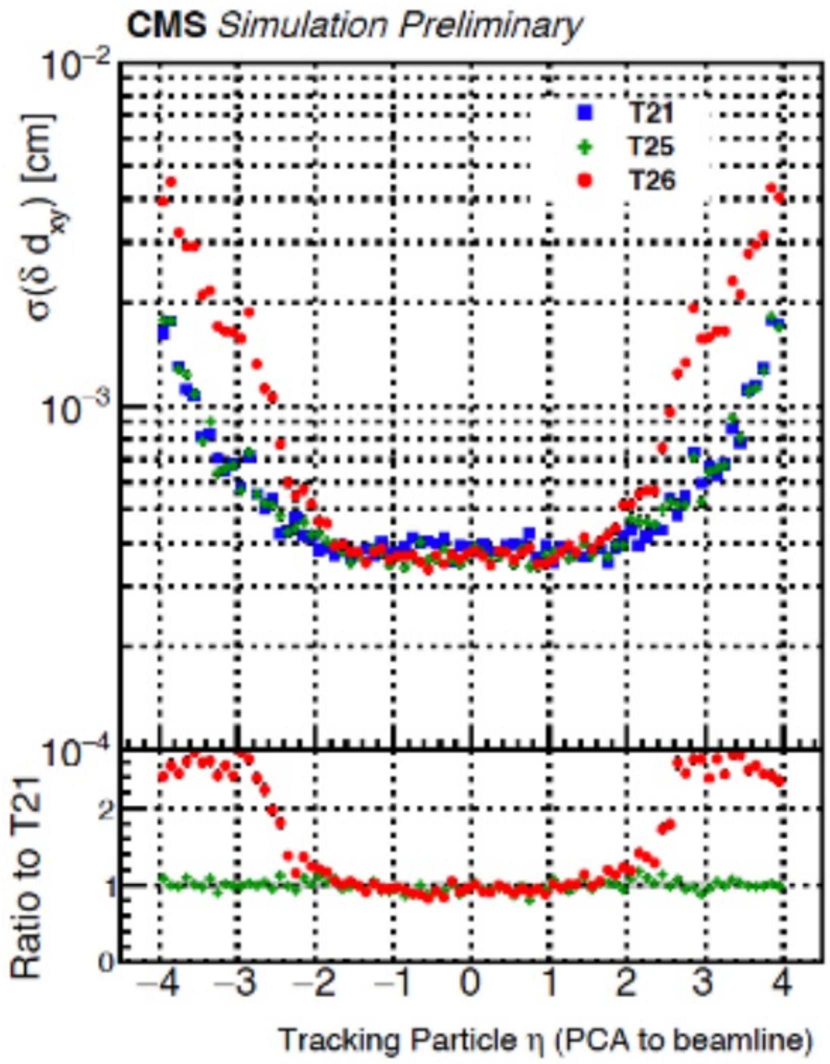
No radiation simulation yet

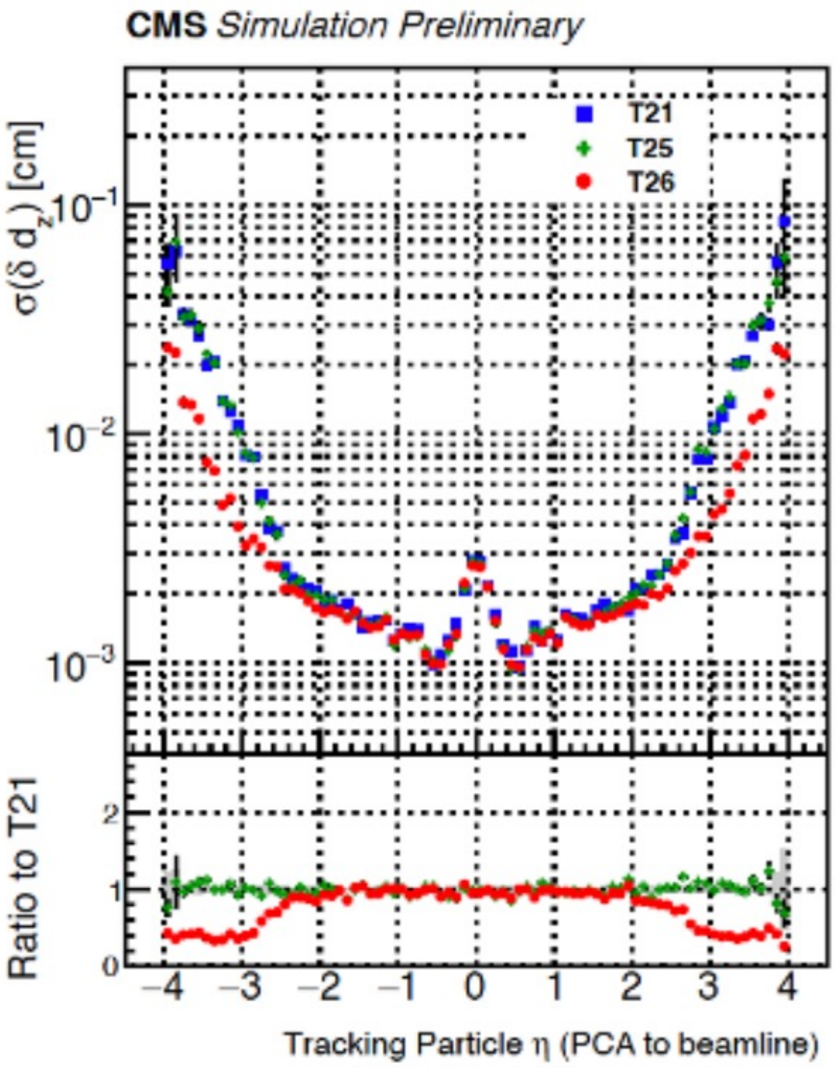


Tracking performance

Events from muon gun (no pile-up, no vtx-smearing)

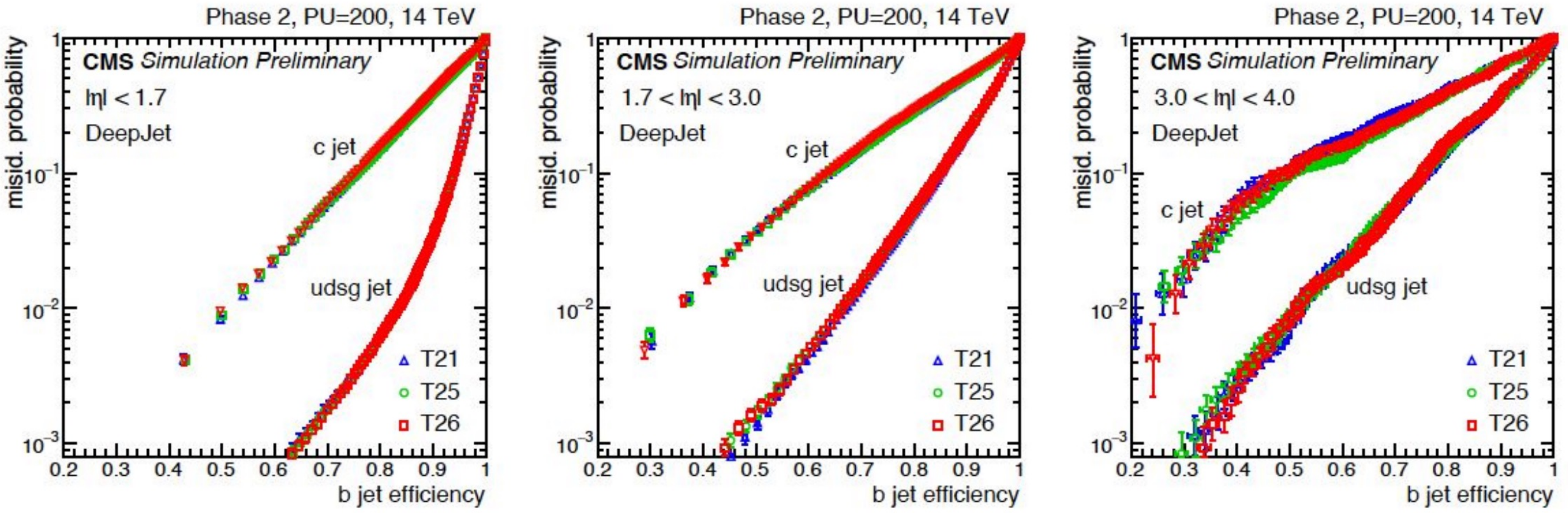
Resolution loss for T26 in d_{xy} is comparable to the gain in d_z





Heavy flavor tagging performance

TTbar+PU200events



Conclusions

- Introduction to HL-LHC and CMS Phase-2 IT project
- Introduction to PixelAV and its use to simulate sensors
- Studied different sensor sizes and sensor technologies
- Developed PixelAV to simulate 3D sensors and avalanche gain effect
- Compared simulations to DESY test beam data
- Studied tracking and heavy flavor tagging in CMSSW