RASER simulation towards 4D tracking detectors

Chenxi Fu[1,2], Jian Feng[3], Dai Zhong[4], Ziyin Cai[4]

On behalf of RASER team

[1]Institute of High Energy Physics, Chinese Academy of Science

[2]University of Chinese Academy of Science

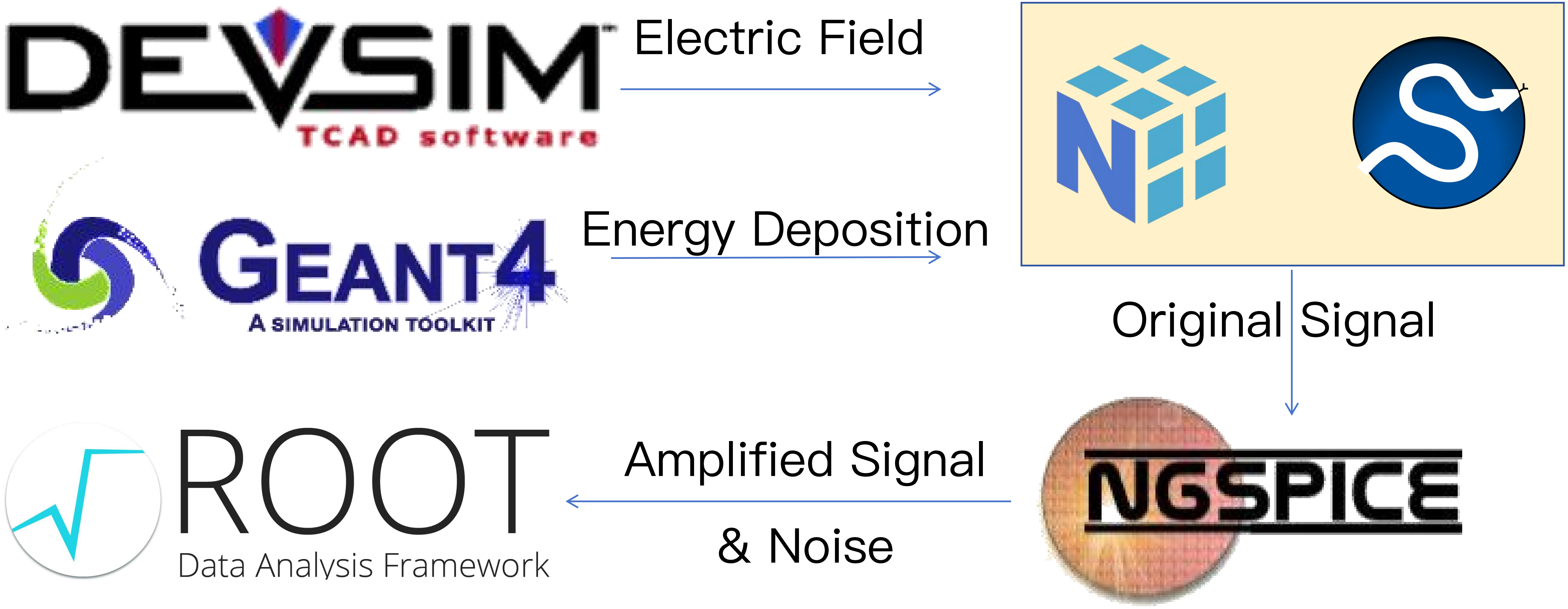
[3]Southwest Jiaotong University

[4]Xiangtan University

Outline

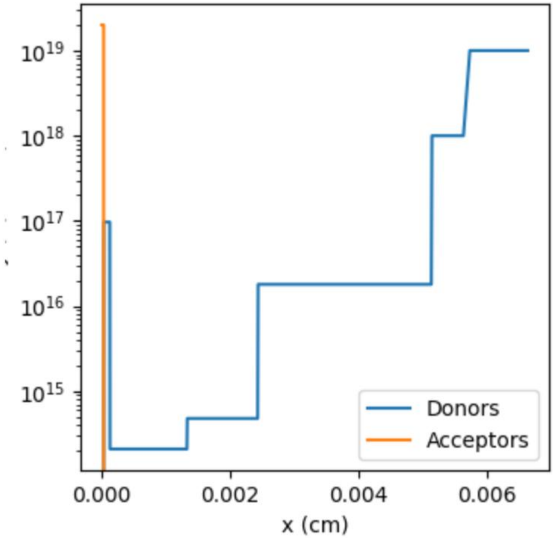
- Device simulation: electric properties
  - electric field, weighting field, IV&CV, irradiation
- single event simulation
  - energy deposition, carrier transport, signal generation, electronics
- multiple event simulation for conventional LGAD detector
  - time resolution
- progresses on simulating strip & pixel detectors
  - cross talk, CoG methods, attempts on simulating space resolution
- Discussion
- Conclution

RASER Simulation Flowchart

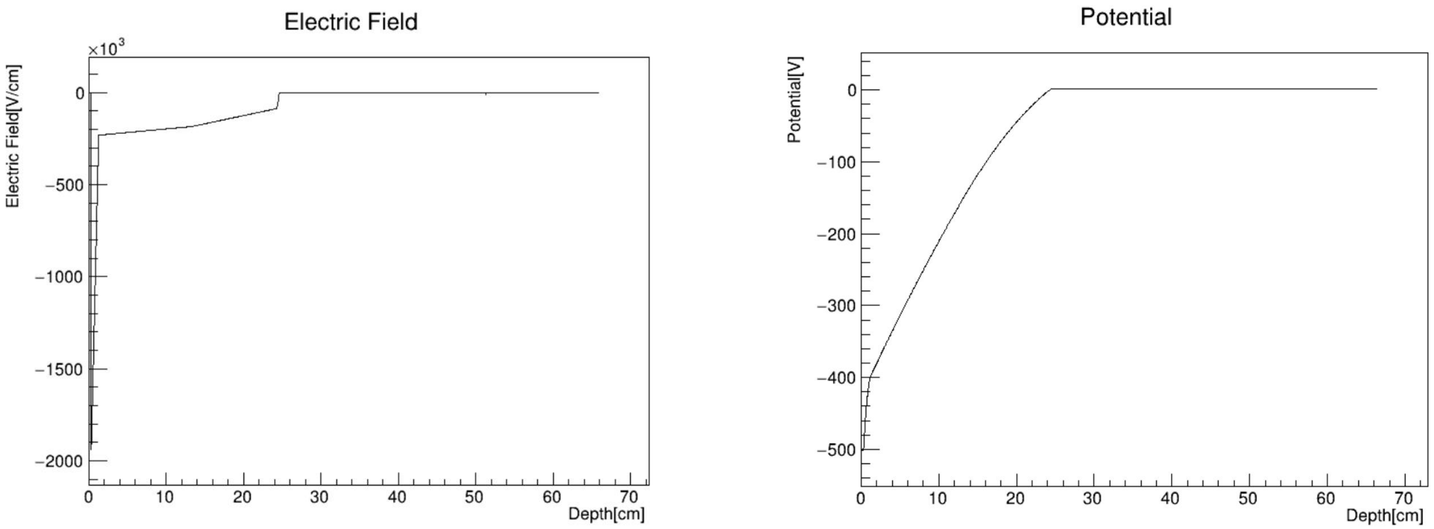


Electric Property Simulation – Electric Field (1)

- Solving the Poisson equation and the carrier continuous equations in steady state
  - Using Scharfetter – Guemmel framework provided by Devsim
    - By Juan Sanchez, http://github.com/devsim/devsim
  - command: raser field <device name>



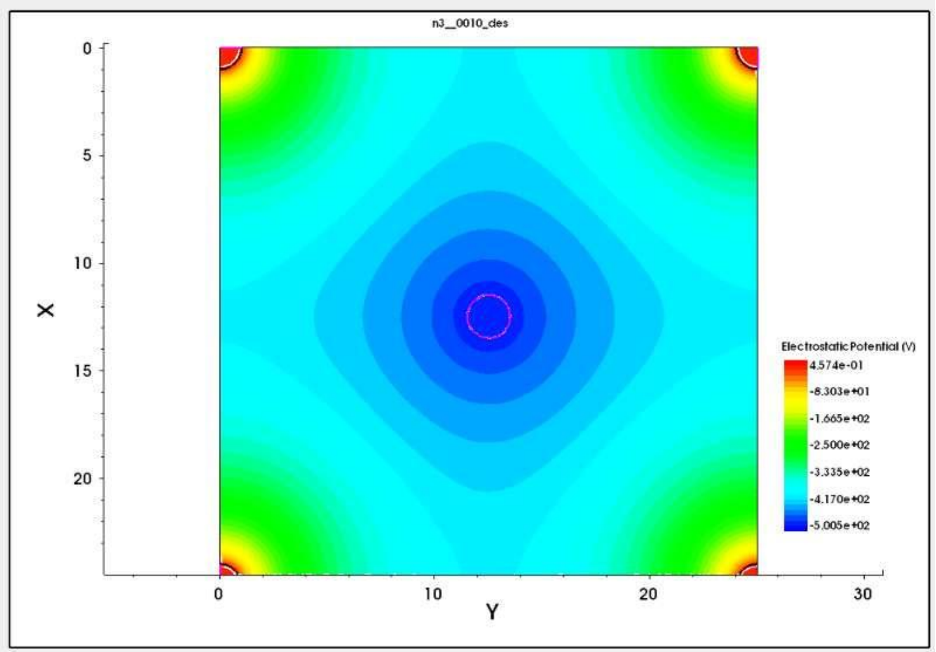
The doping profile of an 1D 4H-SiC LGAD prototype, called SICAR-1, total thickness 66.4um, active thickness 25.1um.

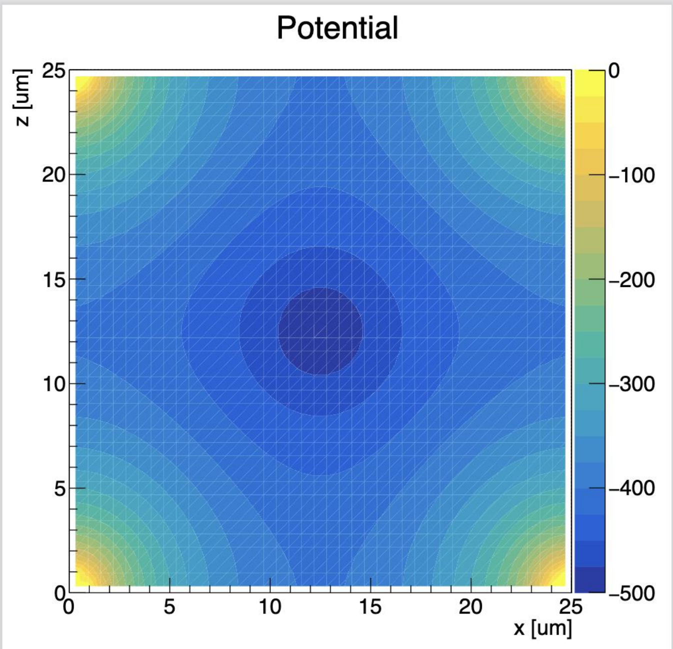


The electric field and the potential of the 4H-SiC LGAD, bias voltage -500V on the top. Under bias voltage enough to deplete the device, the electric field and the potential look just the same as one could obtain from barely the Poisson equation, with doping as charge.

Electric Property Simulation – Electric Field (2)

- Another method: import an Sentaurus .tdr file and convert it into .devsim format
  - By Juan Sanchez, https://github.com/tcaduser/tdr-convert
  - command: raser field -ext <tdr file>

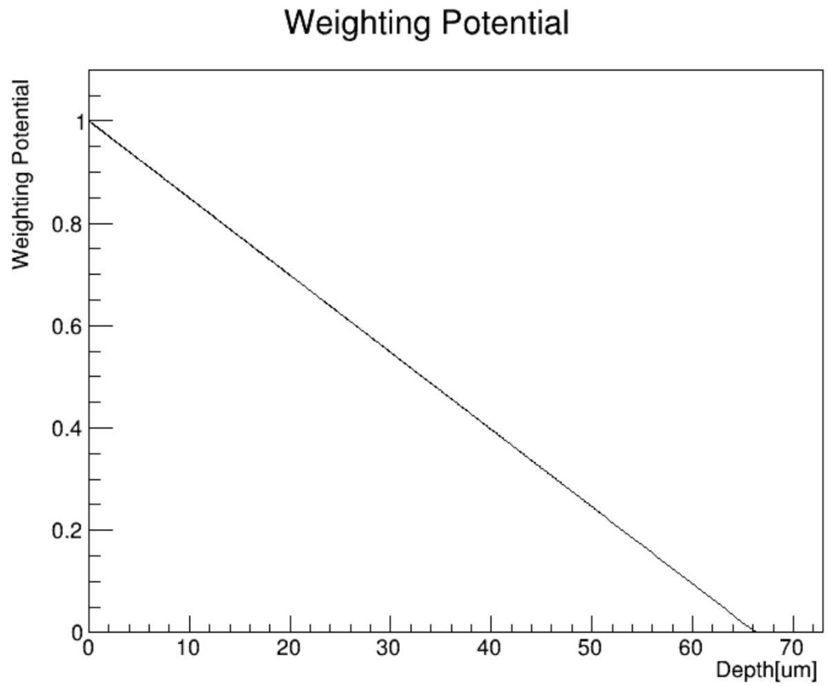


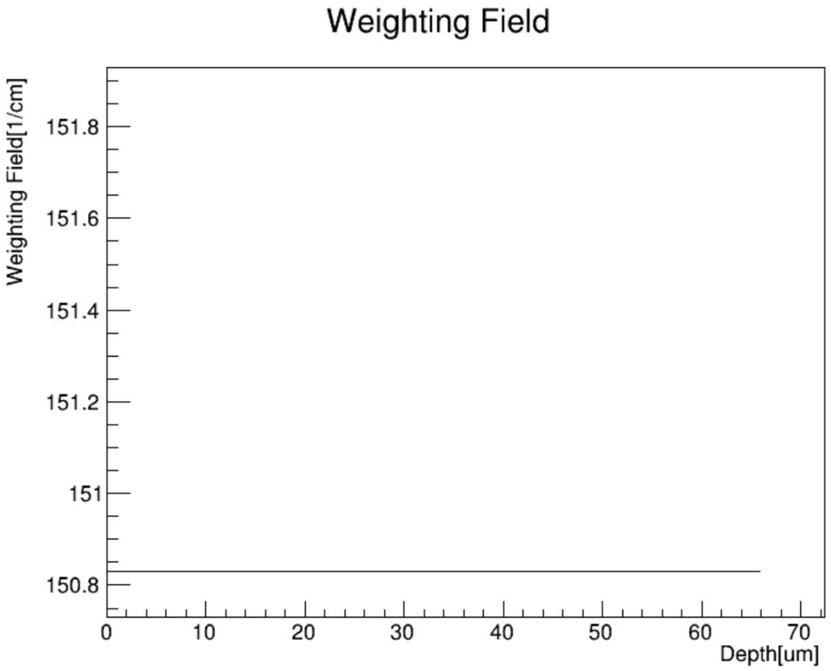


The potential of a 3D pixel detector, bias voltage –500V on the center. On the left is the result provided by Sentaurus, on the right is the converted result drawn by ROOT.TH2F.

Electric Property Simulation – Weighting Potential (1)

- To directly solve
  - charges all removed, 1V bias voltage on specified electrode
  - command: raser field –wf <device name>

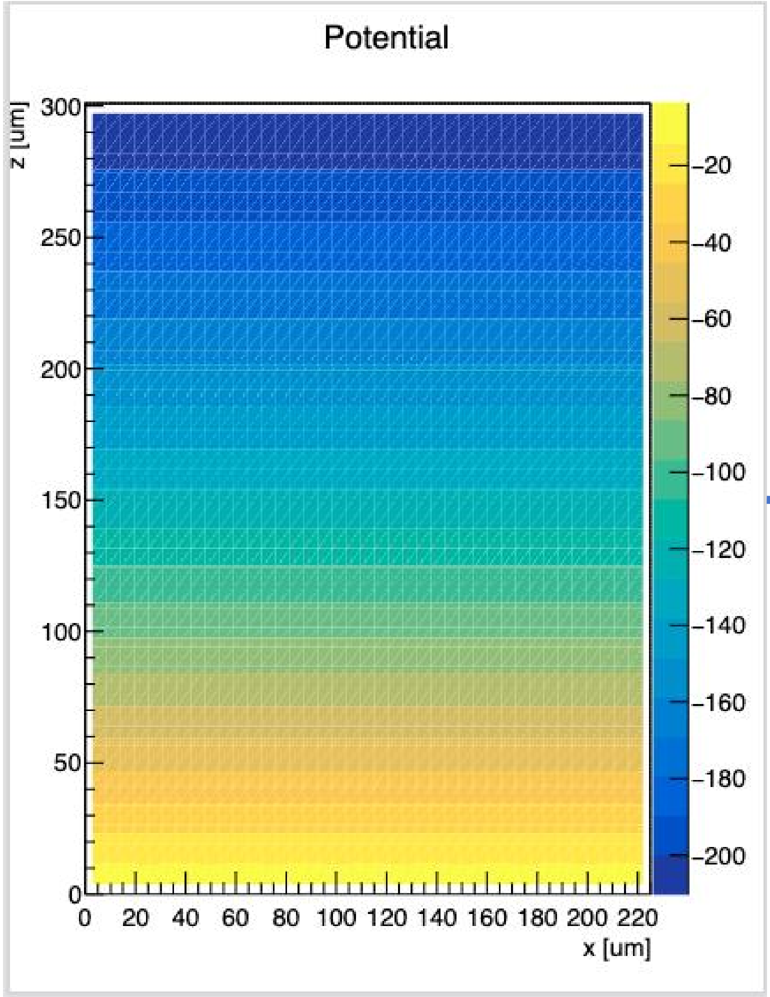


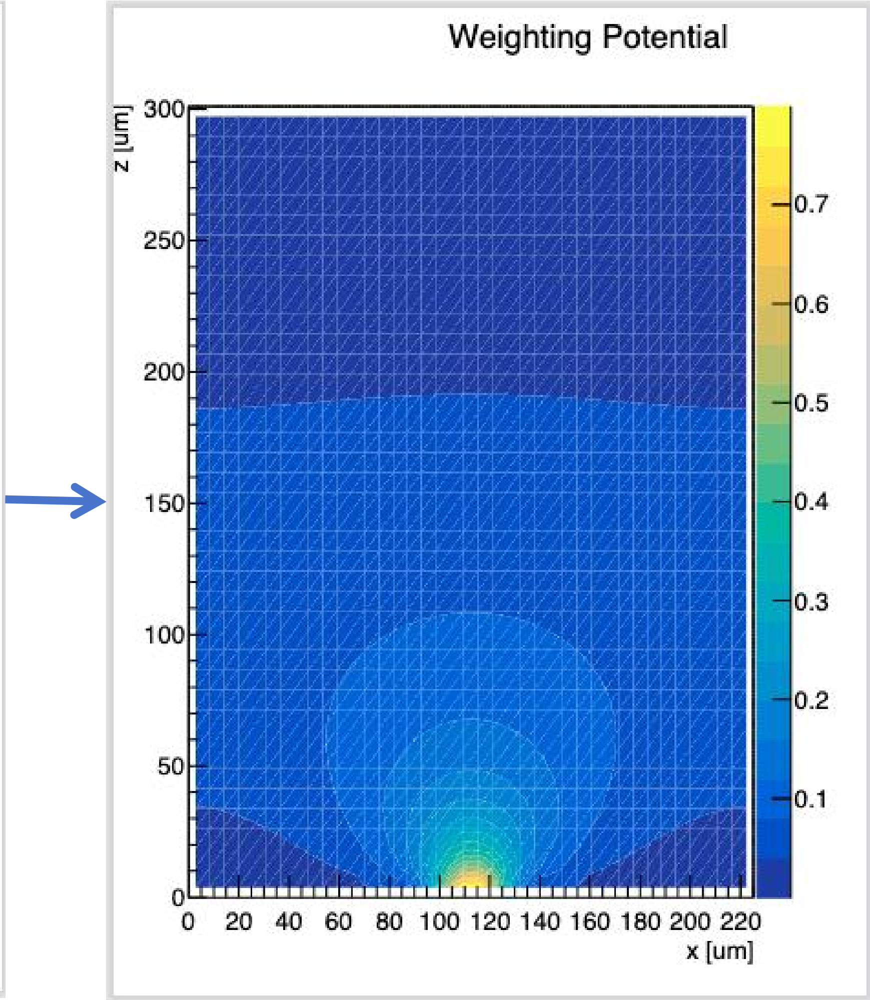


The weighting potential and the weighting field of the 1D 4H–SiC LGAD. Of course it is no more than 1–x/d and 1/d.

Electric Property Simulation – Weighting Potential (2)

- For undepleted device or Sentaurus imported device:
  - calculate the electric potential
  - add another 1V onto the specified electrode
  - take the substraction between the new potential distribution and the old
  - command: raser field <device name> -wf_sub <voltage> <electrode name>

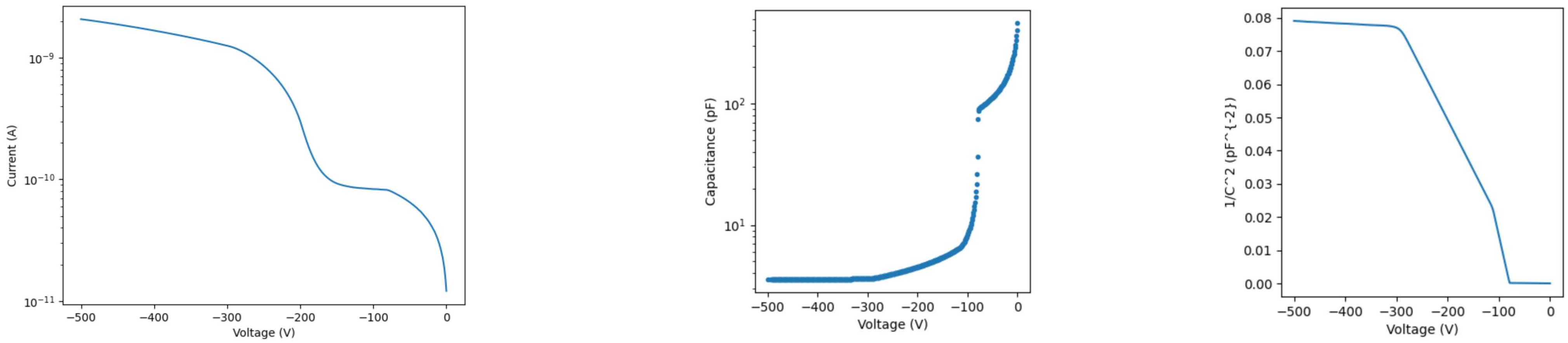




The potential of a 9-electroded strip detector. After adding the extra 1V on the center and subtract the two potentials can we obtain the weighting potential for a quite wide range.

Electric Property Simulation – IV & CV

- IV directly obtained from the ramping of devsim.solve
  - Impact Ionization added for breakdown simulation
  - command: raser field <device name> (the same as the one for electric field)
- CV needs additional AC simulation
  - command: raser field <device name> –cv (also including IV & field results)



Simulated I–V, C–V and C⁻²–V relations of the 4H–SiC LGAD. The area factor is set to be 0.01 cm^2.

Electric Property Simulation – Irradiation

- Based on SRH model and trap assisted tunnelling
- Silicon: Modified HPTM model
  - arXiv:2504.20463
- 4H-SiC: $Z_{1/2}$ and EH_3
  - arXiv:2503.09016

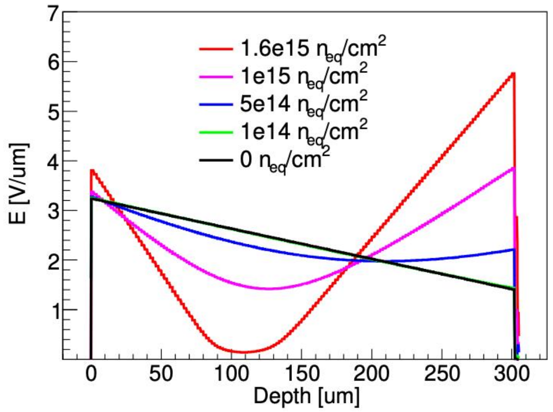


Fig. 3 Simulation results of electrical field distribution with different fluences at 700 V

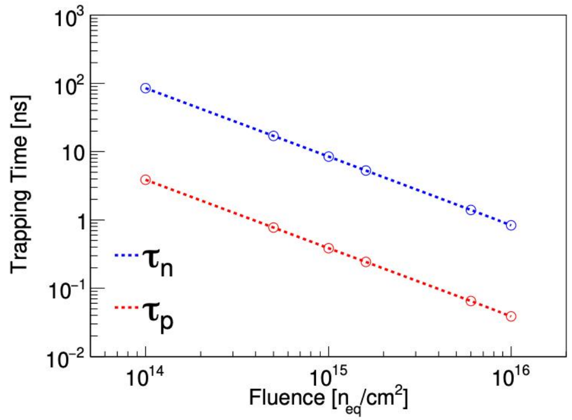


Fig. 4 Simulation results of trapping time with different fluences

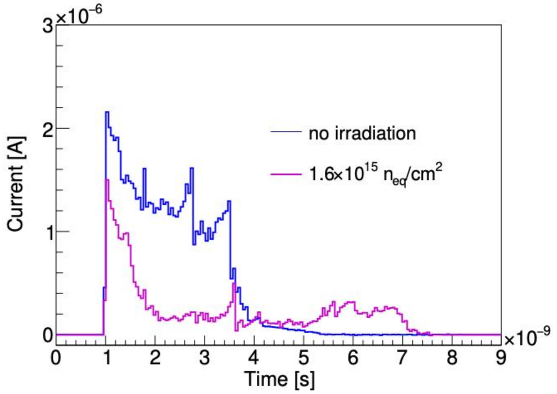
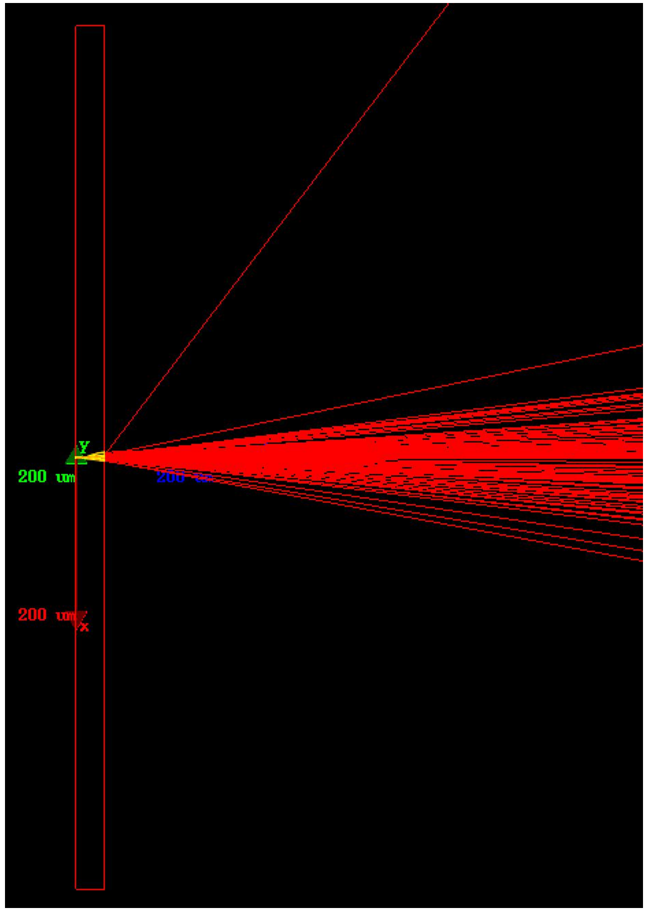


Fig. 5 Simulation of induced current waveform with non-irradiation and 1.6×10^{15} n_{eq}/cm^2 fluence at 500 V

Single Event Simulation – Energy Deposition

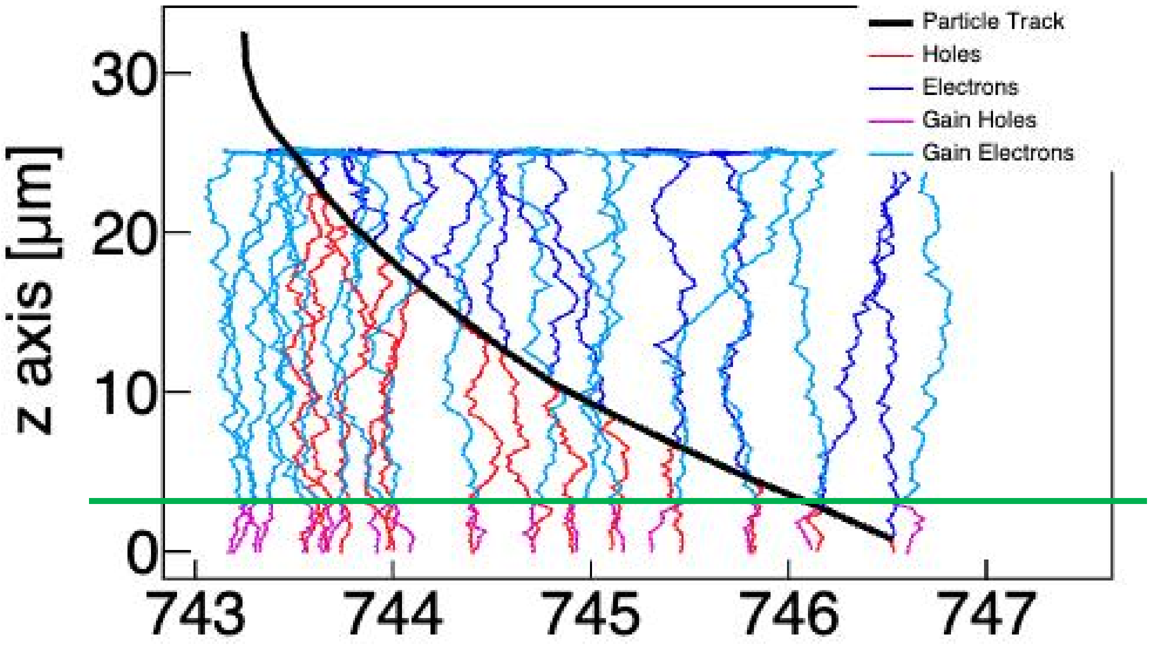
- Use Geant4
  - Construct detector geometry
  - Append a particle gun to the system
  - Generate everal events with different track and energy deposition value
  - Convert deposited energy to excited carriers
    - 3.6 eV per pair for Si, 8.4 eV for 4H-SiC
- command: raser signal <device name>
  - for the whole single event simulation process



A ^{90}Sr beta source is placed on the left of the DUT (the red rectangle). After passing through the DUT, the beta rays are greatly scattered.

Single Event Simulation – Carrier Transport

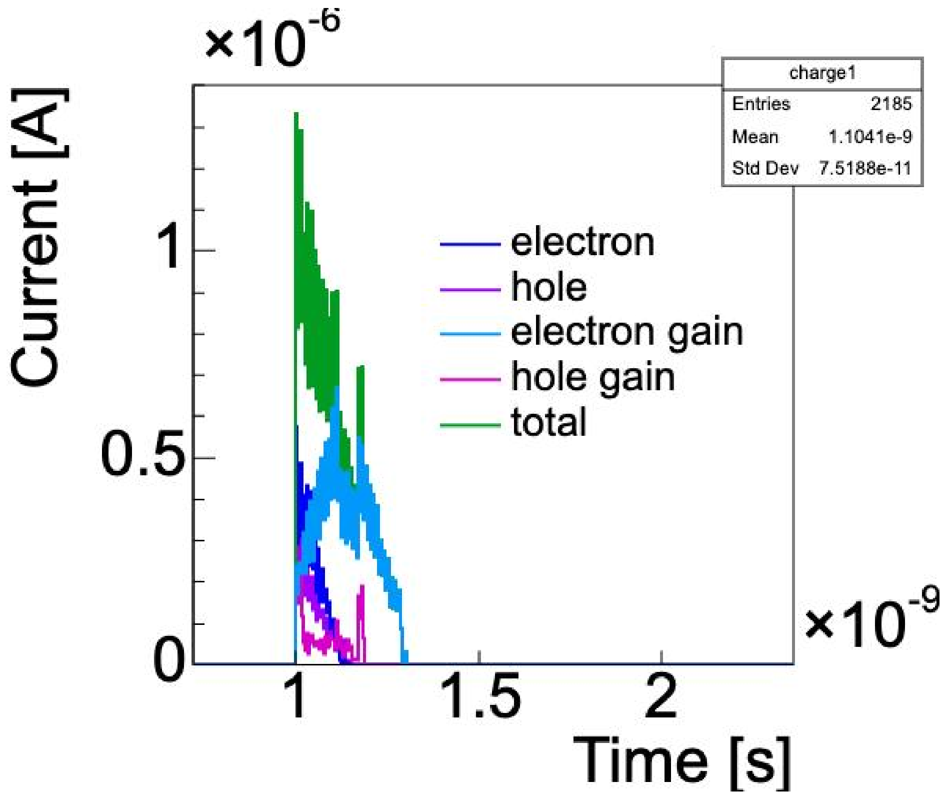
- Use NumPy & SciPy
  - Start carrier drift from the energy depletion track
  - Terminate drift at the device border or low-field region
    - except for non-full-depletion MAPS
  - Release multiplied carriers upon reaching the gain layer for LGADs
    - gain rate previously calculated by integrating the impact ionization coefficient



The black line illustrates how an incoming beta ray passes through the 4H-SiC LGAD. The original carriers (red for holes, blue for electrons) start to drift immediately after generated, and when the holes touch the gain layer (the green line), the gain carrier pairs will be generated from avalanche (magenta for holes, cyan for electrons). In the picture can we see that carriers end drifting on the border of the depleted area.

Single Event Simulation – Signal Generation

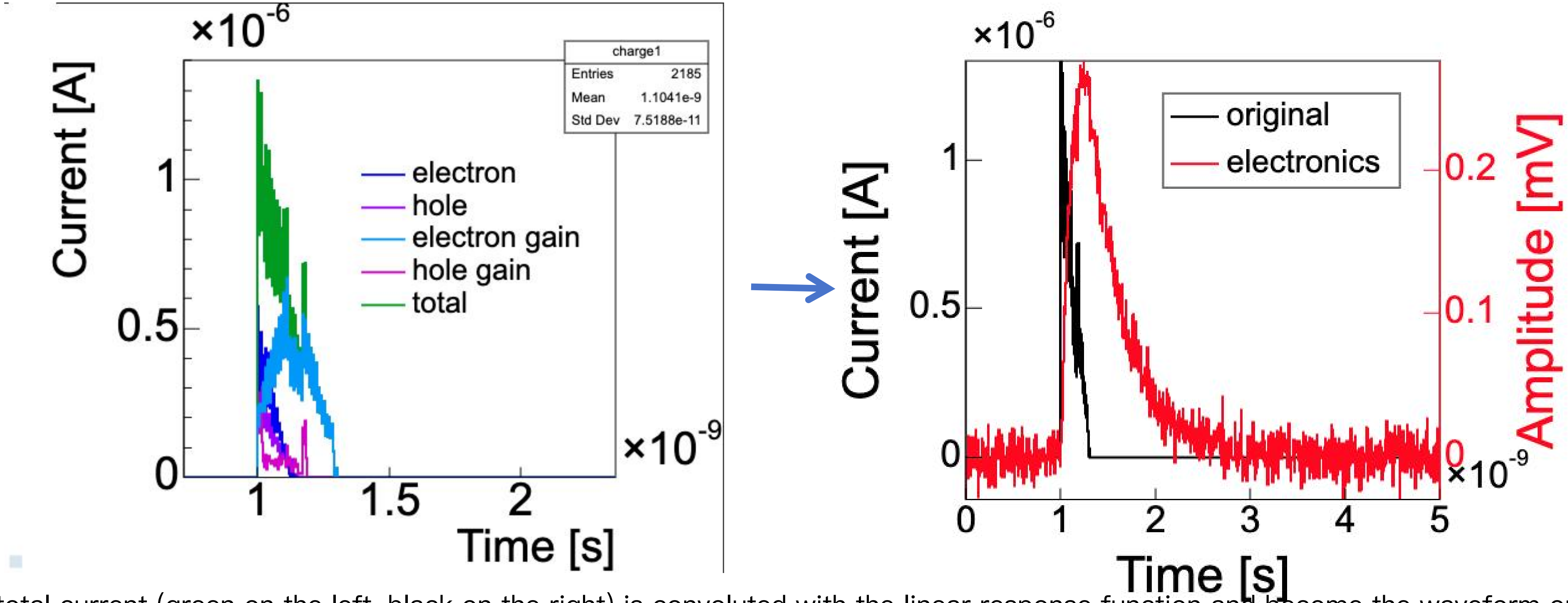
- Use NumPy, SciPy & ROOT
  - Calculate induced current using the Shockley–Ramo theorem
    - I = q*v*Ew
  - Simplify calculation by integrating current over drift time step
    - I*dt = q*dx*Ew = q*dUw
    - Bypass the need for weighting potential differentiation
  - Accumulate signals with TH1F.Fill()



The generated signal by beta ray in a silicon LGAD. The four components and the overall signal are shown in the legend. The gain of the LGAD is 2.1.

Single Event Simulation – Electronics (1)

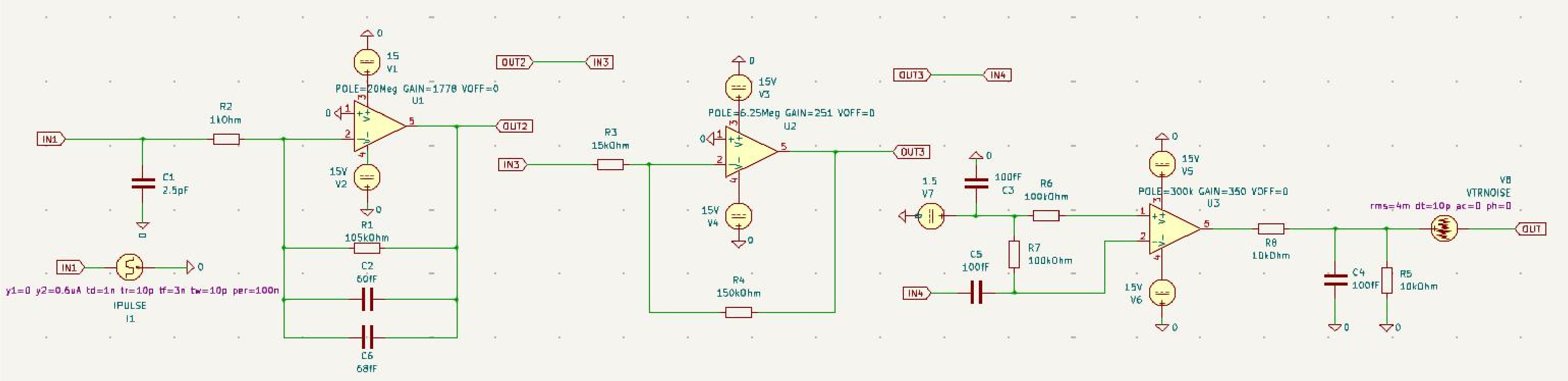
- Convolute the signal with pulse response functions
- Gaussian noise added



The total current (green on the left, black on the right) is convoluted with the linear response function and become the waveform output (red on the right), noise added

Single Event Simulation – Electronics (2)

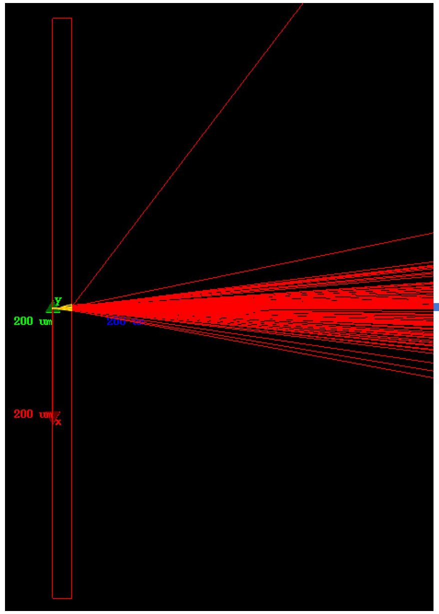
- Use NGSpice by sending the signal into a .cir netlist file
  - NGSpice: https://ngspice.sourceforge.io
  - KiCad: https://www.kicad.org



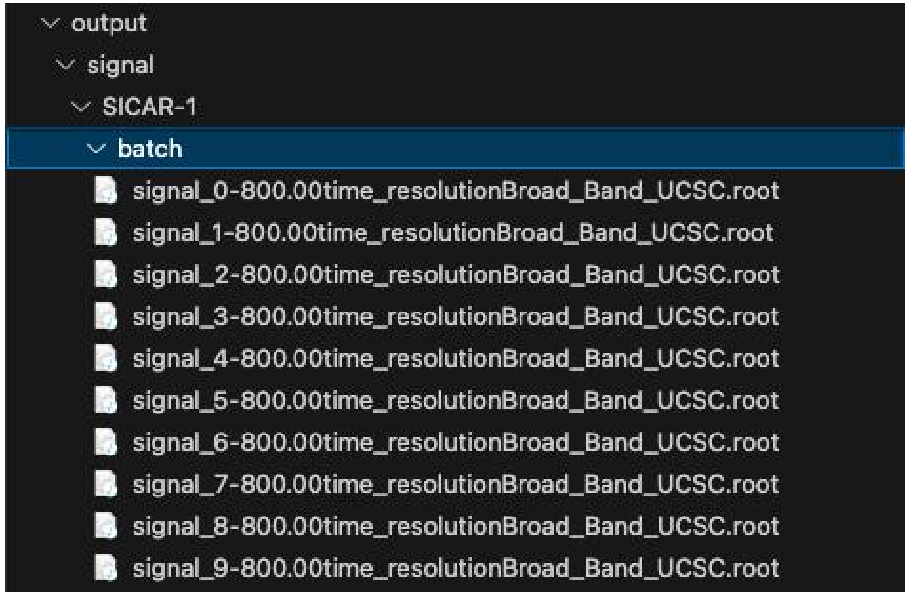
A 3-staged amplifier circuit drawn in KiCad, which could easily converted into an NGSpice netlist, which could be utilized by RASER as the front end electronics. Noise included at the end of the circuit.

Multiple Events – Batch Job

- For the statistical performances, a great amount of events need to be simulated
- Command: raser signal –s <job number> <device name>
  - use –b parameter for HTCondor service on IHEP server



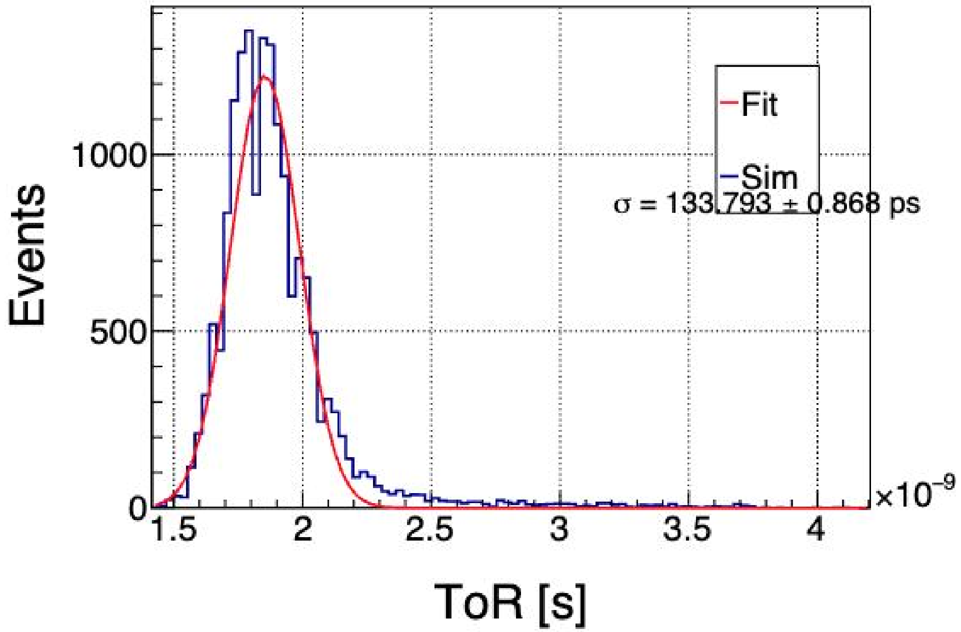
Multiple events in Geant4. We divide the events in to 100 jobs and send them to compute platforms by HTCondor service. The figures on the left and the right correspond to different injection angles.

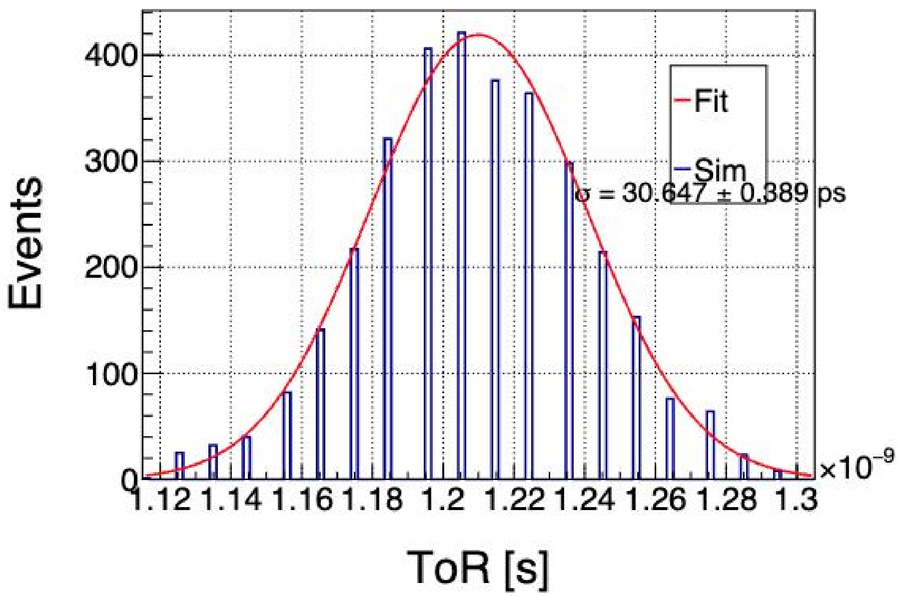


Every running result of a job will be filled into an .root file by the form of ROOT TTree object. The TTrees are able to be further analysed.

Multiple Events – Time Resolution

- Use Constant Fraction Discriminator to obtain arrival time
- Command: raser resolution <device name>

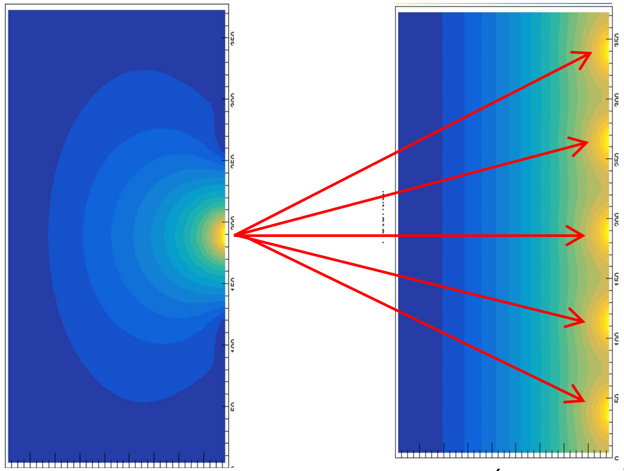




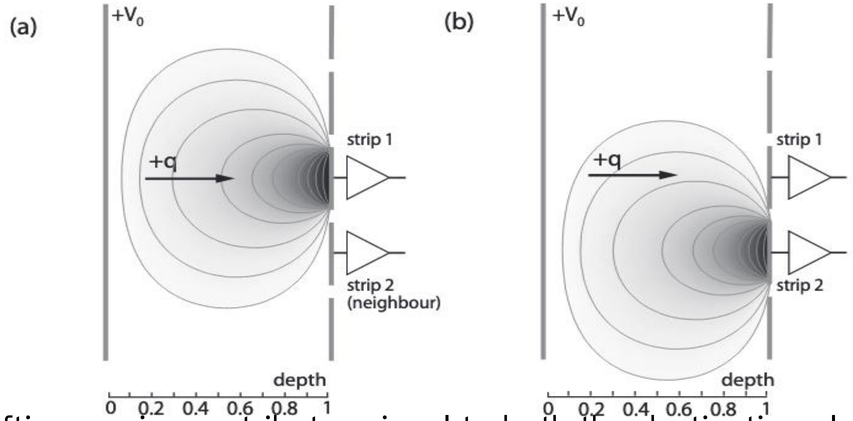
The time resolutions of 4H-SiC PiN (left) and LGAD (right).
The time resolution is simulated and tested in 10.3389/fphy.2022.718071.
The gain and time resolution is simulated in 10.1007/s41605-023-00431-y and tested in 10.1109/TNS.2024.3471863.

Strip Detectors – Adjustments on the Electric Field and Weighting Field

- Field solution simplified
  - by solving only within a periodic unit cell for detectors with numerous electrodes
- Assign this identical field to all electrodes
  - considering the effects from neighbouring ones
- Calculate the signal induced
  - not only on the nearest electrode, but on all others where the weighting potential is non-zero at the carrier's location



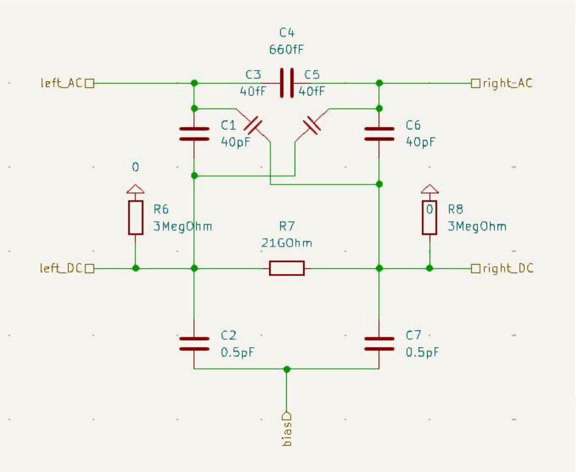
The real electric field is translationally symmetric (periodic), while the weighting field, though not periodic, is identical for every electrode. This permits the use of a single unit-cell field for all channels, eliminating redundant calculations.



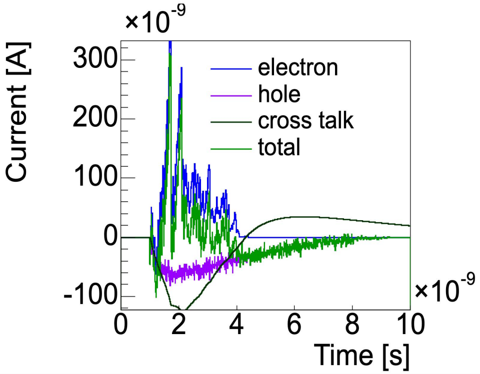
An drifting carrier contributes signal to both the destination electrode and its neighbouring electrode (figure from *Particle detectors: fundamentals and applications* by Hermann Kolanoski & Norbert Wermes)

Strip Detectors – Cross Talk

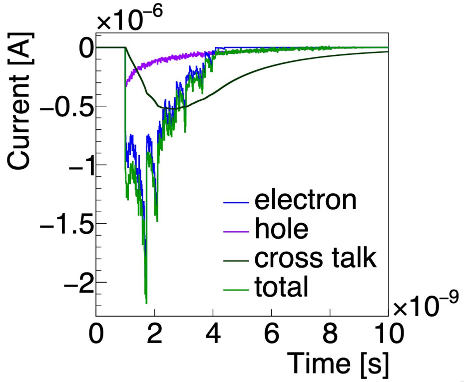
- Parasitic capacitances between neiboughing electrodes will change the output signal
  - Some strip sensor utilizes this feature to reduce readout channels while keeping the space resolution unchanged



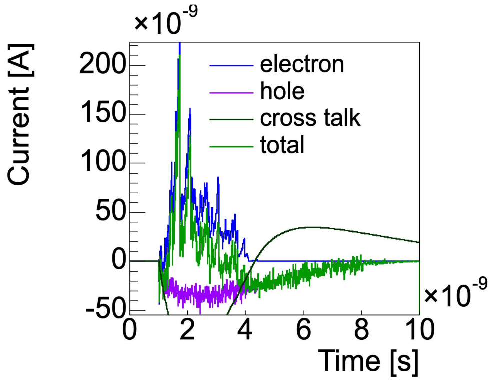
Resistances and capacitances between two readout strips in KiCad.



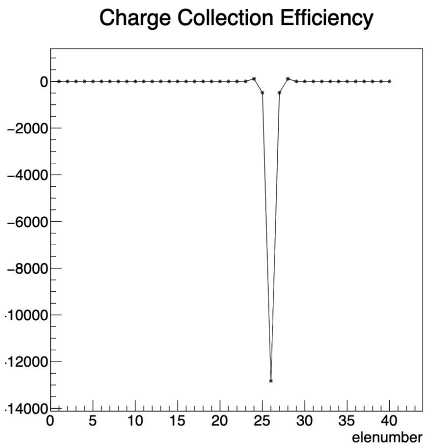
signal of electrode 25



signal of electrode 26



signal of electrode 27

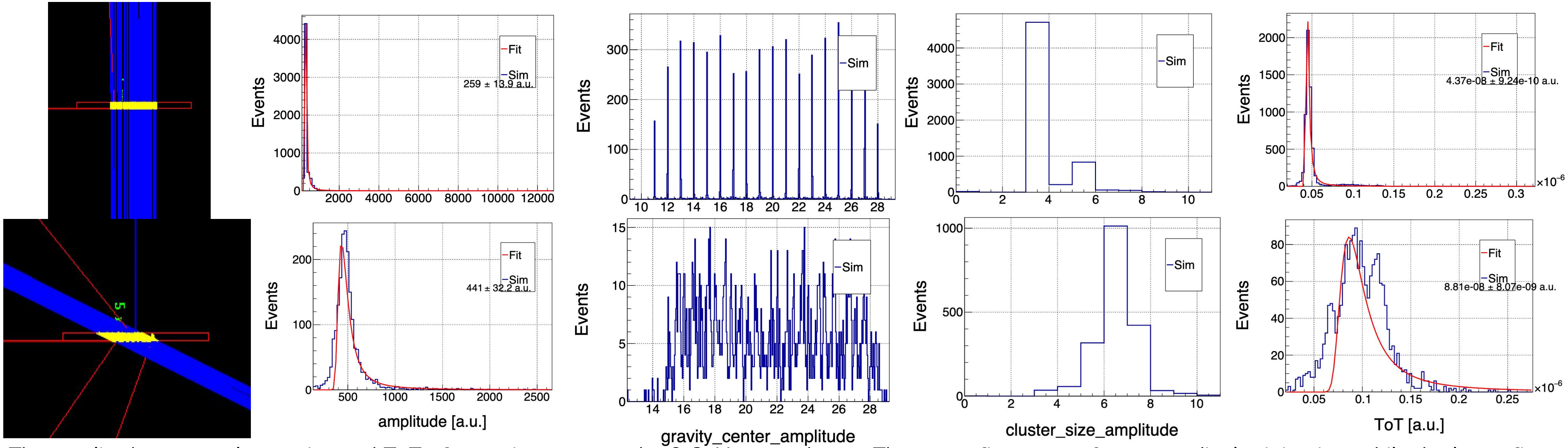


charge collection of the 40–channeled sensor

Signals induced on a 40–channel AC–readout strip sensor (left 3 figures) from a MIP hitting on electrode 26. Without cross talking (merely considering the expansion of the weighting field), the signs of the pulse on the neighbouring electrodes are opposite to the original one. But with cross talking, these signs appear to be the same, and lead to a charge collection with the same sign as the target electrode (right).

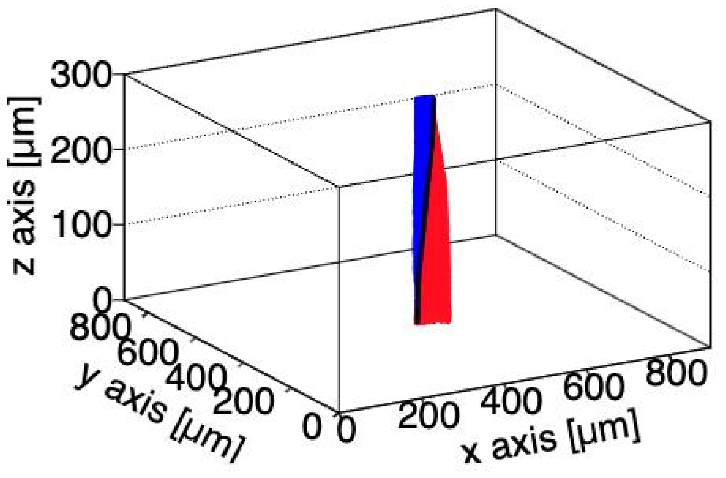
Strip Detectors – Statistics from waveforms

- Every event brings a list of characterized datas
  - total amplitude, gravity center, cluster size, time over threshold, etc.
  - their distributions reflect the procedure of particle-detector interaction

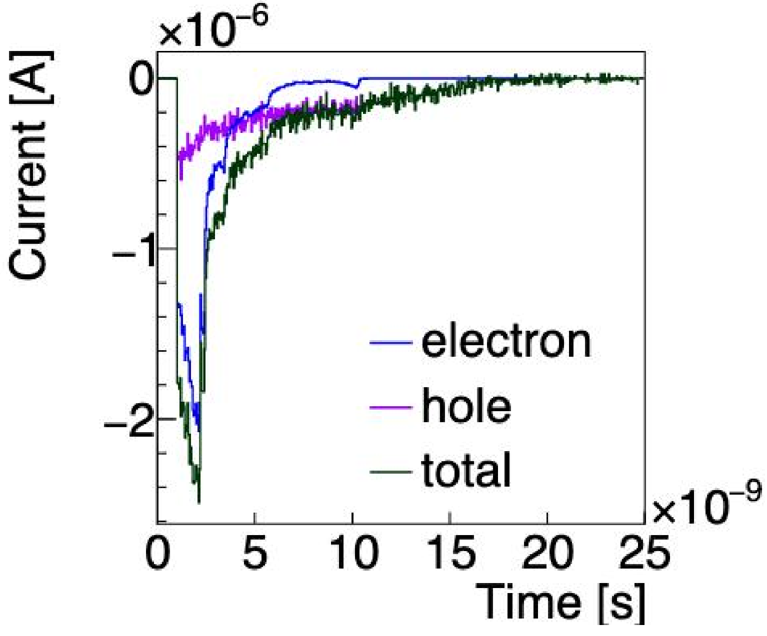


The amplitude, c.o.g., cluster size and ToT of an strip sensor under 3 GeV proton beam. The upper figures are for perpendicular injection, while the lower figures corresopnds to an injection tilted for 60 degrees. Beacuse of the existance of a strong cross talk, the cluster sizes tend to be bigger than other strip devices.

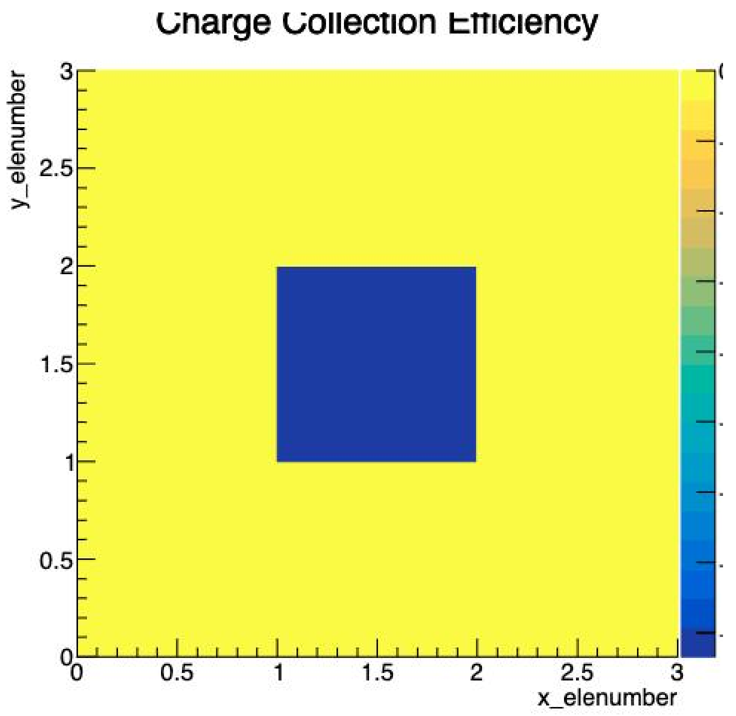
Attempts on Pixel Detectors



A beta particle hitting a pixel detector with volume of 900*900*300 um, pixel size 300*300 um

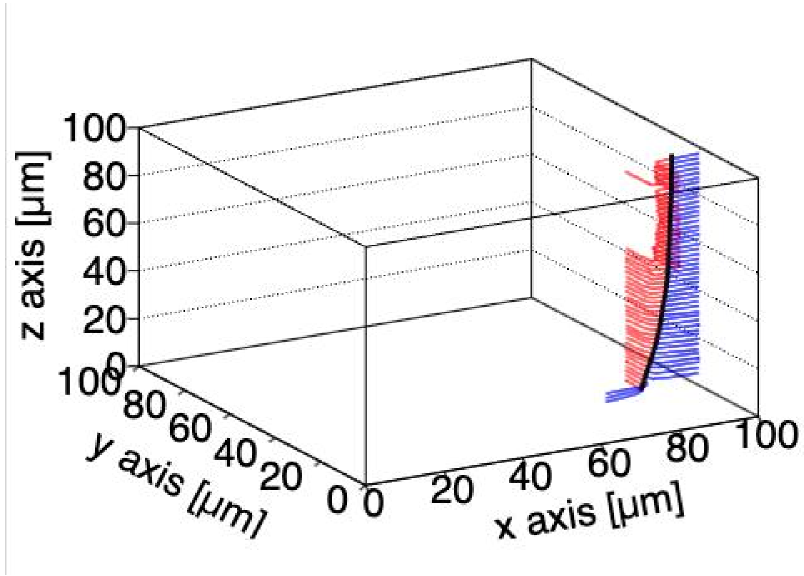


Simulated waveform of the center electrode.

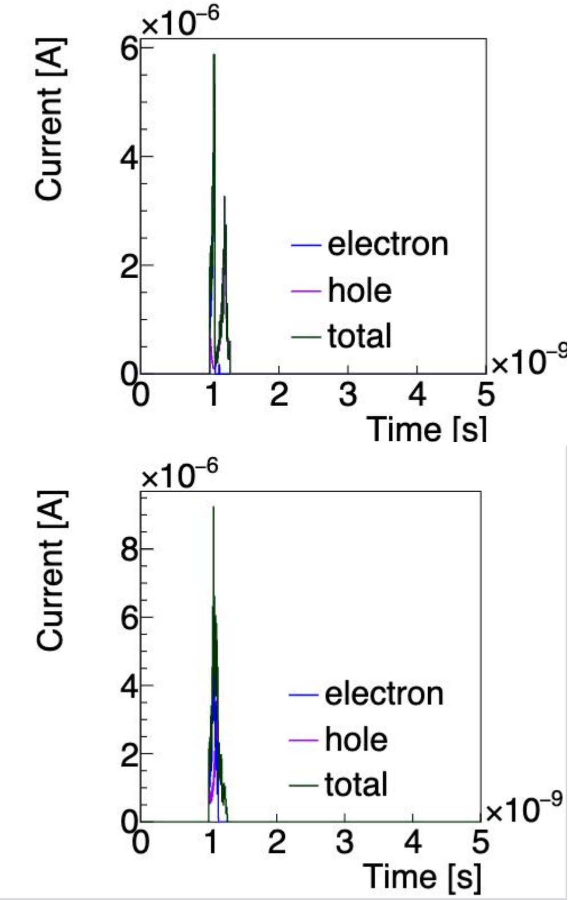


Charge collection result on the pixel detector, not including the effect of charge sharing & cross talk

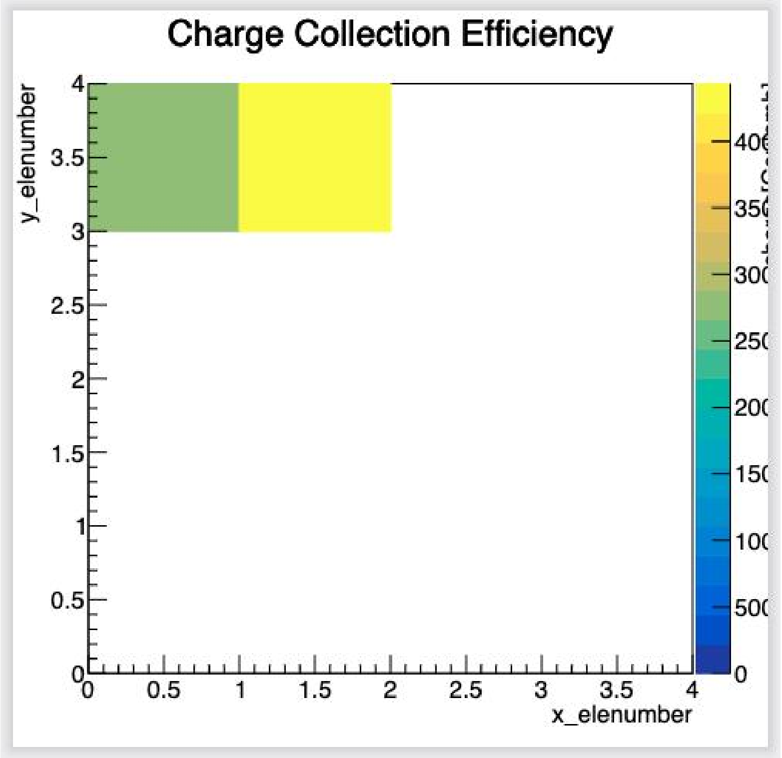
Attempts on 3D Pixel Detectors



A beta particle hitting a 3d pixel detector with volume of 100*100*100 um, pixel size 25*25 um, electrode pillar radius 1um



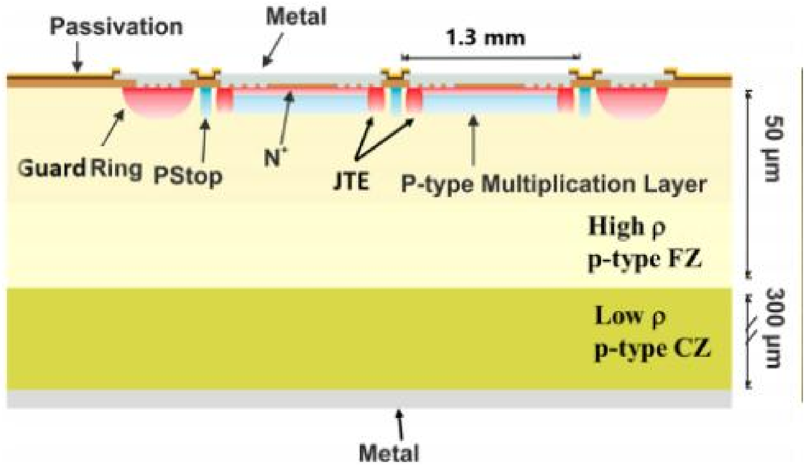
Simulated waveform of the electrode (1,4) and (2,4).

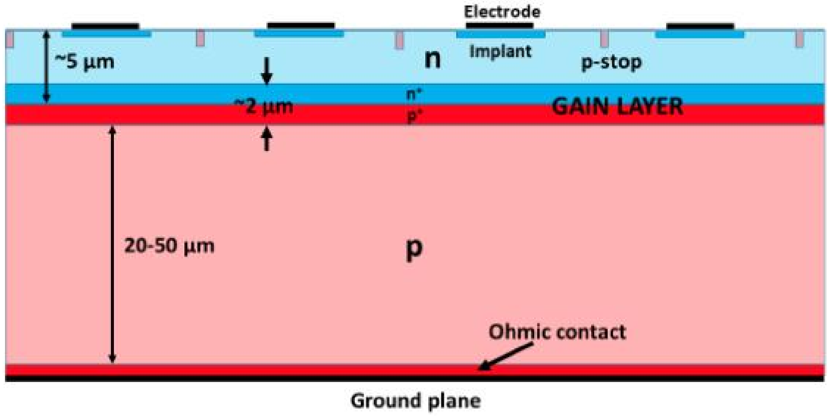


Charge collection result on the 3d pixel detector, not including the effect of charge sharing & cross talk

Discussions For simulating 4D trackers

- (Trenched, DC) LGAD, strip or pixel
  - Need to modify gain carrier generation to take the transformation of the electric field near JTE into account
- Deep Junction LGAD & inverted LGAD, strip or pixel
  - Straight forward
  - Combine the existing gain carrier generation & strip/pixel signal generation

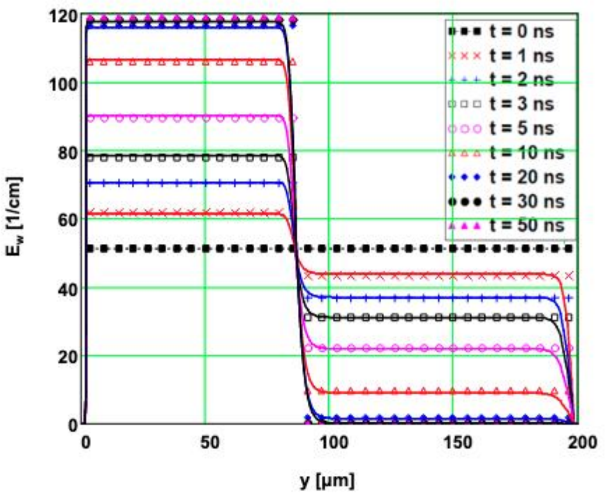




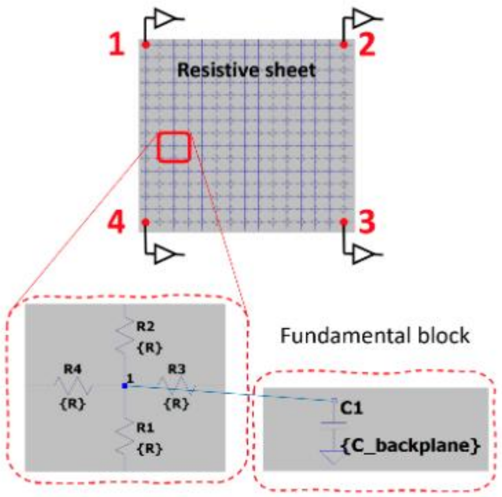
from: Deep-Junction LGAD to achieve high granularity and radiation hardness, Dr. Simone Michele Mazza

Discussions For simulating 4D trackers

- AC-LGAD, DC-RSD
  - Hardest
  - Existing method:
    - 1. Use time-dependent weighting potential (like arXiv:2505.05974)
      - Messy for Sentaurus user
      - time dimension added, simulation time cost greatly increased
    - 2. Use RC network (like arXiv:2508.16324)
      - Carrier injection coordinate might not fit the granularity of the resistive sheet



time-dependent weightig field, picture from 10.1016/j.nima.2019.162418



resistive sheet in spice, picture from arXiv:2508.16324

Conclusion, Next steps & More

- Conclusion
  - RASER is built for conceptual simulation of solid-state detectors
    - mainly focusing on novel designs and applications
  - Through a series of processes, one could obtain several properties of the new designed detector
    - its responce to certain stimulation, its resolution, and its performance under irradiation
- Next steps
  - finish the simulation of strips & pixels
  - extend our simulation to resistive silicon detectors
  - include open-source PDK & fabrication simulation, digital front end, and so on
- Contacts
  - git repo: https://github.com/raser-team/raser
  - pypi: https://pypi.org/project/raser/

Thanks for listening and welcome to contribute!