Pixel Simulations in CMS

M. Swartz
Johns Hopkins University

CMSSW Simulation

- Geant-based CMS “full sim” production MC w/ digitizers for sub-detectors
- Use GEANT4 to simulate charge deposition in pixels
  - ✳ includes fluctuations and large delta rays
- Assumes uniform E-field across sensor substrate
  - ✳ technically correct only at “type inversion”
  - ✳ uniform Lorentz drift
  - ✳ no carrier focusing at n+ implants
  - ✳ does have carrier diffusion
- No trapping or charge induction
  - ✳ dominant effects in radiation damaged sensor
- Works remarkably well for unirradiated (lightly irradiated) sensors
  - ✳ fails badly for heavily irradiated sensors
- Includes readout effects [ROC thresholds and analog response]
- Includes dynamic ROC inefficiencies and dead channels
- FAST
- Soon to include realistic sensor and radiation effects by re-weighting clusters from Pixelav-simulated 2D cluster shapes

Pixelav Detailed Simulation

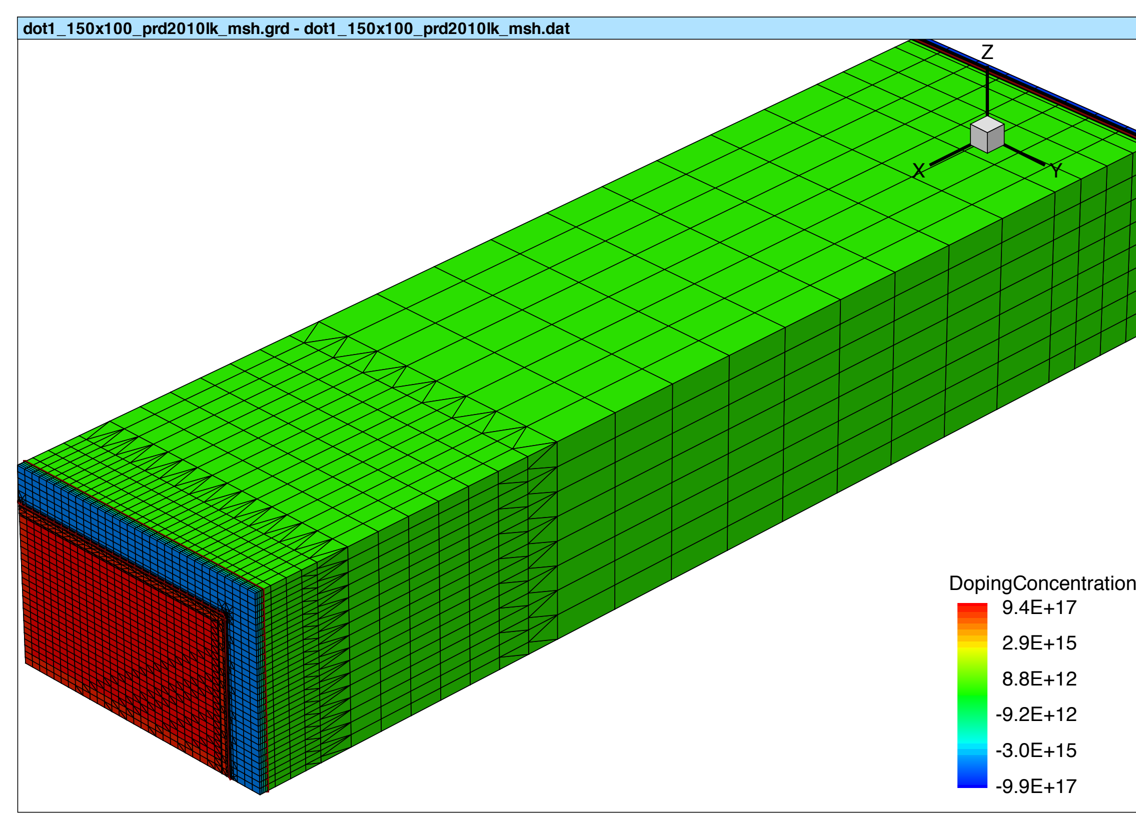
Created to interpret beam tests of irradiated sensors, now used to perform Lorentz calibrations and generate template profile shapes:

- charge deposition model based on Bichsel π-Si x-sections
- delta ray range: Continuous Slowing Down Approx + Nist Estar dedx
- plural scattering and magnetic curvature of delta ray tracks
- carrier transport from Runge-Kutta integration of saturated drift

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

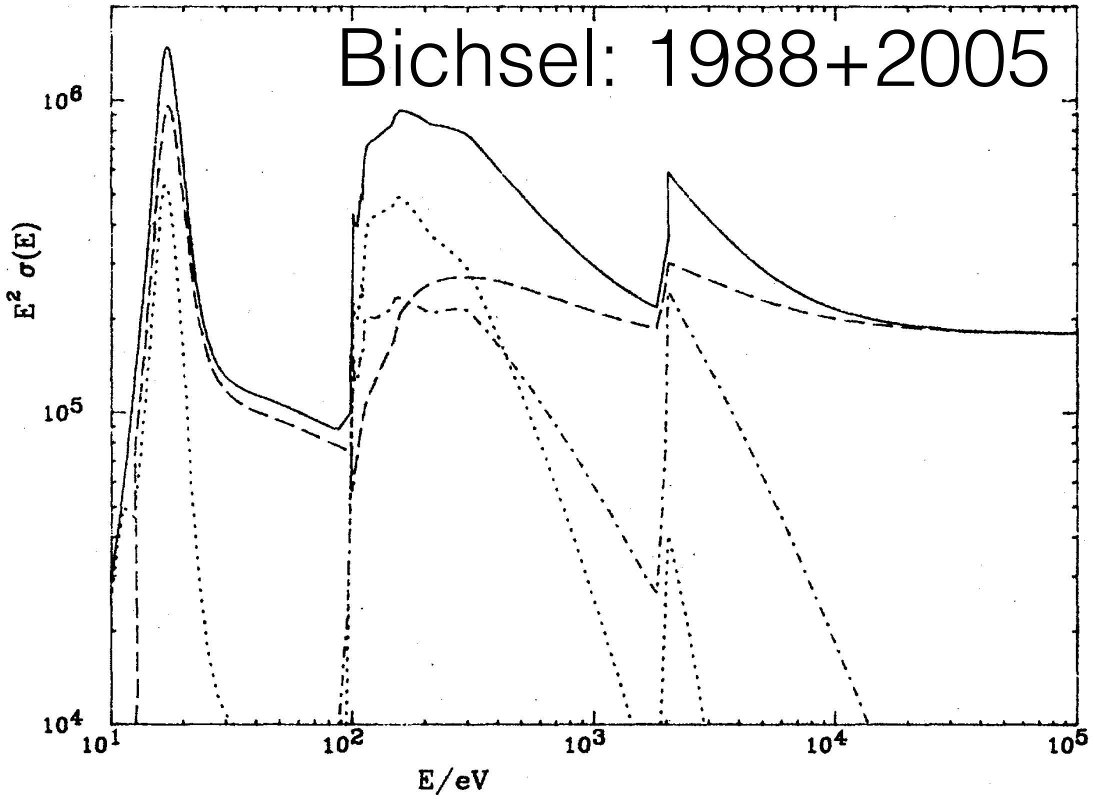
  - ✳ electric field map from ISE TCAD simulation of pixel cell
  - ✳ includes diffusion, trapping, and charge induction on implants

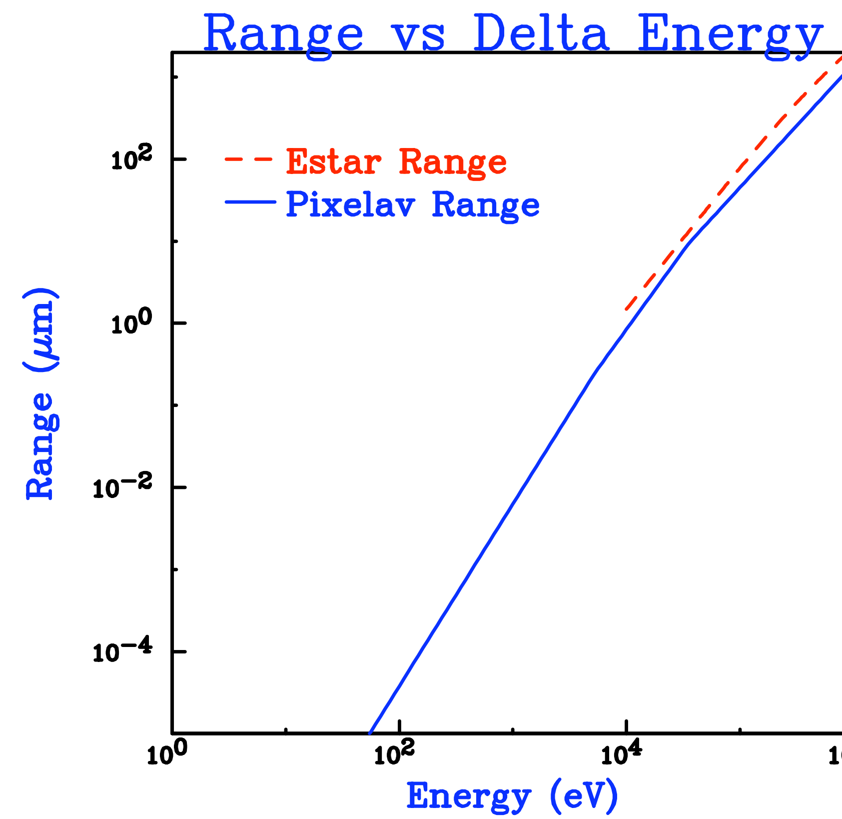
- Electronic Simulation: noise, linearity, thresholds, mis-calibration



Charge Deposition

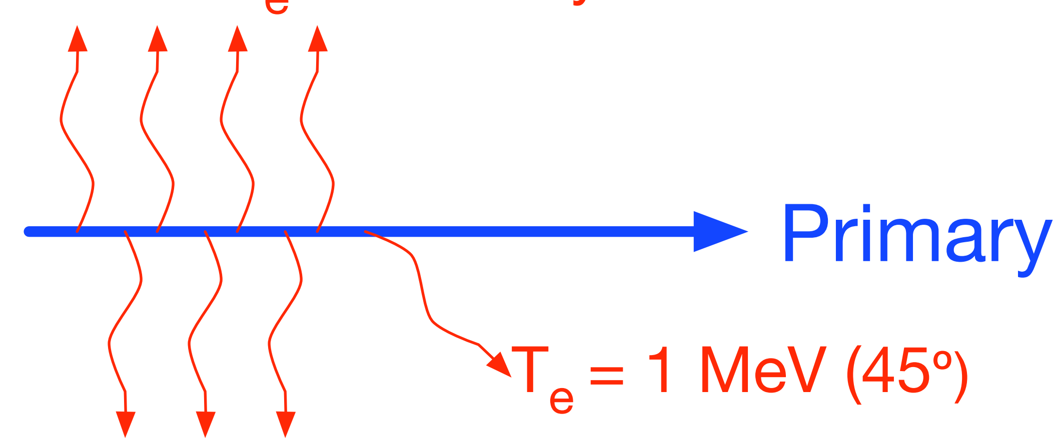
Charge deposition via Bichsel differential cross sections (depend on βγ)



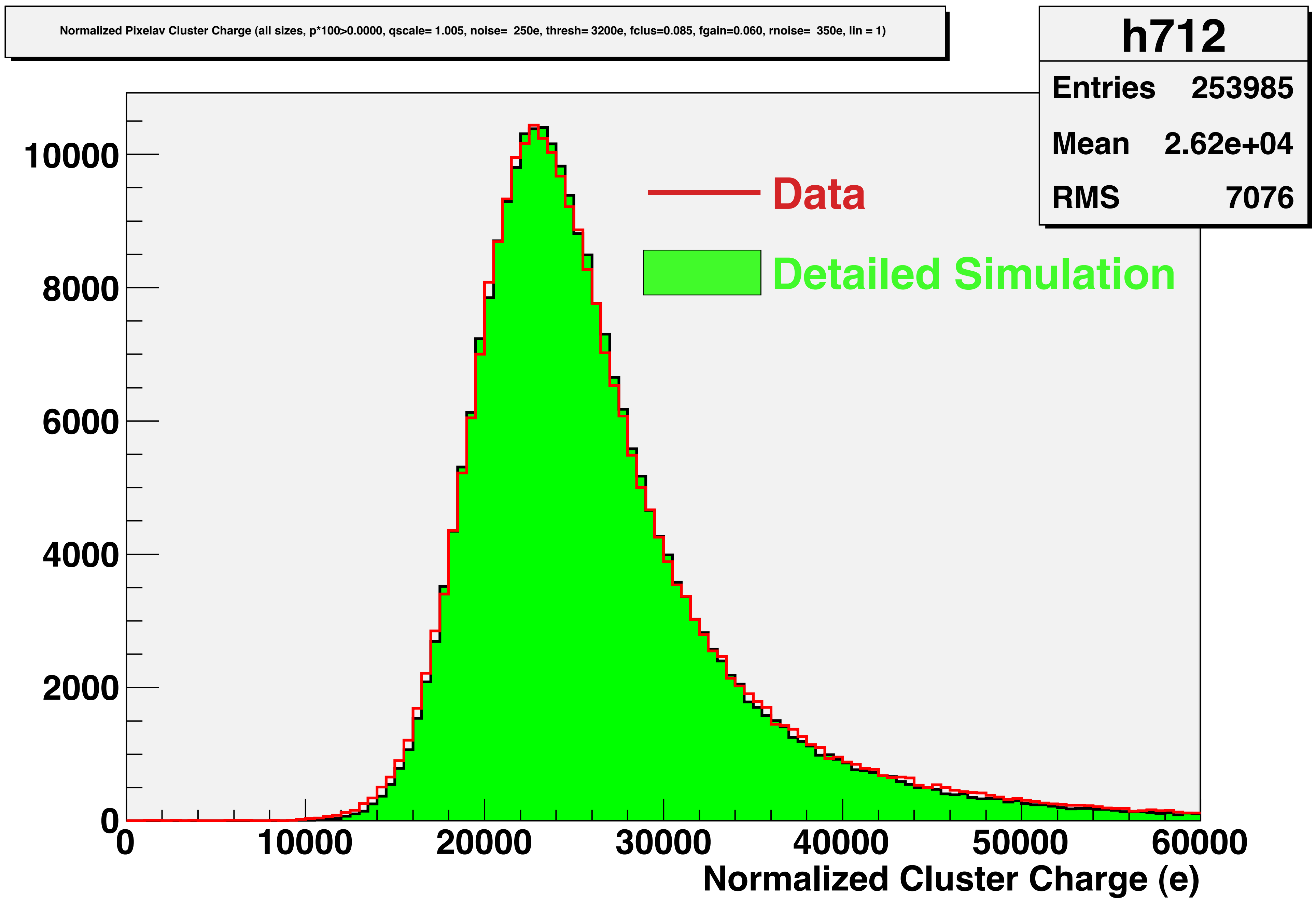


- secondary e- energies from 8 eV to 1 MeV
- highest energy secondaries travel 1-2 mm
- most emitted at 90deg wrt primary
- plural scattering + magnetic curvature
- range/dEdx from NIST Estar calculations
- average of 1 eh pair for each 3.68 eV of deposited ionization

$$\cos\theta_{ep} = \sqrt{\frac{T_e}{T_e + 2m_e}}$$

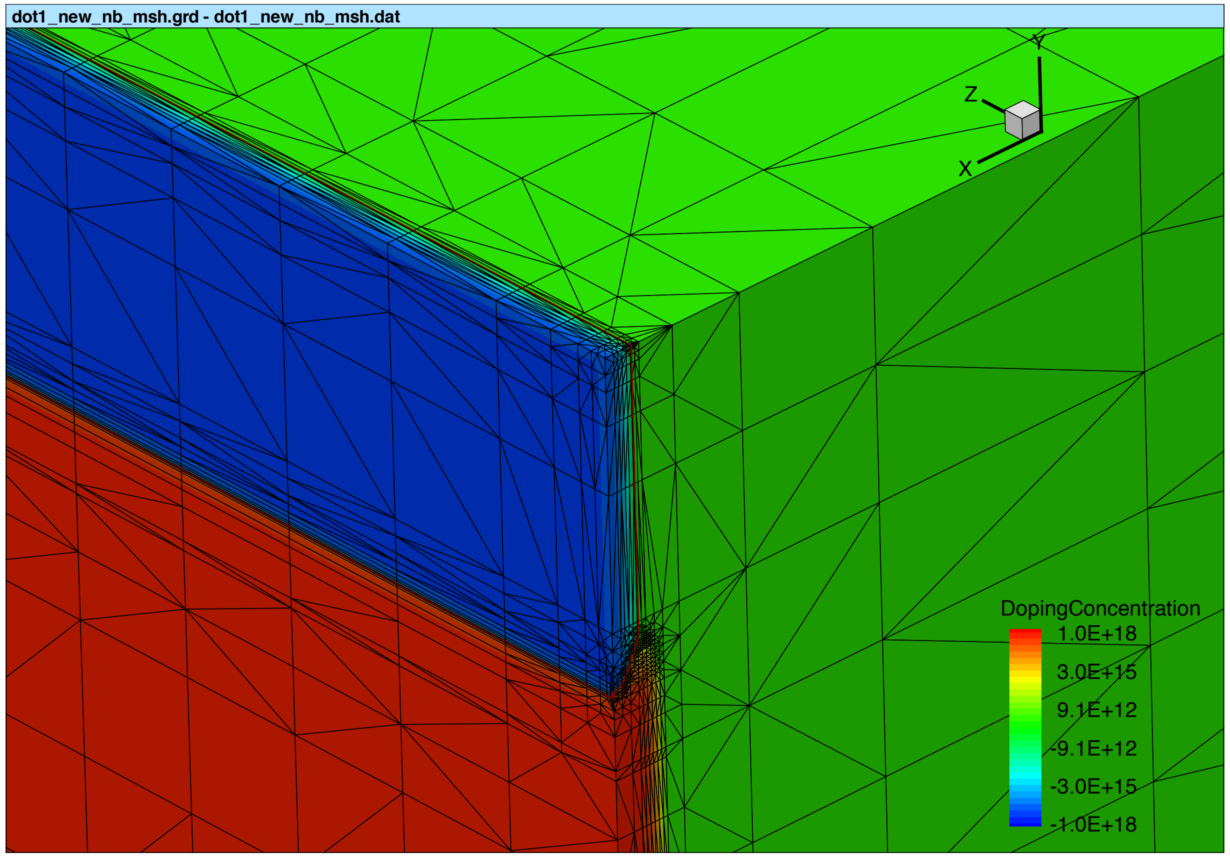


The high energy secondaries ["delta rays"] produce a "tail" at large charge. Plotting Q/track length for a sample of real pixel clusters

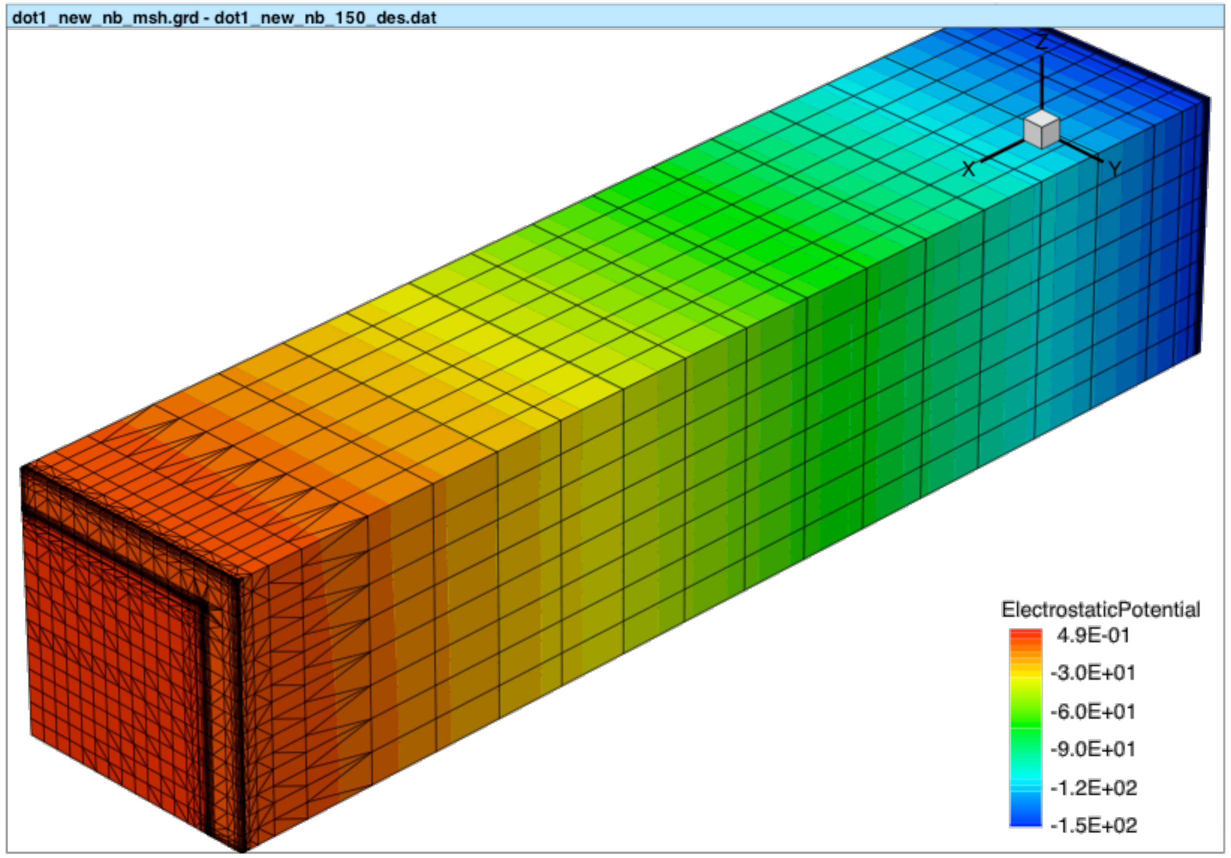


- Model describes observed cluster charge distribution very well
- Charge scale [no free parameters] agrees with measured distribution to within a few x 0.1%

- Electric field calculation: uses TCAD 9.0 software
  - simultaneously solves Poisson and carrier continuity eqs
  - includes lots of semiconductor physics (including SRH)
  - simulate 1/4 (1/2) pixel cell to keep mesh size ~17000 (25000) nodes. This requires 4-fold (2-fold) symmetry.
  - no process simulation, use MESH w/ analytic doping profiles to generate grid and doping files



doping profiles



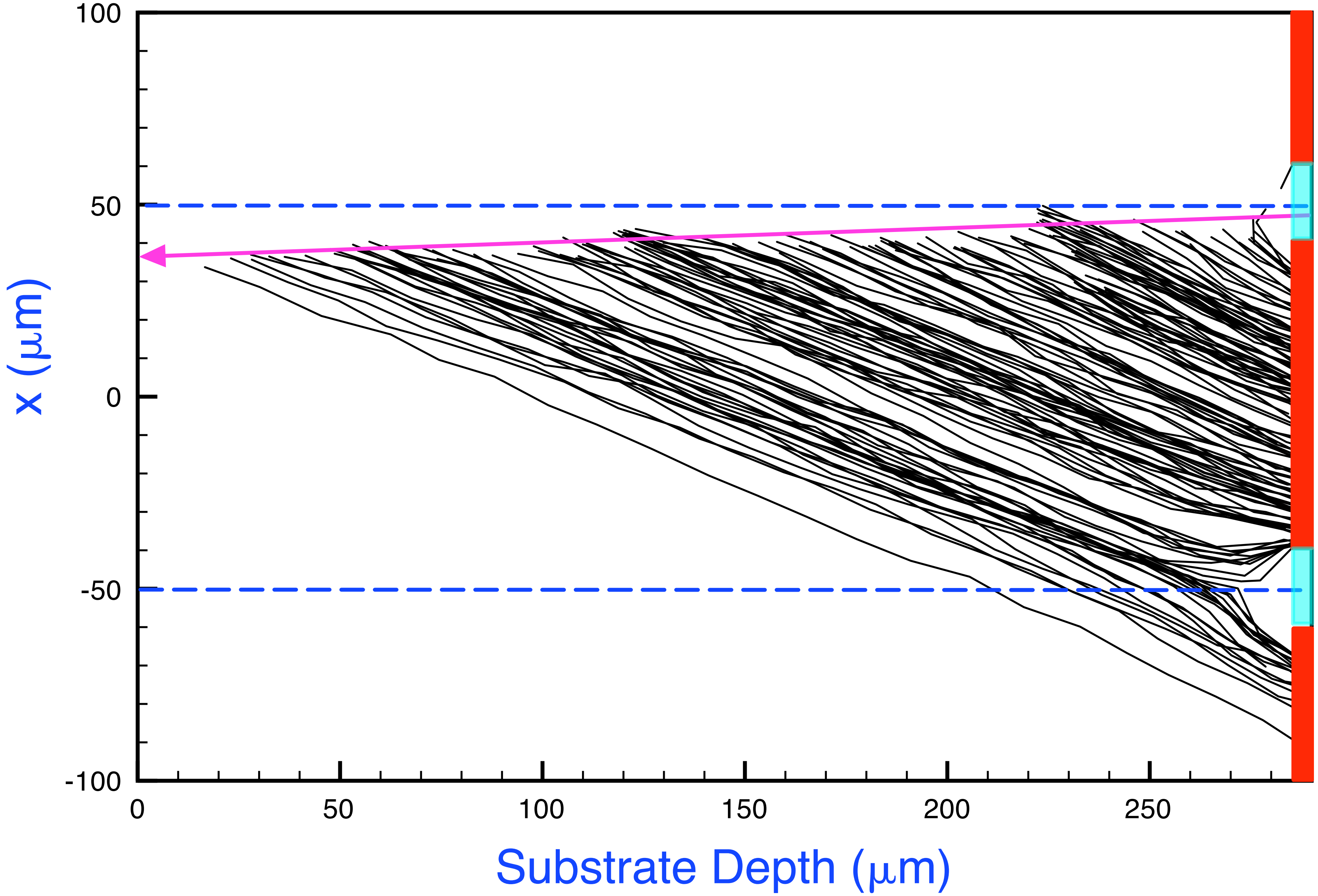
potential distribution

- Transport calculations are done by integrating the fully saturated equation of motion for the carriers [in time]

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

  - each carrier is described by 4-vector (t,x,y,z) [useful for trapping]
  - 5th-order R-K technique w/ adaptive step-sizing is vectorized for x86-64 and ppc processors
  - transport done in full 3D for e /h [new pix/str], or e+h [irr pix/str]
  - incorporates diffusion + trapping (Llubjana trapping rates)
  - signal induced from displaced, trapped charge is calculated from segmented parallel plate cap. model
  - special versions to consider time-dep response functions [eg deconvolution mode for CMS strips]
- Electronics Simulation:
  - includes leakage current and electronic noise
  - readout chip thresholds
  - readout chip analog response from measurements
  - ADC digitization

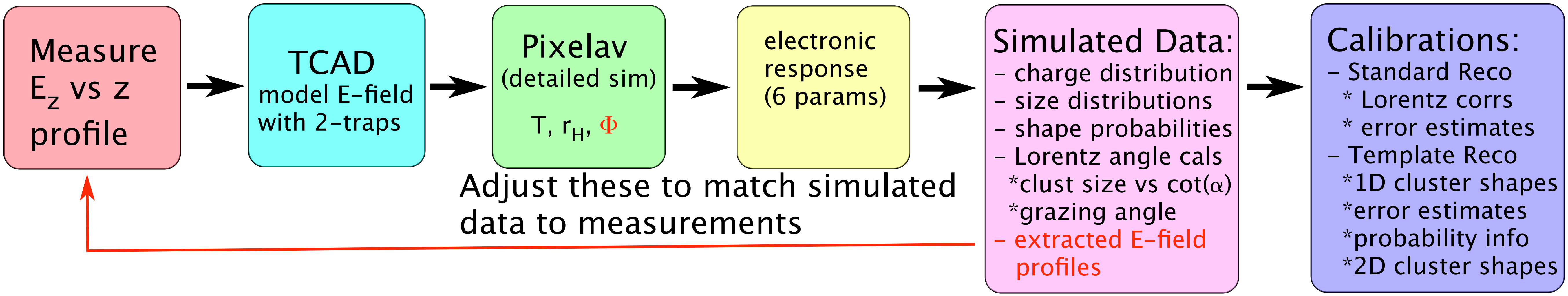
Pixelav Simulated Custer



- simulation transports only 1/10 carriers to save time
  - ✴ the charge fluctuations are large enough that the statistics are unaffected by this
- Figure above is even sparser to aid clarity.

Pixelav in Production

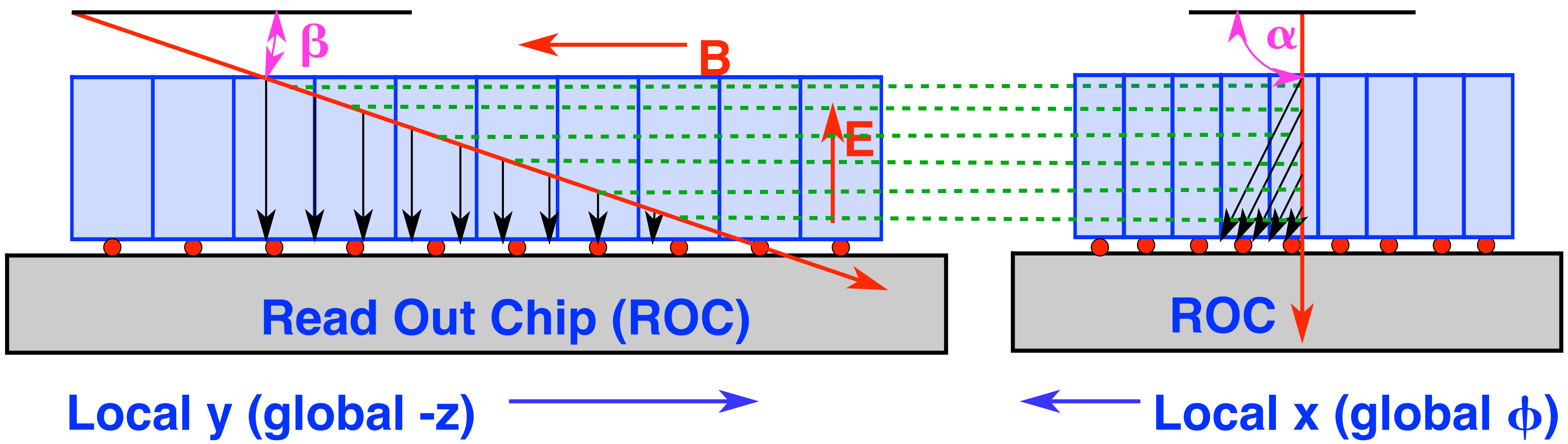
The TCAD+Pixelav simulations are tuned to measured distributions



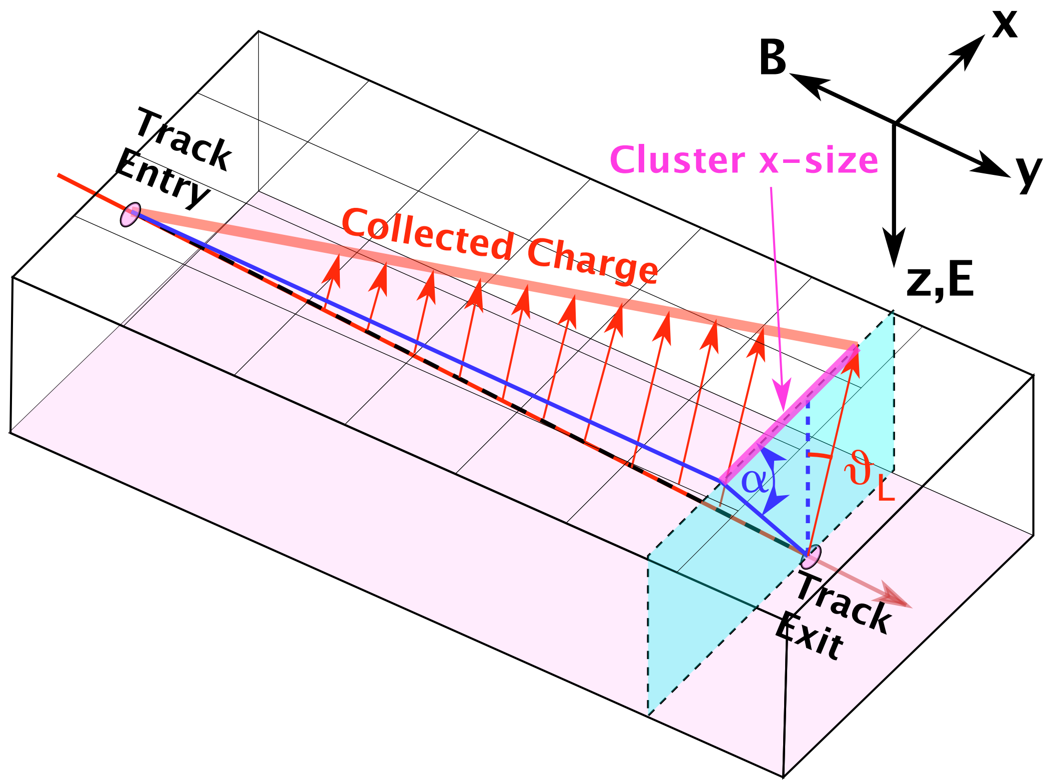
- E-field profiles are extracted from data and compared with simulation
  - adjust TCAD sensor modelling to reproduce measured profiles
- Cluster charge profiles are extracted from data and compared with simulation
  - adjust pixelav trapping parameters to model Q vs depth
- Tuned simulations are used to calibrate the hit reconstruction
  - 1D cluster shapes for the “template algorithm”
  - Lorentz drift corrections for the “generic algorithm”
  - Error estimates for both algorithms
  - 2D cluster shapes for realistic CMSSW simulation re-weighting

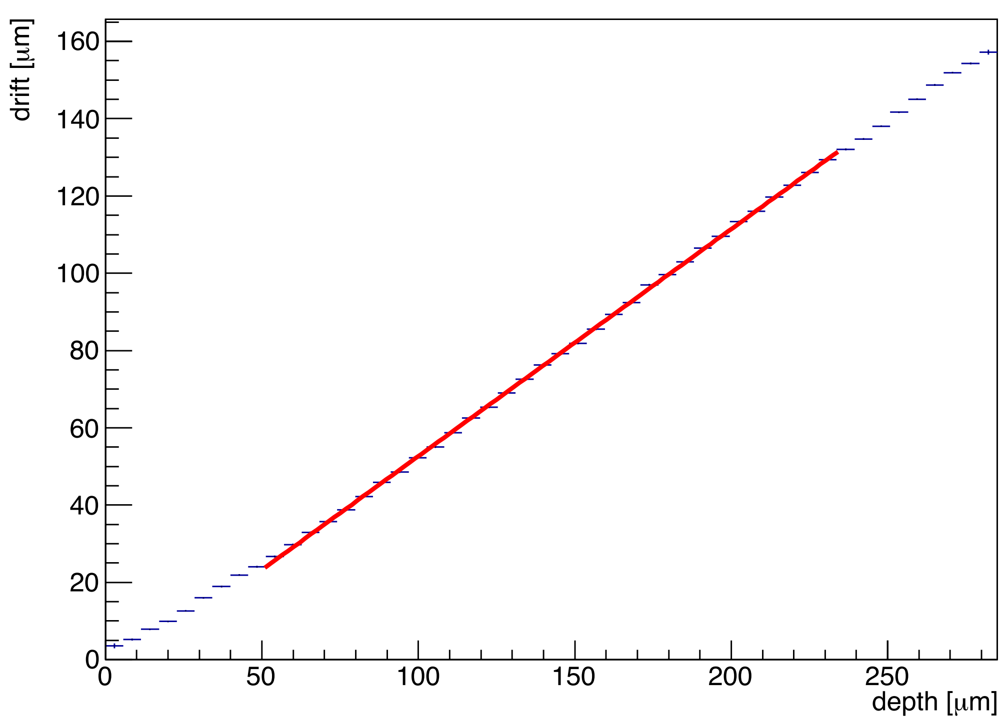
Lorentz Angle Calibration

Drift vs depth [grazing angle technique] was developed by UniZ colleagues to calibrate the Lorentz angle



Accumulate the charge centroid [drift] vs depth for a sample of highly inclined tracks. The angle is the average Lorentz angle



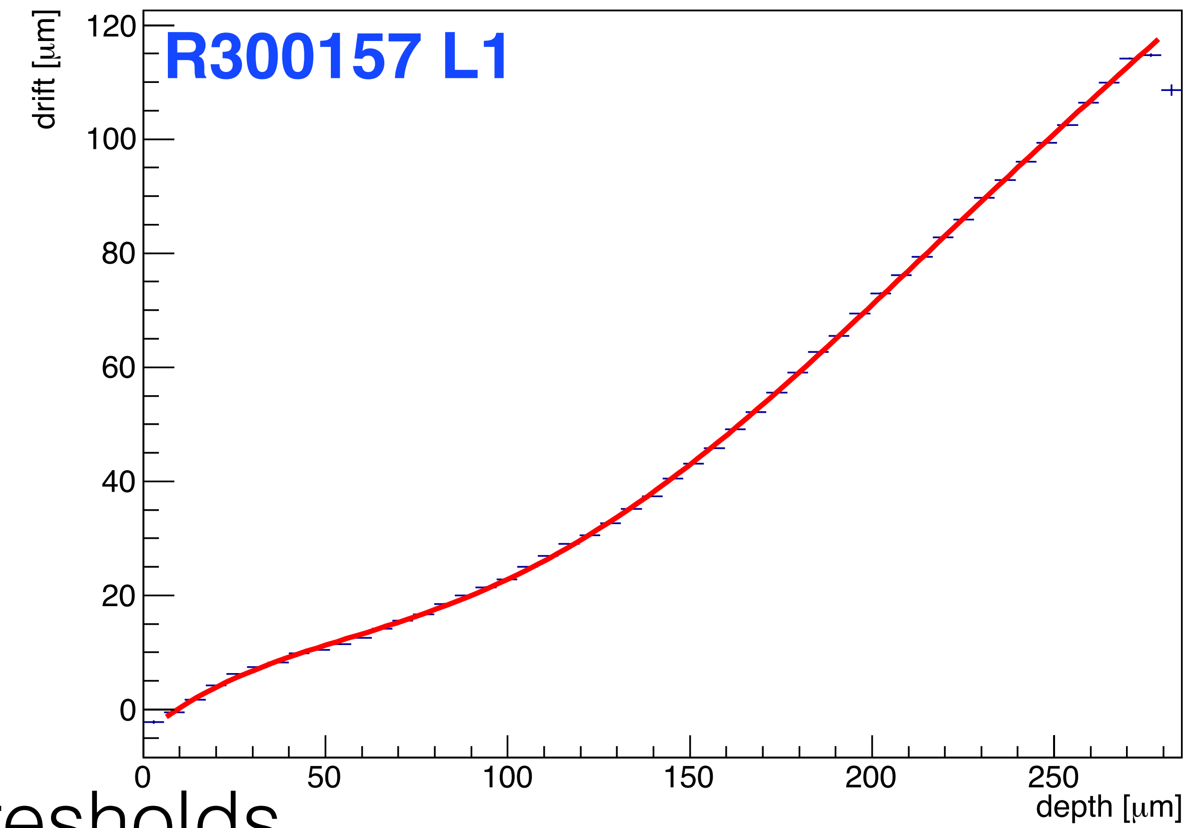


E-Field Measurement and Template/LA Calibration

Take our drift (x) vs Depth (D) data, fit to a polynomial [5th order] and then calculate a local slope [Lorentz Angle] vs D. We then convert it to an E vs D curve from the expression

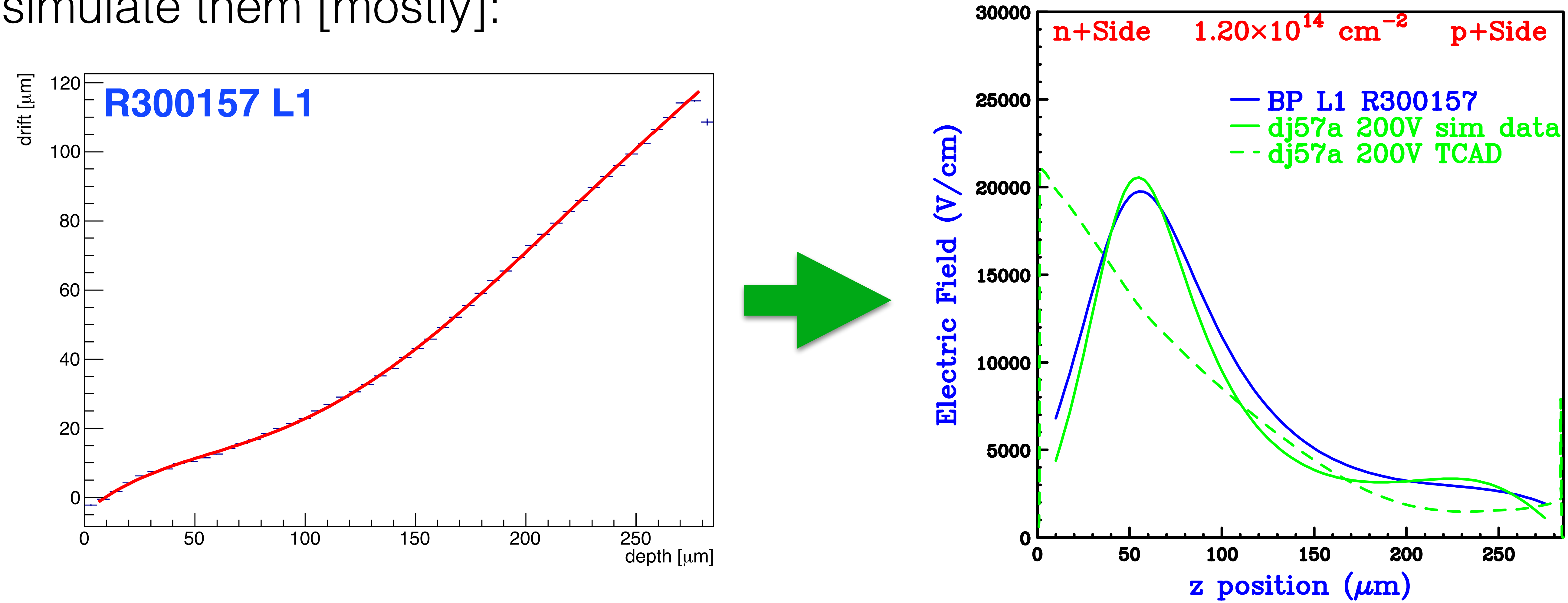
$$\tan\theta_L = \frac{dx}{dD} = r_H \mu(E) B_y$$

$$\rightarrow E = \mu^{-1}\left(\frac{1}{r_H B_y}\frac{dx}{dD}\right)$$



- depends upon the slope dx/dD
  - insensitive to alignment effects
- insensitive to the knowledge of thresholds
- insensitive to trapping [displacement is measured at fixed depth]!
- can be done at operating voltage: no need for bias scans
- extracts information that is sort of comparable to the simulated E-field
  - still need to simulate the extracted fields in this procedure
- Q vs D distributions can then be used to independently adjust the trapping rates for e/h

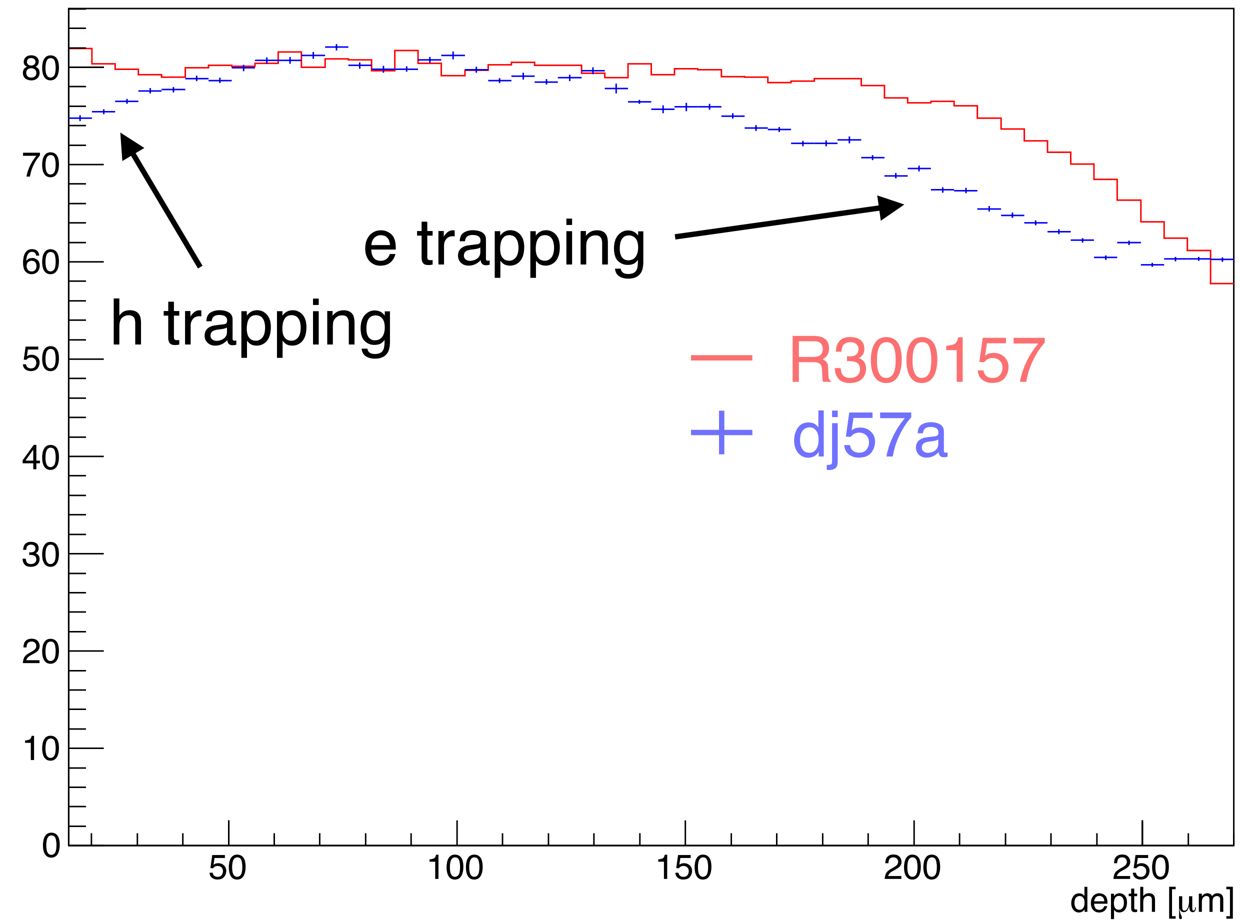
The extracted electric field profile is distorted by focusing near the n+ implant and other systematic effects. The good news is that we can simulate them [mostly]:



- Run 300157 was taken after 11.8 fb^{-1}: $\Phi_Q = 1.2\text{x}10^{14}$ cm^{-2}
  - the neutron equivalent flux [0.6 hardness] $\Phi_{eq} = 0.72\text{x}10^{14}$ cm^{-2}
  - the electric field is well described by our old model dj57a?
    - it was from a sensor that had been exposed to $\Phi_{eq} = 2\text{x}10^{14}$ cm^{-2}

Trapping Measurement

Compare the measured depth profile with the simulated profile



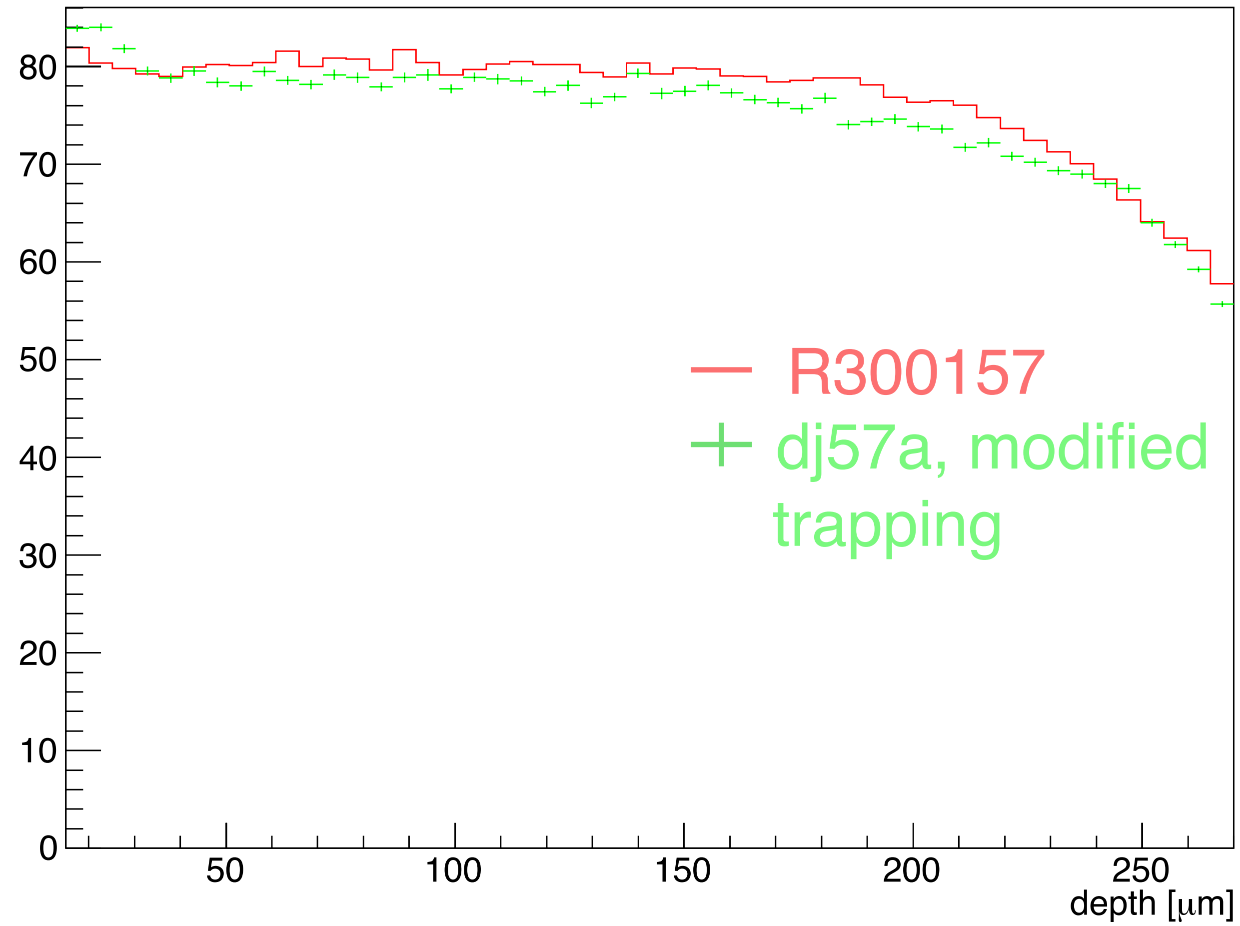
The trapping rates for e and h are both too large!

How much trapping do we expect for $\Phi_Q = 1.2\text{x}10^{14}\ \text{cm}^{-2}$?

In our test beam models, the trapping rates should scale as $0.8\Phi_{eq} = 0.48\Phi_Q = 0.6\text{x}10^{14}\ \text{cm}^{-2}$?

Trapping Measurement

Simulate the dj59a E-field with trapping rates corresponding to 0.6x10^{14} cm^{-2}



- The electric field is evolving faster [differently] than expectations from the beam test models
- Trapping rates appear to be evolving according to the fluence calculation with a hardness factor of 0.6
- The slower evolution of the trapping rates has important consequences for the longevity of the detector

Cluster Reweighting Algorithm

CMSSW simulation produces an array P[ix,iy] of pixel charges for a simulated hit at local coordinates (x,y) and local track angles (cotα,cotβ)

- use 2D template objects to interpolate arrays at (x,y) and (cotα,cotβ)
  - G[ix,iy] - the average cluster shape for an undamaged idealized sensor with a uniform field [match LA to actual V_{bias}]
  - T[ix,iy] - the average cluster shape for a damaged physical sensor
- for each element G[ix,iy] > min value, calculate R[ix,iy]= T[ix,iy]/G[ix,iy]
- if T[ix,iy] > min value and G[ix,iy] < min value, calculate R[ix,iy]= T[ix,iy]/G[ixn,iyn] where ixn, iyn is the nearest pixel with a denominator G[ixn,iyn] > min value (store ixn and iyn too).
- if there are P[ix,iy] for which R[ix,iy] is undefined, set R[ix,iy] to value of nearest defined pixel.
  - handles large delta rays
- calculate output charges O[ix,iy]= R[ix,iy] P[ix(n),iy(n)]

This produces output clusters with the correct average shapes but reflecting the charge fluctuations of the input cluster. For example (at fluence $\Phi=1.2\text{x}10^{15}\ n_{eq}/cm^2$, bias voltage 600V, temperature 263K):

Input clust P[ix,iy]

0	0	0	0	0	0	0	0	0	0	0
0	0	9590	7380	0	0	0	0	0	0	0
0	0	0	9140	12500	11140	11790	12330	1750	0	0
0	0	0	0	0	0	0	0	0	0	0

Input templ G[ix,iy]

0	0	0	0	0	0	0	0	0	0	0
0	93	10696	5584	722	23	0	0	0	0	0
0	0	884	7398	12337	12963	12997	13042	1375	0	0
0	0	0	0	0	0	0	0	0	0	0

Output templ T[ix,iy]

0	0	0	0	0	0	114	159	87	23	0
0	0	8538	5871	754	263	527	495	208	41	0
0	0	48	2856	7512	7069	5894	4705	669	51	0
0	0	0	0	0	125	477	582	258	40	0

Weights R[ix,iy]

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.80	1.05	0.06	0.02	0.04	0.04	0.00	0.00	0.00
0.00	0.00	0.00	0.39	0.61	0.55	0.45	0.36	0.49	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.04	0.00	0.00	0.00

The output cluster is the product of the input cluster and weights

Input clust
P[ix,iy]

0	0	0	0	0	0	0	0	0	0	0
0	0	9590	7380	0	0	0	0	0	0	0
0	0	0	9140	12500	11140	11790	12330	1750	0	0
0	0	0	0	0	0	0	0	0	0	0

Weights
R[ix,iy]

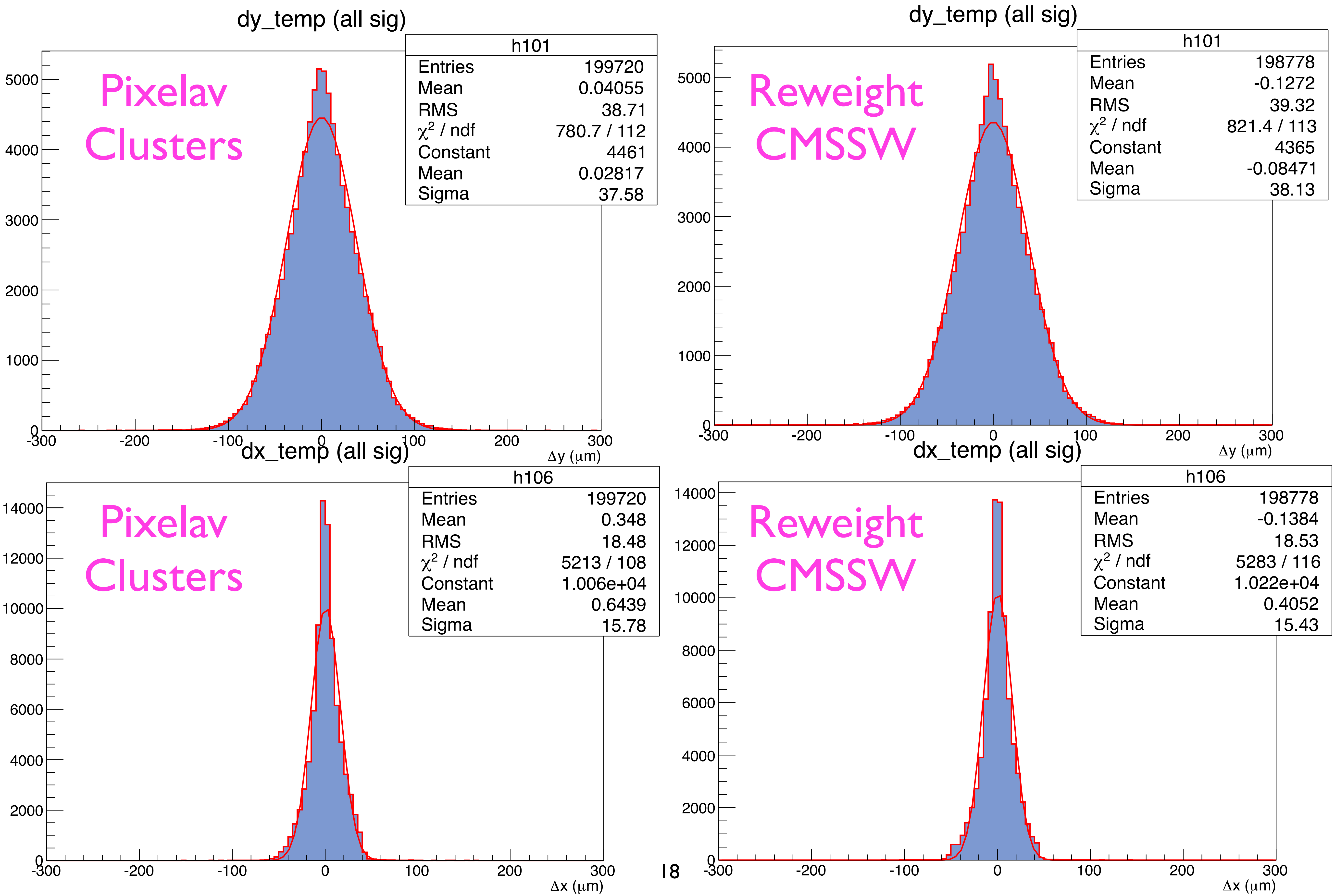
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.80	1.05	0.06	0.02	0.04	0.04	0.00	0.00	0.00
0.00	0.00	0.00	0.39	0.61	0.55	0.45	0.36	0.49	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.04	0.00	0.00	0.00

Output clust
O[ix,iy]

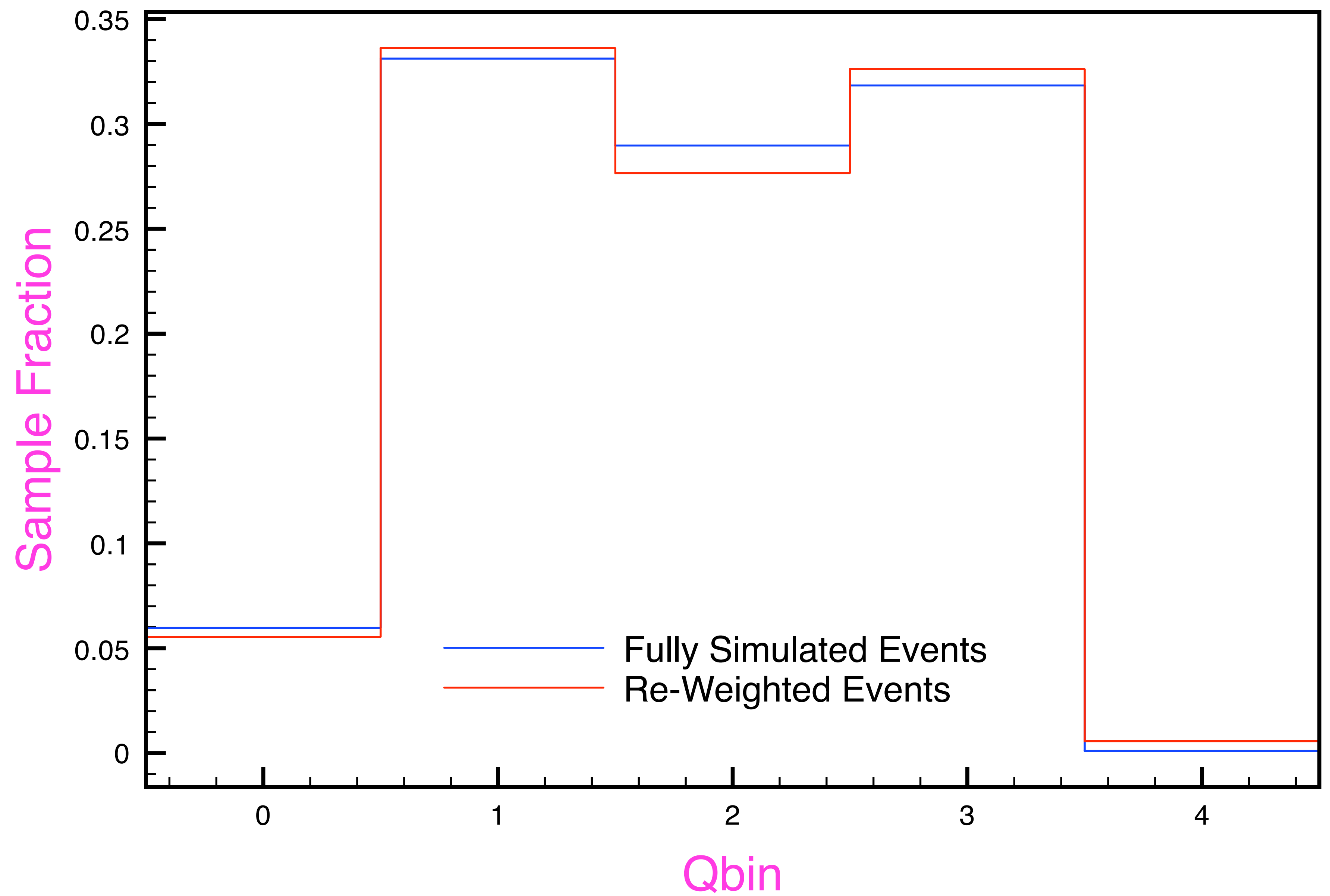
0	0	0	0	0	0	0	0	0	0	0
0	0	7773	7878	764	226	478	468	0	0	0
0	0	0	3583	7728	6169	5429	4516	865	0	0
0	0	0	0	0	0	433	550	0	0	0

- Technique captures the larger induced signals from trapped charge
  - most evident for the longer drifting carriers on the RH side
  - these are mostly less than the ROC threshold on the neigboring pixels but also contribute to the pixels with collected carriers.

Fully simulated $\Phi=1.2\times10^{15}$ n_{eq}/cm^2 clust vs reweighted CMSSW-like clust



Fully simulated Φ=1.2x10^{15} n_{eq}/cm^2 clust vs reweighted CMSSW-like clust



- x/y resolutions of Template Reco are within 2% of fully simulated values
- Binned charge distributions also agree well

Summary

- Pixelav simulation was originally developed to interpret beam test results and is now a key part of the pixel hit reconstruction algorithm
  - ✳ it was developed to model sensor physics as accurately as possible
  - ✳ speed was a secondary consideration, it was never intended for the production simulation of CMS pixel hits
- CMS has developed techniques to tune the Pixelav simulation parameters from collision data
  - ✳ this is important because beam tests are performed under different conditions [eg w/ optimally annealed detectors] with possibly different detector materials than normal operations
    - ‣ models established in beam tests require additional tuning
- The (fast) CMS production pixel simulation is incorporating a reweighting algorithm to include more realistic modelling and irradiation effects
  - ✳ the simulation and the reconstruction will be synchronized by using templated cluster shapes generated from the same models
    - ‣ simulated events will reflect correct resolution effects using the full reconstruction chain