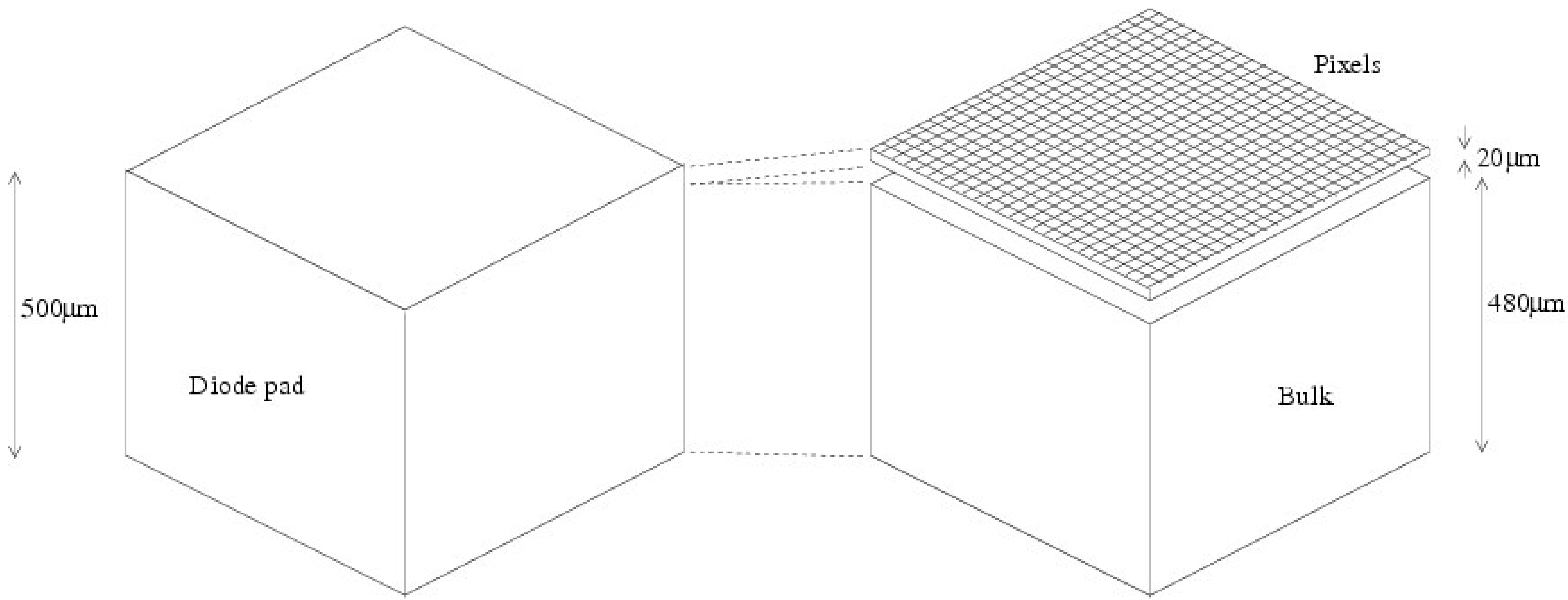
MAPS "Digitisation"

- In principle change from simulation output to "raw" information equivalent to that seen in real data
  - Not "reconstruction", which is non-linearity corrections, clustering, etc; further downstream
- Most of the CPU time in simulation is tracking of physical particles through complicated detector structure
  - Digitisation (and reconstruction) usually much faster
- Ideally would keep things flexible
  - Ability to redo digitisation without rerunning full simulation
  - Allow fast change of parameters for identical events; pixel size, threshold level, noise rate, etc.
- Also want easy comparison with diode pad option
  - Keep diode pad structure, currently $1\times1\text{cm}^2$ pads
  - Within this, subdivide into pixels, assume $50\times50\mu\text{m}^2$ pixels (200×200/pad)

Digitisation concept

- Divide diode pad into pixels (epitaxial layer) and ”bulk pads”
- Keep energy deposits in bulk pads so diode pad energy can be reconstructed in digitisation step for comparison

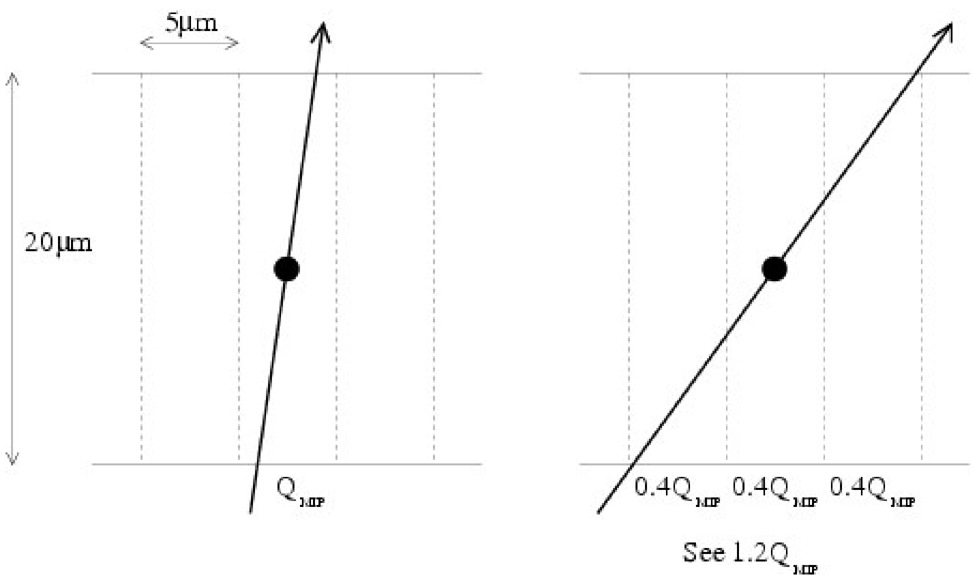


Digitisation input

- Input is the main issue
  - "Standard" interface is LCIO simCalorimeterHit objects
  - Contain energy, centre position of "cell" and two 32-bit int "cellIDs"
- The discussion is on what the "cells" are
  - To be flexible for pixel size, need finer pixellation here
  - Obvious subpixel size would be Giulio's charge mapping size = 5μm
  - Leads to 30M diode pads × 4M subpixels/pad ~ 10^{14} subpixels!
  - Certainly need 64 bits (up to 4×10^{18}) to store subpixel cellID number
  - Positions are floats; may hit precision limit trying to store coordinates ~1m to an accuracy of ~1μm
- If this can be implemented in GEANT4 simulation
  - Energy would then be raw deposited energy ∝ charge in subpixel
  - Digitisation would apply Giulio's mapping to spread charge out
  - Add subpixels in groups to required pixel size
  - Gives charge in each real pixel to compare to threshold

Digitisation input (cont)

- GEANT4 implemention may be hard
  - They never expected so many active cells and such small sizes
- An alternative would be to make simCalorimeterHits with "cells" equivalent to diode pads
  - Have position not as centre of cell but exact location of energy deposit
  - Would need access to geometry database when doing digitisation to translate position into Giulio subpixel
  - Then apply mapping and continue as before
- Complication due to aspect ratio of subpixels
  - Epitaxial layer ~ 20µm > subpixel size ~ 5µm
  - No direction information is stored; loss of imformation

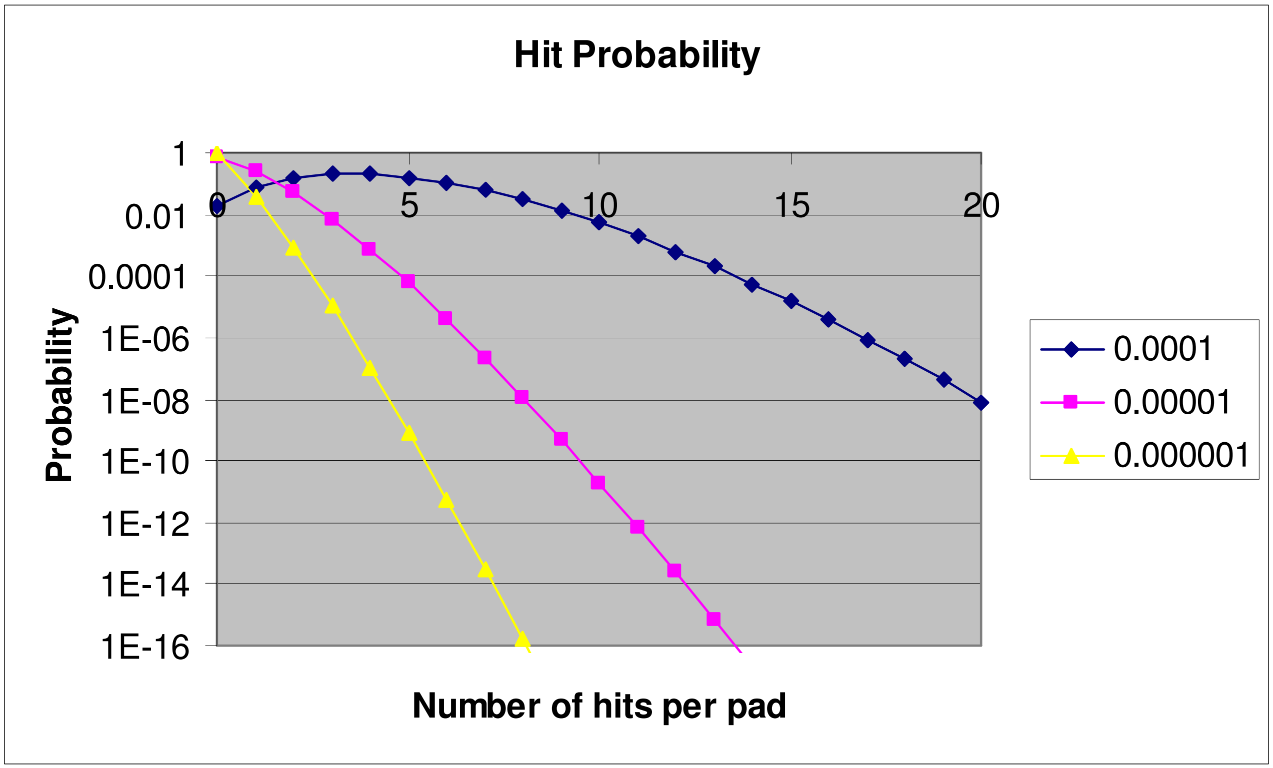


Noise simulation

- Seems trivial to do in principle
  - But practical implementation can be tricky due to 10^{12} pixels
  - Only point where all pixels need to be considered
- For pixels with no charge, probability of a noise hit is constant
  - Assuming no coherent noise, crosstalk, bad channels, etc.
  - No point in generating Gaussian noise and imposing threshold
  - Simply say hit or no hit with correct probability
  - Probability ~ 10^{-5} (or less); total of ~10^{7} noise hits in calorimeter
  - Not trivial to get high precision random number generator at this level; may need to take square-root and use two random numbers
  - Would need ~ 10^{12} random number calls; could be slow
- Better to pick number of noise hits from binomial distribution
  - Work at diode pad level: expect average of 40000×10^{-5} ~ 0.4 hits/event
  - Only need one high precision number for binomial per pad and then two integer values for x and y within pad: total of ~ 5×10^{7} random number calls

Binomial noise distribution

- Assume will generate ~10^6 events
  - Equivalent to $30M \times 10^6 \sim 3 \times 10^{13}$ pads
  - Need probabilities down to ~10^{-13}
  - (Probably noise beyond that due to coherent effects or bad channels anyway)
  - Equivalent to a maximum of ~15 noise hits per pad



Charge noise simulation

- Pixels with physical charge deposited need special treatment
  - Not a fixed probability; depends on charge
  - Noise can push total charge above or below threshold
- Here, do more standard treatment
  - Add Gaussian noise charge to real charge
  - Check against threshold and flag if above
- Can do binomial noise in whole calorimeter first
  - No bias if noise result then discarded for pixels with charge
- Presumably number of hit pixels much less than 10^7/event
  - CPU dominated by noise hit implementation
- Note, noise cannot be done once only in this scheme
  - Need to redo when changing pixel size or threshold, as well as noise rate.

Output of digitisation

- Conceptually, output is a list of hit pixels
  - Need to define how this is to be implemented
- Propose LCIO object (TBD) containing:
  - Diode pad cellID
  - Number of pixels hit in pad
  - List of local int x,y values within pad for hit pixels
- Non-standard LCIO objects cannot be (easily) displayed
  - Propose pad-like output objects also (or contained in the above?)
- Standard LCIO calorimeterHit objects, containing:
  - Diode pad cellID
  - "Energy" $\propto$ number of pixels hit
  - Position of centre of pad
- N.B. Could also have standard diode pad output in very similar format; allows easy comparison

Random numbers

- Digitisation must be reproducible
  - Seed random number generator(s) for every event
  - Seed must be stored in LCIO Mokka output
  - Implies random number generator(s) tied to LCIO
- This is also needed for standard diode pad digitisation
  - Noise must be added for each pad
  - Has to be reproducible for same reasons
- Does LCIO and/or Marlin have this facility?
  - If not, must be added
  - Could be some delay in getting this implemented