cern.ch/allpix-squared

Frequently asked questions and frequently encountered problems

Paul Schütze
Simon Spannagel
Håkan Wennlöf

9/5 -22

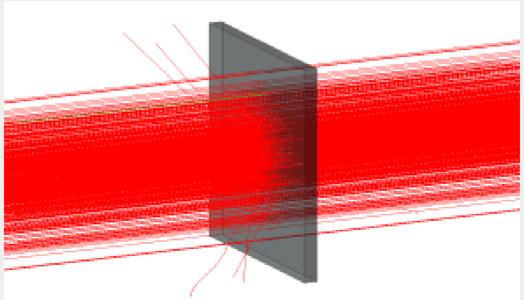
Outline

- Simulation flow can be divided into 3 main parts;
  - Configuration
  - Runtime
  - Analysis
- We will go through **frequently asked questions** and **frequently encountered problems** (and their **solutions**) for each part

Configuration stage FAQ and FEP

Constructing geometry – things to consider

- Make sure that your incident particles actually hit the sensor(s) the way you want
- [VisualizationGeant4] is your friend here; opens the QtGui for Geant4
  - */run/beamOn* is a handy command to see where particles from a source actually end up, compared to the sensor
  - This will also help figuring out the desired source directions
- Passive material: predefined shapes are “box”, “cylinder”, and “sphere”, which is useful for **material approximations**. GDML file import is also available now, for more advanced structures

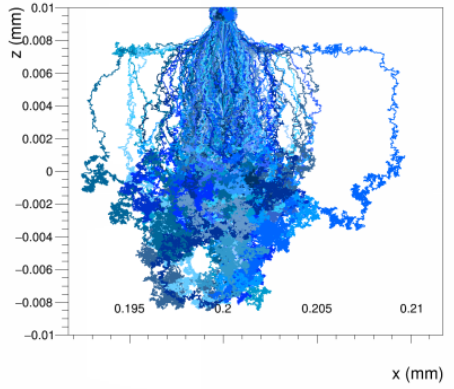


Example result from */run/beamOn 1000* in the QtGui command line

- When importing external electric fields: the collection electrode size definition is important!
  - When **not** doing transient simulations the *implant_size* in Allpix² should be **slightly larger** than in the imported field, to not lose charge due to final diffusion steps

When to use which propagation and transfer?

- A propagation module propagates charges through the sensor
- [ProjectionPropagation] is **fast**, at the cost of precision. Can only be used with **linear** electric fields. Can be used for simple sensor approximations
- [GenericPropagation] is **versatile** and highly customisable. The best choice for **precision**, if transient pulses are not required. Can produce detailed linegraphs of charge movement
- [TransientPropagation] calculates the **induced charge** on neighbouring implants in each step. Creates **current pulses**. Requires weighting potential. Relatively **slow**, but gives a lot of information
  - Allpix² comes with a *generate_potential* tool, that can generate weighting potentials from model descriptions



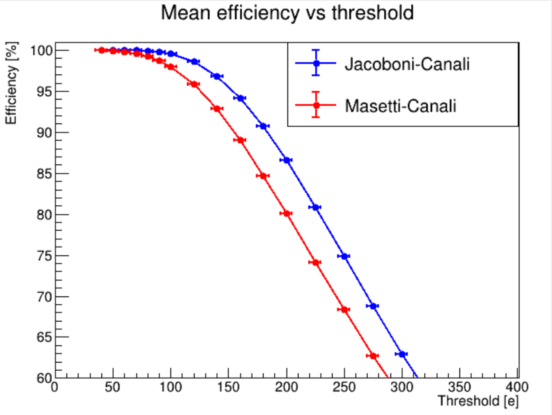
Example linegraph from [GenericPropagation]

When to use which propagation and transfer?

- A transfer module transfers propagated charge to the readout electronics simulation stage (i.e. digitisation)
- [SimpleTransfer] makes a direct mapping of propagated charge to the nearest pixel
  - Can also be set to collect **only from the implant**
    - Important: set the **max_depth_distance** slightly larger than the collection implant depth in an imported field
- [InducedTransfer] **approximates the induced charge** on neighbouring pixels, by taking the difference in weighting potential between charge carrier start and end position. Requires weighting potential
  - Faster than using [TransientPropagation], and useful when a large fraction of particles get trapped (but still induce charge)
- [PulseTransfer] is required to transfer induced **pulses** created by [TransientPropagation]
  - Hence, **use [PulseTransfer] if [TransientPropagation] is used**
  - [PulseTransfer] will also create approximative pulses, if no pulse information is available (e.g. if [GenericPropagation] is used)

Propagation settings

- Propagation modules allow for different **recombination models** and **mobility models**
- Recombination models simulate charge carrier **finite lifetime** due to sensor material doping
  - The combined "**srh_auger**" model is recommended, as it applies to both low and high doping concentrations
- There are many mobility models available for silicon, and they basically split into **doping-dependent** and **doping-independent**
  - Can have a **significant impact** on observables
  - The recommendation on what to use depends a lot on the simulated device and available information (electric field strength, doping concentration, temperature, ...)
  - The "extended Canali model" (***masetti_canali***) is a good choice when an electric field and a significant doping concentration are present
  - It is also possible to use a **custom model**



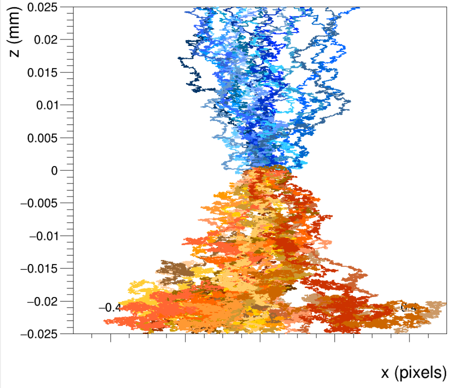
Example study comparing mobility models

How to choose a specific particle source?

- Allpix^2 comes with several pre-defined particle sources
  - [DepositionReader]
  - [DepositionCosmics]
  - [DepositionPointCharge]
  - [DepositionGeant4]
- All useful for different things
  - [DepositionReader] takes in externally generated energy depositions into Allpix^2
  - [DepositionCosmics] uses the Cosmic-ray shower generator (CRY) and Geant4 to generate and transport cosmic rays (see e.g. talk by Max)

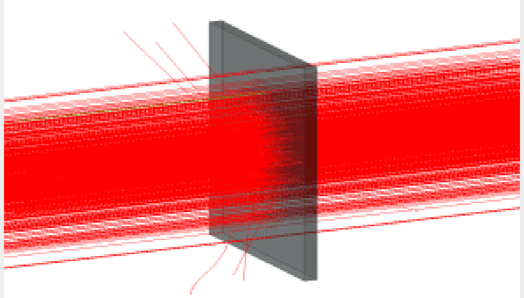
Particle source: [DepositionPointCharge]

- Deposits **user-defined charge** in a single **point** or along a **line**
- Thus removes statistical fluctuations in deposition – very useful for detailed sensor behaviour investigations!
- Two possible modes: Point and MIP
  - “Point” deposits charge in a single point in 3D space
  - “MIP” deposits charge along a straight line
- Deposits a given number of charges; total number for “point”, and charge per length for “MIP”
- Three possible deposition models: “fixed”, “spot”, and “scan”
  - “Fixed” deposits charge in the same position each time
  - “Spot” smears the position over a Gaussian spot
  - “Scan” moves the deposition position homogeneously over **a full pixel cell** (in the middle of the pixel matrix)



Single point deposition in pixel centre with subsequent drift of electrons and holes

Particle source: [DepositionGeant4]



- Generates **Geant4 particles** that can pass through the sensor and deposit energy, taking physical stochastic effects into account
  - Simulates particle decays, secondaries, Landau fluctuations, ...
- Predefined source shapes: beam, point, square, sphere
  - **Beam** is the default; can set many of the beam parameters
    - Particle flavour, beam direction, beam size and divergence, beam profile uniformity and energy, ...
  - Square and sphere: homogeneous emission over surface. Particle directions random by default, but can be set
    - Field inside sphere is isotropic; useful for space application studies
  - Particle flavours can also be radioactive elements; ^{55}Fe, ^{241}Am, ^{90}Sr, ^{60}Co, ^{137}Cs available, with simulated decays
- Can also do **Geant4 macro** for the General Particle Source
  - Can thus define **any** radioactive source, or any source shape
  - Also several different particles at once
  - Iron-55 photon example in backup slides. Link to GPS info here

Default and/or wrong units in configuration

- A common problem, with **big consequences**!
- The framework default units are:
  - Length: mm
  - Time: ns
  - Energy: MeV
  - Temperature: K
  - Charge: e (elementary charge)
  - Voltage: MV
  - Magnetic field: T
  - Angle: radian
- It is recommended to **always** specify the unit, to avoid mistakes
- The default unit might be unexpected (like MV), so **don’t rely on it!**

Mesh conversion – importing field and doping

- When importing fields from e.g. TCAD, the *mesh_converter* tool can be used
- In the mesh_converter configuration file, make sure to:
  - Select the correct **region** in the source file to convert
  - Select the correct **observable** and its matching **unit!**
    - Electric field is typically in V/cm, and doping concentration in /cm/cm/cm
    - Getting them wrong can have a **huge impact!** Using V/cm for a doping concentration in cm^{-3} gives a final doping that is a factor of 10 000 too low
    - Newer $Allpix^2$ versions require units to be given, but **beware** of older versions and their default units
  - Select the **orientation** of the field
    - Tip: enabling output plots in [ElectricFieldReader] and [DopingProfileReader] in the $Allpix^2$ simulation is useful for examining the results
- When loading doping profiles into $Allpix^2$, be mindful of the **depth setting** *(doping_depth)*
  - If the setting doesn't match the profile size, the profile will get stretched or squished in the depth direction, to fit the $Allpix^2$ module setting
- For electric fields, the field outside of the imported mesh is set to 0. For doping profiles, the concentration is kept the same as the **last point within the depth setting**

Mesh conversion – how many divisions?

- The number of divisions relates to the **granularity** of the field in Allpix^2
- The study below shows the effect of different settings on final observables, with an example field for a 20x20x10 μm^3 pixel

Mesh divisions	Mean cluster size	Resolution in x [µm]	Resolution in y [µm]	Efficiency [%]
100x100x100	2.696 ± 0.002	2.35 ± 0.01	2.37 ± 0.01	99.988 ± 0.002
100x100x50	2.817 ± 0.003	2.28 ± 0.01	2.29 ± 0.01	99.993 ± 0.002
100x100x10	2.693 ± 0.003	2.36± 0.01	2.38 ± 0.01	99.986 ± 0.002
300x300x100	2.696 ± 0.002	2.35 ± 0.01	2.37 ± 0.01	99.988 ± 0.002
100x100x100	2.696 ± 0.002	2.35 ± 0.01	2.37 ± 0.01	99.988 ± 0.002
50x50x100	2.695 ± 0.003	2.35 ± 0.01	2.37 ± 0.01	99.987 ± 0.002
20x20x100	2.930 ± 0.003	2.22 ± 0.01	2.24 ± 0.01	99.993 ± 0.002

- Changes in z (depth) have a larger effect than changes in x-y
- In x-y one can make the mesh coarser without losing out (for this particular sensor at least)
  - Having it finer does not slow down simulations however, so don’t be shy about using a fine mesh

Runtime stage FAQ and FEP

General questions and their answers

- "My simulation is taking too long"
  - Propagate more charges together
    - Impact? Good to **make a study** for your sensor type, to see impact
    - Example study on the right for ultra-thin sensor shows little impact on efficiency, but significant impact on resolution
  - Propagation time scales approximately **linearly** with the number of charges propagated together
  - To find which module takes a lot of time: run a few events and look at INFO output at the end
- "My RAM is filling up"
  - Propagate more charges together
  - Reduce buffer size *(buffer_per_worker)*
    - Impact? Takes longer, potentially

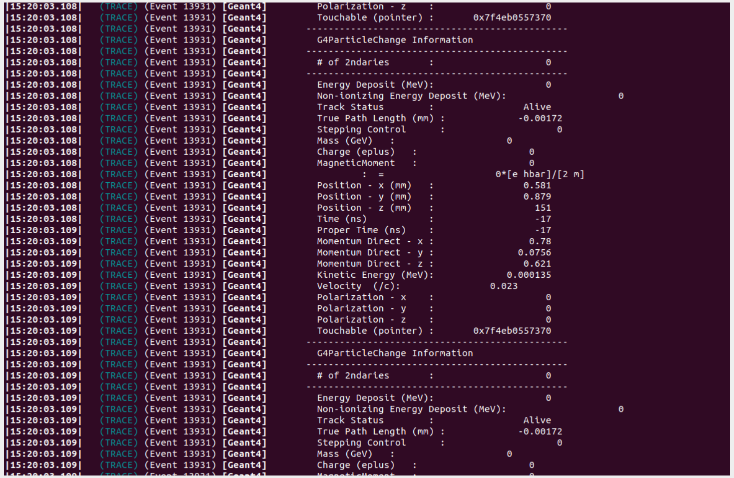
Max. number of charge carries propagated per step	Efficiency [%]	Resolution in x [µm]
1	99.990 ± 0.002	2.29 ± 0.01
5	99.988 ± 0.002	2.35 ± 0.01
10	99.988 ± 0.002	2.44 ± 0.01
25	99.982 ± 0.002	2.70 ± 0.01
50	99.975 ± 0.002	3.09 ± 0.01

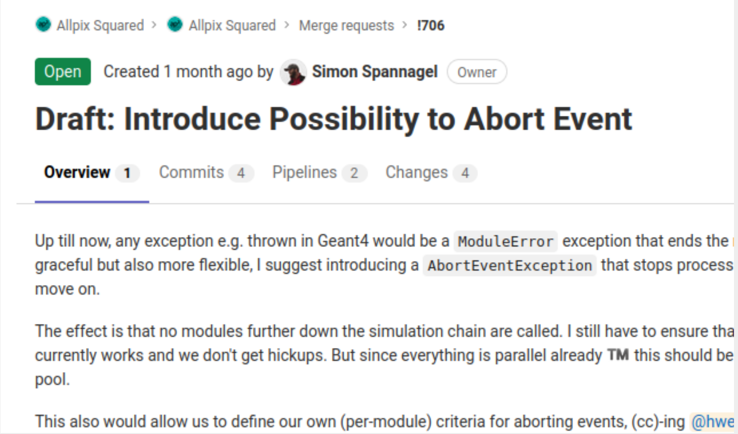
Impact of number of charge carriers propagated together, for a thin MAPS with small collection electrode. See talk by Sara for study details. (Expected signal most probable value of ~800 electrons)

```
|17:46:00.535|  (STATUS) Executed 8 instantiations in 33 seconds, spending 92%
of time in slowest instantiation GenericPropagation:dut
|17:46:00.535|    (INFO)  Module GeometryBuilderGeant4 took 0.0254188 seconds
|17:46:00.535|    (INFO)  Module DepositionGeant4 took 2.00758 seconds
|17:46:00.535|    (INFO)  Module ElectricFieldReader:dut took 0.395379 seconds
|17:46:00.535|    (INFO)  Module DopingProfileReader:dut took 0.0382851 seconds
|17:46:00.535|    (INFO)  Module GenericPropagation:dut took 30.8973 seconds
|17:46:00.535|    (INFO)  Module PulseTransfer:dut took 0.000669902 seconds
|17:46:00.535|    (INFO)  Module DefaultDigitizer:dut took 0.00576612 seconds
|17:46:00.535|    (INFO)  Module ROOTObjectWriter took 0.111498 seconds
```

Simulation stuck in charge deposition stage

- Lately, simulations of multiple sensors have been known to get stuck using the [DepositionGeant4] module
- This seems to be due to a single low-energy secondary particle created by Geant4, and is a Geant4 problem
- Possible solution: We are implementing the possibility to **abort events**
- Temporary workarounds:
  - Change the range cut
  - Try a different physics list
  - Change the random seed





Simulation uses a lot of memory in the propagation and transfer stages

- Using [TransientPropagation] and [PulseTransfer] can lead to large amounts of RAM being used, in events with a **large energy deposition**
  - These events appear randomly, especially in thick sensors and low-momentum incident hadrons

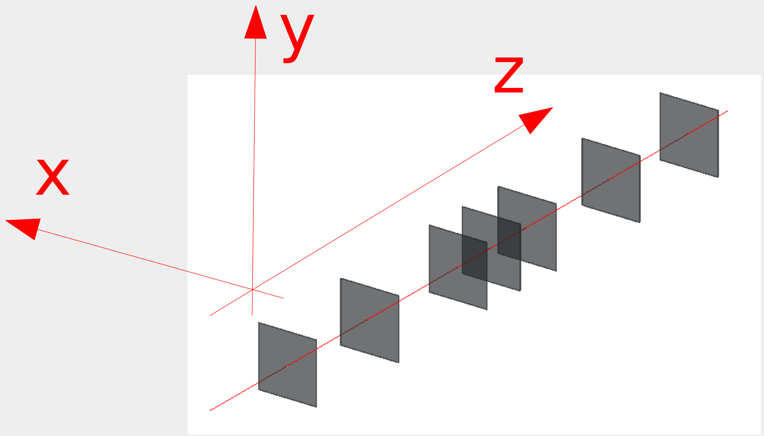
```
(INFO) (Event 1243) [R:DepositionGeant4] Deposited 1319918 charges in sensor of detector detector1
```

- Quite a recently detected issue; we're working on solutions
  - One example is dynamically changing the number of charge carriers propagated together, to reduce the impact of large events

Analysis stage FAQ and FEP

Coordinate system definitions

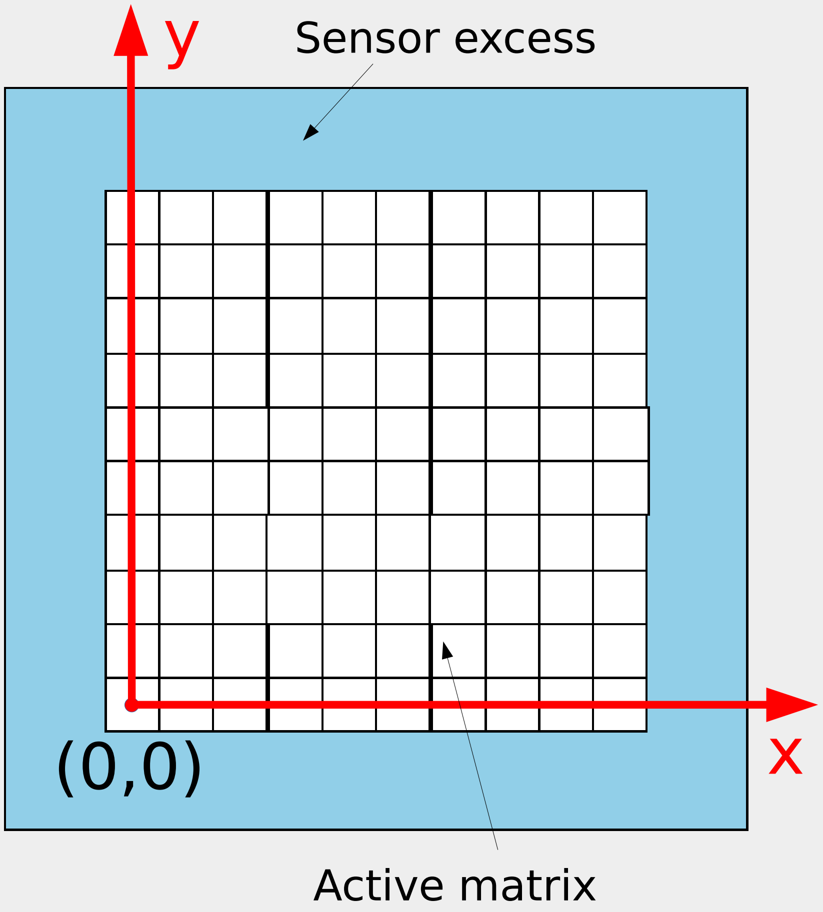
- There is one **global** and one or more **local** coordinate systems
  - The global system holds the **full setup**
  - **Each detector** has a **local** system
- Detectors (and possibly passive material and sources) are **placed in the global system**
  - The system is right-handed and Cartesian
  - Positions relative to the origin chosen by the user
  - With zero rotation of a sensor, the z-axis is orthogonal to the sensor surface (i.e. “along the beam”)
- The local systems are also right-handed and Cartesian
  - More info on next slide



Seven sensors in the global coordinate system

Coordinate system definitions - local

- Origin at **lower left** of pixel grid, in the **centre** of pixel (0,0)
  - The pixel grid lies in the **x-y plane**
  - The centre is defined in **3D**, so z = 0 is located in the **middle of the sensor thickness**
- It is possible to get both local and global coordinates for most objects (e.g. MCParticles)
  - Important to keep in mind which one you are using though!



Frequently encountered problems

- “External analysis script doesn’t work!”
  - Make sure analysis is using the same Allpix^2 version as the simulation run did, and include the correct libraries
- “Output files are too big!”
  - One can select what to include or exclude in [RootObjectWriter]. Including everything uses up a lot of space, and is often not necessary
    - Example: deposited and propagated charges take up a lot of space, but are only useful for post-simulation analysis in special cases
- “Efficiency is strangely low”
  - Check whether **sensor excess** is included. Important in analysis; there can be inactive silicon at the edges
    - Calculating efficiency can go wrong here, with sensor excess hits being included as hits that are expected to generate a signal
    - Hitmaps are useful for checking!
  - Check that the number of pixels assumed in analysis matches the number of pixels in the setup geometry
  - Check that the collection implant in the simulation is in the expected position
    - Output plots from [ElectricFieldReader] are useful here

General hints and tips

Figuring out what's going wrong

- Look for **warnings** and **errors** in the simulation output
    - This will indicate wrongly-configured modules, and misspelled module keywords
- Enable **output plots**; a lot of handy histograms are available to visualise different stages in the simulation flow
- If final results are unexpected: enable **linegraphs** in [GenericPropagation] to see the actual paths of the electrons and holes
    - Note: takes a long time with a fine timestep setting, so only run a few key events
- If a certain event causes issues, **skip_events** under [Allpix] can be used to study a certain event in detail (just make sure that the same random seed is used)

Concluding remarks

- Most problems have solutions!
  - And if not, we'll find a temporary workaround until we have a solution
- Forum is always helpful
- Developers are happy to help, and investigate and sort out any problems

Resources

Website
https://cern.ch/allpix-squared

Repository
https://gitlab.cern.ch/allpix-squared/allpix-squared

Mattermost channel

https://mattermost.web.cern.ch/allpix2

User Forum:

https://cern.ch/allpix-squared-forum/

Mailing Lists:

allpix-squared-users https://e-groups.cern.ch/e-groups/Egroup.do?egroupId=10262858

allpix-squared-developers https://e-groups.cern.ch/e-groups/Egroup.do?egroupId=10273730

User Manual:
https://cern.ch/allpix-squared/usermanual/allpix-manual.pdf

Backup slides

Iron-55 example source

```
# Done as two different sources, of different intensity to match the decay probability
# Let's make them point sources, for now
# Source 1
# 5.9 keV, in 24.4% of decays. This is normalised to intensity 1 by default
/gps/source/intensity 24.4
/gps/particle gamma
/gps/pos/type Point
/gps/pos/centre 0 0 -10.5 mm
/gps/ene/mono 5.9 keV
/gps/ang/type iso
# Theta: 90 to 180 degrees is "forward", such as it is set up now
/gps/ang/mintheta 150 deg
/gps/ang/maxtheta 180 deg
# No need to change phi, keep it isotropic there

# Source 2, in the same position
# 6.5 keV, in 2.85% of decays
/gps/source/add 2.85
/gps/particle gamma
/gps/pos/type Point
/gps/pos/centre 0 0 -10.5 mm
/gps/ene/mono 6.5 keV
/gps/ang/type iso
/gps/ang/mintheta 150 deg
/gps/ang/maxtheta 180 deg
```

```
[DepositionGeant4]
physics_list = FTFP_BERT_LIV
source_type = "macro"
file_name = "./Fe55photons.mac"
max_step_length = 1.0um
number_of_particles = 1
output_plots = true
```