



Allpix² : Un logiciel générique de simulation Monte-Carlo des détecteur pixels semi-conducteurs

Mathieu Benoit

10 Juin 2021



Outline

- Introduction to the Allpix² Framework
- Simulation flow
- Description of modules
 - Energy deposition
 - Charge Transport
 - Transfer
 - Digitization
 - IO and analysis
- Examples
- Ressources



The Allpix² team



Allpix² has been developed and is maintained by

- Koen Wolters, CERN, [@kwolters](#)
- Daniel Hynds, Nikhef, [@dhynds](#)
- Paul Schütze, DESY, [@pschutze](#)
- Simon Spannagel, DESY, [@simonspa](#)

The following authors, in alphabetical order, have contributed to Allpix²:

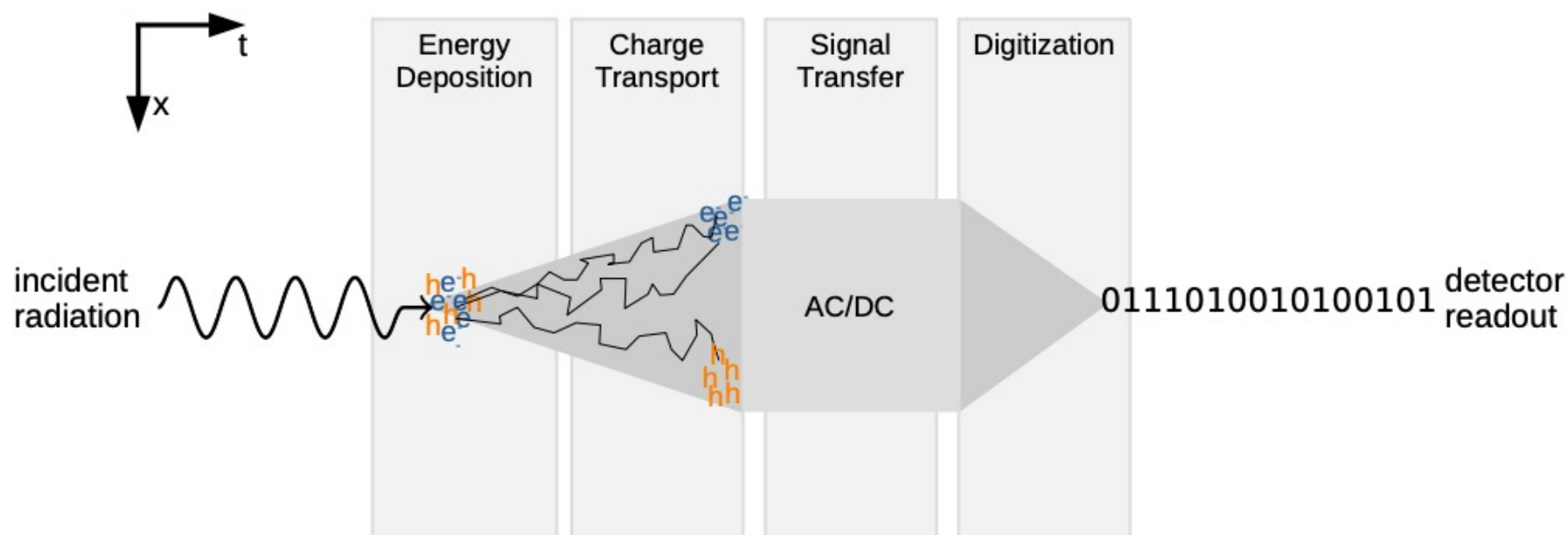
- Mohamed Moanis Ali, Free University of Bozen-Bolzano, [@mmoanis](#)
- Mathieu Benoit, BNL, [@mbenoit](#)
- Thomas Billoud, Université de Montréal, [@tbilloud](#)
- Tobias Bisanz, CERN, [@tbisanz](#)
- Marco Bomben, Université de Paris, [@mbomben](#)
- Koen van den Brandt, Nikhef, [@kvandenb](#)
- Carsten Daniel Burgard, DESY [@cburgard](#)
- Liejian Chen, Institute of High Energy Physics Beijing, [@chenlj](#)
- Katharina Dort, University of Gießen, [@kdort](#)
- Neal Gauvin, Université de Genève, [@ngauvin](#)
- Lennart Huth, DESY, [@lhuth](#)
- Maoqiang Jing, University of South China, Institute of High Energy Physics Beijing, [@mjing](#)
- Moritz Kiehn, Université de Genève, [@msmk](#)
- Stephan Lachnit, Heidelberg University, [@stephanlachnit](#)
- Salman Maqbool, CERN Summer Student, [@smaqbool](#)
- Sebastien Murphy, ETHZ, [@smurphy](#)
- Andreas Matthias Nürnberg, KIT, [@nurnberg](#)
- Sebastian Pape, TU Dortmund University, [@spape](#)
- Marko Petric, CERN, [@mpetric](#)
- Nashad Rahman, The Ohio State University, [@nashadroid](#)
- Edoardo Rossi, DESY, [@edrossi](#)
- Andre Sailer, CERN, [@sailer](#)
- Enrico Jr. Schioppa, Unisalento and INFN Lecce, [@schioppa](#)
- Sanchit Sharma, Kansas State University, [@SanchitKratos](#)
- Xin Shi, Institute of High Energy Physics Beijing, [@xshi](#)
- Ondrej Theiner, Charles University, [@otheiner](#)
- Annika Vauth, University of Hamburg, [@avauth](#)
- Mateus Vicente Barreto Pinto, CERN, [@mvicente](#)
- Andy Wharton, Lancaster University, [@awharton](#)
- Morag Williams, University of Glasgow, [@williamm](#)

The Allpix² Framework

A Modular, Generic Simulation Framework for pixel Detectors

- The framework aims at facilitating the different steps of the simulation of semiconductor detectors
 - **Energy deposition in the detector material** (GEANT4 etc.)
 - **Charge Transport in the semiconductor**
 - **Transfer, Digitization and Analysis**
- The developers aim at implementing the best practices in semiconductor detector simulation in a generic way to provide to the community verified and standardized methods and a development environment for further improvement to simulation methods
 - Framework infrastructure
 - Documentation, examples and code demonstration

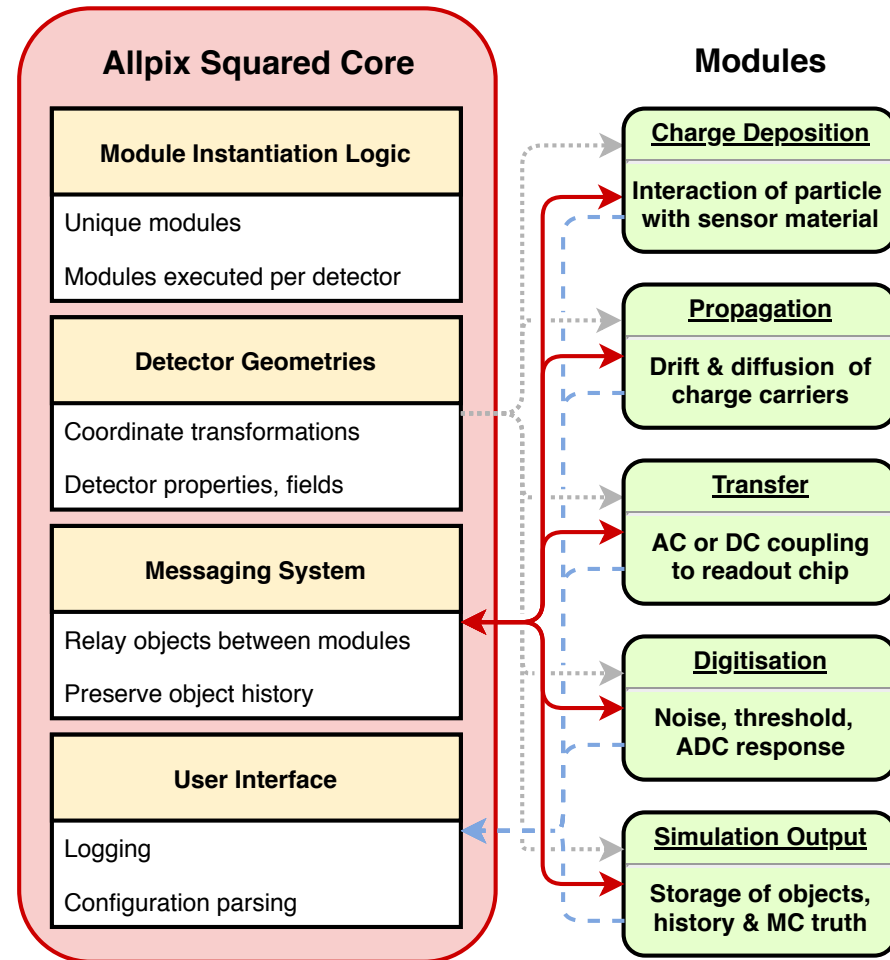
The simulation flow



The Allpix² Framework



The allpix Framework offer a modular way to build a simulation flow and provide the infrastructure (core) to easily transfer information between modules, configure them and log their actions



Framework Configuration

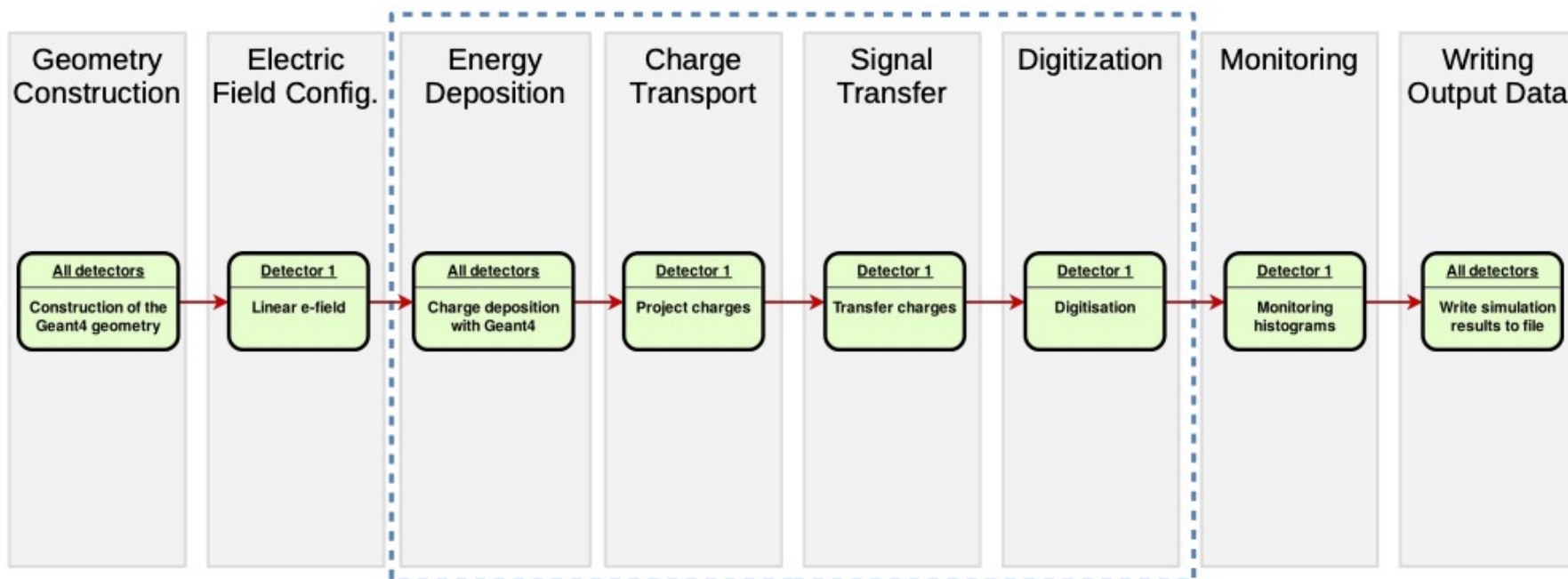
- Framework configured by one file
 - All desired modules listed in order of execution
 - Key-value pairs in TOML-style (extended)
 - Human readable
 - Little overhead (e.g. compared to XML)
- Support for physical units
 - Never ask what units are used – type them out!
 - Automatic conversion to internal units
 - No need for manual conversions in C++

```

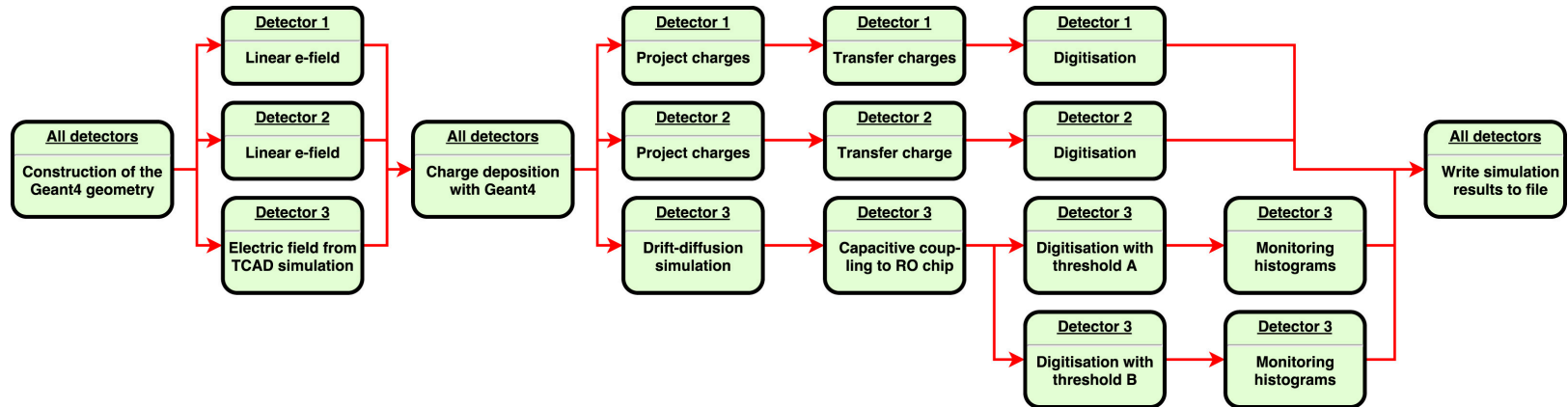
1  [AllPix]
2  log_level = "INFO"
3  number_of_events = 500000
4  detectors_file = "telescope.conf"
5
6  [GeometryBuilderGeant4]
7  world_material = "air"
8
9  [DepositionGeant4]
10 physics_list = FTFP_BERT_LIV
11 particle_type = "Pi+"
12 number_of_particles = 1
13 beam_energy = 120GeV
14 # ...
15
16 [ElectricFieldReader]
17 model="linear"
18 bias_voltage=150V
19 depletion_voltage=50V
20
21 [GenericPropagation]
22 temperature = 293K
23 charge_per_step = 10
24 spatial_precision = 0.0025um
25 timestep_max = 0.5ns
26
27 [SimpleTransfer]

```

Simulation Flow in Allpix² (simple)

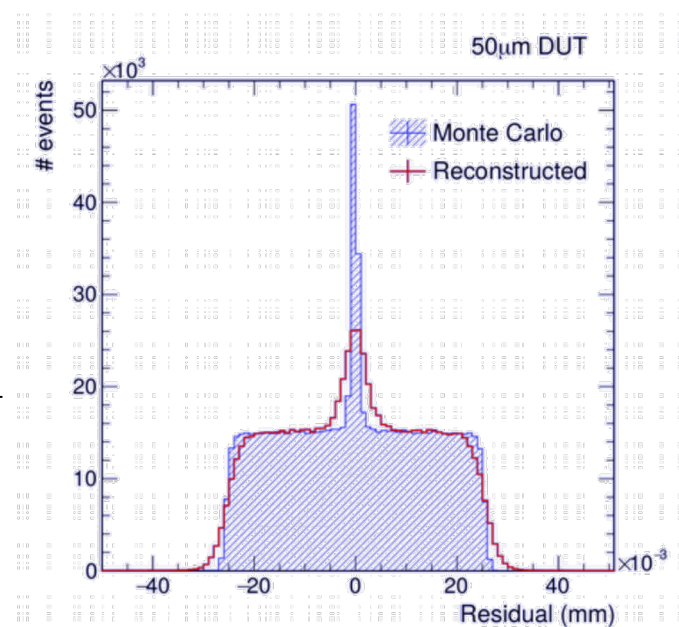


Simulation Flow in Allpix² (Complex)



Messages & Object History

- Data exchanged between modules by means of messages
 - Message holds specific data type and has an origin (detector name, stream name...)
 - Modules bind to specific message (or message type) via central messenger
- Object history
 - For each object the full provenance is recorded using TRef objects
 - PixelHit ← PixelCharge ← PropagatedCharge(s) ← DepositedCharge(s) ← MCParticle(s) ← MCTrack
 - Allows direct relation of each pixel hit from front-end to initial particle(s)
 - Relation between MCParticles enables sorting between primaries (entered sensor from outside) and secondaries (produced within sensor)
 - Answers questions like “where in the detector did the charge carriers of this hit originate from?”



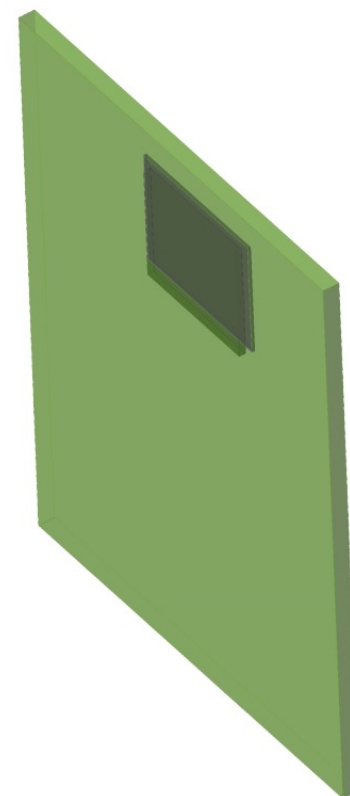
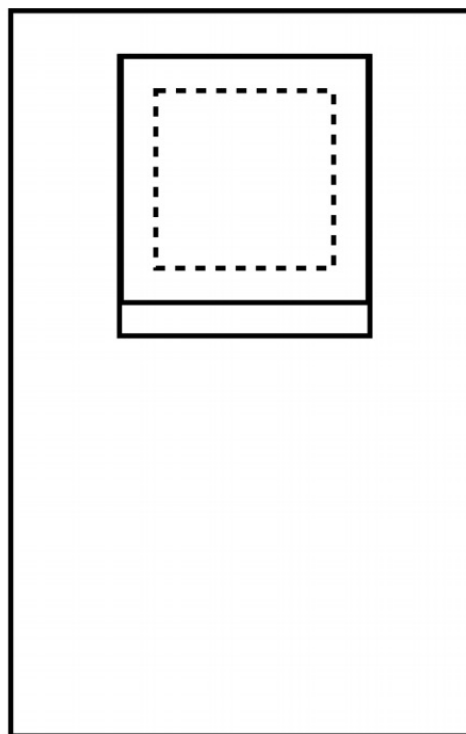
Defining the geometry

The Model: timepix.conf

```

1 type = "hybrid"
2
3 number_of_pixels = 256 256
4 pixel_size = 55um 55um
5
6 sensor_thickness = 300um
7 sensor_excess = 1mm
8
9 bump_sphere_radius = 9.0um
10 bump_cylinder_radius = 7.0um
11 bump_height = 20.0um
12
13 chip_thickness = 700um
14 chip_excess_left = 15um
15 chip_excess_right = 15um
16 chip_excess_bottom = 2040um
17
18 [support]
19 thickness = 1.76mm
20 size = 47mm 79mm
21 offset = 0 -22.25mm

```



Defining the geometry

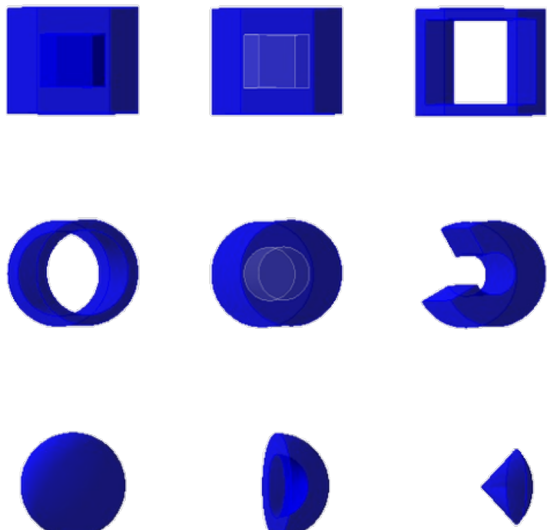


Passive Materials: Things in the Beamline

Paul Schütze, DESY

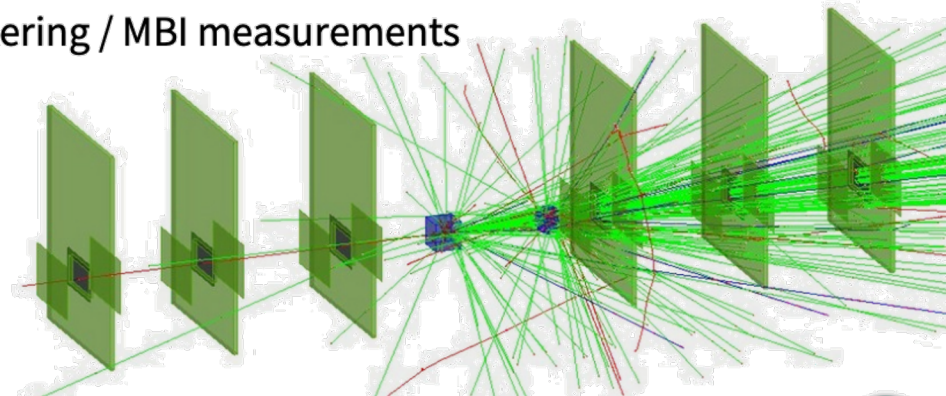
Koen van den Brandt, Nikhef

- Added possibility to define passive material in the geometry
- Different shapes, automatic merging of multiple shapes / hierarchy resolution
- Completely transparent to core framework through new parameter “*role*”



Usage examples:

- Realistic test beam setup (cooling box)
- Calorimeter simulation
- Multiple Scattering / MBI measurements



Defining particle sources and energy deposition model

DepositionGeant4

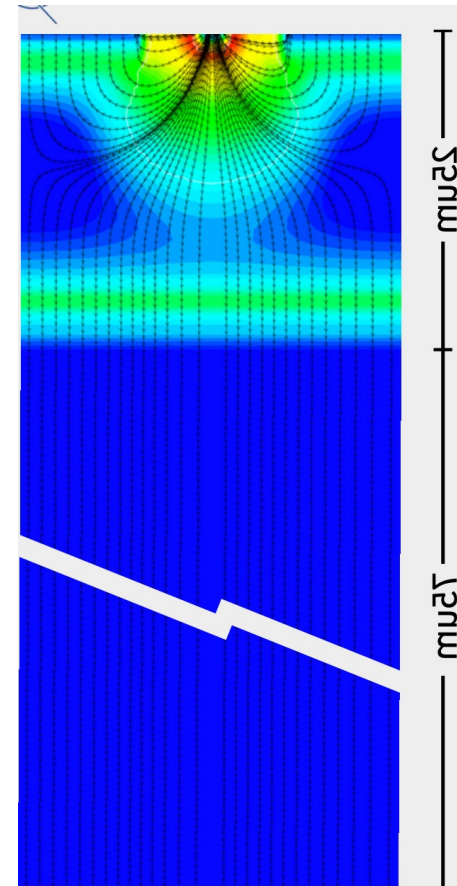
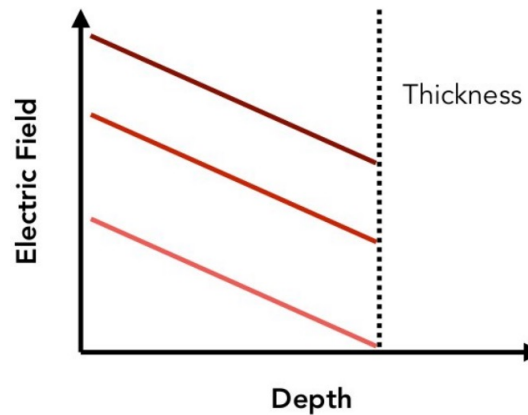
- Interface to Geant4
- Definition of particles and tracking through the setup
- Energy deposition in sensitive material
- Pick ...
 - the type of particles (or radioactive sources)
 - the particle energy
 - the origin and direction of the beam
 - the shape and size of the beam
 - a suitable physics list

```
[DepositionGeant4]
particle_type = "e-"
source_energy = 5GeV
source_type = "beam"
beam_size = 3mm
source_position = 0 0 -200mm
beam_direction = 0 0 1
physics_list = FTFP_BERT_EMZ
```

Defining the electric field

- Modules are provided to model detector with common analytical models for simple simulation
- Tools for converting more complex electric field distribution from TCAD (Sentaurus) are provided to import more complex electric field distribution in the Framework

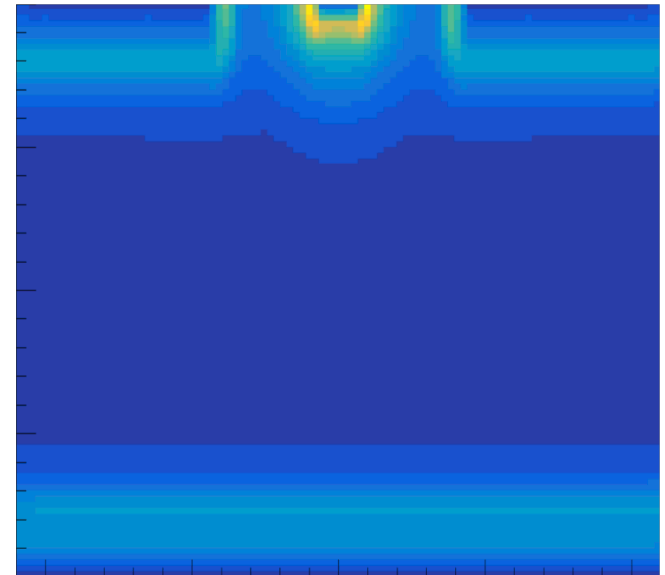
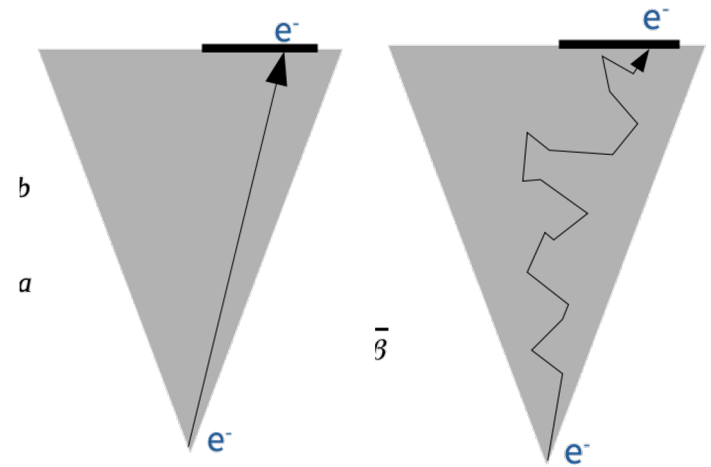
```
[ElectricFieldReader]  
model = "linear"  
bias_voltage = -50V  
depletion_voltage = -30V
```



Propagating deposited charge

The framework provide multiple methods to transport the charge in the defined electric field, from simple to complex

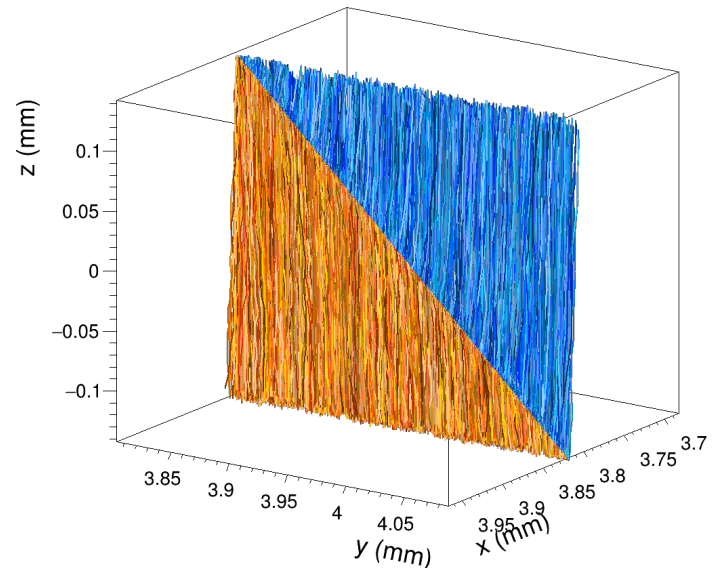
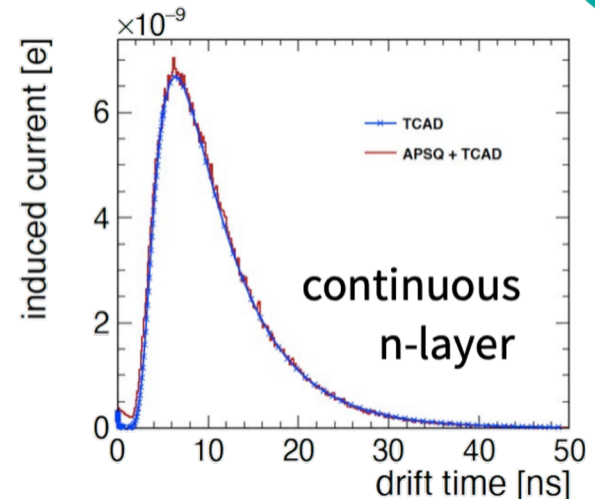
- **Projection Propagation**
 - Simple approximation of transport, good for linear electric field and thick sensors, transport to surface then smear charge to account for diffusion
- **Integration of charge motion**
 - Full numerical integration of charge or fragments trajectory via a RKF45 method, in the electric field, taking into account :
 - Mobility
 - Magnetic and electric field



Transfer charge to signal

The framework provide multiple transfer method from simple to complex to transform the information from charge trajectory to registered signal in the electrode

- **Simple Transfer**
 - Assign charge to closest electrode at the end of trajectory
- **Simple Ramo**
 - Assign charge to each electrode following start and end point and Ramo Potential model or Ramo potential provided via TCAD
- **Full Ramo**
 - Calculate induced pulse from charge trajectory for each electrode using provided Ramo potential



Digitization

Digitizing Pulses: CSADigitizer

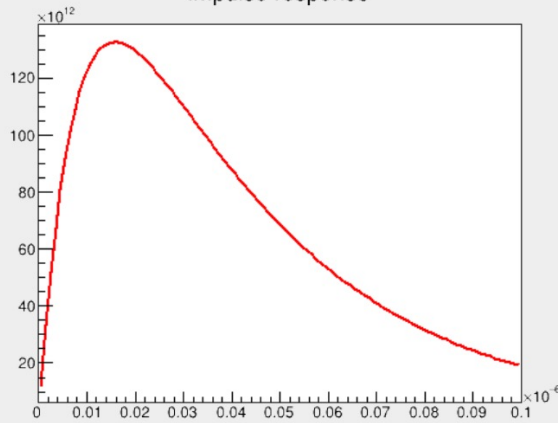


Annika Vauth, UniHH/DESY

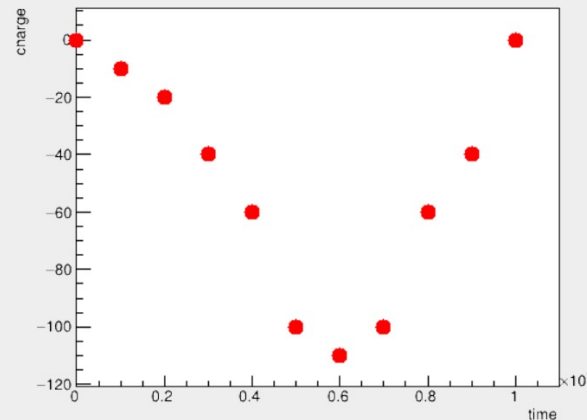
- Implementation of charge-sensitive amplifier with Krummenacher feedback, configuration via:
 - Rise time, feedback time & capacitance – “simple”
 - Detector cap., Krum. current, transconductance – “csa”
- Integrated ToT / ToA sampling on different clocks

```
[CSADigitizer]
model = "simple"
feedback_capacitance = 5e-15C/V
rise_time_constant = 1e-9s
feedback_time_constant = 10e-9 s
integration_time = 0.5e-6s
threshold = 10e-3V
clock_bin_toa = 1.5625ns
clock_bin_tot = 25.0ns
```

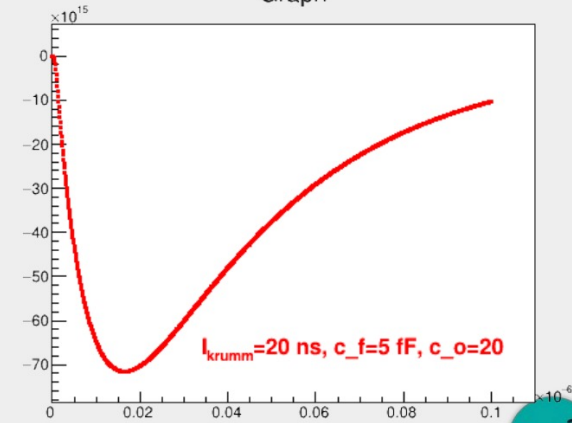
impulse response



pulse



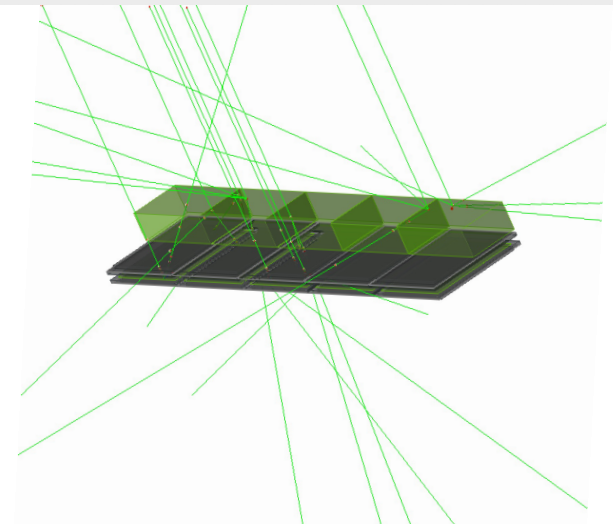
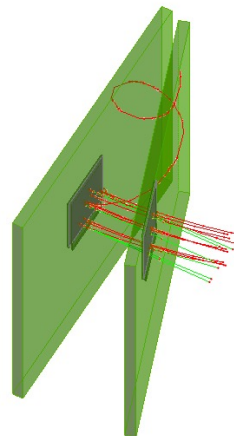
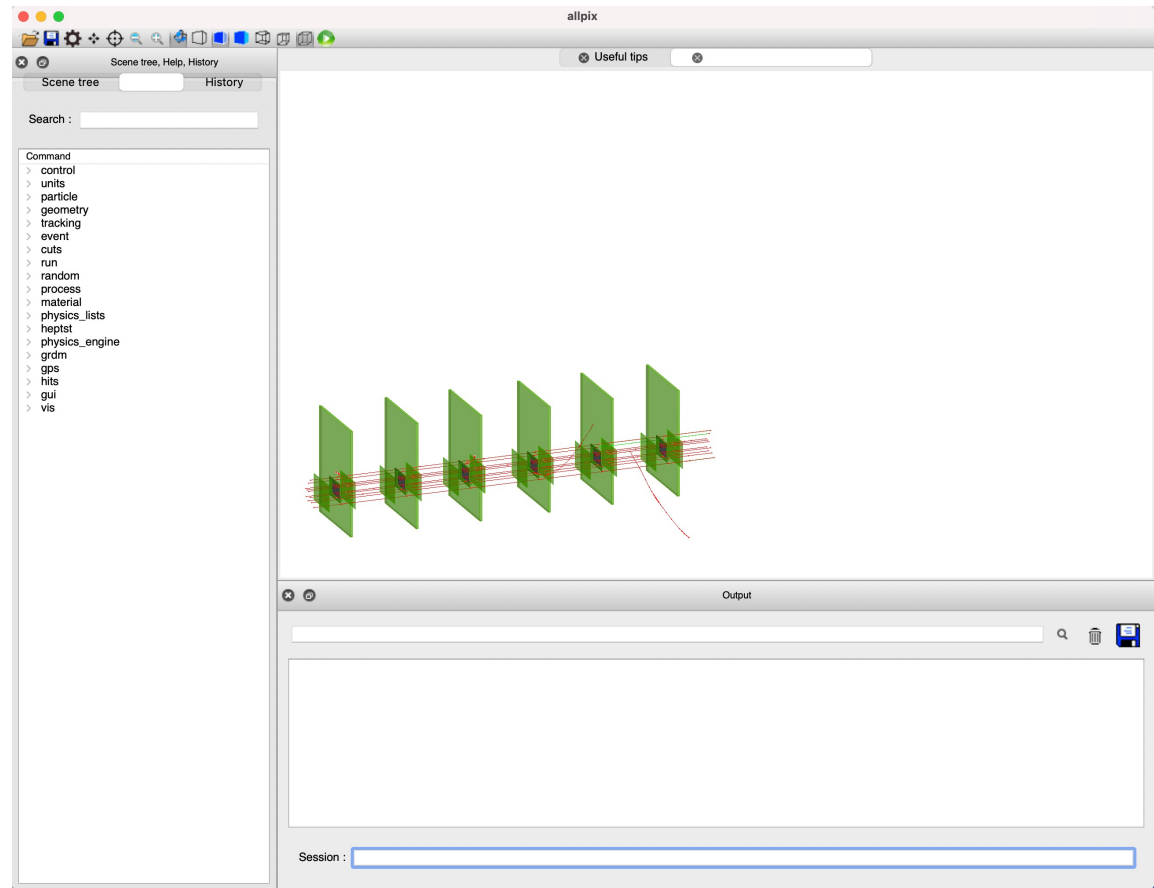
Graph



Visualization

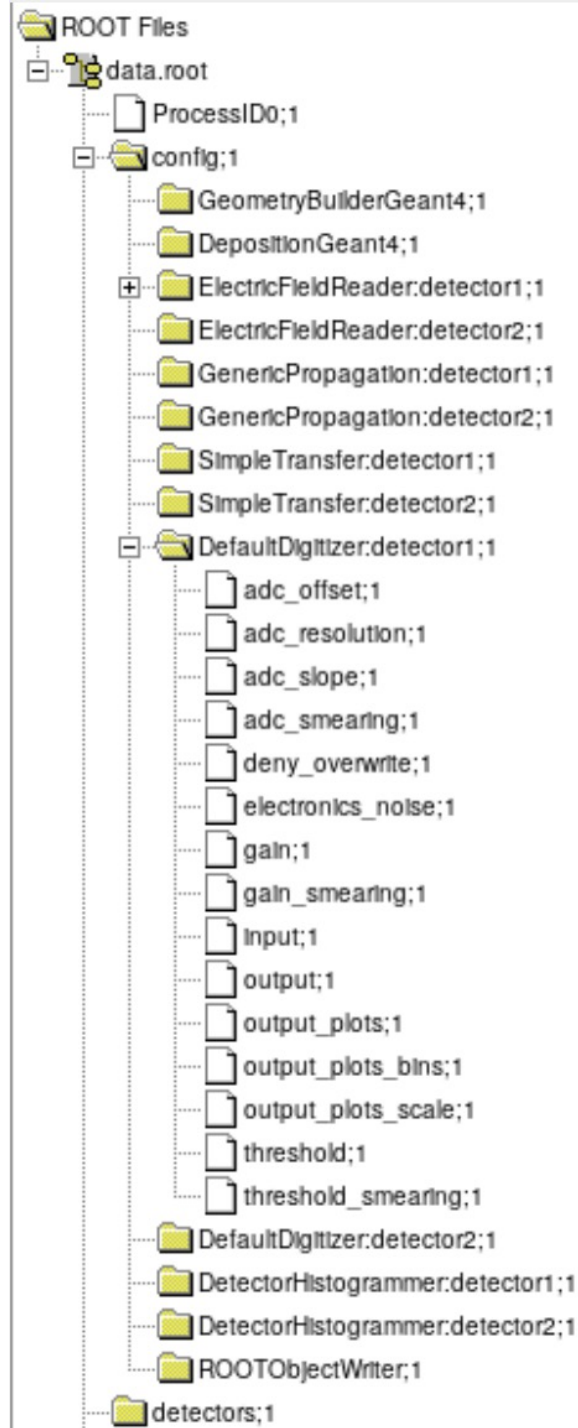
The framework comes with an easy to use visualization Module using the GEANT4 capabilities to illustrate the simulation

- Particle trajectories, charge, secondaries production
- Volumes, passive and active
- Hit pixels, energy deposition



Output of results

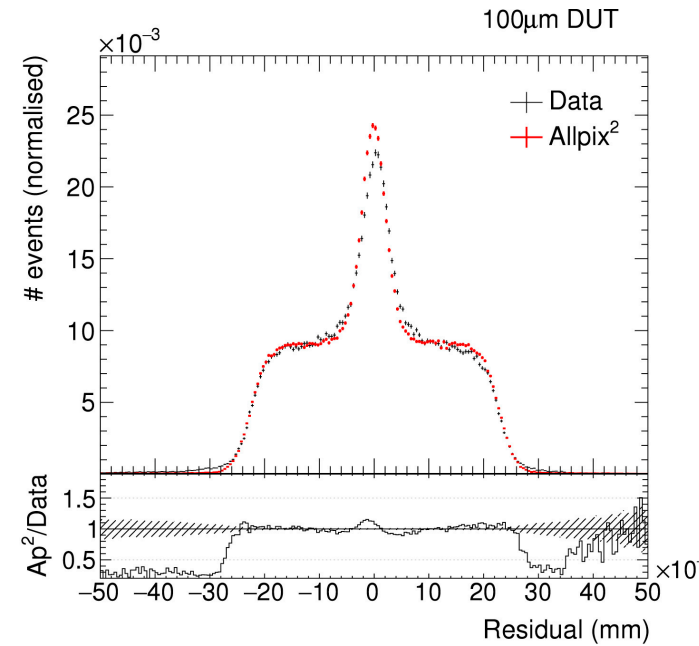
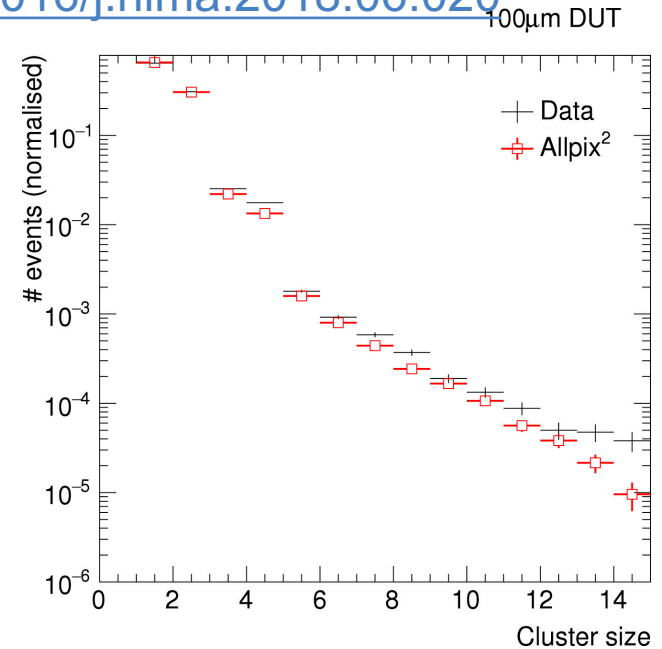
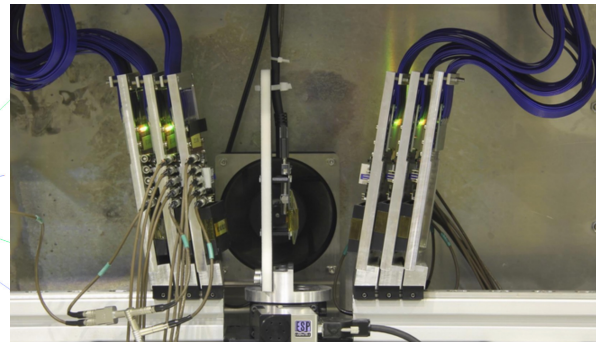
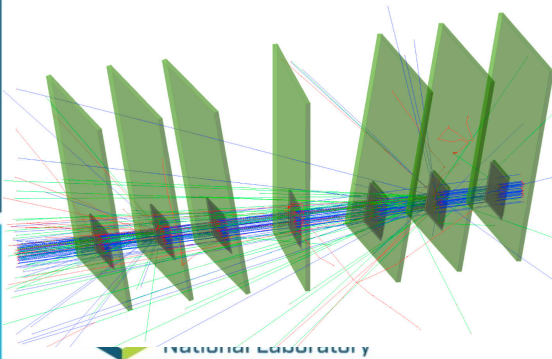
- Different output writers available
 - Eutelescope, RCE, Corryvekran and more
 - Can be use for reconstruction with Telescope software
- Native format: ROOT files with all objects
 - Also contains detectors & sim. parameters
 - Full framework configuration can be reconstructed from single data file
- ROOTObject reader replays data from file
 - Simulate deposition & propagation once
 - Read data from file and quickly repeat final digitization step with different parameters
- Each module reports a set of plots and histogram for rapid analysis of results



Example Application

[NIMA 901 \(2018\) 164 – 172](#)
[doi:10.1016/j.nima.2018.06.020](#)

- Simulation of a beam telescope setup:
CLICdp Timepix3 telescope @ SPS H6
 - Telescope: 6x Timepix3 w/ 300 μm sensors
 - DUT: 1x Timepix3 w/ 50 μm sensor
- Validation of reconstruction
- Different algorithms used:
 - Telescope: projection
 - DUT: successive integration
- Linear electric field approximation
- Very good agreement between data and simulation observed (total charge: Geant4; cluster size: both; residual shape: Allpix²)



Documentation, coding guidelines, peer review

- Focus from the very beginning on well-documented framework
 - Source code documentation for every class, method
 - Doxygen markup for code reference
 - Deployed to the website for tags
 - Extensive User Manual in LaTeX
 - Automatically compiled by CI
 - Module documentation as Markdown
 - Document module parameters algorithms
 - Included in manual via Pandoc
 - Peer review of code for inclusion help maintains code readability and preserve the usability

GenericPropagation

Maintainer: Anika Vuurh (anika.vuurh@dey.de), Simon Spangberg (simon.spangberg@dey.de)

Status: Functional
Input: DepositedCharge
Output: PropagatedCharge

Description

Simulates the propagation of electrons and/or holes through the sensitive sensor volume of the detector. It allows to propagate sets of charge carriers together in order to speed up the simulation while maintaining the required accuracy. The propagation processes for these sets are fully independent and therefore parallelized. The maximum size of the set of propagated charges and thus the accuracy of the propagation carrier control.

The propagation consists of a combination of drift and diffusion simulation. The drift is calculated using the charge carrier velocity derived from the charge carrier mobility parameterization by C. Accornero et al. [1]. The carrier mobility for other electrons is taken automatically chosen based on the type of the charge carrier under consideration. Thus, acceptor with both electrons and holes is treated equally.

The parameters `propagate_electrons` and `propagate_holes` allow to control which type of charge carrier is propagated for their respective electrodes. Either one of the carrier types can be selected, or both can be propagated. It should be noted that this will slow down the simulation considerably, since there are many carriers have to be handled with should only be used where sensible. The direction of the propagation depends on the electric field configuration, which should be assumed that the carrier types selected are actually transported to the implant side. For these electric fields, a warning is issued if possible (see configuration document).

A fourth order Runge-Kutta is falling performed with 50th order series expansion in order to integrate the electric field. After every Runge-Kutta step, the diffusion is accounted for by applying an offset drawn from a Gaussian distribution calculated from the Einstein relation

$$D = \frac{k_B T}{q} \mu$$

using the carrier mobility μ , the temperature T and the time step t . The propagation steps evaluate the set of charges reaches any surface of the sensor.

The propagation module also produces a variety of output plots. These include a 3D line plot of the path of all separately propagated charge carrier sets from their point of deposition to the end of their drift, with nearby paths having different colors. In the coloring scheme, electrons are marked in blue colors, while holes are presented in red. A 2D GIF animation for the drift of individual sets of charges with the use of the path plots in the set can be produced. Finally, the module produces 2D contour animations in all the showing the concentration flow in the sensor. It should be noted that generating the animations should be combined with some other visualization tool.

Dependencies

This module requires an installation of Eigen3.

Parameters

- `temperature`: Temperature of the sensitive device, used to estimate the diffusion coefficient of the diffusion. Defaults to room temperature (293.15K).
- `charge_per_step`: Maximum number of charge carriers to propagate together. Small charge carriers at a specific point in time of the number of charge carriers can not be a value of 10 charges per step is usually default. If this value is not specified.
- `spatial_precision`: Spatial precision to use for the truncation of the Runge-Kutta algorithm. Spatial precision after calculating the uncertainty from the fifth order series method. Default is 10nm. To increase the resolution of the Runge-Kutta integration with respect to the time to increase the truncation of the spatial precision parameter. Default is 10nm.
- `simulation_max`: Maximum steps in time to use for the Runge-Kutta integration regards default to 10 ns.
- `simulation_time`: Time within which charge carriers are propagated. After success is performed for the respective carrier. Defaults to 100 ns. Each crossing time of 20.
- `propagate_electrons`: Select whether electron-type charge carriers should propagate to an anode.
- `propagate_holes`: Select whether hole-type charge carriers should be propagated to a cathode.
- `convergence_epsilon`: Convergence of output plots should be given for every step. The simulation, it is not recommended to enable this option for cases with more than a few.
- `output_plots_step`: Time step to use between two points plotted. Indirectly determines the amount of points plotted. Defaults to increasing phase if not explicitly specified.
- `output_plots_theta`: Viewport angle of the 3D animation and the 2D line graph around the world Z-axis. Defaults to zero.
- `output_plots_phi`: Viewport angle of the 3D animation and the 2D line graph around the world X-axis. Defaults to zero.
- `output_plots_use_point_labels`: Determine if the plots should use points as unit instead of metric length in meters. Defaults to false (thus using the metric system).
- `output_plots_use_equal_scaling`: Determine if the plots should be produced with equal distance scales on every axis (also this implies that some points will fall out of the graph). Defaults to true.
- `output_plots_step_plots`: Determine if output should be plotted as step plots. Defaults to false if enabled the start and the end of the axis will be at the left point between plots.
- `output_animation`: In addition to the other output plots, also write a GIF animation of the charges drifting towards the electrodes. This is very slow and writing the animation takes a considerable amount of time, therefore defaults to false. This option requires `output_plots` to be enabled.
- `output_animation_time_scaling`: Scaling for the animation used to convert the actual simulation time to the time step in the animation. Defaults to 1.0ns, meaning that every nanosecond of the simulation is equal to an animation step of a single second.
- `output_animation_marker_size`: Scaling for the markers on the animation, defaults to one. The markers are already internally scaled to the charge of the step, normalized to the maximum charge.
- `output_animation_color_max_scaling`: Scaling to use for the contour color axis from the theoretical maximum charge at every single plot step. Default is 10, meaning that the maximum of the color scale axis is equal to the total amount of charges (should be less values above this, are displayed in the same maximum color). Parameter can be used to improve the color scale of the contour plots.
- `output_animation_color_markers`: Determine if colors should be for the markers in the animations. Defaults to false.

Usage

An example of generic propagation for all sensors of type "Detector" at room temperature using packets of 20 charges in the following:

```
[GenericPropagation]
type = "Detector"
temperature = 293
charge_per_step = 20
```

ap²

Allpix² User Manual

7 Modules

This section describes all currently available Allpix² modules in detail. This includes a description of the physics implemented as well as possible configuration parameters along with their defaults. For inquiries about certain modules or in documentation, the respective maintainers should be contacted directly. The modules are listed in alphabetical order.

7.1 CSADigitizer

Maintainer: Anika Vuurh (anika.vuurh@dey.de), Simon Spangberg (simon.spangberg@dey.de)

Status: Functional
Input: PixelCharge
Output: PixelHit

Description

Digitization module which translates the collected charges into a digitized signal, emulating a charge sensitive amplifier with Krummenacher feedback. For this purpose, a transfer function for a CSA with Krummenacher feedback is taken from [2]: $W(x) = \frac{x}{(1 + \exp(-x))}$ with $x = R_f C_f \cdot$ rise time constant $\tau_r = \frac{C_{in} C_f}{C_{in} + C_f}$.

The impulse response function of this transfer function is convoluted with the charge pulse. This module can be steered by either providing all contributions to the transfer function as parameters within the cs module, or using a simplified parameterization providing rise time and feedback time. In the latter

- 7.1 CSADigitizer
- 7.2 CapacitiveTransfer
- 7.3 CopyVecWriter
- 7.4 DatabaseWriter
- 7.5 DefaultDigitizer
- 7.6 DepositionGeom4
- 7.7 DepositionPointCharge
- 7.8 DepositionReader
- 7.9 DetectorHistogrammer
- 7.10 ElectricFieldReader
- 7.11 GDMLOutputWriter
- 7.12 GenericPropagation
- 7.13 GeometryBuilderGeom4
- 7.14 InducedTransfer
- 7.15 LCHOWriter
- 7.16 MagneticFieldReader
- 7.17 ProjectionPropagation
- 7.18 PulseTransfer
- 7.19 RCHWriter

Conclusion

The Allpix² Framework, a generic tool for simulation of semiconductor detectors combine the power of GEANT4 with state of the art algorithms for the simulation of transport, transfer and digitization

- Easy to use , no coding required in most case
- Validated widely by the community
- In constant evolution, each version gets better !
- Well documented, including examples

We hope to have convinced you to have a look at the framework !

All the good links !!!

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

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

Docker https://gitlab.cern.ch/allpix-squared/allpixsquared/container_registry 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>

Tutorials

For multiple hands-on tutorials and in-depth presentation on allpix

<https://project-allpix-squared.web.cern.ch/project-allpix-squared/page/publications/>

Users who published so far :

<https://www.scopus.com/results/citedbyresults.uri?sort=plf-f&cite=2-s2.0-85048755761&src=s&imp=t&sid=e2843363bb8a407e98425a734fed7a9e&sot=cite&sdt=a&sl=0&origin=inward&editSaveSearch=&txGid=7bb166821b2dc64286a13edfe8973e2b>