ap²

allpix squared

cern.ch/allpix-squared

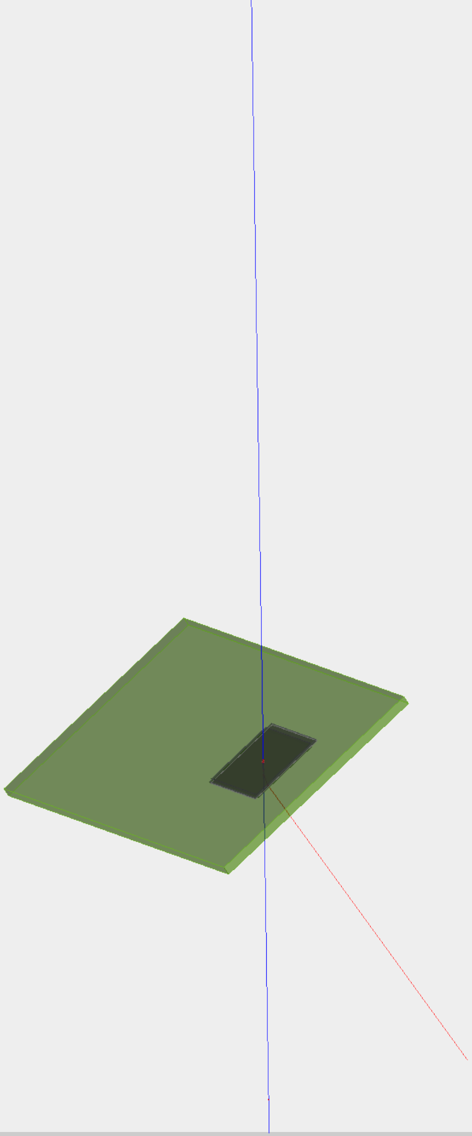
Semiconductor Detector End-to-end Simulations with Allpix Squared: Latest Features, Ongoing Developments, and Application Examples

Paul Schütze
Simon Spannagel
Håkan Wennlöf

04/12 -24

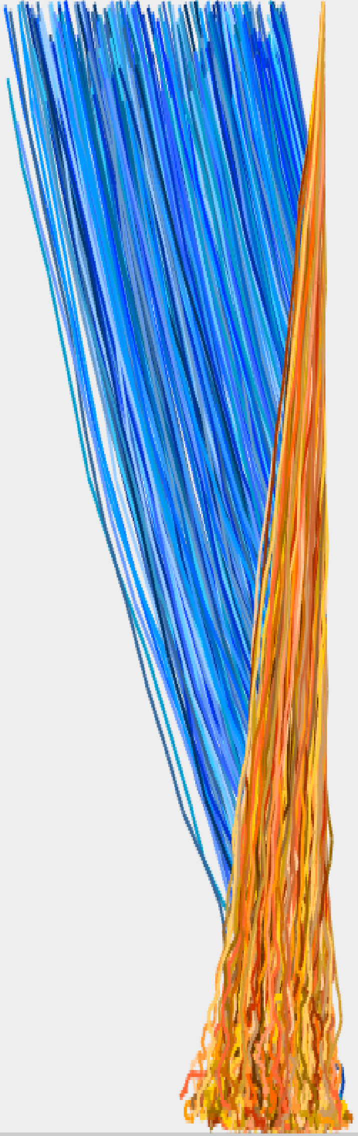
Outline

- Introduction
- Recent addition highlights
  - Briefly
- Recent usage example highlights
  - Focused on work since the last DRD3 week
- Ongoing and planned developments
  - Focused on work since the last DRD3 week
- Summary and outlook

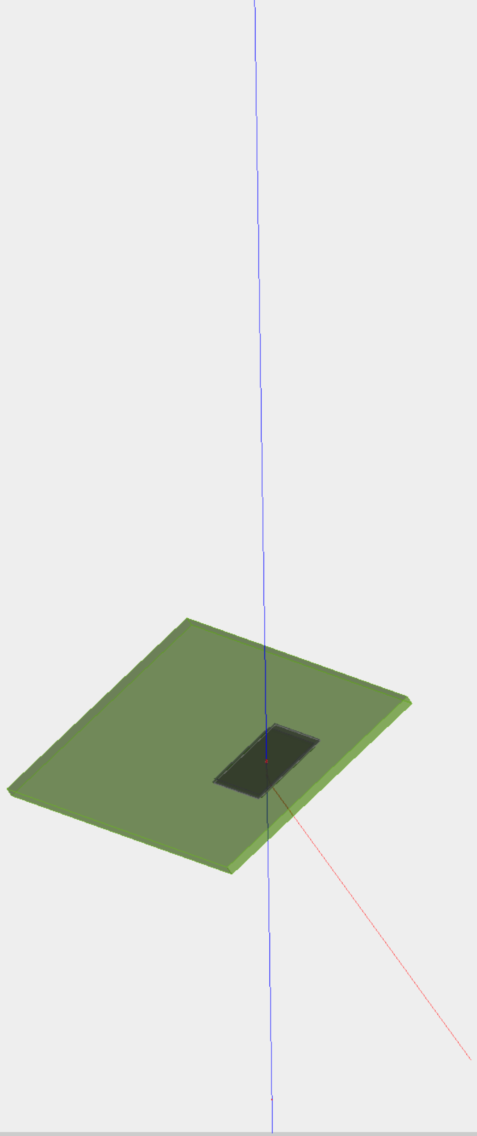


Allpix Squared

- Framework presented in more detail at the last DRD3 week
- **Open-source** simulation framework, based on a **modular** design, written in **modern C++**
- **Complementary** to detailed device simulation; fast, allows for high-statistics samples accounting for stochastic effects in the physics
- Can interface to **Geant4** for energy deposition simulation, **TCAD** for electric fields, **ROOT** for I/O, event generators via **HepMC3**
- **Low entry barrier:** extensive documentation, public forum, human-readable configuration files (no coding or code-reading required for use)
- Allows for **Monte Carlo simulations** of pixellated detectors
  - Can simulate the **full detector hit chain;** energy deposition, charge carrier propagation and transfer, and digitisation
- Multiple-detector setups can be simulated, giving realistic simulations of e.g. test beam applications



Recent addition highlights



Recent additions

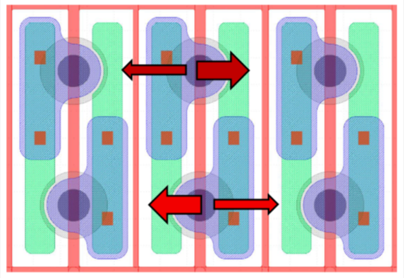
- Plenty of bugfixes
  - Fixing sneaky errors turning up in corner cases
- Capacitative transfer updates
  - Now handles implants like the other transfer modules
  - Allows mirroring of coupling matrix for odd/even row and column asymmetry (useful for e.g. “bitten design”)
- Allow configuration of field units when importing electric fields
- Store more information in objects
  - Geant4 particle interaction process subtype
  - Total deposited energy of an MCParticle

Allpix Squared / Merge requests / !1127
Capacitative transfer bugfix and update

Allpix Squared / Merge requests / !1129
Improved handling of getting ints with units (safeguarding against future mistakes)

Allpix Squared / Merge requests / !1145
Changed mobility value from integer 1417 to 1417.0

Allpix Squared / Merge requests / !1125
Updated the Mandic Trapping Model used in Allpix2



Allpix Squared / Merge requests / !1148
Option to mirror coupling matrix every second row (or column)

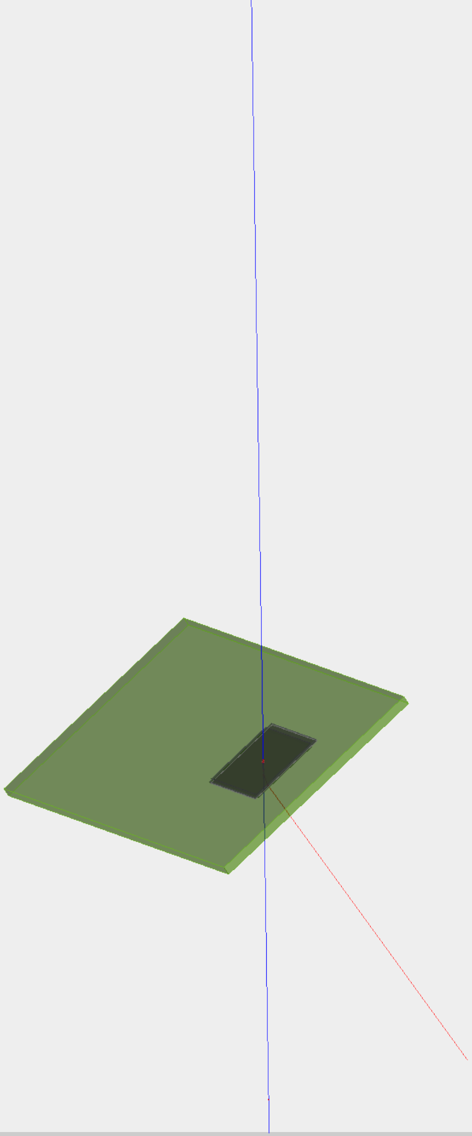
Allpix Squared / Merge requests / !1146
ElectricFieldReader: allow configuration of field units

Allpix Squared / Merge requests / !1128
DepositionGeant4: store G4Process->GetProcessSubType

Allpix Squared / Merge requests / !1144
Add total deposited energy to MCParticle

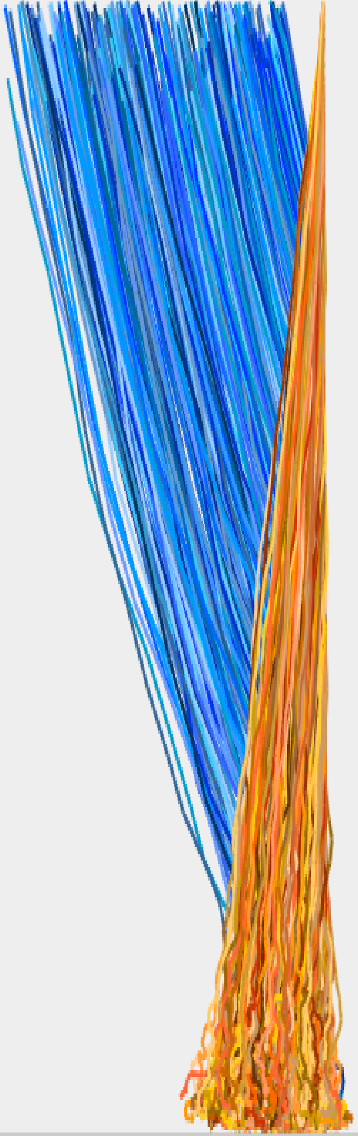
Black boxes are hyperlinks to the merge requests

Usage example highlights



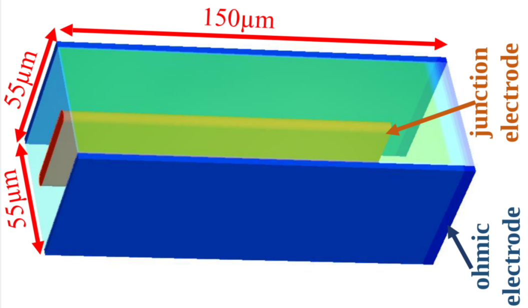
Usage examples

- The framework is **actively used** by the DRD3 community and beyond
- Examples:
  - Simulation of CMOS strip sensors
    - See talk tomorrow (I. Zatocilova, N. Davis et al.)
  - TCT studies of diamond detectors (F. Ishaqzai)
  - Impact ionisation modelling in LGADs (A. Visibile, P. Skomina)
  - Digitisation modelling for radiation damage levels (e.g. this earlier talk by J. Dandoy)
  - Extensive simulations of MAPS in 180 nm and 65 nm CIS
- Use across working groups in DRD3, and also in **DRD7**



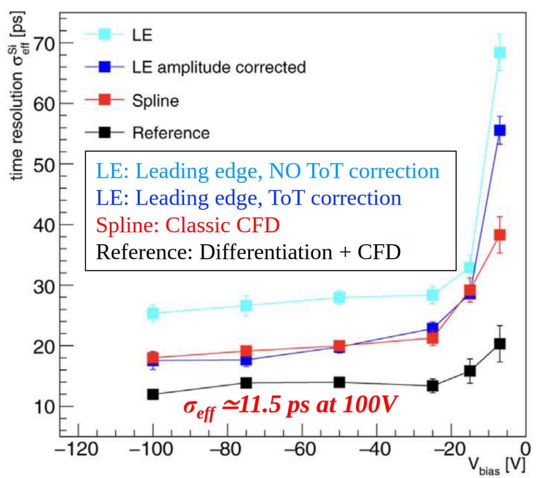
3D-trench electrode sensors for timing

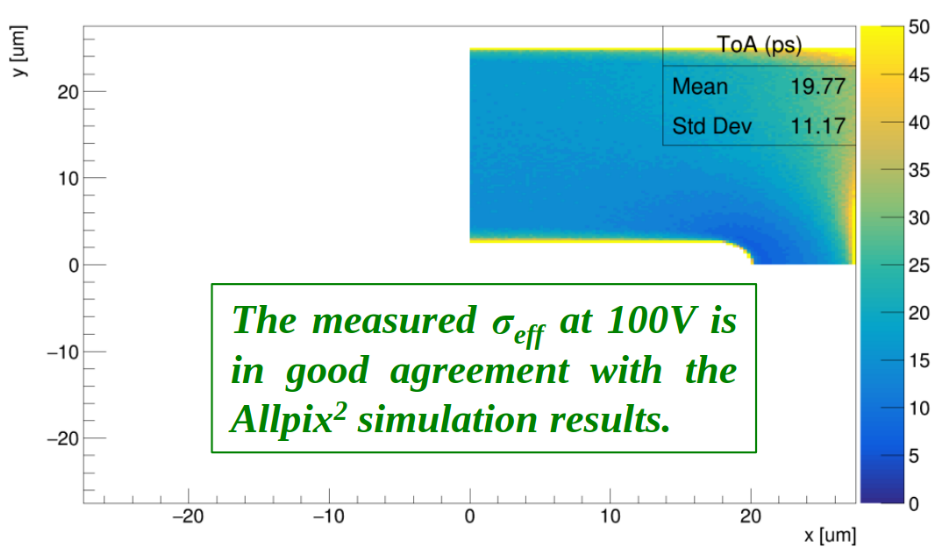
- Investigating **new designs of 3D sensors** (J. Ye, PIXEL 2024)
  - 3D-trenched sensors for tracking + timing
  - Fast, and radiation hard
  - Part of the TimeSPOT project
- Electric fields imported from TCAD into Allpix^2
- Fine **scans of charge deposition** performed, and **time-of-arrival information extracted**
  - Charges deposited at points in x-y at a given depths, with step length of 0.25 µm
  - **Matches experimental data** well



Work by Jixing Ye, University of Trento

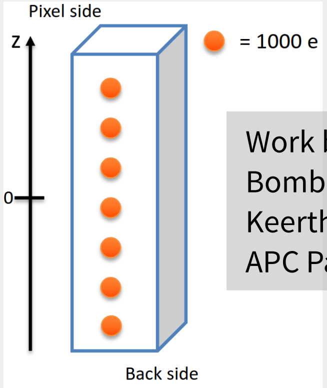
https://indico.in2p3.fr/event/32425/contributions/142767/





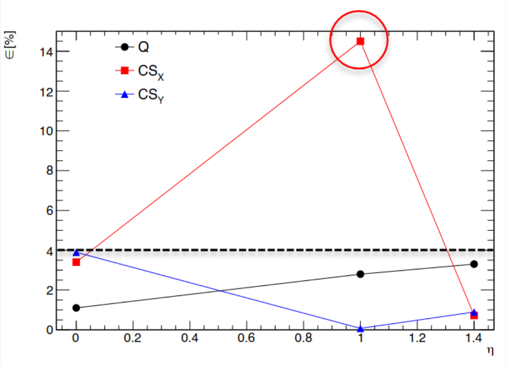
https://indico.in2p3.fr/event/32425/contributions/142767/

A lightweight algorithm to model radiation damage effects in MC events for HL-LHC experiments



Work by Marco Bomben and Keerthi Nakkalil, APC Paris

- HL-LHC experiments require **fast and precise MC production**, taking radiation damage effects into account
- Full digitisation simulation takes a long time; faster way needs to be found
- Simulating point depositions across a sensor to **generate look-up tables** for where a deposited charge ends up, and how much of the signal is lost , for different irradiation levels (M. Bomben, K. Nakkalil)
- Agreement with full (and much slower) MC simulations within a few percent for most of the investigated eta range
  - **Promising method** for future simulations!



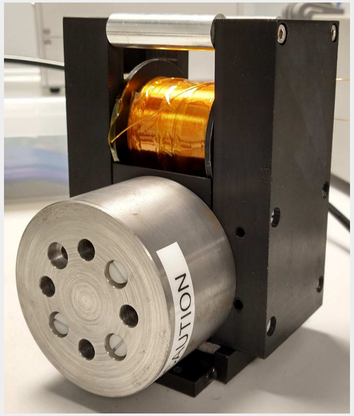
Agreement at few % level but for transverse cluster size at η = 1

https://indico.in2p3.fr/event/32425/contributions/142752/

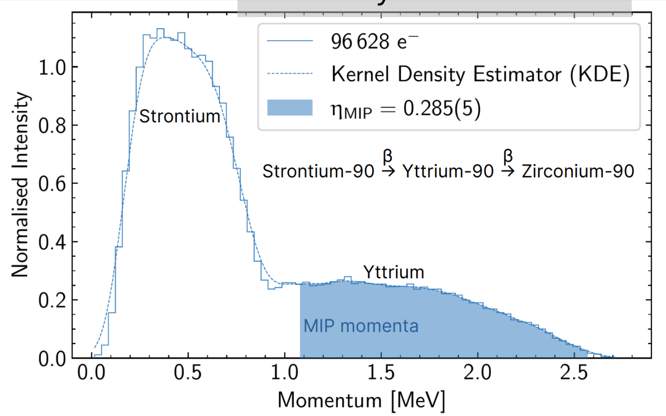
Optimisation of a minimum ionising particle generator

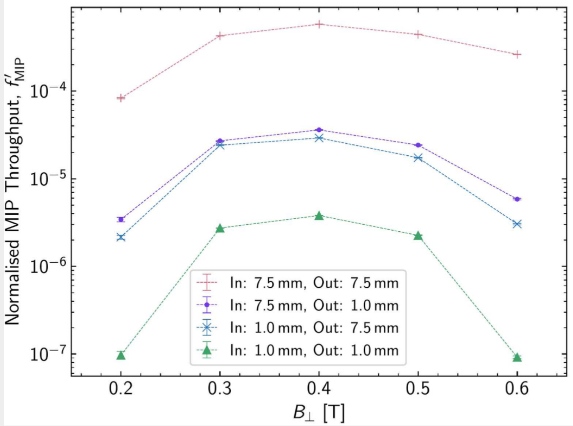
Work by Samuel Wood, University of Oxford

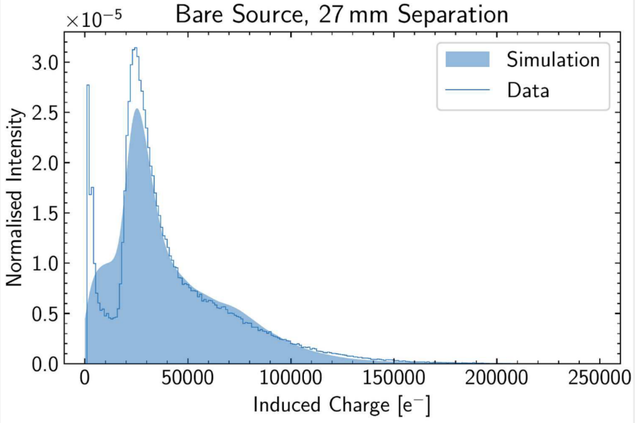
- Using a **strontium-90 source** and a **dipole magnet** to select electron energy from the source
- Using Allpix Squared to investigate this design (S. Wood, APSQWS5)
- Simulations show a **good match with data**, enabling simulations to be reliably used for optimisation



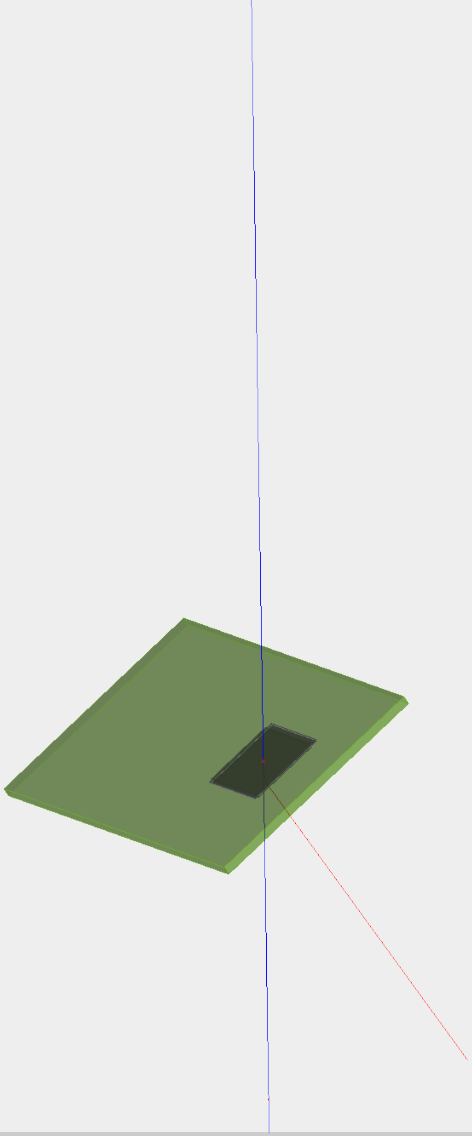
https://indico.cern.ch/event/1346382/contributions/5928860/







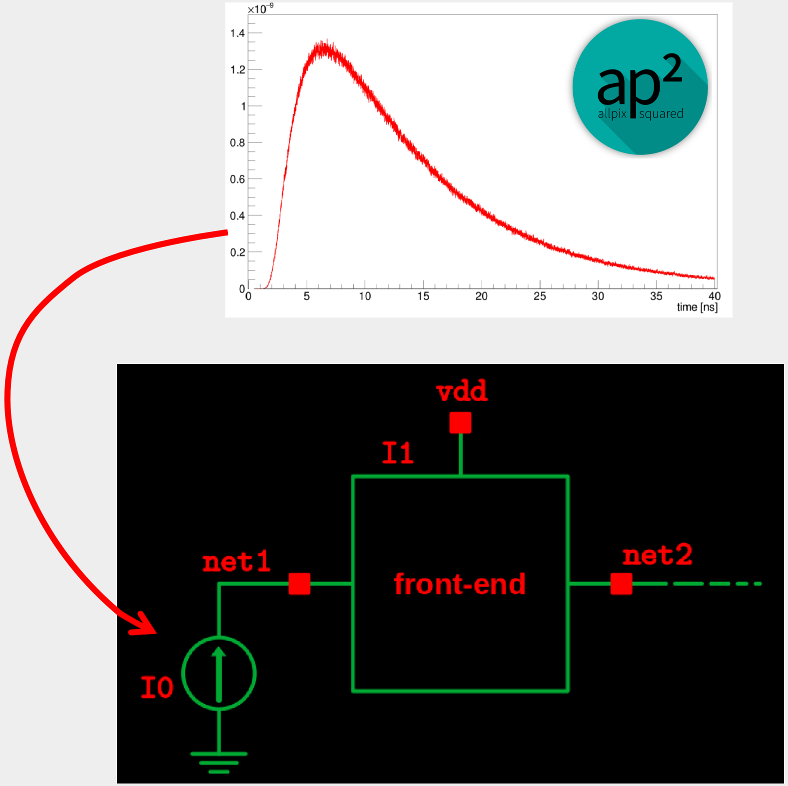
Ongoing and planned developments



Integrating microelectronics simulations

Work by Elio Sacchetti, Manuel Del Rio Viera, Kennedy Caisley, Simon Spannagel

- First step to link physics simulations with **integrated circuit** simulations (merge request 1147)
- Allows **advanced front-end electronics simulations** to be applied to the results of current pulse simulations from Allpix Squared
- Benefit from accuracy of both physical and electrical simulations
  - Leaving the electrical simulation **transparent for the user**
- For now, there is a module that writes files that can be used for front-end simulations for signals generated by Allpix Squared using e.g. the Cadence Spectre SPICE simulator



https://indico.cern.ch/event/1470471/#4-wp1-integrating-microelectro

Integrating microelectronics simulations

- New module implemented that writes a **netlist** that can be used as input for SPICE simulations
  - Text file describing **components and connections of a circuit**
  - Also contains parameters for SPICE simulation
- Module takes a **netlist template**, and adds each current pulse from the Allpix Squared simulation as a **current source** to the circuit
  - Done for each event
  - Generates and saves a **new netlist**
  - Output netlist can then be used as input for SPICE simulations

Netlist example

```
---- Header ----

subckt front_end in out vdd
          ----
          ----
          ----
ends front_end

I0 (net1 0) isource type=pwl wave=[---]
I1 (net1 net2) front_end

simu1 tran stop=100n ……
simulatorOptions ……
```

```
[SPICENetlistWriter]
target = SPECTRE
netlist_template = "./netlist1.scs"
source_type = isource
source_name = I0 # Current source
subckt_name = I1 # Subcircuit name
common_nodes = vdd
```

https://indico.cern.ch/event/1470471/#4-wp1-integrating-microelectro

Integrating microelectronics simulations

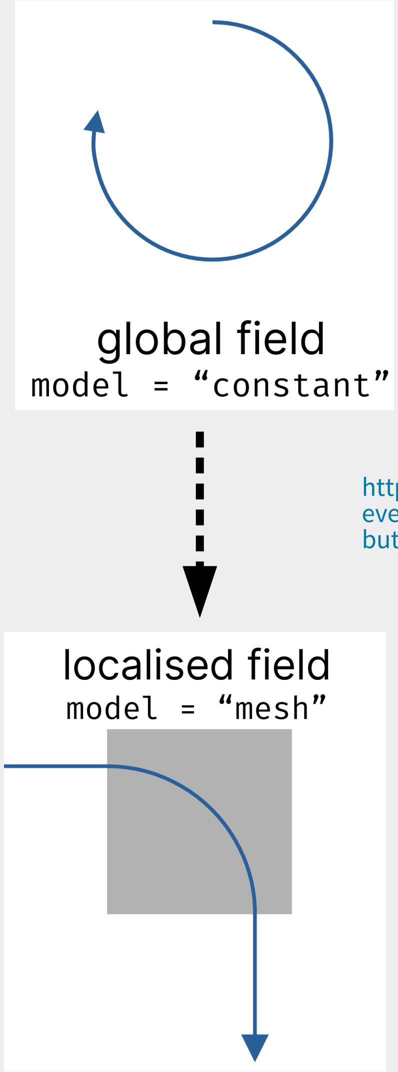
- Module adds a current source for each activated pixel in an event
- **Still to do:** perform the microelectronics simulation as part of the Allpix Squared chain
  - Integrating with e.g. ngspice
  - Writing final outputs to a ROOT file, compatible with the current Allpix Squared way
- Will simplify **simulations and exchange with electronics designers**
  - Easy to simulate detector performance with a netlist from a newly-designed front-end, without full insight into the design
  - Easy to test a new FE under realistic particle interaction conditions

Microelectronics integration flow
1. Read the netlist template containing the front-end description
2. Identifies the instances
3. Writes the time + current data in the isource declaration line (wave=[…])
4. Increments the instances with <pixel_address>
5. Writes a new netlist file containing all the new info.
6. - Not yet implemented – performs the microelectronics simulation in the Allpix^2 event and writes the output in the ROOT file

https://indico.cern.ch/event/1470471/#4-wp1-integrating-microelectro

Non-uniform magnetic fields

- Currently, only uniform magnetic fields filling the full simulated world are possible in Allpix Squared
- Some applications (e.g. the MIP generator on slide 11) require **local magnetic field variations**
- This is being implemented: will allow **loading of magnetic fields from mesh files**, into a Geant4 world
  - Affects both primary particles and charge carriers within sensors
  - Fields can be created by **finite-element calculations**, or be **analytical**
  - Analogous to reading in an electric field in a sensor

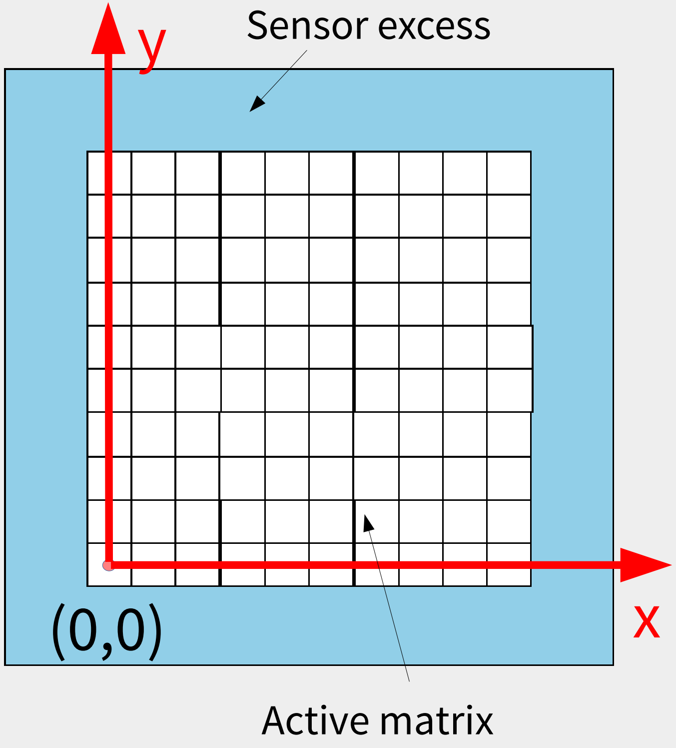


Work by Samuel Wood, University of Oxford

https://indico.cern.ch/event/1346382/contributions/5928860/

Guard ring/sensor excess field inclusion

- Currently, only the field of the **sensitive pixels** is simulated
  - Sufficient for most applications, but does not necessarily represent matrix **edge effects** well
- Sensor excess is included, but without field (to enable scattering and charge carrier motion to/from edge region)
- There have been requests in the Forum to also include fields outside of the active matrix
- Implementations of this are **under discussion**
  - Importing separate electric field and applying it outside of the matrix
  - Symmetry and interface not entirely straightforward

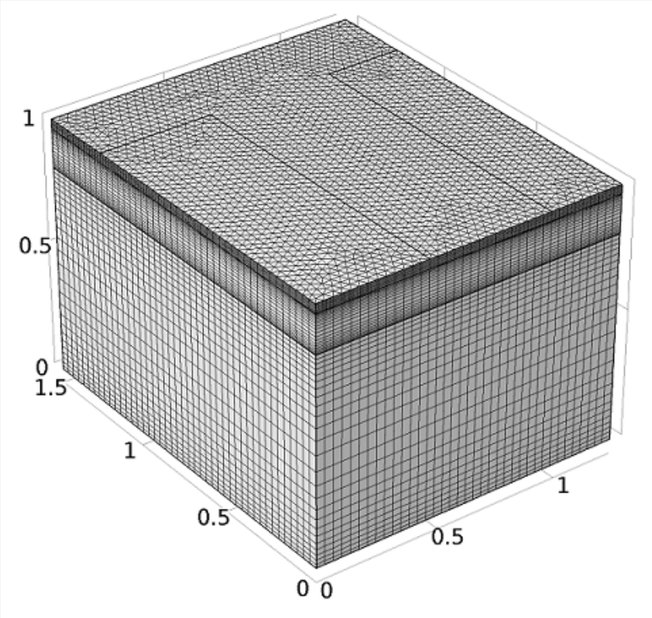


COMSOL field parsing

Work by Rickard Brunskog, KTH

- Some users utilise **COMSOL Multiphysics** to simulate the electric fields and weighting potentials of sensors
- Currently there is no implementation to import this into Allpix Squared
- There is a Python script base in the Forum for field conversion
- This will be implemented more solidly into the framework

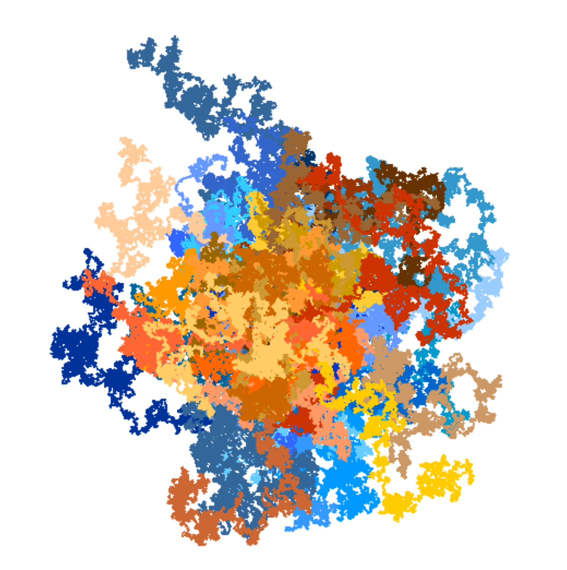
COMSOL



https://www.comsol.com/blogs/how-to-perform-a-3d-analysis-of-a-semiconductor-device

Charge repulsion/self-interactions

- First steps in adding charge carrier density effects
- The problem: charge carriers are treated independently in Allpix Squared
  - Realistic in cases with small deposits, but **requires correction** at high charge densities
  - Recalculating the electric field fully each step is **not feasible** for Monte Carlo simulations – need a simplification
- First implementation by M. Benoit for simple projection propagation
- Ideas being explored:
  - Increasing **initial extent** of charge cloud based on charge
  - Adding modification constant for the diffusion
  - Moving charges apart based on "charge within radius" for each step



Charge carrier movement under zero electric field

Work by M. Benoit, ORNL

Summary and outlook

- Framework under **active development** and **widely used** in the community
  - Use and further development important part of several proposed projects
- Release of version 3.2.0 on the horizon
- Working towards **integration with sophisticated front-end electronics simulations**
- Contributions from the community welcome and encouraged!
  - Tutorials for usage held at e.g. the BTTB workshop

6th Allpix Squared User workshop

7th to 9th of May 2025 at Nikhef, Amsterdam
(Wednesday through Friday, week 19)

Save the date, more info will come soon!

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

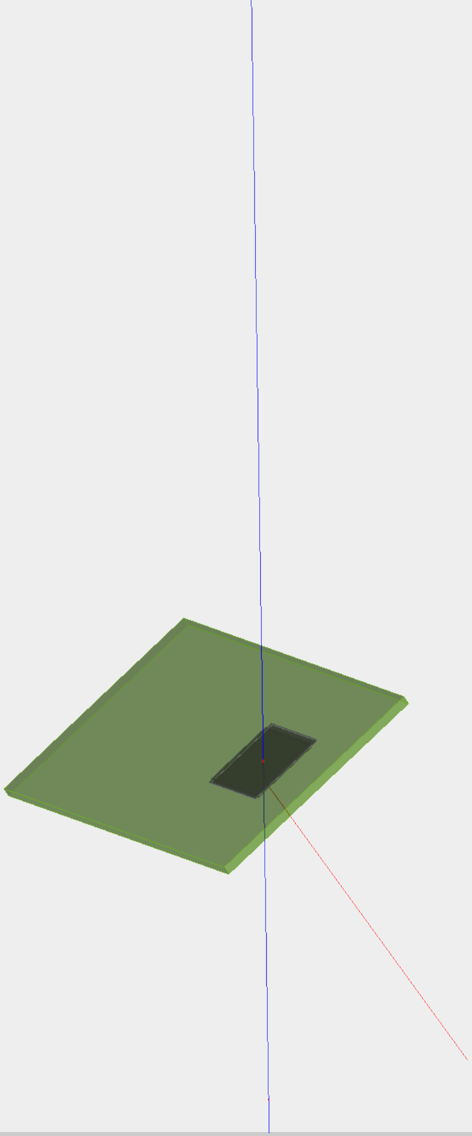
User Manual:
https://cern.ch/allpix-squared/usermanual/allpix-manual.pdf

6th Allpix Squared User workshop

7th to 9th of May 2025 at Nikhef, Amsterdam
(Wednesday through Friday, week 19)

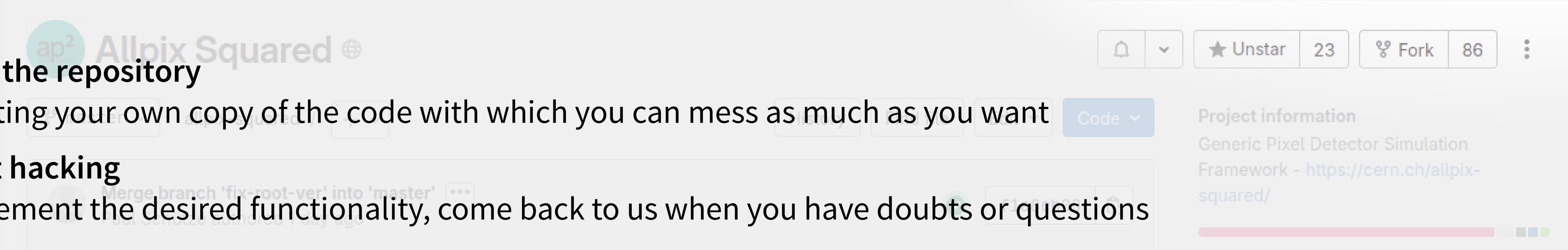
Save the date, more info will come soon!

Backup slides



How To Contribute – A Cookbook

- **Get in touch** – mail, forum, issue tracker, …
  Let's discuss the idea, maybe we have input, maybe others are working on it already



- **Fork the repository**
  Creating your own copy of the code with which you can mess as much as you want
- **Start hacking**
  Implement the desired functionality, come back to us when you have doubts or questions

- **Make sure the CI passes**
  Enable the CI in your fork and publish your new code there – check that the CI works!
- **File a Merge Request**
  This provides us a central point to discuss and review all your code changes

- **See your code being merged and published!**

… 69 contributors & counting

Babar Ali, Czech Technical University in Prague
Mohamed Moanis Ali, GSOC2019 Student
Jay Archer, University of Wollongong
Mathieu Benoit, ORNL
Thomas Billoud, Université de Montréal
Tobias Bisanz, CERN
Bogdan-Mihail Blidaru, Heidelberg University
Sebbe Blokhuizen, Stockholm University
Marco Bomben, Université de Paris
Koen van den Brandt, Nikhef
Ben Bruers, DESY
Carsten Daniel Burgard, DESY
Maximilian Felix Caspar, DESY
Liejian Chen, IHEP Beijing
Dominik Dannheim, CERN
Naomi Davis, DESY
Manuel Alejandro Del Rio Viera, DESY
Malinda de Silva, DESY
Mauricio Donatti, Brazilian Synchrotron Light Laboratory
Katharina Dort, University of Gießen
Neal Gauvin, Université de Genève
Yajun He, DESY
Ryan Heller, LBNL

Lennart Huth, DESY
Daniel Hynds, University of Oxford
Francisco-Jose Iguaz-Gutierrez, Synchrotron SOLEIL
Maoqiang Jing, IHEP Beijing
Moritz Kiehn, Université de Genève
Rafaella Eleni Kotitsa, Université de Genève
Stephan Lachnit, DESY
Hugo Natal da Luz, Czech Technical University in Prague
Salman Maqbool, CERN Summer Student
Stefano Mersi, CERN
Ryuji Moriya, CERN Summer Student
Sebastien Murphy, ETHZ
Andreas Matthias Nürnberg, DESY
Sebastian Pape, TU Dortmund University
Marko Petric, CERN
Florian Michael Pitters, HEPHY
Radek Privara, Palacky University Olomouc
Renato Quagliani, CERN
Nashad Rahman, The Ohio State University
Sabita Rao, GSDocs2020 Student
Daniil Rastorguev, DESY
Edoardo Rossi, DESY
Sara Ruiz Daza, DESY

Jihad Saidi, Université de Genève
Andre Sailer, CERN
Tasneem Saleem, Synchrotron SOLEIL
Arka Santra, Weizman Institute
Christian Scharf, HU Berlin
Enrico Jr. Schioppa, Unisalento and INFN Lecce
Sebastian Schmidt, FAU Erlangen
Paul Schütze, DESY
Sanchit Sharma, Kansas State University
Xin Shi, Institute of High Energy Physics Beijing
Petr Smolyanskiy, Czech Technical University Prague
Viktor Sonesten, GSOC2018 Student
Simon Spannagel, DESY
Reem Taibah, Université de Paris
Ondrej Theiner, Charles University
Annika Vauth, University of Hamburg
Mateus Vicente Barreto Pinto, Université de Genève
Håkan Wennlöf, DESY
Andy Wharton, Lancaster University
Morag Williams, University of Glasgow
Koen Wolters
Samuel Wood, University of Oxford
Jixing Ye, University of Trento

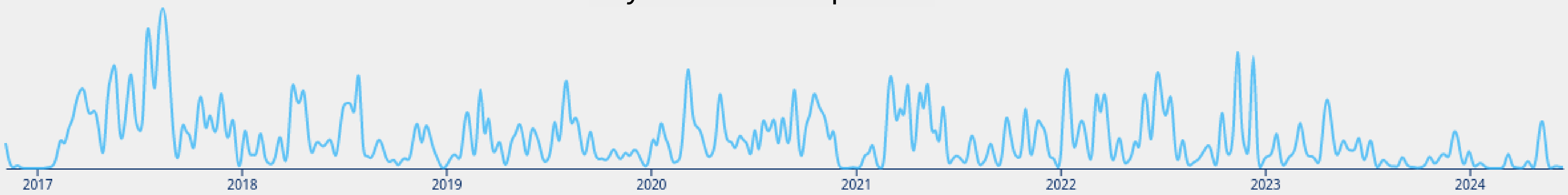
The Allpix² Framework

- Development of framework started within **CLICdp Collaboration**
- Now > 7 years of development with
  - 52 releases, current version 3.1.1
  - 5 user workshops
  - Close to 70 code contributors

Development based on four principles:

I. Integration of Existing Toolkits
II. Well-Tested & Validated Algorithms
III. Low Entry Barrier for New Users
IV. Clean & Maintainable Code

7 years of development



8882 commits | Last commit ≈ 4 days ago

Well-Tested & Validated Algorithms

Simulations provide insights into physical processes – but only if they model them correctly!

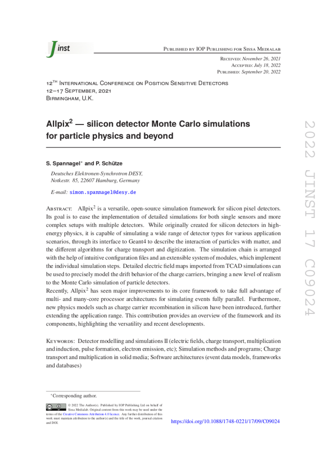
- Validation of algorithms is a crucial and time-consuming process
- So far 5 User workshops for exchange of the community, discussions, planning…

- Validating as much as possible against data
- Publishing reference studies including full simulation configuration used
- Providing automated tests for every new feature

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

NIMA 964 (2020) 163784
doi:10.1016/j.nima.2020.163784

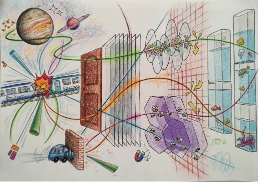
NIMA 1031 (2022) 166491
doi:10.1016/j.nima.2022.166491



JINST 17 (2022) C09024
doi:10.1088/1748-0221/17/09/C09024

In preparation…

In preparation…
arXiv:2408.00027



Semiconductor Detector MC Simulations DRD

Community Meeting 22-23/03/2023
Implementation of TF3
Solid State Detectors

- Complexity of detectors increases, more and more technologies available,
- different approaches combined (e.g. monolithic + LGAD)
  - Necessity of MC simulations growing
  - Some sensors / setups impractical to simulate in TCAD (time limitation, stochastics)
  - Community needs *common* flexible, tested & supported MC simulation tools

- Using Monte Carlo methods to describe detector response is not new
- Creation & proliferation of many different codes for detector simulation
  - Experiment-specific
  - Specialized on specific detectors
  - Inclusion only of effects relevant to that one simulation
  - Written as part of a PhD thesis, abandoned afterwards

- Would be great to collate features in commonly maintained software (->SM)
  - Having several tools is valuable as testbed for algorithms
  - Well-maintained & supported common software will significantly ease use in community