ANTS2 toolkit:
Detector optimization and experimental data processing for position sensitive scintillation detectors

Andrey Morozov (LIP-Coimbra)

Coimbra, November 2018

ANTS2 development team

Developers:

Andrey Morozov
Vladimir Solovov
Raimundo Martins (not active anymore)

Neural network module:

Francisco Neves

Consultant:

Vitaly Chepel

Work on ANTS2 has started in 2013 based on

- ANTS package developed at the neutron detector group of LIP
- B-spline LRF parameterization algorithms of V.Solovov

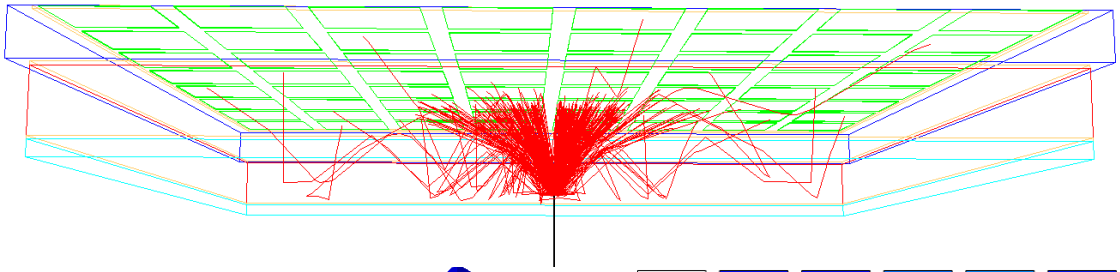
Content

- Position sensitive scintillation detectors (PSSD)
- ANTS2 highlights
- Reconstruction module
- Simulation module
- Advanced optimization tools
- Experimental data import
- Implementation details
- Future?

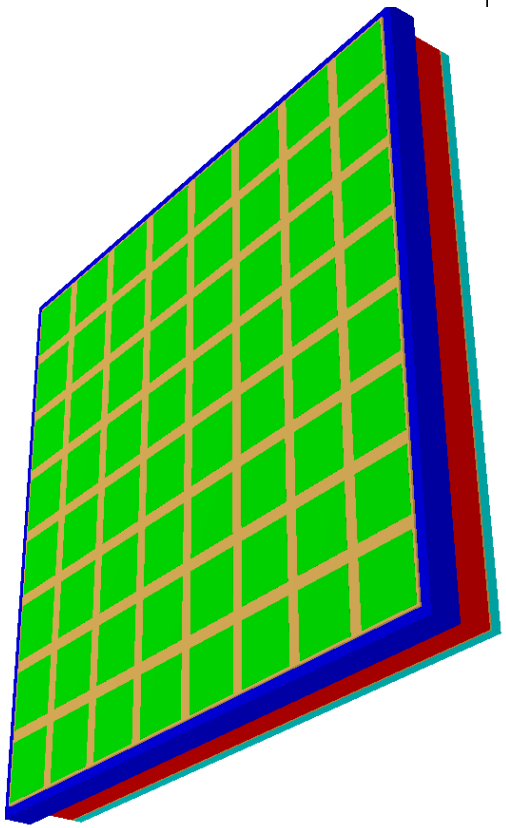
Position sensitive scintillation detectors (PSSD)

PSSD examples

Medical gamma camera



Gamma ray interactions result in emission of optical photons.



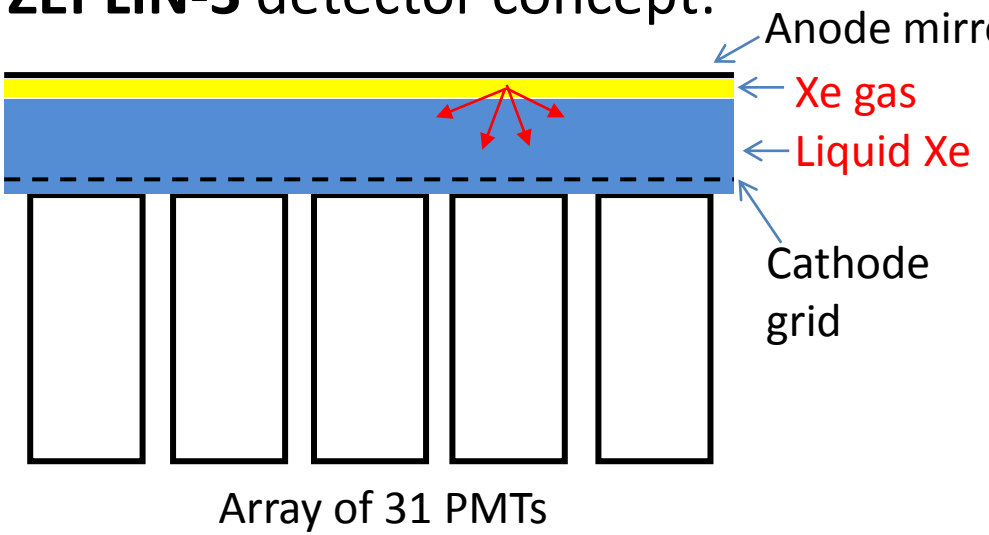
0	1	1	11	13	2	0	0
0	1	12	58	78	9	1	1
0	0	12	68	97	6	1	0
1	0	1	13	10	3	1	0
0	1	0	1	0	1	0	0
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1
0	0	0	0	1	0	0	0

Distribution of the sensor signals is used to reconstruct **XY position and energy** of the scintillation event.

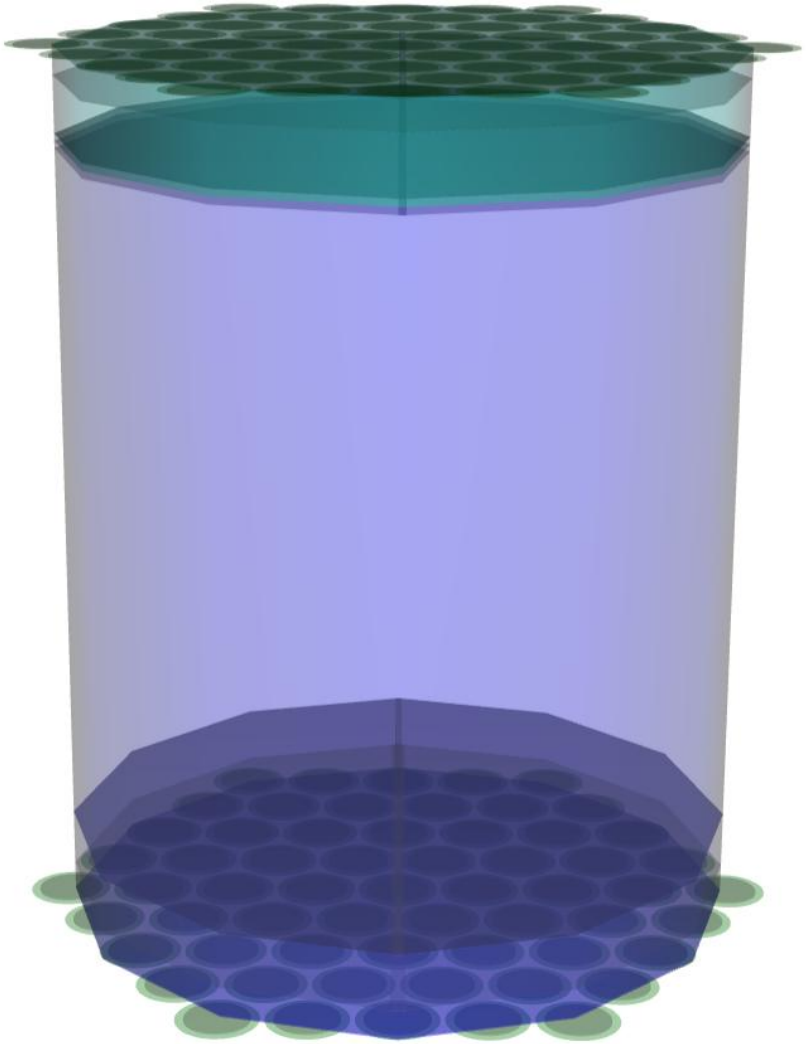
Another example with the same geometry:
thermal neutron detector with ^{6}Li glass scintillator

Dual-phase dark matter detectors

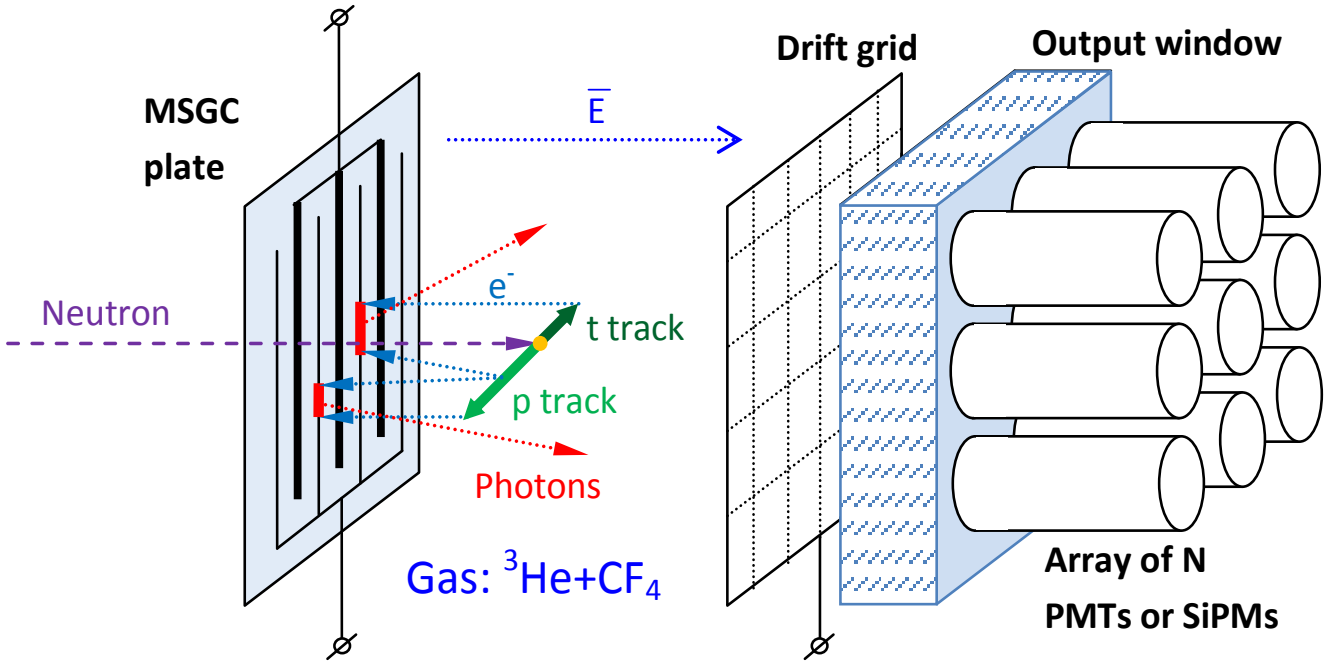
ZEPLIN-3 detector concept:

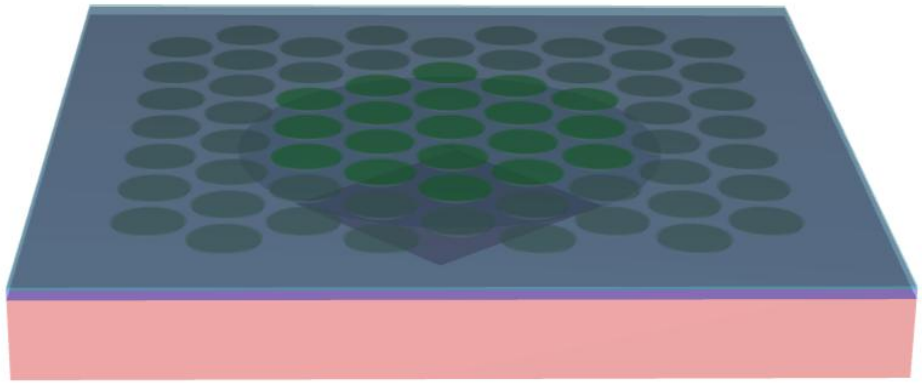


ANTS2 model of LUX detector:



Thermal neutron scintillation detectors



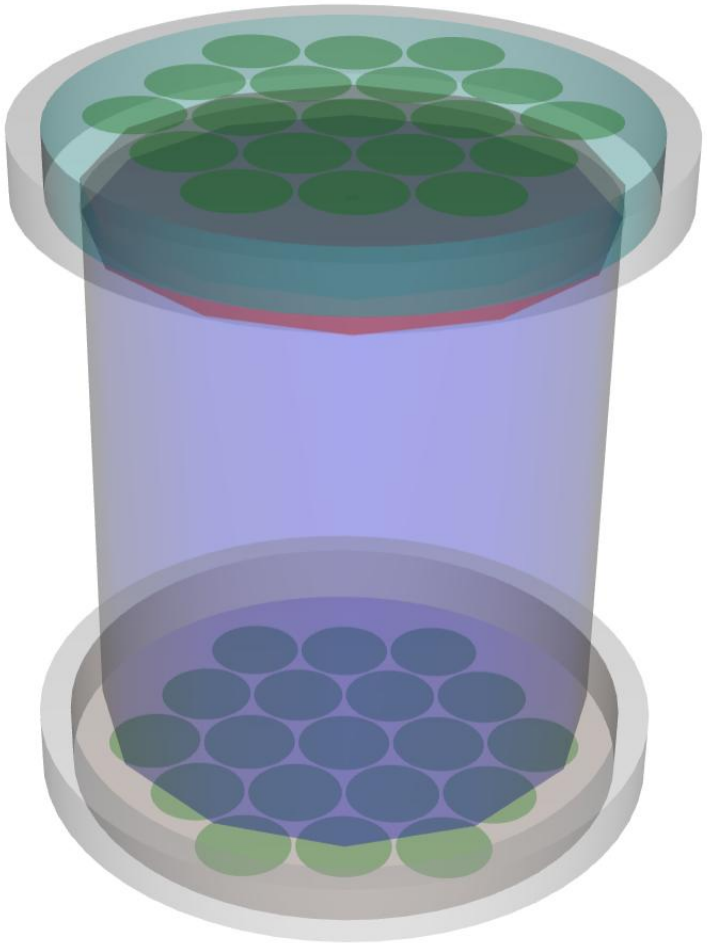


ANTS2 model of GSPC-19 detector

Neutrino detectors

Design is “inherited” from the dual-phase dark matter detectors:

Coherent scattering of a neutrino on a Xe atom results in recoil of the atom, leading to generation of primary ionization in liquid xenon.



ANTS2 model of the RED100 neutrino detector

ANTS2 highlights

ANTS2 highlights

ANTS2 targets:

- Development and optimization of PSSDs
- Optimization of event reconstruction techniques
- Reconstruction of response of light sensors
- Reconstruction of events for experimental datasets

ANTS2 highlights

Simulations

- 3D, custom detector geometry
- Generation and tracking of gamma rays, neutrons and positive ions
- Primary and secondary scintillation
- Time- and wavelength-resolved tracking of optical photons
- Photon scattering, wavelength shifters
- Signal generation for PMTs and SiPMs

Scintillation event reconstruction (XYZ + energy):

- Statistical algorithms
  - Contracting grids on GPUs (real-time operation)
- Artificial neural networks
- kNN-based reconstruction

ANTS2 highlights

Experimental data processing

- Import and preprocessing of experimental data
- Event discrimination tools (noise and multiple event rejection)

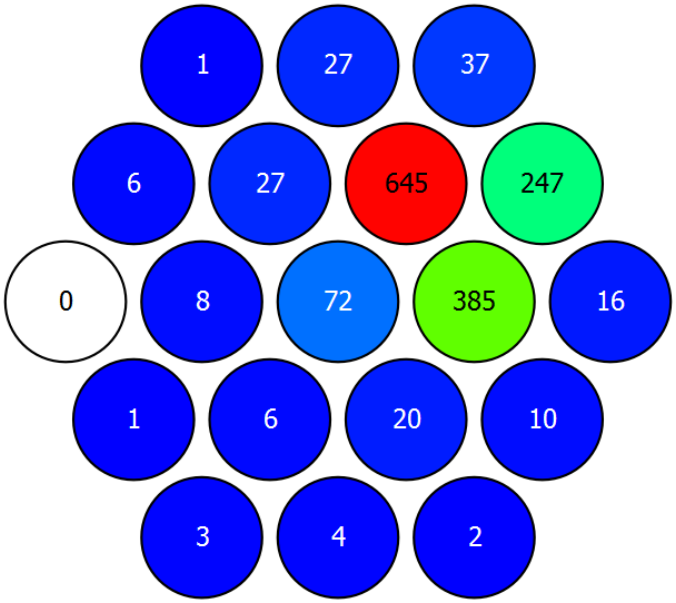
Detector response reconstruction

- B-spline and analytic parameterization of sensor response
- Grouping of the sensor to take advantage of the array symmetry
- Iterative reconstruction of the detector response
- A framework of tools for determination of the sensor gains

Collection of tools for characterization of PSSD performance

Reconstruction module

Center of Gravity method



Light sensor signal distribution for a scintillation event

Traditional approach for a PSSD:

Center of Gravity (CoG) method

$$x = \frac{\sum X_i S_i}{\sum S_i} \qquad y = \frac{\sum Y_i S_i}{\sum S_i}$$

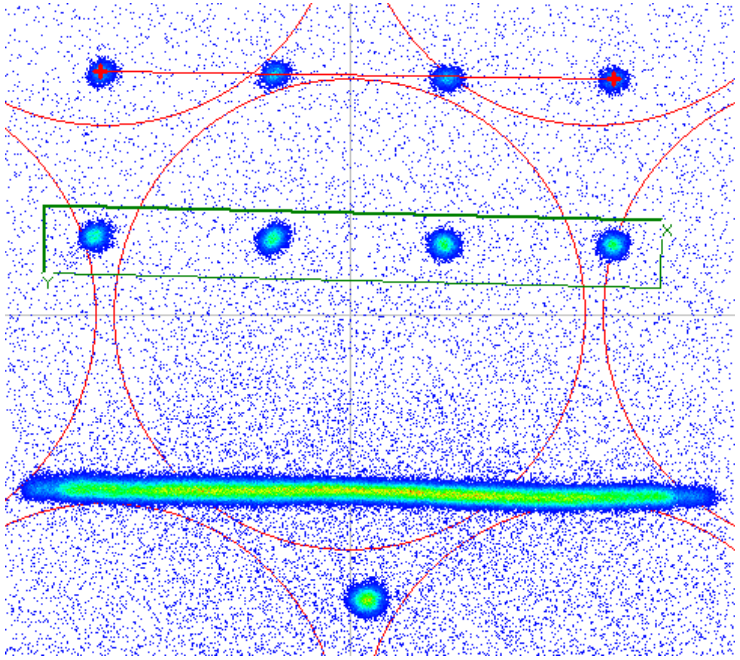
Robust, Simple to implement

Drawbacks:

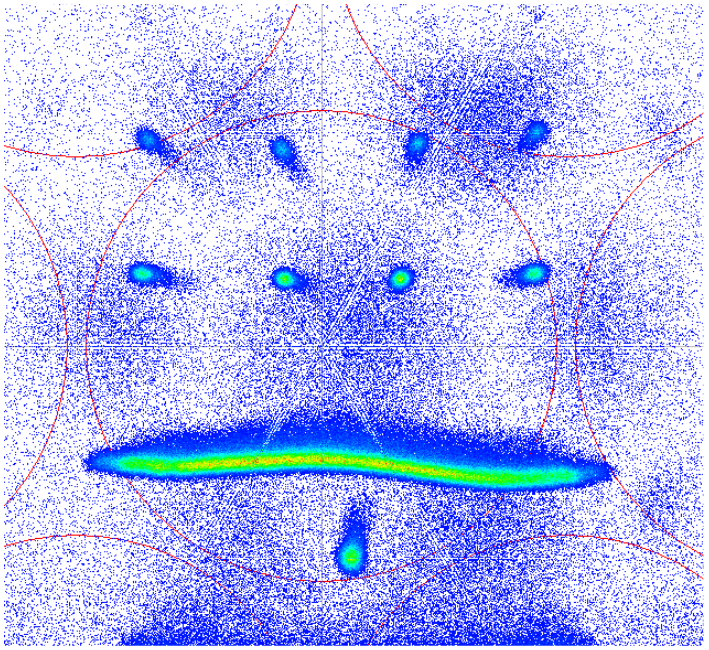
- Systematic distortions in position and energy
- Sensitive to gain drift
- Resolution typically below the limit given by the photon statistics
- Noise event discrimination can be difficult

Center of gravity is sufficient?

Expectation:



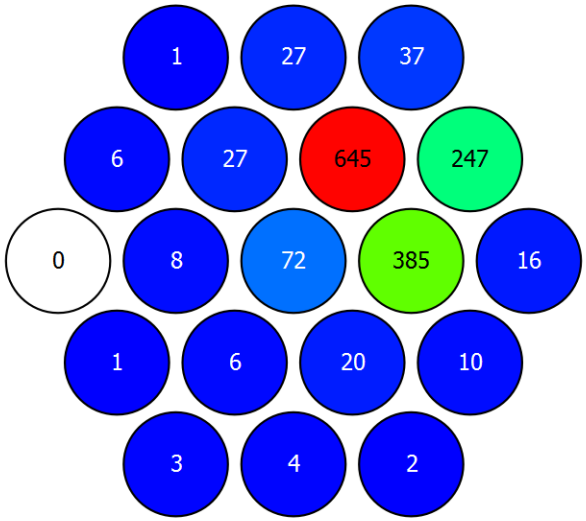
Center of Gravity reconstruction
(PMT gains are already equalized!):



Could be a long journey to meet the expectations!

Statistical reconstruction

Measured signals

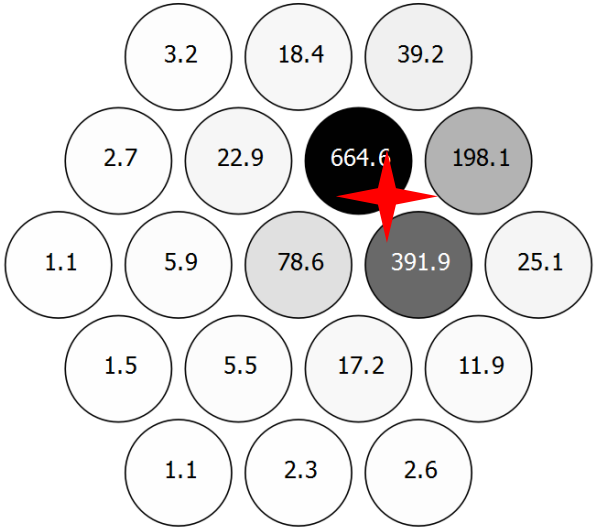


Finding the best match between the measured sensor signals s_i and the expected signals $S_i(x, y, energy)$

Potentially distortion-free and can give the best possible resolution

Requires detailed knowledge of the detector response

Expected signals



In ANTS2 we parameterize the response of the detector with **Light Response Functions**

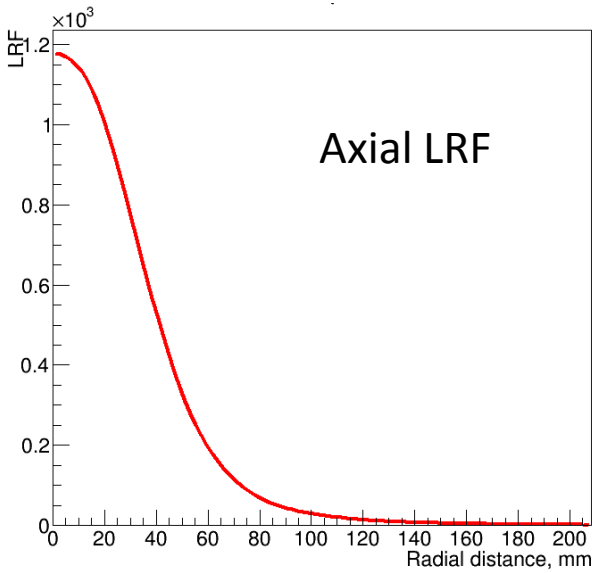
LRF describes the dependence of the average signal of a sensor on the light source position.

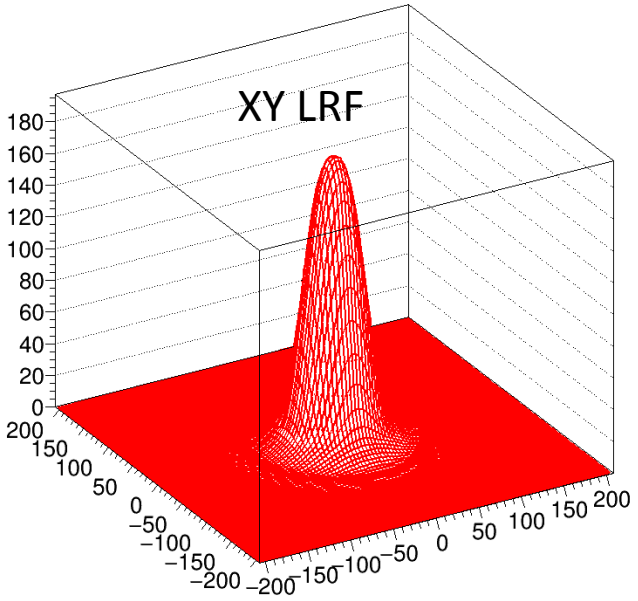
LRF parameterization in ANTS2

1. Cubic B-spline parameterization

LRF types:

- *Axial*: Functions of the R: radial distance between the source and the sensor center in the XY plane)
- *XY*: Functions of X and Y
- *3D axial*: Functions of R and Z
- *Slice 3D*: A set of *XY* LRFs for several Z planes
- *Full 3D*: Functions of X, Y and Z





LRF parameterization

2. Analytic functions through script

A script can be used to define parameterization function.

The coordinate options are similar to the B-spline parameterization schemes.

```
function eval(r) {
    return A * Math.exp( -0.5 * r * r / S2) + C
}
```

3. Plug-ins

It is possible to provide custom LRF types using the plug-in interface of ANTS2.

Advanced options:

The LRF of each sensor can be configured as a sum of several LRF components, each of which is parameterized with one of the schemes described above.

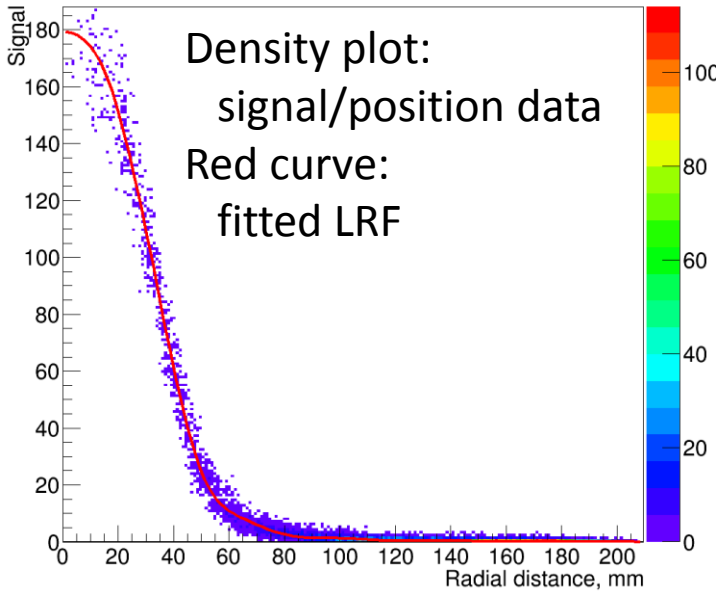
LRF fitting

Need two datasets:

- **signals** of the sensors
- corresponding source **positions**

The position data can be

- **true** positions of the light source
  (simulations or scan with a pencil beam)
- **reconstructed** positions



ANTS2 can take into account the symmetry of the sensor arrays
to group light sensors and use a common LRF profile for several sensors.

B-spline library allows to set restrains on the LRF profile, e.g.:

- non-increasing function
- always positive
- flat-top

Statistical reconstruction in ANTS2

How to find the position which gives the best match between the sensor signals and their values predicted by the LRF model?

- Need a way to characterize the **difference** in the signal sets
  - Least Squares (euclidean norm)
  - Maximum Likelihood (logarithm of the likelihood function, assuming Poisson distribution of the sensor signals)

- Need a **search** algorithm
  - Minimizer (Simplex or Migrad algorithms from Minuit2 library)
  - Contracting grids
    - On CPU using multithreading
    - On GPU using CUDA (custom kernerls):
      reached rates of ~1e6 events per second for 37 PMT detector!

Scripting tools (with multithreading capabilities) are provided to implement custom solutions

ANN and kNN-based reconstruction

Artificial neural networks (ANN)

- Configures and trains the network using a calibration dataset
- Uses the trained network for event reconstruction
- The cascade algorithm allows to automatically find the best ANN topology.

kNN searches

- Operates in a multidimensional space, where
  each dimension represents signal value of one of the sensors
- Detector response model is represented by a set of calibration events
- During reconstruction, the position of an event is determined
  as a centroid of the known positions of the closest calibration events, found by the kNN search.

Scan is a must for calibration?

All methods discussed above **require** calibration datasets with sensor signals and **known source positions!**

Can be very difficult / unpractical / impossible to provide.

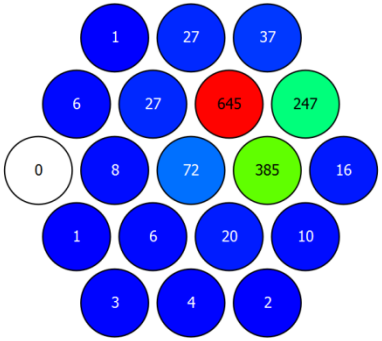
Is it possible to soften the requirements and use
only flood field calibration datasets
(no information of the true event positions)?

Iterative reconstruction of LRFs

For an array of **N** sensors, one event results in **N equations**
Event reconstruction finds X + Y + Energy → **3 variables**

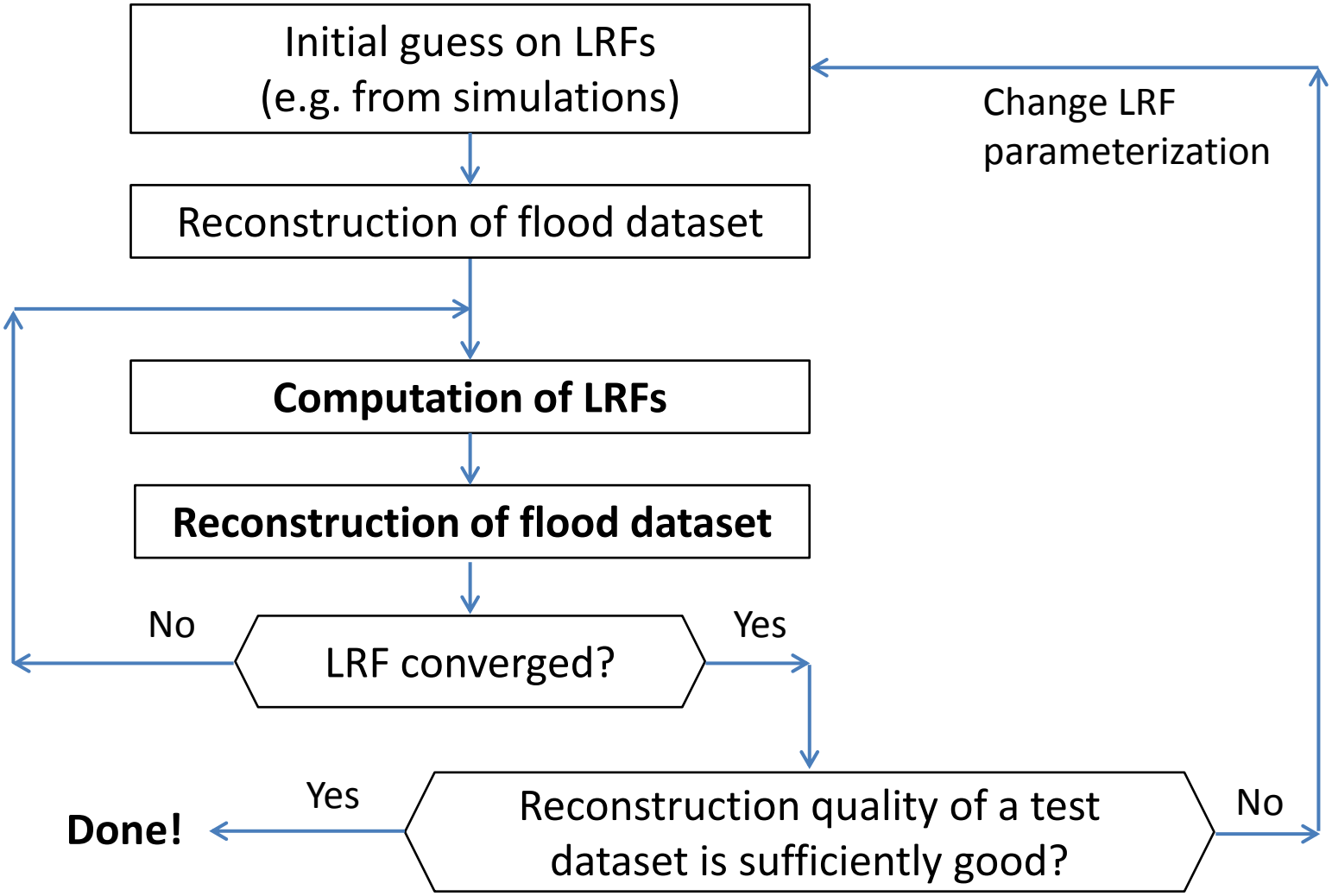
If N is much larger than 3 we have an over-determined system

The distribution of the signals also “encodes” information on the detector spatial response.



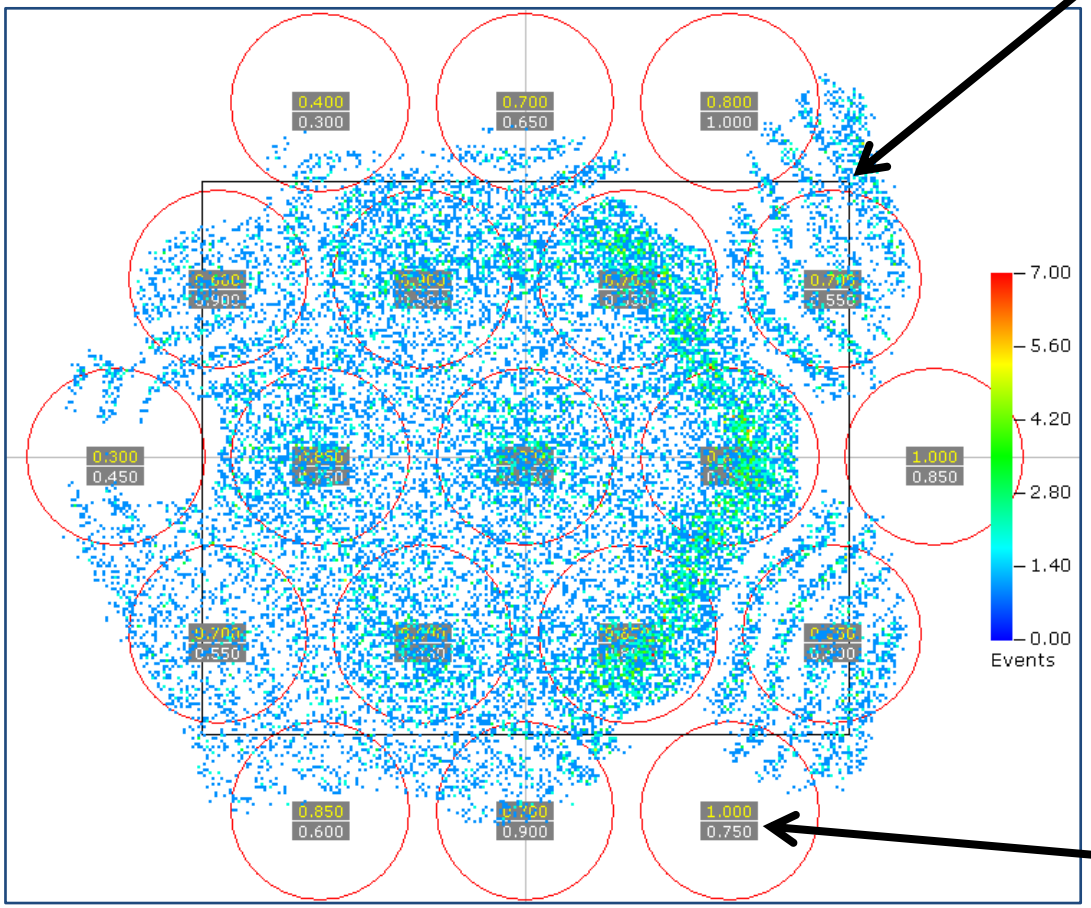
Using many events, distributed over the entire detector active area (**flood field**), it may be possible to **reconstruct LRFs** for individual sensors.

Iterative method in a nutshell

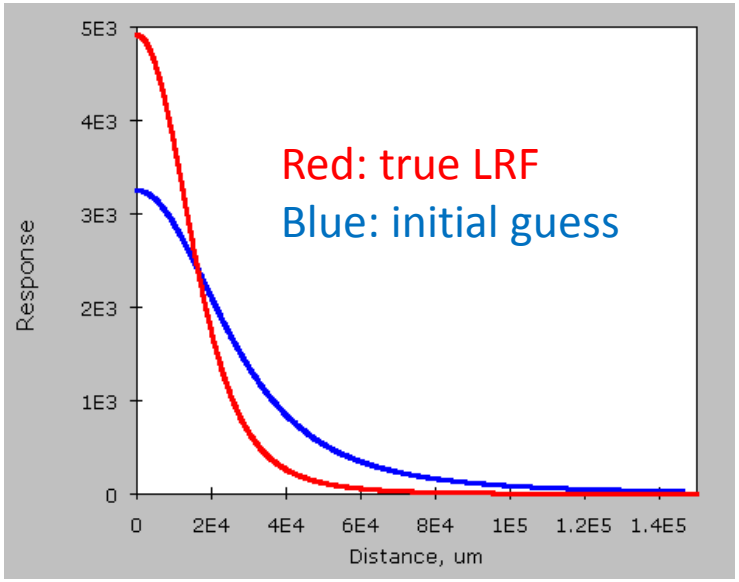


Iterative method

Initial guess on LRF



Flood field: events uniformly cover the outlined area (black rectangle)
Energy spread: 20% FWHM

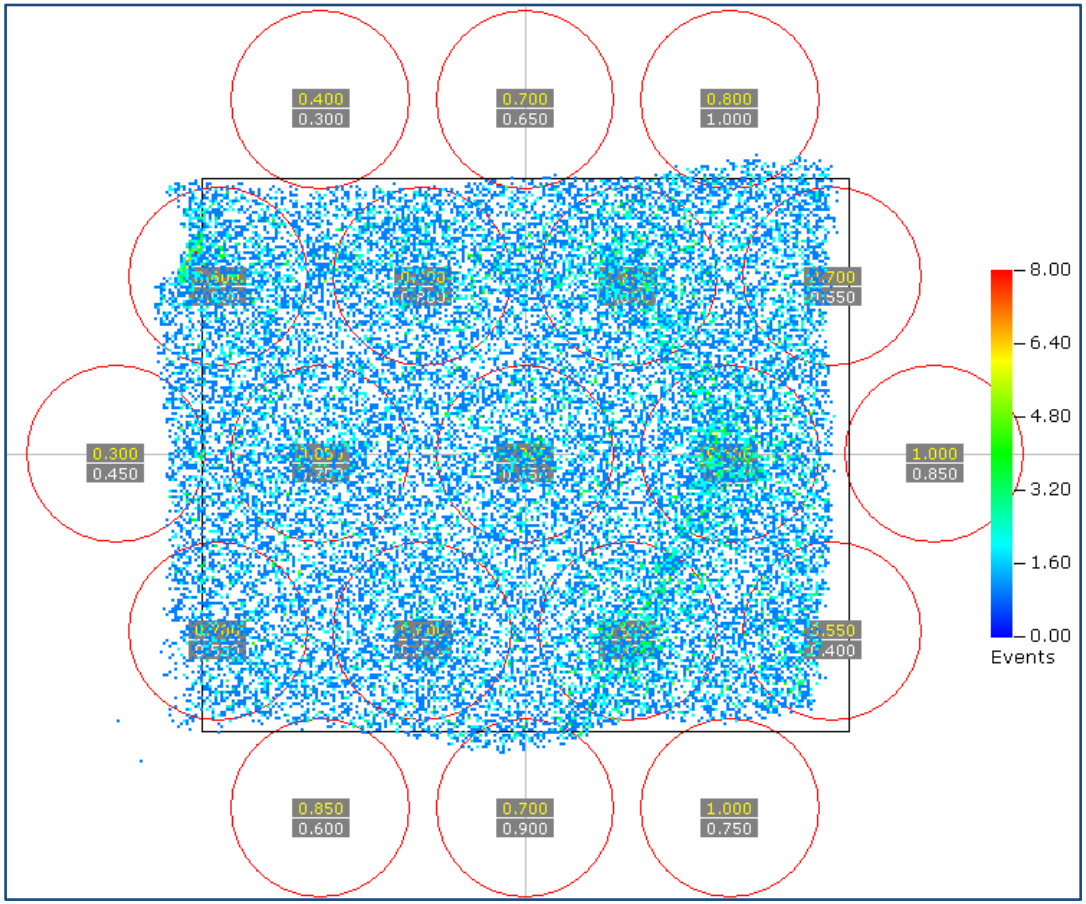


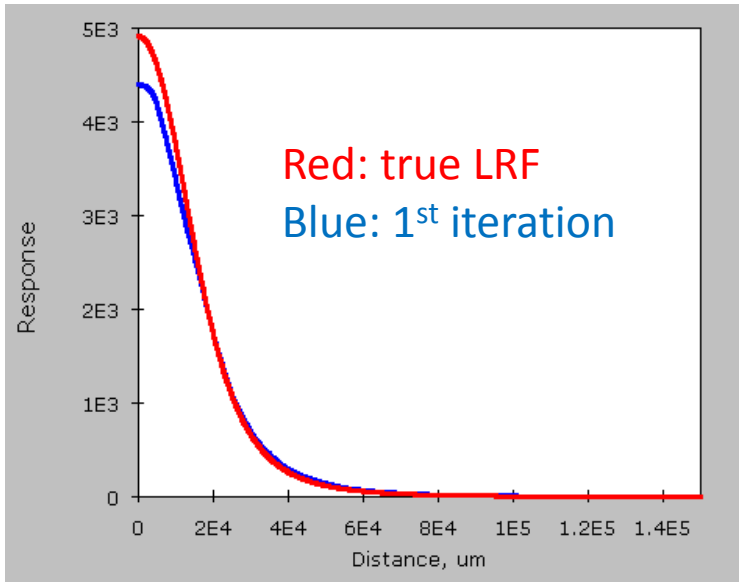
Initial gains: <u>random</u>, up to 30% off the true values

Color scale: density of reconstructed events

Iterative method

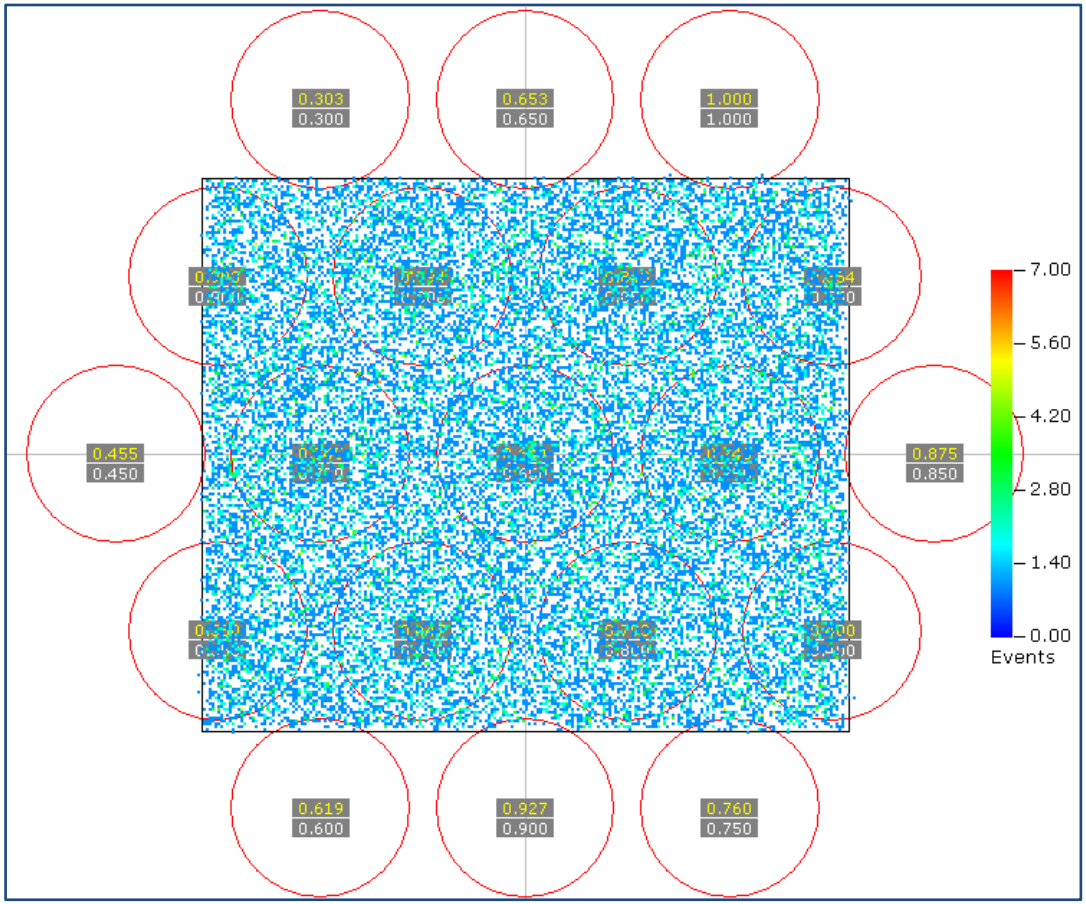
After one iteration



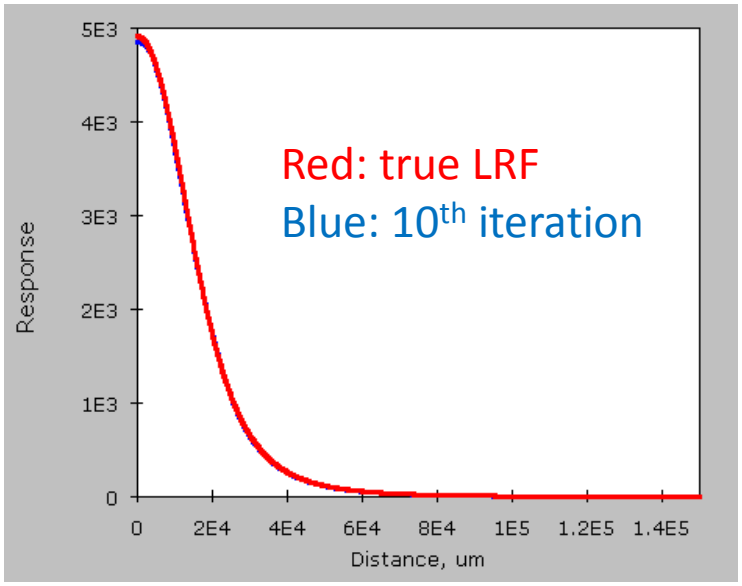


Iterative method

After 10 iterations



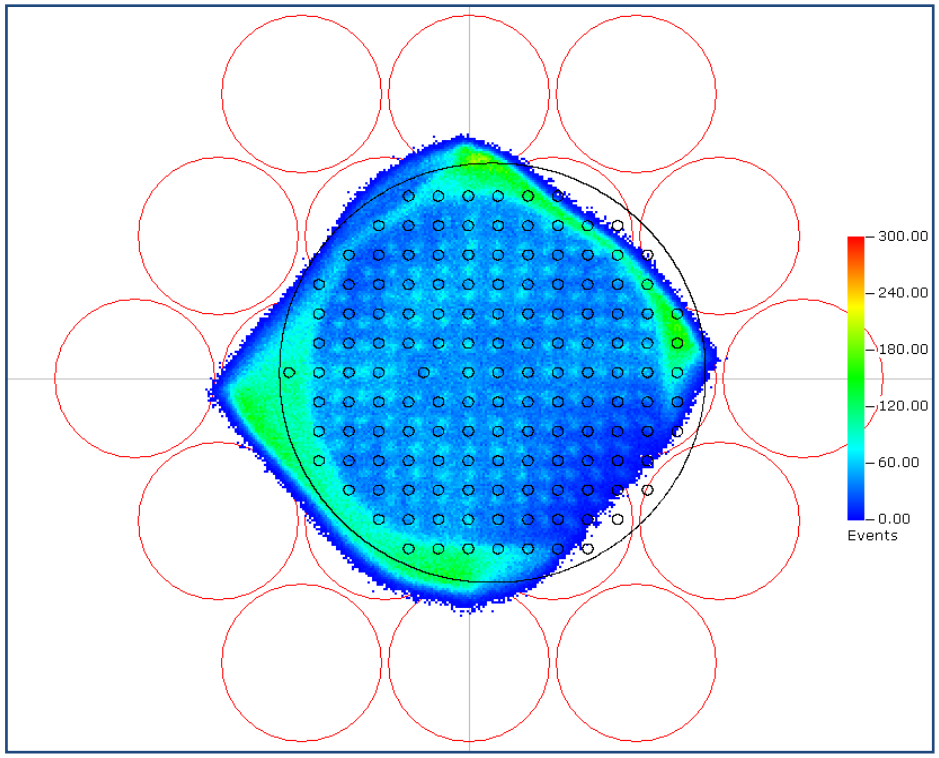
Nearly perfect match!



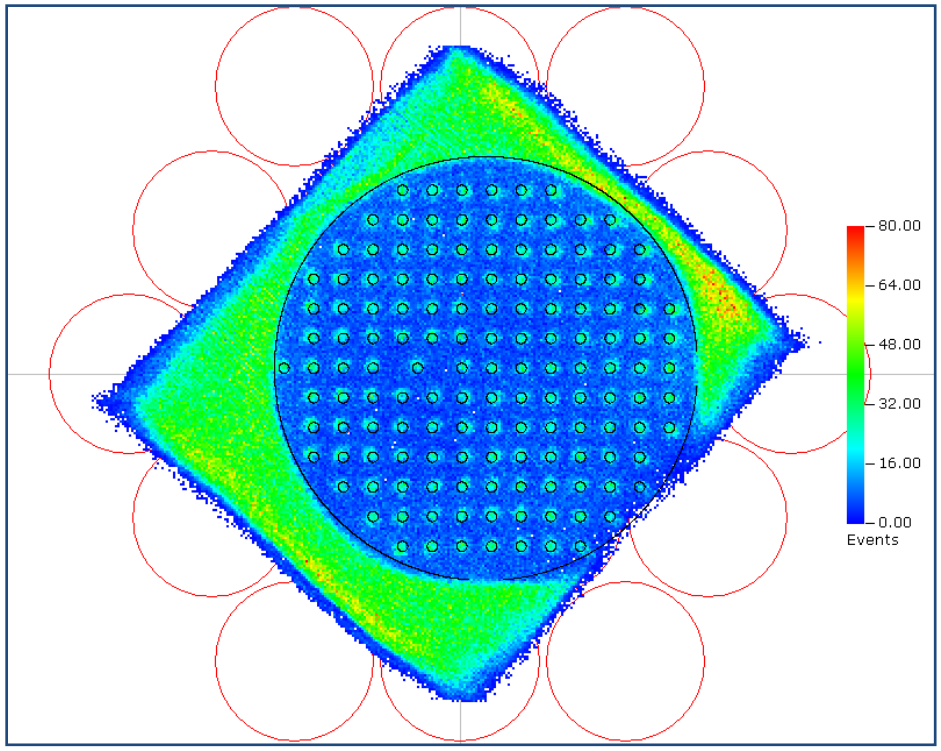
Reconstructed gains are within 5% from the true values.

GSPC-19 neutron detector

Good match between the mask contour and the reconstructed image for **experimental data:**



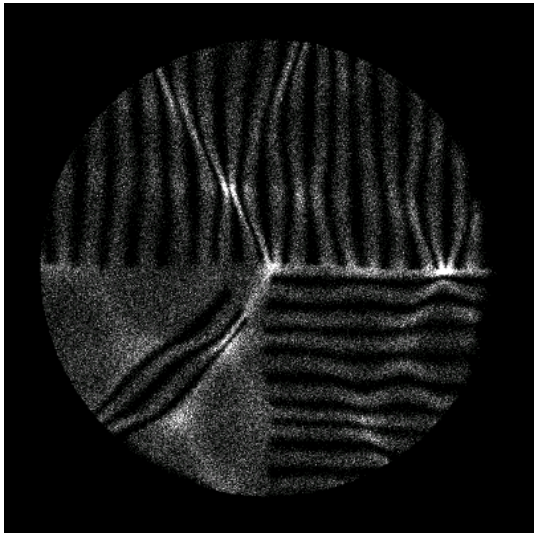
CoG reconstruction

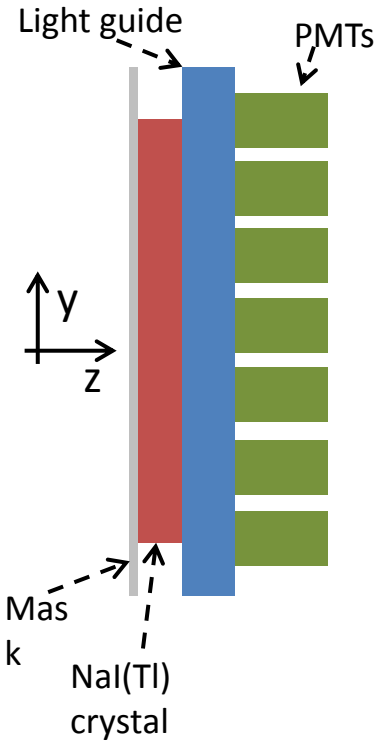


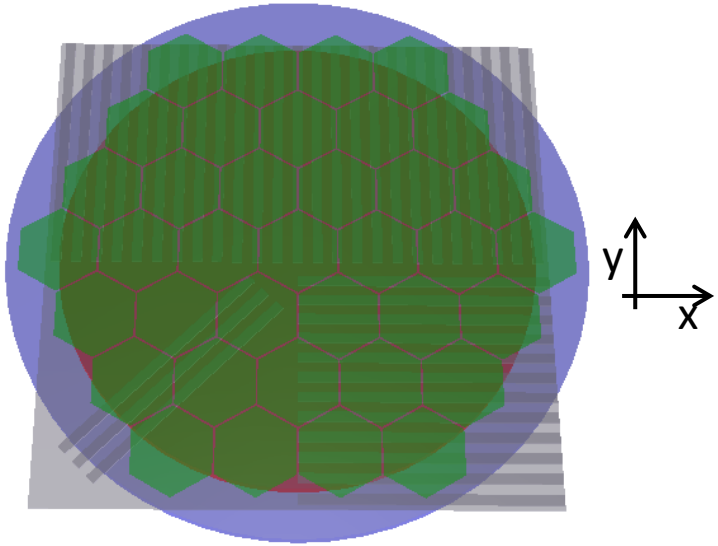
Reconstruction using LRFs obtained with the **iterative method**

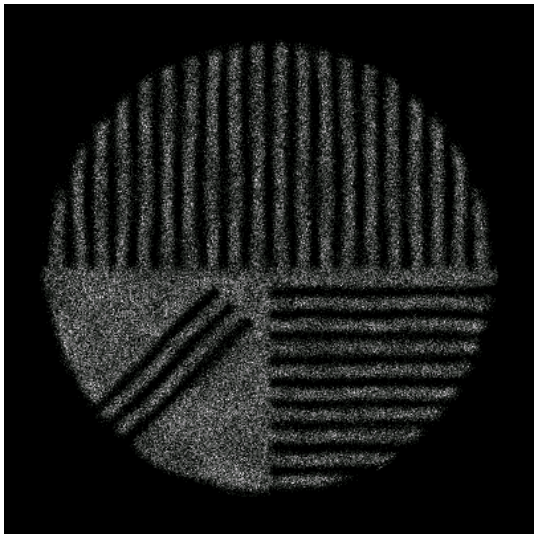
Clinical gamma camera

Reconstruction using
the initial guess on the LRFs



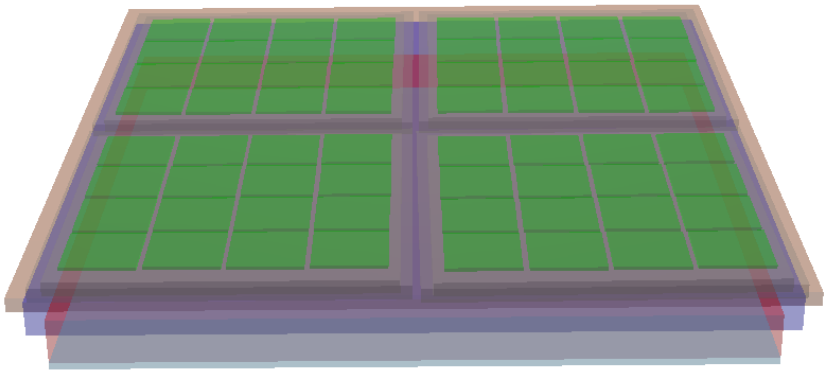


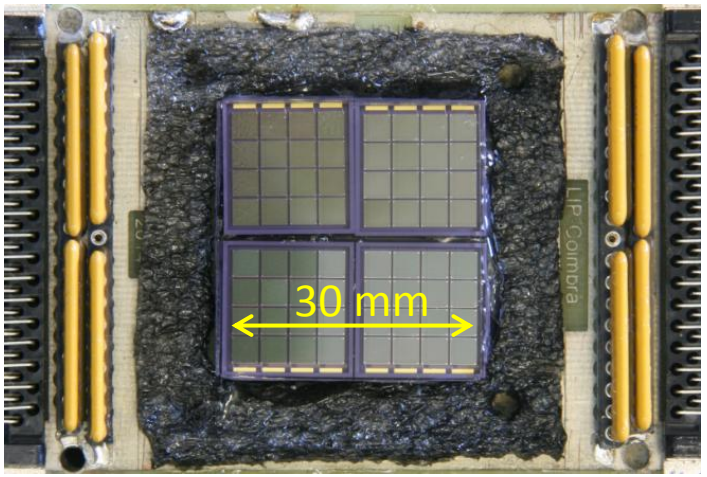


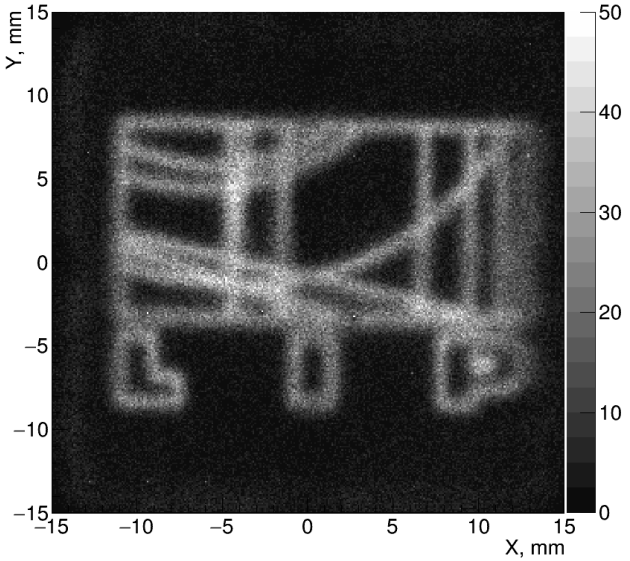


... and after 30 iterations.

Compact gamma camera







Reconstruction using the LRFs obtained after 12 iterations

Simulation module

Simulation module

What do we expect from the simulation module?

- Fully custom 3D detector geometry
- Fresnel / Snell's + custom properties of light scattering on material interfaces
- Simulate primary/secondary scintillation and PMT/SiPM signal generation
- Very fast photon tracing
- Optional feature: simulate propagation and interactions of gamma rays, neutrons and positive ions with detector materials

Decided against using Geant4 and have chosen to build a custom simulator based on **TGeoManager** 3D navigator from **CERN ROOT**

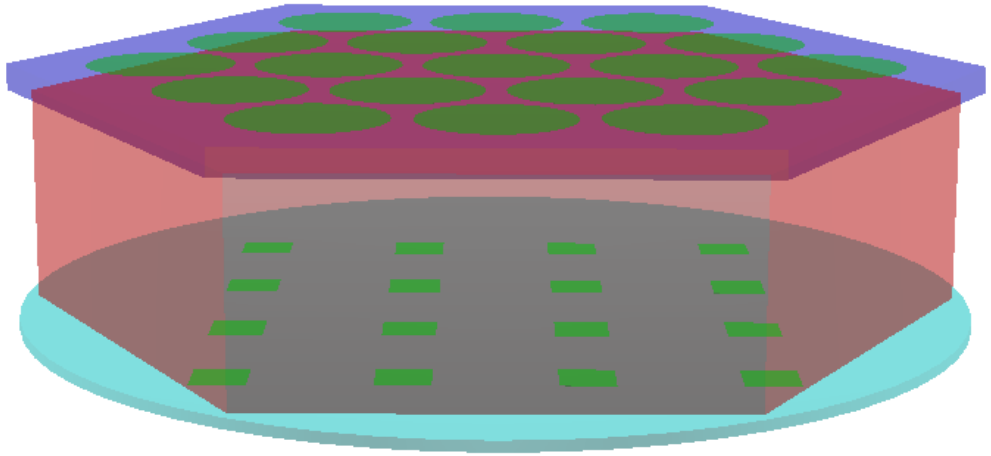
Attempt to follow the "controlled complexity" approach

Detector geometry

ANTS2 offers three approaches to configure the detector geometry.

1. **Detector as a stack of *slabs***

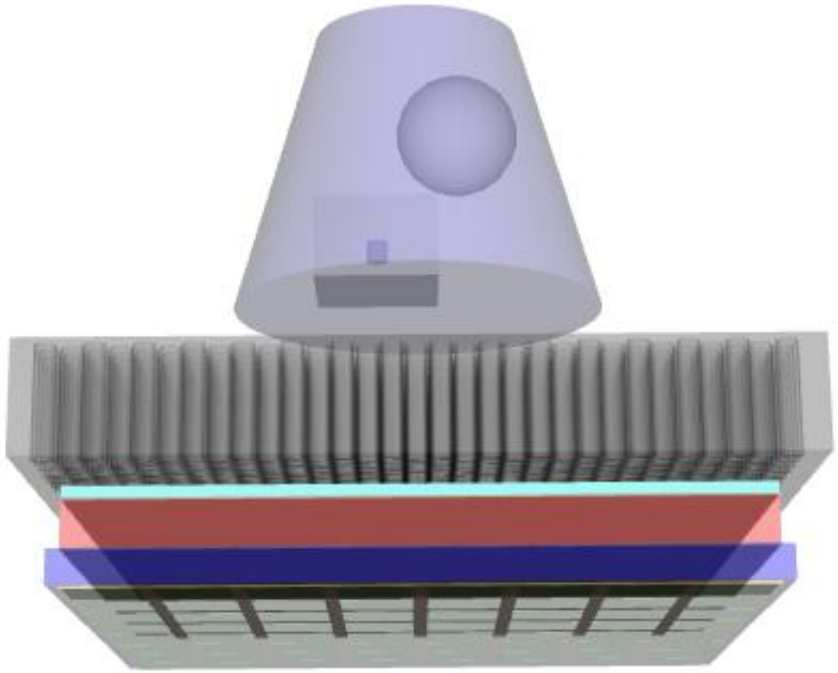
Detector is considered to be a stack of *slabs* "sandwiched" between two arrays of sensors. *Slabs* can be cylinders or polygons of arbitrary number of edges.



Very quick to configure – well suited for detector prototyping phase!

Detector geometry

2. **Custom TGeoShape-based objects**
   A new object can be placed inside any other at an arbitrary position and with an arbitrary orientation.
   Extremely flexible: Tens of shapes supported by the TGeoManager, including composite shapes (union, subtraction and intersection)



Tools are provided for handling of:

- Arrays of objects
- Stacks
- Compound lightguides
- Grids/meshes of wires

Geometry can be build staring from *slabs* defined with the first method.

Detector geometry

3. **Import from a GDML file**

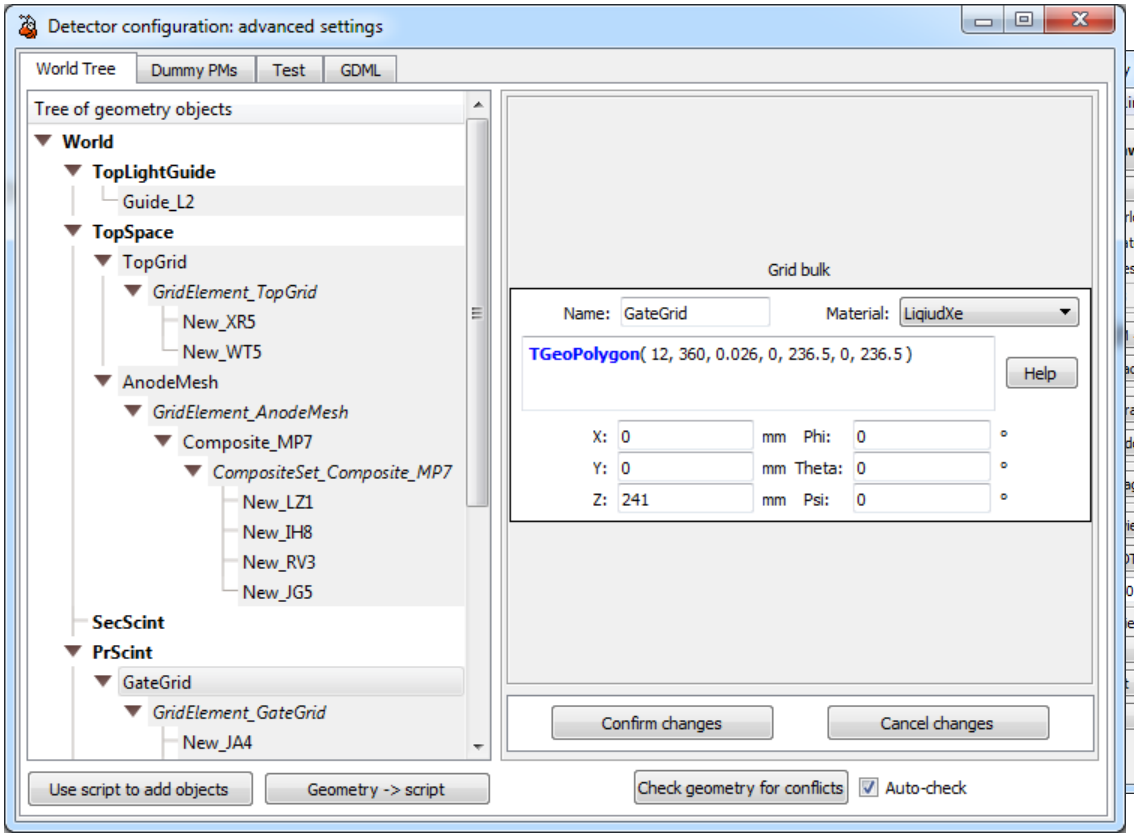
   Geometry defined in a GDML file (XML formatted) can be directly imported to ANTS2 or converted to ANTS2 system.

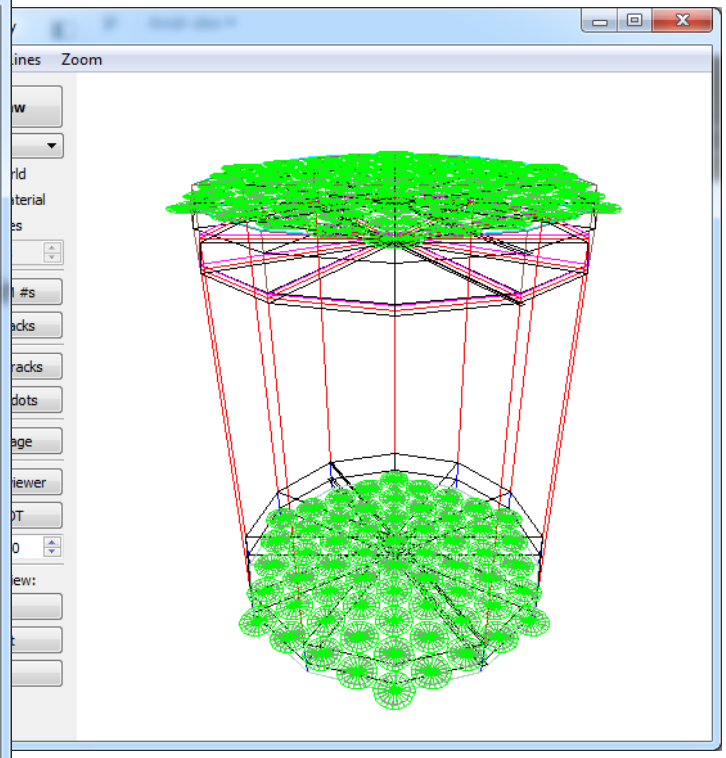
   GDML standard is supported by Geant4 (and GeantV) and CERN ROOT TGeo navigator.

Detector geometry

Geometry can be defined or modified:

- in an interactive GUI
- using scripting system (JavaScript or Python)
- directly in the configuration files (JSON format: attribute + value pairs)





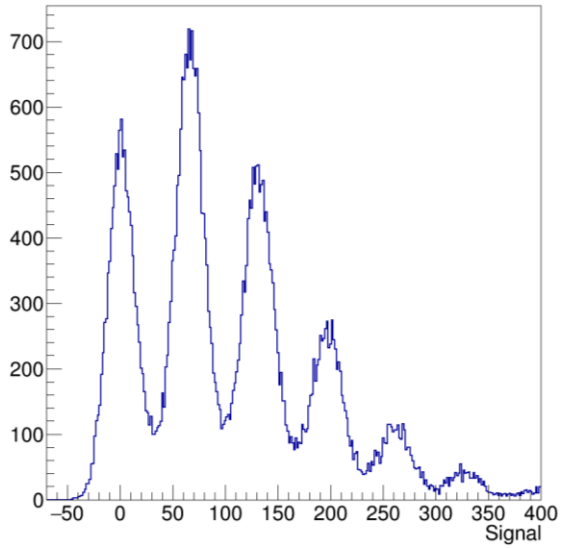
Optical sensors

Photomultiplier tubes (**PMT**) and silicon photomultipliers (**SiPM**):

The general properties of a sensor *model* are shared by all individual sensors of that type: the **size/shape** (cylinder, box, polygon, spherical segment) and the **material** of the optical interface.

If required, the following options can also be configured:

- Photon detection efficiency (optionally function of wavelength, angle of incidence and position on the sensor)
- The number of microcells, dark count and optical crosstalk (SiPMs only)
- Signal readout properties
  - single photoelectron spectrum
  - electronic noise
  - signal digitalization properties



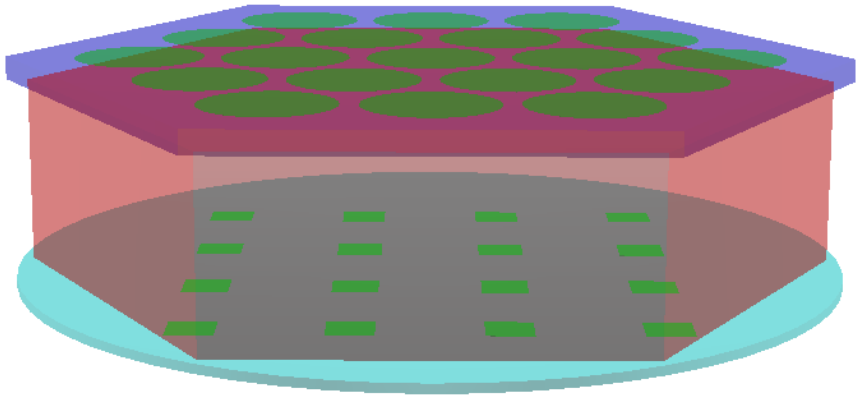
ANTS2 simulation: distribution of signals from a SiPM sensor

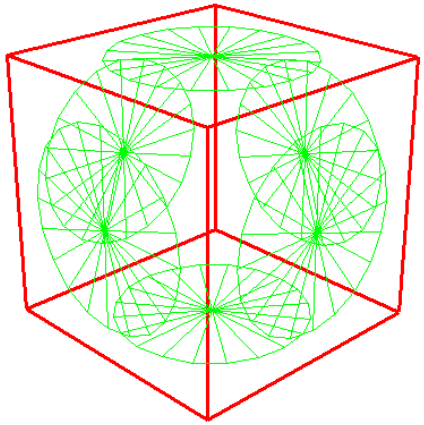
Arrays of sensors

Regular mode:
• All sensors are automatically positioned in square or hexagonal packing in the XY plane parallel to the *slabs*
• The Z position is automatically updated on all changes in the *slabs*

Custom mode:
• The XYZ position, orientation and sensor *model* are configured individually

The photon detection properties defined for a sensor *model* can be overridden for each individual sensor.





Simulation modes

ANTS2 offers two main simulation modes:

Optical photons only

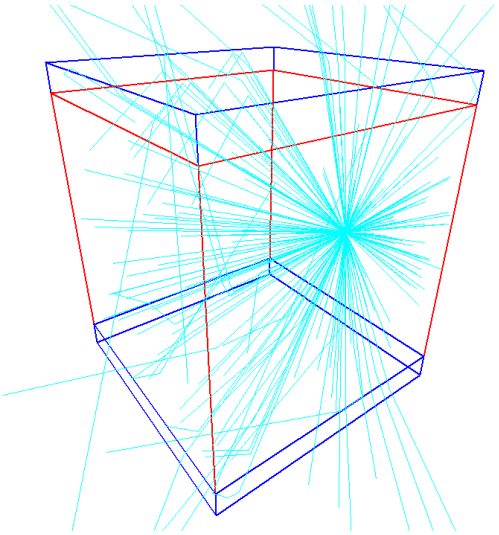
- To be used when it is sufficient to simulate each event as a "*photon bomb*" – isotropic emission of a photon from a given source.
- This mode does not require much configuration efforts – very well suited during detector prototyping phase.

Particles + optical photons

- Used to simulate interaction of particles with the detector and generate optical photons in primary/secondary scintillation, according to the properties of the interacting particle and the materials.
- These simulations produce more accurate results but require additional configuration efforts.

Photon source mode

Individual events are simulated starting from generation of photons from a point source (or along a line for secondary scintillation)



The number of photons is either fixed or distributed according to the user-defined Poisson, normal, gamma or custom distribution.

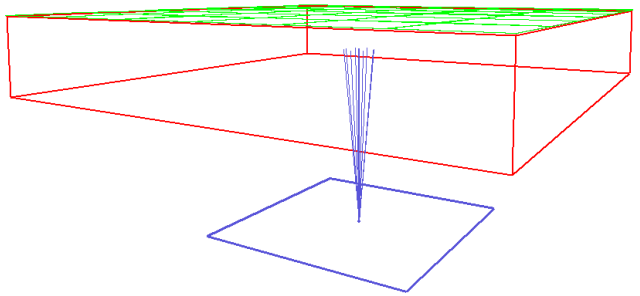
GUI offers a simple way to generate an event dataset, with the source positions being:

- a fixed position
- a regular array of positions (scan)
- random positions in a region with configured constrains (flood)

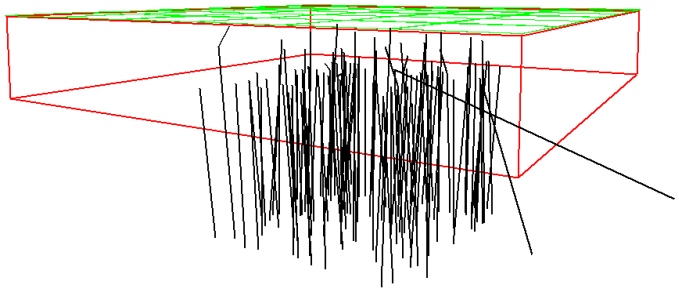
Particle source mode

Individual events are simulated starting from generation of particles from a *particle source(s)*.

A source emit particles of a given type with the initial energy which is fixed or sampled from a given distribution.



A source shape can be: a point, linear, square, round, box or cylinder.
Emission is uniform over the shape.



Emission from a source can be collimated: the particles are generated with initial direct

Alternative mode: particle generation from a file or a script-based generator

Interaction processes

ANTS2 can currently simulate the following interactions:

- **Gamma rays**
  - Photoelectric absorption
  - Compton scattering
  - Pair production
- **Positive ions**
  - Continuous energy deposition
- **Neutrons**:
  - Absorption (including capture + following decay)
  - Elastic scattering (including elastic coherent scattering from polycrystalline materials using NCrystal library)

All particles are considered to move in straight lines between the interactions.

Materials

Every volume of the detector geometry have to be associated with one of the user-defined materials. Materials have three groups of properties

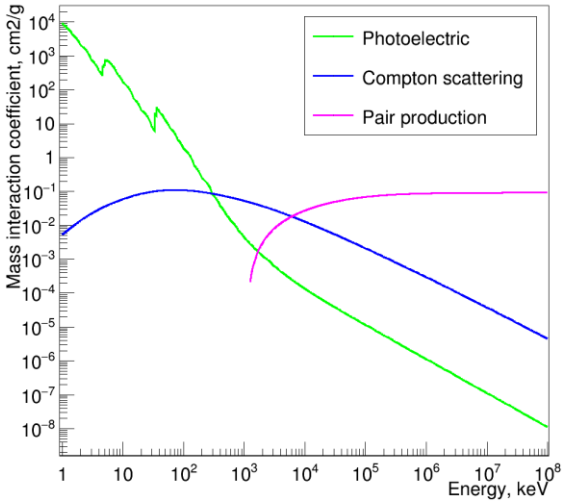
1. Particle interaction properties

- Configured for each material-particle pair, relevant for the planned simulations.
- For pairs with undefined interaction properties ANTS2 assumes that the material is transparent for the particle.

For new materials the interaction data have to be provided by the user!

ANTS2 have a set of tools to import data from

- XCOM database (gamma rays)
- TRIM/SRIM output (positive ions)
- IAEA database aggregator (neutrons)



Materials

2. Optical properties

- Refractive index (scalar and/or vs. wavelength)
- Bulk absorption (scalar and/or vs. wavelength)
- Rayleigh mean free path
- Reemission probability for photons absorbed by the material

3. Primary scintillation properties

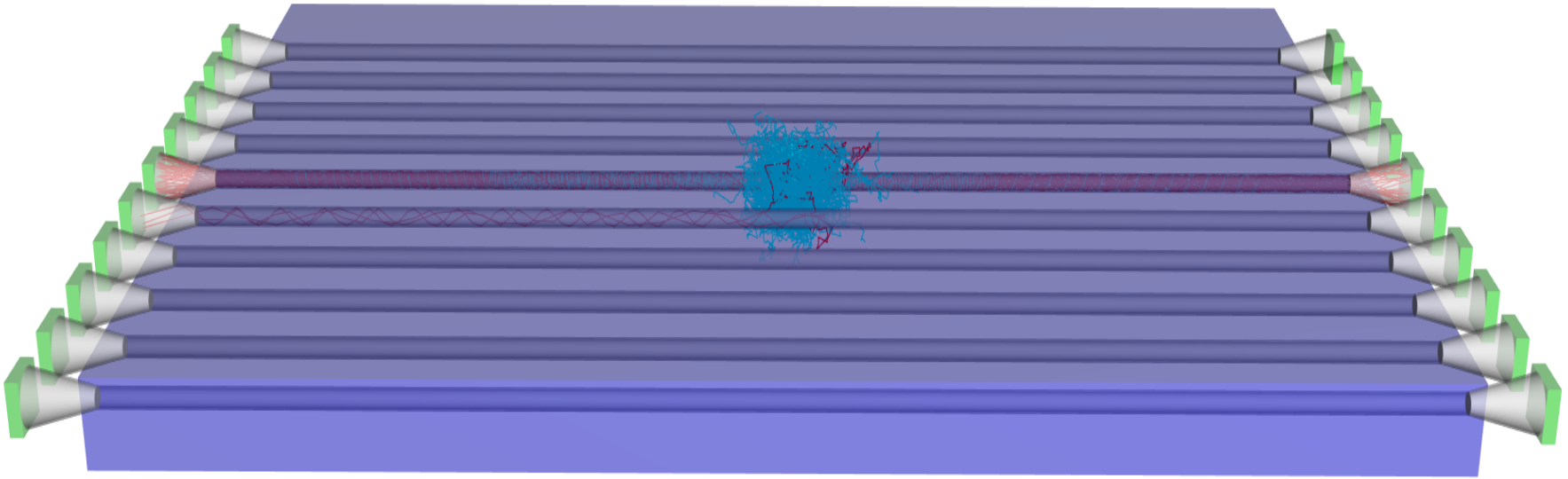
- Emission spectrum
- Emission decay time (several components in raise and decay is possible)
- Photon yield
- Intrinsic energy resolution

Photon tracking

3D navigation and calculation of the normal to the surfaces are performed with **TGeoManager** class of **CERN Root**.

Photon tracking implements the following processes:

- Reflection / transmission (Fresnel equations, Snell's law)
- Absorption (with possibility of reemission)
- Rayleigh scattering
- Custom rules for optical interfaces (next slide)

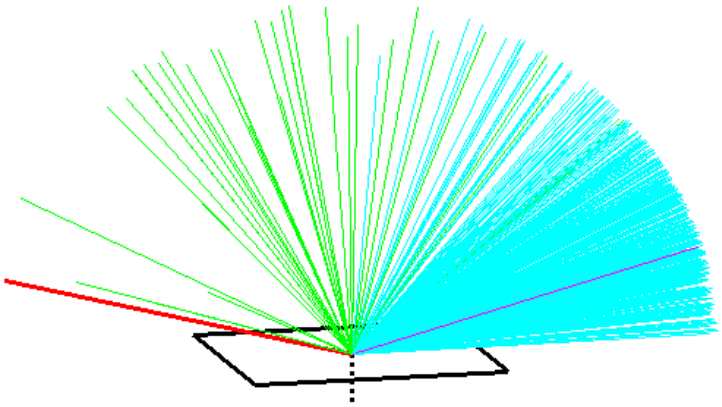


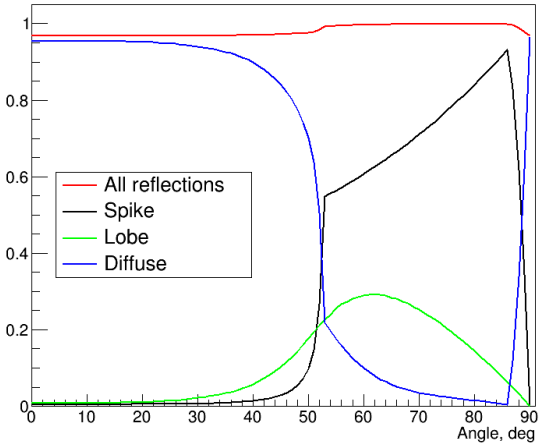
ANTS2 simulation: photon "tracks" in wavelength shifting fiber neutron detector

Custom rules for optical interfaces

In ANTS2 it is possible to define custom rules for behavior of optical photons at the borders of geometrical volumes.

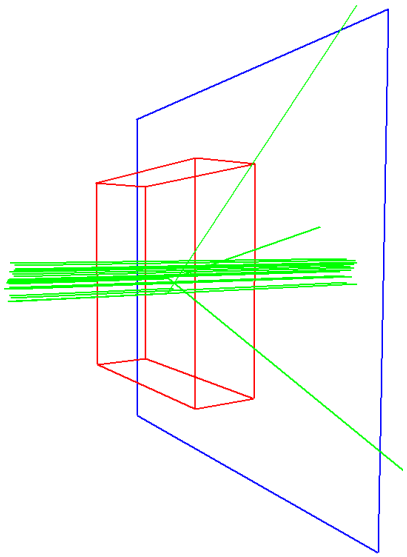
- The rules are defined for a pair of materials (*from* and *to*).
- In the simplest case, the rule includes the probability of absorption, specular reflection and Lambertian scattering.
- More advanced models describe light scattering on rough surfaces (e.g. the model of Claudio Silver from dark matter group), dielectric-metal interface and wavelength shifters.
- Optical interface can also be defined with a script.

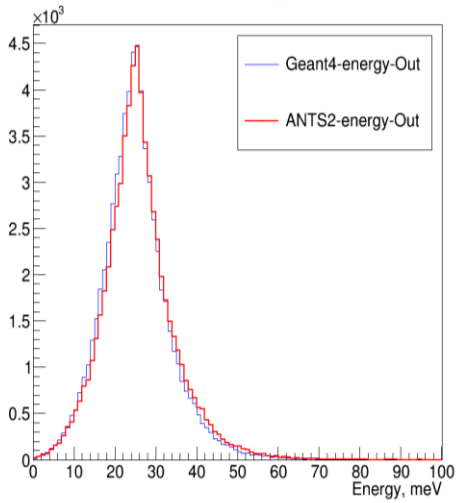




Simulation tools

- *Monitor* objects:
  Record statistics of the passing optical photons or particles.
  Give access to the distributions of time, position, angle of incidence, and wavelength (energy).





- "Photon" script unit:

  Detailed history of photon tracing can be collected and accessed for individual photons selected according to two configurable lists of processes: "must have" and "should not have".

- "Depo" script unit:

  A detailed history of tracking, energy deposition and termination can be accessed for each particle.

Semi-automatic detector optimization

Advanced optimization tools

A multi-parameter optimization is often required during detector development.

The brut-force approach:

Define a grid of parameter values and perform simulation + analysis for each grid node. Find the best node.

Can be very time consuming!

ANTS2 offers a more effective alternative:

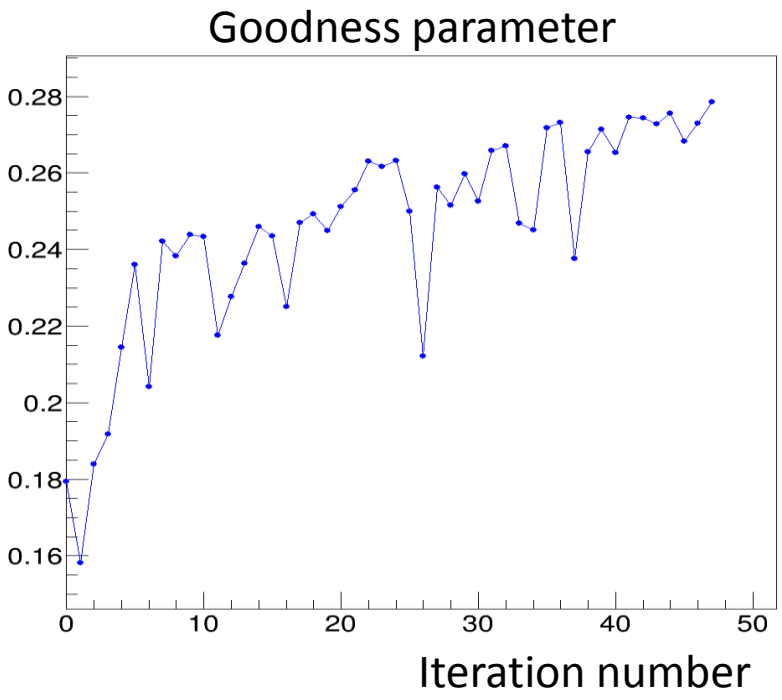
Configure the minimizer (running, e.g., Simplex algorithm) to use a minimization function defined in ASNTS2 script. On each call from the minimizer the function:

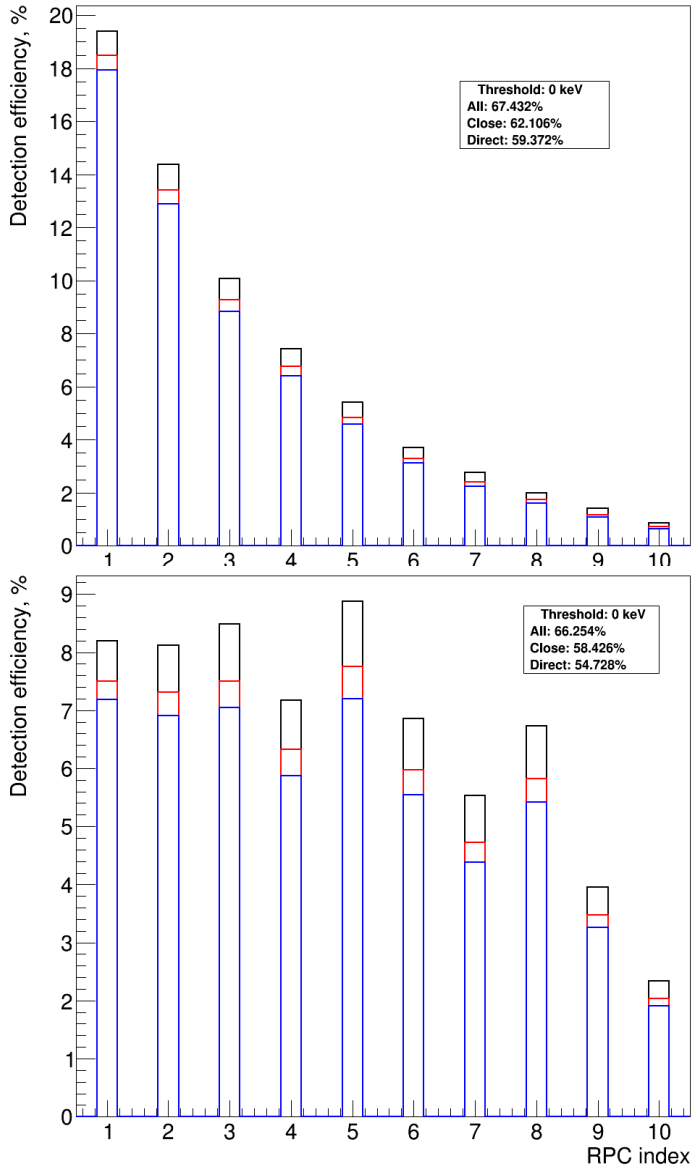
- modifies the detector geometry
- runs a simulation providing a set of events
- reconstructs the events
- performs analysis and returns the goodness value to the minimizer.

Semi-automatic detector optimization

Boron10-RPC neutron detector:

Optimize 5 neutron converter thicknesses to equalize as much as possible the count rate of 10 RPCs but keep the total detection efficiency as high as possible.





Experimental data import

Event data import

In ANTS2 event data can be loaded from text file(s):
every line of the event file has to list the signal values of all sensors.

The line can also contain information on the position of the event
(X, Y and Z coordinates) and its energy.

The imported data can be preprocessed in ANTS2:

- Pedestal subtraction
  (a tool is provided to calculate the pedestals)
- Scaling to equalize sensor gains or to convert signals to photoelectrons
  (a tool is provided to evaluate the relative gains and estimate the signal per single photoelectron)
- Suppression of events with sensor saturation

Event discrimination tools

Event discrimination can be performed based on:

- Signal values of individual sensors or sum signal of all sensors
- Reconstructed and/or loaded event energy
- Chi2 of the event reconstruction
- Reconstructed or loaded position of the event
- Average distance to the neighbors calculated with the kNN method

Scripting tools allow to perform custom cuts using multiple criteria

Implementation details

Implementation

ANTS2 is written in C++ using Qt framework
and requires installation of CERN ROOT.

ANTS2 runs on Linux (native Windows version is deprecated) but due to the docker support can run at Windows and Mac too.

Optional libraries:

EIGEN3:	for faster LRF fitting
CUDA toolkit:	for GPU-based statistical reconstruction
NCrystal:	for elastic neutron scattering simulation
FANN:	for ANN reconstruction
FLANN:	for kNN reconstruction and event rejection

Open source and more detailed information can be found at:

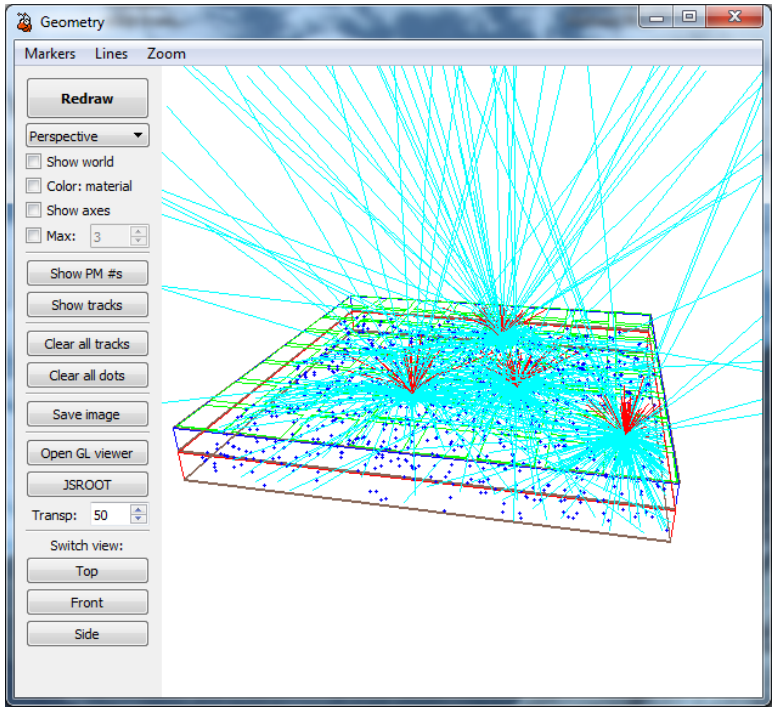
https://github.com/andrmor/ANTS2

http://coimbra.lip.pt/ants/ants2

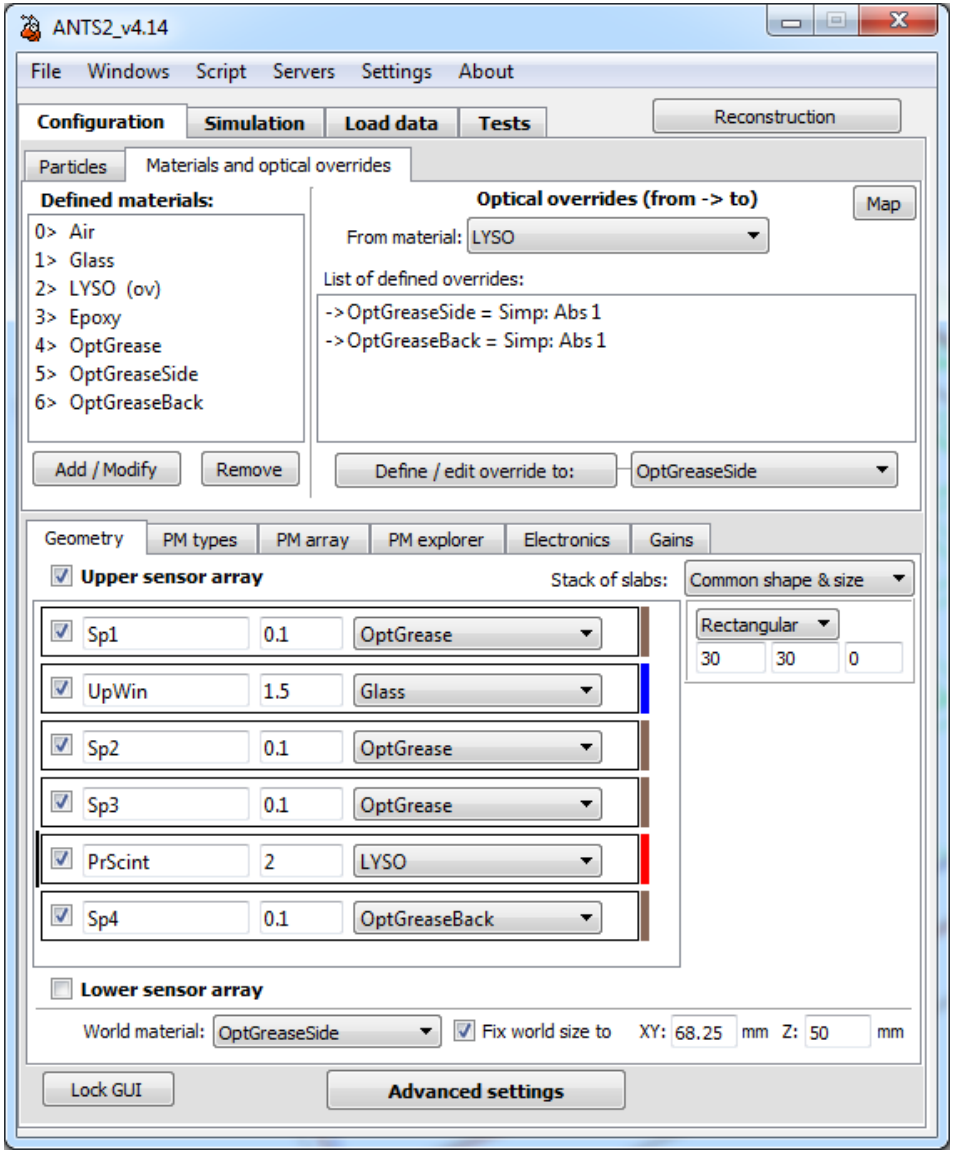
ANTS2 GUI

GUI was developed in **Qt** framework

Visualization of the detector geometry (plus tracks and markers) as well as the histograms and graphs with the results are performed with **CERN Root** package included as a library.



Geometry window



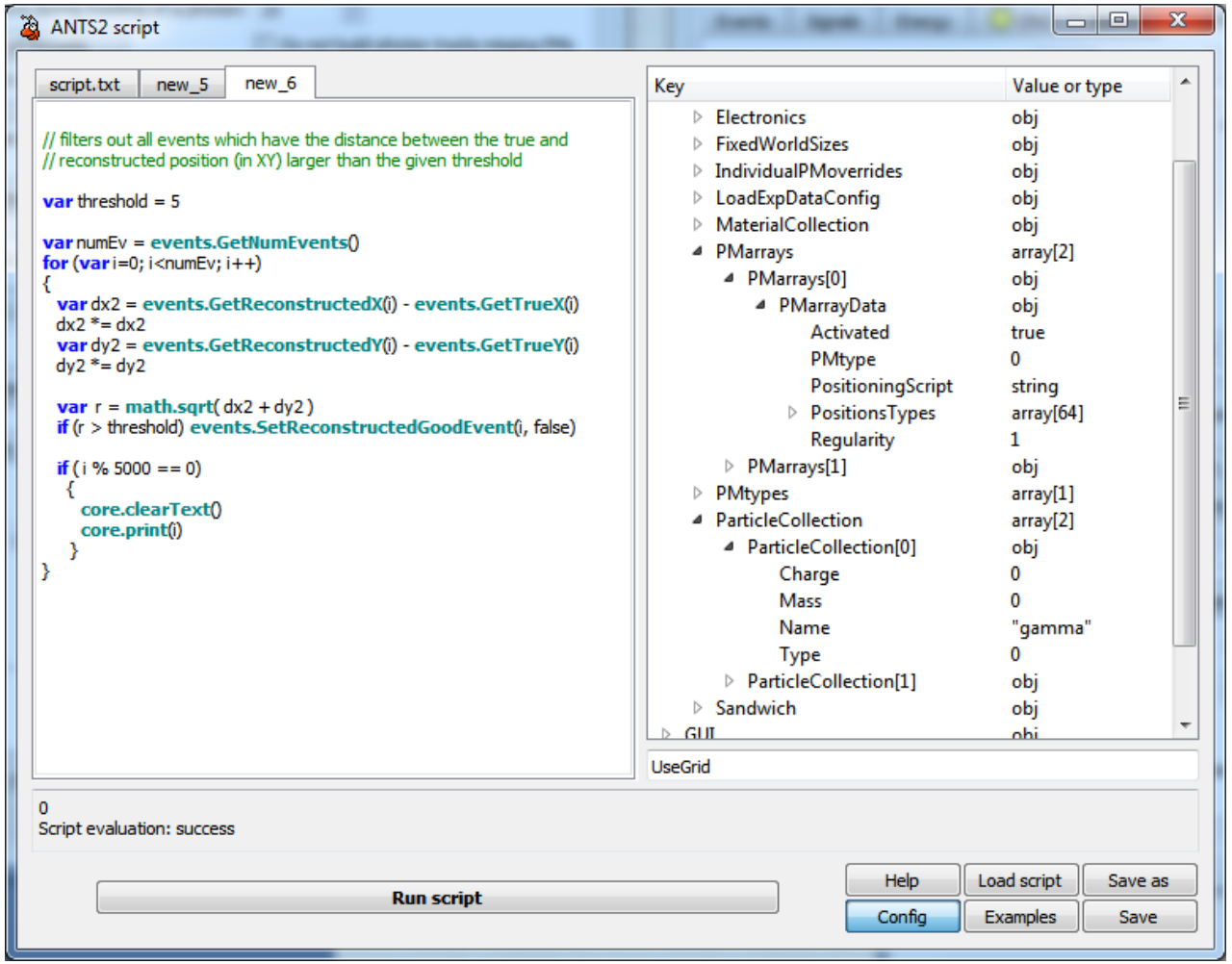
Main window, detector configuration tab

Scripting tools

Scripting system of ANTS2 uses JavaScript (QScript module of Qt) or Python (optional, has to be installed separately)

Scripts can be executed

- In GUI (script window)
- In batch mode (by providing the name of the file with the script to the ANTS2 executable)
- Through the WebSocket interface of the ANTS2 server



Script window of ANTS2

Distributed simulation/reconstruction

Simulations and reconstructions can be very time-consuming!

ANTS2 offers a simple way (literally one click in GUI) to distribute work over a grid of computers using WebSocket protocol and custom ANTS server dispatcher program.

Our colleagues at LZ collaboration has recently run ANTS2 in batch mode at a cluster of 1000 computers using docker installation.

Docker

Future?

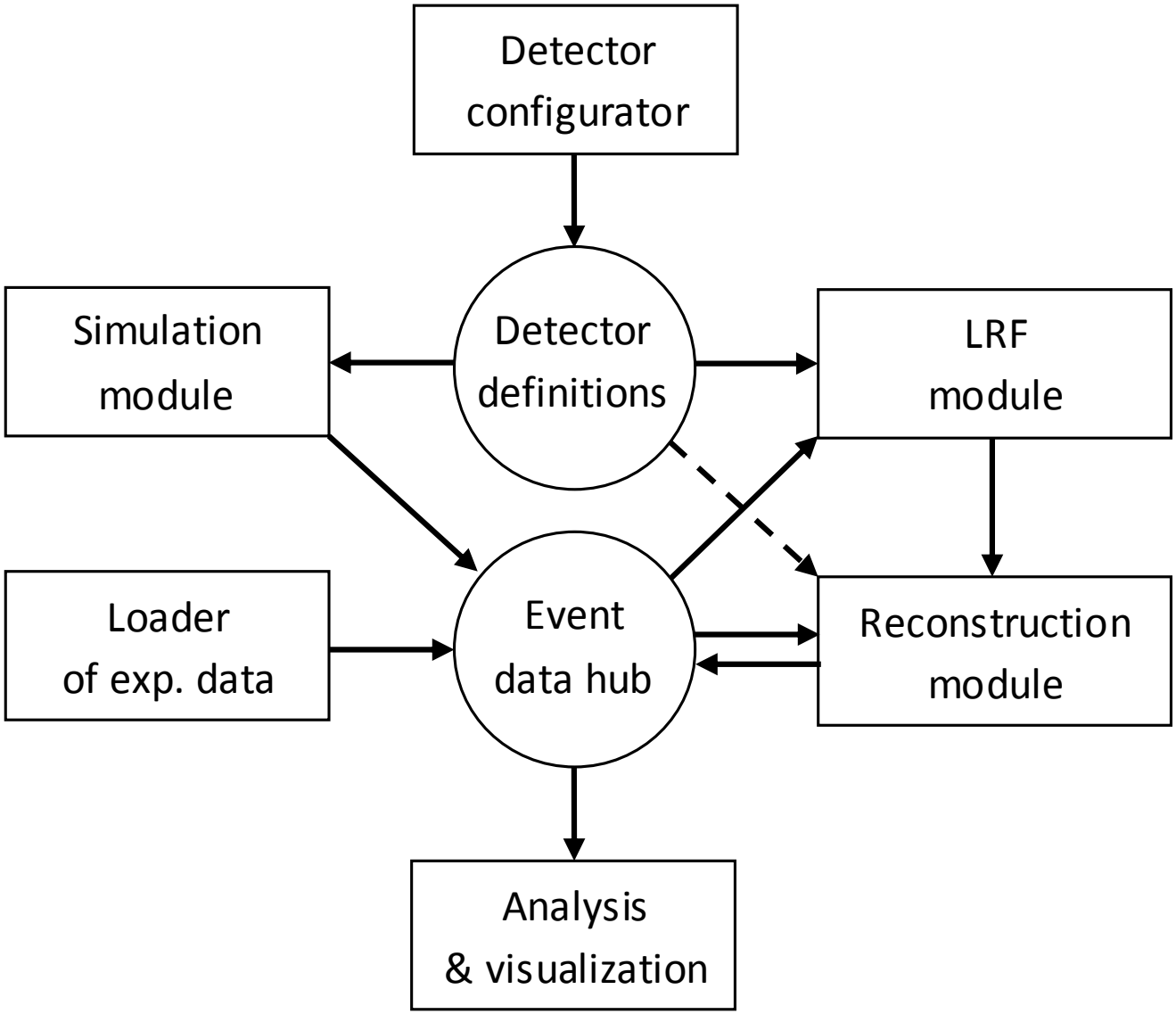
Near future:

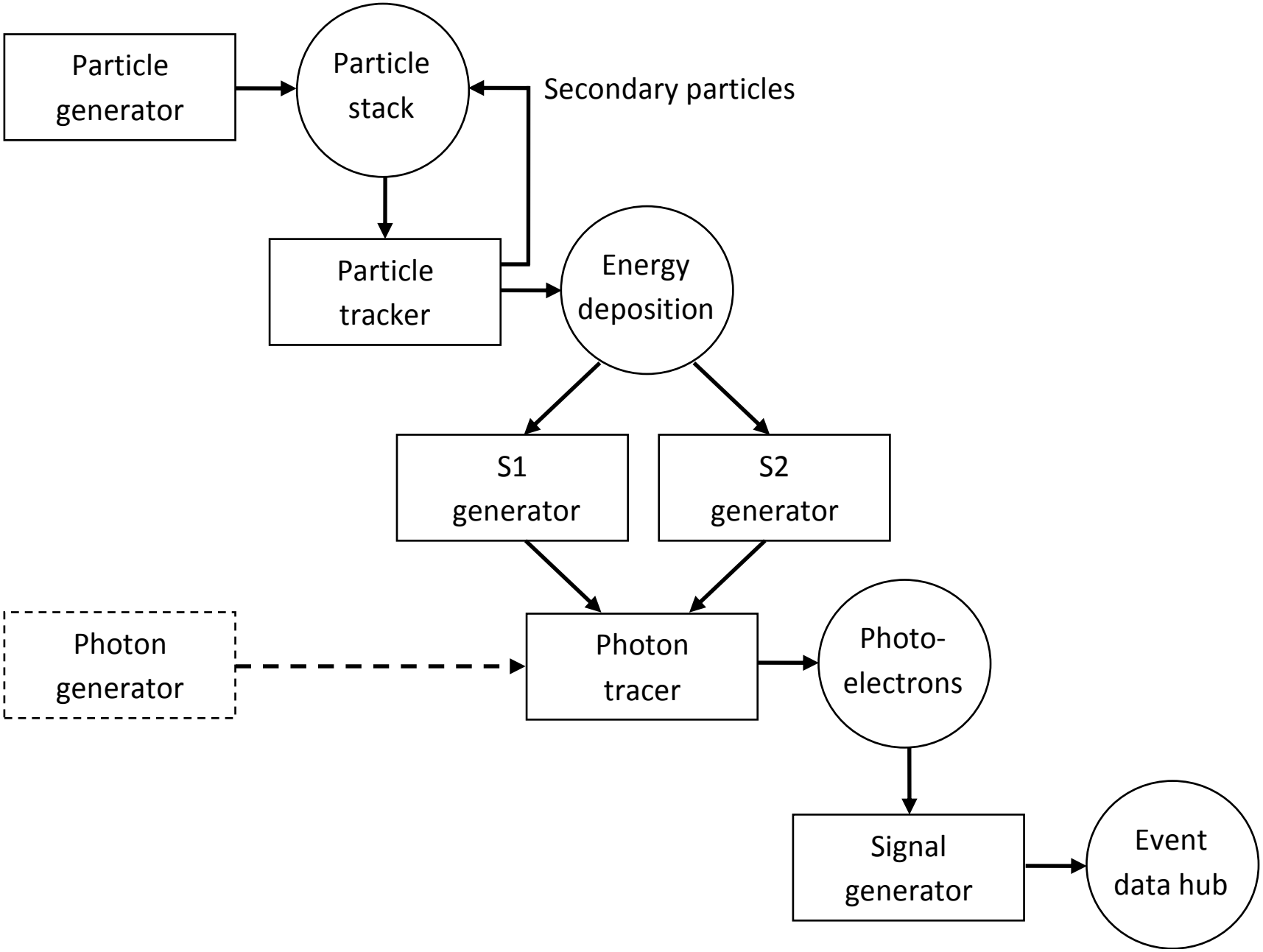
We need a campaign of writing / updating toolkit documentation

Large-scale projects:

- Replace TGeoManager with VecGeom (the navigator of GeantV)
  Faster: SIMD architecture, updated navigation algorithms
- ANTS2-Geant4 interface
  To be able to delegate tracking of particles to Geant4 and get back the energy deposition data
- Tracking of optical photons on GPUs
  An attempt to dramatically speed up photon tracing – need to code a custom tracker based on algorithms already available in VecGeom
- Reconstruction and discrimination tools for events with multiple photon sources
- Web interface for outreach and didactics

Backup slides


Detector configurator
Simulation module
Detector definitions
LRF module
Loader of exp. data
Event data hub
Reconstruction module
Analysis & visualization


Particle generator
Particle stack
Secondary particles
Particle tracker
Energy deposition
S1 generator
S2 generator
Photon generator
Photon tracer
Photo-electrons
Signal generator
Event data hub

Photon polarization is NOT considered!

n – refractive index

α – bulk absorption coefficient