Toolkit for simulation of Detector's Charging Up/Down in **MPGD**s

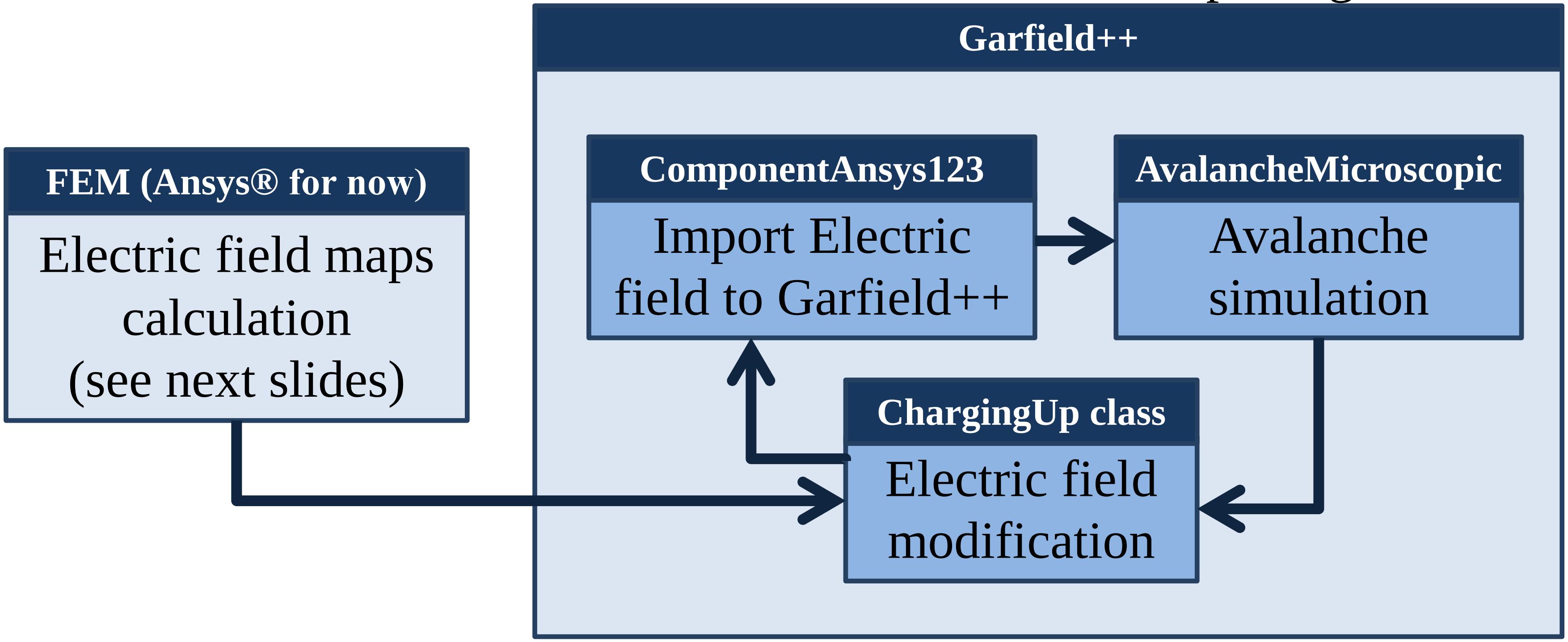
RD-51 Collaboration meeting

Michael Pitt[1], P.M.M. Correia[2], S. Bressler[1], L. Arazi[3], A.E.C Coimbra[1], D. Shaked-Renous[1], C.D.R Azevedo[2], J.F.C.A Veloso[2] and A. Breskin[1]

[1] Weizmann Institute of Science, Israel
[2] University of Aveiro, Portugal
[3] Ben-Gurion University of the Negev, Israel

Overview

- Initially developed by Rob Veenhof and Aveiro group (2014 JINST 9 P07025)
- The aim of the tool is to study the effect of charge accumulation on detector's insulating surfaces (2018 JINST 13 P01015)
- Available on https://github.com/pmcorreia/Garfpp-chargingup.git
- Based on the superposition principle (see next slides)
- Applicable for any MPGD geometry
- It's a c++ class interfaced with Garfield++ simulation package:



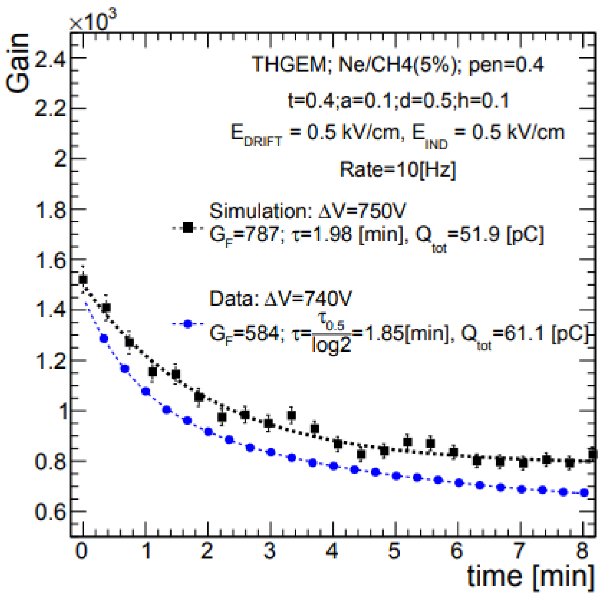
Overview - examples

- Gain variation due to accumulated charges can be simulated using Garfield++ interfaced with the toolkit.
- This allows studies of physics performance of detectors incorporating insulating surfaces

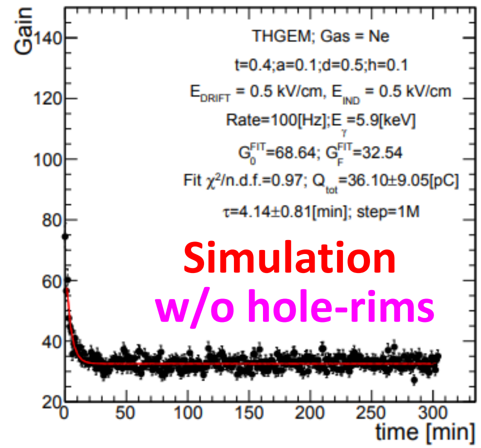
Overview - examples

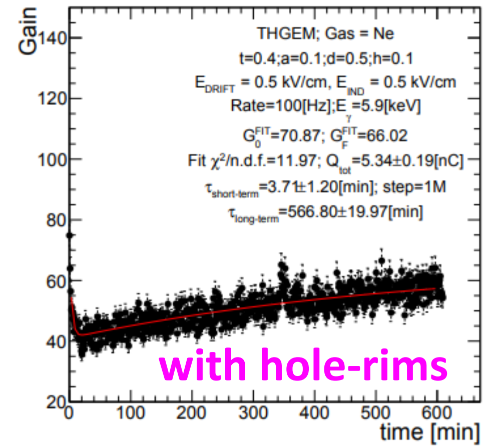
- Gain variation due to accumulated charges can be simulated using Garfield++ interfaced with the toolkit.
- This allows studies of physics performance of detectors incorporating insulating surfaces

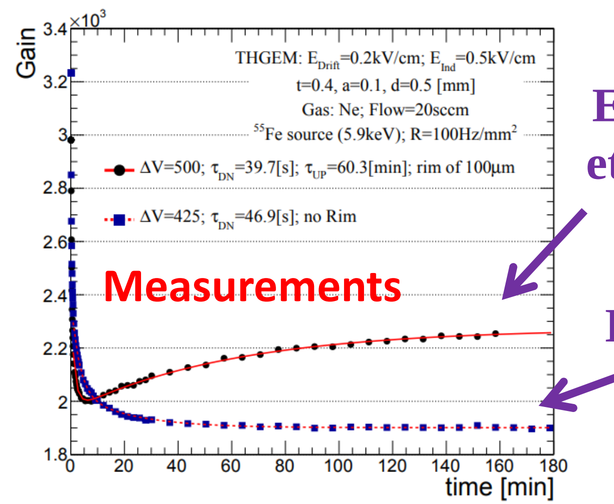
Example: Study of initial gain stabilization in THGEM with different geometry/conditions



MP et al., 2018 JINST 13 P03009







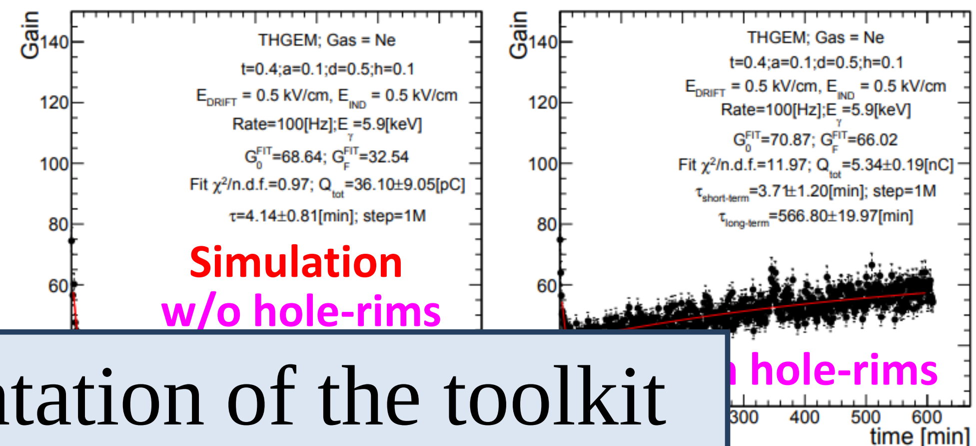
Electrode with an etched hole-rim of 0.1mm

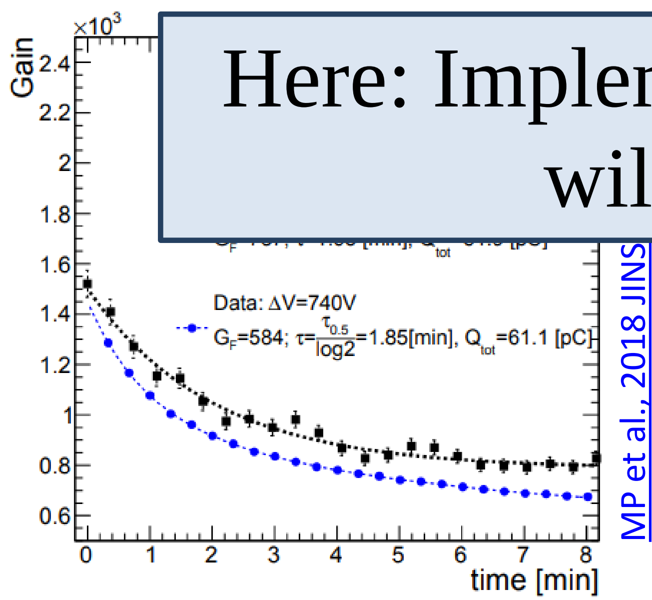
Electrode without hole-rim

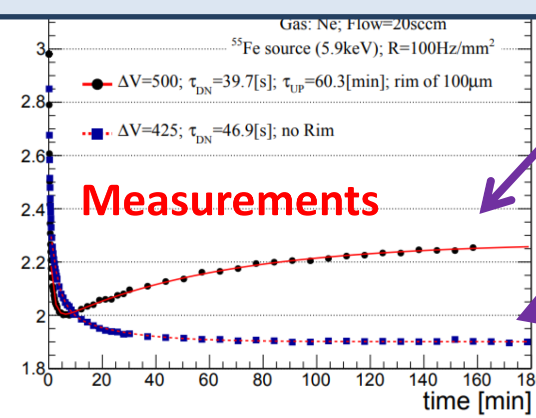
Overview

- Gain variation due to accumulated charges can be simulated using Garfield++ interfaced with the toolkit.
- This allows studies of physics performance of detectors incorporating insulating surfaces

Example: Study of initial gain stabilization in THGEM with different geometry/conditions







Here: Implementation of the toolkit will be reviewed

Superposition principle

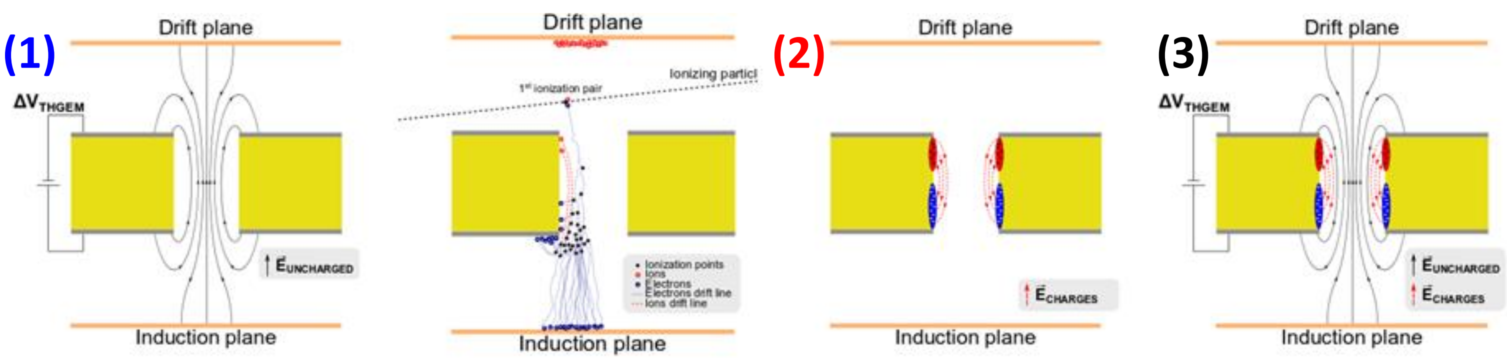
The simulation of charging-up rely on the superposition principle

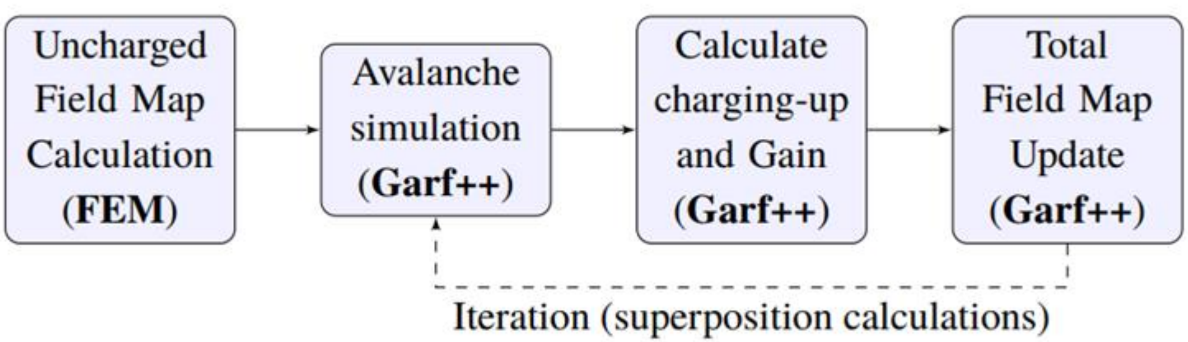
The tool is able to superpose field maps provided by the FEM program:

(1) Electric field calculation due to **applied voltages** on detector's electrodes

(2) Electric field calculation due to **electrical charges** on detector's surfaces

(3) Superposing **(1)** with **(2)** to obtain a new field map



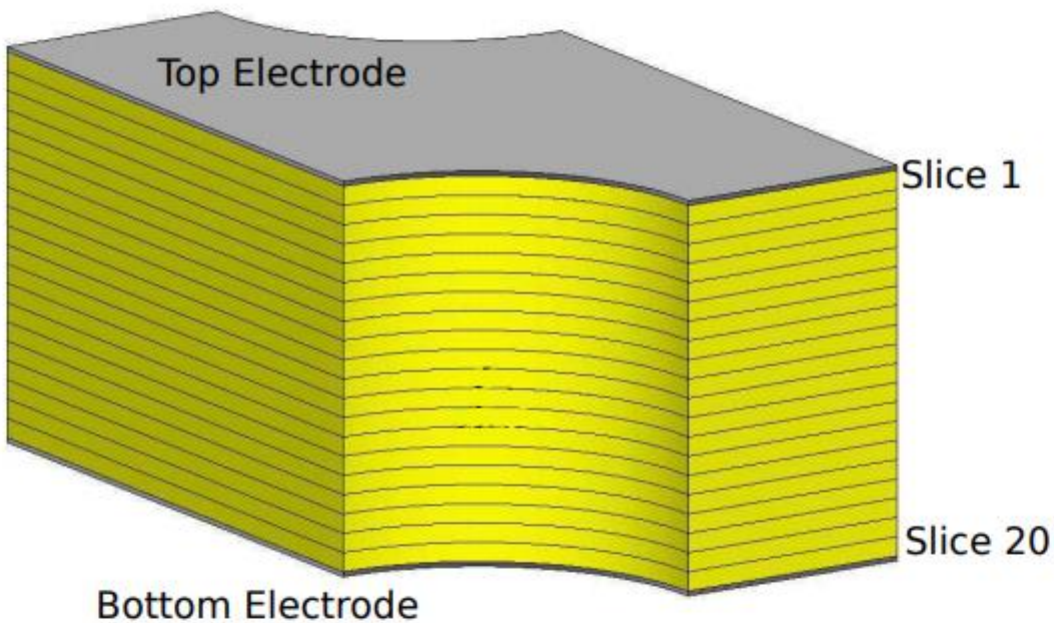


Superposition principle - example

- Design your detector geometry, calculate field maps when:
  1. **Voltage is applied on detector's electrode**
  2. **A slice is charged with a single unit charge**

Ansys®

Example: THGEM divided into 20 slices




```
PRNSOL_900V.lis
PRNSOL_slice10.lis
PRNSOL_slice11.lis
PRNSOL_slice12.lis
PRNSOL_slice13.lis
PRNSOL_slice14.lis
PRNSOL_slice15.lis
PRNSOL_slice16.lis
PRNSOL_slice17.lis
PRNSOL_slice18.lis
PRNSOL_slice19.lis
PRNSOL_slice1.lis
PRNSOL_slice20.lis
PRNSOL_slice2.lis
PRNSOL_slice3.lis
PRNSOL_slice4.lis
PRNSOL_slice5.lis
PRNSOL_slice6.lis
PRNSOL_slice7.lis
PRNSOL_slice8.lis
PRNSOL_slice9.lis
```

Implementation in Garfield++

"ChargingUpAnsys" class allows to manipulate with Ansys field maps:

Feed the class with the list of map files

Calculate the charges to be added to the slices

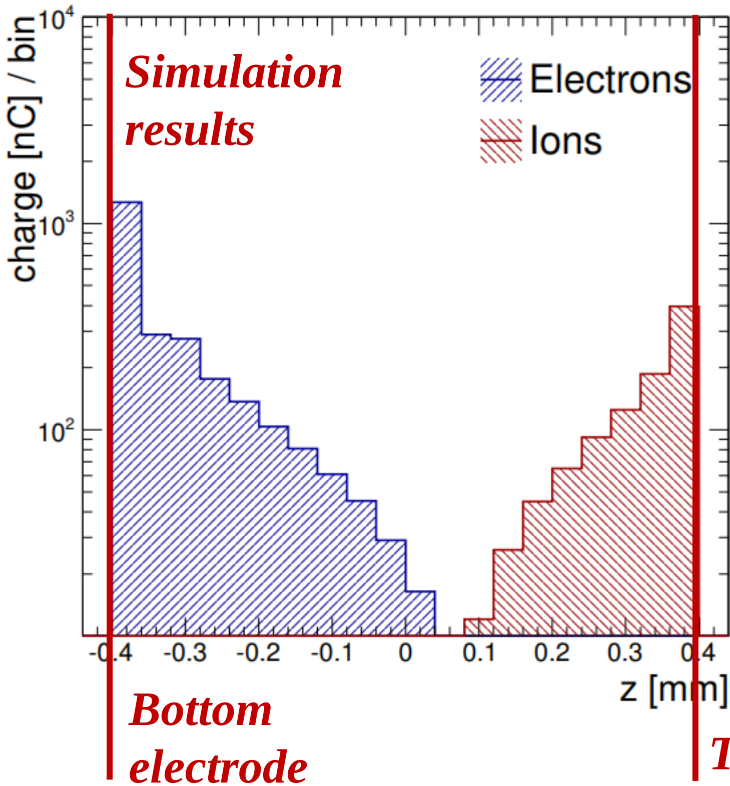
Update field map

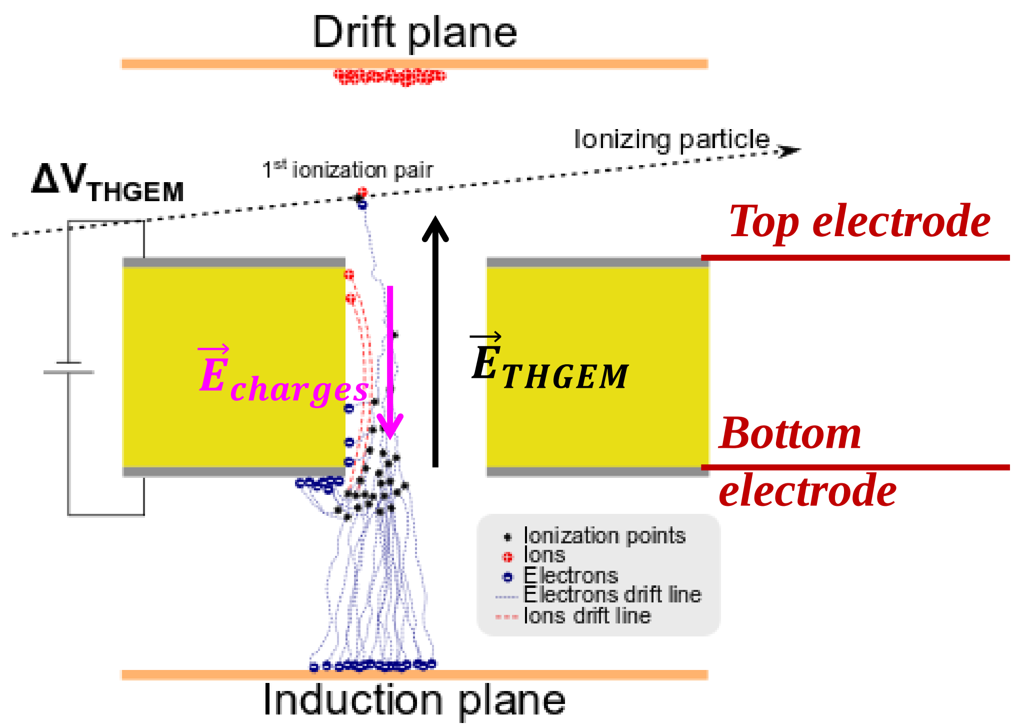
```
#include "ChargingUpAnsys.hh"
using namespace Garfield;

int main(int argc, char * argv[]){
  double ChargesVector[nSlices];

  //(...);

  ChargingUpAnsys file(mapfilesdir, nSlices, ChargesVector, gasstr, vgem, npe);
  if (!file.checkSlicesFieldMaps()){
        std::cout<<"Error 1, files don't exist"<<std::endl;
        exit(0);
  }
  file.loadSlicesFieldMaps();


  double simlatedCharges[ nSlices];

  //for loop over iterations
  // (...) after avalanches calculation and calculation of the
  // number of accumulated electrons and ions
  file.DownCharge(DownChargeLambda)
  file.UpdateFieldMap(simulatedCharges);
  file.SaveKaptonChargesFile(iter);
  file.printCurrentCharges();

  //end of the for loop
}
```

In a new iteration, modified field map will be used

Simulation setup: THGEM case

- In the GitHub entry an example for THGEM is provided
- In the example, n_p avalanches simulated, and charge that end-up on the insulating surfaces (Q^{up}) is stores in `double simlatedCharges[ nSlices];`
- The actual amount of charge = $Q^{up} \times step$ (see next slide)





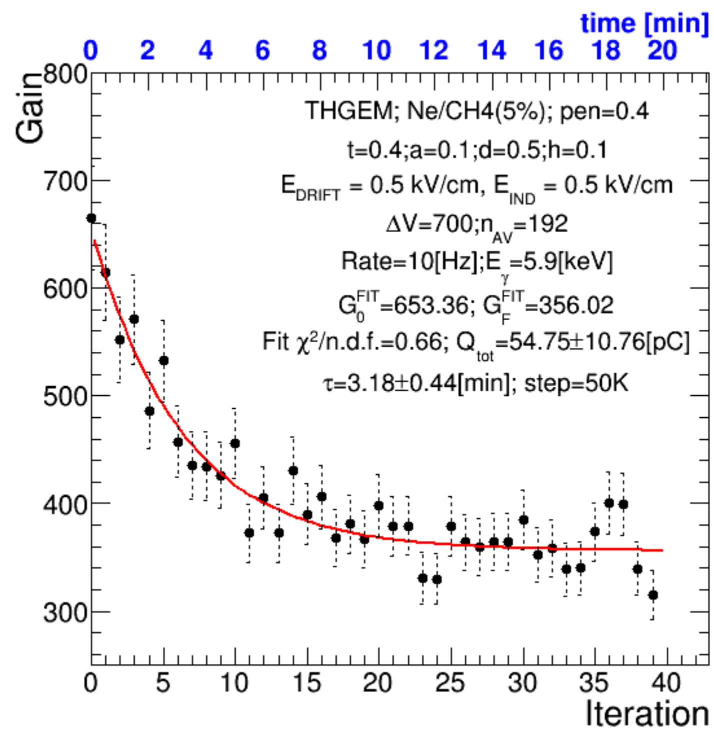
Simulation setup: THGEM case

- Since the total charge accumulated on the slices is small to significantly modify electric field, then one can:
  - Continue to iterating, adding charges (Millions of iteration)
  - Multiply charges by constant value to speed up the process
- Step size is a fixed parameter usually large for high voltages

Gain stabilization as a function of the number of the simulated iterations can be converted to actual time by:

$$\boldsymbol{t[sec]} = \frac{step}{n_p \times R[Hz]} \times n_{iter}$$

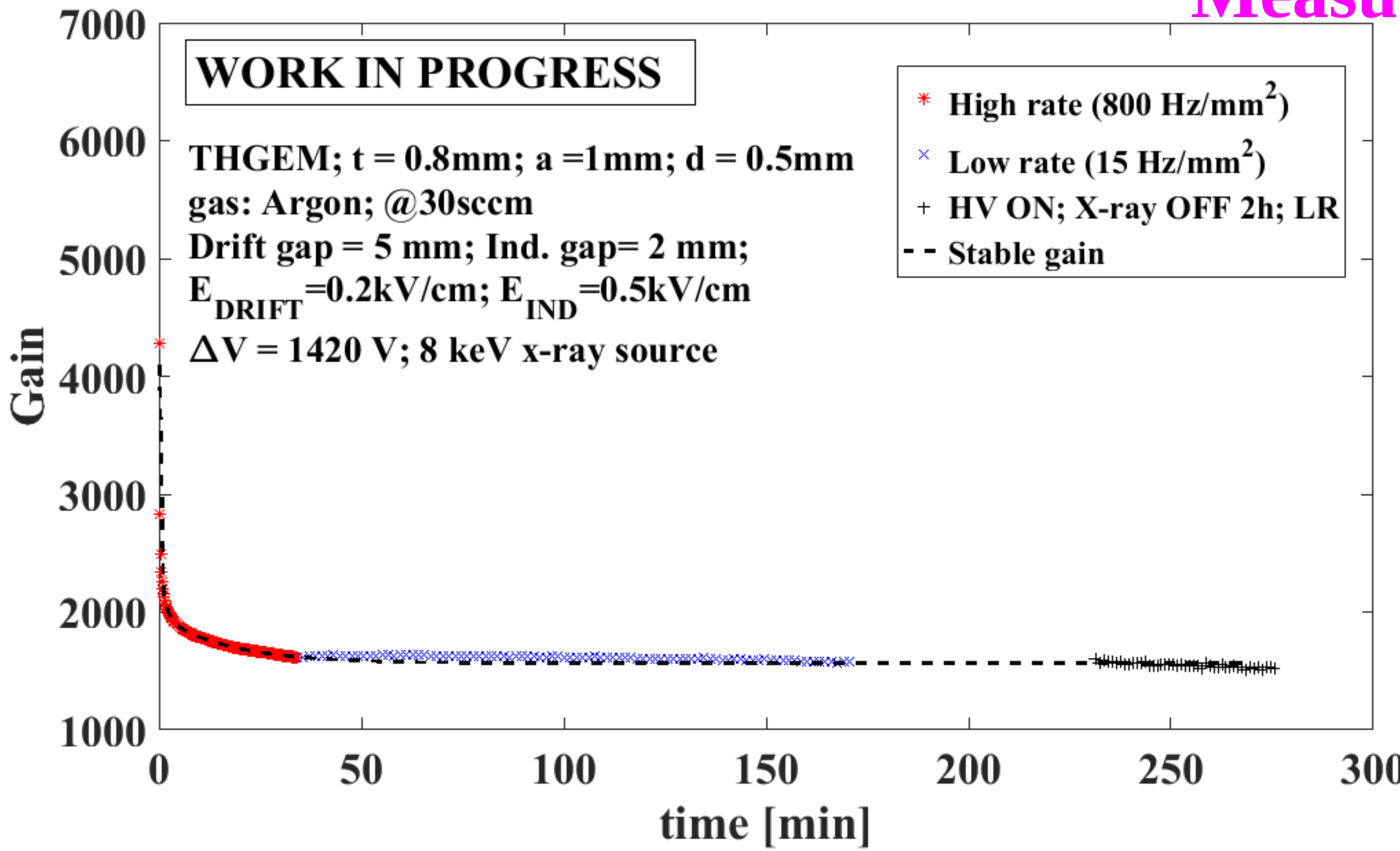
Gives the ability to compare to experimental results



Down charging – Motivation

- Measurement in pure noble gases showed that after a sable gain is achieved, further changes in irradiation rate are not affecting the detector's gain (furthermore, the stable gain value is rate independent – see slide 15)

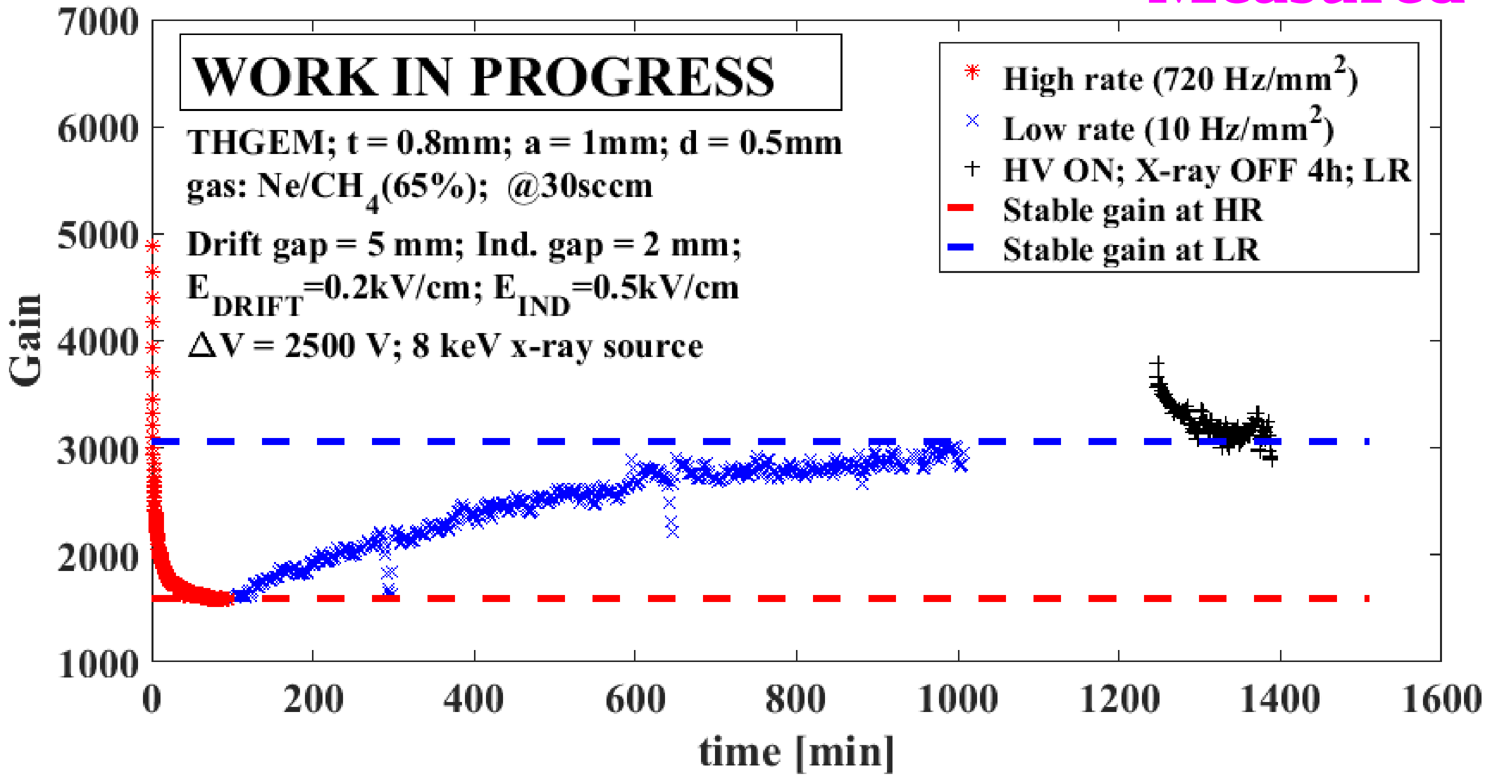
Measured gain



Down charging – Motivation

- With the same operational detector condition, **BUT** different gas mixture – gain stabilization is no longer rate independent.
- This might be attributed to charge evacuation via electro-negative gas molecules.

Measured gain



Charge evacuation model

- Introduce down-charging mechanism: $\Delta Q = Q^{up} - \boldsymbol{\lambda Q}$
- Charge evacuation rate is currently fixed by a user

```
//for loop over iterations
// (...) after avalanches calculation and calculation of the
// number of accumulated electrons and ions
file.DownCharge(DownChargeLambda)
file.UpdateFieldMap(simulatedCharges);
file.SaveKaptonChargesFile(iter);
file.printCurrentCharges();

//end of the for loop
```

- In the absence of charging up, $Q^{THGEM} = Q^{THGEM} \times e^{-\lambda \cdot n_{iter}}$
- Then one can extract the down-charging parameter using gain stabilization time τ_{DN}, by

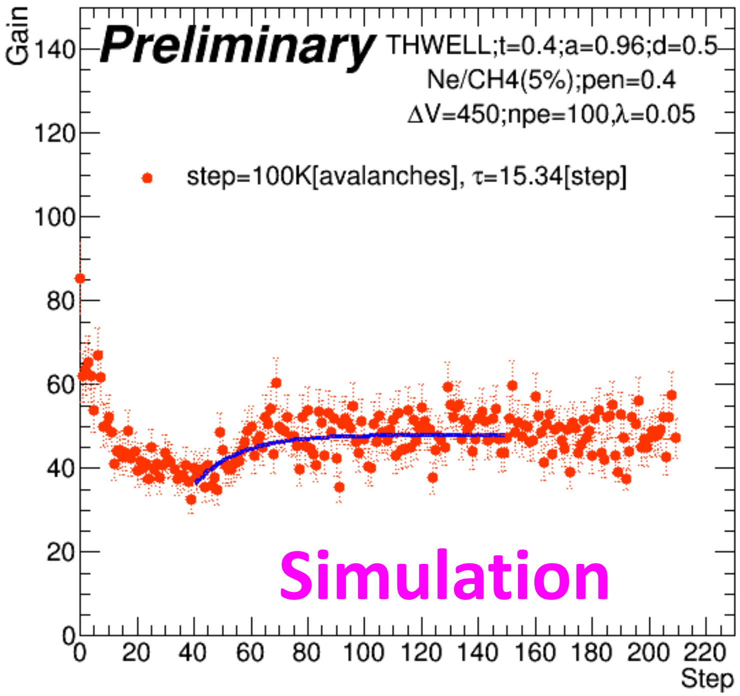
$$\lambda = \frac{1}{\tau_{DN}} \times \frac{step}{n_p \cdot R[Hz]} [\frac{1}{s}]$$

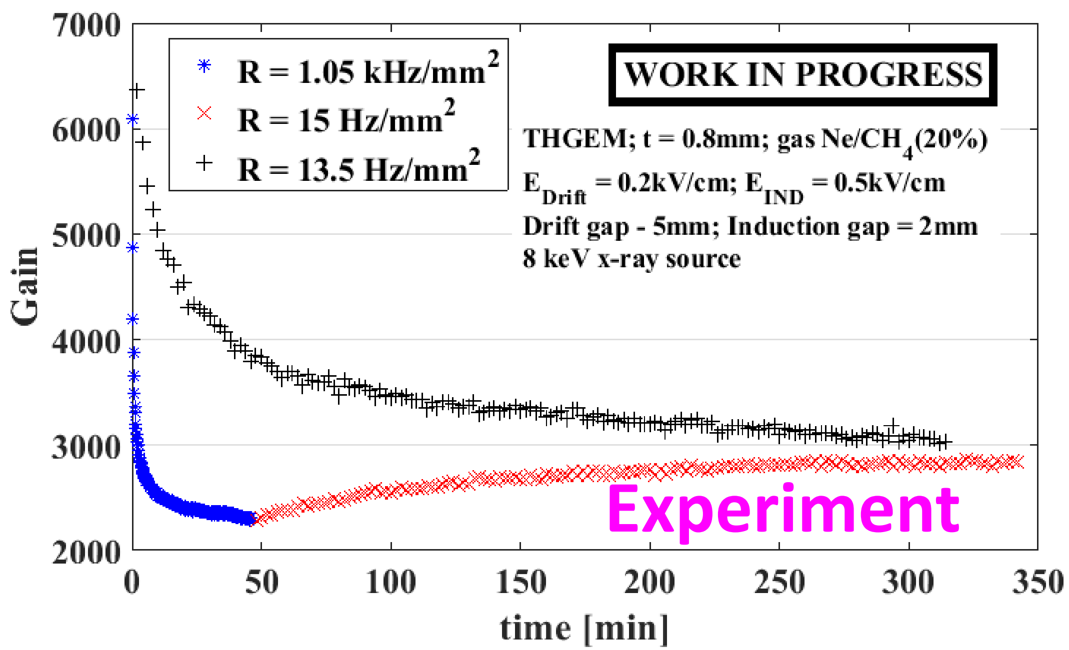
- Work is ongoing to determine evacuation rate for various gas mixtures

Charge evacuation model

- Introduce down-charging mechanism: $\Delta Q = Q^{up} - \lambda Q$
- Charge evacuation rate is currently fixed by a user

```
//for loop over iterations
// (...) after avalanches calculation and calculation of the
// number of accumulated electrons and ions
file.DownCharge(DownChargeLambda)
file.UpdateFieldMap(simulatedCharges);
file.SaveKaptonChargesFile(iter);
file.printCurrentCharges();

//end of the for loop
```





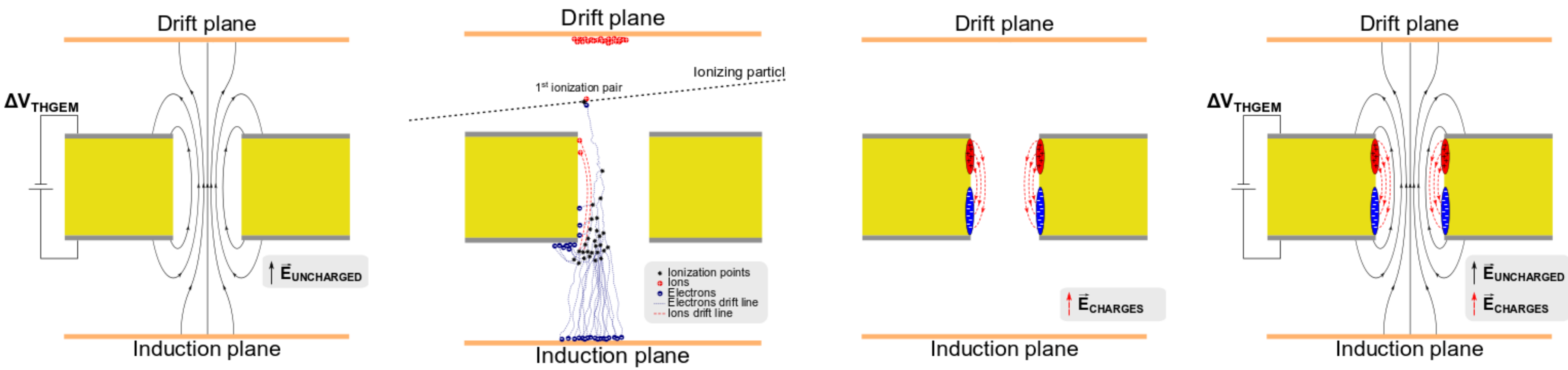
Summary

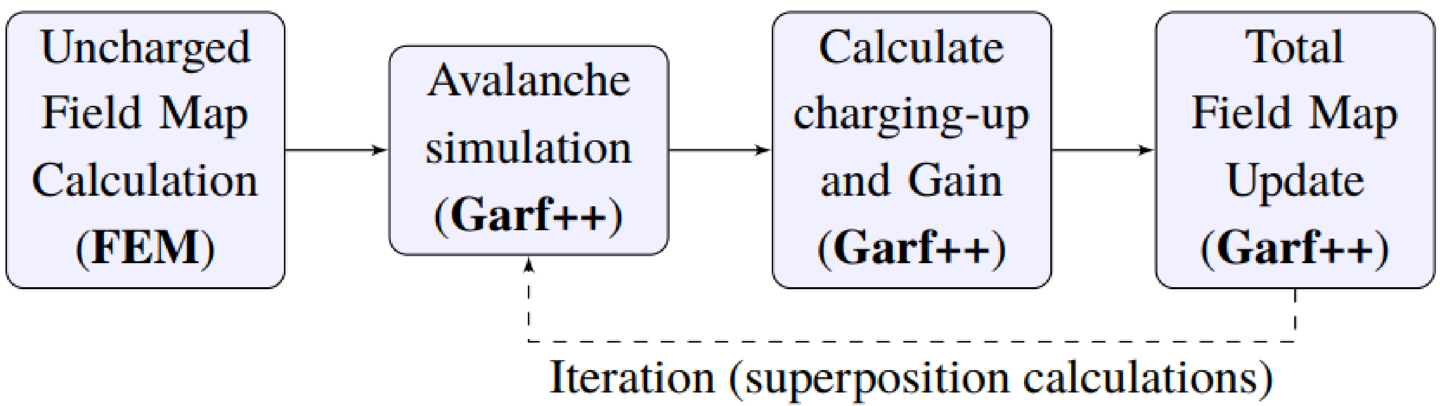
- The toolkit is available for the study of charging up/down in detector elements incorporating insulating materials
- Applicable to any MPGD geometry
- The Tool has been used in studies of GEM and THGEMs.
- Permits electric field variations within Garfield++ package.
- Down-charging is currently tested, up to now it is up to the user to fix the rate.

Back up

Principle of superposition

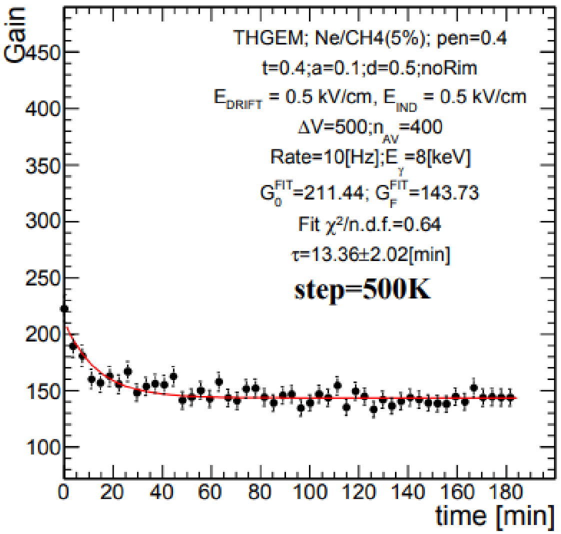
The simulation of charging-up rely on the superposition principle

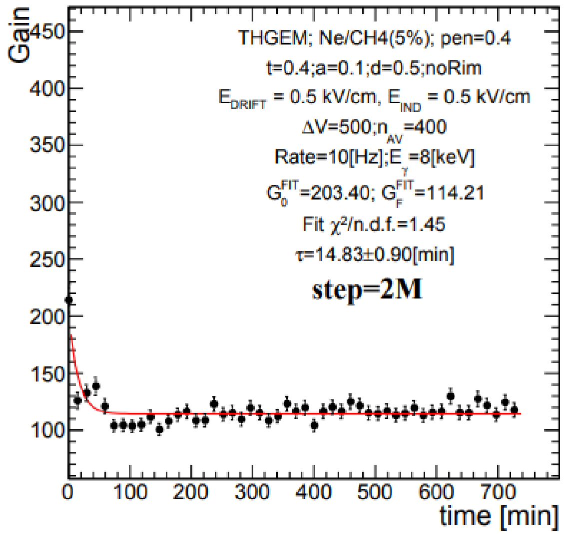


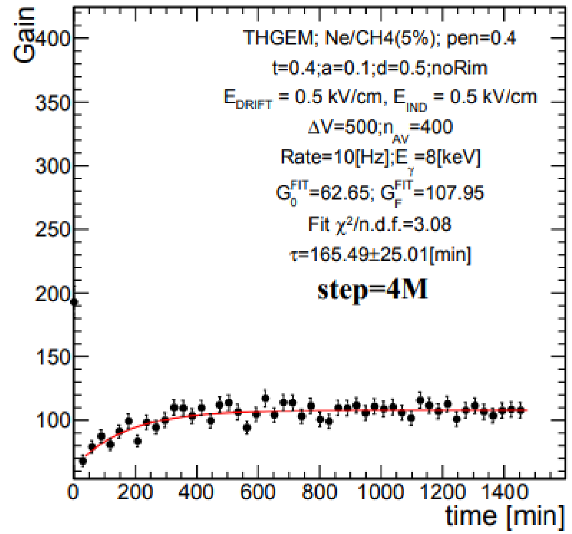


Simulation setup: Limitations

- Step size is a fixed parameter usually large for high voltages, one should scan different steps untill the correct range is found







Simulation setup: Limitations

- Step size is a fixed parameter usually large for high voltages, one should scan different steps untill the correct range is found