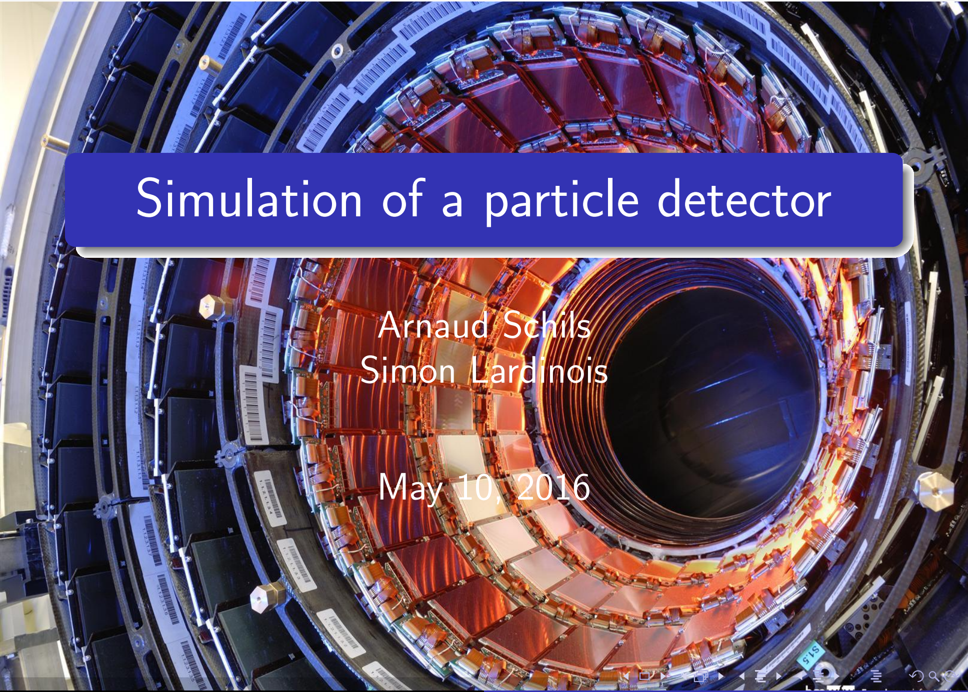


Simulation of a particle detector

Arnaud Schils
Simon Lardinois

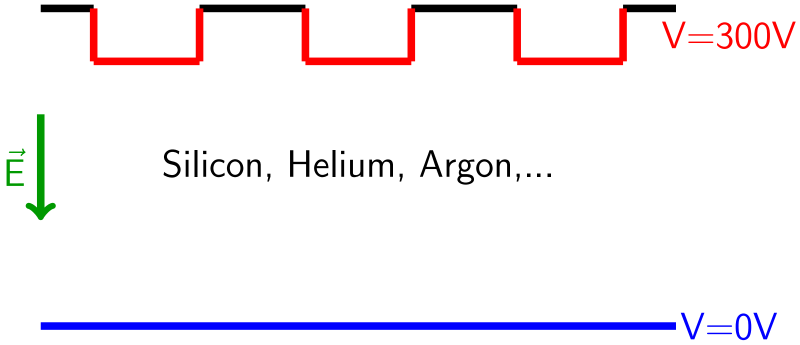
May 10, 2016

Summary

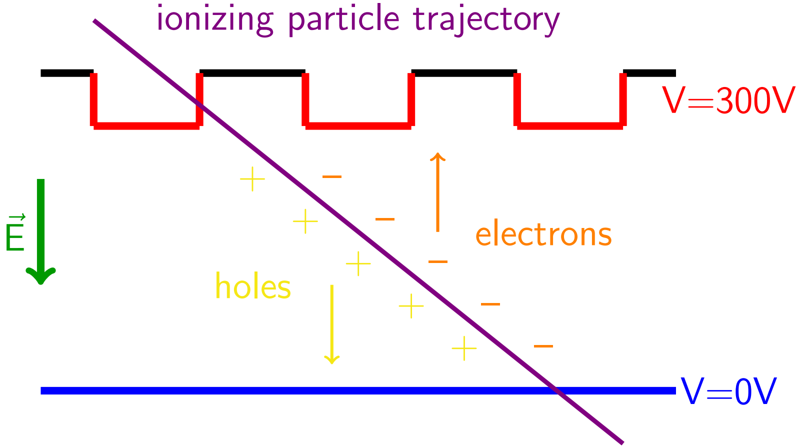
1. Problem statement
   - Particle detector
   - Media ionization
   - Drift velocity
   - Shockley Ramo theorem
   - Townsend avalanche
2. The software *pdetect*
   - Features and technical characteristics
   - Geometries and boundary conditions
   - Computing the potential in the detector
   - Computing the current induced by a particle
3. Conclusion

Problem statement

The detector



Problem statement
Media ionization



Problem statement
Drift velocity

Charges drift with velocity

$$|\vec{v}| = \mu|\vec{E}|$$

where μ is the mobility

- Electrons mobility is much higher than hole mobility
- mobility decreases when fields of 104 Vcm^{-1} and higher are applied due to saturation

Instantaneous current generated by one moving charge q at speed $\vec{v}$

$$i_{e,h} = -q\vec{v} \cdot \vec{E}_w$$

Weighting field $\vec{E}_w = \vec{E}$ computed when applying

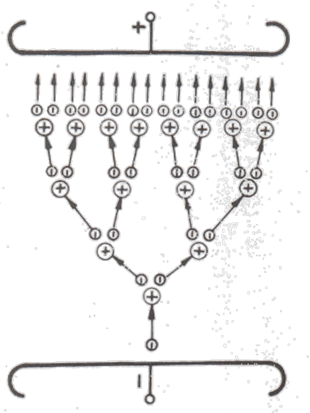
- $1V$ to the measurement electrode
- $0V$ to the other electrodes

Instantaneous current induced by the particle

$$i_{tot} = \sum_{holes} i_h + \sum_{electrons} i_e$$

Problem statement
Townsend avalanche

- e^- extract e^- from media molecules
- In gas detectors when $|\vec{E}| > 10^6\mathrm{Vm}^{-1}$



Problem statement

Summary: what do we need to compute the current?

$$i = -q\vec{v} \cdot \vec{E}_w \qquad \vec{v} = \frac{q}{|q|}\mu \vec{E}$$

$$\implies i = -|q|\mu \vec{E} \cdot \vec{E}_w$$

$$\vec{E} = -\nabla V \qquad \vec{E}_w = -\nabla V_w$$

$\implies$ to compute i you need V and V_w
$\implies$ Solve $\nabla^2 V = 0$ and $\nabla^2 V_w = 0$ for corresponding boundary conditions

Pdetect software

Features

- Simulates both silicon and gas detectors
- Computes $\vec{E}$ and $\vec{E}_w$ for three different 2D geometries
- Computes i_{tot} for any particle trajectory
  - Simulates Townsend avalanche and mobility saturation
- Outputs:
  - plots of $V(x,y), V_w(x,y)$, $\vec{E}(x,y)$, $\vec{E}_w(x,y)$
  - points $(t, i_{tot}(t))$ in a text file

Pdetect software

Technical characteristics

Developed in C++ (5209 lines of code)

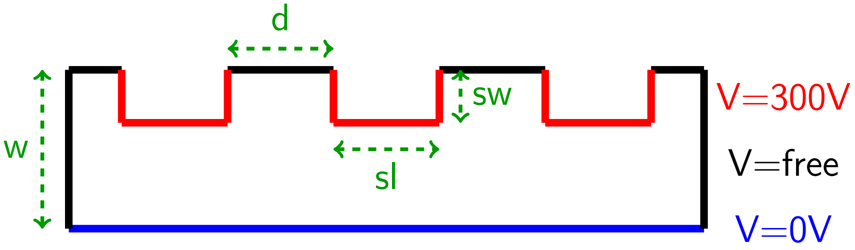
Object oriented programming (inheritance, generic programming, composition,...) advantages:

- avoids code duplication
- add features to a module with few modifications to other modules
- improve code readability

Fast: multithreading, adaptive grid refinement,...

Pdetect software

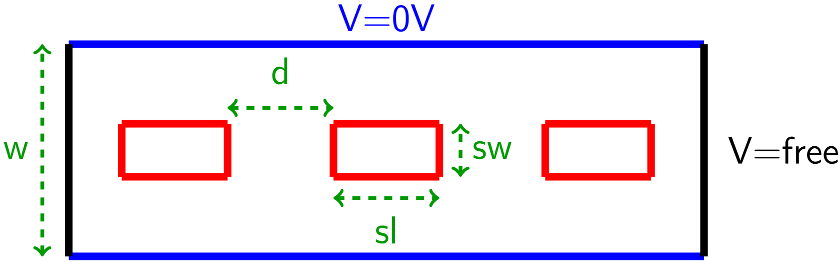
Detector geometry and boundary conditions (I)



- number of strips
- strip length sl
- strip width sw
- strip potential V
- detector width w
- inter-strips distance d

Pdetect software

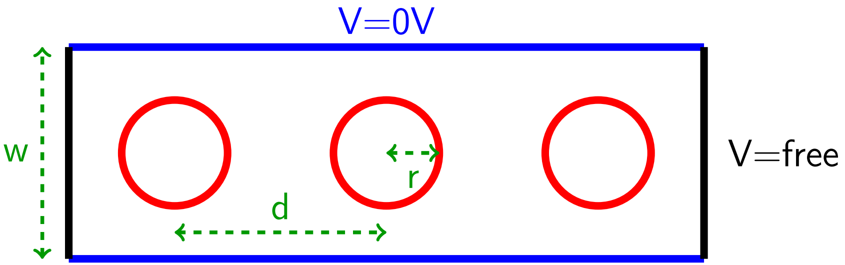
Detector geometry and boundary conditions (II)



- number of strips
- strip length sl
- strip width sw
- strip potential V
- detector width w
- inter-strips distance d

Pdetect software

Detector geometry and boundary conditions (III)



- number of strips
- strip radius r
- detector width w
- strip potential V
- inter strips centers distance d

Finite element method solves $\nabla^2 V = 0$ numerically

- function domain is divided in subdomains called finite elements
- piecewise interpolation: one polynomial fits the restriction of the searched function on one finite element
- build a linear system with the polynomials coefficients
- solves the system thanks to constraints:
    - satisfy $\nabla^2 V = 0$
    - match imposed domain boundary conditions
    - boundary conditions between polynomials defined on adjacent finite elements

Use deal.ii to solve the Laplace eq. (fast thanks to multithreading)

Inputs:

1. Detector geometry
2. Boundary conditions
3. Max relative error on V

Outputs for each FE:

1. a funct approximating V
2. a bound on the error on V
3. a funct approximating ∇V

Perform this process two times

- get $\vec{E} = -\nabla V$
- get $\vec{E}_w = -\nabla V_w$ with other boundary conditions

Electric field constant in most part of the detector

- quick computation with large FE sufficient

But Electric field varies strongly close to the strips

- much smaller FE required to get precise results

Uniform coarse grid composed of large FE
$\Longrightarrow$ results not precise enough close to the strips

Uniform dense grid composed of small FE
$\Longrightarrow$ waste of time

Iterative process

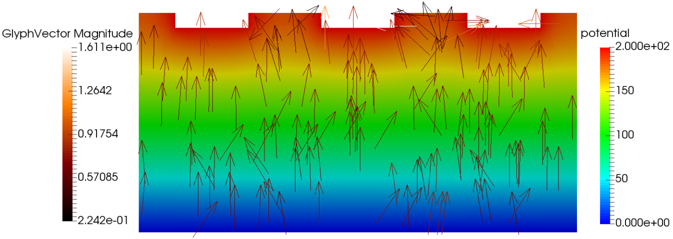
1. Starts with the coarsest grid
2. Compute V and error at each FE
3. If max error $>$ max tolerated error
    1. refine only cells of the grid with highest errors
    2. return to step 2

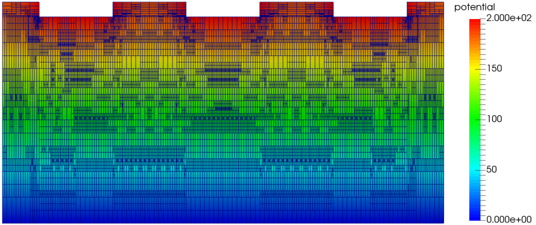
Final grid is composed of small cells only where the Electric field varies strongly

Pdetect software

Serrated rectangle: potential and electric field

Strip potential = 200V, detector width = 300μm, strip length = 100μm, strip width = 20μm, inter-strips distance = 100μm, max relative error = 0.9%

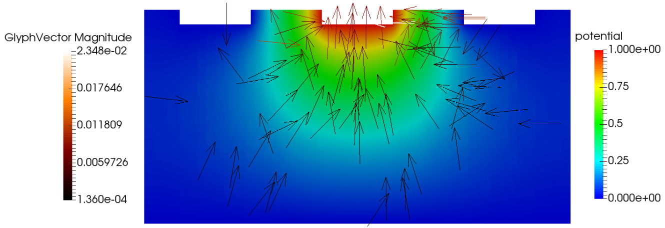


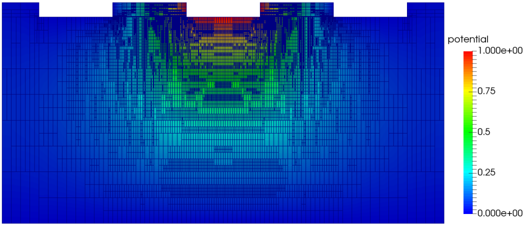


Pdetect software

Serrated rectangle: weighting potential and weighting electric field

Strip potential = 200V, detector width = 300μm, strip length = 100μm, strip width = 20μm, inter-strips distance = 100μm, max relative error = 0.9%

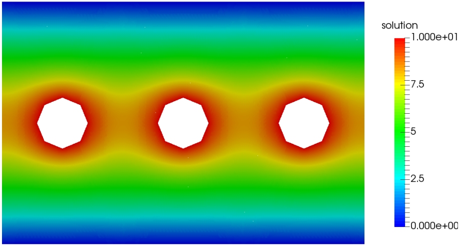


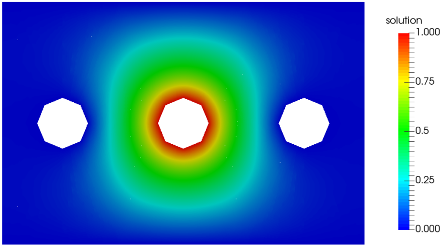


Pdetect software

Circular strips: potential and weighting potential

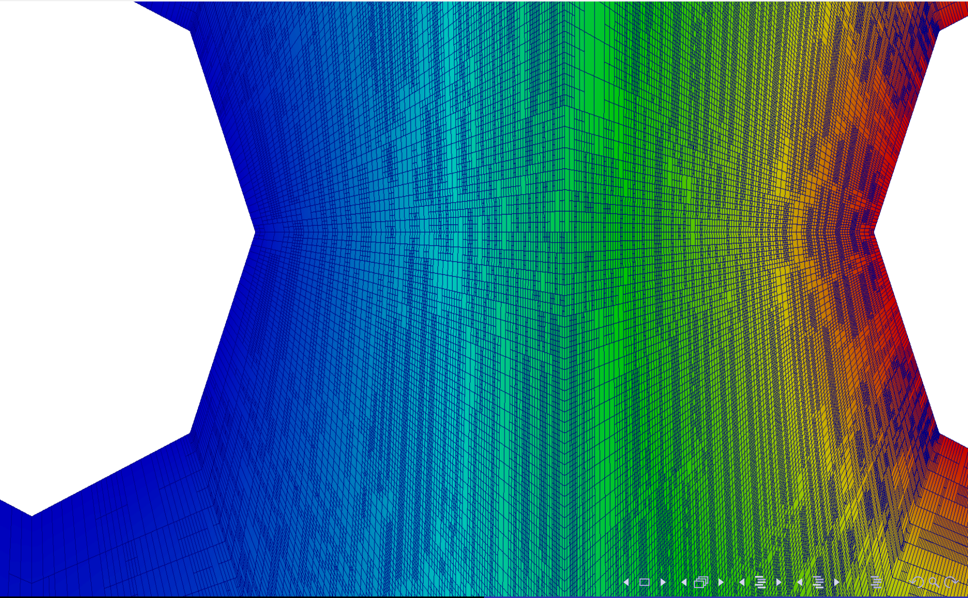
Detector width $= 200\mu$m, number of strips $= 3$, radius $= 21\mu$m, inter strips distance $= 50\mu$m, potential $= 10$V, max relative error 1%





Pdetect software

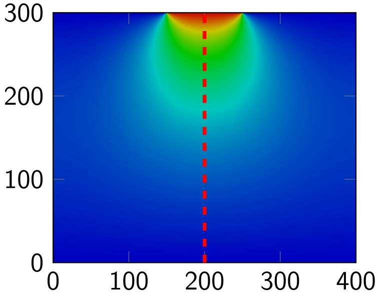
Circular strips: example of adaptive grid

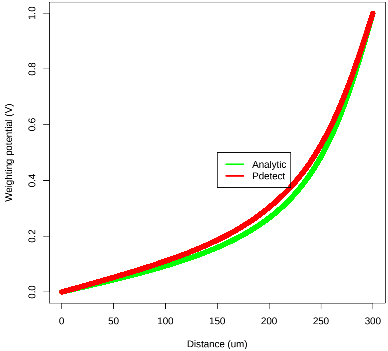


Pdetect software

Potential: numerical vs analytical solution

Potential along a vertical line (middle of the detector)
Pitch $= 400\mu$m, strip length $= 100\mu$m, width $= 300\mu$m, $V_{strip} = 1$V



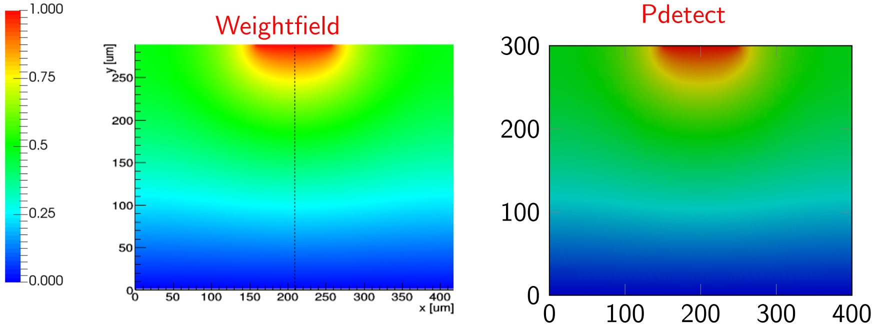


- The two curves perfectly match for a different strip width
- Strip width used to compute analytical solution is unknown

Pdetect software

Potential: comparison with Weightfield

V=1V, 1 strip, pitch = 400μm, strip length = 100μm, width = 300μm



User must choose

- particle trajectory (all supported)
- a precision level affecting
    - the time interval Δt of the simulation
    - the number of punctual charges

Initial charges are grouped among punctual charges

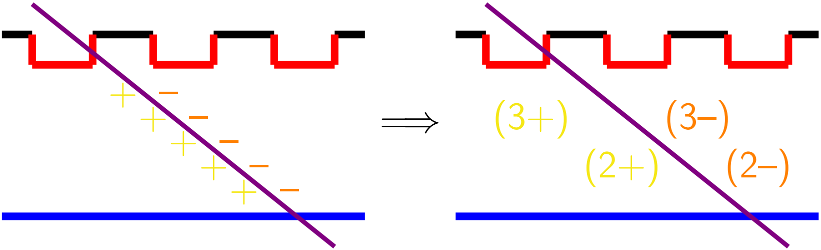
- equidistant and uniformly distributed along the trajectory
- punctual charge $= \Sigma$ physical charges at that point
- fewer than real charges $\implies$ faster computation

Pdetect software

Punctual / virtual charges

10 physical charges

4 virtual charges



Pdetect software

Computing the current: iteration

$t = 0$

while there are charges inside detector

$i_{tot} = 0$

for each punctual charge

1. computes $\vec{v}$ using $\vec{E}$
2. computes i using $\vec{v}$ and $\vec{E}_w$
   - $i_{tot} = i_{tot} + i$
3. move punctual charge
   - $(x, y) = (x, y) + (\Delta t \times v_x, \Delta t \times v_y)$

save (t, i_{tot})

$t = t + \Delta t$

Pdetect software

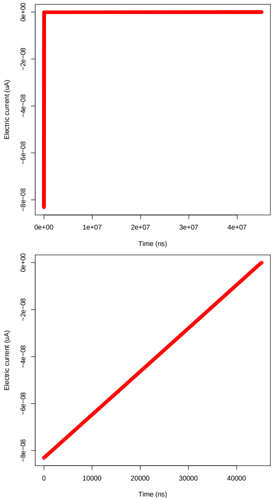
Computing the current: adaptative Δt

In gas detectors holes slower than e^- (1000 times)

- small Δt: fine to capture electrons current but far too slow to compute holes current
- large Δt: capture quickly holes current but miss electrons current

Solution: at each iteration select Δt depending on max speed at previous iteration

$$\Delta t = \frac{width}{max\_speed\_y \times 2^{precision\_level}}$$



Current computation algorithm accesses $\vec{E}(x, y)$ and $\vec{E}_w(x, y)$ very often $\implies$ access must be fast

The problem: find the cell (FE) in which the point (x, y) belong

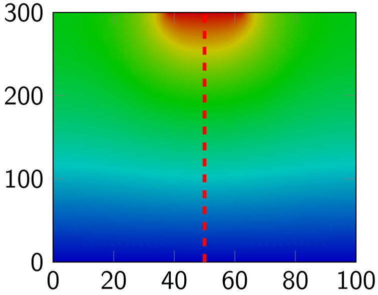
Running through each cell is too slow: $\mathcal{O}(n)$ time complexity ($n = \#$cells)

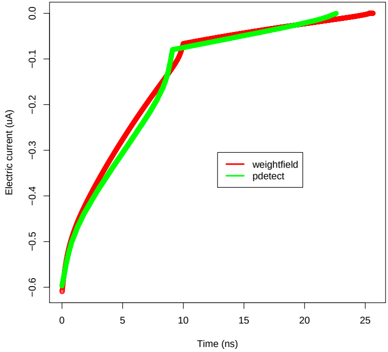
$\implies$ rtree data structure allows to access the right cell with $\mathcal{O}(\log(n))$ time complexity

Pdetect software

Computing the current: Pdetect vs Weightfield

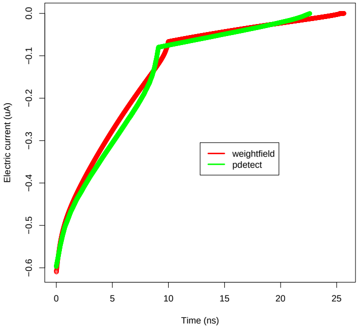
Current induce by particle passing vertically (middle of the detector)
1 strip, pitch = 100μm, strip length = 25μm, width = 300μm, V_{strip} = 100V





Pdetect software

Computing the current: Pdetect vs Weightfield



Some differences, due to effect in Weightfield:

- Depletion voltage
- Gain
- Temperature
- Different number of hole/electron

Conclusion

- This work is a first step to the development of a properly designed software to simulate particle detectors
- Computed potentials match analytical solution and Weightfield results
- Computed current for silicon detectors match Weightfield results
- Computed current for gas detectors need to be validated
- Solving the Laplace equation is fast for 2D geometries
- The slowest phase is the current computation

Future work

- Validate results for gas detectors and Townsend avalanche
- Simulation of additional physical effects: non uniform energy deposition by the particle in the detector,...
- Additional 2D geometries: gaussian curves as corners
- 3D geometries
- Graphic user interface
- Clusters, GPUs
- Simulation of the electronic
- Dependance regarding pressure and temperature
- External magnetic field
- ...

Number of hole-electron pairs

$$N = \frac{dE/dx}{w} \times L$$

- dE/dx energy deposited per unit of length
- w energy to produce one hole-electron pair
- L distance covered in the detector

Appendix
Saturation

$$\mu_s = \mu \left(1 + \left(\frac{\mu|E_y|}{v_{sat}}\right)^{\beta}\right)^{-\frac{1}{\beta}} \qquad (1)$$

- β is a constant
- μ is the mobility when not considering saturation
- E_y is the component of the electric field along the axis orthogonal to the anode and cathode
- v_{sat} the saturation velocity, it depends on the media and the temperature.

During Δt the initial electric charge q_0 is multiplied:

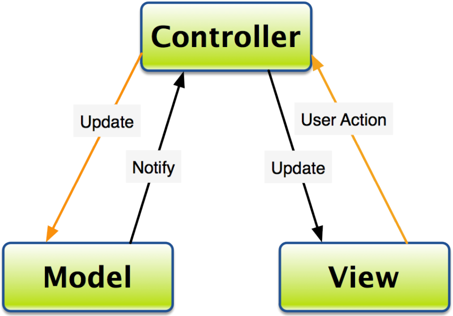
$$q = q_0 e^{\alpha \Delta x} \tag{2}$$

- Δx: distance covered by the e^- during Δt
- α is the first Townsend coefficient

$$\alpha = ap\ e^{-bp/E}$$

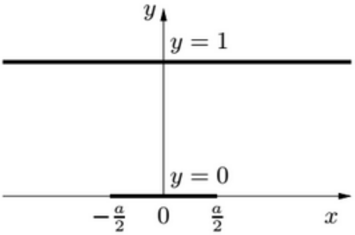
a, b: constants depending on the gas media
p: pressure in the detector
E: norm of the electric field at the charge position

- model manages the data, logic and rules of the application
- view: any output representation of information, multiple views of the same information are possible
- controller accepts input and converts it to commands for the model or view



Appendix

Potential analytical solution

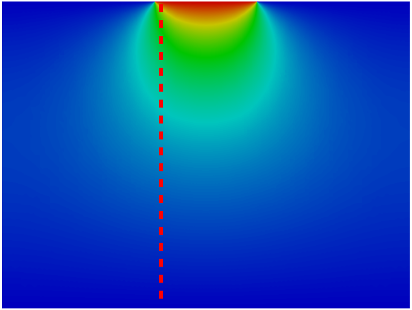


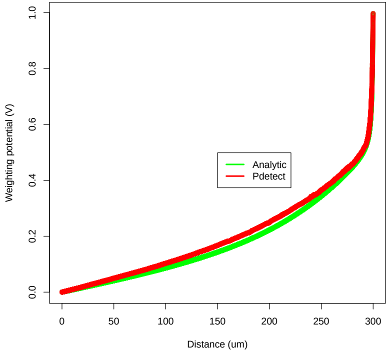
$$\phi_w(x, y) = atan\left(\frac{\sin(\pi y)\sinh(\pi \frac{a}{2})}{\cosh(\pi x) - \cos(\pi y)\cosh(\pi \frac{a}{2})}\right)$$

Appendix

Potential: numerical vs analytical solution (I)

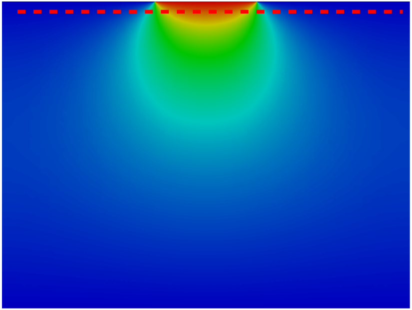
Potential along a vertical line (edge of the strip)
Pitch $= 400\mu$m, strip length $= 100\mu$m, width $= 300\mu$m, $V_{strip} = 1$V

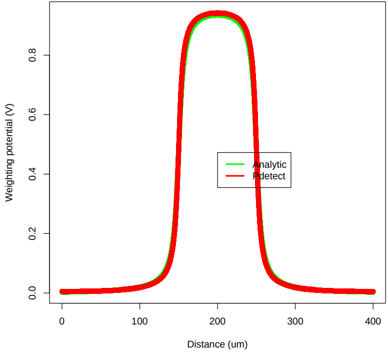




Appendix
Potential: numerical vs analytical solution (II)

Potential along a horizontal line (below the strip at y=298 μm)
Pitch = 400μm, strip length = 100μm, width = 300μm, V_{strip} = 1V

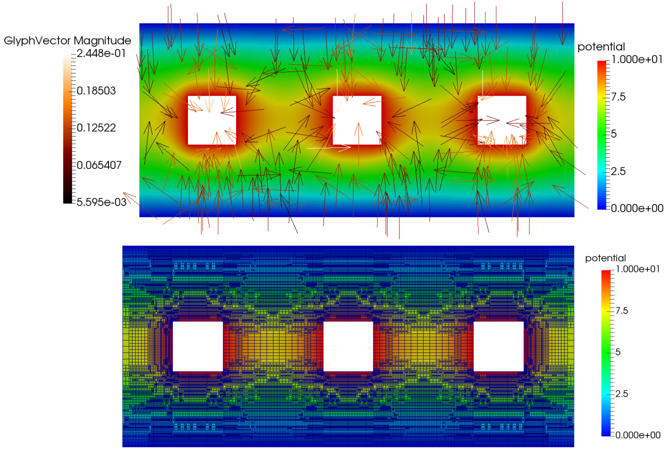




Pdetect software

Rectangular strips: potential and electric field

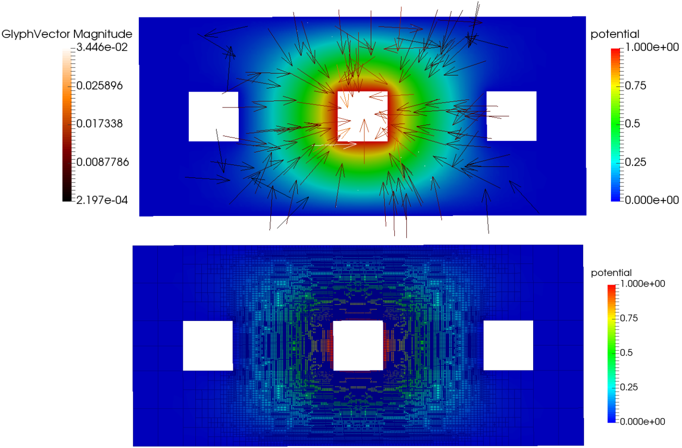
Detector width = 200 μm, strip length = 50 μm, strip width = 50 μm, inter strips distance = 100μm, number of strips = 3, potential = 10V, max relative error = 0.9%



Pdetect software

Rectangular strips: weighting potential and weighting electric field

Detector width $= 200\ \mu$m, strip length $= 50\ \mu$m, strip width $= 50\ \mu$m, inter strips distance $= 100\mu$m, number of strips $= 3$, potential $= 10$V, max relative error $= 0.9\%$



- Kelly error estimator
- Integration of the "jump" of gradient of the function on the sides.

$$\eta_K^2 = \sum_{F \in \delta K} c_F \int_{\delta K_F} \left(a \frac{\delta u_h}{\delta n} \right)^2 do$$

- Upper bound of the error
- Modified version of Kelly error system
- Reliable for Poisson equation

More information: see the deal.ii Kelly error estimator page