Description and simulation of physics of Resistive Plate Chambers

Vincent Français

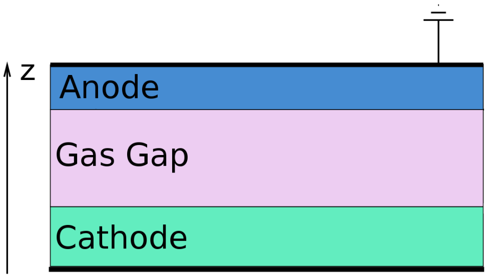
Laboratoire de Physique Corpusculaire de Clermont Ferrand

Université Blaise Pascal - CNRS/IN2P3

Basic single-gap design

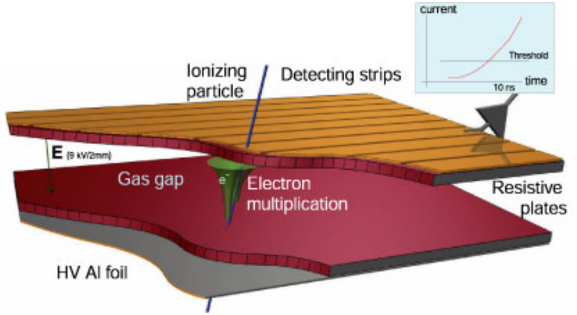
- Gap is 12 mm wide
- Glass anode and cathode ($10^{12}\ \Omega\, cm$, $\epsilon_r \sim 7$), 7 and 11 mm wide

- HV of 6.9 kV between plates (57.5 kV/cm)



Operation

- Particle crossing the detector $\rightarrow$ ionisation
- Freed electric charges drift and multiply under the influence of the HT $\Rightarrow$ electronic (Townsend) avalanche
- Electric signal arise on the electrode by induction



Gaseous mixture

- The gaseous mixture is maybe the most vital part of a RPC as it influences many key characteristics :
  - $\rightarrow$ ionisation (number of electrons freed)
  - $\rightarrow$ multiplication gain
  - $\rightarrow$ electron drift velocity (influences signal amplitude and timing)

- usually mixture is composed of 3 gases :
  1. ionizing gas $\sim 95\%$
  2. UV quencher gas $\sim 4\%$
  3. electron quencher gas $\sim 1\%$

- mixture used for this presentation :
  1. TFE $C_2H_2F_4$ 93%
  2. CO_2 5%
  3. SF_6 2%

State of the art and objectives

- Monte-Carlo simulation is essential in detector development $\rightarrow$ allows to predict characteristics and responses
- Simulations for RPC are not widespread and often incomplete
  - $\hookrightarrow$ unadapted mathematical distribution (Polya) which lacks physical interpretation
  - $\hookrightarrow$ overlook important phenomena

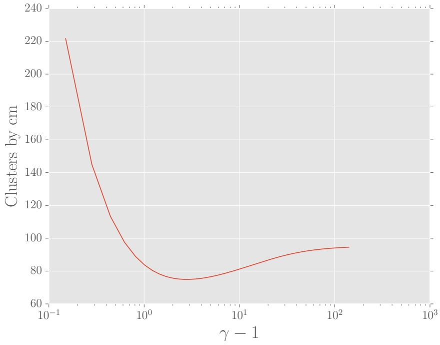
- modelise the main processes of an electronic avalanche
- develop a full, fast and multi-threaded Monte-Carlo simulation
- portable, easily modifiable and usable on various hardwares

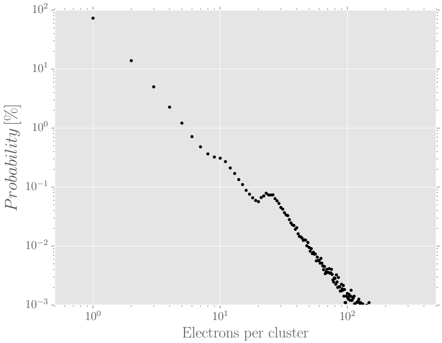
Ionisation

- charged particle crossing the gas gap $\rightarrow$ ionisation
- each ionisation event $\Rightarrow$ electron clusters
- charge deposit characterized by two things

$\rightarrow$ the number of clusters by unit of length

$\rightarrow$ the probability distribution for the number of electrons by cluster



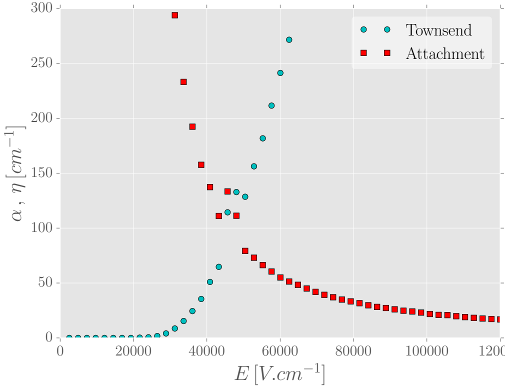


Electron multiplication

- electrons drift under the influence of the electric field and multiply by interaction with gas molecules (avalanche)
- evolution of the number of electrons conditioned by two coefficient :

Townsend coefficient $\alpha \rightarrow$ probability to multiply

Attachment coefficient $\eta \rightarrow$ probability to get attached



Avalanche development model (W. Riegler and C. Lippmann)

- average numbers of e^- and positive ions :

$$\bar{n}(x) = e^{(\alpha-\eta)x}$$

$$\bar{p}(x) = \frac{\alpha}{\alpha-\eta}\left(e^{(\alpha-\eta)x} - 1\right)$$

- stochastic multiplication and attachment for one e^-

$$n = \begin{cases} 0, & s < k\frac{\bar{n}(x)-1}{\bar{n}(x)-k} \\ 1 + floor\left[\ln\left(\frac{(\bar{n}(x)-k)(1-s)}{\bar{n}(x)(1-k)}\right) \cdot \frac{1}{\ln\left(1-\frac{1-k}{\bar{n}(x)-k}\right)}\right], & s > k\frac{\bar{n}(x)-1}{\bar{n}(x)-k} \end{cases}$$

with s a random number $\in [0, 1)$, $k = \eta/\alpha$

here x is the drifted distance

Multiplication procedure

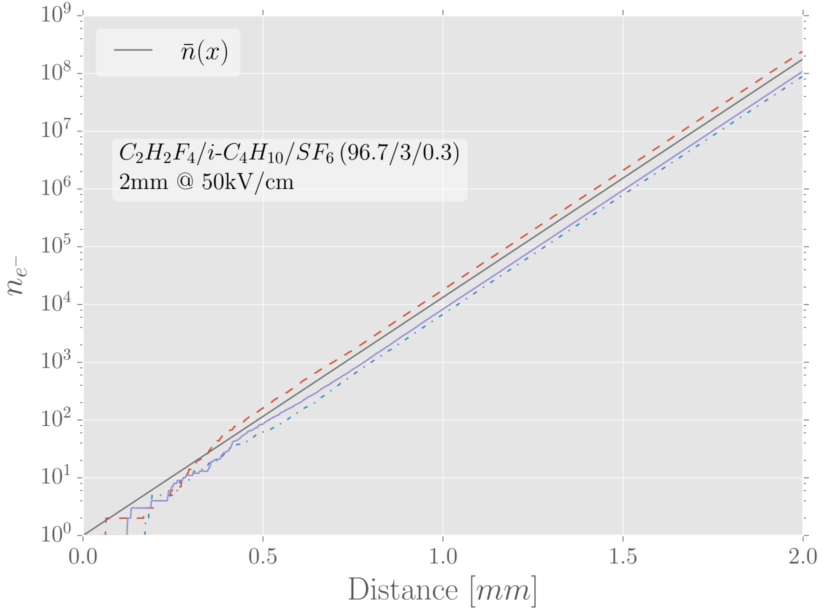
- gas gap divided into N steps of Δx ($\sim \mu m$)
- clusters are put into their respective bin

Case of one cluster at x_0

- n_0 electrons present at x_0
- each one of the n_0 electrons will multiply according to the previous formula and we find n_1 electrons at $x = x_0 + \Delta x$
- In the same way, the n_1 electrons will multiply and we find n_2 electrons at $x = x_0 + 2\Delta x$

$\rightarrow$ This procedure is iterated until all the electrons reach the anode

Multiplication procedure

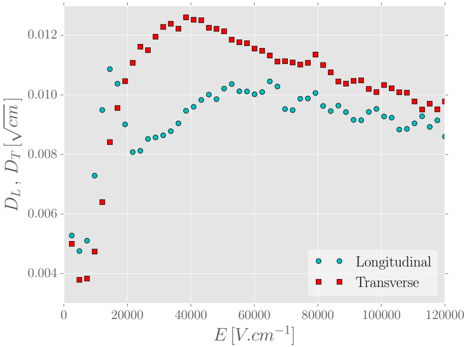


Diffusion

- Thermal diffusion motion superposed by drift motion $\Rightarrow$ anisotropic diffusion

$$\varphi_L = \frac{1}{\sqrt{2\pi}\sigma_L} \exp\left(-\frac{(z - z_0)^2}{2\sigma_L^2}\right)$$

$$\varphi_T = \frac{1}{\sigma_T^2} \exp\left(-\frac{(r - r_0)^2}{2\sigma_T^2}\right)$$

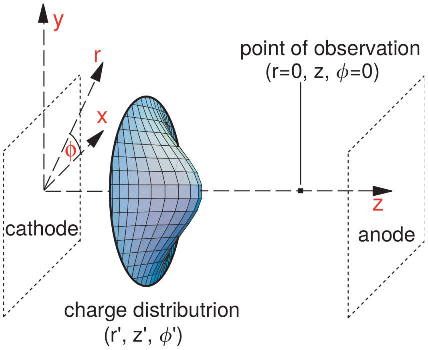


- diffusion characterized by their diffusion coefficient D_L and D_T and drifted distance l

$$\sigma_{L,T} = D_{L,T}\sqrt{l}$$

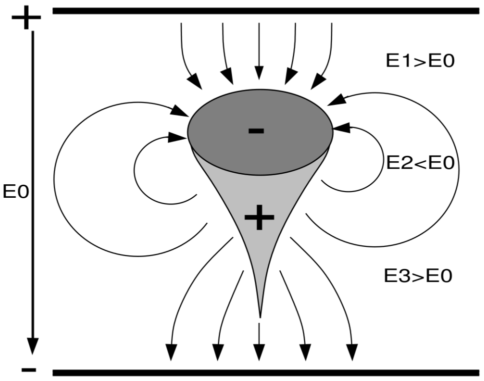
Transverse diffusion

⇒ Transversal : we consider the charges to be contained in a disk with a Gaussian radial distribution (φ_T) with $\sigma = D_T\sqrt{l}$ where l is the drifted distance



Space Charge Effect

- When the number of charges in avalanches is high enough they influence the electric field and thus the values of α and $\eta \implies$ Space Charge Effect



Space Charge Effect

- When the number of charges in avalanches is high enough they influence the electric field and thus the values of α and $\eta \Longrightarrow$ Space Charge Effect

Approximation to *feel* its impact : charges lie in sphere of radius r_d

$$E_r = \frac{e_0 \, n_e}{4\pi \, \varepsilon_0 \, r_d^2}$$

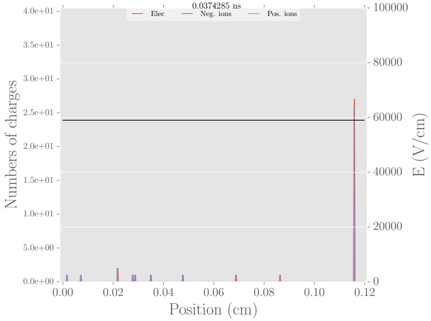
$n_e = 10^6 \quad r_d = 0.1mm$
$\Rightarrow E_r = 1.5 \, kV/cm$

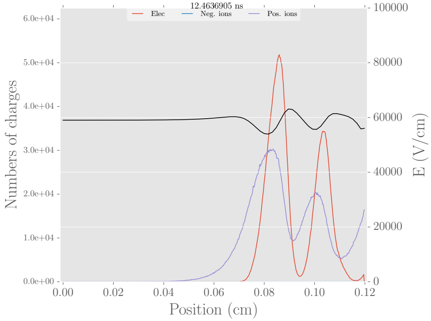
3% of typical RPC field ($\sim 50 \, kV/cm$) $\rightarrow 10\%$ change in coefficients (and so in multiplication gain)

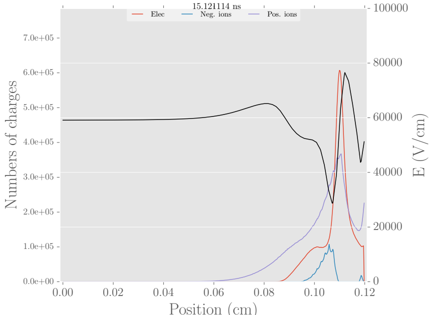
- Space Charge Effect leads to a saturation of the number of produced electrons

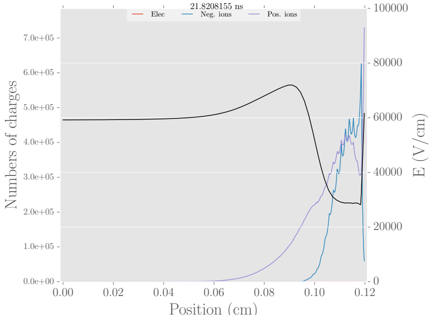
- Fully modelised by computing the field of all the charges in gas gap

Space Charge Effect illustration









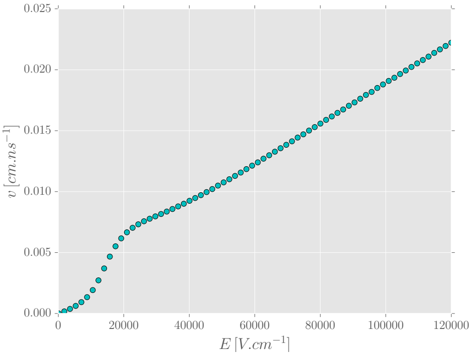
Signal Induction

- Output signal is only due to the movement of electrons in the electric field

$\rightarrow$ electrons in gas are not collected on electrodes as they are absorbed by resistive layer

$\rightarrow$ electrons movement induces charges on electrodes

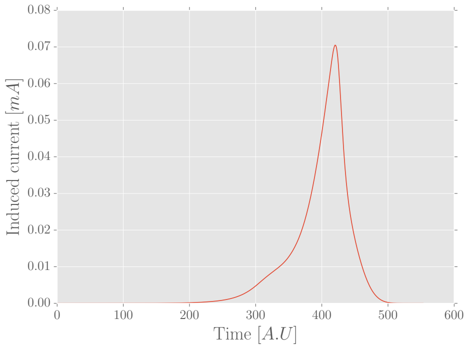
$\rightarrow$ ions don't contribute due to their small drift velocity

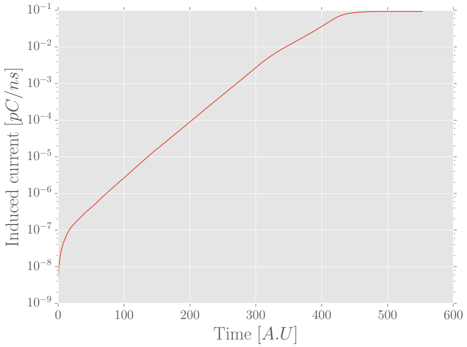


- We use the generalised Ramo's theorem to compute induced signals

Preliminary results

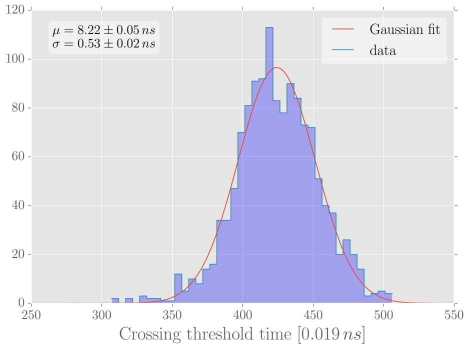
- Cathode 0.11 cm, Anode 0.07 cm, Gap 0.12 cm, HV 57.5 kV/cm
- Glass @ $10^{12}\,\Omega cm$

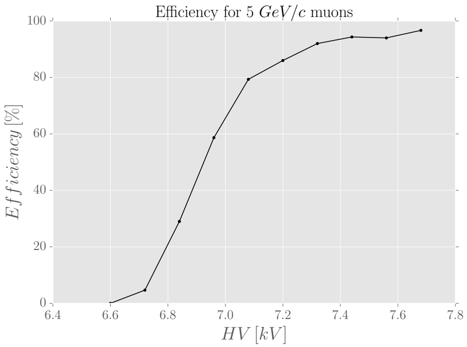




Preliminary results

- Cathode 0.11 cm, Anode 0.07 cm, Gap 0.12 cm, HV 57.5 kV/cm
- Glass @ $10^{12}\,\Omega cm$





Conclusion and perspective

- Resistive Plate Chambers are widely use, yet there is no proper simulation
- Working on a model taking into account the main physics processes
- Still work in progress but taking shape
- Aim to be a basic code for further RPC development and to be hooked in a more global detector simulation chain
- Room for improvements
  - $\rightarrow$ main bottleneck comes from the pseudo-random numbers generation
  - $\rightarrow$ using GPU (CuRand or Thrust) PRNG may give a significant speedup

BACKUPS

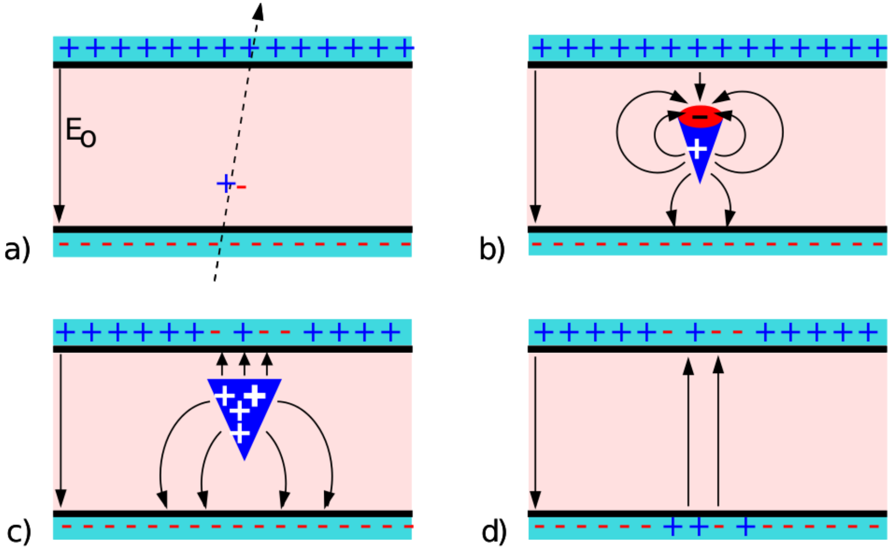
The simulation and libraries used

- Compiled on a server with an old GCC (4.4.7) $\Rightarrow$ No C++11 atm

- UNIX POSIX thread library
- Using the ThreadFactory (P. Schweitzer) to spawn threads and allocate events (one thread reserved for output writing)
- RngStreams (L'Ecuyer) for random number generation
- Using Garfield framework (rev. 418) with HEED (1.01) and Magboltz (9.01) for electron gas transport parameters and particle-gas interactions
- Gnu Scientific Library (QUADPACK) for integral computation, can also use python with scipy (more precise but much slower)
- TinyXml2 for configuration file parser

- Except Garfield (which use ROOT) and GSL, doesn't rely on a lot of libraries, all included in src

Electronic avalanche



Avalanche development model continued

$$n = \begin{cases} 0, & s < k\frac{\overline{n}(x)-1}{\overline{n}(x)-k} \\ 1 + \ln\left(\frac{(\overline{n}(x)-k)(1-s)}{\overline{n}(x)(1-k)}\right)\frac{1}{\ln\left(1-\frac{1-k}{\overline{n}(x)-k}\right)}, & s > k\frac{\overline{n}(x)-1}{\overline{n}(x)-k} \end{cases} \qquad \alpha, \eta > 0$$

$$n = \begin{cases} 0, & s < \frac{\alpha x}{1+\alpha x} \\ 1 + \ln\left[(1-s)(1+\alpha x)\right]\frac{1}{\ln\left(\frac{\alpha x}{1+\alpha x}\right)}, & s > \frac{\alpha x}{1+\alpha x} \end{cases} \qquad \alpha = \eta$$

$$n = \begin{cases} 0, & s < e^{(-\eta x)} \\ 1, & s > e^{(-\eta x)} \end{cases} \qquad \alpha = 0$$

Central Limit theorem

CPU-intense procedure $\Rightarrow$ very time consuming !

$\rightarrow$ Unadapted to the simulation of a large number of event

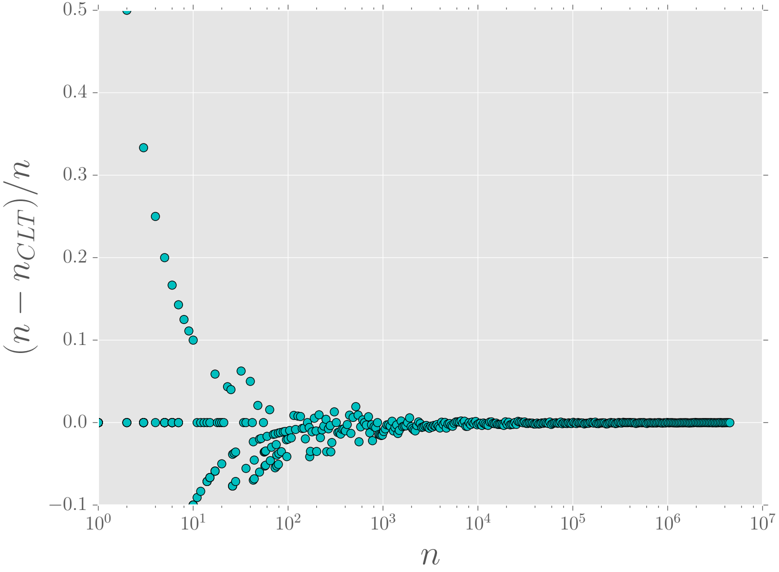
$\rightarrow$ We make use of the Central Limit Theorem :

when n_i is big enough we draw n_{i+1} from a gaussian

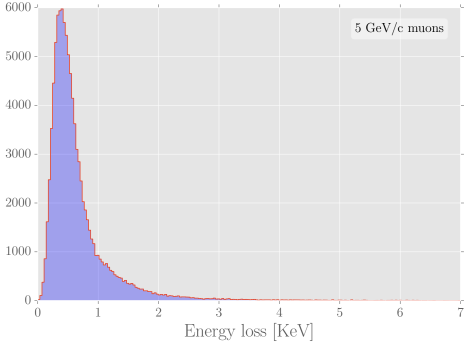
$$\mu = n_i \bar{n}(\Delta x) \qquad \sigma_{CLT} = \sqrt{n_i}\sigma(\Delta x)$$

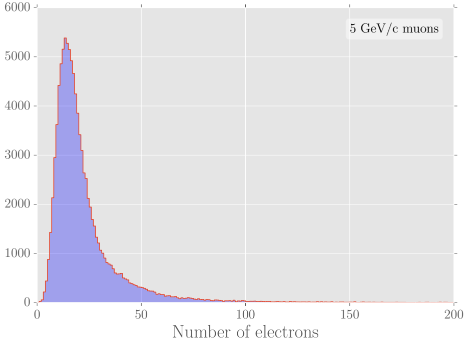
$$\sigma^2(\Delta x) = \left(\frac{1+k}{1-k}\right) \bar{n}(\Delta x)\left(\bar{n}(\Delta x) - 1\right)$$

Central Limit theorem



Primary ionisation





Space Charge Effect - potential

$$\begin{aligned}\Phi(r,\phi,z,r',\phi',z') &= \frac{Q}{4\pi\varepsilon_2}\left[\frac{1}{\sqrt{P^2+(z-z')^2}} - \frac{(\varepsilon_1-\varepsilon_2)}{(\varepsilon_1+\varepsilon_2)\sqrt{P^2+(z+z')^2}}\right.\\ &\quad - \frac{(\varepsilon_3-\varepsilon_2)}{(\varepsilon_3+\varepsilon_2)\sqrt{P^2+(2g-z-z')^2}}\\ &\quad \left. + \quad \frac{1}{(\varepsilon_1+\varepsilon_2)(\varepsilon_2+\varepsilon_3)}\int_0^\infty d\kappa\, J_0(\kappa P)\,\frac{R(\kappa,z,z')}{D(\kappa)}\right],\end{aligned}$$

$$0 \le z \le g\,;$$

$$\begin{aligned}D(\kappa) &= (\varepsilon_1+\varepsilon_2)(\varepsilon_2+\varepsilon_3)\left(1-e^{-2\kappa\,(p+q)}\right)\\ &- (\varepsilon_1-\varepsilon_2)(\varepsilon_2+\varepsilon_3)\left(e^{-2\kappa\,p}-e^{-2\kappa\,q}\right)\\ &- (\varepsilon_1+\varepsilon_2)(\varepsilon_2-\varepsilon_3)\left(e^{-2\kappa\,(p-g)}-e^{-2\kappa\,(q+g)}\right)\\ &+ (\varepsilon_1-\varepsilon_2)(\varepsilon_2-\varepsilon_3)\left(e^{-2\kappa g}-e^{-2\kappa\,(p+q-g)}\right)\end{aligned}$$

$$\begin{aligned}&R(\kappa;z,z') = \\ &(\varepsilon_1+\varepsilon_2)^2(\varepsilon_2+\varepsilon_3)^2\left[e^{\kappa(-2p-2q+z-z')} + e^{\kappa(-2p-2q-z+z')}\right] - \\ &(\varepsilon_1+\varepsilon_2)^2\,(\varepsilon_2-\varepsilon_3)^2\,e^{\kappa(-4g-2q+z+z')} - \\ &4\varepsilon_1\,\varepsilon_2(\varepsilon_2+\varepsilon_3)^2\,e^{\kappa(-2q-z-z')} - (\varepsilon_1-\varepsilon_2)^2\,(\varepsilon_2+\varepsilon_3)^2\,e^{\kappa(-2p-z-z')} - \\ &\left({\varepsilon_1}^2-{\varepsilon_2}^2\right)(\varepsilon_2-\varepsilon_3)^2\,e^{\kappa(-4g+z+z')} + \\ &\left({\varepsilon_1}^2-{\varepsilon_2}^2\right)(\varepsilon_2+\varepsilon_3)^2\left[-e^{\kappa(-2p-2q-z-z')}+e^{\kappa(-2p+z-z')}+e^{\kappa(-2p-z+z')}\right] - \\ &4\left({\varepsilon_1}^2-{\varepsilon_2}^2\right)\varepsilon_2\,\varepsilon_3\,e^{\kappa(-2p-2q+z+z')} - 4\,(\varepsilon_1+\varepsilon_2)^2\varepsilon_2\,\varepsilon_3\,e^{\kappa(-2p+z+z')} + \\ &(\varepsilon_1-\varepsilon_2)^2\left({\varepsilon_2}^2-{\varepsilon_3}^2\right)e^{\kappa(-2g-z-z')} + 4\,\varepsilon_1\,\varepsilon_2\left({\varepsilon_2}^2-{\varepsilon_3}^2\right)e^{\kappa(2g-2p-2q-z-z')} + \\ &(\varepsilon_1+\varepsilon_2)^2\left({\varepsilon_2}^2-{\varepsilon_3}^2\right)\\ &\left[-e^{\kappa(-2g-2q+z-z')}-e^{\kappa(-2g-2q-z+z')}+e^{\kappa(-2g-2p-2q+z+z')}\right] + \\ &\left({\varepsilon_1}^2-{\varepsilon_2}^2\right)\left({\varepsilon_2}^2-{\varepsilon_3}^2\right)\\ &\left[e^{\kappa(-2g-2q-z-z')}-e^{\kappa(-2g+z-z')}-e^{\kappa(-2g-z+z')}+e^{\kappa(-2g-2p+z+z')}\right].\end{aligned}$$

Space Charge Effect computation

- The unit charge is assumed to be contained in a disc perpendicular to the z-axis, so its electric field is

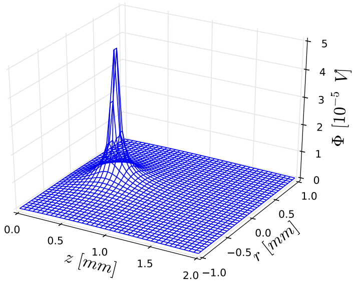
$$\overline{E}(z,l,z') = -\int_0^\infty \varphi_T(r',l)\,\frac{\partial\phi(z,r',z')}{\partial z}\,r'dr'$$

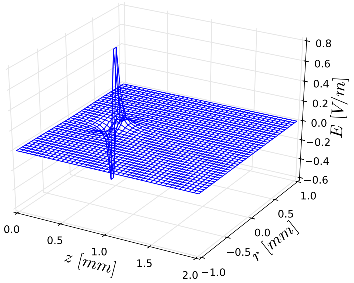
- Then the total space charge field at z is given by summation of all the discs :

$$E_{SC}(z) = \sum_{n=0}^{N} q_n \overline{E}(z_n, l_n, z'_n)$$

Very time consuming !

Space Charge Effect





$\rightarrow$ Need to compute an integral inside another integral (semi improper) $\Rightarrow$ Very time consuming

$\rightarrow$ Values of $\overline{E}$ are loaded in memory from a pre-computed table. Using interpolation during simulation

Ramo’s theorem

$$\boxed{i = e_0\, E\, v_e}$$

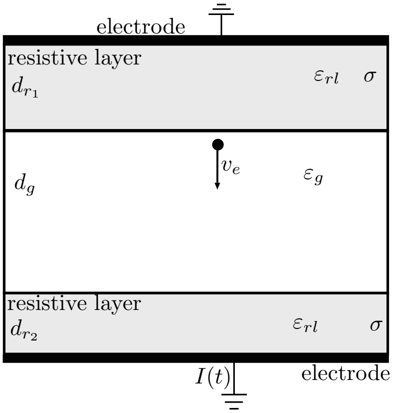
$\hookrightarrow$ doesn’t hold in case we have resistive materials $\Longrightarrow$ time-dependent fields

$\Rightarrow$ Maxwell’s equations in quasi-static approximation, for medium with time- and space-dependent permittivity and conductivity (sparing some ugly algebra we have)

$$i(t) = \frac{Q}{V_0} \int_0^t E_\Psi(\vec{x}(t'), t - t')\dot{x}(t')dt'$$

- $\vec{E}_\Psi$ is the weighting field, ie the field in detector if all conductors grounded but one put to voltage V_0. Depends only on detector geometry

Weighting field

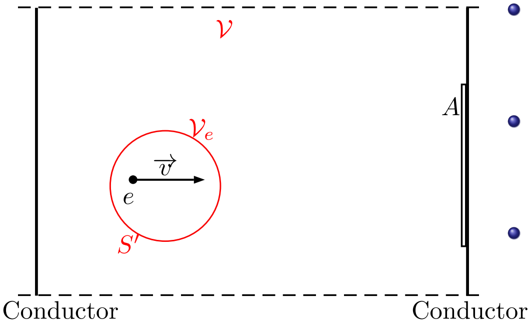


- single gap chamber with resistive layers of permittivity $\varepsilon_{rl} = \varepsilon_r \varepsilon_0$, gas of $\varepsilon_g \sim \varepsilon_0$

$$\frac{E_\Psi(t)}{V_0} = \frac{\varepsilon_r}{(d_{r_1} + d_{r_2}) + \varepsilon_r d_g} \delta t$$

$$i(t) = e_0 \, N(t) \, v_e \, \frac{\varepsilon_r}{(d_{r_1} + d_{r_2}) + \varepsilon_r d_g}$$

Ramo's theorem



- Make use of Green's theorem with volumes $\mathcal{V}$ (detector) and $\mathcal{V}_e$ (surrounding the electron)
- V is potential between conductors (removing space $\mathcal{V}_e$), V_e potential including electron
- consider conductors are grounded except A which is put to $1V$ and electron is removed : $V \rightarrow V' \;\; V_e \rightarrow V'_e$

- playing with Green's theorem with potentials defined above we get

$$Q_A = -e_0 \cdot V'_e$$

$$i = \frac{dQ_A}{dt} = -e_0 \cdot \frac{dV'_e}{dt} = -e_0 \cdot \frac{\partial V'_e}{\partial x} \frac{dx}{dt}$$

$$i = e_0 \, E \, v_e$$

Performance and bottlenecks

- Average execution time between 1 and 8 mins
  - $\rightarrow$ Depends heavily on detector geometry and HV

- Main bottleneck : Pseudo-random number generation
  - $\rightarrow$ need to draw a random number by electron at each simulation step
  - $\rightarrow$ typically 500-600 simulation steps, at its peak an avalanche can contain up to 10^8 electrons
  - $\hookrightarrow$ GPU computing could be a solution

Pseudo-Random Number Generation

- Parallel (multi-thread) simulation $\Rightarrow$ each thread compute an event
- Each thread needs to have its own independent stochastic streams to achieve reproducibility and avoid stochastic streams overlap
- We use the RngStreams package (MRG32k3a) by L'Ecuyer
  $\rightarrow$ Produces 2^{64} non-overlapping streams of length 2^{127}

- Performance of RngStreams (i5-3230M CPU @ 2.60GHz) :

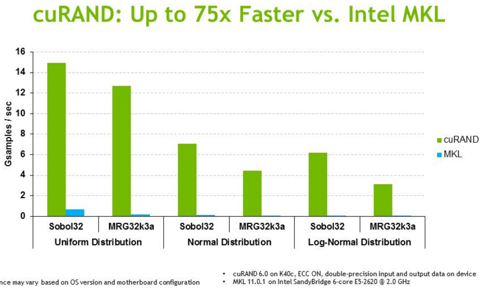
  10^6 numbers : ~0.45sec

  10^7 numbers : ~1sec

  10^8 numbers : ~3.3sec

  10^9 numbers : ~32sec

$\rightarrow$ hence the interest of using CUDA



from NVidia Cuda developer site