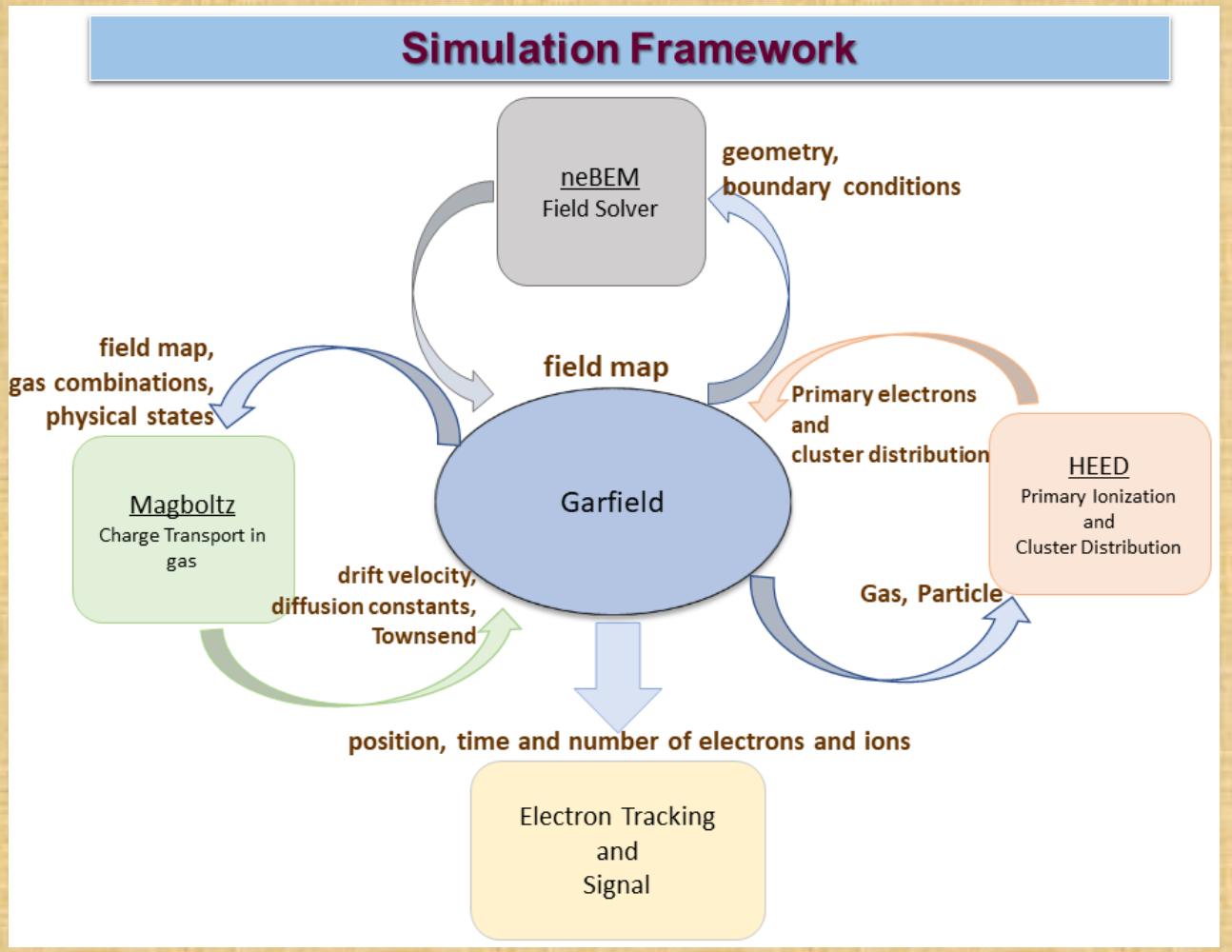
# Algorithm for Electric Field Calculation in Garfield++ with neBEM

Supratik Mukhopadhyay

on behalf of the neBEM team

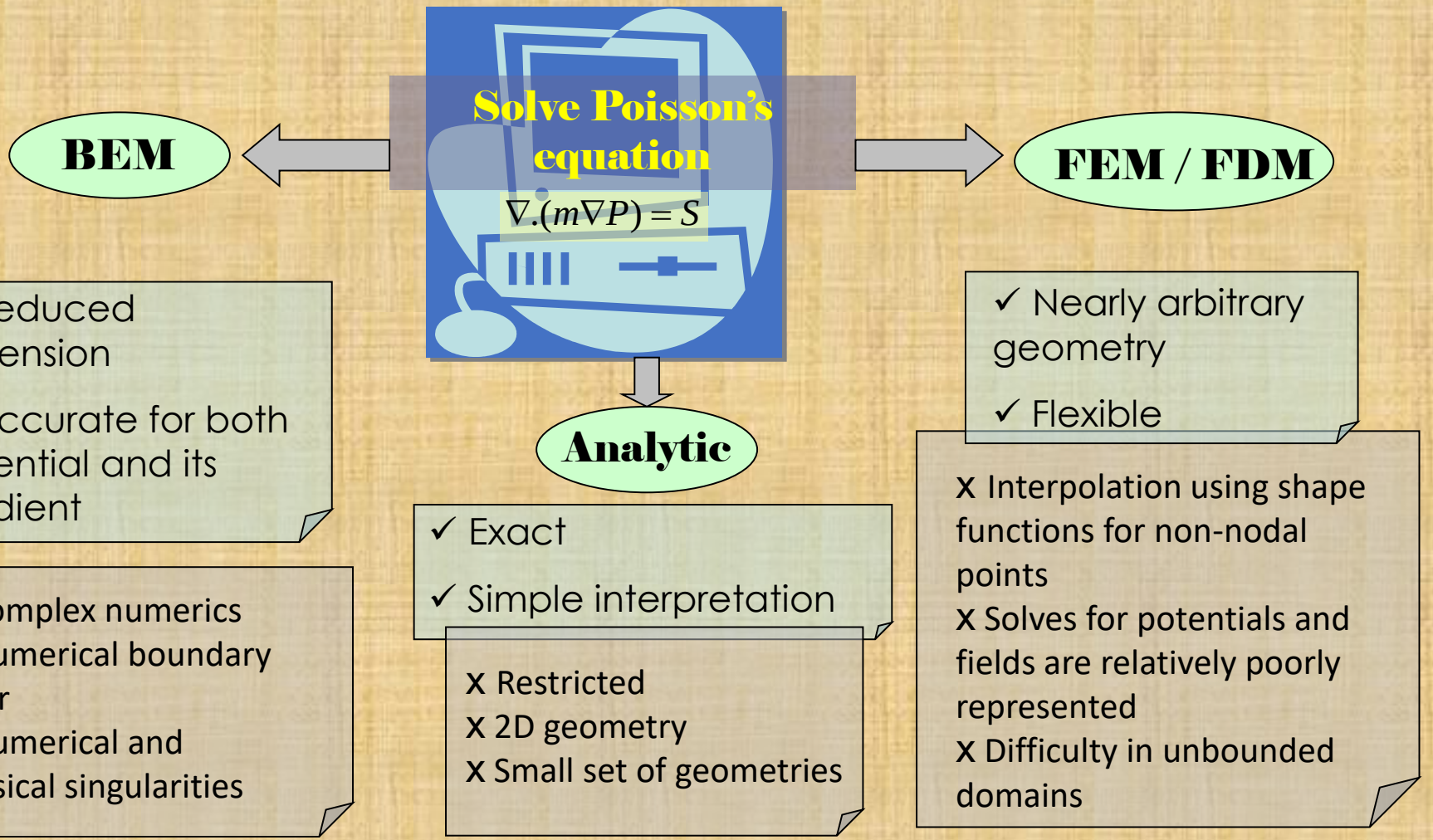
# RD51 simulation framework since 2009




Each and every component is open-source and free

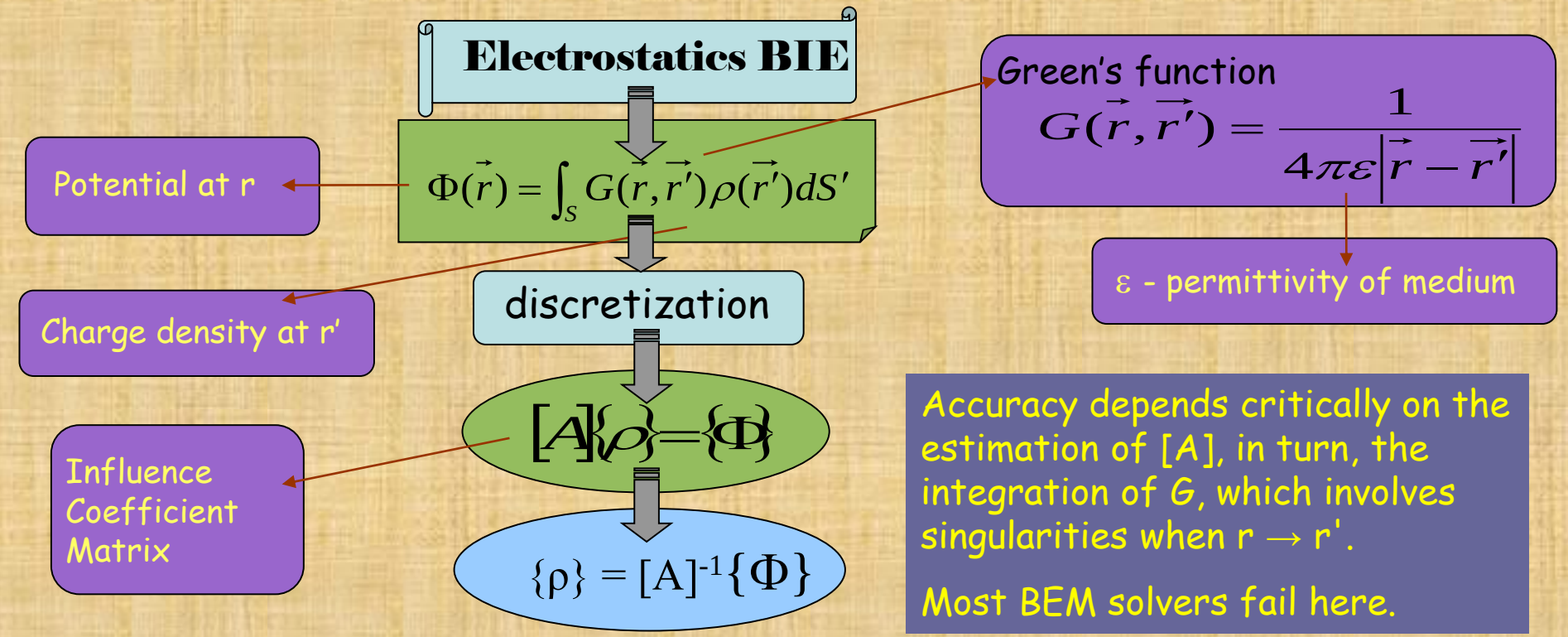
Garfield: garfield.web.cern.ch, Heed: heed.web.cern.ch,
Magboltz: magboltz.web.cern.ch, neBEM: nebem.web.cern.ch

# Field Solver Possibilities



# Solution of 3D Poisson's Equation using BEM

- Numerical implementation of boundary integral equations (BIE) based on Green's function by discretization of boundary.
- Boundary elements endowed with distribution of sources, doublets, dipoles, vortices.




Accuracy depends critically on the estimation of [A], in turn, the integration of G, which involves singularities when r → r'.

Most BEM solvers fail here.

# Conventional BEM

**Major Approximations**

- While computing the influences of the singularities, the singularities modeled by a sum of known basis functions with constant unknown coefficients.
- The strengths of the singularities solved depending upon the boundary conditions, modeled by shape functions.

**Numerical boundary layer**

*Constant element approach*

Singularities assumed to be concentrated at centroids of the elements, except for special cases such as self influence.

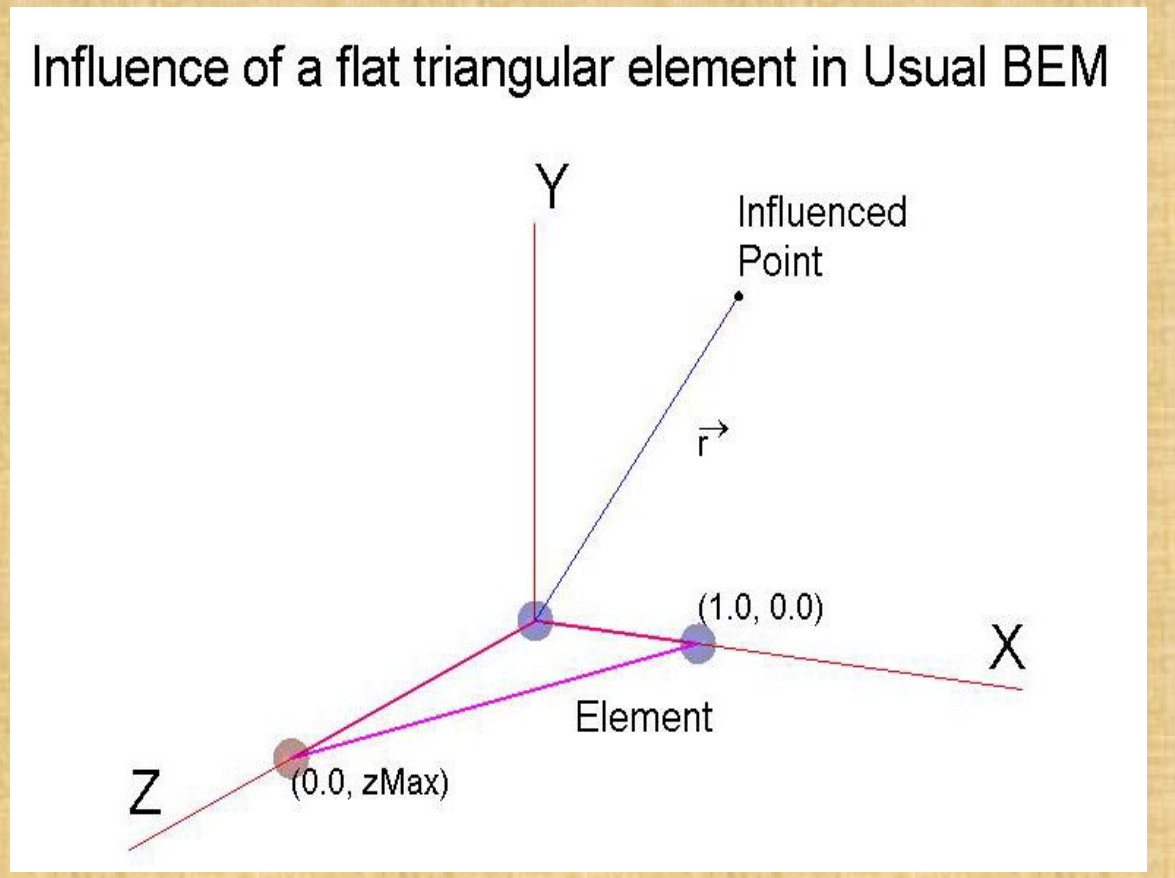
Mathematical singularities can be removed: Sufficient to satisfy the boundary conditions at centroids of the elements.

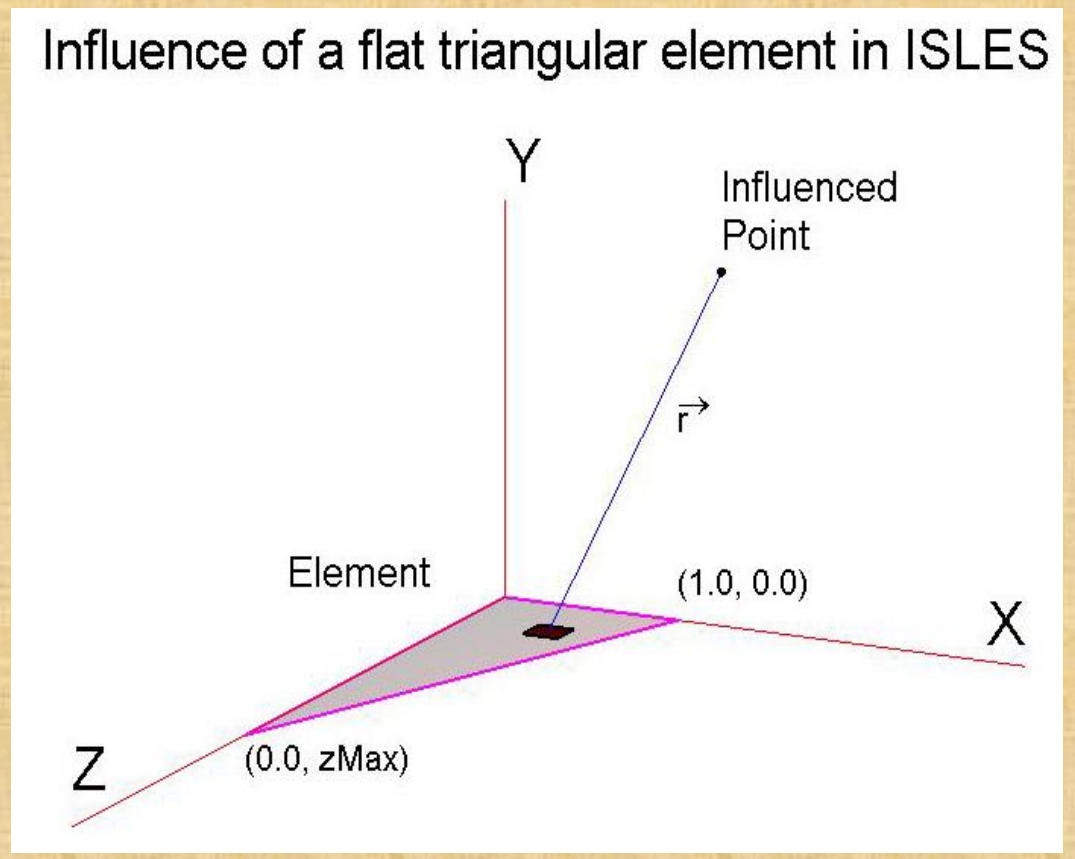
**Difficulties in modeling physical singularities**

- geometric singularity
- boundary condition singularity

# Contrast of approaches

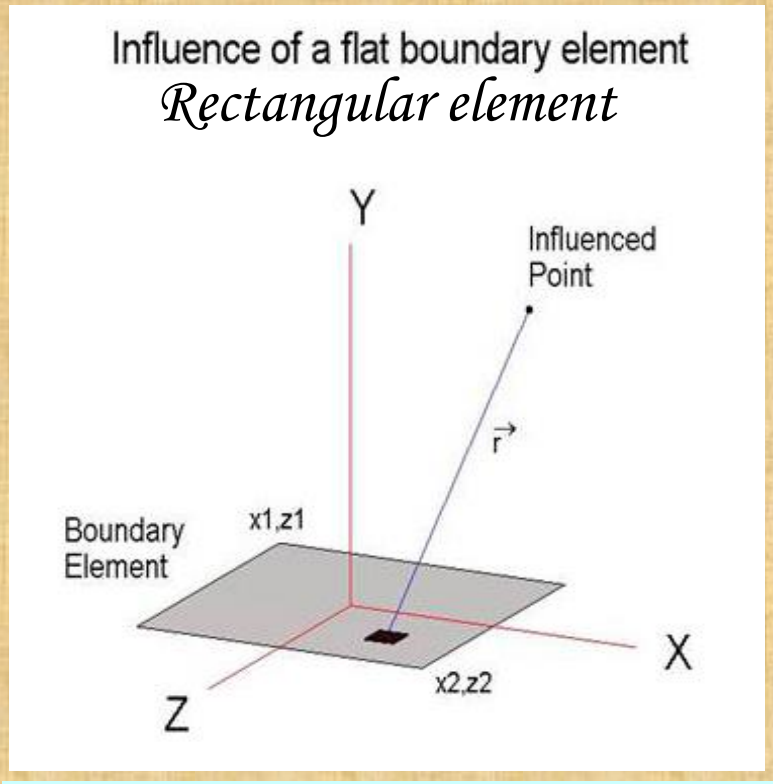
nodal versus distributed



# Foundation expressions of ISLES

*Inverse Square Law Exact Solutions*




$$\Phi(X,Y,Z) = \int_{x1}^{x2}\int_{z1}^{z2} \frac{dxdz}{\sqrt{(X-x)^2+(Y-y)^2+(Z-z)^2}}$$

*Value of multiple dependent on strength of source and other physical consideration*

$$\Phi(X,Y,Z) = \frac{1}{2}\times\left\{\begin{array}{l} 2\times(X\,|\,Z\,|\,x_i\,|\,z_j)\times\ln\left(\frac{D_{i,j}-(X\,|\,Z-x_i\,|\,z_j)}{D_{m,n}-(X\,|\,Z-x_m\,|\,z_n)}\right) \\ +iS_j|Y|\times\left[\tanh^{-1}\left(\frac{R_j-iI_i}{D_{i,j}|Z-z_j|}\right)-\tanh^{-1}\left(\frac{R_j+iI_i}{D_{i,j}|Z-z_j|}\right)\right] \end{array}\right\} - 2\pi Y$$

4 log terms

4+4 complex $\tanh^{-1}$ terms

$$D_{i,j} = \sqrt{(X-x_i)^2+Y^2+(Z-z_j)^2}$$
$$R_i = Y^2+(Z-z_i)^2$$
$$I_i = (X-x_i)|Y|$$
$$S_i = Sign(Z-z_i)$$

May need translation and vector rotation

# neBEM approach

**Analytic expressions of potential and force field at any arbitrary location due to a uniform distribution of source on flat *rectangular* and *triangular* elements. Using these two types of elements, surfaces of any 3D geometry can be discretized.**

**Restatement of the approximations**

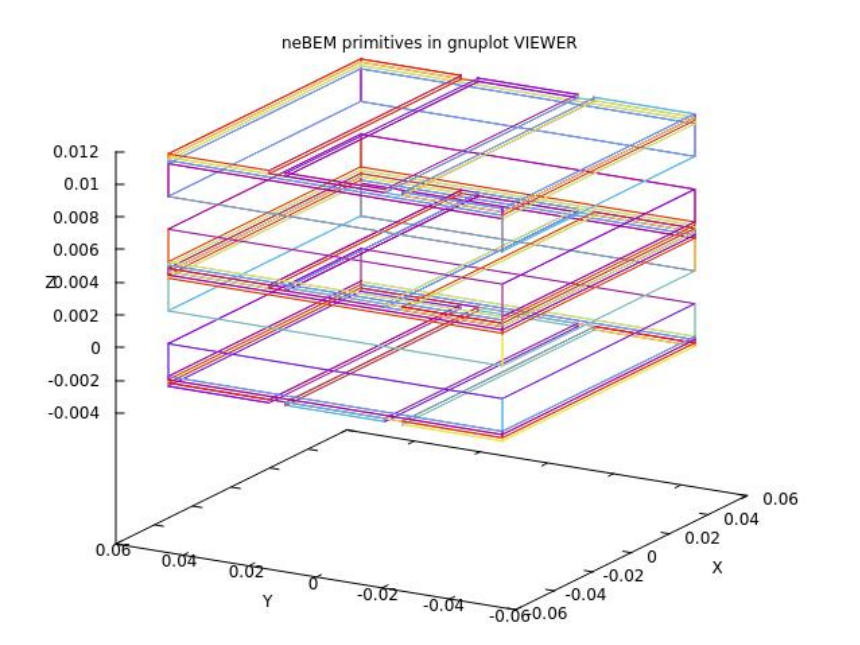
- Singularities distributed uniformly on the surface of boundary elements
- Strength of the singularity changes from element to element.
- Strengths of the singularities solved depending upon the boundary conditions, modeled by the shape functions

**ISLES library and neBEM Solver**

***Foundation expressions are analytic and valid for the complete physical domain***

# Before neBEM starts working …

- The device is created before neBEM begins its work.
  - Simple Constructive Solid Geometry (CSG) is used at present.
  - Has huge scope of improvement.
- Device creation can be done by any code that specifies a 2D / 3D geometry in terms of few basic primitives – lines, rings, discs, triangular areas, rectangular areas.
  - It is important to have primitives with non-overlapping materials. This can be a difficult criterion to meet.
- Specify
  - Potentials on different conductors.
  - Dielectric permittivity of insulating components.



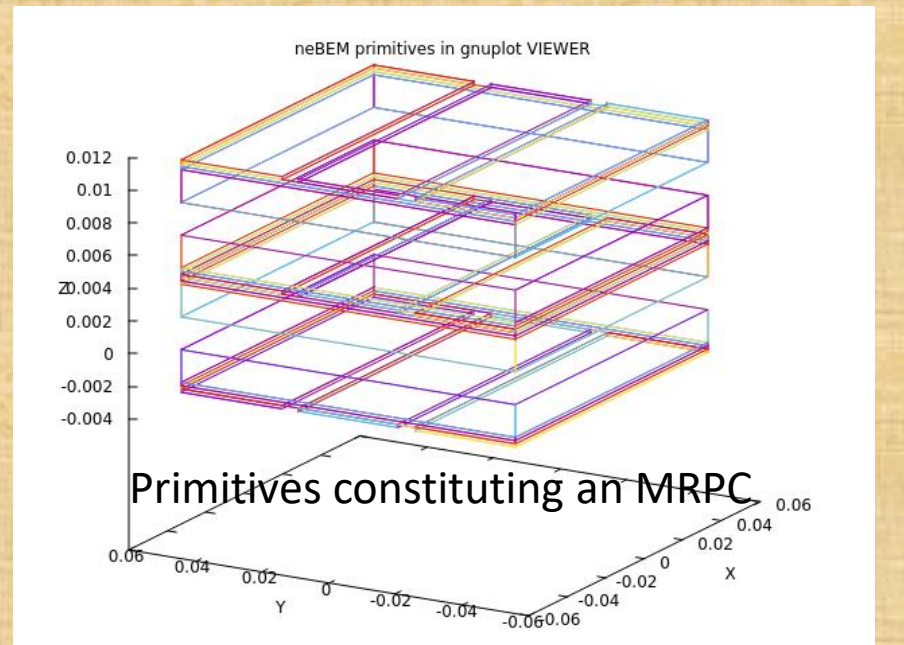

An MRPC represented by simple primitives.
Two gas gaps, nine readout electrodes.
Necessary glass sheets, graphite coatings, mylar sheets are all there.

# neBEM algorithm

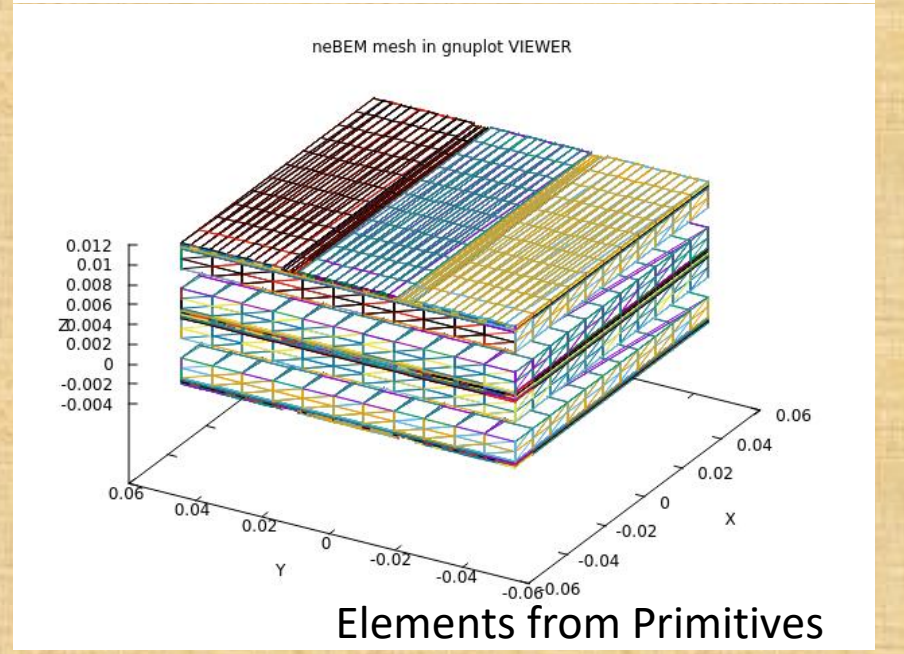
1. Discretize primitives into small elements.
   - charge density on each element is considered uniform.
2. Assign boundary conditions on these elements (usually in terms of potential, or field).
3. Prepare the influence coefficient matrix (potential and field at the $j^{th}$ element due to unit charge density on the $i^{th}$ element).
4. Incorporate influence due to charged dielectrics and space charge accumulation.
5. Solve for the charge density on each element that satisfies the prescribed boundary condition.
6. Use the charge density on device boundaries and other known charges (space charge, charging up etc.) to compute potential and field at any arbitrary point.

# 1 & 2) Discretize primitives

- Discretize primitives to create elements.
  - Line primitives give rise to line elements.
  - For area primitives, try to create as many rectangular elements as possible, since triangular elements are computationally more demanding.
  - Try to minimize aspect ratio of the elements.
- Using knowledge of primitives, specify boundary conditions on elements.
- Has huge scope of improvement.




Primitives constituting an MRPC

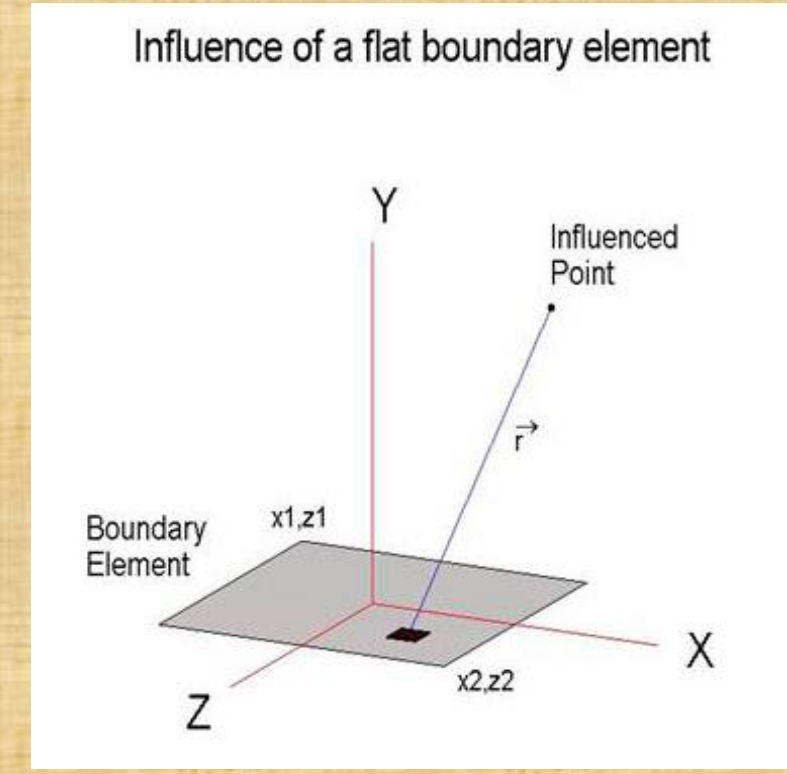



Elements from Primitives

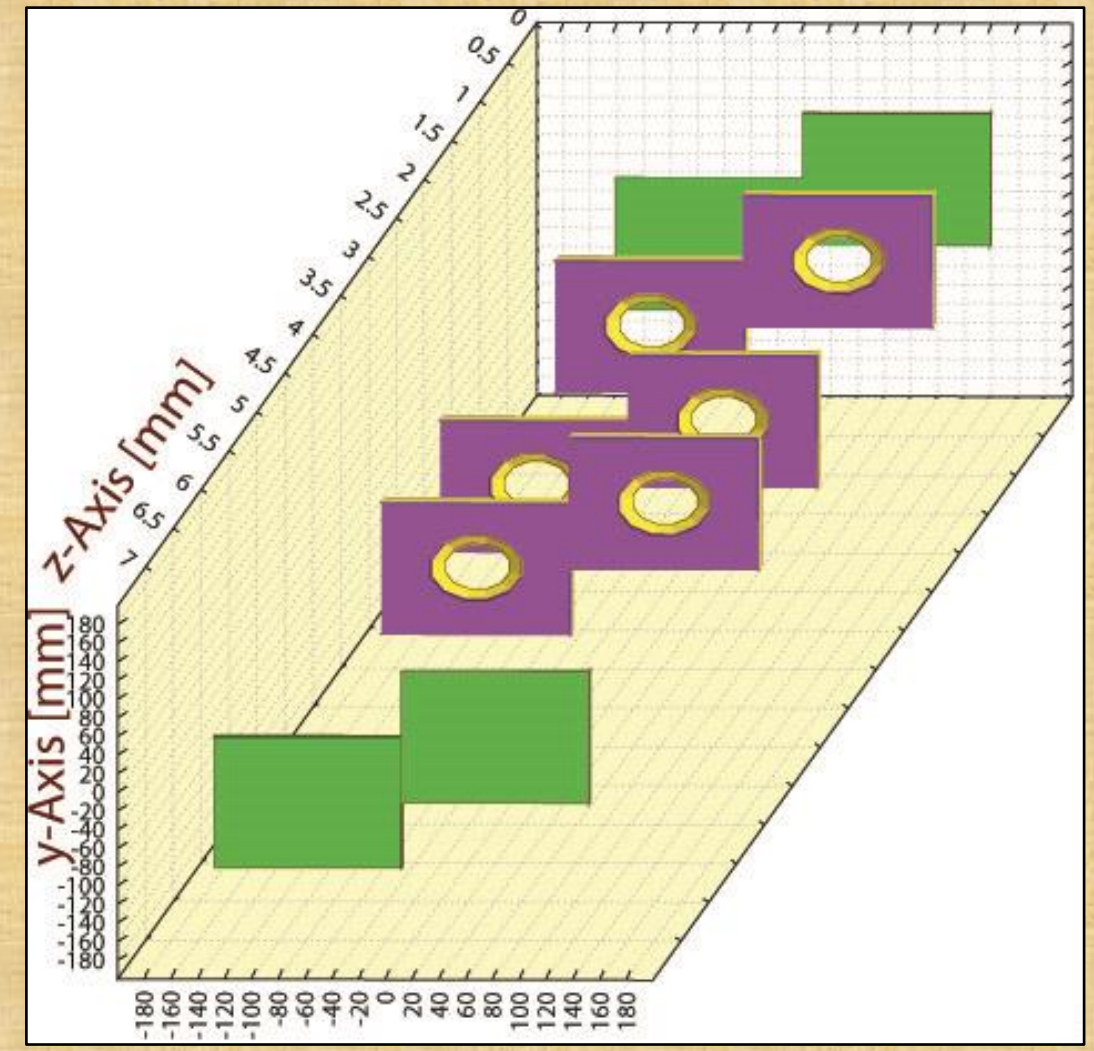
# 3, 4 & 5) Influence coefficient matrix generation, decomposition and inversion

- Influence coefficient matrix: Extensive use of ISLES Library.
- Matrix decomposition using LU, or SVD: GSL and NR.
- Solve for charge densities that satisfy the boundary conditions.

- Each step is time-consuming and resource hungry.
- Evaluation of many mathematical functions, conditional statements.
- Coordinate transformations (Global to element-local and vice-versa).
- Intense work going on to accelerate all these processes
  - OpenMP implemented partially.
  - GPU (NVidia, CUDA) in progress.



# Reduced-Order Modelling (ROM)




**Typical scenario for a Triple-GEM**

| | |
|---|---|
| **Foil thickness :** | **50 µm** |
| **Copper thickness :** | **5 µm** |
| **Hole dia (outer) :** | **70 µm** |
| **Hole dia (inner) :** | **50 µm** |
| **Hole pitch :** | **140 µm** |
| **(staggered)** | |
| **Gap configuration :** | **3:1:2:1 (mm)** |

**Repetitions in X and Y: ~100**

Base device

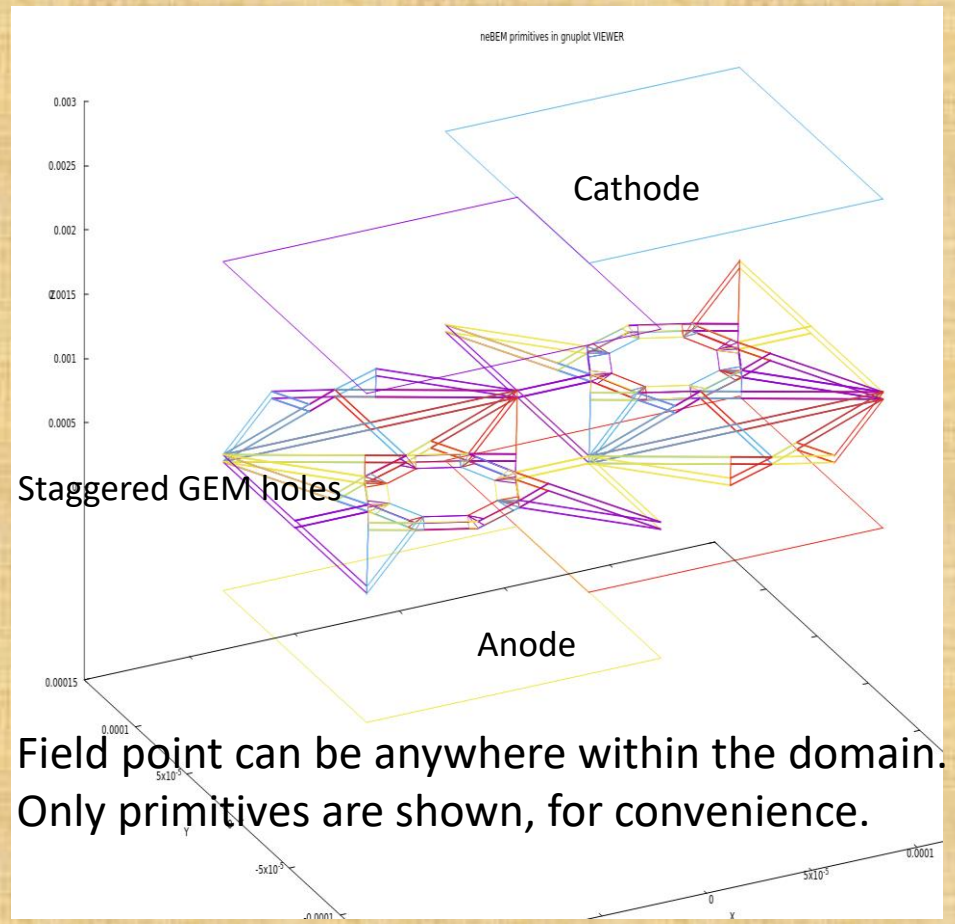
*Can we ignore the variation of charge density on a virtual GEM that is far away from the base device?*

*Repetitions that are far from the field point, consider their structure to be made of primitives, rather than elements.*

Has good potential. Present implementation being improved.

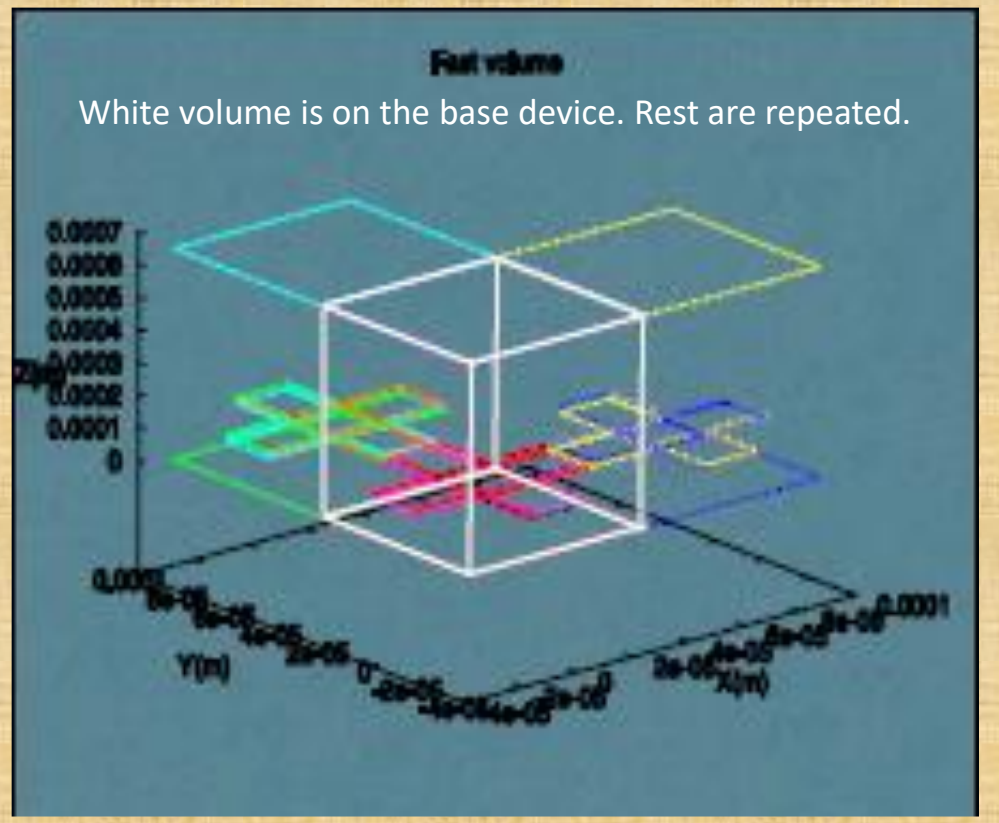
# 6) Evaluation of electric potential / field

- Large number of field points where properties are evaluated.
  - Influenced by all the elements / primitives.
  - Charges due to charging up process and space charge accumulation need to be considered.
  - Other physical / chemical processes can complicate matter further.
- Extensive use of the ISLES library (influence for charges represented as points, lines, areas, rings, discs).
- Very recently, Heinrich modified a part of the code that has accelerated computation by good measure.
- Intense activity here as well:
  - OpenMP has been partially implemented.
  - GPU computations are under way.



# Fast Volume is a way out

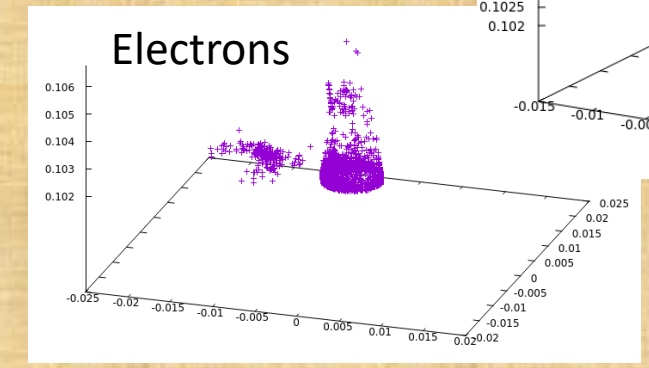
- Potential and field provided on a cartesian 3D map.
- Potential and field evaluated at arbitrary points by trilinear interpolation.
- Implementation available but can be improved further.
- This part of the work will be taken up once OpenMP and GPU implementations are in place.

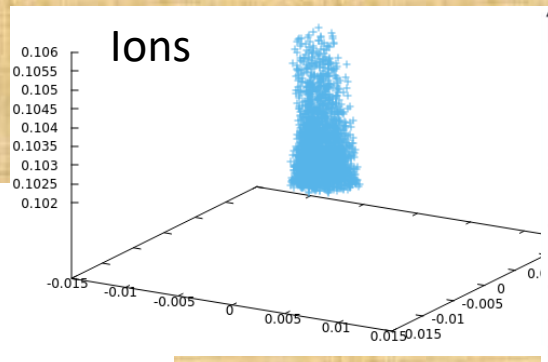


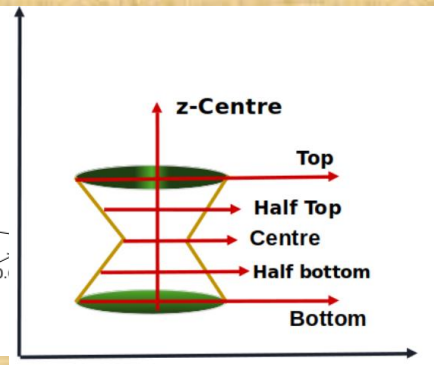

Sorry for the state of the figure – could not locate a better one

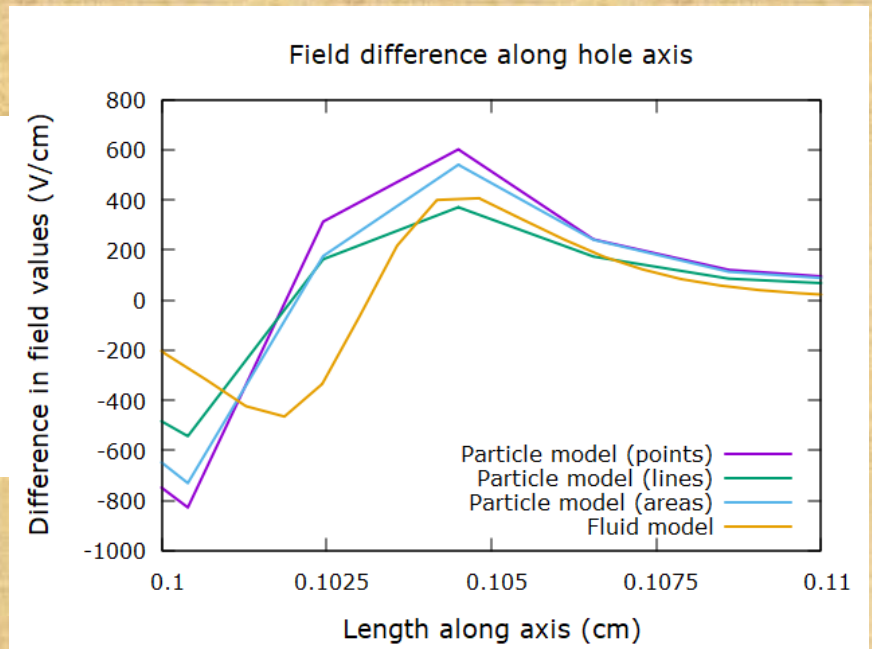
# Space charge and charging up

- These additional charges accumulated in the gas volume, or the insulating boundaries, have the option of being represented as
  - Points, lines, rings, discs and, triangular and rectangular areas.
  - Different representations (including fluid model) lead to similar results, thankfully!
- Especially GPU computation can accelerate this part of the computation and work in under way.



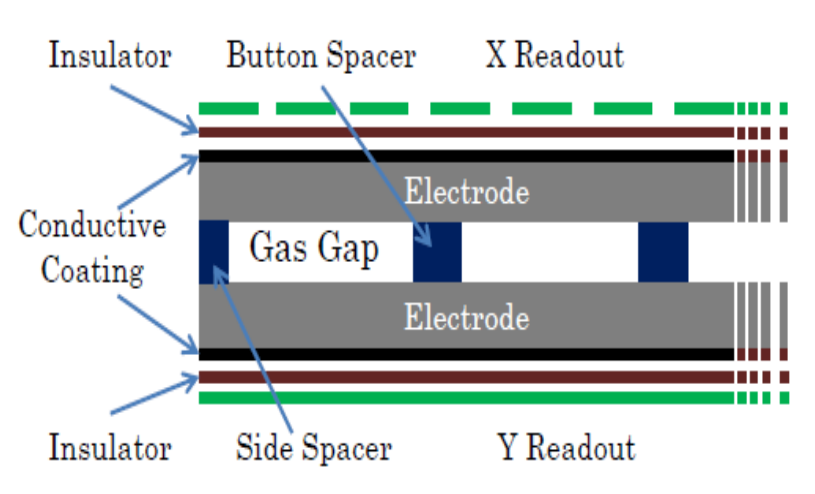



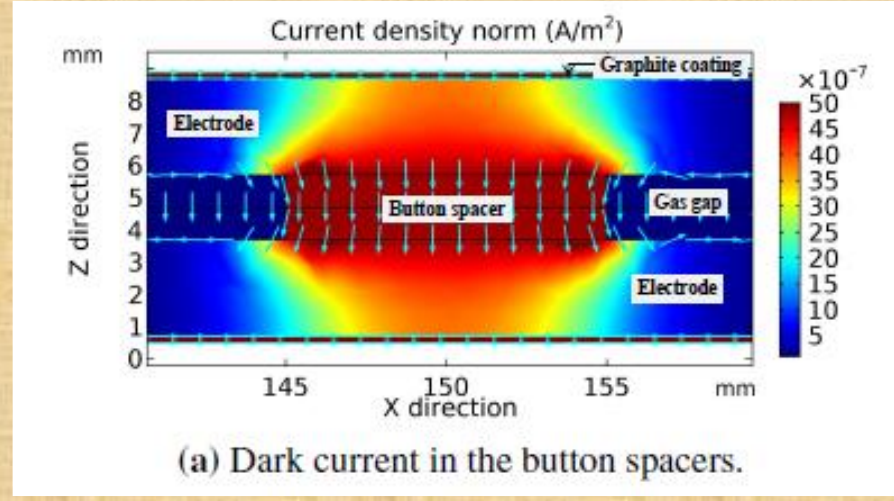
# Resistive materials

- This is one aspect that we have not worked upon using Garfield++.
  - FEM packages have been used to compute the effects of such materials.
  - The necessary governing equations will be implemented in neBEM.
- Excellent work has recently been done in the community on signal induction through resistive materials.
  - Hope to implement such models in neBEM. External support welcome.




RPC material studies using FEM (Comsol)

Dark current through a button spacer

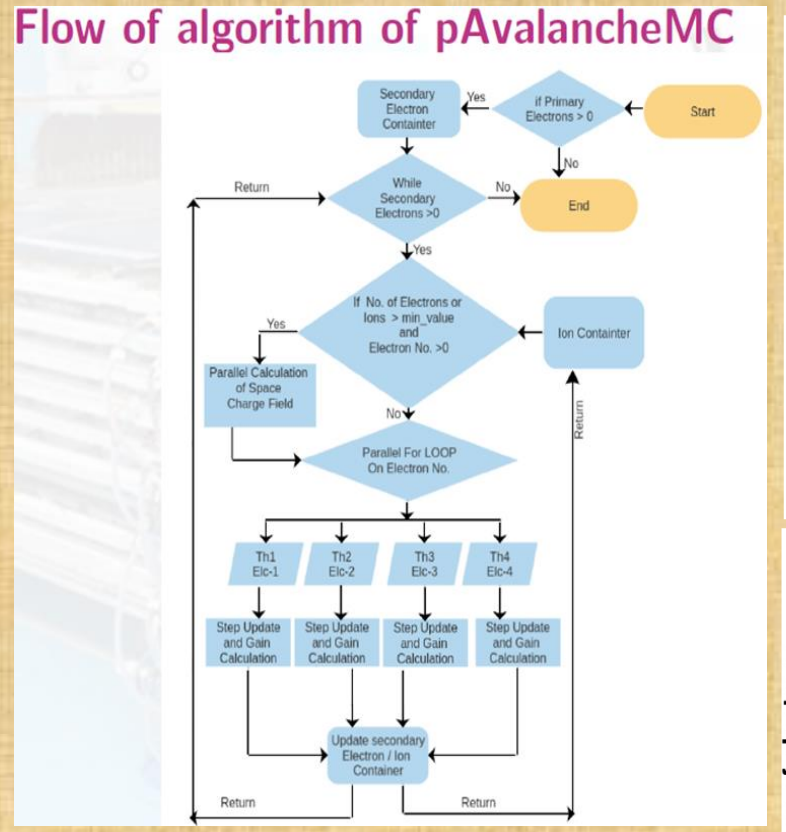



(a) Dark current in the button spacers.

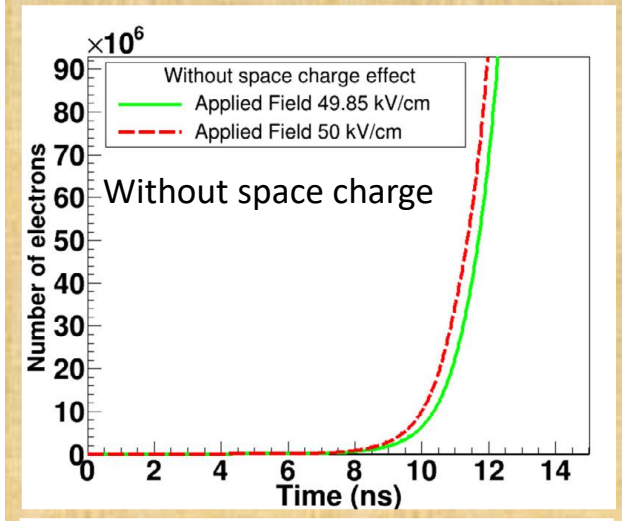
# RPC space charge using OpenMP

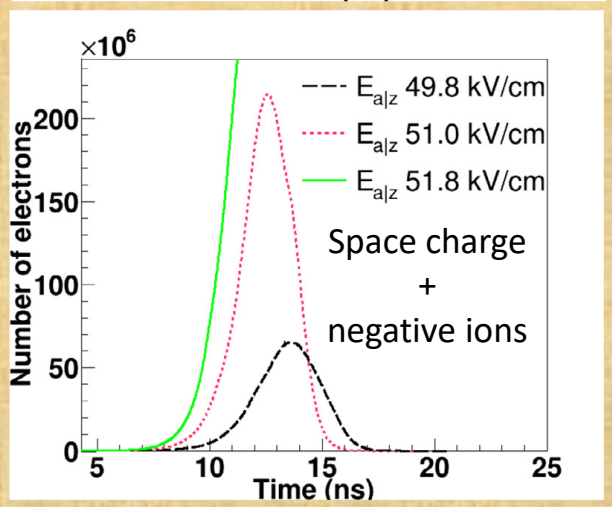
- In Garfield++, a new class has been added to the existing framework:
- `pAvalancheMC` (loosely based on class `AvalancheMC`)
- The new class contains several new functions such as
  - `SetNumberOfThreads(20)` – carries out OpenMP parallelization.
  - `SpaceChargeEffectOn()`
  - `SetMinSpCharge(1e4,0)`
  - `SetGridElements(dthta, dx, dy, dz, dr)`
  - `SetElectrodePropertise(thickness, thickness, gasgap, epsilon, true);`
  - `SetElectrodeLocations(electrode_Center1_alongz, electrode_Center2_alongz, gas_Center_alongz);`
  - `GlobalTimeWindow(time);`
- etc …
- Till now, specific to RPCs.

Parallelization of AvalancheMC




OpenMP implementation in neBEM is undergoing further improvement.



# GPU-CUDA in neBEM

- Garfield++ built using CUDA.
- OS: Ubuntu 22.04, GPU h/w: NVidia Quadro K2200, Libraries: cuda, cuBLAS.
- Impressive initial performance observed in matrix multiplications within neBEM.
- One PhD student working. Expect several interesting developments in the near future.

# Present team

- Bhattacharya, Purba
- Dey, Tanay
- Dutta, Shubhabrata
- Majumdar, Nayana
- Mukhopadhyay, Supratik
- Schindler, Heinrich
- Veenhof, Rob

Happily acknowledge support from members of the RD51 / DRD1 collaboration.

- Thanks a lot for your patience!