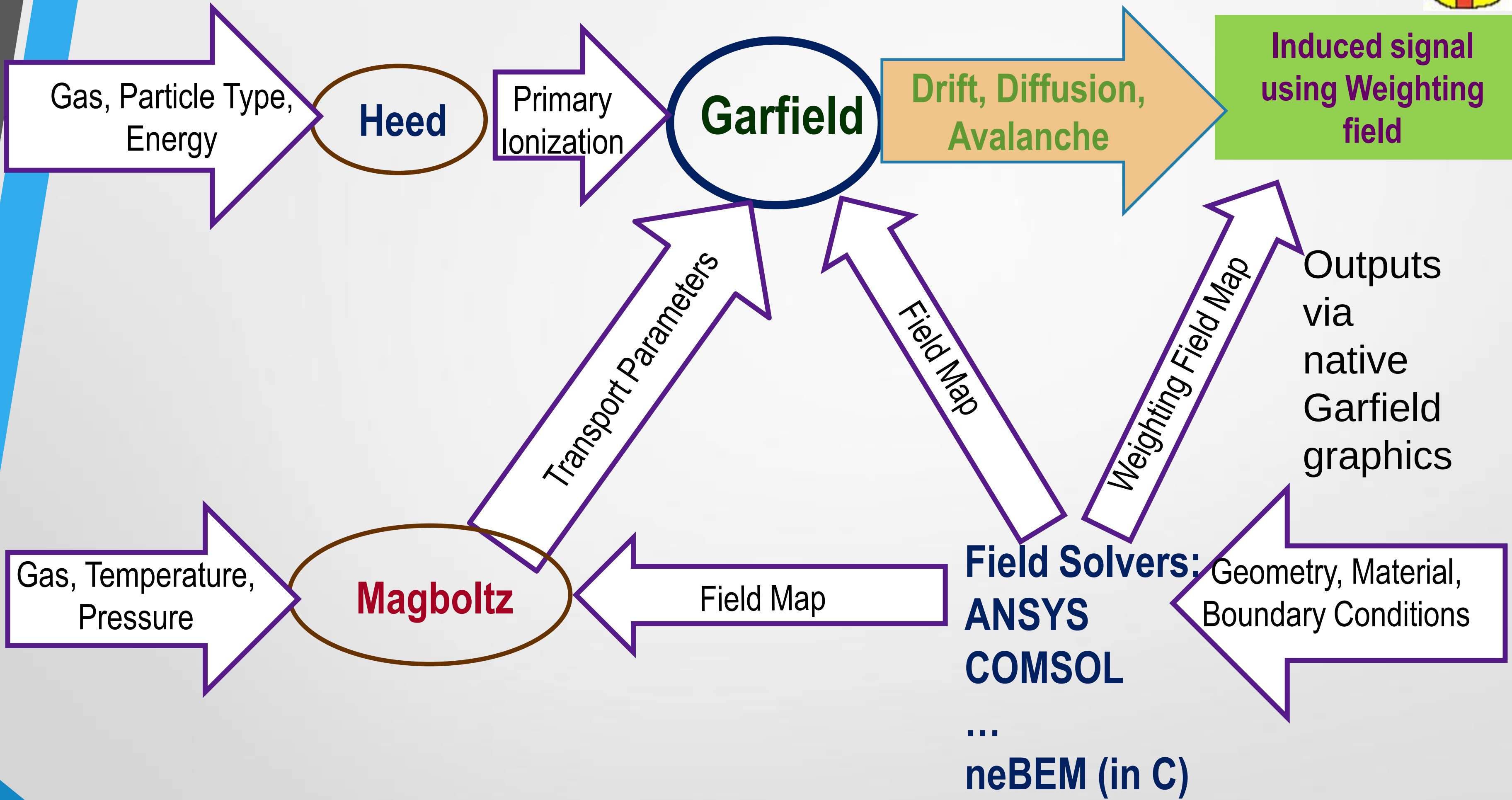
neBEM for Garfield++

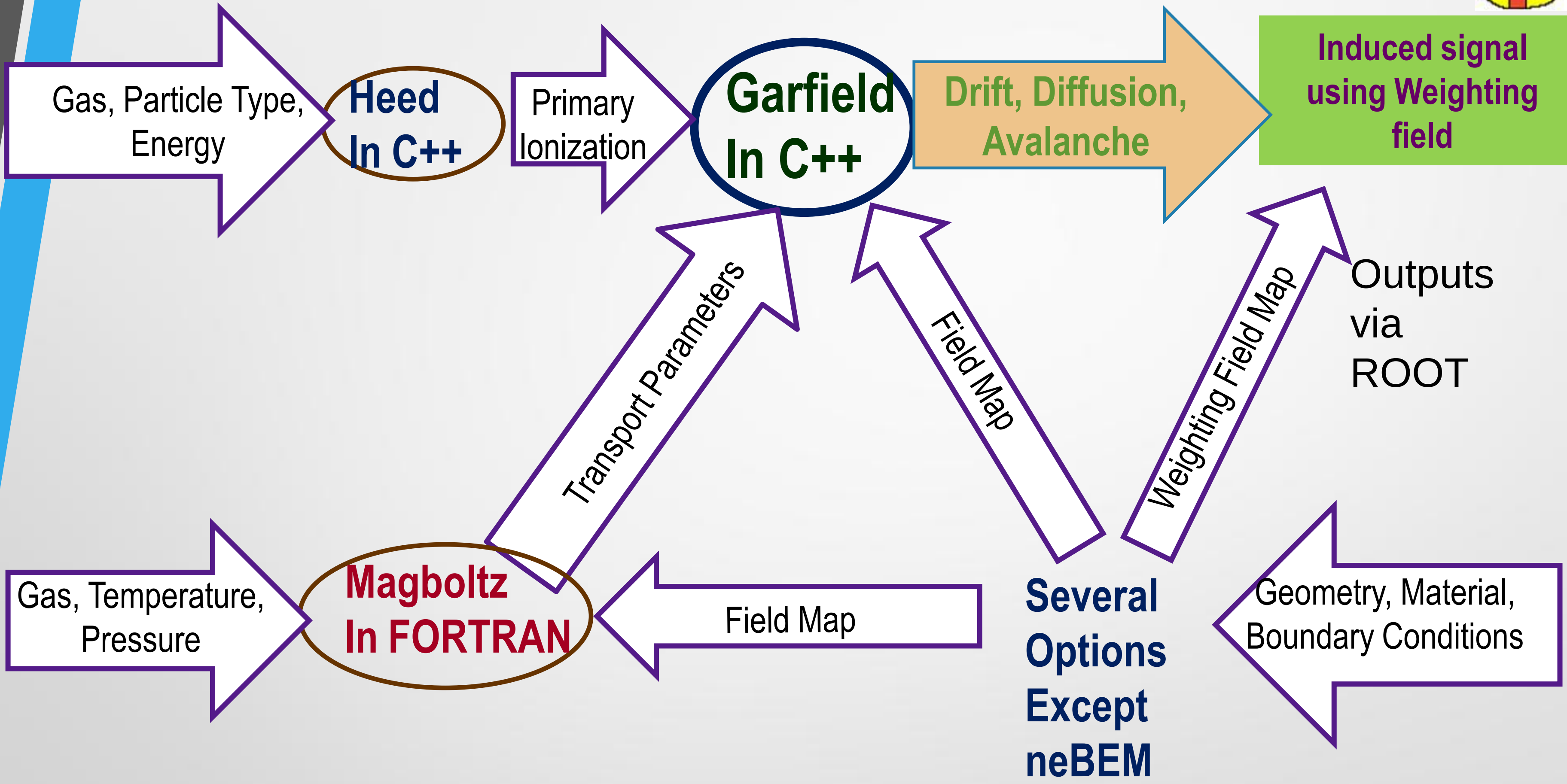
Supratik Mukhopadhyay
ANP Division, SINP

Earlier RD51 simulation framework in FORTRAN


Gas, Particle Type, Energy
Heed
Primary Ionization
Garfield
Drift, Diffusion, Avalanche
Induced signal using Weighting field
Transport Parameters
Field Map
Weighting Field Map
Outputs via native Garfield graphics
Gas, Temperature, Pressure
Magboltz
Field Map
Field Solvers: ANSYS COMSOL … neBEM (in C)
Geometry, Material, Boundary Conditions

Not only field maps which involve interpolation, neBEM could be used to evaluate field at any arbitrary location using the charge densities on various interfaces

Garfield migrates to Garfield++


Gas, Particle Type, Energy
Heed In C++
Primary Ionization
Garfield In C++
Drift, Diffusion, Avalanche
Induced signal using Weighting field
Outputs via ROOT
Transport Parameters
Field Map
Weighting Field Map
Gas, Temperature, Pressure
Magboltz In FORTRAN
Field Map
Several Options Except neBEM
Geometry, Material, Boundary Conditions

Development and Maintenance of Garfield++ (Fortran version Garfield is still available, but not actively supported):

Garfield++ is a collection of classes for the detailed simulation of small-scale detectors.

Improvements in Garfield++

Simulation Improvements:

Transport:

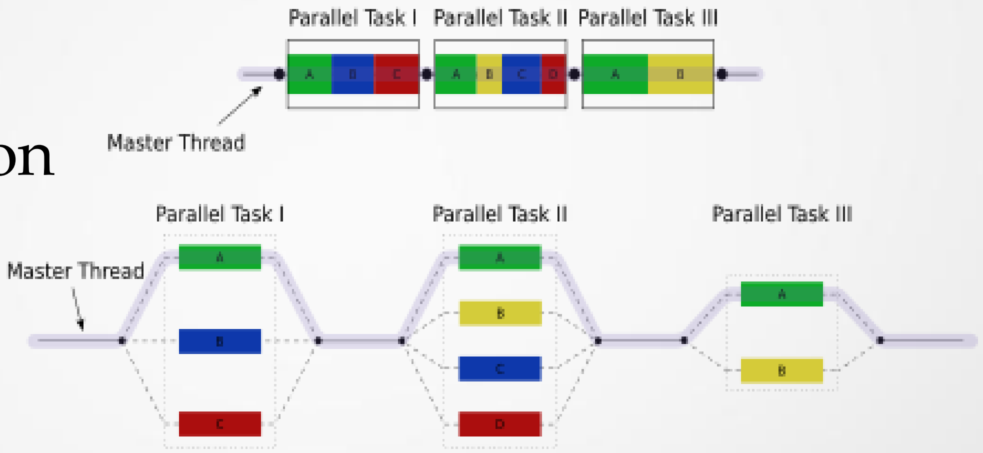
- ion mobility and diffusion, measurement and modeling
- ongoing update of electron cross sections
- e-ion recombination process in Xe
- thermal motion

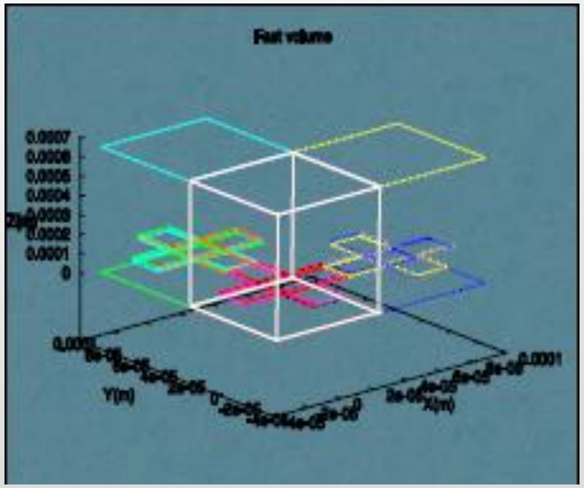
Photons:

- update in UV emission
- inclusion of IR production
- photon trapping and resulting excitation transport
- photon absorption in the gas (gas feedback)
- photon absorption in and electron emission from walls (feedback)
- photo cathodes

Improvments in neBEM

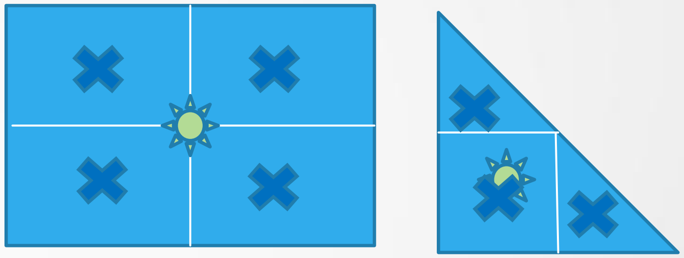
- Open Multi-Processing (OpenMP): an Application Programming Interface (API).
- Fast volume for both physical and weighting field
- Adaptive modelling

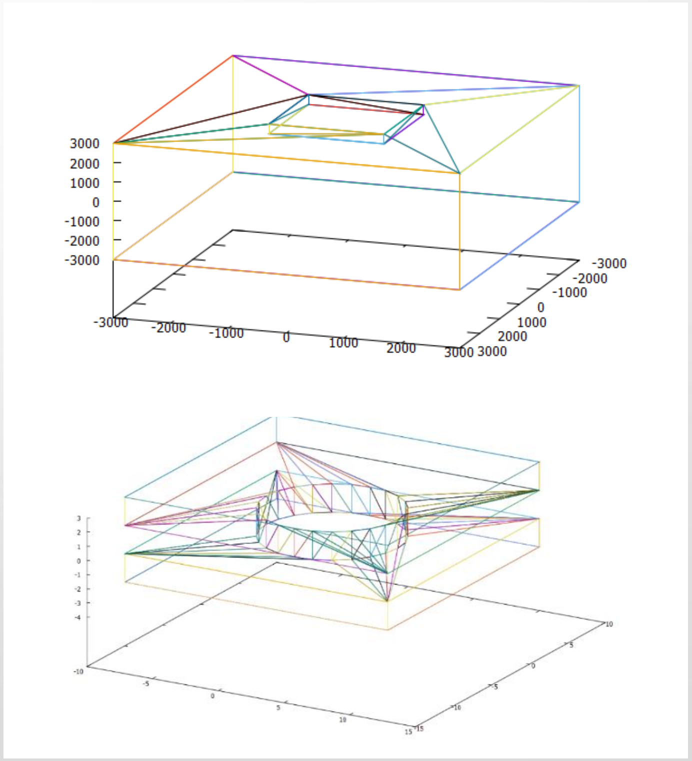




Improvements in neBEM

- Error estimation: boundary condition being evaluated at non collocation points (OptEstimateError; can lead to effective adaptive meshing)
- Charging up: electrons and ions are being assigned to elements on which they get deposited (OptChargingUp)
- Space charge: basics are ready since long
- Charge dispersion, in slow progress



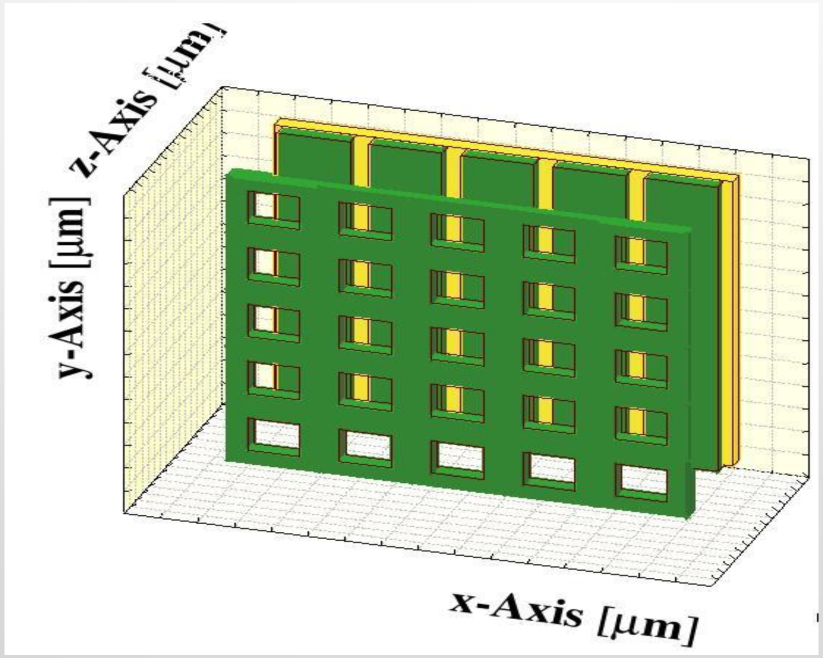


Garfield++ using neBEM

- Since September 2018, we are working on developing an interface between Garfield++ and neBEM.
- Two approaches being pursued at present.
  - neBEM direct interface to Garfield++:
    - Almost the entire geometry modeler for the FORTRAN version of Garfield has been rewritten in C++
    - neBEM will use this native geometry modeler and will be strongly coupled to Garfield++ as before.
    - ComponentNeBem3d, a new class, is being written for this purpose.
  - neBEM as a provider of field map:
    - Already implemented using Garfield (FORTRAN). Initially based on ComponentVoxel, a special class ComponentNeBem3dMap has been written for this purpose and is working. It, naturally, needs further improvements.
    - Being pursued using Geant4

neBEM requirments

- neBEM computes the charge density at various interfaces that leads to the satisfaction of a given set of boundary conditions.
  - needs the geometrical and material properties of the interfaces.
  - needs the boundary conditions at the interfaces.



Direct interface to Garfield++

- Garfield++ provides neBEM with
  - List of existing volumes and their properties
  - Interfaces among these volumes where charge densities will appear through the application of potential.
    - To be supplied as non-overlapping rectangles or right-triangles.
    - Should contain boundary conditions, material properties etc.
  - Discretization requirements
  - Repetition information (periodicity)
  - Computational specifications (solution method to be adopted, number of cores to be used, adaptive meshing requirements and so on)

Direct interface to Garfield++

- A new class (`ComponentNeBem3d`) is being introduced.
- Instantiation of an object of this class and its initialization will prompt
  - Setting up of primitives, elements, boundary conditions
  - Computation of influence matrix
  - Computation of charge densities on each element
- After the charge densities are computed, electric field can be computed at any desired location.

- The development is at an early stage but the crucial phase of identifying the overall design is complete.
- Successful completion of this approach will need some serious coding efforts in the coming few months.

neBEM as a field mapper

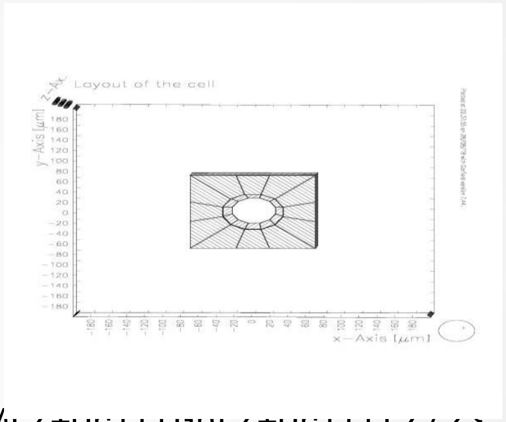
- In order to achieve computational efficiency, neBEM has since long been used as field mapper to Garfield (FORTRAN): `FastVolume` provides field map on a 3D rectangular grid
- Garfield++ has a class (`ComponentVoxel`) that can use a field-map on a 3D rectangular mesh.
- It has been possible to combine these two so that voxelized field map generated using neBEM can be used by Garfield++ for its computations.
- As a further improvement along this line, a special class, `ComponentNeBem3dMap`, has been implemented and is working fine.

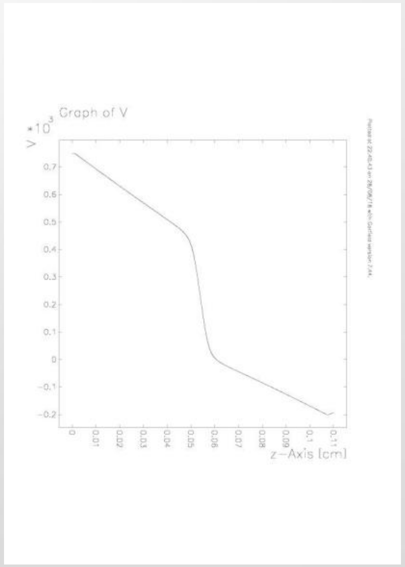
neBEM as field-mapper

- Use FORTRAN Garfield script along with neBEM to generated a field map.
- Use ComponentNeBem3dMap class to read this field map into Garfield++
- Carry out device simulation

Garfield script with neBEM

```
global DriftVolt = -200.0
global TopCuVolt = 0.0
global BotCuVolt = 450.0
global IndVolt = 750.0

&CELL

nebem target-element-size 0.0010

solids
box centre {OrigX} {OrigY} {OrigZ+IndLZ+IndGpLZ+GEMLZ+DriftGpLZ+DriftLZ/2}
...
        half-lengths {GemLX/2.} {GemLY/2.} {DriftLZ/2.} ...
        conductor-1 ...
        voltage {DriftVolt}
box centre {OrigX} {OrigY} {OrigZ+IndLZ/2} ...
        half-lengths {GemLX/2.} {GemLY/2.} {IndLZ/2.} ...
        conductor-1 ...
        voltage {IndVolt}
…
period x = 0.0140
period y = 0.0140
nebem  lu-inversion ...
       min-elem 1 max-elem 11 ...
       x-periodic-copies 10 ...
       y-periodic-copies 10 ...
       keep-inverted-matrix ...
       new-model
```





MapInfo file

- 0.1 => Map version
- 1 => OptMap
- 0 => OptStaggerMap
- 29 => nX
- 29 => nY
- 201 => nZ
- -7.000000e-03 7.000000e-03 => xMin and xMax
- -7.000000e-03 7.000000e-03 => yMin and yMax
- 0.000000e+00 1.060000e-01 => zMin and zMax
- 0.000000e+00 => xStagger
- 0.000000e+00 => yStagger
- 0.000000e+00 => zStagger
- MapFPR.out => Field map data file

Field map generated by neBEM

```
# X(cm)    Y(cm)     Z(cm)      FX(V/cm) FY(V/cm) FZ(V/cm) Pot(V)      Region
-0.007     -0.007    0       -59.699737       -62.793173       -3680.567  749.94216    2
-0.007     -0.007    0.00053    -59.292323       -60.249582       -6.1935511        749.93135    2
-0.007     -0.007    0.00106    -58.907324       -60.102944       6561.2783 749.54879    0
-0.007     -0.007    0.00159    -58.513258       -59.342215       6517.8987 746.08165    0
                                          …
                                          …
-0.007     -0.007    0.05088    67.514592 6.4217114 4502.7969 450.57853    0
-0.007     -0.007    0.05141    310.60821 -22.552155       -169.91559        450.07853    3
-0.007     -0.007    0.05194    -57.119348       23.024326 90184.929 410.37054    5
-0.007     -0.007    0.05247    -129.30656       38.688416 90029.226 362.61138    5
-0.007     -0.007    0.053      -95.699053       34.574837 89935.345 314.91835    5
-0.007     -0.007    0.05353    -43.257569       19.450469 89888.944 267.26113    5
-0.007     -0.007    0.05406    10.460695 -0.39916935      89879.216 219.61884    4
-0.007     -0.007    0.05459    64.139432 -19.673695       89903.112 171.97272    4
-0.007     -0.007    0.05512    114.78391 -32.75224  89964.131 124.30394    4
-0.007     -0.007    0.05565    136.4748  -33.245928       90073.575 76.591848    4
-0.007     -0.007    0.05618    7.3069507 -10.813104       90239.718 28.805869    4
-0.007     -0.007    0.05671    -145.68192       12.256835 -99.116446        -0.069678531        6
-0.007     -0.007    0.05724    -21.765962       4.186861  2414 -0.63534521         0
```

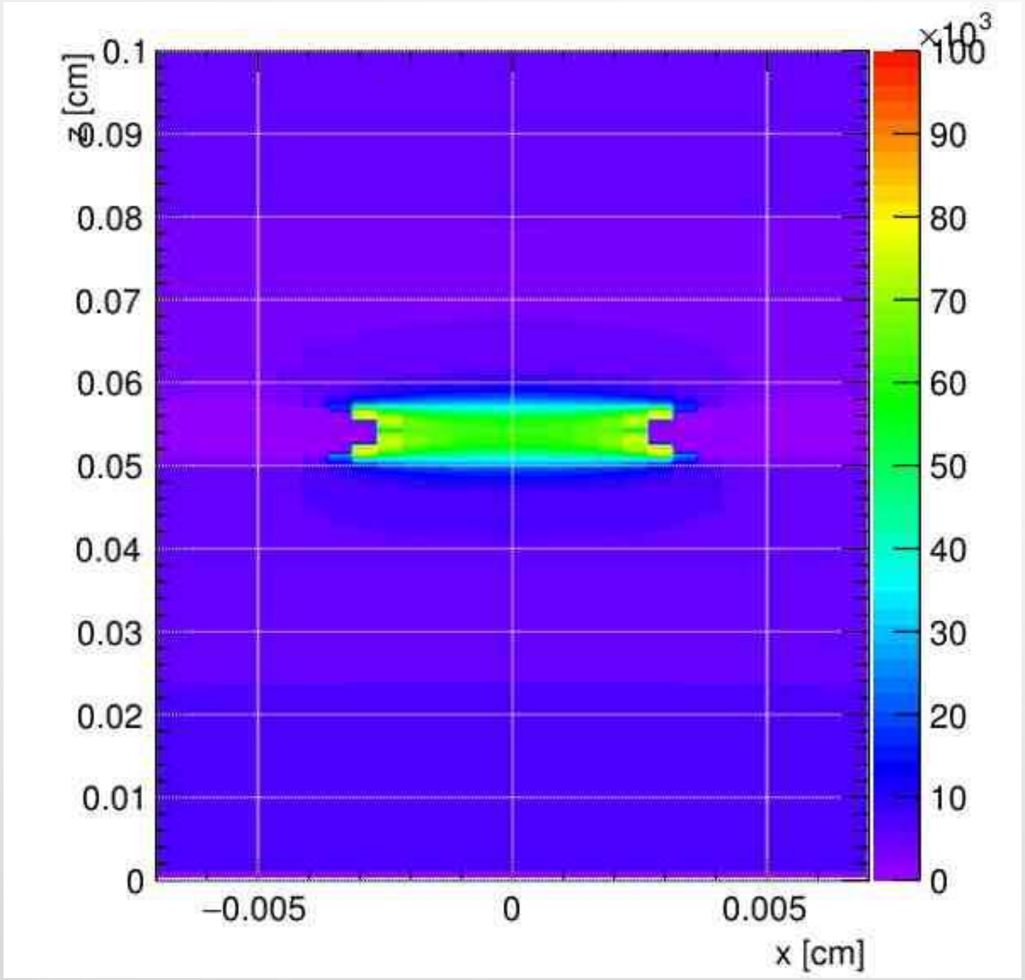
ComponentNeBem3dMap to read field-map

```
ComponentNeBem3dMap* eField = new
ComponentNeBem3dMap();
eField->LoadMapInfo("MapInfo.out", version,
optmap, optstgrmap, nX, nY, nZ, xMin, xMax,
yMin, yMax, zMin, zMax,
xStgr, yStgr, zStgr, mapfile);

eField->SetMesh(nX, nY, nZ, xMin, xMax, yMin,
yMax, zMin, zMax);

eField->LoadElectricField(mapfile, "XYZ", true,
true, 1.0, 1.0, 1.0);
```

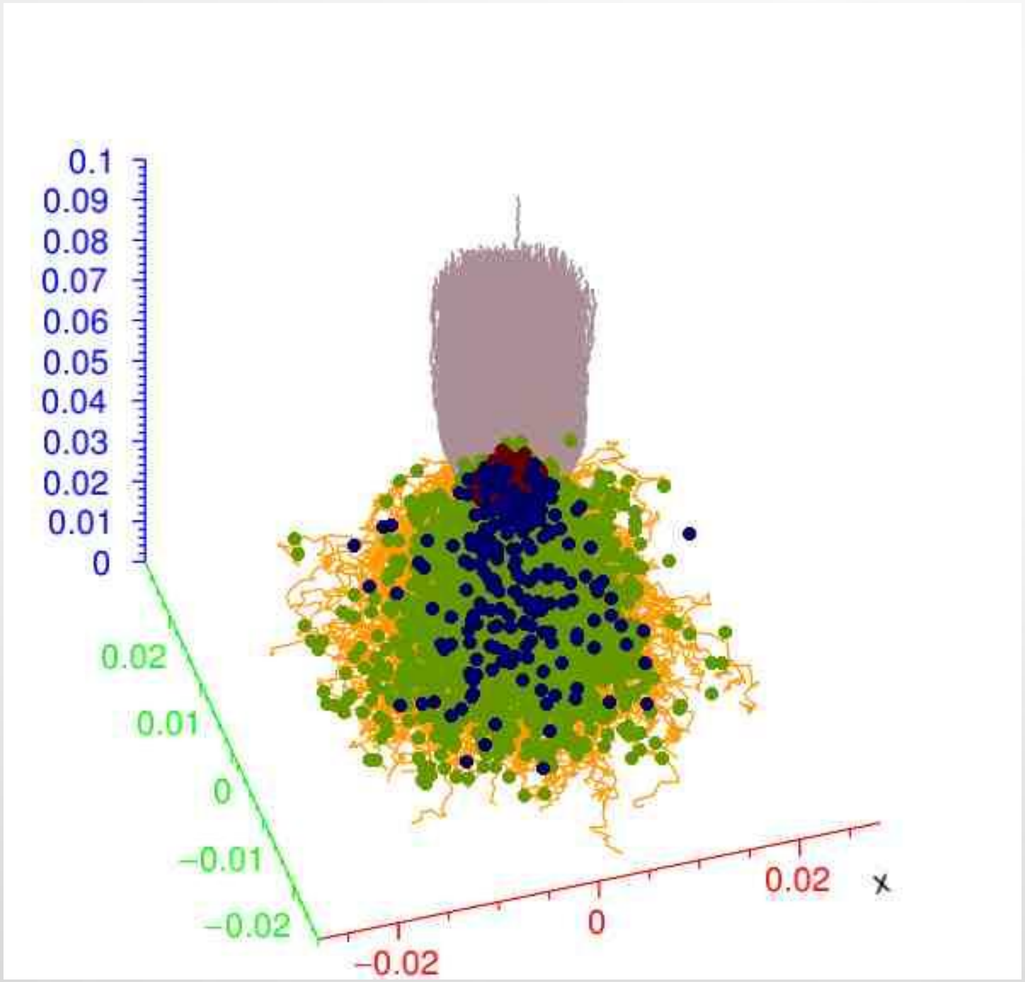
Device simulation

```
// Create the sensor.
 Sensor* sensor = new Sensor()
…

AvalancheMicroscopic* aval = new AvalancheMicroscopic();
  aval->SetSensor(sensor);
  aval->EnableAvalancheSizeLimit(1000);

  AvalancheMC* drift = new AvalancheMC();
  drift->SetSensor(sensor);
  drift->SetDistanceSteps(2.e-4);
…
 for (int i = nEvents; i--;) {
    std::cout << "\nEVENT: " << i<<"\n";
…
 aval->AvalancheElectron(x0, y0, z0, t0, e0, 0., 0.,
0.);
…
```

Field map result



Avalanche in GEM



neBEM field mapping using Geant

- This could be interesting because the community is well-conversant with Geant4 and neBEM can easily act as an electric field solver for a large number of 3D geometries
- Discussions going on – we are hopeful …

Summary

- Direct integration of neBEM into Garfield++ is ongoing
    - Design details are in place
    - Major coding efforts in the coming months
- neBEM as a field-map provider to Garfield++ is already successful
    - Minor improvements necessary in very near future
    - Major improvements in computational (speed and storage) expected in the long term
- neBEM as a field-mapper using more elaborate geometry modelers seems to be on the way,

The SINP TEAM

- Deb Sankar Bhattacharya (Wurzberg)
- Purba Bhattacharya (Weizzmann)
- Sudeb Bhattacharya (Retd)
- Jaydeep Dutta
- Abhik Jash (NISER)
- Anil Kumar
- Vishal Kumar
- Nayana Majumdar
- Supratik Mukhopadhyay
- Prasant Kumar Rout
- Promita Roy
- Mohammed Salim (AMU)
- Sandip Sarkar
- Muzamil Ahmad Teli (KU)
- Sridhar Tripathy

Acknowledgments

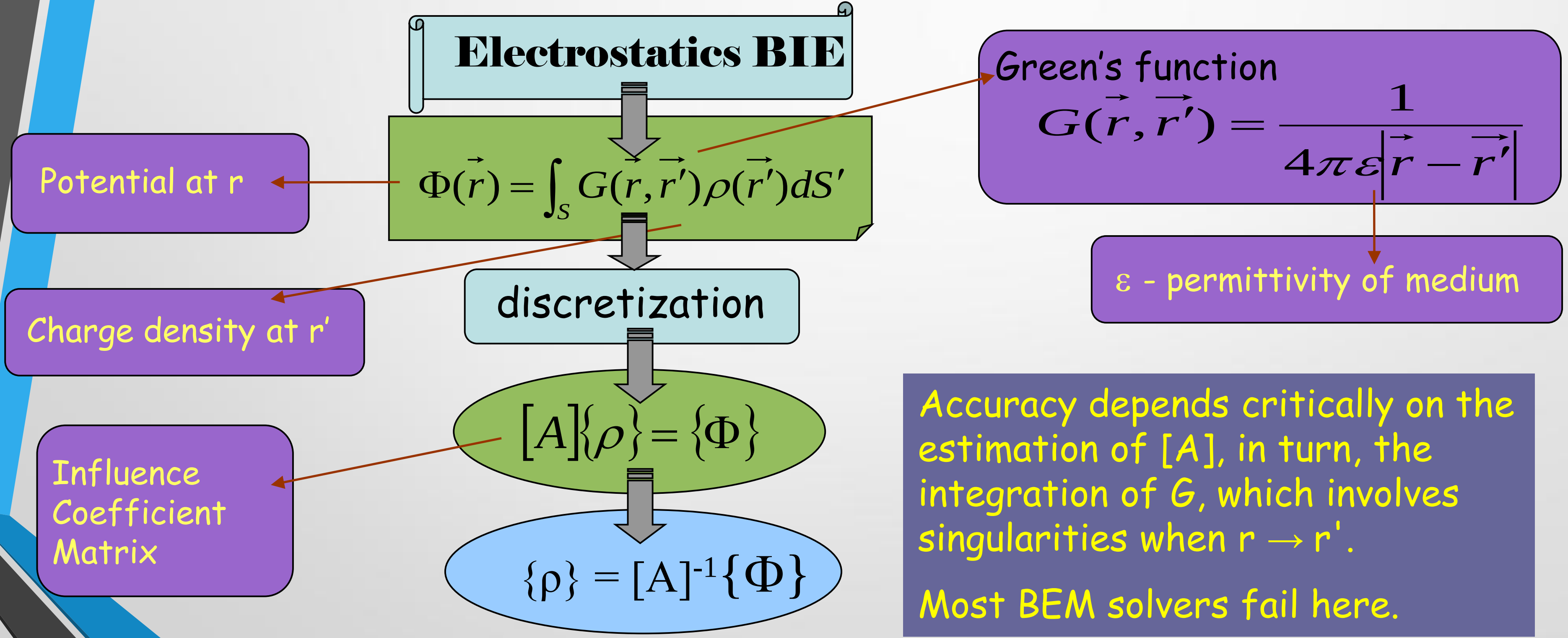
- Rob and Heinrich
- Leszek and the RD51 team

Thank you
for your kind attention!

Backup Slides

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.



$$\Phi(\vec{r}) = \int_S G(\vec{r},\vec{r}')\rho(\vec{r}')dS'$$

$$G(\vec{r},\vec{r}') = \frac{1}{4\pi\varepsilon|\vec{r}-\vec{r}'|}$$

$$[A]\{\rho\} = \{\Phi\}$$

$$\{\rho\} = [A]^{-1}\{\Phi\}$$

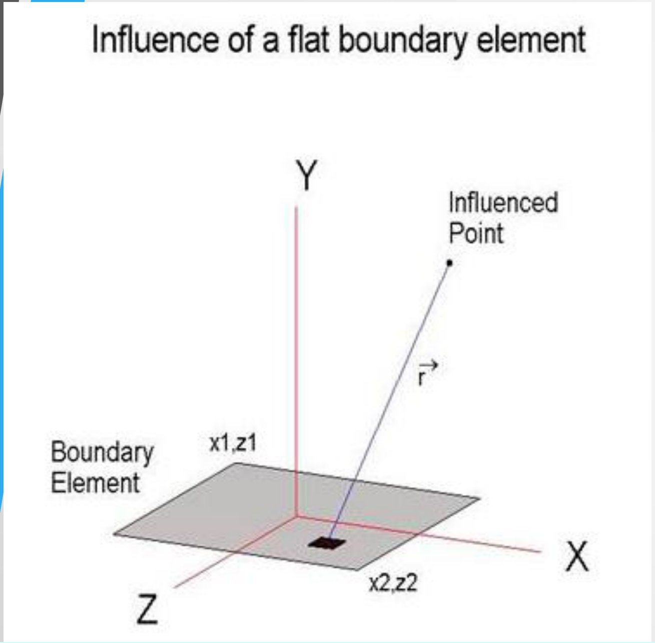
Accuracy depends critically on the estimation of [A], in turn, the integration of G, which involves singularities when $r \rightarrow r'$.

Most BEM solvers fail here.

Foundation expressions of ISLES

Inverse Square Law Exact Solutions

Rectangular element



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

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

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

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

nearly exact Boundary Element Method (neBEM)

A formulation based on green's function that allows the use of exact close-form analytic expressions while solving 3d problems governed by Poisson's equation. It is very precise even in critical near-field regions, and microscopic length scale.

It is easy to use, interface and integrate neBEM

Stand-alone
A driver routine
An interface routine
Post-processing, if necessary

Garfield
Garfield prompt
Garfield script

Charge density at all the interfaces

Potential at any arbitrary point

Field at any arbitrary point

Capacitance, forces on device components properties can be obtained by post-processing

neBEM@CERN

http://nebem.web.cern.ch