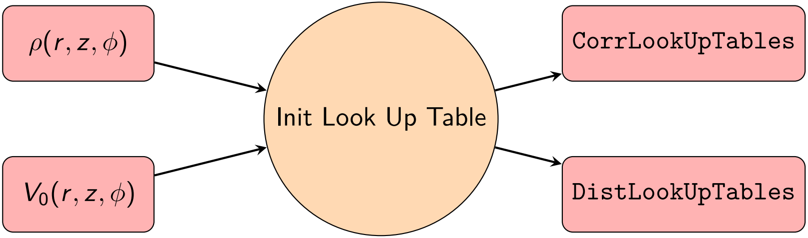
Correction and Distortion look-up tables by following the electron drift, ATO-112

Rifki Sadikin, Jens Wiechula, Marian Ivanov, Kai Schweda

November 24, 2015

Table of contents

Correction/Distortion Framework



- `GetCorrection`$(x[], roc, dx[]) \leftarrow$ `CorrLookUpTables`
- `GetDistortion`$(x[], roc, dx[]) \leftarrow$ `DistLookUpTables`

Distortion Algorithm Sketch

1. $\nabla^2 V(r, z, \rho) = -\frac{1}{\epsilon_0}\rho(r, \phi, z)$

↓

2. $\vec{E}(r, \rho, z) = -\nabla V(r, \rho, z)$

↓

3. $$\hat{\delta}_{rE}(r_i, \phi_j, z_k) = c_1 \int_{z_k}^{z_{k-1}} \frac{E_r}{E_z} dz + c_2 \int_{z_k}^{z_{k-1}} \frac{E_\phi}{E_z} dz$$
$$r\hat{\delta}_{\phi rE}(r_i, \phi_j, z_k) = c_2 \int_{z_k}^{z_{k-1}} \frac{E_r}{E_z} dz - c_1 \int_{z_k}^{z_{k-1}} \frac{E_\phi}{E_z} dz$$
$$\hat{\delta}_z(r_i, \phi_j, z_k) = \int_{z_k}^{z_{k-1}} \frac{v'(E)}{v_0}(E - E_0) dz$$

↓

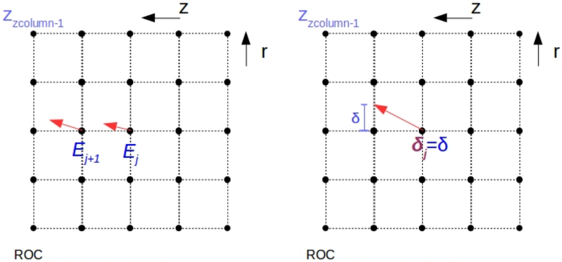
4. Follow the driftline from $(r_i, \phi_j, z_k) \rightarrow (r_i + \delta r_i, \phi_j + \delta\phi_j, z_0 + \delta z_k)$

↓

5. Assign distortion $\mathrm{Dist}(r_i, \phi_j, z_k) = (\delta r, \delta phi, \delta z)$

Local Distortion

- Local distortion is calculated based on formulation in ALICE-INT-2010-016 (Simplified Langevin Eq)
- This integration is in z direction. Let suppose we want to calculate local distortion at (r_i, z_j, ϕ_m), we require $\vec{E}(r_i, z_{j+1}, \phi_m)$ and $\vec{E}(r_i, z_j, \phi_m)$ are known.



(a) Electric Field Vectors (b) Local Distortion

Local Distortion (2)

Integration

We can calculate definite integrations for each distortions respect of z from z_j to z_{j+1} as follows:

$$\int_{z_j}^{z_{j+1}} \frac{E_r}{E_z}(r_i, z_j, \phi_m)dz \approx \frac{-1}{\mathrm{ezField}} \frac{h_z}{2.0} \left(E_r(r_i, z_j, \phi_m) + E_r(r_i, z_{j+1}, \phi_m)\right)$$

$$\int_{z_j}^{z_{j+1}} \frac{E_\phi}{E_z}(r_i, z_j, \phi_m)dz \approx \frac{-1}{\mathrm{ezField}} \frac{h_z}{2.0} \left(E_\phi(r_i, z_j, \phi_m) + E_\phi(r_i, z_{j+1}, \phi_m)\right)$$

$$\int_{z_j}^{z_{j+1}} E_z(r_i, z_j, \phi_m)dz \approx \frac{h_z}{2.0} \left(E_z(r_i, z_j, \phi_m) + E_z(r_i, z_{j+1}, \phi_m)\right)$$

The code sniplet:

```
// = Electric Field (V/cm) Magnitude ~ -400 V/cm;
Double_t ezField = (fgkCathodeV-fgkGG)/fgkTPCZ0;
localIntErOverEz = (gridSizeZ/2.0)*((*eR)(i,j)+(*eR)(i,j+1))/(-1*ezField) ;
localIntEphiOverEz = (gridSizeZ/2.0)*((*ePhi)(i,j)+(*ePhi)(i,j+1))/(-1*ezField) ;
localIntDeltaEz = (gridSizeZ/2.0)*((*eZ)(i,j)+(*eZ)(i,j+1)) ;
```

CERN ALICE LIPI

Local Distortion (3)

After we have local integrations for ellectric fields in each direction, local distortion $\hat{\delta}(r_i, z_j, \phi_m)$ is calculated by simplified Langevin equation:

$$\hat{\delta}_{rE}(r_i, z_j, \phi_m) = c_0 \int_{z_j}^{z_{j+1}} \frac{E_r}{E_z} dz + c_1 \int_{z_j}^{z_{j+1}} \frac{E_\phi}{E_z} dz$$

$$r\hat{\delta}_{\phi E}(r_i, z_j, \phi_m) = -c_1 \int_{z_j}^{z_{j+1}} \frac{E_j}{E_j} dz + c_0 \int_{j_j}^{z_{j+1}} \frac{E_\phi}{E_z} dz$$

$$\hat{\delta}_z(r_i, z_j, \phi_m) = \int_{z_j}^{z_{j+1}} \frac{v'(E)}{v_0}(E - E_0) dz$$

Where c_0 and c_1 are constants (see the ALICE-INT-2010-016 for further details).

```
(*distDrDz)(i,j) = fC0*localIntErOverEz   + fC1*localIntEphiOverEz;
(*distDphiRDz)(i,j) = fC0*localIntEphiOverEz - fC1*localIntErOverEz ;
(*distDz)(i,j) = localIntDeltaEz*fgkdvdE*fgkdvdE;
```

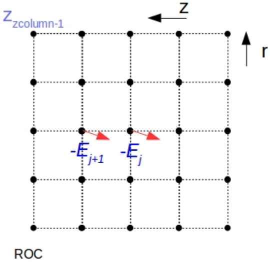
Local Correction

Let suppose we want to calculate local correction at $(r_i, \mathbf{z_{j+1}}, \phi_m)$. Then we can calculate definite integrations for each directions in respect of z from z_{j+1} to z_j as follows:

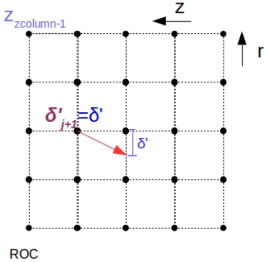
$$\int_{z_{j+1}}^{z_j} \frac{E_r}{E_z}(r_i, z_j, \phi_m)dz \approx -1 * \frac{-1}{\text{ezField}} \frac{h_z}{2.0} \left(E_r(r_i, z_j, \phi_m) + E_r(r_i, z_{j+1}, \phi_m) \right)$$

$$\int_{z_{j+1}}^{z_j} \frac{E_\phi}{E_z}(r_i, z_j, \phi_m)dz \approx -1 * \frac{-1}{\text{ezField}} \frac{h_z}{2.0} \left(E_\phi(r_i, z_j, \phi_m) + E_\phi(r_i, z_{j+1}, \phi_m) \right)$$

$$\int_{z_{j+1}}^{z_j} E_z(r_i, z_j, \phi_m)dz \approx -1 * \frac{h_z}{2.0} \left(E_z(r_i, z_j, \phi_m) + E_z(r_i, z_{j+1}, \phi_m) \right)$$



(a) Opposite Electric Field Vectors



(b) Local Correction

Local Correction (2)

- Local correction at $\hat{\delta}'(r_i, \mathbf{z_{j+1}}, \phi_m)$ is calculated by simplified Langevin equation:

$$\hat{\delta}'_{rE}(r_i, z_{j+1}, \phi_m) = c_0 \int_{z_{j+1}}^{z_j} \frac{E_r}{E_z} dz + c_1 \int_{z_{j+1}}^{z_j} \frac{E_\phi}{E_z} dz$$

$$r\hat{\delta}'_{\phi E}(r_i, z_{j+1}, \phi_m) = -c_1 \int_{z_{j+1}}^{z_j} \frac{E_j}{E_j} dz + c_0 \int_{z_{j+1}}^{z_j} \frac{E_\phi}{E_z} dz$$

$$\hat{\delta}'_{z}(r_i, z_{j+1}, \phi_m) = \int_{z_{j+1}}^{z_j} \frac{v'(E)}{v_0} (E - E_0) dz$$

- For implementation, we use the fact that

$$\hat{\delta}'_{rE}(r_i, z_{j+1}, \phi_m) = -1 * \hat{\delta}_{rE}(r_i, z_j, \phi_m)$$

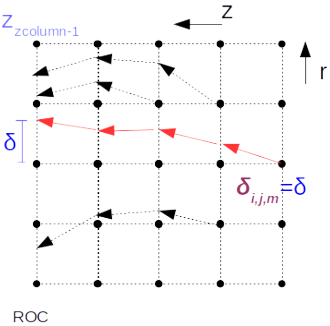
$$r\hat{\delta}'_{\phi E}(r_i, z_{j+1}, \phi_m) = -1 * r\hat{\delta}_{\phi E}(r_i, z_j, \phi_m)$$

$$\hat{\delta}'_{z}(r_i, z_{j+1}, \phi_m) = -1 * \hat{\delta}_{z}(r_i, z_j, \phi_m)$$

```
(*corrDrDz)(i,j+1) = -1* (*distDrDz)(i,j) ;
(*corrDphiRDz)(i,j+1) = -1* (*distDphiRDz)(i,j);
(*corrDz)(i,j+1) = -1* (*distDz)(i,j);
```

CERN ALICE LIPI

Global Distortion/Following the driftline



(a) Global Distortion

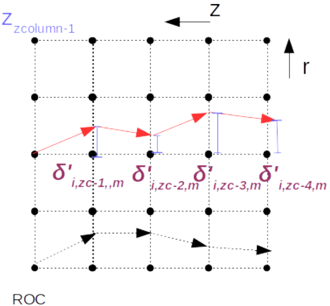
Interpolation is used for computing intermediate local distortion.

Global Distortion

Algorithm 1 Global Distortion

```
for m = 0 to PhiSlice-1 do
    for j = 0 to Zcolumn-2 do
        for i = 0to i= Row-1 do
            δ_rE(r_i, φ_m, z_j) = 0, δ_φrE(r_i, φ_m, z_j) = 0
            δ_z(r_i, φ_m, z_j) = 0, r = r_i, φ = φ_m, z = z_j
            for jj = j to Zcolumn-2 do
                (δ̂_rE, δ̂_φrE, δ̂_z) = interpolate(r, z, φ)
                φ+ = δ̂_rφE/r, r+ = δ̂_rE z = z_{jj-1} + δ̂_z
                δ_rE(r_i, φ_m, z_j)+ = δ̂_rE
                δ_φrE(r_i, φ_m, z_j)+ = δ̂_φrE
                δ_z(r_i, φ_m, z_j)+ = δ̂_z
            end for
        end for
    end for
end for
```

Global Correction/Following the driftline



(b) Global Correction

Interpolation is used for computing intermediate local correction.

Global Correction

Algorithm 2 Global Correction

for j = Zcolumn - 2 **to** 0 **do**
  for m = 0 **to** PhiSlice-1 **do**
    for i = 0 **to** i= Row-1 **do**
      $\phi = \phi_m + \delta'_{\phi rE}(r_i, \phi_m, z_{j-1})/r$
      $r = r_i + \delta'_{rE}(r_i, \phi_m, z_{j-1})$
      $z = z_{j-1} + \delta'_z(r_i, \phi_m, z_{j-1})$
      $(\hat{\delta}'_{rE}, \hat{\delta}'_{\phi rE}, \hat{\delta}'_z) = \texttt{interpolate}(r, \phi, z, \hat{\delta}')$
      $\delta'_{rE}(r_i, \phi_m, z_j) += \hat{\delta}'_{rE}$
      $\delta'_{\phi rE}(r_i, \phi_m, z_j) += \hat{\delta}'_{\phi rE}$
      $\delta'_z(r_i, \phi_m, z_j) += \hat{\delta}'_z$
    end for
  end for
end for

Computation Cost of Naive Algorithm

	Time in s			
Size	$(17)^2 + 18$	$(33)^2 + 2 * 18$	$(65)^2 + 3 * 18$	$(129)^2 + 4 * 18$
Poisson Solver	0.01	0.13	2.49	59.85
Electric Field Calc	0.00	0.00	0.01	0.13
Local Dist/Corr	0.00	0.01	0.03	0.25
Global Dist/Corr	0.04	0.59	8.65	139.26
Lookup Table Fill	0.74	0.76	0.74	0.84

Global Distortion

Since $\delta'(r_i, z_j, \phi_m) = -\delta(r_i, z_j, \phi_m)$, we could use Global Correction for interpolation in Global Distortion. So, we remove one inner loop in Global distortion

Algorithm 3 Global Distortion

for j = ZColumn-2 **to** 0 **do**
  for m = 0 **to** PhiSlice-1 **do**
    for i = 0 **to** i= Rrow-1 **do**
      $r = r_i, \phi = \phi_m, z = z_{j-1}$
      $(\delta_{rE}, \delta_{\phi rE}, \delta_z) = -\texttt{interpolate}(r, \phi, z, \delta\prime)$
      $(\hat{\delta}_{rE}, \hat{\delta}_{\phi rE}, \hat{\delta}_z) = \texttt{interpolate}(r + \delta_{rE}, \phi + \delta_{\phi rE}/r, z + \delta_z, \hat{\delta})$
      $\delta_{rE}(r_i, \phi_m, z_j) += \hat{\delta}'_{rE} \ \delta_{\phi rE}(r_i, \phi_m, z_j) += \hat{\delta}'_{\phi rE} \ \delta_z(r_i, \phi_m, z_j) += \hat{\delta}'_z$
    end for
  end for
end for

Computation Cost of kUseInterpolation Algorithm

	Time in s			
Size	$(17)^2 + 18$	$(33)^2 + 2 * 18$	$(65)^2 + 3 * 18$	$(129)^2 + 4 * 18$
Poisson Solver	0.01	0.13	2.49	59.85
Electric Field Calc	0.00	0.00	0.01	0.13
Local Dist/Corr	0.00	0.01	0.03	0.25
Global Dist/Corr	0.04 (**0.01**)	0.59 (**0.04**)	8.65 (**0.38**)	139.26 (**4.18**)
Lookup Table Fill	0.74	0.76	0.74	0.84

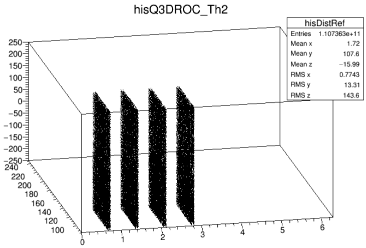
Unit Test for Distortion-Correction

- Generate points for testing in the volume $(\texttt{xlist}, \texttt{ylist}, \texttt{zlist})$
- For each points (x_i, y_j, z_k), do the following:
  1. $(\delta x_{\texttt{dist}}, \delta y_{\texttt{dist}}, \delta z_{\texttt{dist}}) = \texttt{GetDistortion}(x_i, y_j, z_k)$
  2. $(x'_i = x_i + \delta x_{\texttt{dist}}, y'_j = y_j + \delta y_{\texttt{dist}}, z'_k = z_k + \delta z_{\texttt{dist}})$
  3. $(\delta x_{\texttt{corr}} \delta y_{\texttt{corr}}, \delta z_{\texttt{corr}}) = \texttt{GetCorrection}(x'_i, y'_j, z'_k)$
  4. $(x''_i = x'_i + \delta x_{\texttt{corr}}, y''_j = y'_j + \delta y_{\texttt{corr}}, z''_k = z'_k + \delta z_{\texttt{corr}})$
  5. $\texttt{errDistCorr}_{i,j,k} = \frac{\sqrt{(x_i - x''_i)^2 + (y_j - y''_j)^2 (z_k - z''_k)^2}}{3}$
- We also collected individual difference in $r, r\phi, z$ direction.

Experiment 1

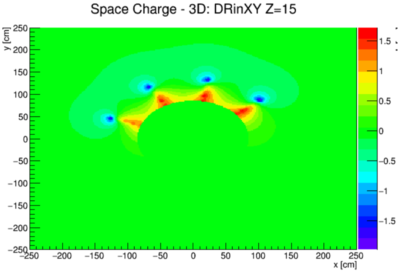
Input

- Space charge distribution as follows:

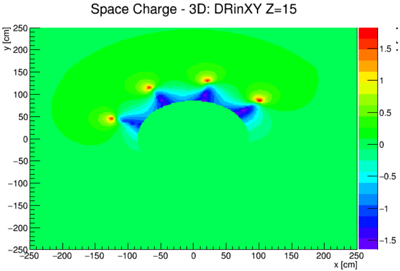


- Boundary values zeros
- Granularity: $129 \times 129 \times 144$

Distortion and Correction Map (1)

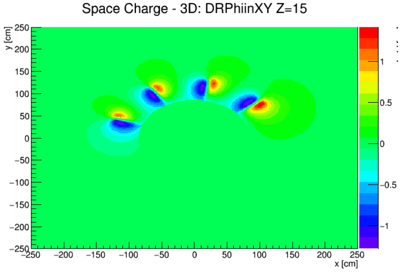


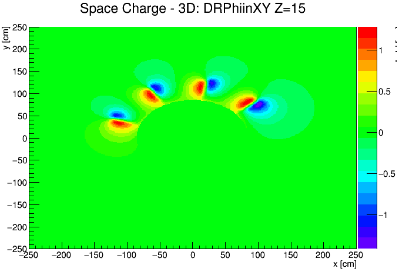
Distortion Map



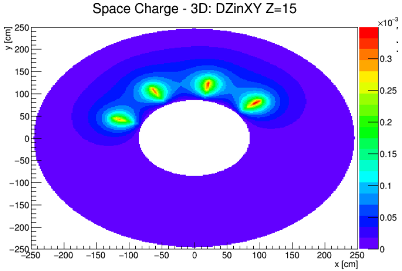
Correction Map

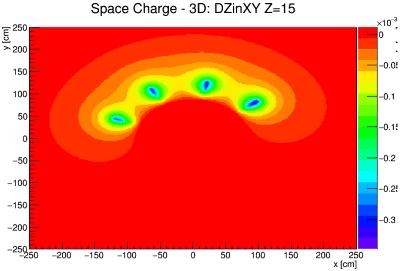
Distortion and Correction Map (2)



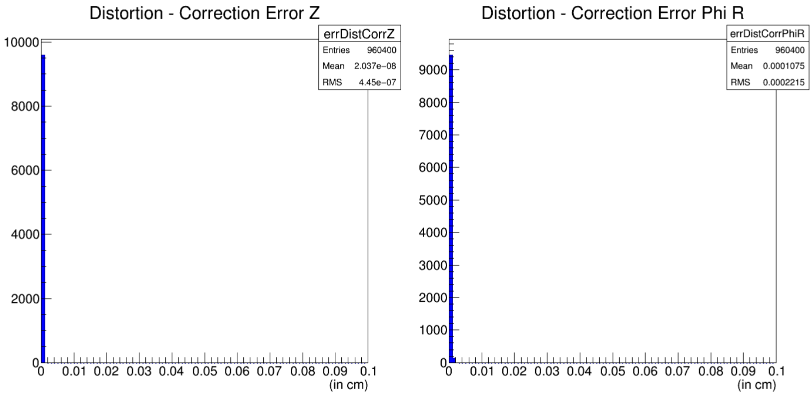


Distortion and Correction Map (3)

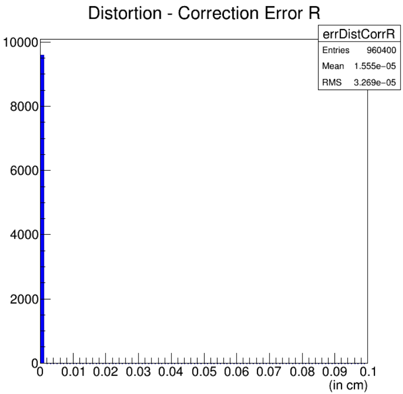




Error Dist-Corr Z, Phi (129,129,144)



Error Dist-Corr R,



Correctness Testing

- Use a pair of known distribution for potential and charge.
- Compare to ANSYS/Garfield.
- Work in progress.

Thank You