3D Field and Detector Response Calculations With Boundary Element Method

Brett Viren

Physics Department

BROOKHAVEN NATIONAL LABORATORY

DUNE FD Sim/Reco
August 1, 2016

Outline

Detector Response

- A single charge passing near a LArTPC wire **induces** a **current waveform** which is then **shaped and digitized**.
  - Collection wires too, truly collected charge makes a short, current spike.
- The waveform **of a single charge** depends, in fine detail, on the instantaneous values of the:
  - **charge velocity** (vector), ie, drift velocity.
  - **charge location** relative to the **wire of interest** as well as **every other electrode** in the vicinity.
- In reality, we measure an **ensemble of superimposed waveforms** from multiple, **distributed charges**.
- We must **deconvolve** the ensemble waveforms with some **average response function** to produce a good measure of **distribution of drifting charge**.

$\rightarrow$ the more the average response function matches the the true, instantaneous response function the better the deconvolved LArTPC image.

Instantaneous Induced current

Current in k^{th} wire due to charge q at position $\vec{r}(t)$ at time t:

$$i_k(\vec{r}(t)) = q \times (\vec{E}_{weight,k} \cdot \vec{v})$$
$$\vec{v} = \mu \vec{E}_{drift}(\vec{r}(t))$$

Drift field $\vec{E}_{drift}$ from applied high voltage

Weight field $\vec{E}_{weight,k}$ constructed *Shockley-Ramo* field.

Drift Velocity $\vec{v}(t)$ determines (mean) drift path of an electron.

Mobility $\mu = \mu(E_{drift})$ of the drifting charge in LAr.

($\vec{E}_{weight,k}$ calculated by setting wire k to 1V and all other electrodes to 0V.)

The Game

1. Define electrode geometry (eg, some wires + cathode).
2. Calculate scalar and vector fields:
   - ϕ_{drift} and its $\vec{E}_{drift}$ for the given geometry.
   - $\phi_{weight,k}$ and its $\vec{E}_{weight,k}$ for each wire of interest.
   - v velocity vector field.
   - i instantaneous scalar current field.
3. Step through vector vector field while sampling current field.
4. Explore ways to average sampled currents to form **average response functions**.

The Boundary Element Method (BEM)

1. Discretize electrode surfaces with a triangular mesh.
2. Define potential boundary conditions on mesh elements.
3. Integrate Laplace equation $\nabla^2\phi = 0$.
4. Fit integral equation to boundary values.
5. Evaluate at points in the volume to get $\phi(\vec{r})$.

A lot of math and code: rely on GMSH for meshing (1-2) and BEM++ for solving/evaluating (3-5).

Element Methods: Boundary vs. Finite

BEM

- Meshes the surfaces.
- Fast for low surface-to-volume. ($\vec{E}_{weight,k}$)
- Performance relies on relatively new math discoveries.
- Relatively few software implementations.

This work.

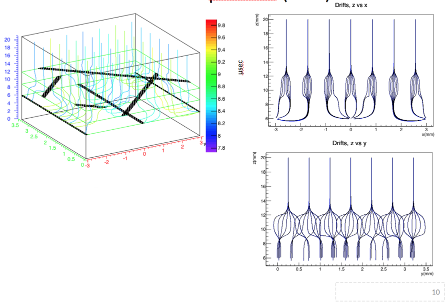
FEM

- Meshes the volume.
- Fast for high surface-to-volume. ($\vec{E}_{field}$)
- Adaptive meshes can improve performance.
- Many implementations, heavily used in industry.

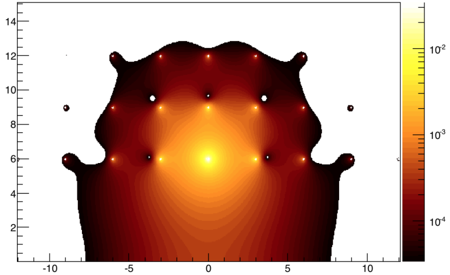
Past 2D and new 3D $\rightarrow$

Plug for Leon Rochester's FEM Work

Drifts in µBooNE (FEM)



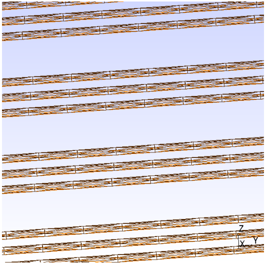
FEM Weighting field (2D)



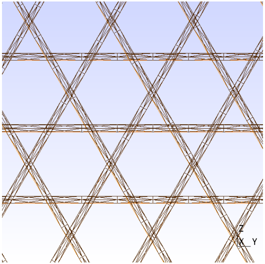
- Contemporaneous and collaborative work with Leon Rochester @ SLAC.
- Custom C++/ROOT FEM code inspired by ICARUS' FORTRAN codes.
- Realizes the benefit of the FEM approach (eg, near-surface precision).

→ Potential BEM/FEM hybrid to get the best of both.

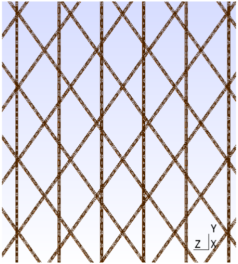
Wires Meshes



"Parallel":
3mm pitch and gap
all wires parallel



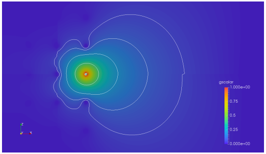
"MicroBooNE"-like:
3mm pitch and gap
60° angles for U/V.



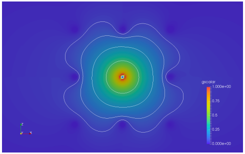
"DUNE"-like:
5mm pitch and gap
35.7° angles for U/V.

Wire generation is parameterized so easy to explore different wire patterns.

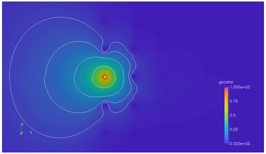
Parallel Wires - Slice Through Weighting Potentials



U plane



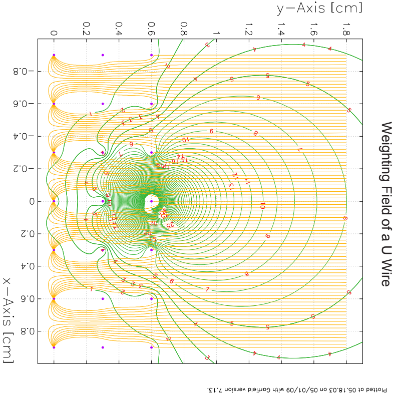
V plane



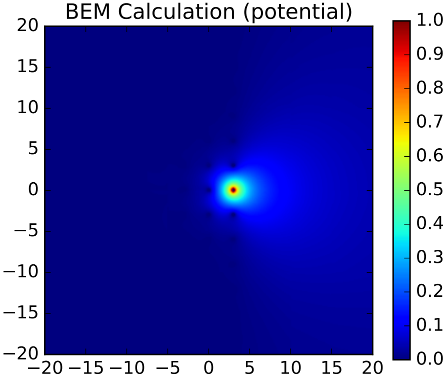
W plane

- X-Z slice through plane of symmetry (Y=0).
- Color shows weighting potential: 0-100%.
- Lines: 5%, 10%, 20%, 40% weights.

Weighting Potential - 2D vs “2D”



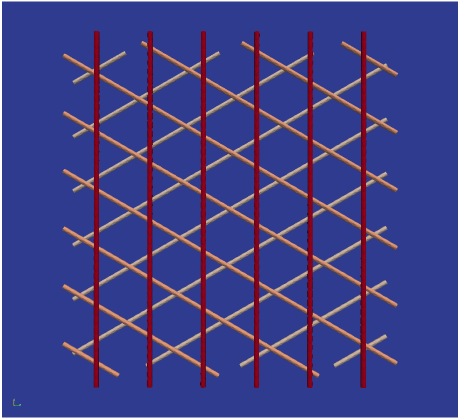
Garfield 2D calculation from Bo

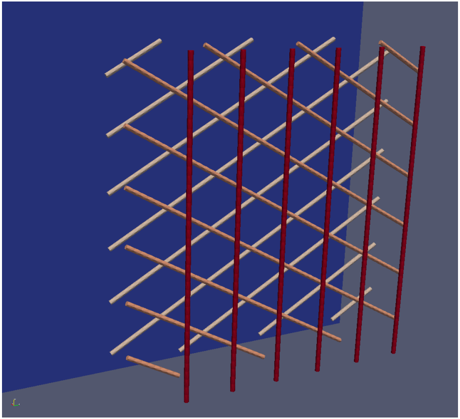


3D BEM, parallel wires, sliced at Y=0.

Initial, qualitative agreement.
More checks would be good but, good enough to continue.

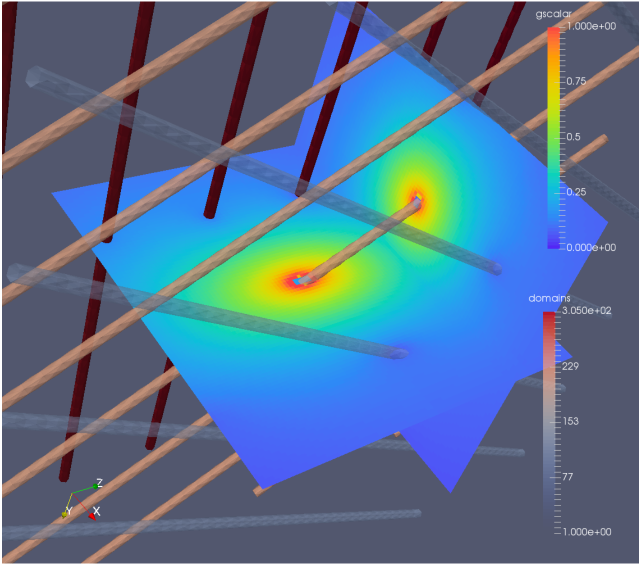
Wire Geometry with Cathode Plane



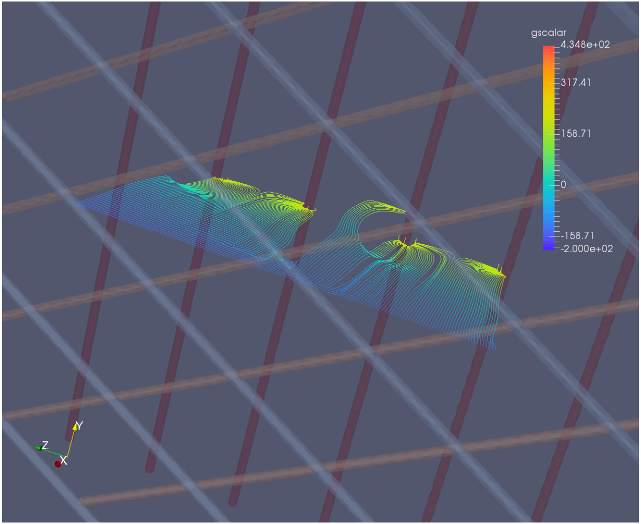


Wires parameterized by pitch, angle, bounding box, radius.

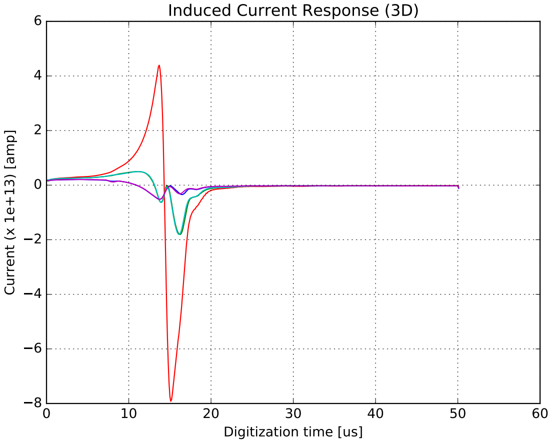
V-plane $\vec{E}_{weight}$ Slices

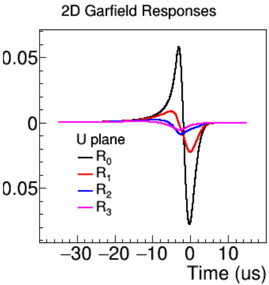


Paraview stepping from line source



Initial Response Function Results





2D equivalent (from uB signal note)

Response for charge near a U-plane wire and charge passing near its nearest and next-nearest wires.

Preliminary, hot off the press!

Next steps

- More validation with 2D and Leon's 3D FEM work.
- Initial results **unfortunately** very similar to 2D!
  - Just 2D→3D not sufficient!?
  - Go to +/- 10 neighbor wires for good 2D-deconvolution.
  - Computation increases ∼ linearly with number of wires
    → (for BEM, FEM is N^2!).
- Implement **protoDUNE/SP and DUNE FD** geometries!
  - Many more weighting fields needed due to non-uniform wire crossing pattern
  - Need ∼30 unique $\vec{E}_{weight,k}$ instead of 3.
  - Can look at "edge effects" such as at APA boundaries.

Anyone wanting to get involved is welcome!

LARF - **L**iquid **Ar**gon TPC **F**ield Calculator

Some features:

- Handles pretty much everything with a simple **command line interface** and **configuration file**.
- Standard Python installation.
    - Heavy lifting: BEM++, GMSH and Numpy.
- Results stored in Sqlite3 database with result provenance.
- Multiple, common data export methods (`.npz`, `.vtk`)
    - Playes well with paraview!

→ **users/developers welcome**.

Code and docs on GitHub:

`https://github.com/brettviren/larf`.

Warning: docs need a refresh, contact me before reading so I can clean up.

extras

Stepping

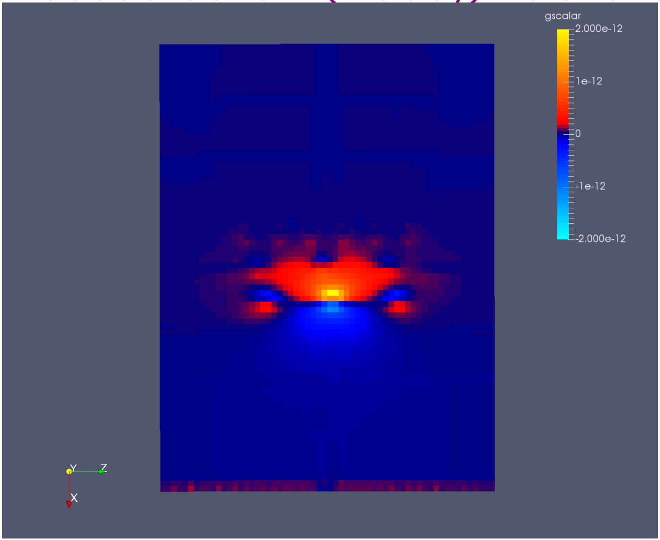
Evolution of position $\vec{r}_k \equiv \vec{r}(t_k)$ from step k to $k+1$:

$$\vec{r}_k \rightarrow \vec{r}_{k+1} = \vec{r}_k + \vec{v}(\vec{r}_k) \times \Delta t_k$$

Use **Adaptive Runge-Kutta + Cash/Karp** ($\sim 5^{th}$ order)

- From Numerical Recipes
- Take two different 6th order steps.
- Distance between each estimates error.
- Adjust Δt_{k+1} based on error
- Hugely more efficient than simple RK4
- Must still watch for stepping to get stuck at maximum.

Induced Current (Krabby) Field for U Wire



U-wire at center, drift goes upward, Y-Z slice, 0.5 mm voxels, current in Amps.