γ – LASER Mode
Beam Monitoring & Scintillation Detectors

John Hallford

University College London

13/07/2020

The incidence of these particles on a detector plane orthogonal to a beampipe is described by this equation:

$$x = R(1 - cos(sin^{-1}(\frac{z_m}{R}))) + tan(sin^{-1}(\frac{z_m}{R}))z_d$$

We also must create a provision to account for the possible rotation of the detector in the Y axis by an angle of θ. Separating the components of the final x value into x_1, x_2 and x_3, we sum them to obtain the 'global' x-value of the hit.

$$x_1 = R(1 - cos(\phi))$$

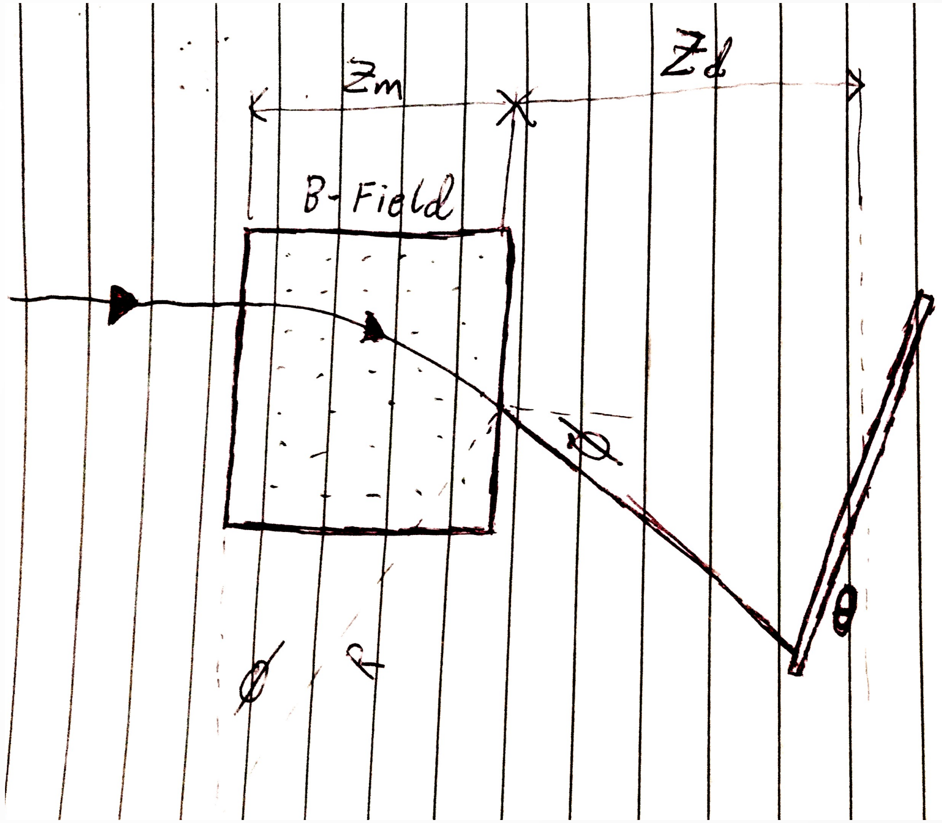
$$x_2 = tan(\phi)z_d$$

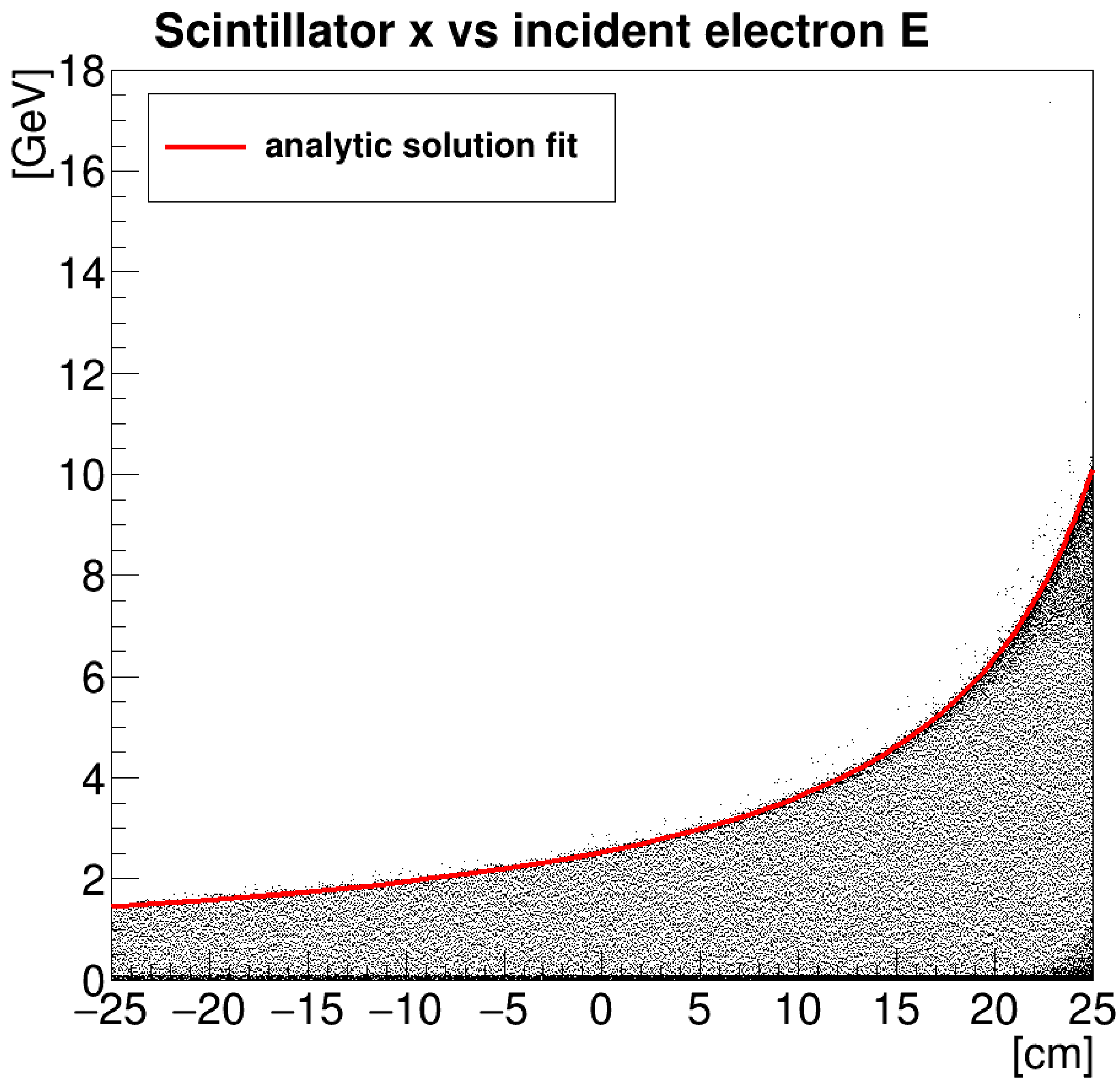
$$x_3 = \frac{tan(\theta)tan(\phi)(x_{detector} - x_1 - x_2)}{1 + tan(\theta)tan(\phi)}$$

Where $\phi = sin^{-1}(\frac{z_m}{R})$. From there mapping to the detector's 'local' x coordinates is elementary: $x_{local} = x_{global} - x_{detector}/cos(\theta)$

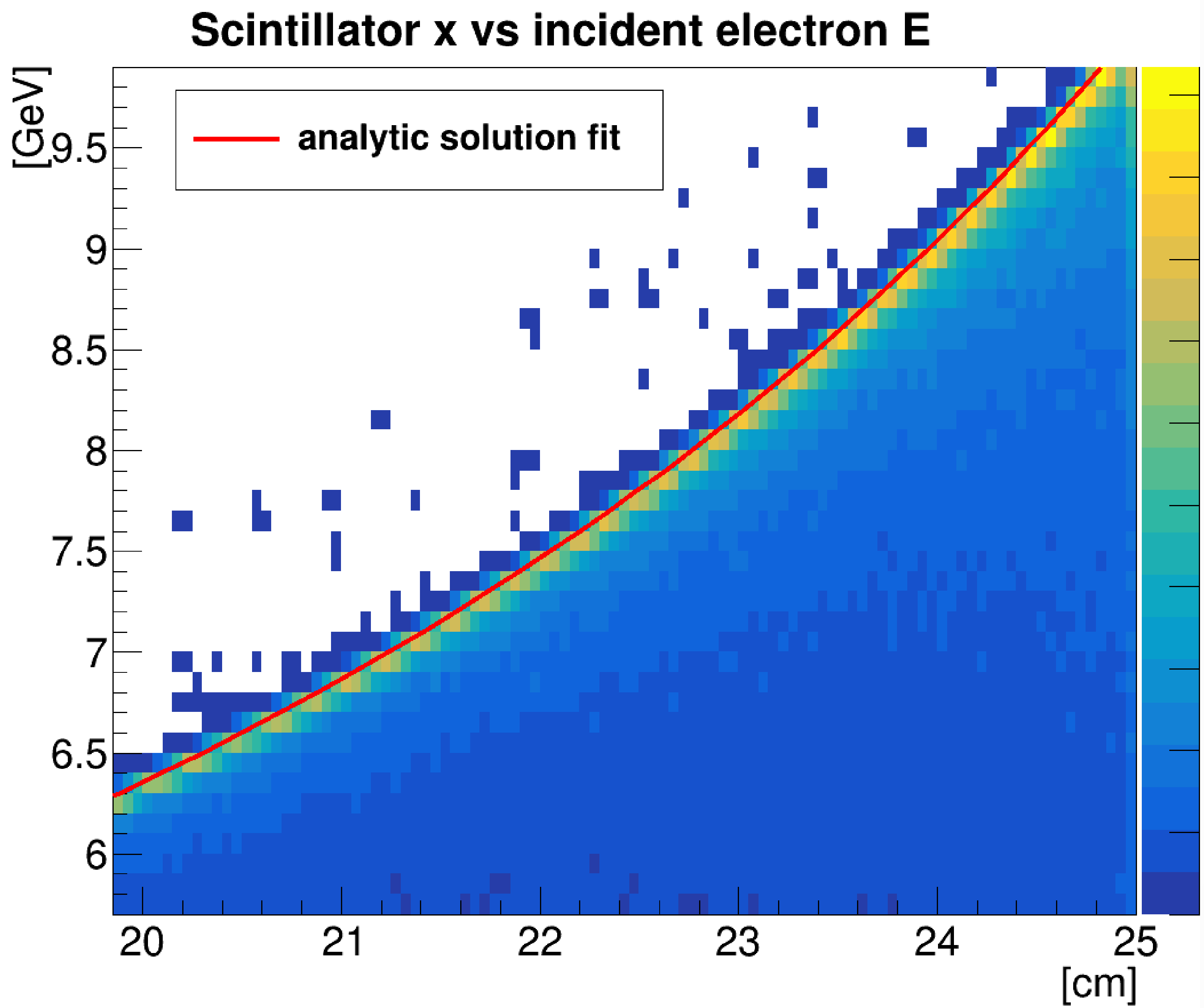
R = E / Bc

(Energy in eV)





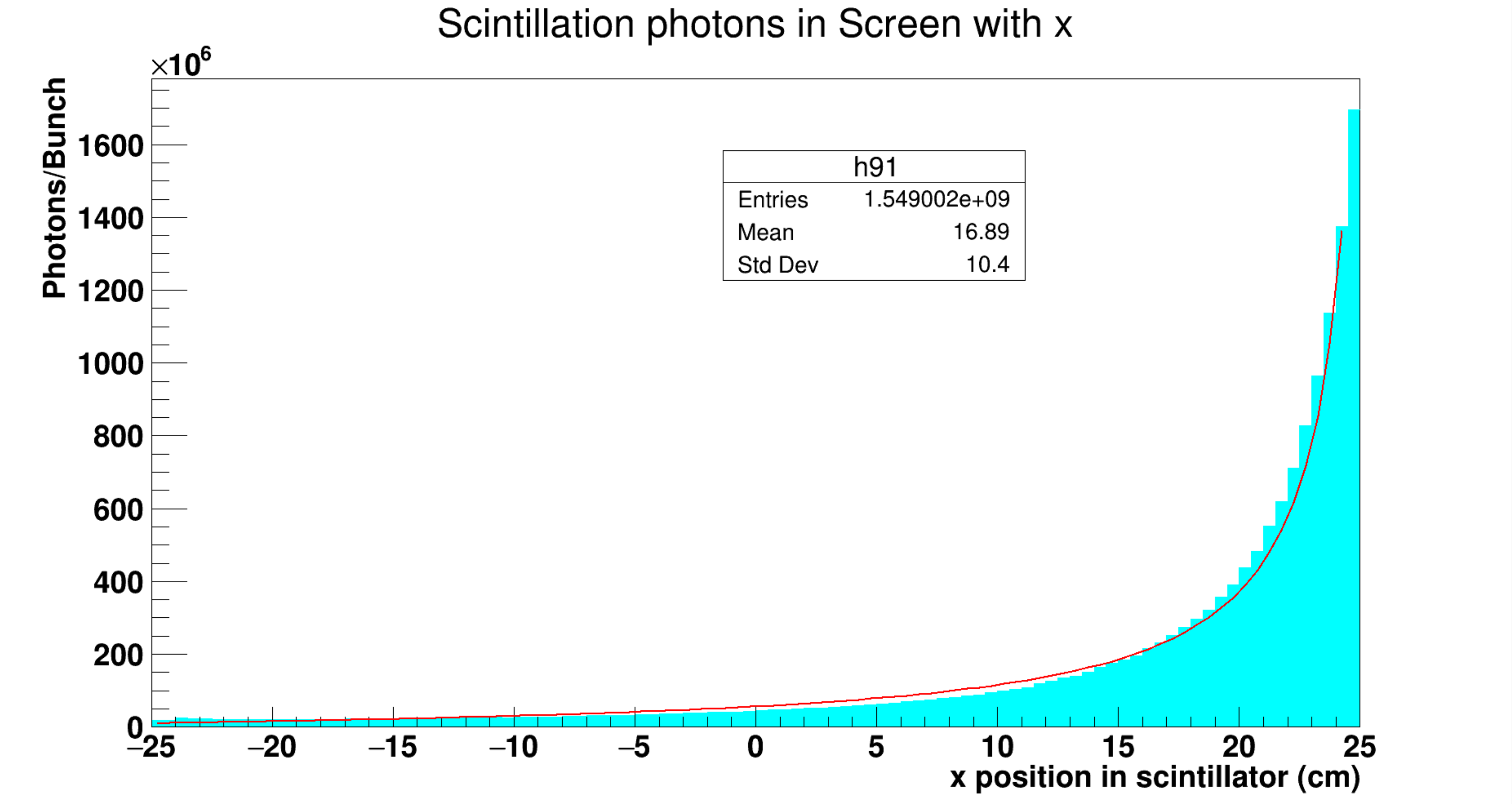
Function overlaid on GEANT electron energy vs hit for scintillator screen. Clearly in good agreement.



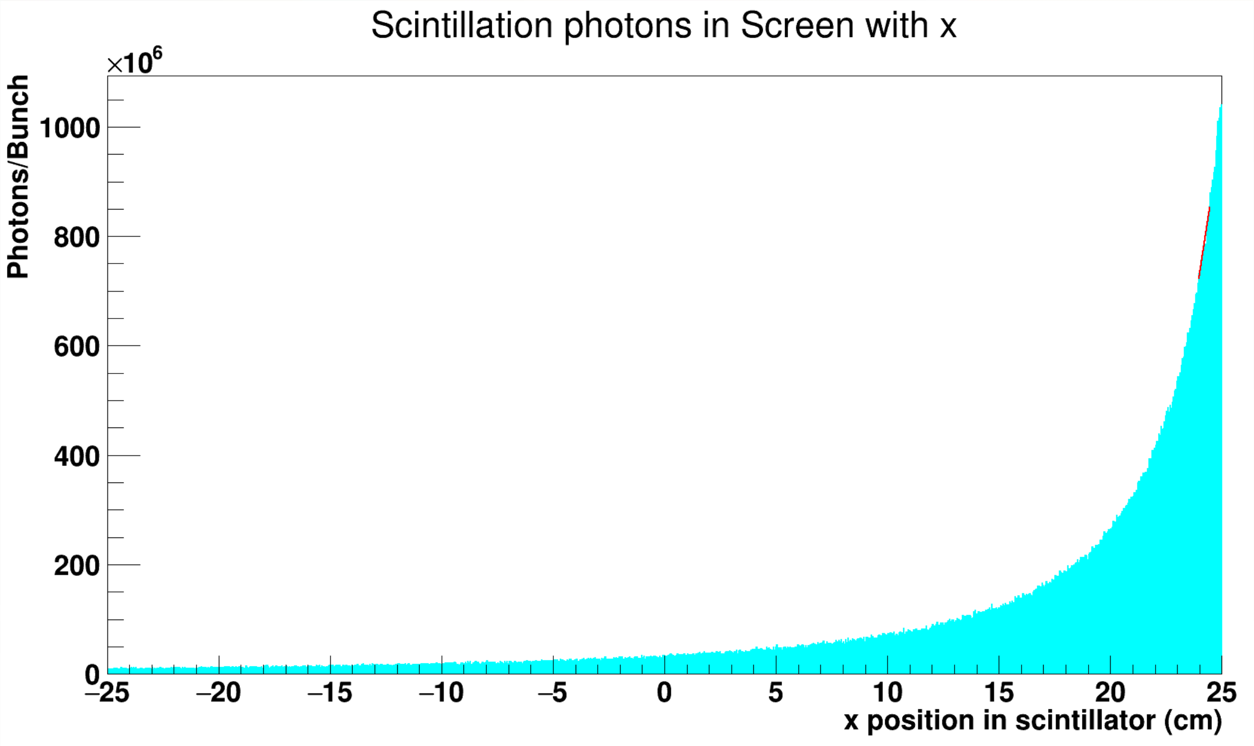
This should lend validity to the equation as it agrees with simulation, and vice versa. A minimal angular spread in electrons after target looks like a good assumption for our purposes. Need to remember both models assume constant B-field box (& no fringe field)

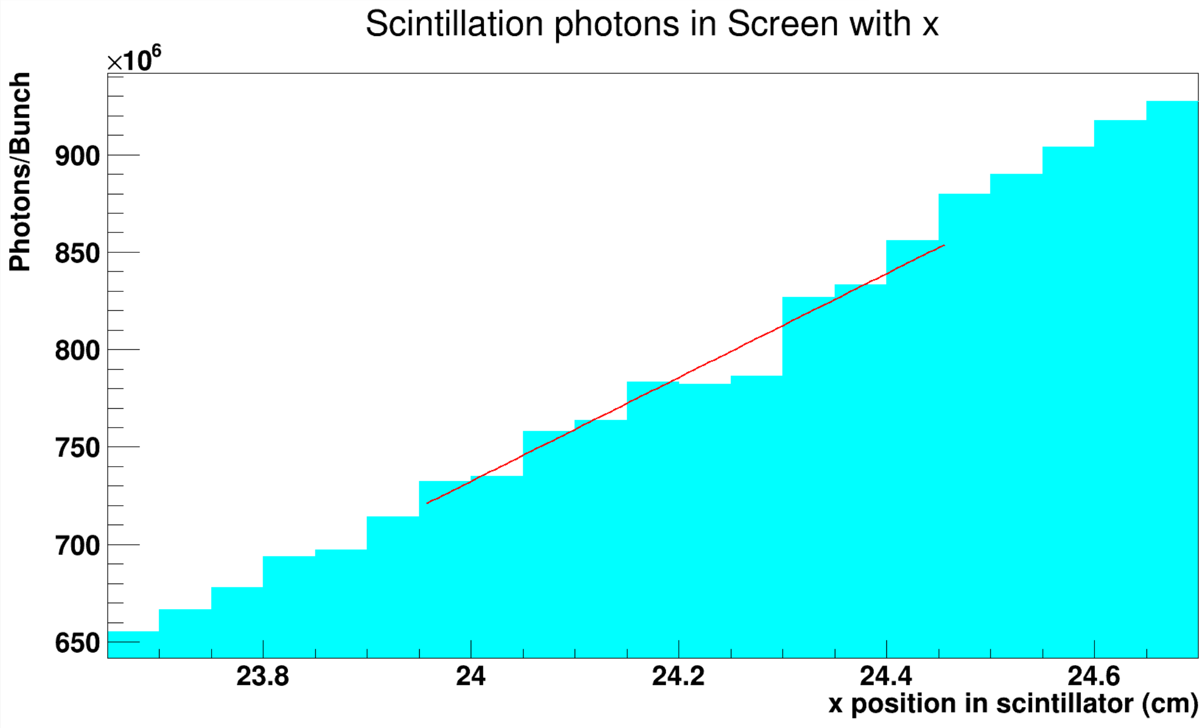
From here, I have looked at the simulated light output profile across detector ‘x’, and used the previous function to determine x-position for some E. For chosen intervals of E, find the corresponding interval in x, and find the integral of Scintillation light within. Then need to divide N_photons by photons/Electron for each electron energy, to find an electron E spectrum.

From there we can take E_beam – E_e to gather a Photon spectrum

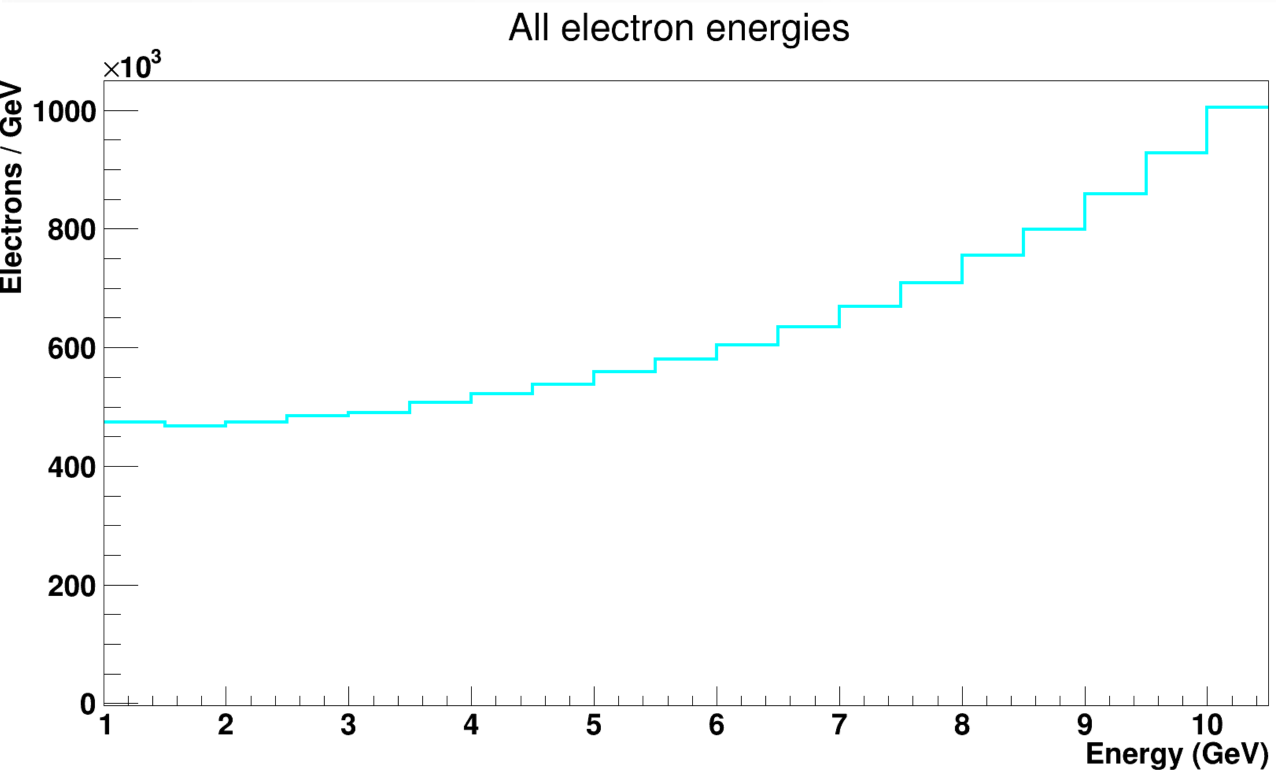


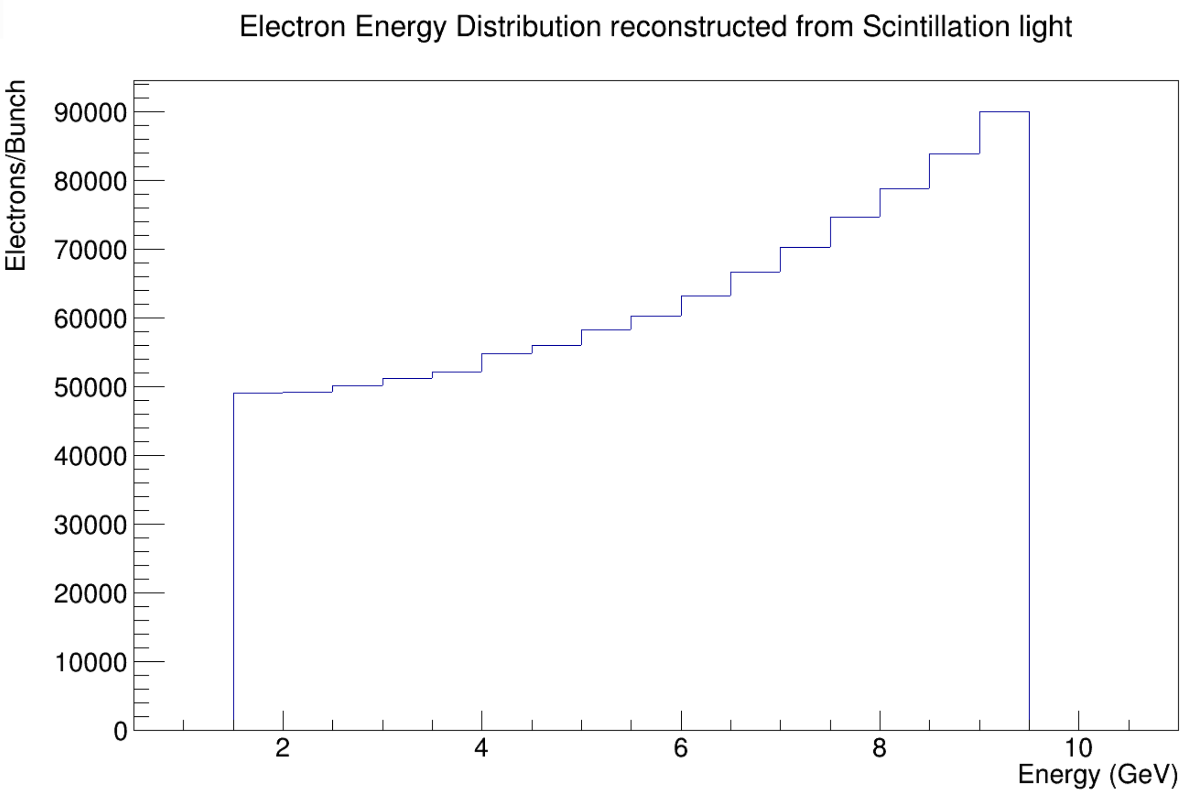
Functional fit of the output Scintillation light (with eg. exp, inverse 1/x) has not been satisfactory. So, I have constructed linear fits only between the intervals (bins)





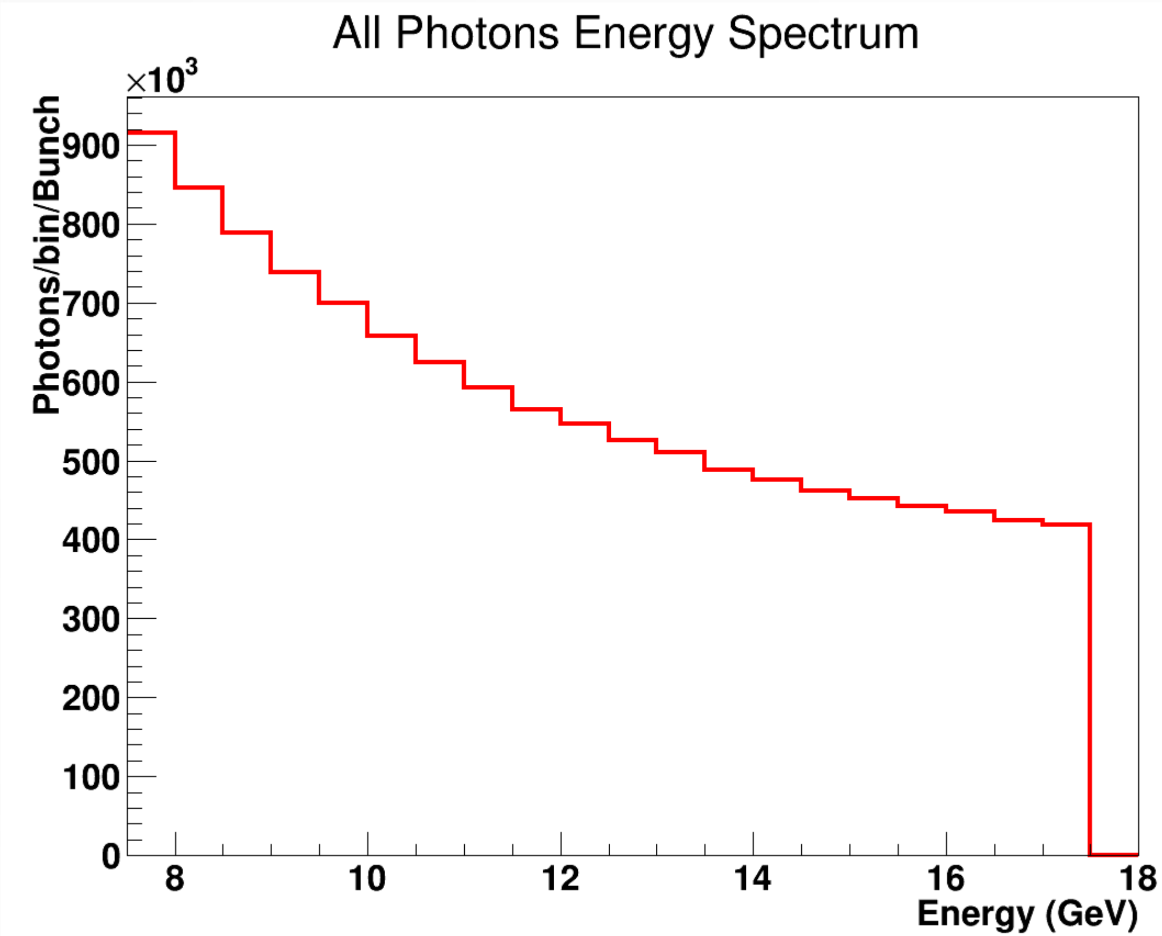
The method of approximating here looks reasonable for the small intervals, by eye at least, even at the steeper section of the spectrum.

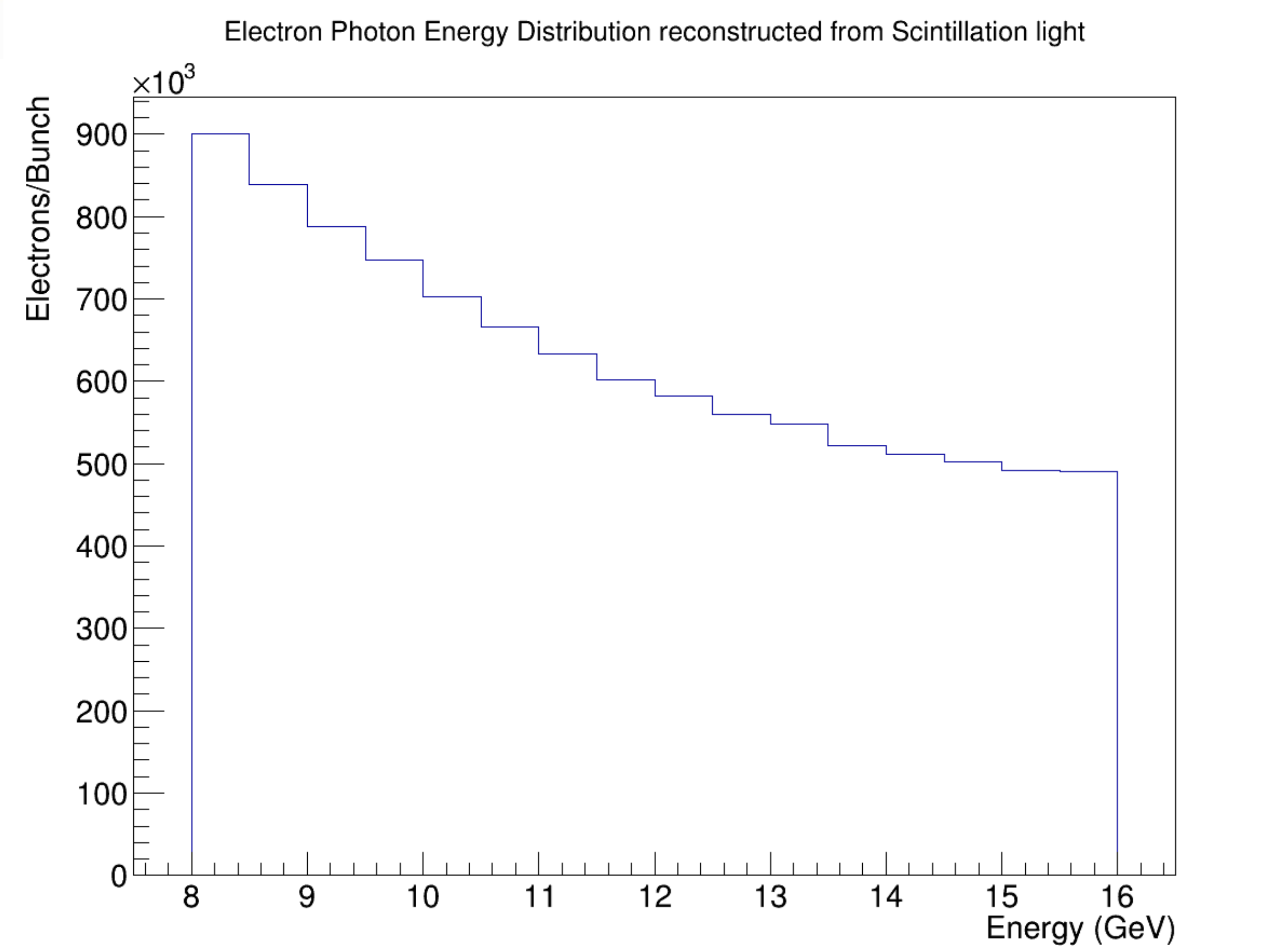




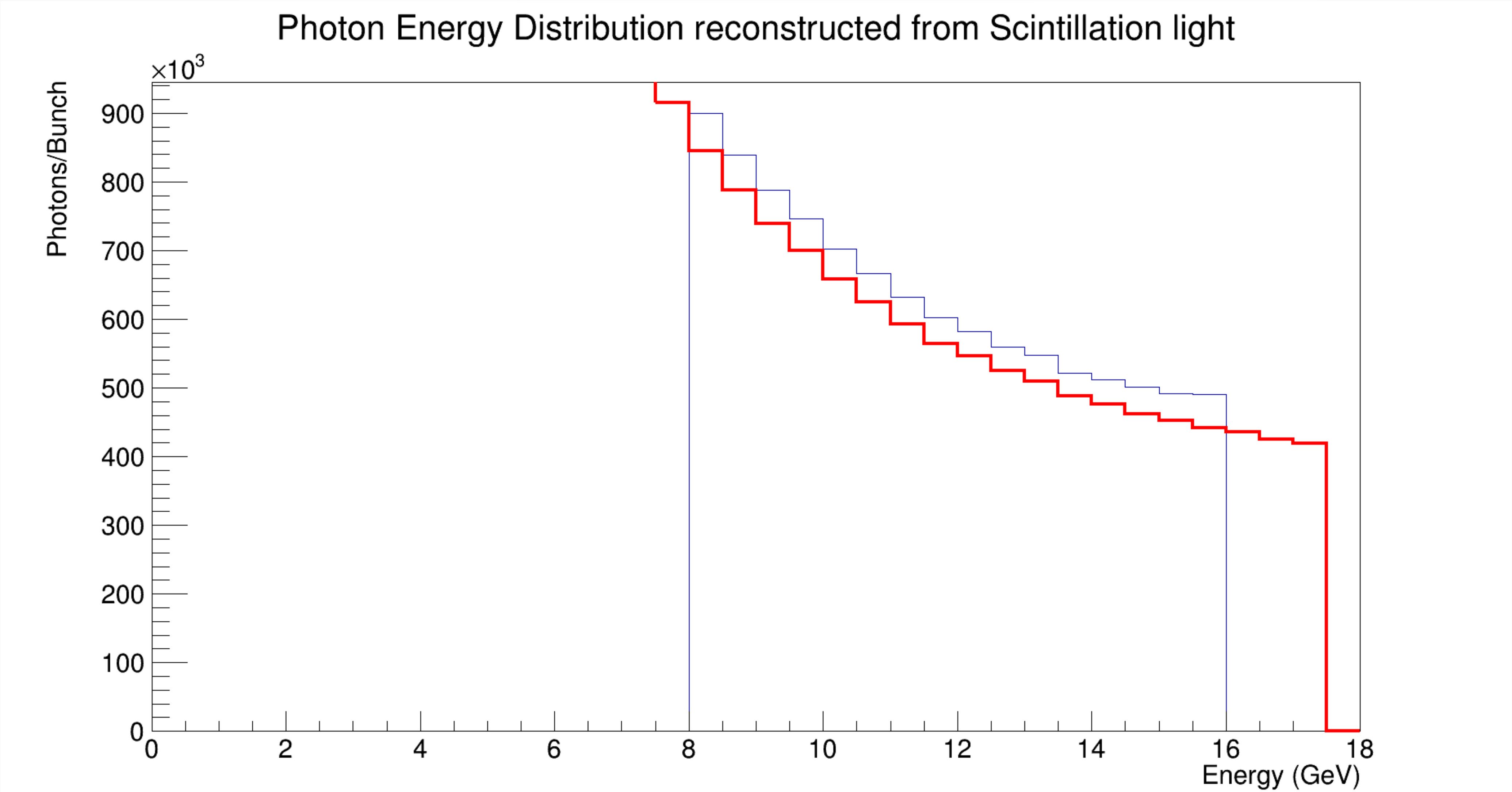
'Truth' electron distribution on left. Reconstructed electron E on Right. We have a shape that looks comparable.

Reconstructed spectrum now looks wrong by factor $\sim$ 10. Process is validated (visually/qualitatively), but need to fix & understand the order of mag. error





Comparing again only shape. (We are cheating and have really scaled the reconstructed spectrum by 10). If we look carefully we see the recon. Low-E trend flattens quicker due to contribution of pair production &



Consistent underestimate from this method?

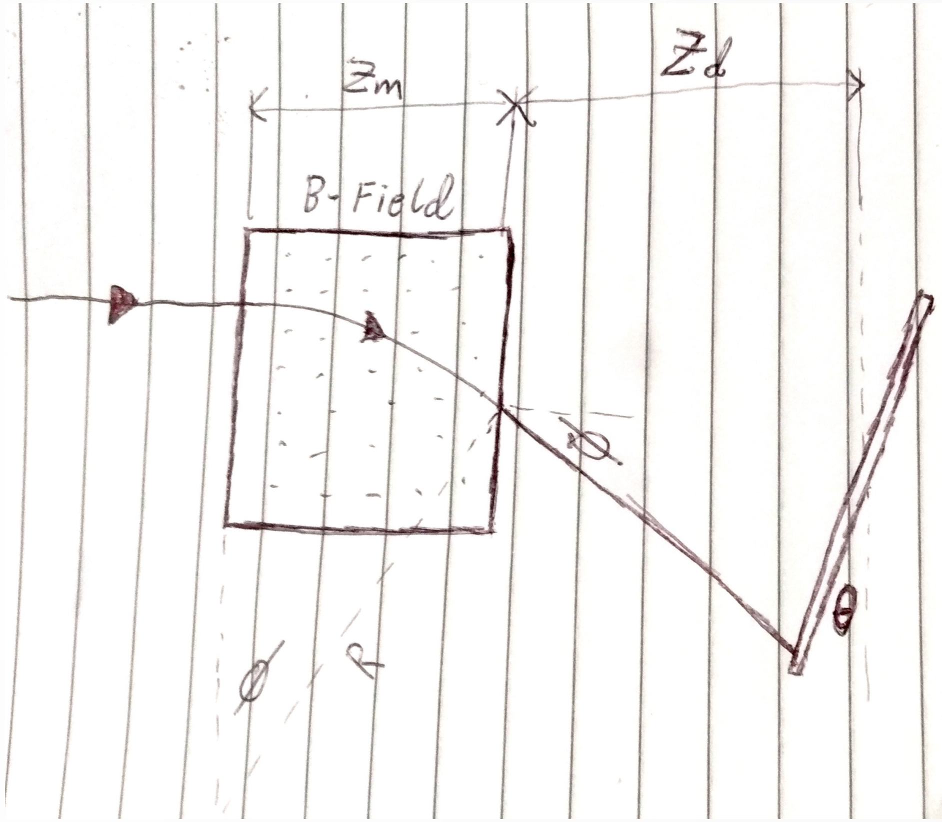
As analysed before, ∼ 10% of Brem events are more complex than $E_\gamma = E_{beam} - E_e$

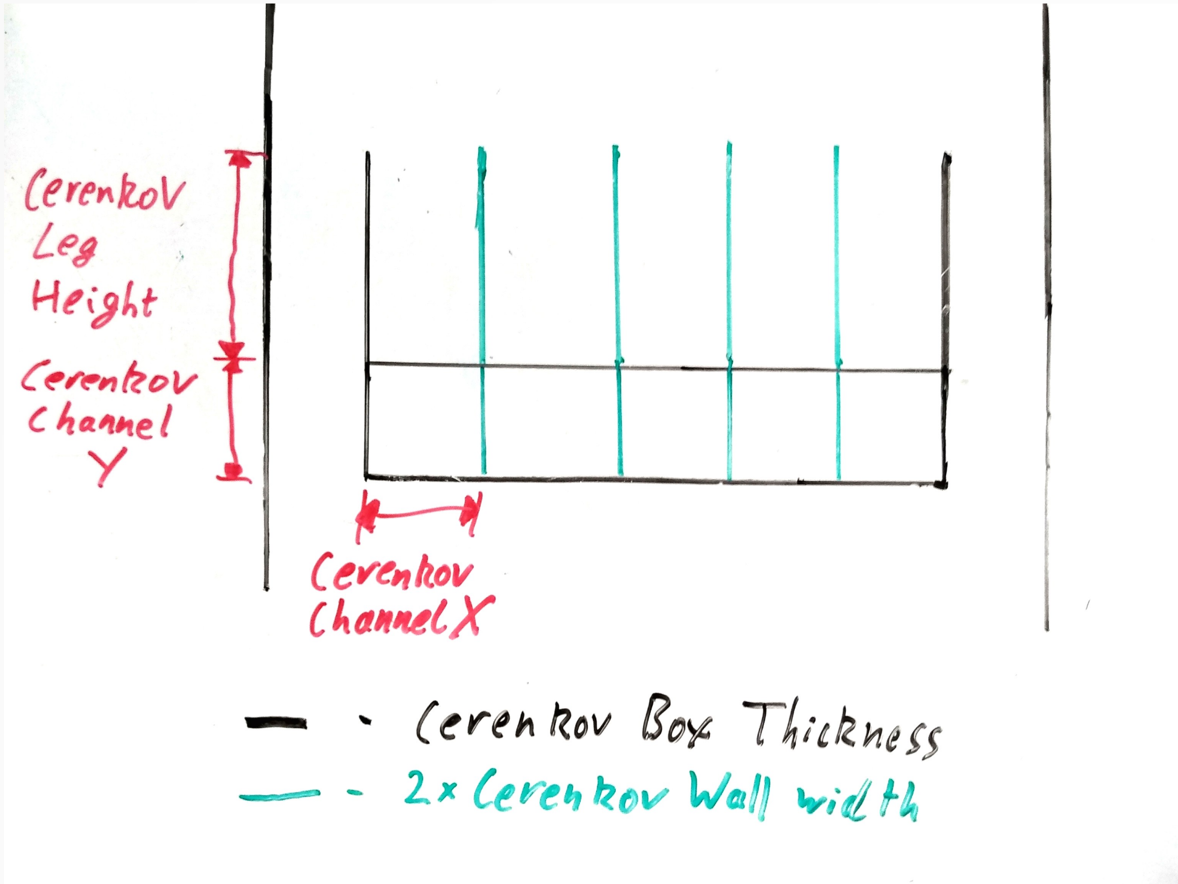
Still to do:

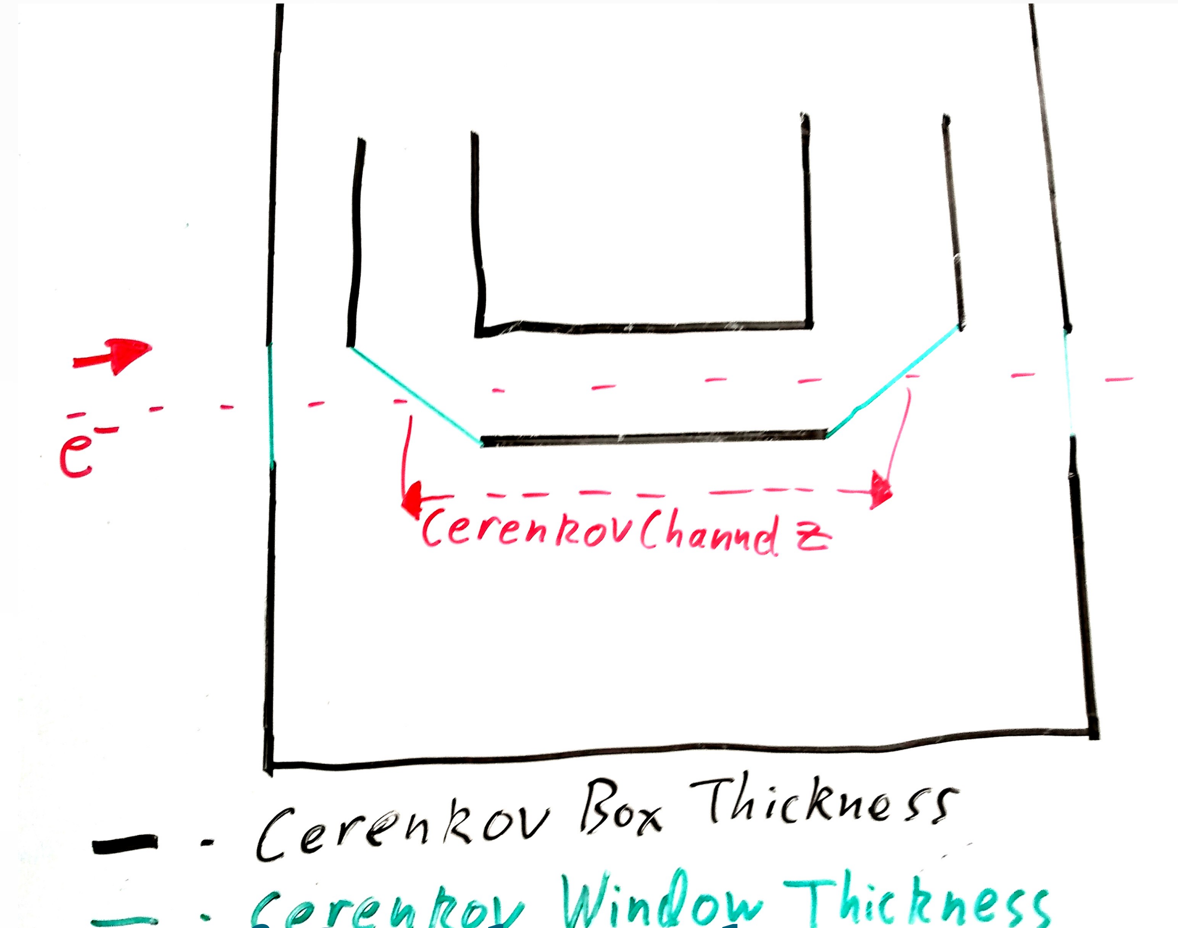
Need to incorporate an angular correction, as particles not perpendicular to screen travel further by 1/(cos(ϕ-θ)) (but this effect should be small)

Will re-do simulation and analysis to eliminate this order-of-magnitude error

But more importantly, writing for CDR chapter!

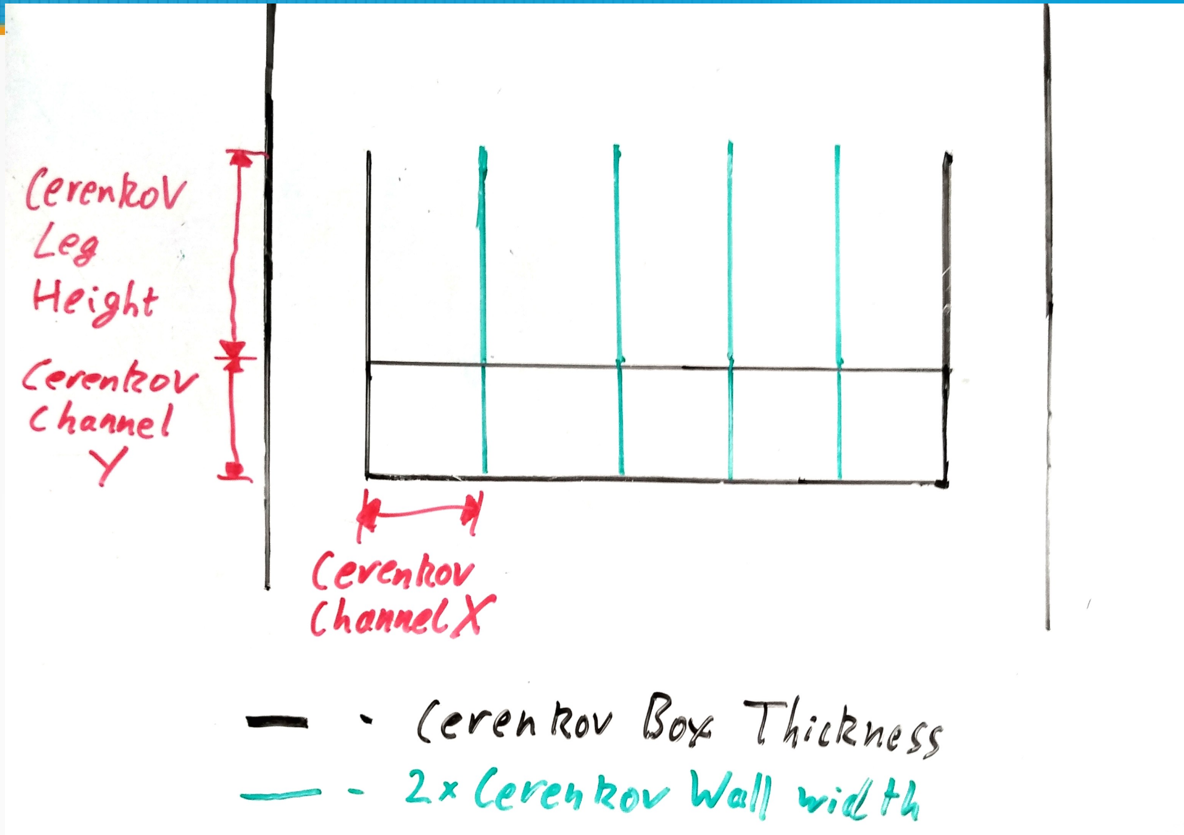


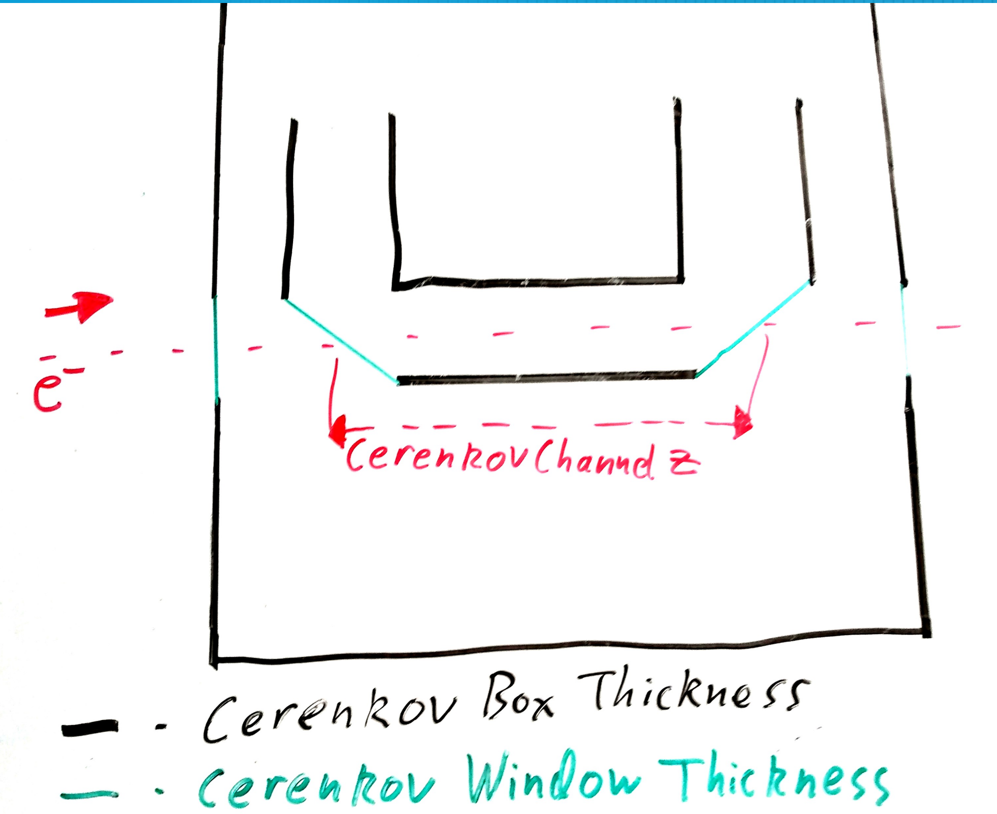




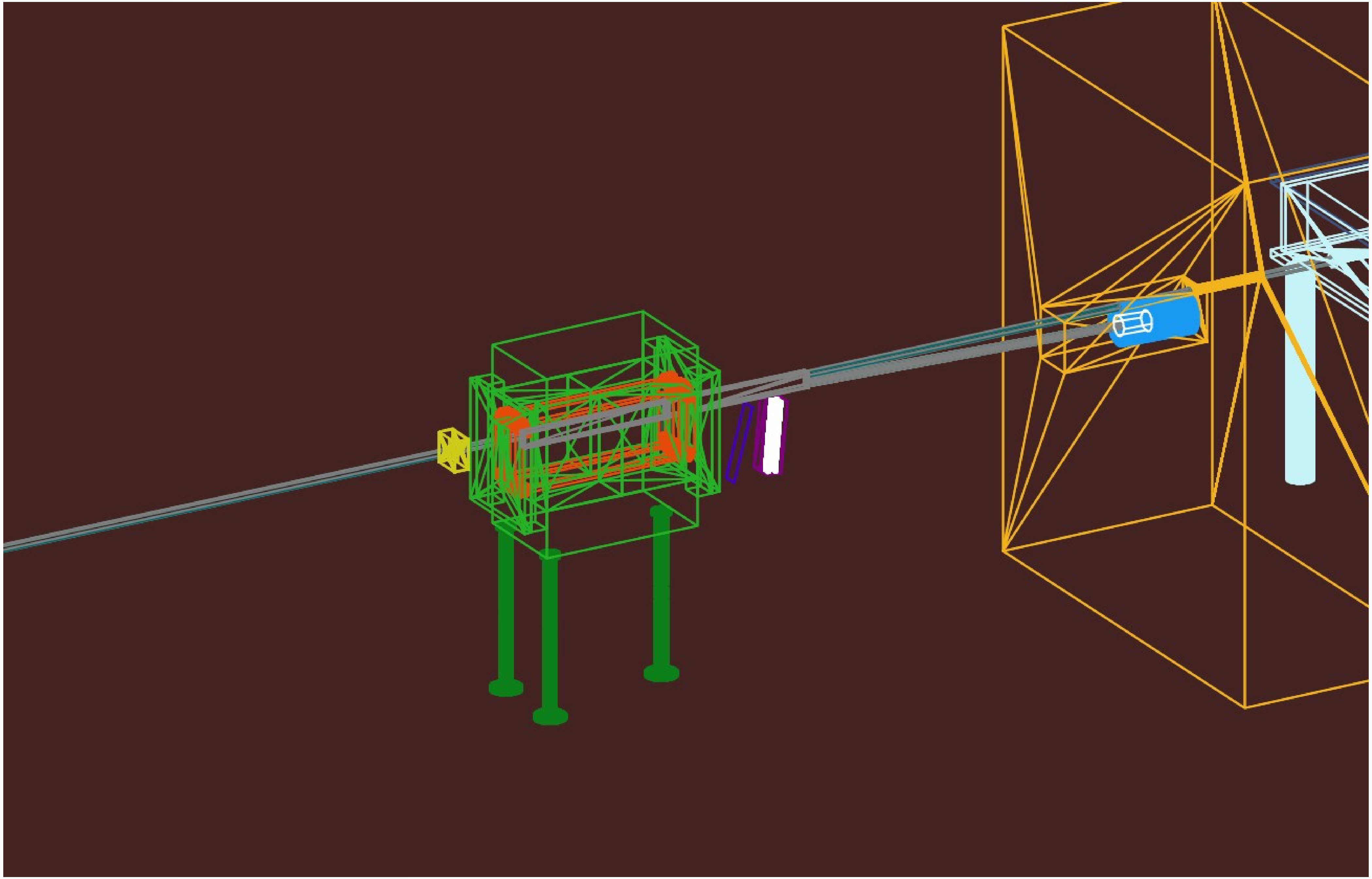
x-y (left) and z-y (right) diagrams of Cerenkov volume.

Upon rotation and placement beside beampipe, ‘x’ and ‘y’ directions interchange



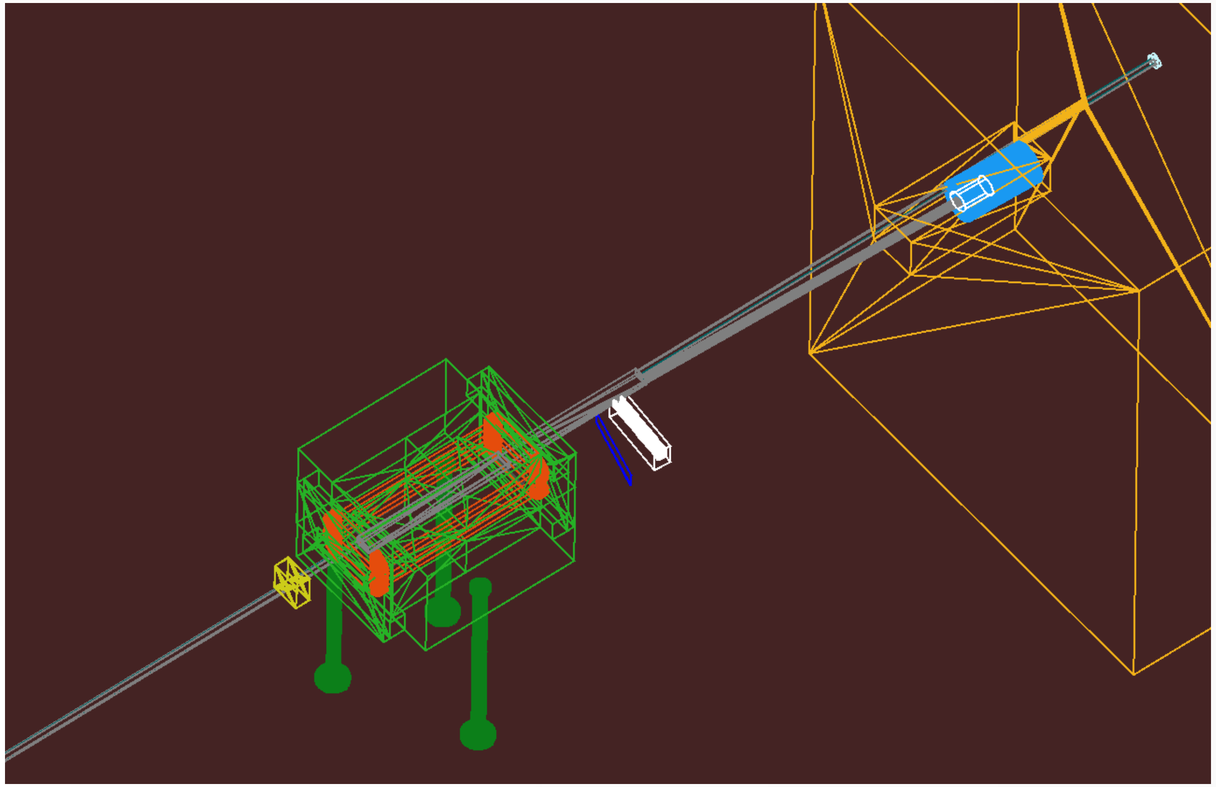



```
//since Magnet / Brem detectors are rotated 90 degrees, X dimensions here (Scint/Cerenkov) map to global Y dimensions & vice versa
ScintCerenkovPhysics = false;
ScintAngle = 15. *deg;
ScintXpos = -360. *mm;
ScintZpos = DumpMagnetZpos + (1520./2. + 275.) *mm; //  defining in relation to position of the beam-dump magnet and its length
CerenkovAngle = 5. *deg;
CerenkovXpos = -350. *mm;
CerenkovZpos = DumpMagnetZpos + (1520./2. + 500.) *mm;
ScintX = 500. *mm;
ScintY = 100. *mm;
ScintZ = 1. *mm;
CerenkovWallWidth = 0.15 *mm;
CerenkovBoxThickness = 1. *mm;
CerenkovWindowThickness = 0.3 *mm;
CerenkovchannelX = 9. *mm;
CerenkovchannelY = 9. *mm;
CerenkovchannelZ = 50. *mm;
CerenkovChannels = 50;         // will crash if not even!! unfortunately..
CerenkovLegHeight = 50. *mm;
ScintMaterial = "G4_GADOLINIUM_OXYSULFIDE";
CerenkovMetal = "Aluminium";
CerenkovMedium = "HeliumGas";
```



Backup

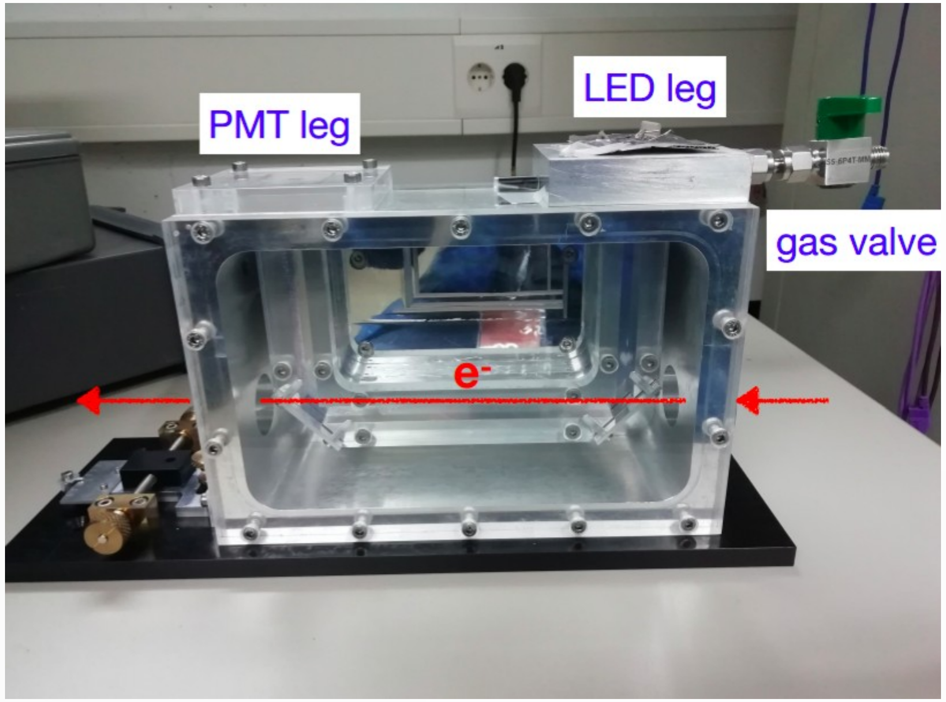
Scintillator/Cerenkov in GEANT4

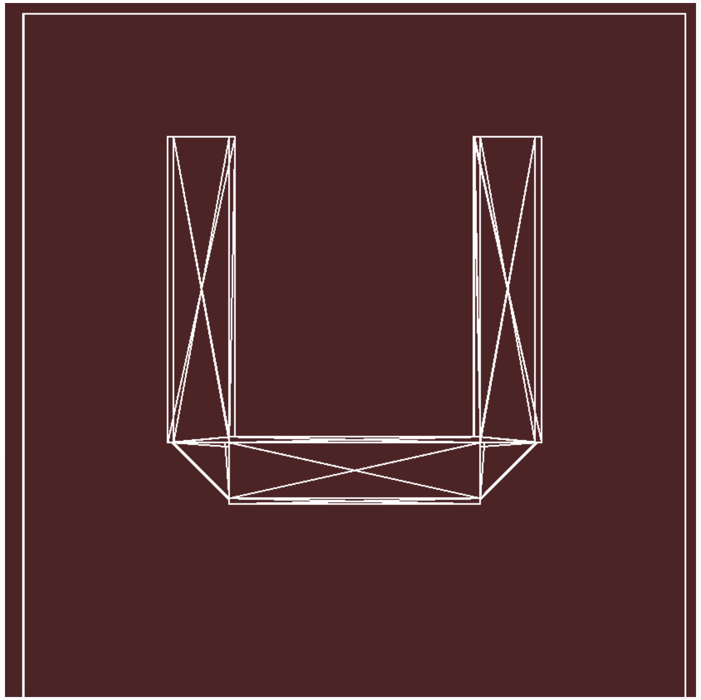


Have implemented geometries for an example simple Scintillating screen and more detailed Cerenkov Devices. Instantiated by

DetectorConstruction::ConstructScintCerenkov()

Takes key parameters from LXSetUp.cc





Scintillator/Cerenkov in GEANT4

I have set up Cerenkov and Scintillation Physics which can be turned on or off, and collect data for histograms (HistoManager) in SteppingAction.

These processes slow down simulation but should not affect validity of background simulation at IP.

Will push a version with these detectors to lxsim git stash (new branch) with physics/histograms commented out

```
ScintCerenkovPhysics = false;
ScintAngle = 15. *deg;
ScintXpos = -350. *mm;
ScintZpos = DumpMagnetZpos + (1520./2. + 300.) *mm; //  defining in relation t
o position of the beam-dump magnet and its length
CerenkovAngle = 5. *deg;
CerenkovXpos = -340. *mm;
CerenkovZpos = DumpMagnetZpos + (1520./2. + 500.) *mm;
ScintX = 500. *mm;
ScintY = 100. *mm;
ScintZ = 1. *mm;
CerenkovWallWidth = 0.15 *mm;
CerenkovBoxThickness = 1. *mm;
CerenkovWindowThickness = 0.3 *mm;
CerenkovchannelX = 9. *mm;
CerenkovchannelY = 9. *mm;
CerenkovchannelZ = 50. *mm;
CerenkovChannels = 50;          // will crash if not even!! unfortunately..
CerenkovLegHeight = 50. *mm;
ScintMaterial = "Polystyrene";
CerenkovMetal = "Aluminium";
CerenkovMedium = "ArgonGas";
```

Scintillator/Cerenkov in GEANT4

These materials take a fair load of memory (compared to the rest of the geometry)

So, given time, I can make a simplified version, if necessary

This was done in lxsim retrieved Sunday. Code has changed since (Dump + magnet rotated 90°) and needs tweaks

```
Start closing geometry.
G4GeometryManager::ReportVoxelStats -- Voxel Statistics

    Total memory consumed for geometry optimisation:   405 kByte
    Total CPU time elapsed for geometry optimisation: 0.01 seconds

    Voxelisation: top CPU users:
    Percent   Total CPU    System CPU       Memory  Volume
    -------   ----------   ----------     --------  ----------
     100.00         0.01         0.00          96k  World
       0.00         0.00         0.00           1k  logicTAUICContainer
       0.00         0.00         0.00          24k  logicTypMBMagnetContainer
       0.00         0.00         0.00          24k  logicTypMBMagnetContainer
       0.00         0.00         0.00           0k  logicComptonDetContainer
       0.00         0.00         0.00           1k  logicBPipeComptonContainer
       0.00         0.00         0.00           0k  logicDetContainer
       0.00         0.00         0.00           0k  logicBremsTargetContainer
       0.00         0.00         0.00           1k  logicGMContainer
       0.00         0.00         0.00           0k  logicGammaTargetContainer

    Voxelisation: top memory users:
    Percent     Memory      Heads    Nodes   Pointers    Total CPU    Volume
    -------   --------     ------   ------   --------   ----------    ----------
      63.54       257k        647     2892       9828         0.00    CerenkovWingLogical
      23.69        96k        212     1260       2838         0.01    World
       5.79        23k         57      286        779         0.00    logicTypMBMagnetContainer
       5.79        23k         57      286        779         0.00    logicTypMBMagnetContainer
       0.32         1k          7       10         44         0.00    logicGMContainer
       0.27         1k          4       13         28         0.00    logicTAUICContainer
       0.19         0k          3        9         19         0.00    logicBPipeComptonContainer
       0.15         0k          3        5         20         0.00    logicVCContainer
       0.07         0k          1        3          8         0.00    logicComptonDetContainer
       0.07         0k          1        3          8         0.00    logicDetContainer
G4WT5 > /run/verbose 2
G4WT5 > /run/setCut 10 um
```