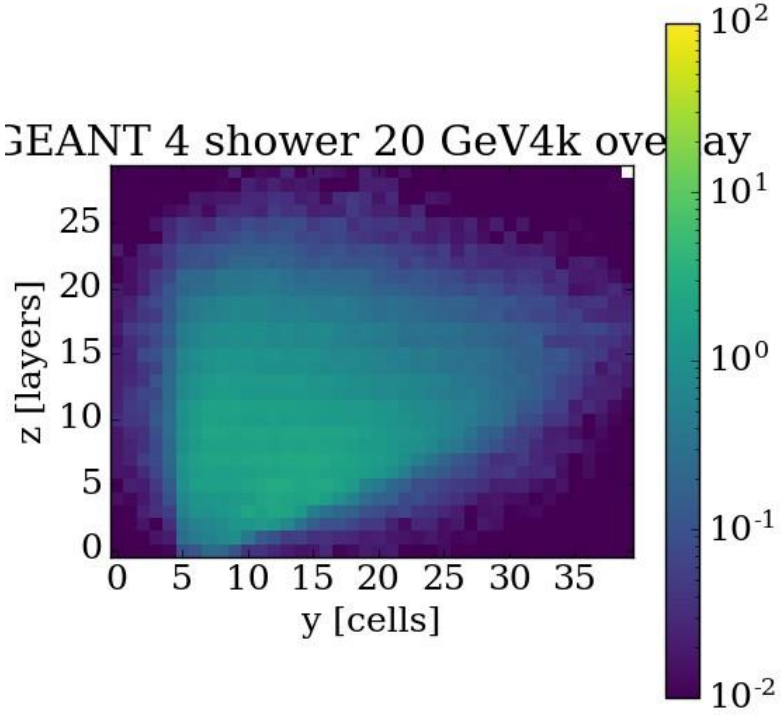
Conditioning on angle of incidence

- Fixed energy- 20 GeV
- Vary polar angle- no transformation (i.e. same initial point of incidence at calorimeter center)
- Shifted gun position in y: 6^{th}-7^{th} bin
- Angular range: 30-90 degrees
- Calorimeter block: 30x30x40



None-uniform in z

- ILD ECAL layers aren’t evenly spaced in z!
- Was causing systematic shift in angle

```
<dimensions inner_r="1.804800000e+03" outer_z="2.350000000e+03" />
<layer repeat="1" thickness="7.065000000e+00" absorberThickness="6.540000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="3.125000000e+00" absorberThickness="2.600000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="9.345000000e+00" absorberThickness="8.820000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="3.125000000e+00" absorberThickness="2.600000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="9.345000000e+00" absorberThickness="8.820000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="3.125000000e+00" absorberThickness="2.600000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="9.345000000e+00" absorberThickness="8.820000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="3.125000000e+00" absorberThickness="2.600000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="9.345000000e+00" absorberThickness="8.820000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="3.125000000e+00" absorberThickness="2.600000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="9.345000000e+00" absorberThickness="8.820000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="3.125000000e+00" absorberThickness="2.600000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="9.345000000e+00" absorberThickness="8.820000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="3.125000000e+00" absorberThickness="2.600000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="9.345000000e+00" absorberThickness="8.820000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="3.125000000e+00" absorberThickness="2.600000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="9.345000000e+00" absorberThickness="8.820000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="3.125000000e+00" absorberThickness="2.600000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="9.345000000e+00" absorberThickness="8.820000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="3.125000000e+00" absorberThickness="2.600000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="1.144500000e+01" absorberThickness="1.092000000e+01" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="5.225000000e+00" absorberThickness="4.700000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="1.144500000e+01" absorberThickness="1.092000000e+01" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="5.225000000e+00" absorberThickness="4.700000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="1.144500000e+01" absorberThickness="1.092000000e+01" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="5.225000000e+00" absorberThickness="4.700000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="1.144500000e+01" absorberThickness="1.092000000e+01" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="5.225000000e+00" absorberThickness="4.700000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="1.144500000e+01" absorberThickness="1.092000000e+01" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
<layer repeat="1" thickness="5.225000000e+00" absorberThickness="4.700000000e+00" cellSize0="5.088333333e+00" cellSize1="5.088333333e+00" />
```

PCA- numpy magic

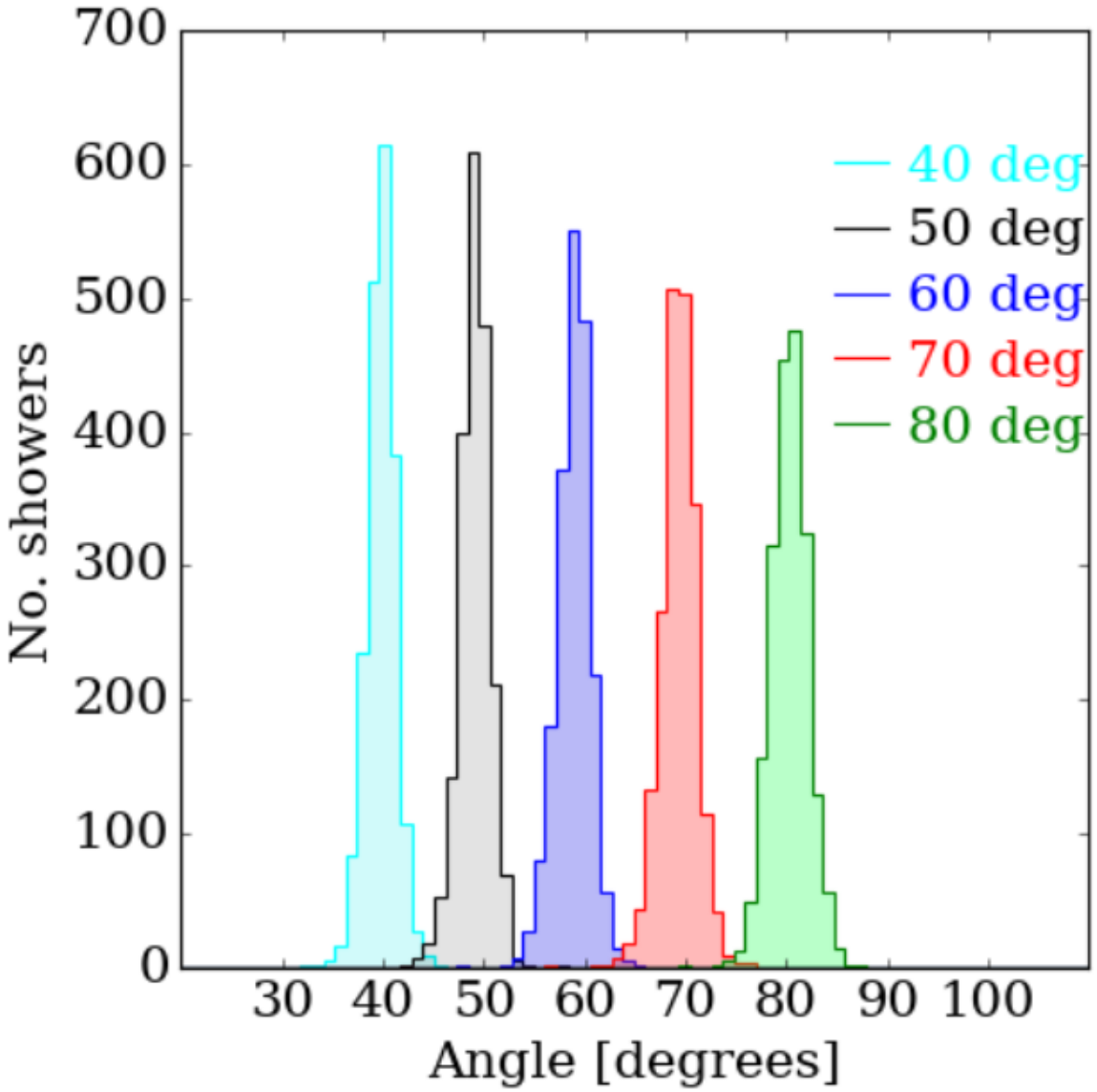
The (symmetric) matrix representing the inertia tensor of a collection of masses, m_i, with positions (x_i, y_i, z_i) relative to their centre of mass is

$$\mathbf{I} = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{pmatrix},$$

where

$$I_{xx} = \sum_i m_i(y_i^2 + z_i^2), \qquad I_{yy} = \sum_i m_i(x_i^2 + z_i^2), \qquad I_{zz} = \sum_i m_i(x_i^2 + y_i^2),$$
$$I_{xy} = -\sum_i m_i x_i y_i, \qquad I_{yz} = -\sum_i m_i y_i z_i, \qquad I_{xz} = -\sum_i m_i x_i z_i.$$

Angular distribution for GEANT4 showers



Backup- code

```
#img_no = 120
g4_theta_40 = np.zeros(2000)
# need to significantly speed up above code to make it usable
# create calorimeter mesh
z, x, y = np.mgrid[0:30, 0:30, 0:40]
print(np.shape(z))
z_pos_true = np.zeros(30)
for i in range(30):
    z_pos_true[i] = np.sum(z_pos[0:i])

#z_stacked_1 = np.vstack((z_pos, z_pos, z_pos, z_pos, z_pos, z_pos, z_pos, z_pos, z_pos, z_pos, z_pos, z_po
#z_stacked = np.vstack((z_stacked_1, z_stacked_1, z_stacked_1, z_stacked_1, z_stacked_1, z_stacked_1, z_sta
z_stacked_2 = np.vstack((z_pos_true, z_pos_true, z_pos_true, z_pos_true, z_pos_true, z_pos_true, z_pos_true
z_2d_stacked = np.transpose(z_stacked_2)

z_mesh = np.zeros((30, 30, 40))
for i,element in enumerate(z):
    z_mesh[:, i, :] = z_2d_stacked



for img_no in range(2000):
    if np.isnan(g4_deg_40[img_no]).any() == True:
        continue
    energy_sum = np.sum(g4_deg_40[img_no], axis=None)
    weight_sum_x = np.sum(g4_deg_40[img_no]*x*x_pos, axis=None)
    weight_sum_y = np.sum(g4_deg_40[img_no]*y*y_pos, axis=None)
    weight_sum_z = np.sum(g4_deg_40[img_no]*z_mesh, axis=None)

    mean_x = weight_sum_x/energy_sum
    mean_y = weight_sum_y/energy_sum
    mean_z = weight_sum_z/energy_sum


    I_zz = np.sum(g4_deg_40[img_no]*((x*x_pos-mean_x*np.ones((30, 30, 40)))**2 +(y*y_pos-mean_y*np.ones((30
    I_yy = np.sum(g4_deg_40[img_no]*((x*x_pos-mean_x*np.ones((30, 30, 40)))**2 +(z_mesh-mean_z*np.ones((30,
    I_xx = np.sum(g4_deg_40[img_no]*((y*y_pos-mean_y*np.ones((30, 30, 40)))**2 +(z_mesh-mean_z*np.ones((30,
    I_xy = -np.sum(g4_deg_40[img_no]*(x*x_pos-mean_x*np.ones((30, 30, 40)))*(y*y_pos-mean_y*np.ones((30, 30
    I_xz = -np.sum(g4_deg_40[img_no]*(x*x_pos-mean_x*np.ones((30, 30, 40)))*(z_mesh-mean_z*np.ones((30, 30,
    I_yz = -np.sum(g4_deg_40[img_no]*(y*y_pos-mean_y*np.ones((30, 30, 40)))*(z_mesh-mean_z*np.ones((30, 30,


    # create moment of inertia tensor by concatenating arrays
    I_yx = I_xy
    I_zx = I_xz
    I_zy = I_yz

    I = np.array([I_xx, I_xy, I_xz, I_xy, I_yy, I_yz, I_xz, I_yz, I_zz])
    I = I.reshape((3,3))

    eigenvalues, eigenvectors = np.linalg.eig(I)

    order = np.argsort(eigenvalues[::-1])

    m = abs(eigenvectors[1,order[0]])/abs(eigenvectors[2,order[0]])

    g4_theta_40[img_no] = np.degrees(np.arctan2(abs(eigenvectors[1,order[0]]), abs(eigenvectors[2,order[0]]
```