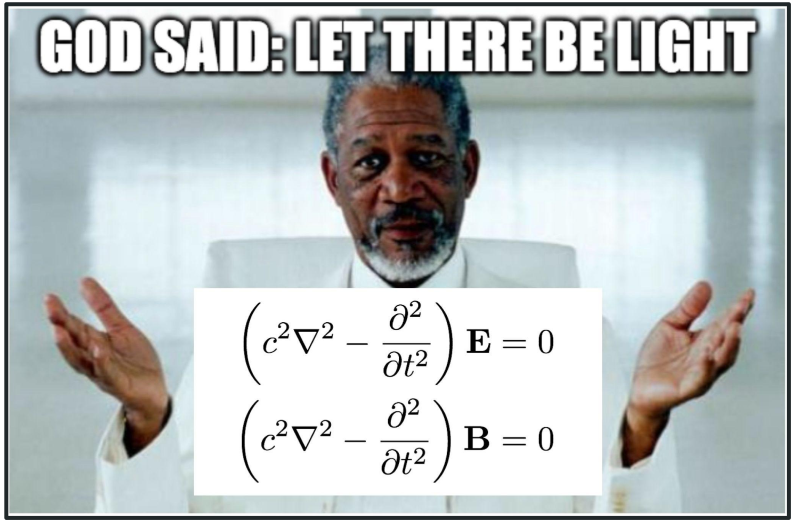
Computational E&M Project

Kyle Gourlie & Madalyn Gragg

Problem Statement

- Determine the following:
  - Electric Potential (V)
  - Electric Field (E)
  - Free and bound charge



- For a solid chunk of metal held at +V in the middle of a spherical tank of water. The chunks must be the following shapes:
  - Cube
  - Cone
  - And More (see prep. activity and HW problem)

Building Blocks (E&M)

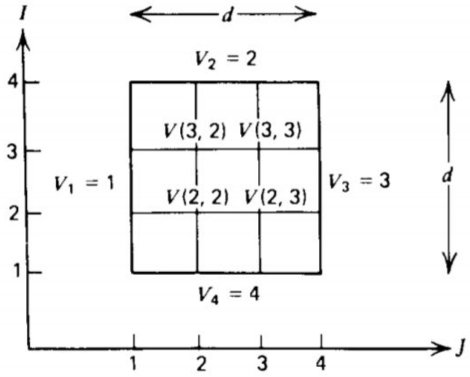
Derivation on next slides

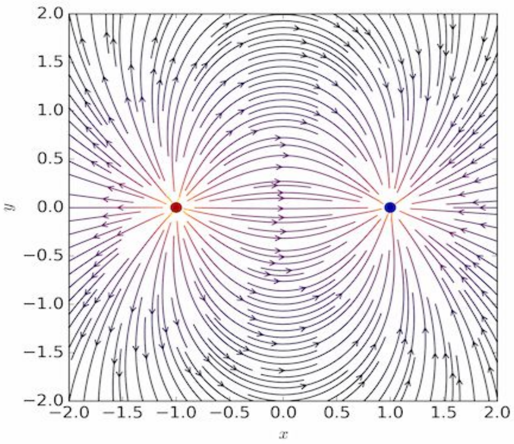
Electric Potential Scalar Field: Numerical differentiation using Method of Relaxation

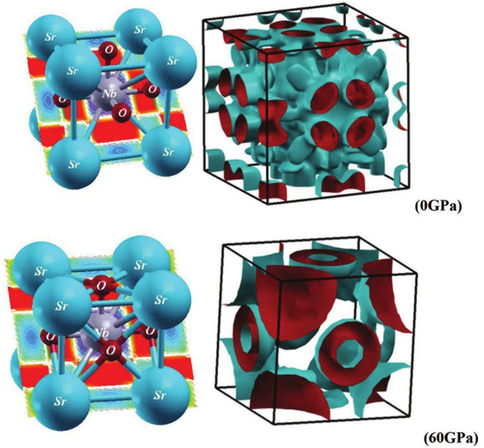
Electric Vector Field: Negative gradient (np.gradient) of potential. -∇V=E

Charge Density (Bound and Free) :

Electric field multiplied by constants (ε and X_e)







Dielectric:

- Charges must be inputted into the dielectric in order for free volume charge to form. We can assume this is not occurring.
- Griffths states that free volume charge is proportional to bound volume charge given a linear media (which water is).
- We can assume that the only charges on the surface will be bound.

Conductor:

- Griffths 2.5 Conductors states: interior volume charge is equal to 0 and that charge can only be located on the surface of a conductor.
- The charge located on the surface of the conductor is free charge since the surface is an equipotential.
- This means we only need to solve for surface free charge.

Dielectric:

- Charges must be inputted into the dielectric in order for free volume charge to form which we can assume is not occurring.

Conductor		Dielectric	
ρ_f	0	ρ_f	0
ρ_B	0	ρ_B	0
σ_f		σ_f	0
σ_B	0	σ_B	

charge can only be located on the surface of a conductor.

- The charge located on the surface of the conductor is free charge since the surface is an equipotential.
- This means we only need to solve for surface free charge.

Math Tidbits : Deriving Equation for Bound Charge

(1) $P = \varepsilon E - D$ ← Definition of Polarization

(2) $D = \varepsilon_o E$ ← Definition of Displacement Field

Equations from Griffths

Math Tidbits : Deriving Equation for Bound Charge

(1) $$P = \varepsilon E - D$$

(2) $$D = \varepsilon_o E$$

$$P = \varepsilon E - \varepsilon_o E$$

Plug (2) into (1)

$$P = E(\varepsilon - \varepsilon_o)$$

Simplify

Equations from Griffths

Math Tidbits : Deriving Equation for Bound Charge

(1) $P = \varepsilon E - D$

(2) $D = \varepsilon_o E$

$P = \varepsilon E - \varepsilon_o E$ ← Plug (2) into (1)

$P = E(\varepsilon - \varepsilon_o)$ ← Simplify

$P = E\varepsilon_o\left(\varepsilon_{H2O} - 1\right)$ ← Use and simplify with: $\varepsilon_{H20} = 1 + X_e$

$\mathrm{P} = \mathrm{E}\varepsilon_o(1 + \mathrm{X_e} - 1)$

Equations from Griffths

Math Tidbits : Deriving Equation for Bound Charge

(1) $$P = \varepsilon E - D$$

(2) $$D = \varepsilon_o E$$

$$P = \varepsilon E - \varepsilon_o E$$

$$P = E(\varepsilon - \varepsilon_o)$$

$$P = E\varepsilon_o\left(\varepsilon_{H2O} - 1\right)$$

$$\mathrm{P} = \mathrm{E}\varepsilon_o(1 + \mathrm{X_e} - 1)$$

$$\mathrm{P} = \mathrm{E}\varepsilon_o\mathrm{X_e}$$

We can assume when we are on the surface that the polarization will be pointing normal to the surface of the dielectric.

$$\sigma_{\mathrm{Bound}} = \mathrm{P} \cdot \mathrm{n_{hat}}$$

$$\sigma_{\mathrm{Bound}} = \mathrm{E}\varepsilon_o\mathrm{X_e} \cdot \mathrm{n_{hat}}$$

Electric Field at interior and exterior of the dielectric will already be solved for.

Equations from Griffths

Math Tidbits : Deriving Equation for Free Charge

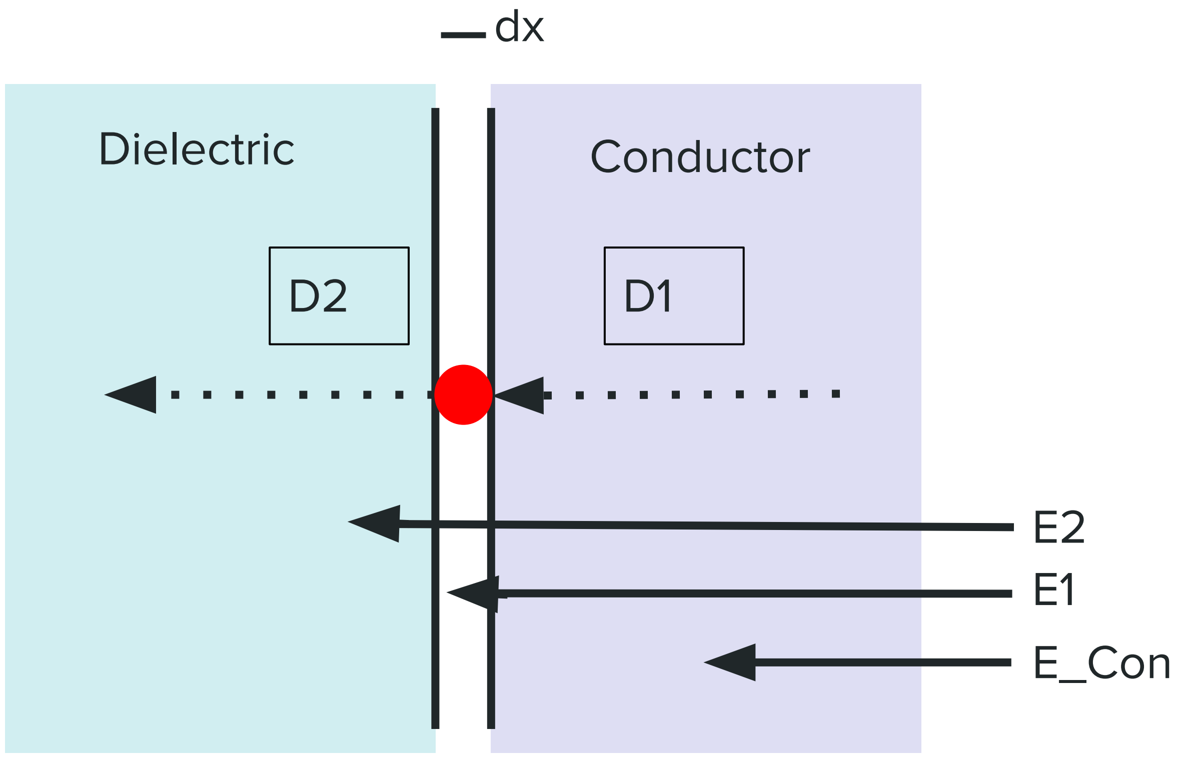
$$D_1 = D_2$$

$$\varepsilon_0 E_1 = D_1$$

$$\varepsilon E_2 = D_2$$

$$\varepsilon_0 E_1 = \varepsilon E_2$$

$$E_1 = \frac{\varepsilon}{\varepsilon_0} E_2$$



Math Tidbits : Deriving Equation for Free Charge

$$\mathrm{D}_1 = \mathrm{D}_2$$

$$\varepsilon_0 E_1 = D_1$$

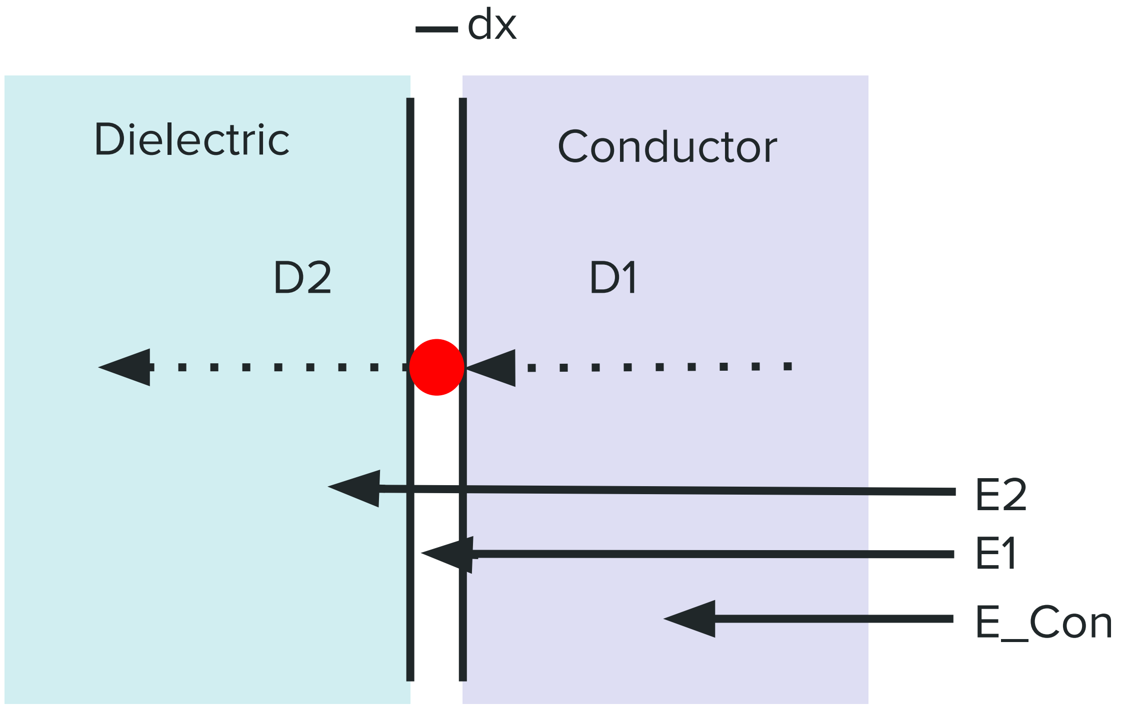
$$\varepsilon E_2 = D_2$$

$$\varepsilon_0 E_1 = \varepsilon E_2$$

$$E_1 = \frac{\varepsilon}{\varepsilon_0} E_2$$

$$E_1 - E_{Con.} = \frac{\sigma_f}{\varepsilon_0}$$

$$E_1 = \frac{\sigma_f}{\varepsilon_0}$$



Applying Boundary Condition

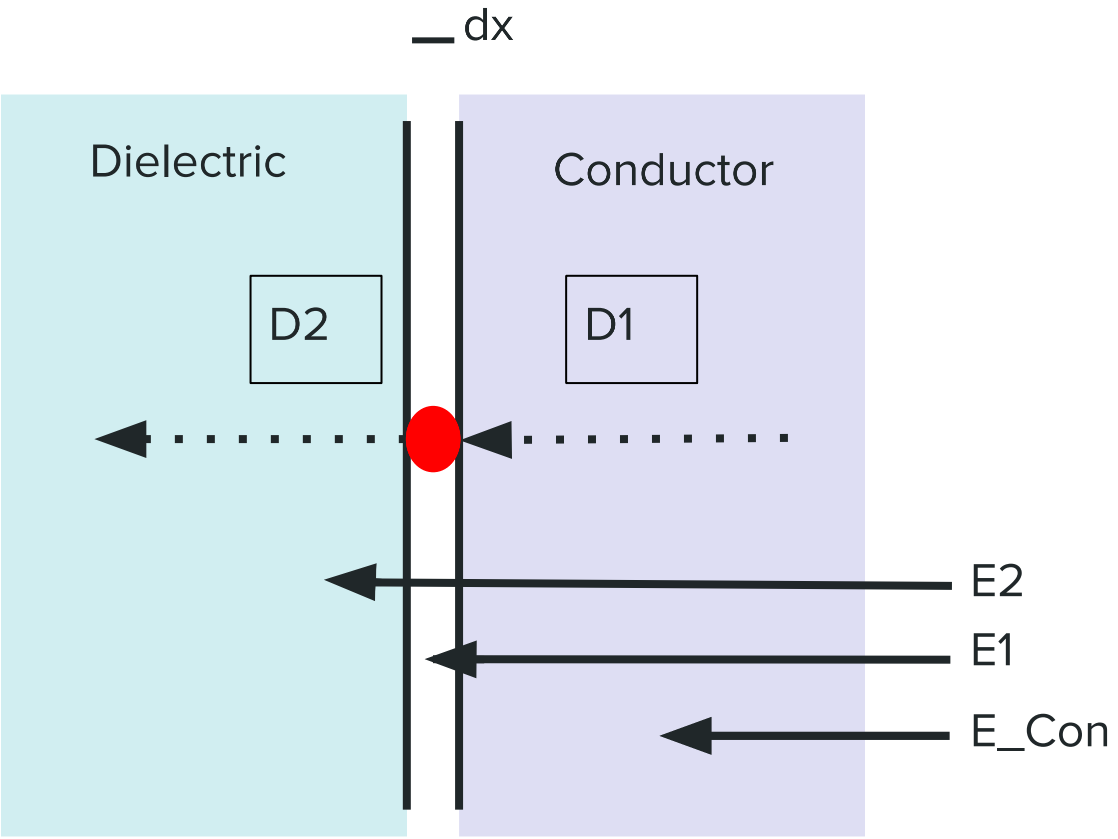
Math Tidbits : Deriving Equation for Free Charge

$$\frac{\sigma_f}{\varepsilon_0} = \frac{\varepsilon}{\varepsilon_0} E_2$$

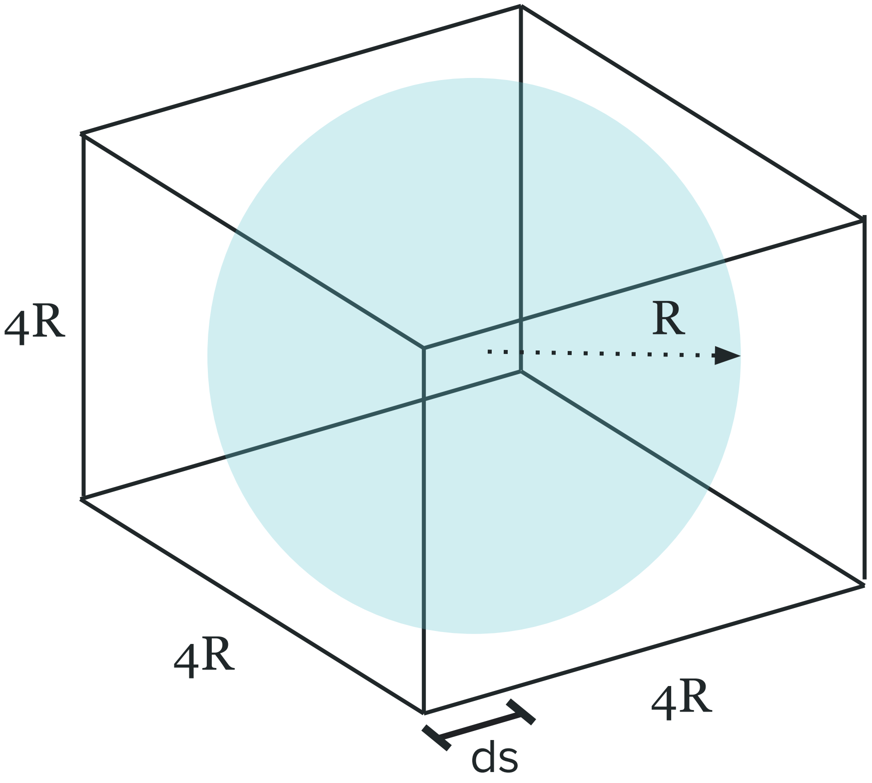
$$\sigma_f = \varepsilon E_2$$

$$\sigma_f = \varepsilon_0 (1 + X_e) E_2$$

E2 will be a known value.



Solution Part One : Encoding Environment



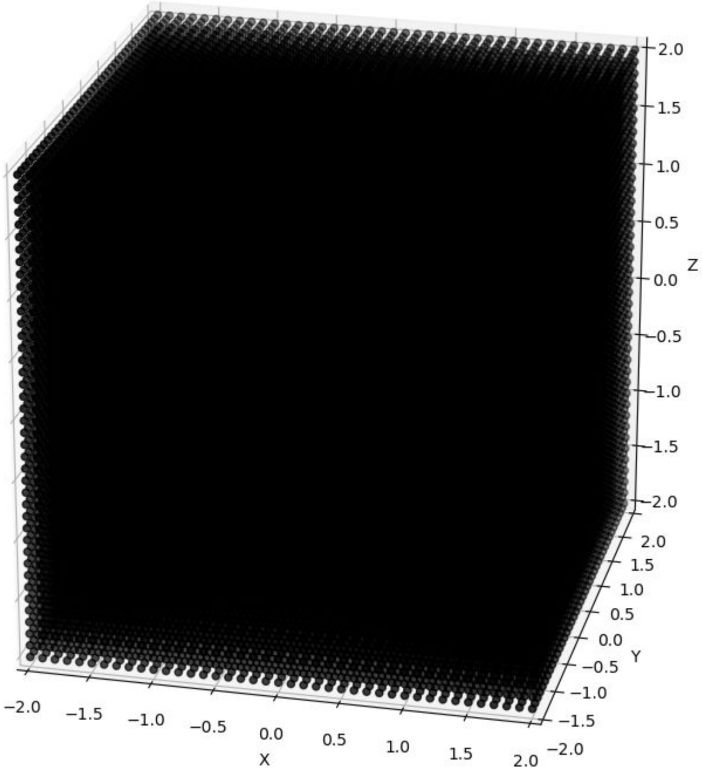
- **Environment:**

np.meshgrid ➔ created X,Y,Z coordinates that ranged that 2R to - 2R in the x, y, and z directions. Step size for meshgrid: Δh or Δs.

- **Dielectric Sphere:**

Used a mask with a conditional statement: $R^2 \geq x^2 + y^2 + z^2$. x, y, and z are points in the environment but we collapse them to the interior of the sphere by making their max values equal to R^2.

Environment

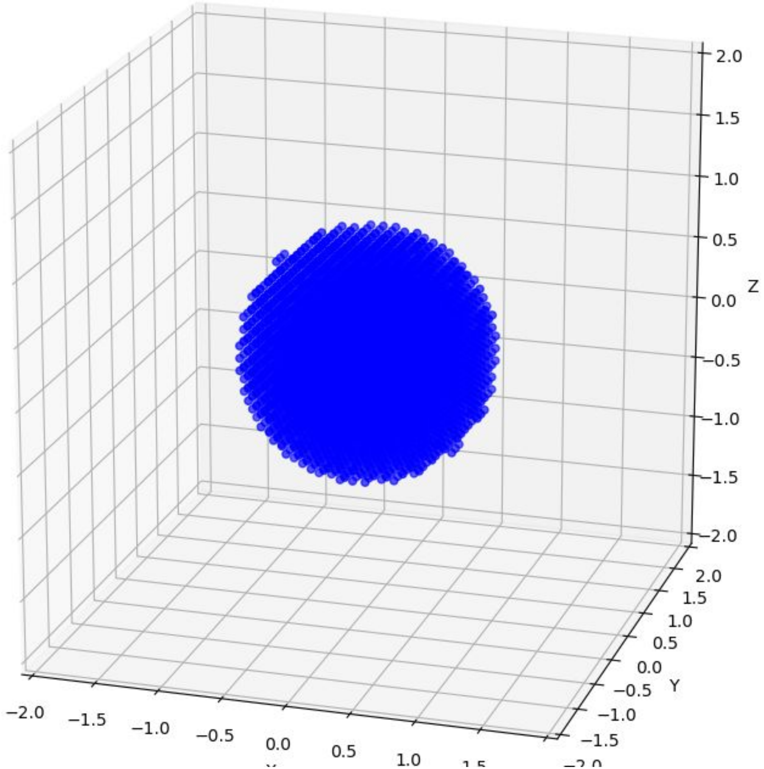



```
R = 1
ds = 0.06
```

```
x = np.arange(-2*R,2*R+ds,ds)
y = np.arange(-2*R,2*R+ds,ds)
z = np.arange(-2*R,2*R+ds,ds)
X,Y,Z = np.meshgrid(x,y,z,indexing='ij')
```

- User inserts R and ds values to determine the size of their simulation.
- Generate X, Y, Z values.
- Plug into meshgrid.

Water: Dielectric Sphere

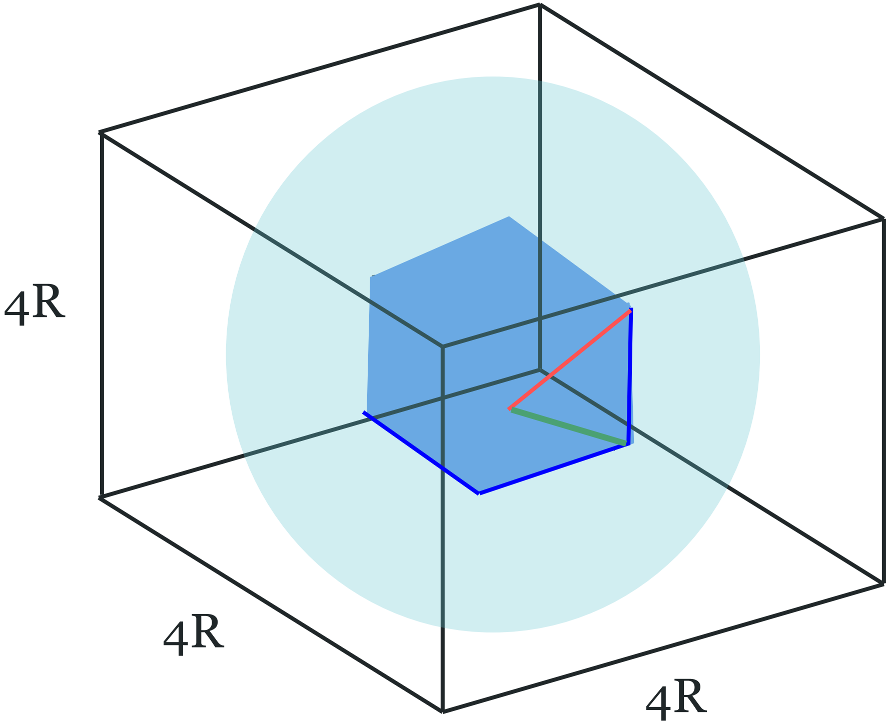



```
perm_index = np.where(X**2 + Y**2 + Z**2 <= R**2)
other_index = np.where(X**2 + Y**2 + Z**2 > R**2)
```

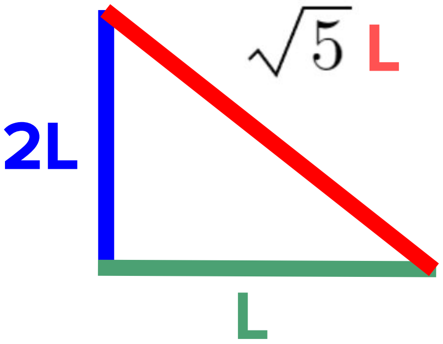
- Define “perm(ittivity) index” using conditional statement previously stated
- Define “other index” to represent all other points in environment.

Solution Part Two : Cube Solution

```python
if l*np.sqrt(5) >= R:
    print('Your chosen sidelength of the cube is larger than the radius of the medium of the water')
    return None
cube_index = np.where((np.abs(X) <= l) & (np.abs(Y) <=l) & (np.abs(Z) <= l))
volt[cube_index] = V
```

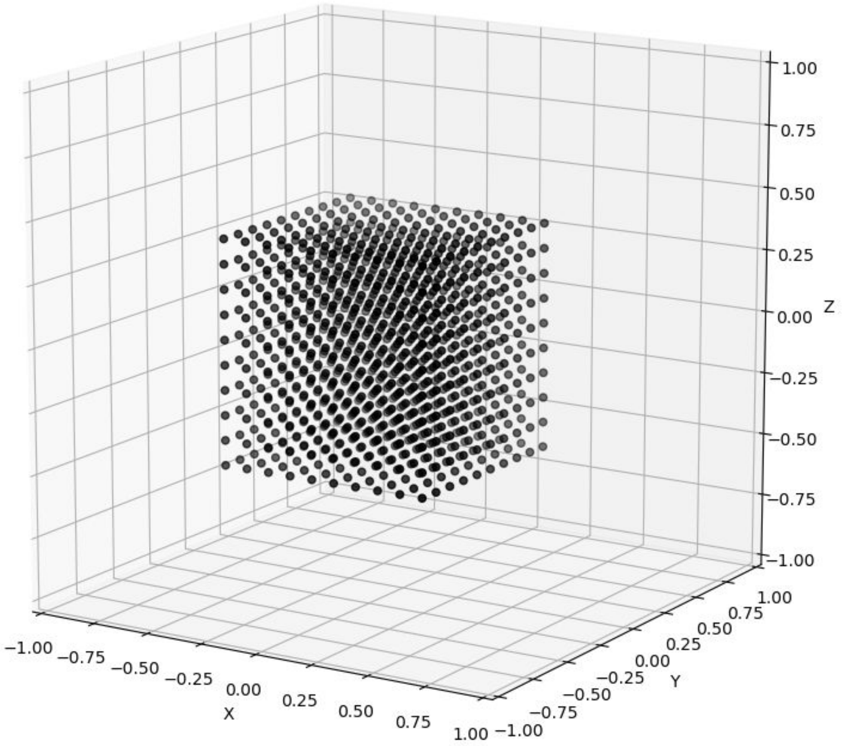


- **Blue** edges have lengths 2L.

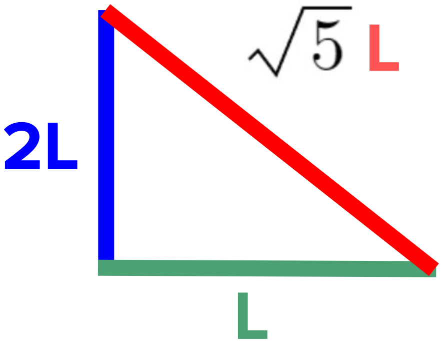


- The length of the cube's body diagonal (sqrt(5) L) should not be larger than the radius of the dielectric sphere.

```python
if l*np.sqrt(5) >= R:
    print('Your chosen sidelength of the cube is larger than the radius of the medium of the water')
    return None
cube_index = np.where((np.abs(X) <= l) & (np.abs(Y) <=l) & (np.abs(Z) <= l))
volt[cube_index] = V
```



- **Blue** edges have lengths 2L.

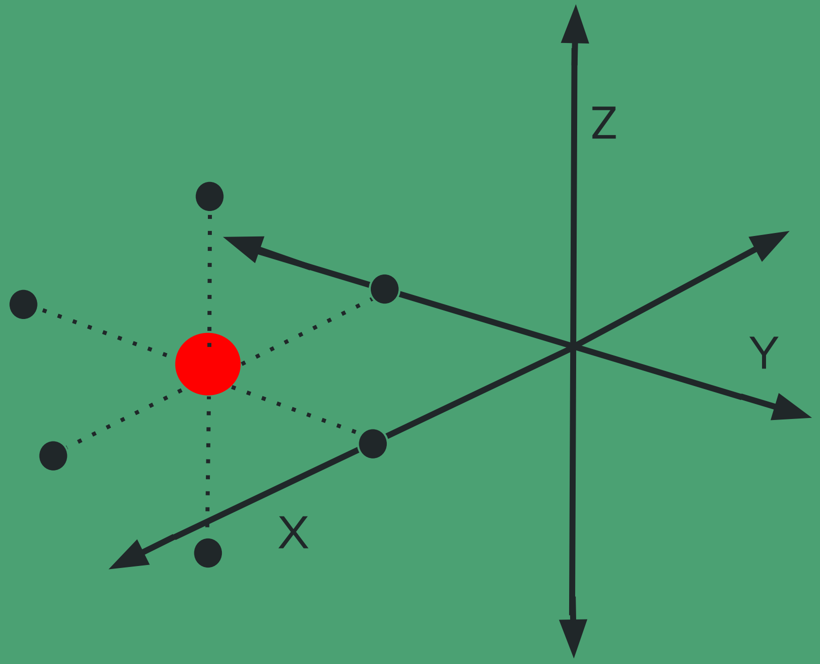


- The length of the cube's body diagonal (sqrt(5) L) should not be larger than the radius of the dielectric sphere.

Method of Relaxation

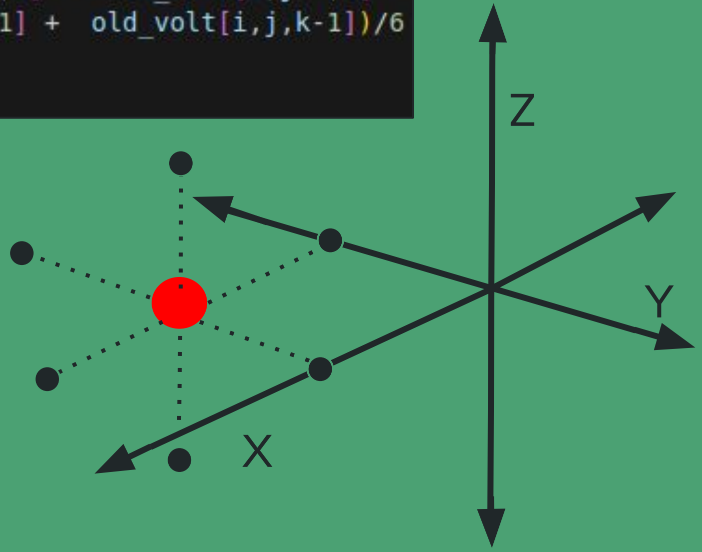
$$V(x',y',z') = \frac{V(x'+\Delta s,y',z')+V(x'-\Delta s,y',z')+V(x',y'+\Delta s,z')+V(x',y'-\Delta s,z')+V(x',y',z'+\Delta s)+V(x',y',z'-\Delta s)}{6}$$

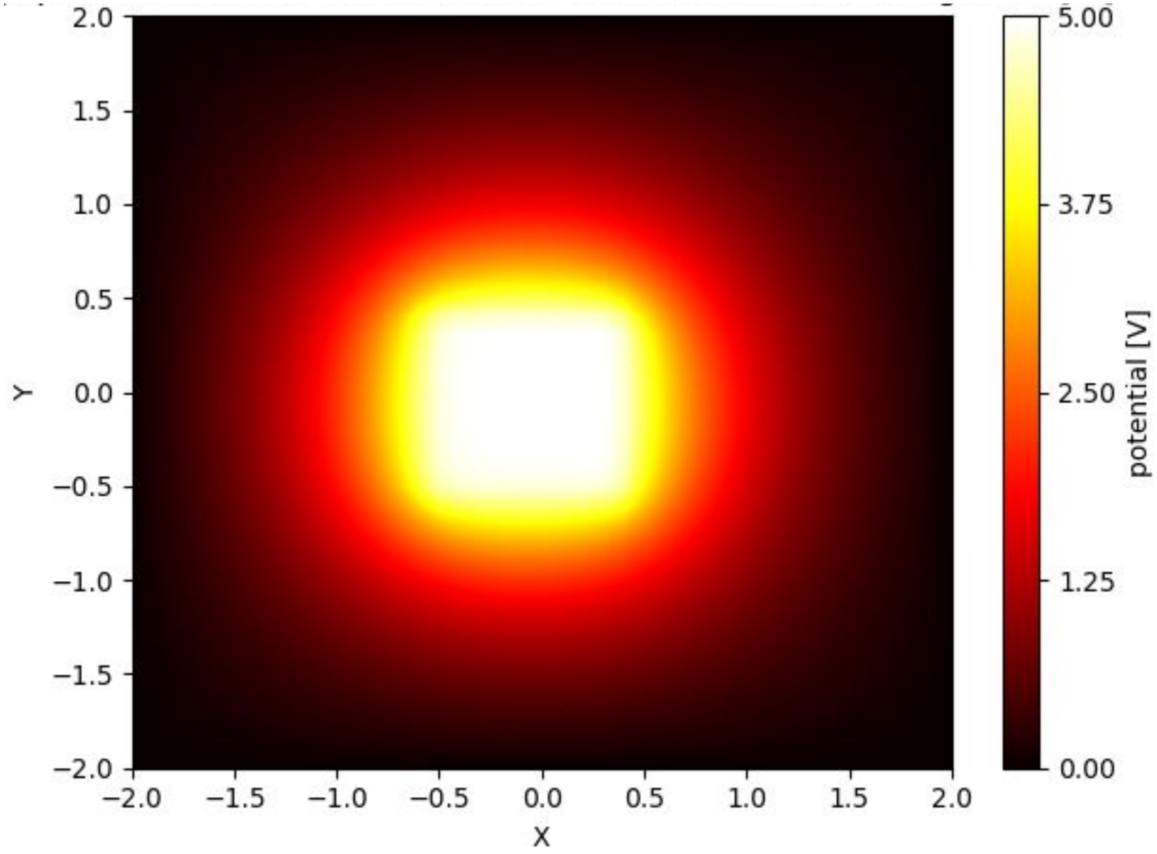
- Average about two points in x, y, and z directions separated by a step size Δs.
- Determines potential at that specific point (denoted by primes) not everywhere in space.
- Must be repeated over all position values.

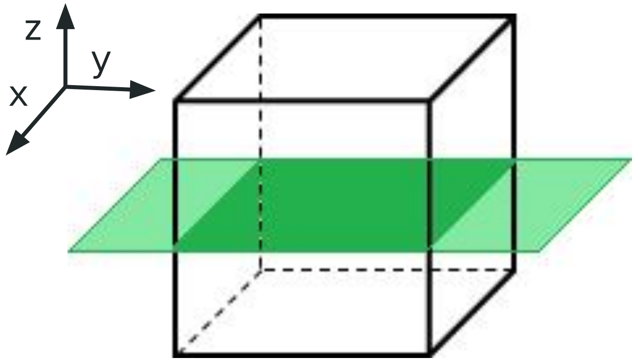


Method of Relaxation

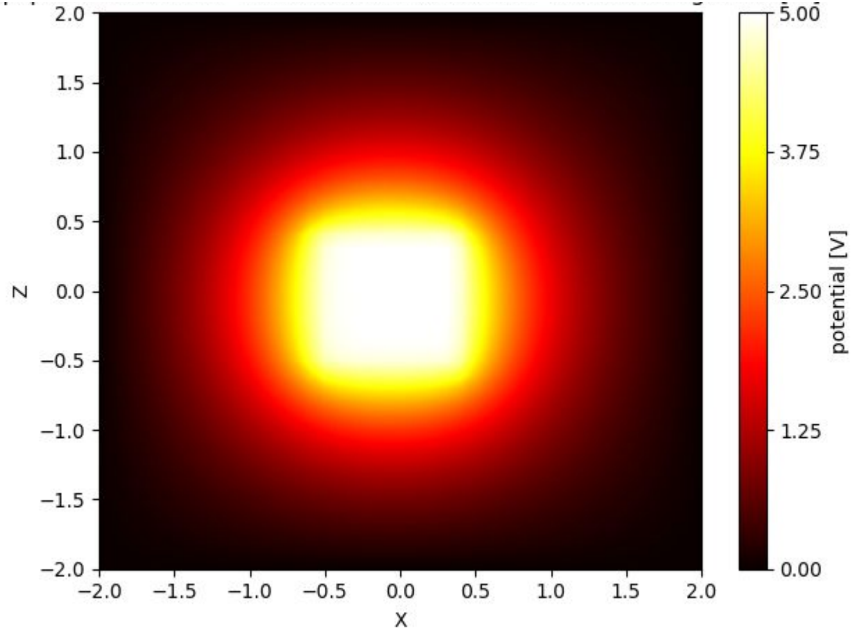
```python
def relax(volt,index, num):
    count = 0
    while(num != count):
        old_volt = volt
        volt[index] = V
        for i in range(1,volt.shape[0]-1):
            for j in range(1,volt.shape[1]-1):
                for k in range(1,volt.shape[2]-1):
                    volt[i,j,k] =  (old_volt[i+1,j,k] +  old_volt[i-1,j,k] +  old_volt[i,j+1,k] +
                                    old_volt[i,j-1,k] + old_volt[i,j,k+1] +  old_volt[i,j,k-1])/6
        count = count + 1
    return volt
```

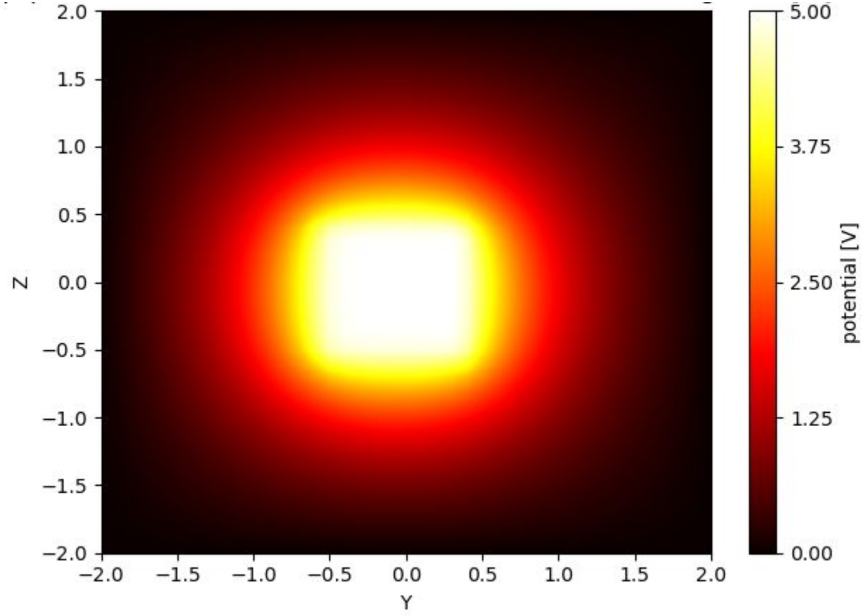






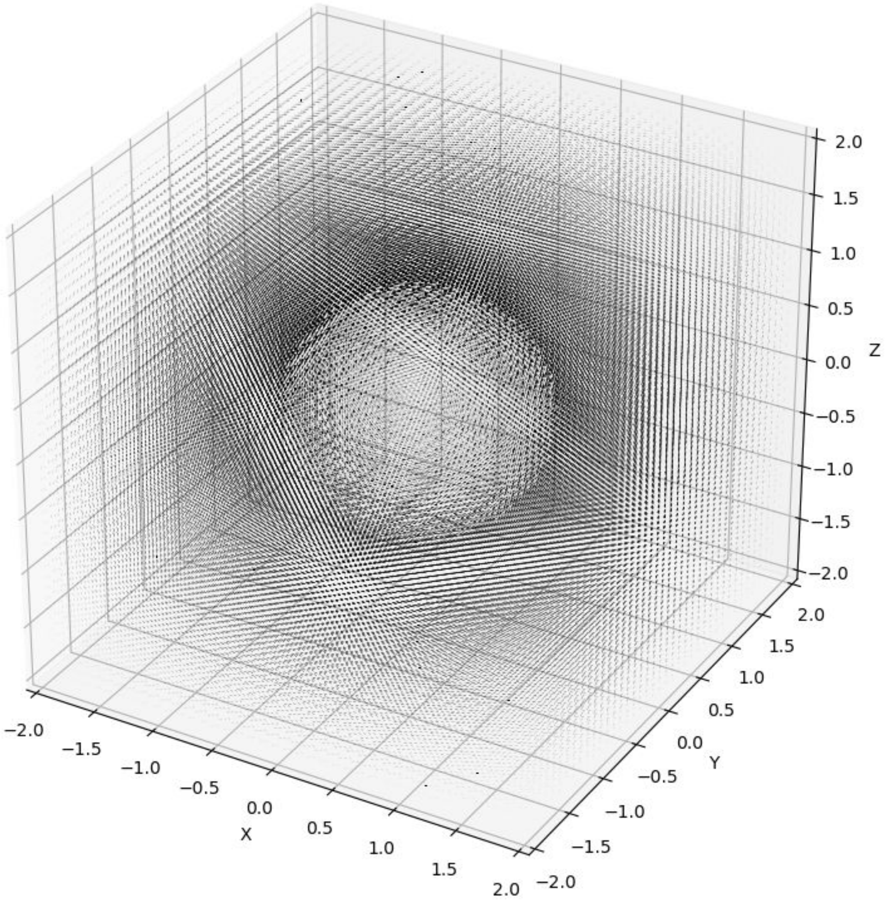
- Potential decreases as radius increases.
- Looks like a point charge from a far distance away.





- Along all axes (X,Y,Z), the potential looks the same which makes physical sense since a cube has three four-fold symmetry axes.

- Potential within cube is the same everywhere as expected from a conductor.



Electric field was generated everywhere in space but it was immensely hard to sense make. INSTEAD, we made cross sectional cuts along regions of interest. (shown in incoming slides)

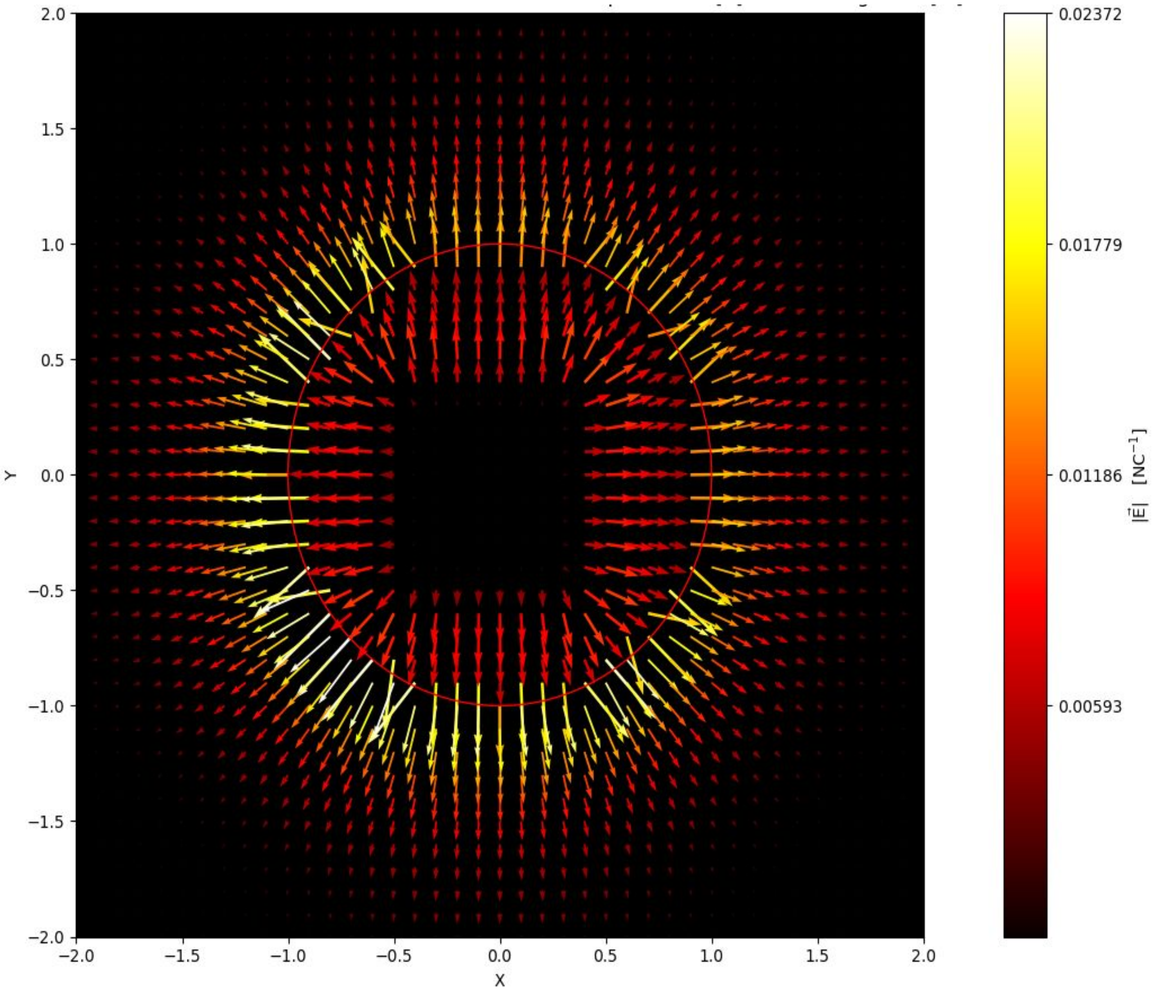
Between Dielectric & Environment:

- The electric field vectors got smaller as the radius value increases.

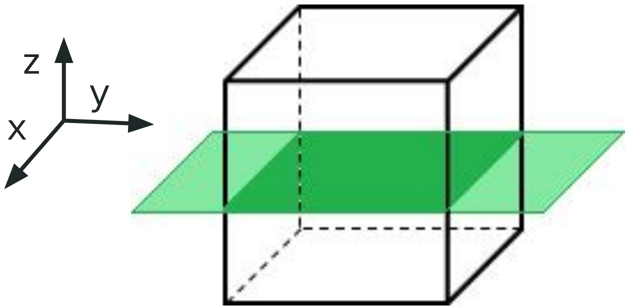
Environment Outwards:

- The electric field vectors got smaller as the radius value increases.

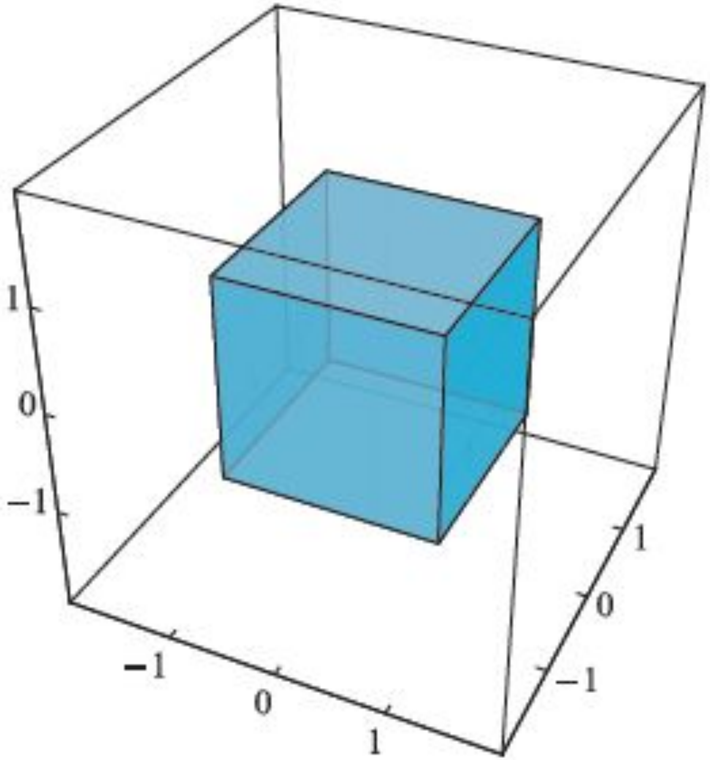
- It is difficult to see the conductor but we expect the conductor to have no electric field vectors (Chpt. 2 Griffiths - Conductors)



- Electric field does not exist within conductor
- Dielectric severely weakens electric field as r increases.
- Electric field gets stronger at dielectric/environment barrier and then decreases as r approaches infinity.



Determining Free & Inner Bound Charge Density Computationally



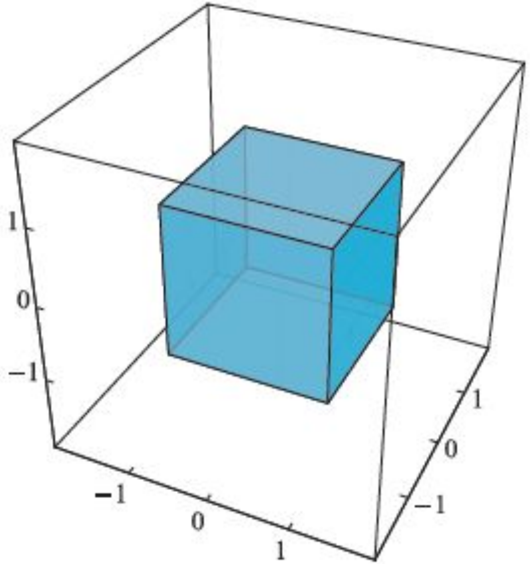
- Mask **(Blue)** used to outline position of conductor can be expanded by a step size to access value right above or below its surface. This will be a new mask **(Black)**.

- Subtract **Blue** mask away from **Black** mask to access position, potential, electric field, and X_e values at *just* the surface/boundary.

- Since we are on the surface we can assume the electric field vectors are normal.

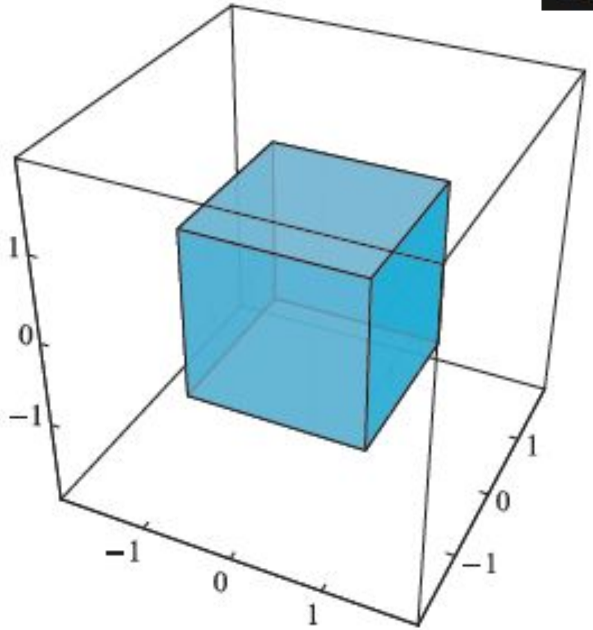
Determining Free & Inner Bound Charge Density Computationally

```
pos_cube_mask = np.zeros((x.size,y.size,z.size),dtype = bool)
pos_cube_mask[cube_index] = True
pos_cube_mask[1:-1,1:-1,1:-1] = (pos_cube_mask[2:,1:-1,1:-1] | pos_cube_mask[0:-2,1:-1,1:-1]
| pos_cube_mask[1:-1,2:,1:-1] | pos_cube_mask[1:-1,0:-2,1:-1] | pos_cube_mask[1:-1,1:-1,2:]
| pos_cube_mask[1:-1,1:-1,0:-2])
pos_cube_mask[cube_index] = False
```



- Fill cube mask with zeros.
- Extend said masks by 1 [Just ensure the cube is attached at all edges].
- Subtract out original mask.
- Yield surface of conductor or dielectric.

Determining Free & Inner Bound Charge Density Computationally




```
x_E_cube_bound_in = x_E_cube_in[pos_cube_mask]
y_E_cube_bound_in = y_E_cube_in[pos_cube_mask]
z_E_cube_bound_in = z_E_cube_in[pos_cube_mask]
mag_E_cube_bound_in_one = np.sqrt(x_E_cube_bound_in**2 +
                                  y_E_cube_bound_in**2 + z_E_cube_bound_in**2)
sig_cube_b = mag_E_cube_bound_in_one*e_0*x_e
```

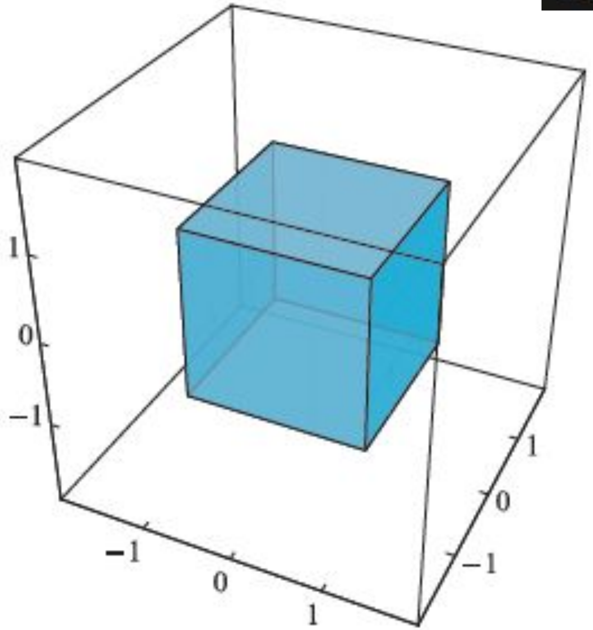
$$\sigma_{\mathrm{Bound}} = \mathrm{E}\varepsilon_o \mathrm{X_e} \cdot \mathrm{n_{hat}}$$

- Take split up x, y, z directions for bound charge.
- Determine E value by calculating magnitude.
- Multiply by e_o and X_e.
- Reminder: Assuming electric field is pointing in the norma direction to the conductor's surface.

Determining Free & Inner Bound Charge Density Computationally

```
x_E_cube_bound_in = x_E_cube_in[pos_cube_mask]
y_E_cube_bound_in = y_E_cube_in[pos_cube_mask]
z_E_cube_bound_in = z_E_cube_in[pos_cube_mask]
mag_E_cube_bound_in_one = np.sqrt(x_E_cube_bound_in**2 +
                                  y_E_cube_bound_in**2 + z_E_cube_bound_in**2)
sig_cube_b = mag_E_cube_bound_in_one*e_0*x_e
```

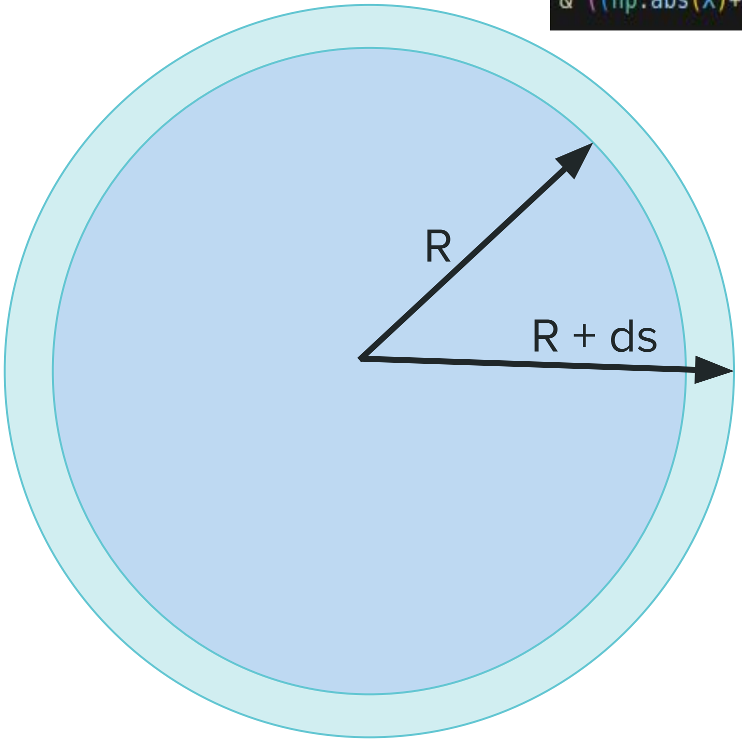
$$\sigma_{\mathrm{Bound}} = \mathrm{E}\varepsilon_o \mathrm{X_e} \cdot \mathrm{n_{hat}}$$



- Take split up x, y, z directions for bound charge.
- Determine E value by calculating magnitude.
- Multiply by e_o and X_e.
- Reminder: Assuming electric field is pointing in the norma
direction to the conductor's surface.

Computationally Determining Outer Bound Charge

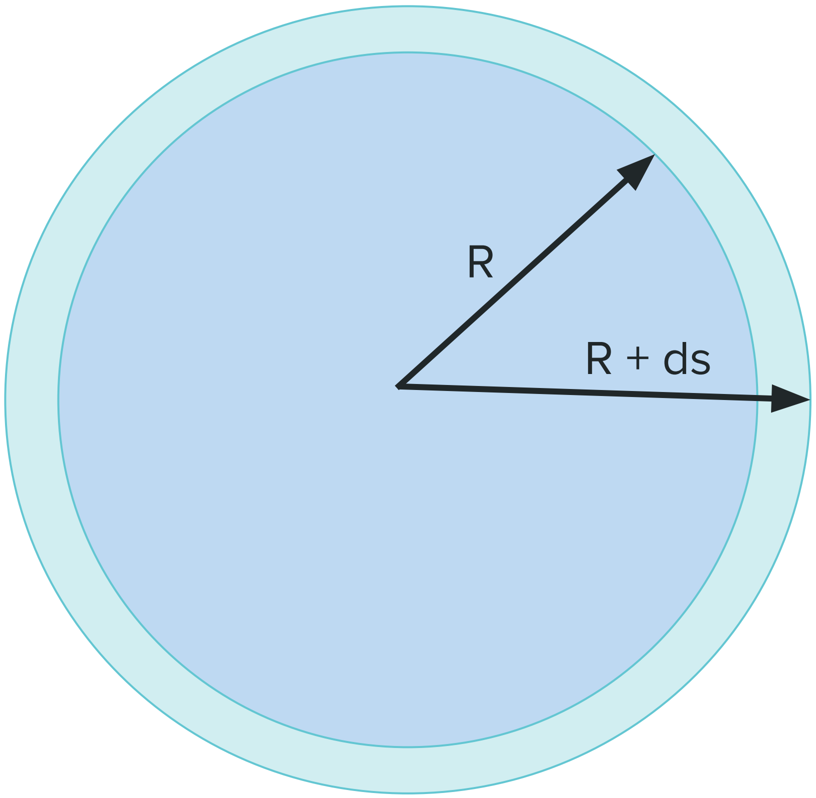
```
scale = 2
sphere_perm = np.where(((np.abs(X))**2 + (np.abs(Y))**2 + (np.abs(Z))**2 < (R)**2)
& ((np.abs(X)+scale*ds)**2 + (np.abs(Y)+scale*ds)**2 + (np.abs(Z)+scale*ds)**2 > (R+scale*ds)**2))
```



- Similar process as with cube for free surface charge on conductor or inner bound charge on dielectric.

- Replace cube with sphere.

Computationally Determining Outer Bound Charge



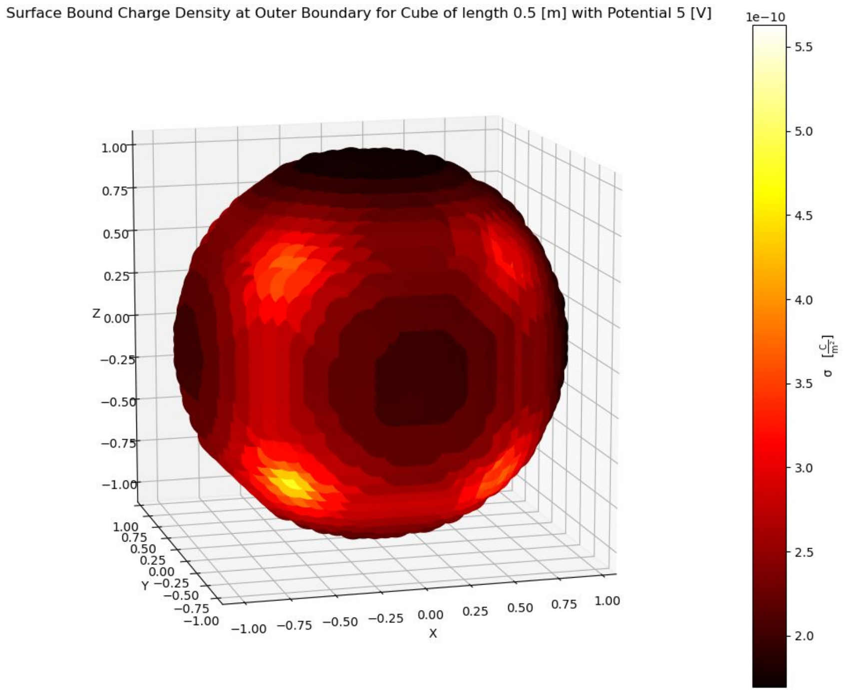

```
x_E_cube_bound_out = x_E_cube_out[sphere_perm]
y_E_cube_bound_out = y_E_cube_out[sphere_perm]
z_E_cube_bound_out = z_E_cube_out[sphere_perm]
dot_x = x_E_cube_bound_out * X[sphere_perm]
dot_y = y_E_cube_bound_out * Y[sphere_perm]
dot_z = z_E_cube_bound_out * Z[sphere_perm]
sig_cube_out_b = (e_0*x_e/R) * (dot_x + dot_y + dot_z)

sig_cube_f = norm_E_cube_bound_in_one*e_0
```

$$\sigma_{Bound} = E\varepsilon_o\chi_e \cdot r_{hat}$$

$$\sigma_{Bound} = E\varepsilon_o\chi_e \cdot \left(\frac{\mathrm{x}}{\mathrm{R}}\mathrm{x}_{hat} + \frac{\mathrm{y}}{\mathrm{R}}y_{hat} + \frac{\mathrm{z}}{\mathrm{R}}z_{hat}\right)$$

- No longer assume E is normal, must be in r_hat direction. Converted into cartesian to match with environment.

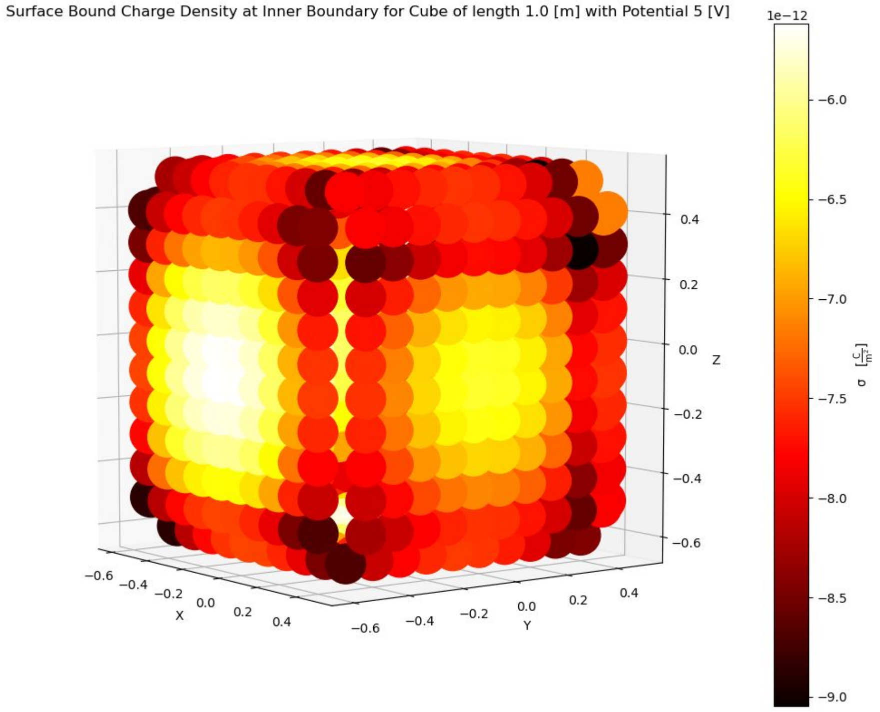


- The dielectric has a larger bound charge distribution at the same location as the corners from the solid cube.

- The spherical symmetry of the dielectric should cause the bound surface charge to look relatively uniform.

- Limitations caused by creating discrete points. Accuracy of charge distribution is inhibited.

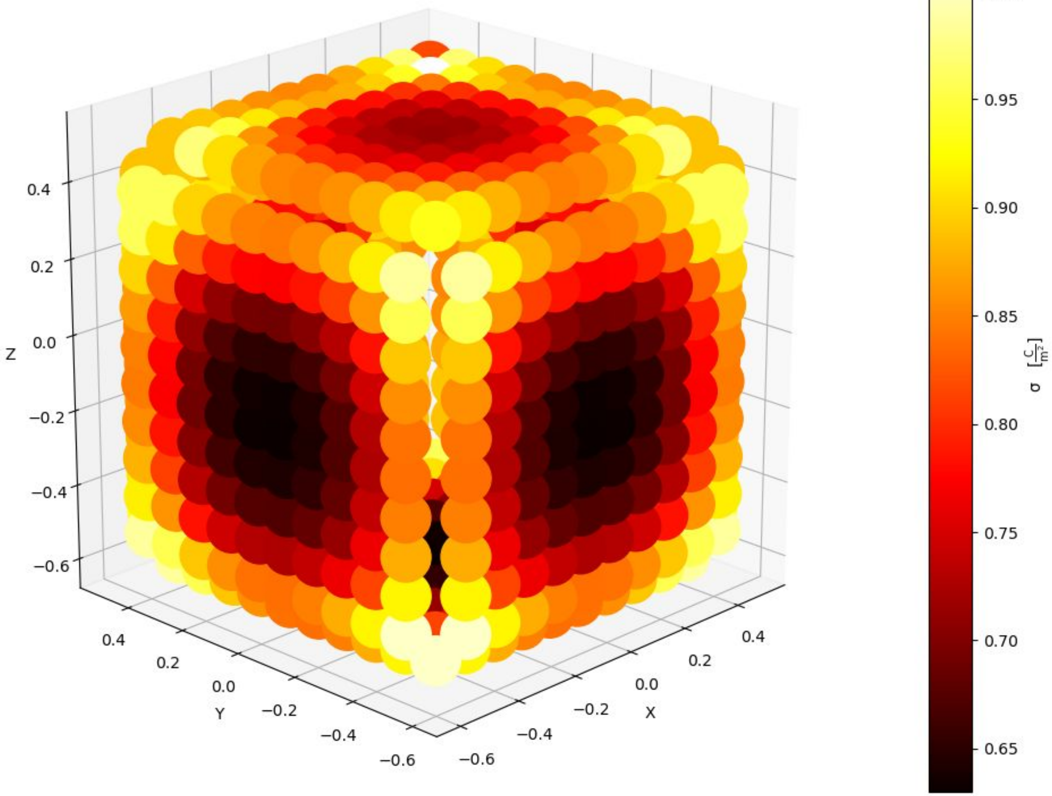
Determining Inner Bound Charge Density Computationally



- Corners of cube have the largest bound charge. Edges of cube have larger density then the center of the faces.
- Center of the faces have uniform charge density
  - Agrees with how we expect charges to gather on a non-spherical object.

Determining Free Charge Density Computationally

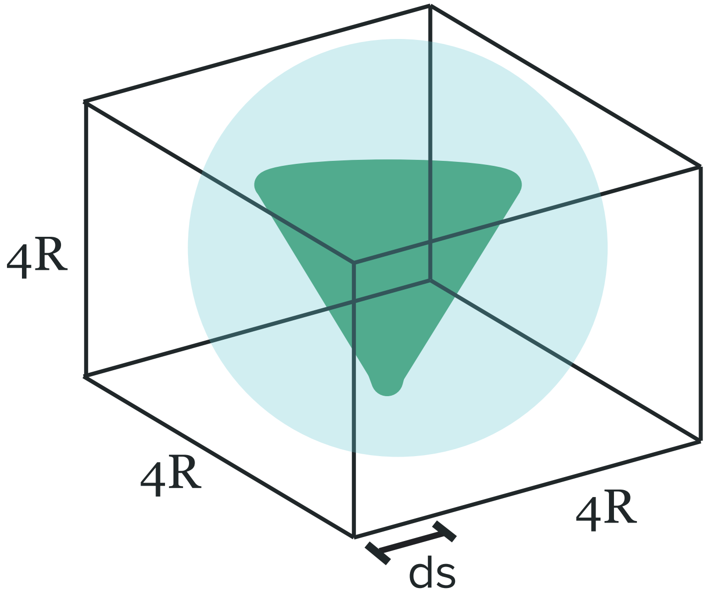
```
sig_cube_f = norm_E_cube_bound_in_one*e_0
```



- Consistent with Inner Bound Surface Charge 3D Model.
- Corners and edges have most intense charge distribution.
- Surfaces have minimal charges.
- Positive charges at edges.

Solution Part Three : Cone Solution

```python
def cond_cone(R,h,volt):
    if h >= R:
        print('Your chosen height of the cone is larger than the the radius of the medium of the water')
        return None
    rr = (h*np.sqrt(R**2-h**2)) / (R*np.cos(np.arctan(np.sqrt(R**2-h**2)/h)))
    rr = 0.5*rr
    cone_index = np.where((Z >= 0) & (Z <=h) &
                          (X**2 + Y**2 <= R**2 - h**2) & (X**2 + Y**2 <= (Z*rr/h)**2))
    volt[cone_index] = V
    return cone_index, volt, rr
```



$$x^2 + y^2 = R^2 - z'^2$$

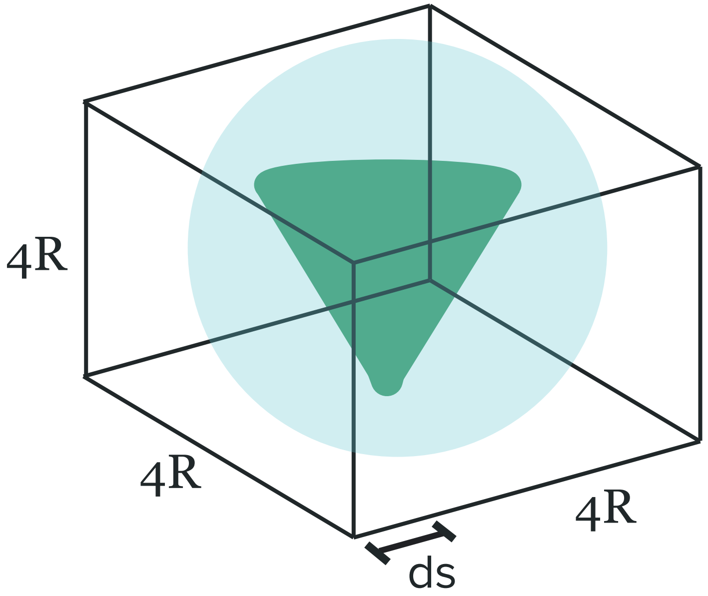
$$-h \leq z' \leq h$$

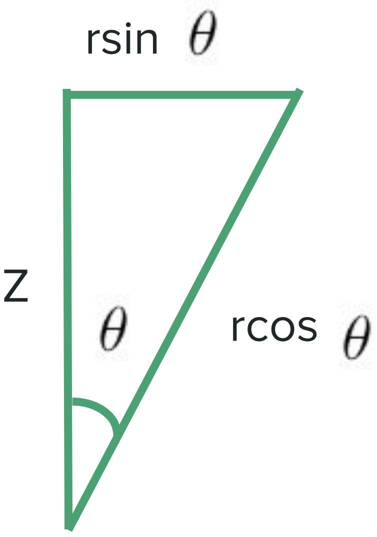
Putting restrictions on cylinder

$$x^2 + y^2 = \frac{z^2}{h^2} r^2$$

Cone equation we will map onto cylinder equation above

```python
def cond_cone(R,h,volt):
    if h >= R:
        print('Your chosen height of the cone is larger than the the radius of the medium of the water')
        return None
    rr = (h*np.sqrt(R**2-h**2)) / (R*np.cos(np.arctan(np.sqrt(R**2-h**2)/h)))
    rr = 0.5*rr
    cone_index = np.where((Z >= 0) & (Z <=h) &
                          (X**2 + Y**2 <= R**2 - h**2) & (X**2 + Y**2 <= (Z*rr/h)**2))
    volt[cone_index] = V
    return cone_index, volt, rr
```

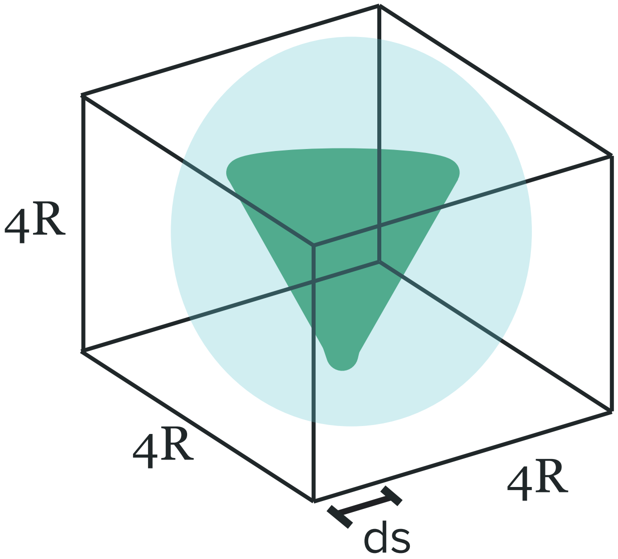


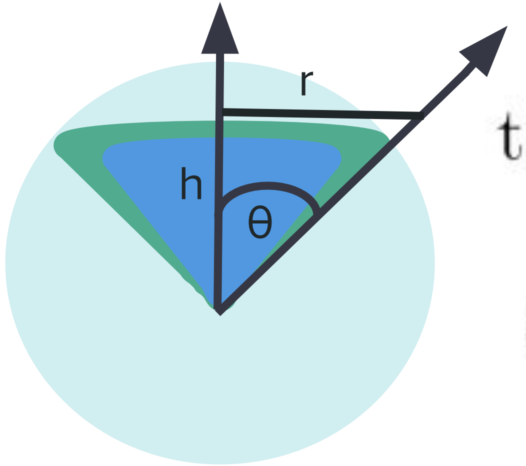


$$\tan\theta = \frac{s}{z} = \frac{\sqrt{x^2 + y^2}}{h}$$

- Ratio between arbitrary height and radial value with cap at the R/h ratio (set in previous slide).

```python
def cond_cone(R,h,volt):
    if h >= R:
        print('Your chosen height of the cone is larger than the the radius of the medium of the water')
        return None
    rr = (h*np.sqrt(R**2-h**2)) / (R*np.cos(np.arctan(np.sqrt(R**2-h**2)/h)))
    rr = 0.5*rr
    cone_index = np.where((Z >= 0) & (Z <=h) &
                          (X**2 + Y**2 <= R**2 - h**2) & (X**2 + Y**2 <= (Z*rr/h)**2))
    volt[cone_index] = V
    return cone_index, volt, rr
```



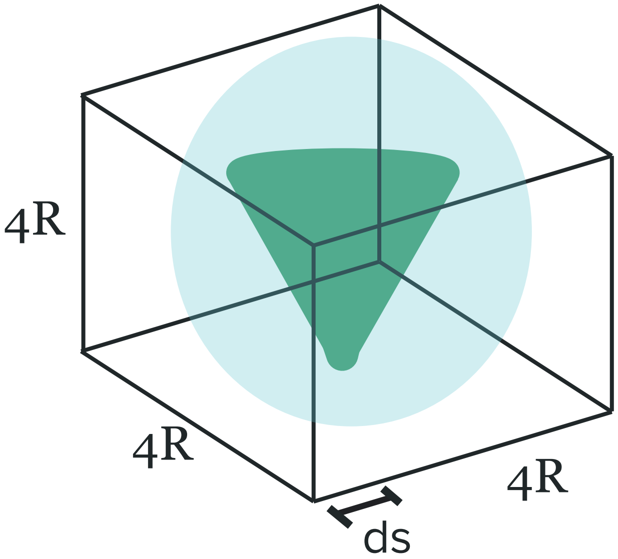


$$\tan(\theta) = \frac{\sqrt{R^2 - z'^2}}{h}$$

$$\theta = \tan^{-1}\left(\frac{\sqrt{R^2 - z'^2}}{h}\right)$$

θ directly depends on h and vice-versa

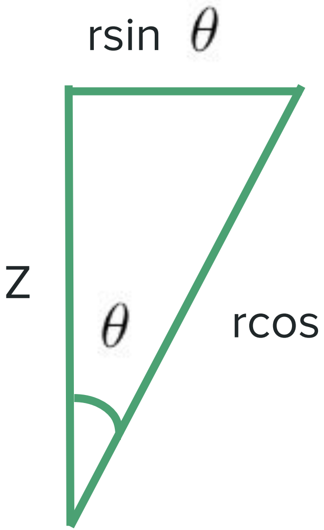
```python
def cond_cone(R,h,volt):
    if h >= R:
        print('Your chosen height of the cone is larger than the the radius of the medium of the water')
        return None
    rr = (h*np.sqrt(R**2-h**2)) / (R*np.cos(np.arctan(np.sqrt(R**2-h**2)/h)))
    rr = 0.5*rr
    cone_index = np.where((Z >= 0) & (Z <=h) &
                          (X**2 + Y**2 <= R**2 - h**2) & (X**2 + Y**2 <= (Z*rr/h)**2))
    volt[cone_index] = V
    return cone_index, volt, rr
```



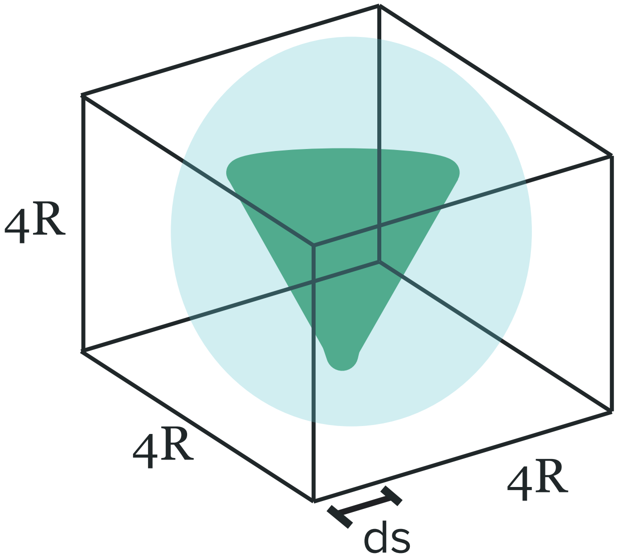
$$x^2 + y^2 = \frac{z^2}{h^2} r^2$$

$$R^2 + z'^2 = \frac{R^2 \cos^2(\tan^{-1}\left(\frac{\sqrt{R^2 - z'^2}}{h}\right))}{h^2} r^2$$

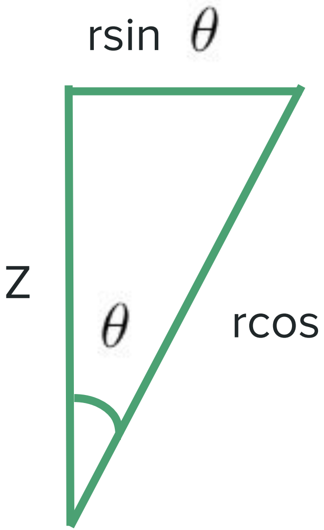
$$R^2 + h^2 = \frac{R^2 \cos^2\left(\tan^{-1}\left(\frac{\sqrt{R^2 - z'^2}}{h}\right)\right)}{h^2} r^2$$



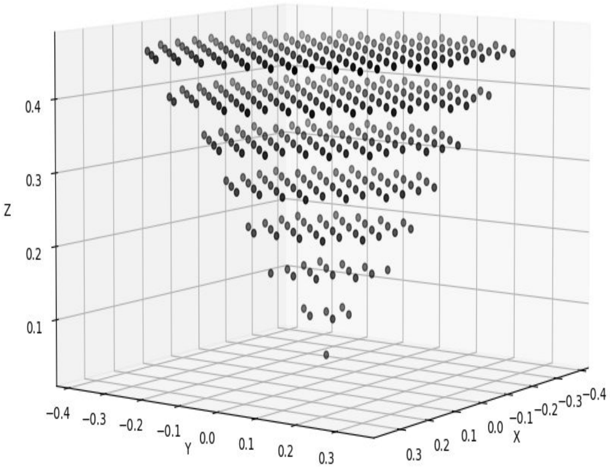
```python
def cond_cone(R,h,volt):
    if h >= R:
        print('Your chosen height of the cone is larger than the the radius of the medium of the water')
        return None
    rr = (h*np.sqrt(R**2-h**2)) / (R*np.cos(np.arctan(np.sqrt(R**2-h**2)/h)))
    rr = 0.5*rr
    cone_index = np.where((Z >= 0) & (Z <=h) &
                          (X**2 + Y**2 <= R**2 - h**2) & (X**2 + Y**2 <= (Z*rr/h)**2))
    volt[cone_index] = V
    return cone_index, volt, rr
```



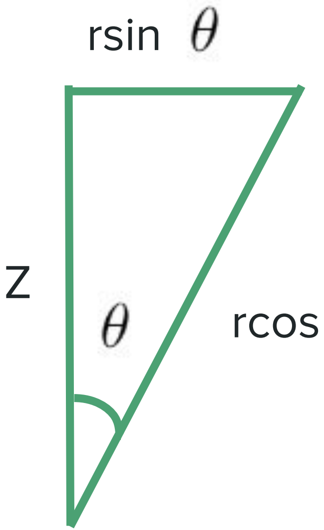
$$\mathrm{r} = \frac{h\sqrt{R^2 - h^2}}{R\cos(\tan^{-1}\left(\frac{\sqrt{R^2 - z'^2}}{h}\right))}$$

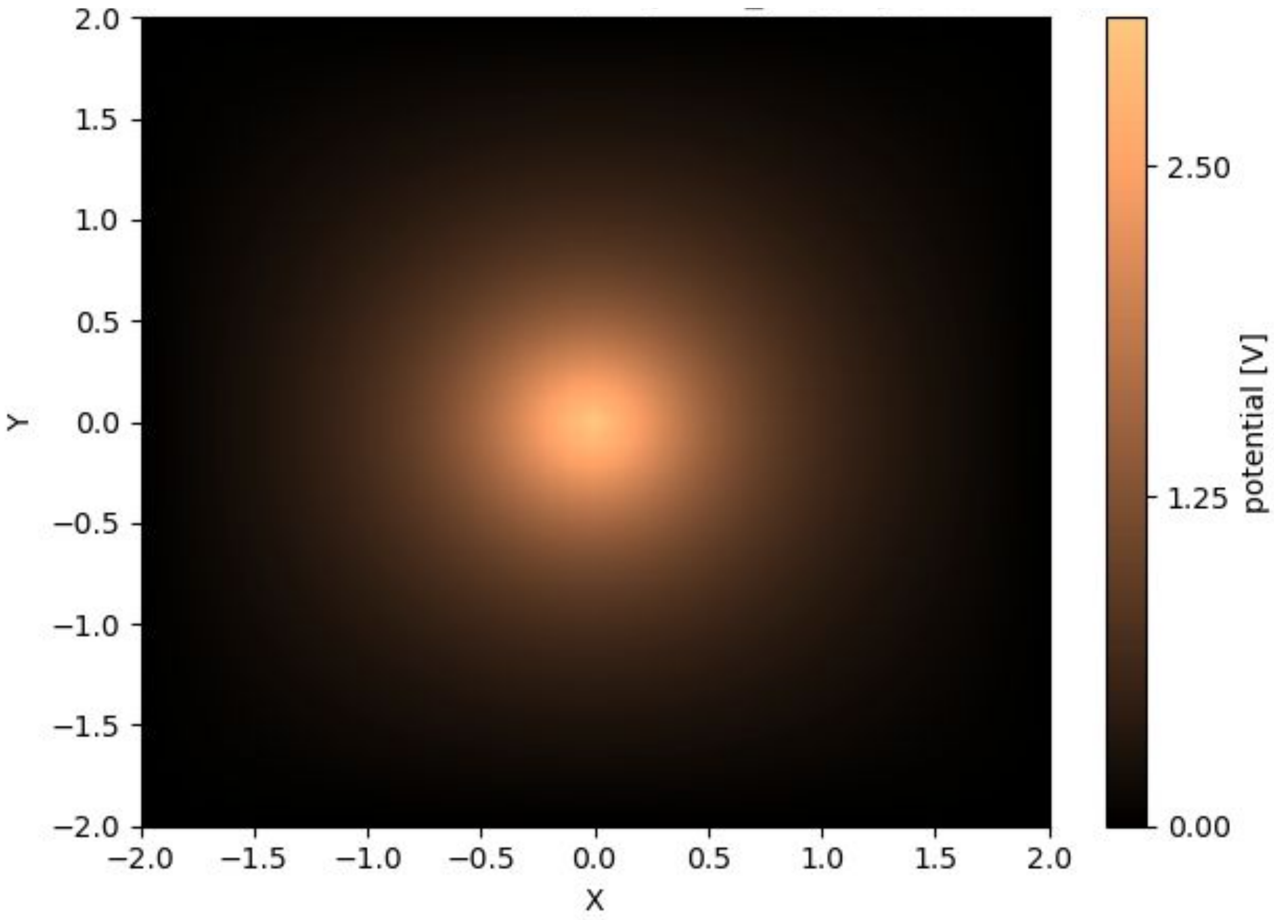


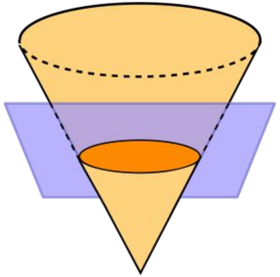
```python
def cond_cone(R,h,volt):
    if h >= R:
        print('Your chosen height of the cone is larger than the the radius of the medium of the water')
        return None
    rr = (h*np.sqrt(R**2-h**2)) / (R*np.cos(np.arctan(np.sqrt(R**2-h**2)/h)))
    rr = 0.5*rr
    cone_index = np.where((Z >= 0) & (Z <=h) &
                          (X**2 + Y**2 <= R**2 - h**2) & (X**2 + Y**2 <= (Z*rr/h)**2))
    volt[cone_index] = V
    return cone_index, volt, rr
```



$$r = \frac{h\sqrt{R^2 - h^2}}{R\cos\left(\tan^{-1}\left(\frac{\sqrt{R^2 - z'^2}}{h}\right)\right)}$$

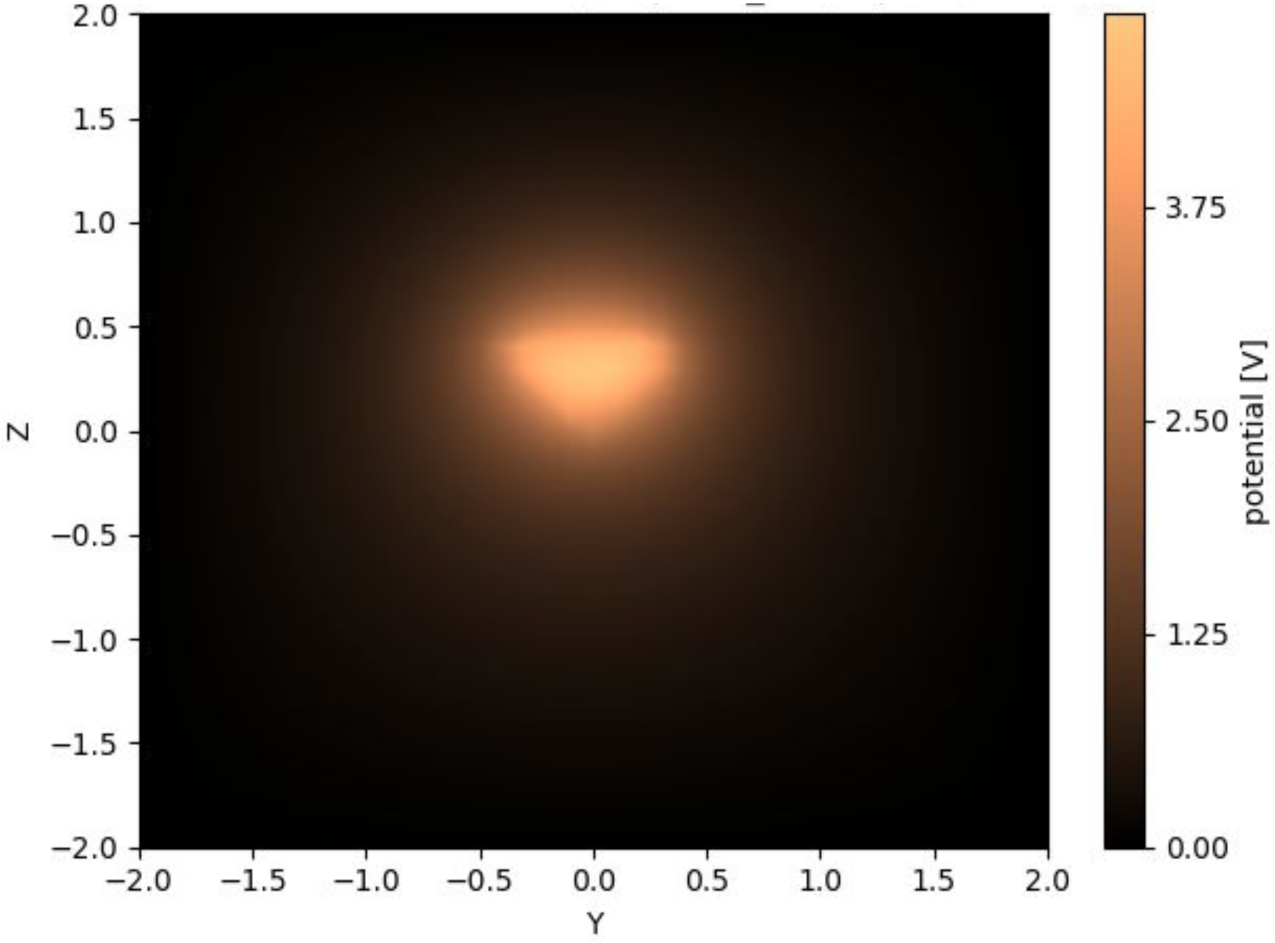




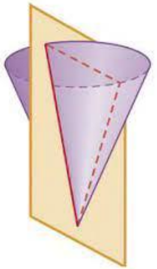


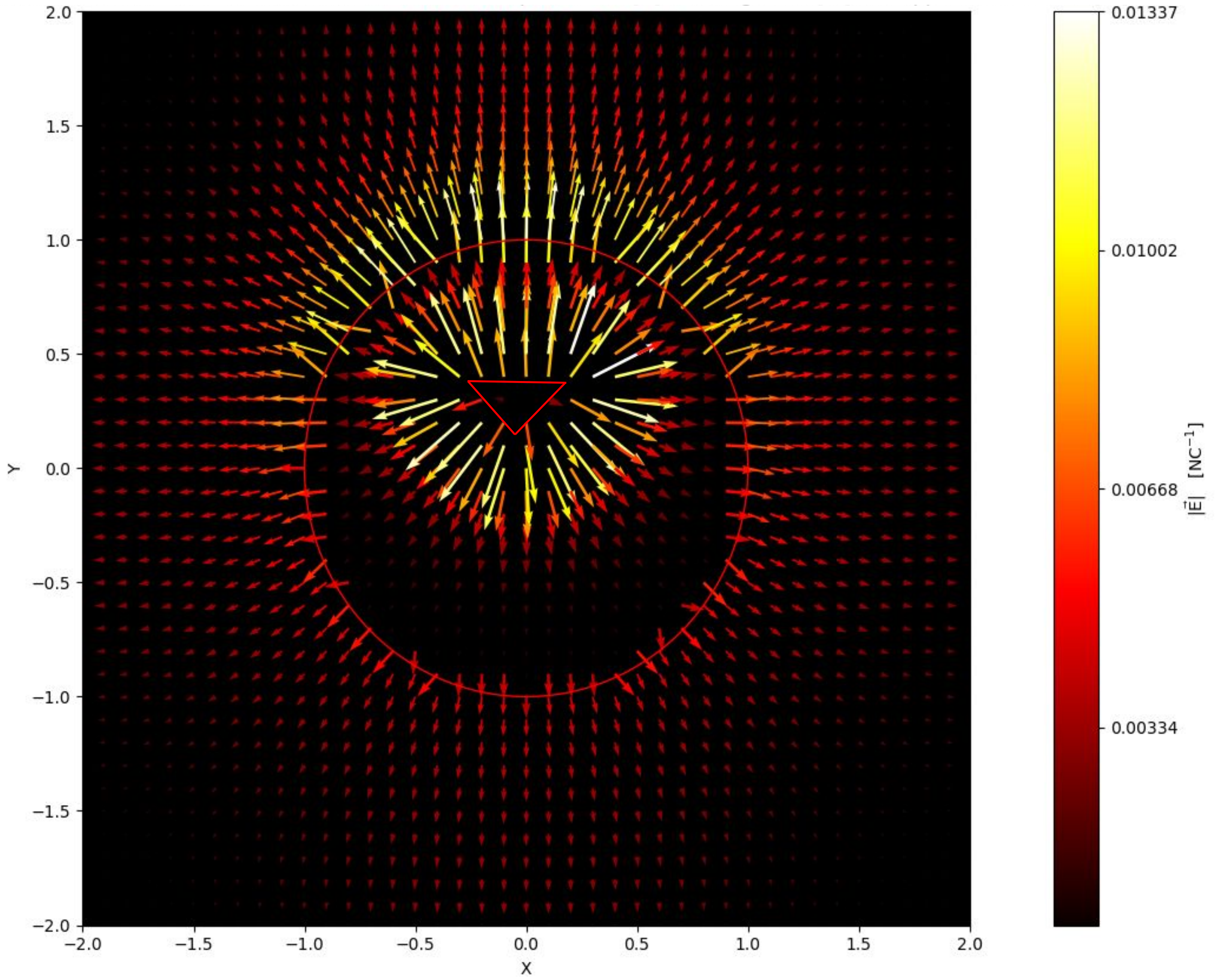
- Electric potential is constant throughout solid section

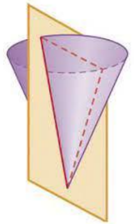
- Electric potential decreases as r gets larger/approaches to infinity.



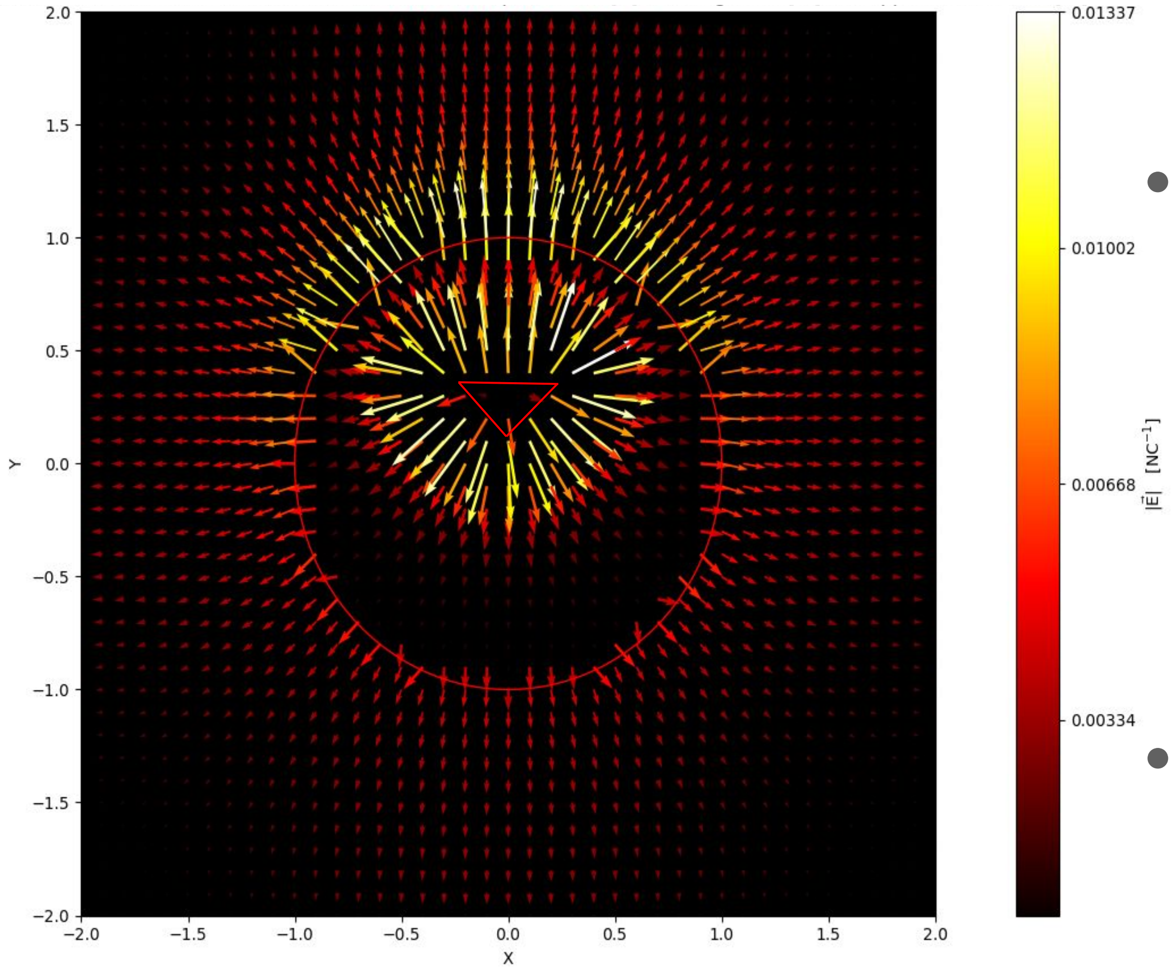
- Electric potential is constant throughout solid section
- Electric potential decreases as r gets larger/approaches to infinity.
- Both cross sections through the central axis creates the same potential.

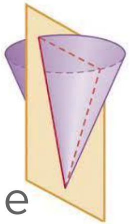




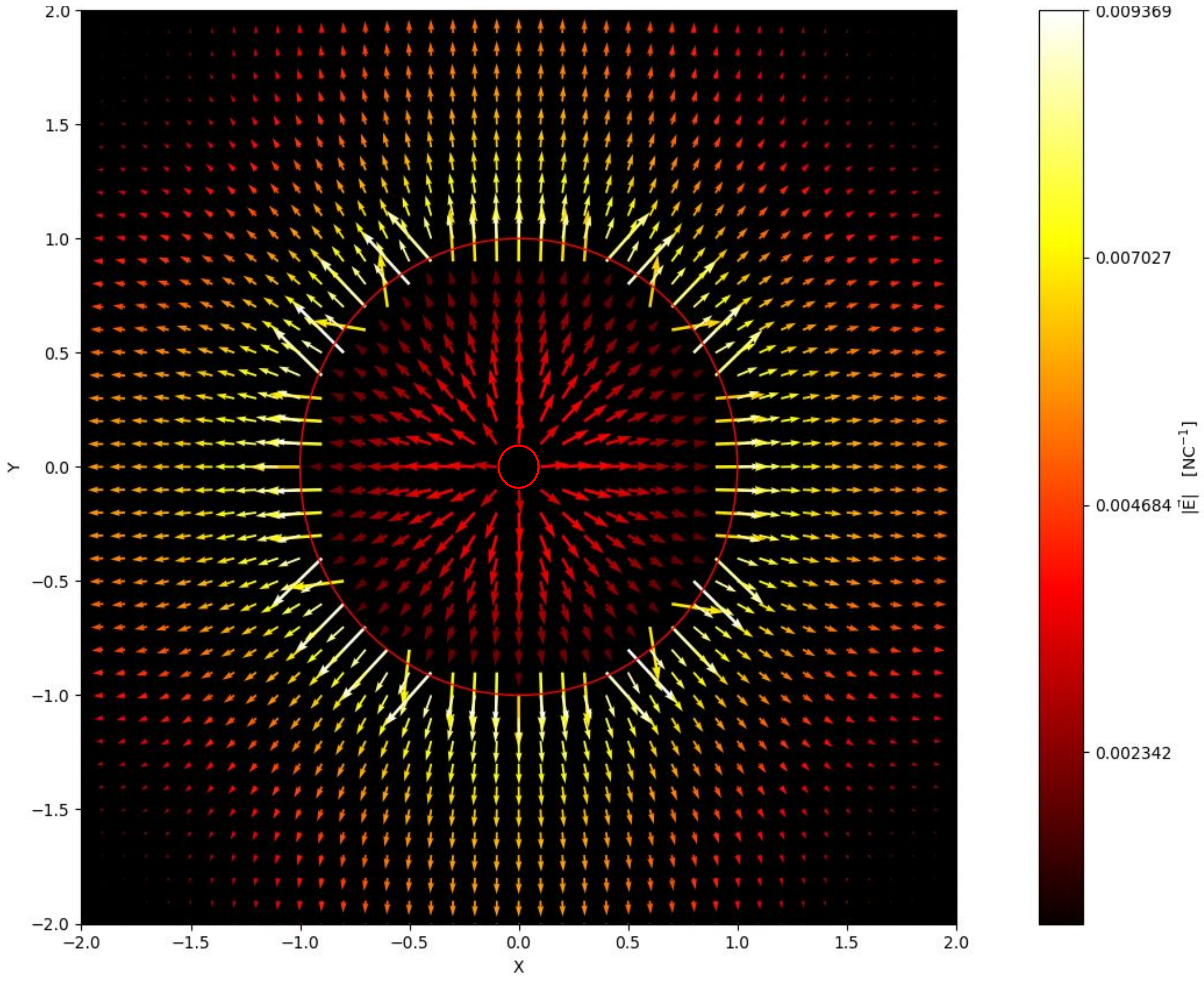


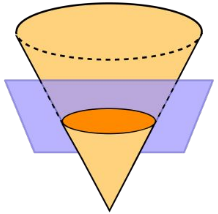
- Cone edge begins in center of the sphere and move towards top of sphere.
- Electric field does not exist inside the cone.
- Electric field quickly dies off inside dielectric as r increases.





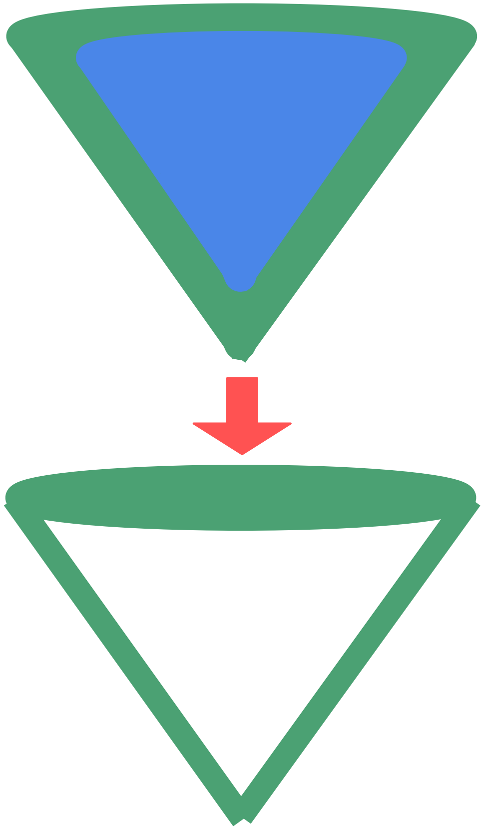
- By the time we get to the outer edge of the dielectric, the electric field looks like it emits from a sphere.
  - As r increases the original shape of the cone should matter less and less.
- Electric field is stronger on top half where we placed the cone inside the spherical water tank.





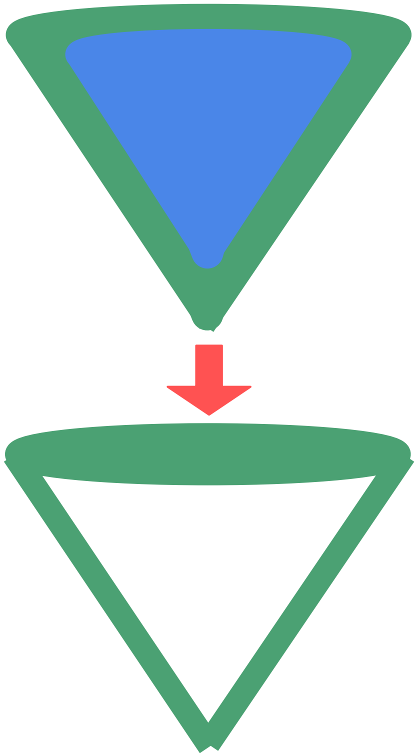
- Same as previous cross section but with symmetry about the X and Y axis. (caused by circular cut)
- Central axis of cone goes through middle of dielectric.
- Electric field has uniform strength at r value.

Determining Free Charge & Inner Bound Density Computationally



- **Blue Cone** represents original mask.
- **Green Cone** represents new mask that is one step size larger.
- We can subtract out the inner blue cone from the outer green cone to get a surface.
- This surface mask can access values on the surface of the cone.
- Since we are close to the surface, we can assume that the electric field value will primarily point in the normal direction relative to the mask.

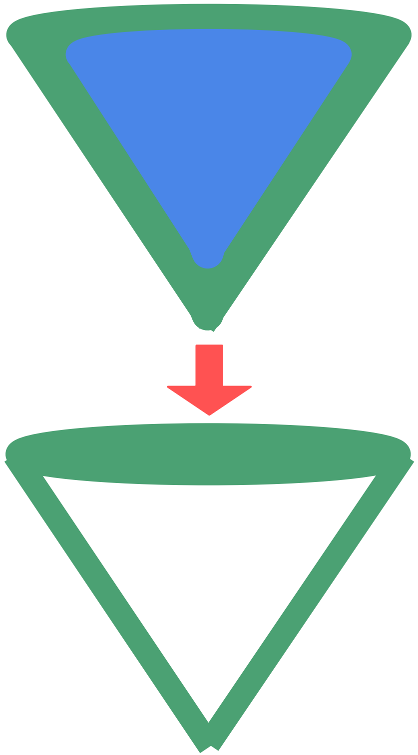
Determining Free Charge & Inner Bound Density Computationally



```
pos_cone_mask = np.zeros((x.size,y.size,z.size),dtype = bool)
pos_cone_mask[cone_index] = True
pos_cone_mask[1:-1,1:-1,1:-1] = (pos_cone_mask[2:,1:-1,1:-1] | pos_cone_mask[0:-2,1:-1,1:-1]
| pos_cone_mask[1:-1,2:,1:-1] | pos_cone_mask[1:-1,0:-2,1:-1]
| pos_cone_mask[1:-1,1:-1,2:] | pos_cone_mask[1:-1,1:-1,0:-2])
pos_cone_mask[cone_index] = False
```

- Fill cone mask with zeros.
- Extend said masks by 1 [Just ensure the cone is attached at all edges].
- Subtract out original mask.
- Yield surface of conductor or dielectric.
    - Same process as with the conducting cube.

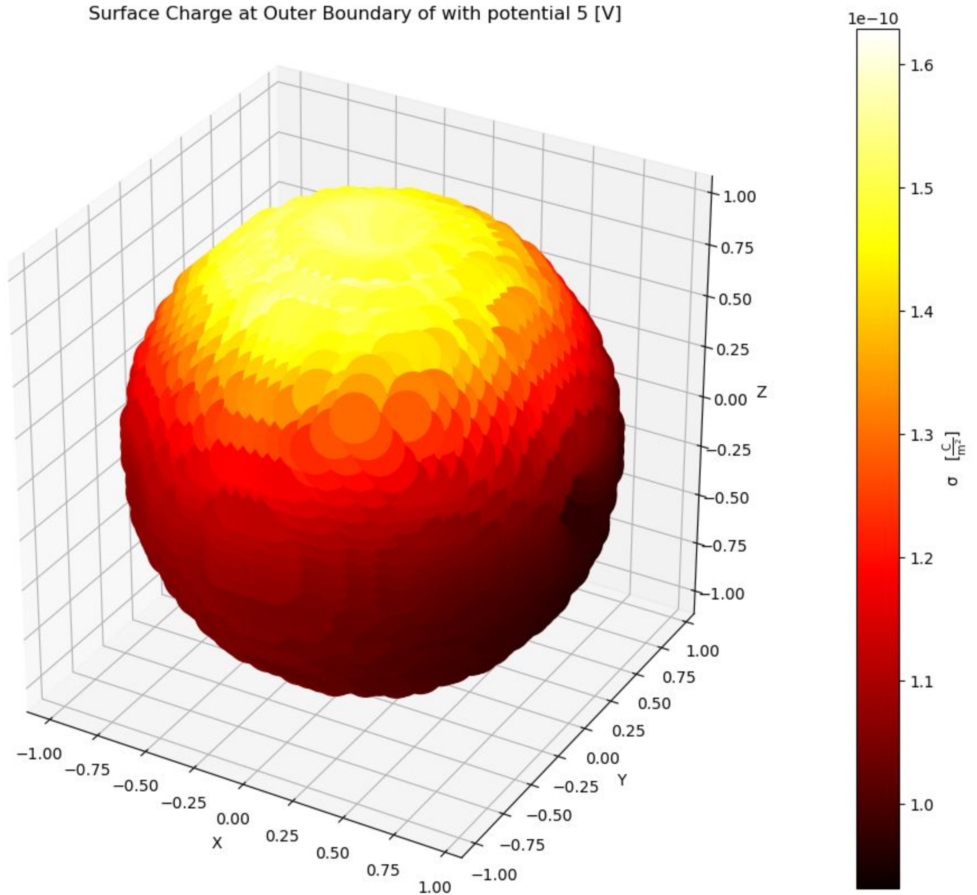
Determining Free Charge & Inner Bound Density Computationally



```
x_E_cone_bound_out = x_E_cone_out[sphere_perm]
y_E_cone_bound_out = y_E_cone_out[sphere_perm]
z_E_cone_bound_out = z_E_cone_out[sphere_perm]
dot_x_cone = x_E_cone_bound_out * X[sphere_perm]
dot_y_cone = y_E_cone_bound_out * Y[sphere_perm]
dot_z_cone = z_E_cone_bound_out * Z[sphere_perm]
sig_cone_out = (e_0*x_e/R) * (dot_x_cone + dot_y_cone + dot_z_cone)
```

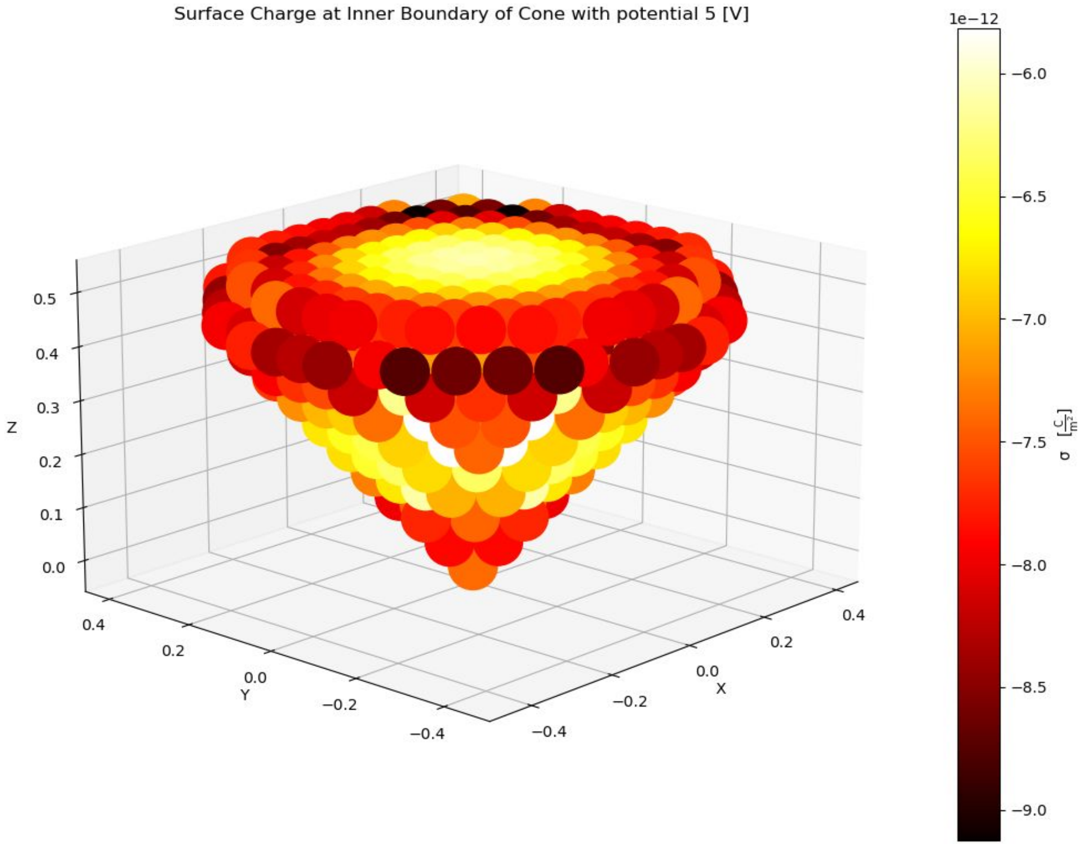
$$\sigma_{\mathrm{Bound}} = \mathrm{E}\varepsilon_o \mathrm{X_e} \cdot \left(\frac{\mathrm{x}}{\mathrm{R}}\mathrm{x}_{hat} + \frac{\mathrm{y}}{\mathrm{R}}y_{hat} + \frac{\mathrm{z}}{\mathrm{R}}z_{hat}\right)$$

- Take split up x, y, z directions for bound charge.
- Determine E value by calculating magnitude.
- Multiply by e_o and X_e.
- Reminder: Assuming electric field is pointing in the normal direction to the conductor's surface.



- The dielectric has a larger bound charge distribution at the top of the sphere which makes sense since the cone is located in the top hemisphere.
- Limitations caused by creating discrete points. Accuracy of charge distribution is inhibited.

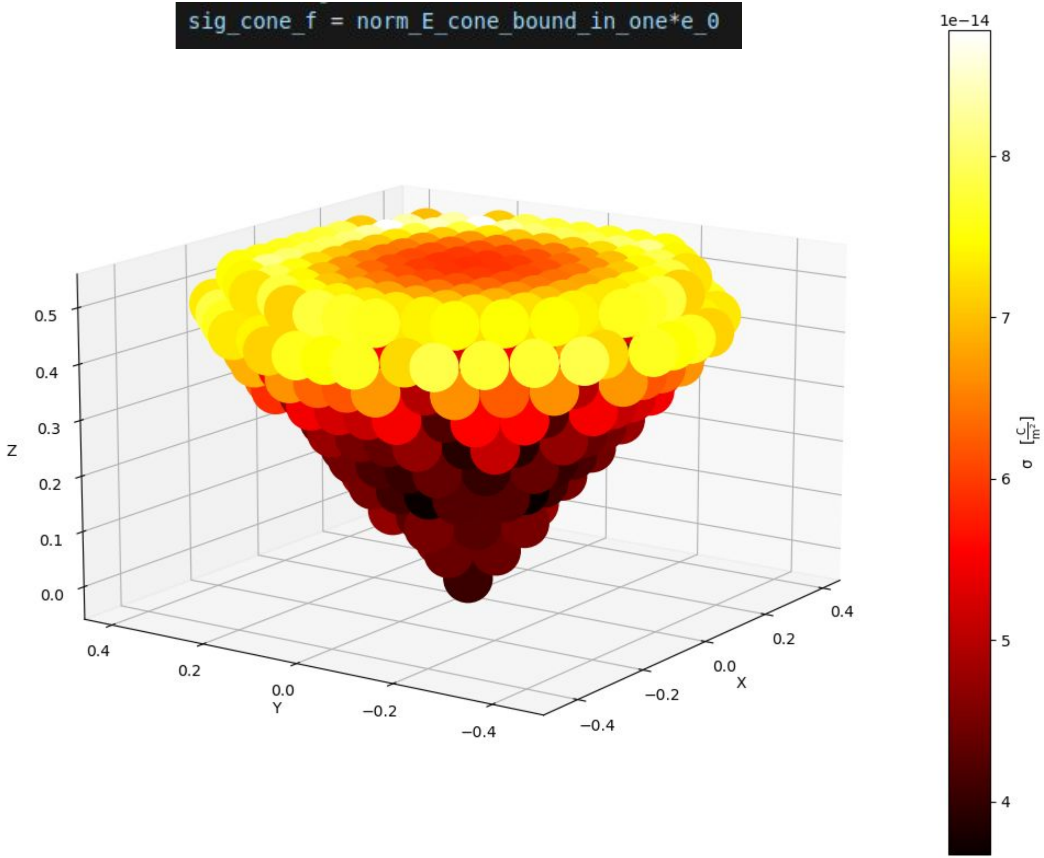
Determining Inner Bound Charge Density Computationally



- Edges of cone have the largest bound charge density.
- Center of the faces have uniform charge density.
  - Agrees with how we expect charges to gather on a non-spherical object.

Determining Free Charge Density Computationally

```
sig_cone_f = norm_E_cone_bound_in_one*e_0
```



- Consistent with Inner Bound Surface Charge 3D Model.
- Edges have most intense charge distribution.
- Surfaces have minimal charges.
- Positive charges at edges.

Thank You for Listening!

:-) Link to Video: https://youtu.be/MI1HW4wBoJc