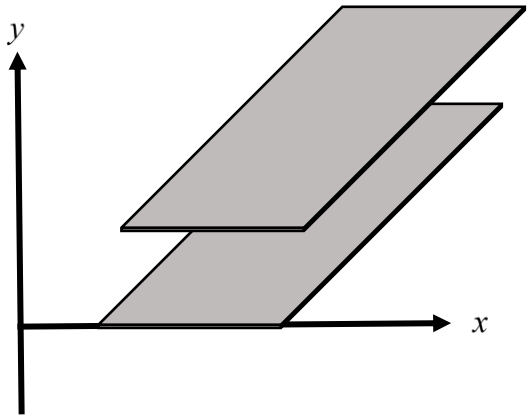
Computer Homework 15

A parallel plate capacitor shown in the figure, has an infinite length in z-direction. The distance between the plates is $d = 2\ \text{mm}$, the width of the plates is $L = 5\ \text{mm}$, and the voltage on the top plate is 100 Volt and -100 Volt on the bottom plate. The template code uses Laplacian to calculate the potential and electric field on x-y plane and plots both. Considering the edge effect, calculate numerically the capacitance per meter length in z-direction and compare the result to the ideal value without edge effect.

You need to finish (1) the Laplacian Solver, (2) finding the charge Q per meter length in z-direction on the plate, and (3) the capacitance.




```
from visual import *

N = 101
dx, dy = 1E-2/(N-1), 1E-2/(N-1)
dx2, dy2 = dx*dx, dy*dy
L, d= 5E-3, 2E-3

def solve_laplacian(u, u_cond, dx, dy, Niter=800):
    V = array(u)
    for i in range(Niter):
        V[u_cond] = u[u_cond]
        V[1:-1,1:-1] = 0          # 0 replaced by code you need to write
    return V

def get_field(V, dx, dy):
    Ex, Ey = gradient(V)
    Ex, Ey = -Ex/dx, -Ey/dy
    return Ex, Ey

u = zeros([N, N])
u[N/2-int(L/dx/2.0):N/2+int(L/dx/2.0), N/2 - int(d/dy/2.0)] = -100.0
u[N/2-int(L/dx/2.0):N/2+int(L/dx/2.0), N/2 + int(d/dy/2.0)] = 100.0
u_cond = not_equal(u, 0)

V = solve_laplacian(u, u_cond, dx, dy)

scene = display(title='dipole', height=1000, width=1000, center = (N*dx/2, N*dy/2, 0))
box(pos = (N*dx/2 , N*dy/2 - d/2 - dy    , 0), length = L, height = dy/5, width = 2*dx)
box(pos = (N*dx/2 , N*dy/2 + d/2 - dy    , 0), length = L, height = dy/5, width = 2*dx)
for i in range(N):
    for j in range(N):
        point = box(pos=(i*dx, j*dy, 0), length = dx, height= dy, width = dx, color=((V[i,j]+100)/200,(100-V[i,j])/200,0.0) )

Ex, Ey = get_field(V, dx, dy)

for i in range(1, N-1, 2):
    for j in range(1, N-1, 2):
        ar = arrow(pos = ( i*dx, j*dy, 2*dx), axis = (Ex[i,j]/6E8, Ey[i,j]/6E8, 0), shaftwidth = dx/4.0, color=color.black)


#find Q, find C_nonideal = Q/(delta V)
#Comepare C_nonideal to C_ideal
```