**Integral Equation and Method of Moment**

|  |  |
| --- | --- |
| Date | Revision |
| 3/10/2017 | Basic |
|  |  |

1. **Introduction**

Walk through the step by step process to utilize Integral Equation and Moment Method to approximate current distribution. A simple example will be illustrated along the way.

1. **Integral Equation**

Pocklington’s and Hellen’s Equations are the two famous integral equations used in Moment Method. A simple example from Textbook will be illustrated to help understanding. Let’s look at the electrostatic charge distribution along a very thin wire.

In electrostatic, a linear charge distribution creates an electric potential, which is given by:

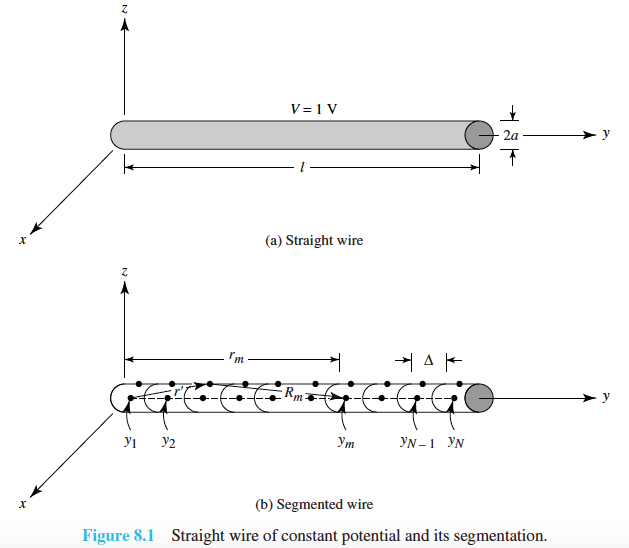


Figure1 [1]

In general, we are given and asked to find in class; however, practical situation ask for finding under was given since voltage is something we can easily measure or find. The charge distribution wouldn’t be uniform throughout the wire which makes it hard. Now, here comes integral equation and moment method into play.

As I understood, Moment method first requires you to have a set of subdomain functions which also known as the basis functions. For visualization purpose, let’s picture pulse functions which are defined:

that they are defined only on small section of domain and where is the differential quantity to encounter the small interval. Note that Dirac delta function can only be defined on a point whereas is defined on an interval whose length is .

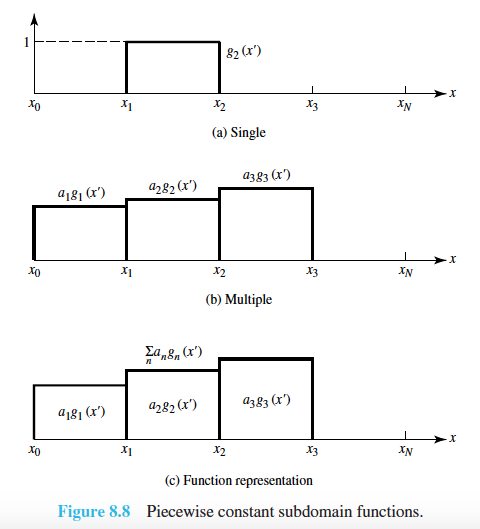


Figure2 [1]

As shown in above figure, ’s magnitude is just unity and is the coefficients that allow to represent the actual height of the desired function at that particular point. If can be found, then we can approximate the desired function and as the number of n increase, the accuracy also increases or become smoother.

Then, let’s represent with this basis function and coefficient representation:

then, equation (1) becomes

The integral equation for charge distribution is done. Considering the Figure1

for simplicity, let’s assume that the voltage across the wire is , then

An important assumption should be made in order to continue analysis, that is defining how current flows along the wire. Let’s assume that current is only on the surface of the wire. Note that current is the source of this integral equation. Now, as we make a number of observation at each point in wire, we can build a matrix equation.

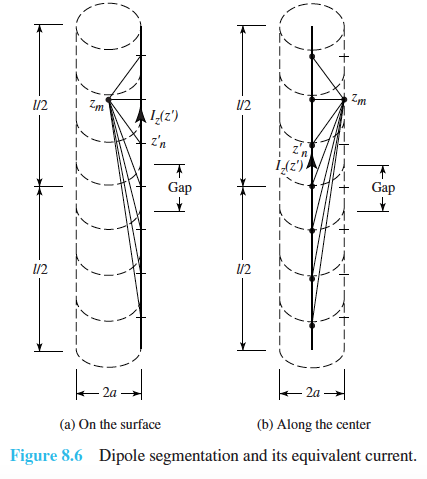


Figure3 [1]

Figure3 illustrates how observation point and source point are related which further suggest how to build a matrix out of it.

If we expand the equation (6),

Equivalent matrix equation would be,

what should be carefully noted is that the impedance matrix which has the form as:

since the voltage is just constant unity,

and the current contains the coefficients

In practice, first define and look for the equation for then current distribution can be found by:

1. Textbook Example

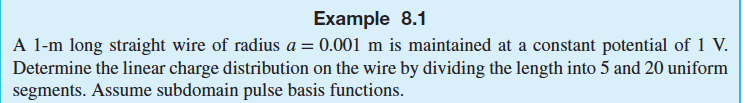
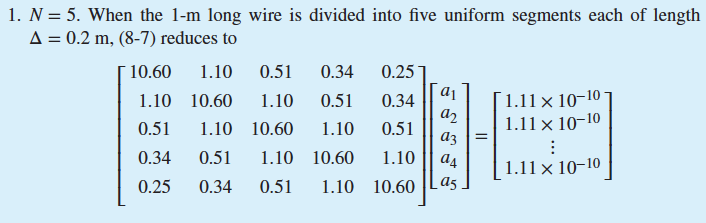
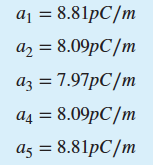


Figure4 [1]

The impedance of matrix is



and the coefficients are



Let’s see if Python code can duplicate this result.

**import** numpy **as** np  
**from** scipy.integrate **import** quad  
**from** numpy.linalg **import** inv  
**import** matplotlib.pyplot **as** plt  
  
pi = np.pi  
epsilon = 8.854e-12  
mu = 4\*pi\*1e-7  
l = 1  
V = 1  
a = 1e-3 # radius of wire  
  
**def sin**(x):  
 **return** np.sin(x)  
  
**def cos**(x):  
 **return** np.cos(x)  
  
**def ln**(x):  
 **return** np.log(x)  
  
**def sqrt**(x):  
 **return** np.sqrt(x)  
  
N = 5  
dy = l/N  
  
position = np.zeros(N)  
**for** m **in** range(N):  
 position[m] = (m)\*dy + dy/2  
print('position =',position)  
Zmn = np.zeros((N,N))  
temp3 = np.zeros(N)  
  
**for** m **in** range(N):  
 **for** n **in** range(N):  
 ym = position[m]  
  
 temp1 = **lambda** y: 1 / sqrt( (position[m] - y) \*\* 2 + a \*\* 2)  
 temp = quad(temp1, yn-(dy/2), yn+(dy/2))[0]  
 temp3[n] = temp  
 Zmn[m,n] = temp  
  
# from textbook  
tz = [[10.60,1.1,0.51,0.34,0.25],[1.1,10.6,1.1,0.51,0.34],[.51,1.1,10.6,1.1,0.51],[0.34,0.51,1.1,10.60,1.1],[0.25,0.34,0.51,1.1,10.6]]  
  
  
v = np.zeros(N)  
**for** m **in** range(N):  
 v[m] = 4\*pi\*epsilon  
  
#ty = inv(tz)  
#ti = np.dot(ty,v)  
  
  
# calculate current distribution  
ymn = inv(Zmn)  
i = np.dot(v,ymn)  
  
pos = np.zeros(N)  
**for** m **in** range(N):  
 pos[m] = (m+1)\*dy  
print('pos = ',pos)  
  
plt.figure()  
plt.plot(position,np.abs(i), drawstyle = 'steps-mid')  
plt.title('Current Distribution along the wire')  
plt.xlabel('distance [m]')  
plt.ylabel('Magnitude of Current')  
plt.show()  
  
print(Zmn)

The output log of python code as the follow:

|  |
| --- |
| /Users/JangSambong/anaconda/bin/python3.5 /Users/JangSambong/Desktop/TAMU/ECEN451/hw7/MoM.py  position = [ 0.1 0.3 0.5 0.7 0.9]  [[ 10.59668473 1.09859007 0.51082385 0.33647175 0.25131423]  [ 1.09859007 10.59668473 1.09859007 0.51082385 0.33647175]  [ 0.51082385 1.09859007 10.59668473 1.09859007 0.51082385]  [ 0.33647175 0.51082385 1.09859007 10.59668473 1.09859007]  [ 0.25131423 0.33647175 0.51082385 1.09859007 10.59668473]]  Process finished with exit code 0 |

Which shows exactly the same result as provided in the textbook. The plot was also made:

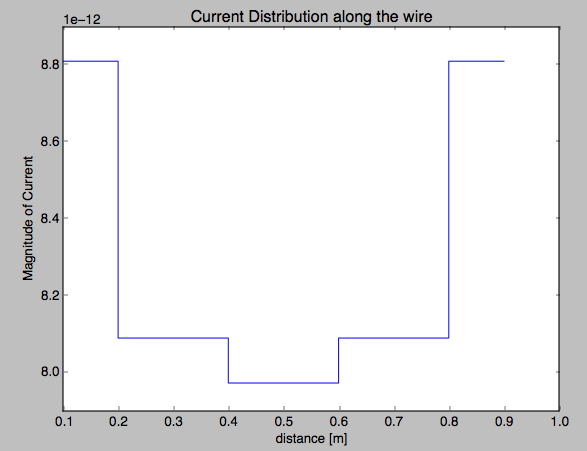


Figure5 [1]

Built-in function in Python for plotting couldn’t yield the perfect plot. Thus, more sophisticated user-defined function can allow the same plot as introduced in the textbook. I will come back and try to fix this issue when I feel like I have to do.