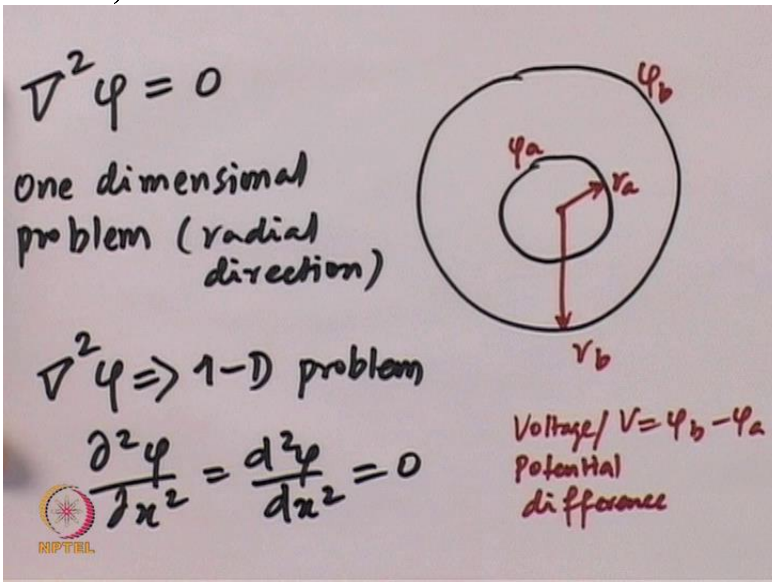
Computational Electromagnetics and Applications
Professor Krish Sankaran
Indian Institute of Technology Bombay
Exercise No 13
Finite Element Method -II

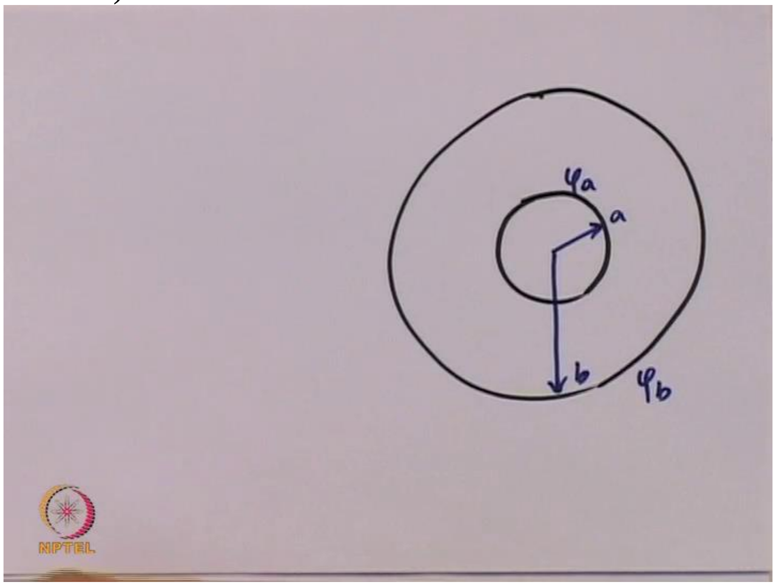
(Refer Slide Time: 00: 20)



$\nabla^2 \varphi = 0$
One dimensional problem (radial direction)
$\nabla^2 \varphi \Rightarrow$ 1-D problem
$\frac{\partial^2 \varphi}{\partial x^2} = \frac{d^2 \varphi}{dx^2} = 0$
φ_b
φ_a
r_a
r_b
Voltage/ Potential difference $V = \varphi_b - \varphi_a$

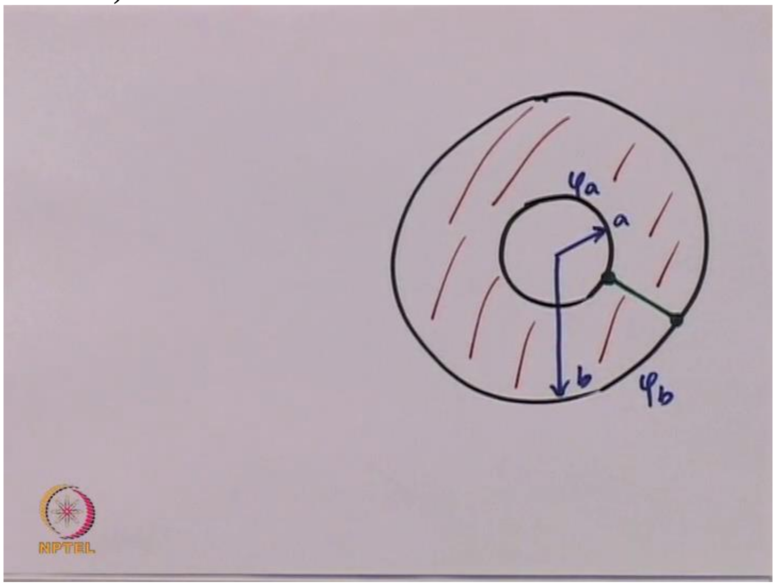
We are now going to look into the problem which we started in the earlier module the simple coaxial capacitor and we are going to use Finite element method to simulate this problem. So let us start the geometry and then I will describe you what will be the process in which we are going to do this problem using finite element method.

(Refer Slide Time: 00: 40)



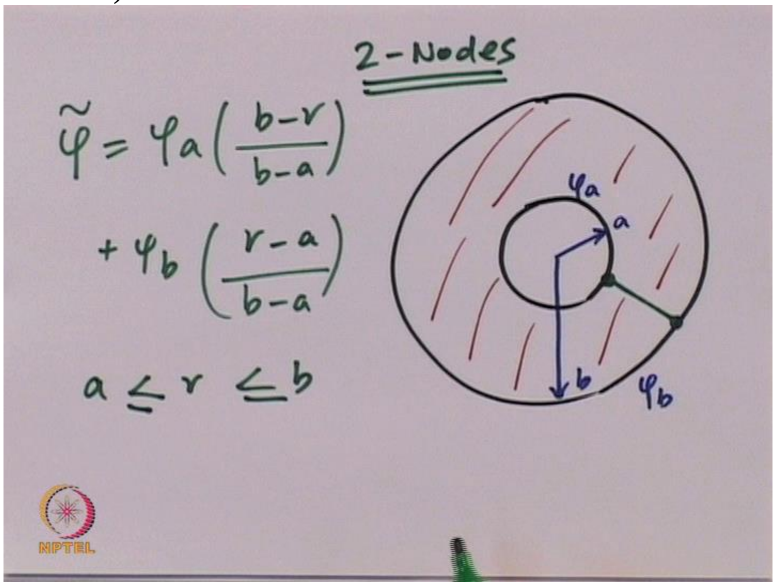
So the problem geometry is we have an outer conductor and we have an inner conductor the dimensions of the outer and the inner conductor are also described in the previous module so we are going to do it again. So the dimension of the inner conductor is going to be a radius and the outer conductor is going to be b. And the potential on the inner conductor is going to be Phi a and the potential on the outer conductor is going to be Phi b. For the starting simple Finite element approximation we will start to take the radial direction and we consider only two points on this radial direction.

(Refer Slide Time: 01: 34)



So the point 1 is going to be on the inner conductor and the point 2 is going to be on the outer conductor. When we do that the very simple approximation using finite element method for calculating the potential on any point in between this area so what we are interested is we are interested in computing the potential in the area between the two conductors. So the area marked in red.

(Refer Slide Time: 02: 05)



2-Nodes

$$\tilde{\varphi} = \varphi_a\left(\frac{b-r}{b-a}\right) + \varphi_b\left(\frac{r-a}{b-a}\right)$$

$$a \leq r \leq b$$

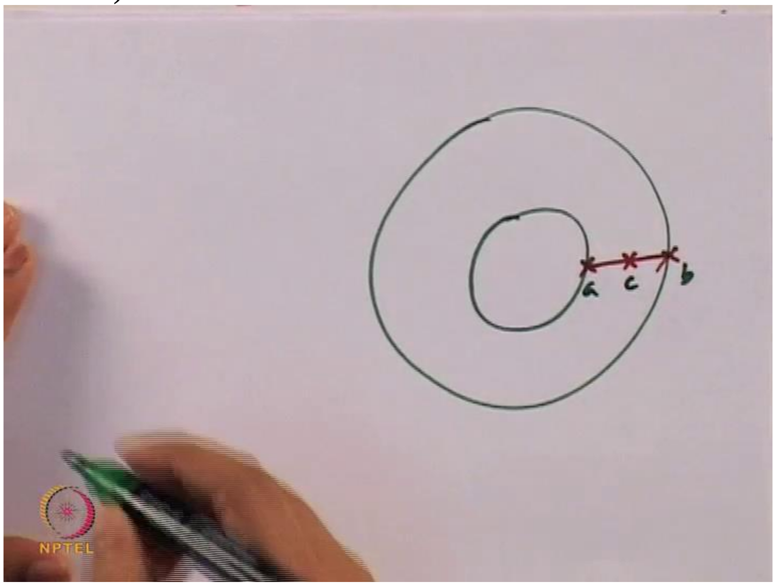
And now we are going to use the simple finite element approximation to compute that. So our first simple approximation will be; so Phi tilde is equal to phi a multiplied by some coefficients plus Phi b multiplied by some coefficients. And the question is what are those coefficients and how are they calculated. So we know the distance between the two points so this is going to be a and this is going to be b and distance between the two points is going to be b by a.

So the weighting functions or the coefficients what we have will have b minus a as the denominator. And the numerator is going to depend on for Phi a it is going to depend on b minus r so it is going to be any value from, so r is going to go from any value from a to b and so b minus r will be the waiting function i have to give for the first one.

So b minus r. and the numerator for the second value that is Phi b is going to be r minus a. We know that a is less than r is less than b or equal to. So a is less than or equal to r is less than or equal to b. So this is the way we are going to compute the value for two nodes, so this is for 2 nodes.

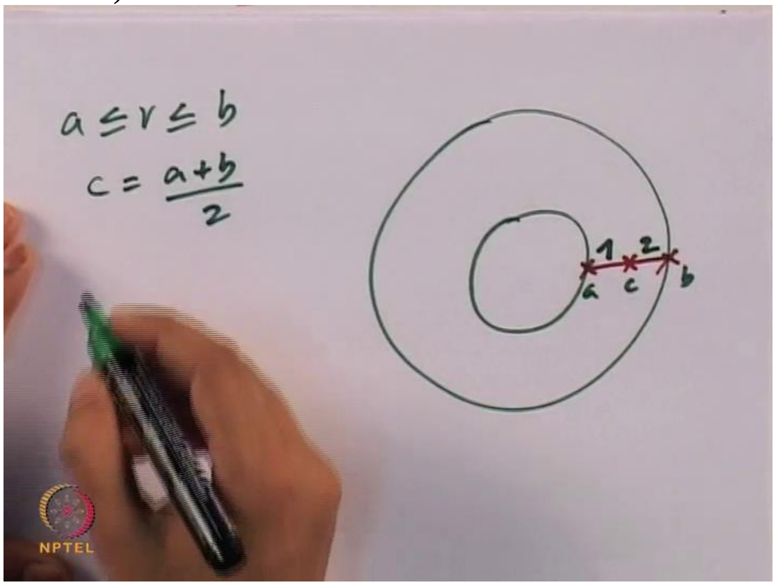
So this is a very simple straight forward application of Finite element method to compute the value of the potential on any points in between r equal to a to r equal to b. So this is the approximation that we are going to take. So if we extend this to let us say three nodes. So I am going to describe that step by step.

(Refer Slide Time: 04: 29)



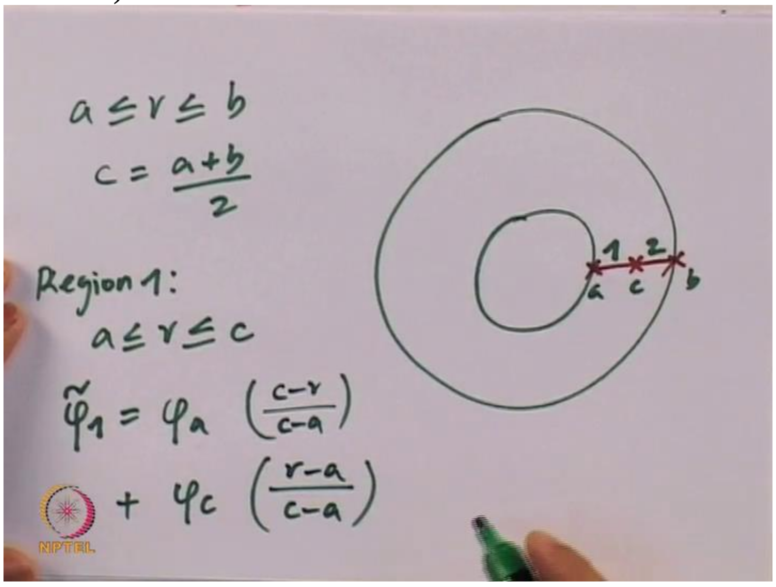
So the three nodes will be again we will draw the geometry the inner conductor and now the three nodes are going to be positioned such that the first node is going to be on the conductor a inner conductor, the second node is going to be on the outer conductor and the third node is going to be exactly in between a and b. So if this is a, this is b and this is c.

(Refer Slide Time: 05: 02)



What we have a is less than or equal to r is less than or equal to b. But there is also going to be a point that is called c that is in between a and b, so c will be equal to a plus b by 2. So we do that we can split this problem into two regions. So this is region number 1, this is region number 2, the first one. So accordingly we will have two sets of values that we can compute for Phi.

(Refer Slide Time: 05: 40)



$a \leq r \leq b$

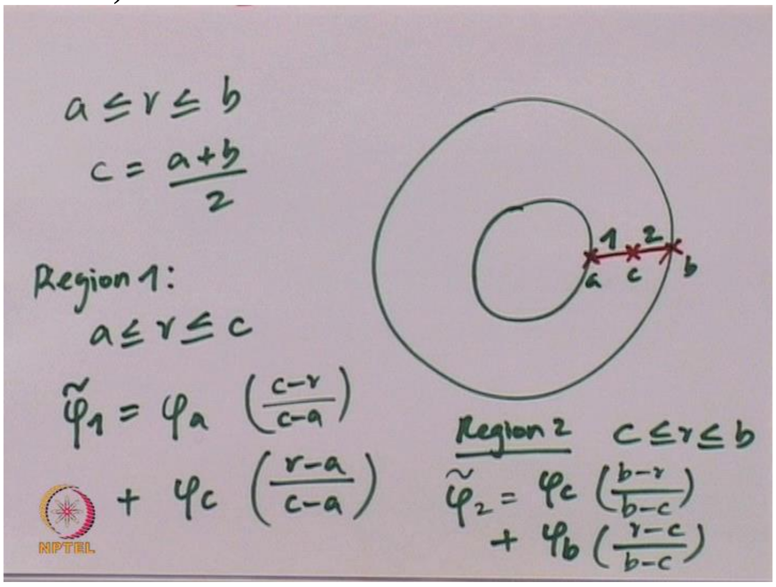
$c = \frac{a+b}{2}$

Region 1:

$a \leq r \leq c$

$\tilde{\varphi}_1 = \varphi_a \left(\frac{c-r}{c-a}\right) + \varphi_c \left(\frac{r-a}{c-a}\right)$

So let me write down for region 1, or a value will be less than or equal to r or is less than or equal to c. So this is this region. So it starts from the inner conductor and goes until the middle between the inner conductor and the outer conductor. c is given by a plus b by 2. For this region we can compute the Phi approximation 1 that is equal to like before we had the first value and the second value multiplied by certain coefficients write.

So similarly we will have the first value whereas for instead of Phi b we will have the value in the middle because that is our point of consideration. So we will have Phi a multiplied by certain coefficients plus Phi c multiplied by certain coefficients and I said the denominator will always be the distance value. So that is going to be c minus a c minus a and the numerator for the first one will be c minus r, and the second value it will be r minus a.
(Refer Slide Time: 06: 53)

$a \leq r \leq b$

$c = \frac{a+b}{2}$

Region 1:

$a \leq r \leq c$

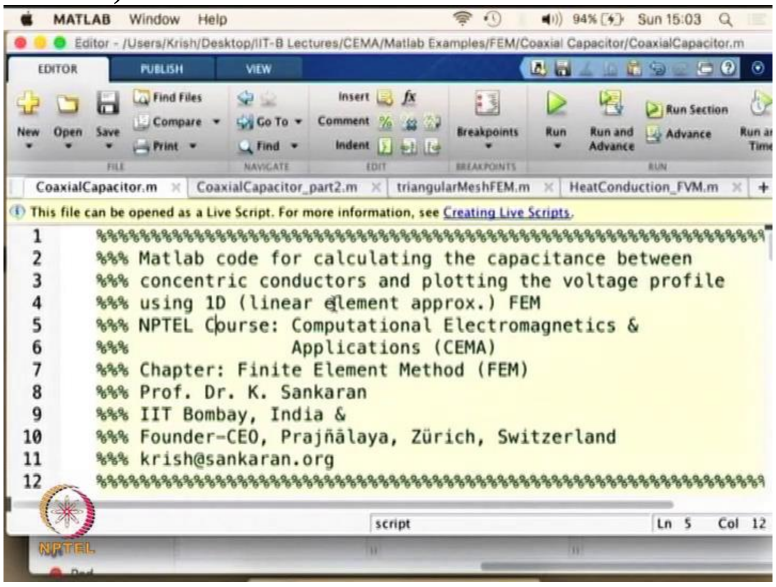
$\tilde{\varphi}_1 = \varphi_a \left(\frac{c-r}{c-a}\right) + \varphi_c \left(\frac{r-a}{c-a}\right)$

Region 2 $c \leq r \leq b$

$\tilde{\varphi}_2 = \varphi_c \left(\frac{b-r}{b-c}\right) + \varphi_b \left(\frac{r-c}{b-c}\right)$

So similarly we can also write it for region 2 which is nothing but c less than or equal to r is less than or equal to b. There value will be Phi tilde 2 that is equal to we have two values Phi c multiplied by certain coefficients plus Phi b multiplied by certain coefficients. And the denominators will be the distance so we will have b minus c b minus c. And on the top we will have b minus r for the first case then for the second one it will be r minus c. What you see is the kind of waiting and this waiting is going to depend on the geometry of the space that we are considering. So it is geometrical waiting, and this is how we are going to do the simple approximation for three points.
And similarly what you can see is we can keep on increasing the number of points and we will be having more and more accuracy. So let us go into the code with this background and see how this simulation how this problem can be simulated using the Matlab environment.

(Refer Slide Time: 08: 09)

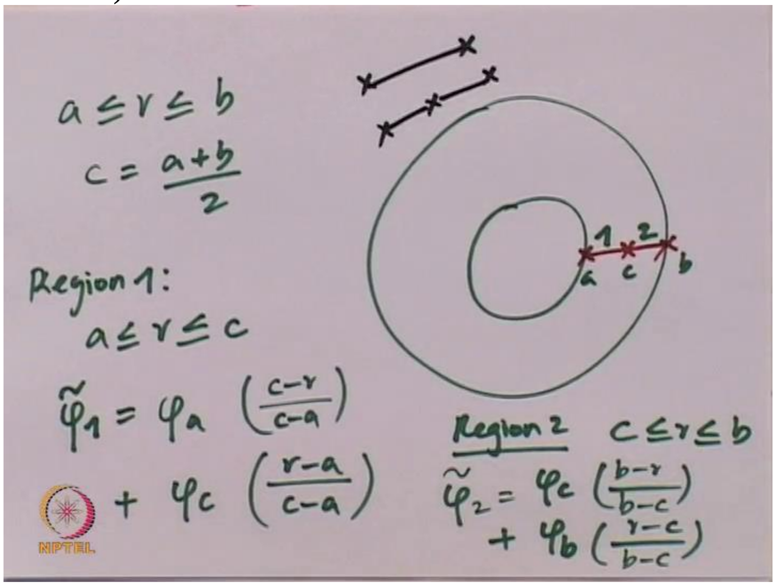


This file can be opened as a Live Script. For more information, see Creating Live Scripts.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Matlab code for calculating the capacitance between
%%% concentric conductors and plotting the voltage profile
%%% using 1D (linear element approx.) FEM
%%% NPTEL Course: Computational Electromagnetics &
%%%                 Applications (CEMA)
%%% Chapter: Finite Element Method (FEM)
%%% Prof. Dr. K. Sankaran
%%% IIT Bombay, India &
%%% Founder-CEO, Prajñālaya, Zürich, Switzerland
%%% krish@sankaran.org
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

So let us start with the code, so the code we are going to use is the coaxial capacitor it is going to calculate the capacitance between the concentric conductors and plotting the voltage profile using 1D linear approximation. We call it linear approximation because the points or the nodes are joint in this case always by lines.

(Refer Slide Time: 08: 37)



$$a \leq r \leq b$$

$$c = \frac{a+b}{2}$$

Region 1:

$$a \leq r \leq c$$

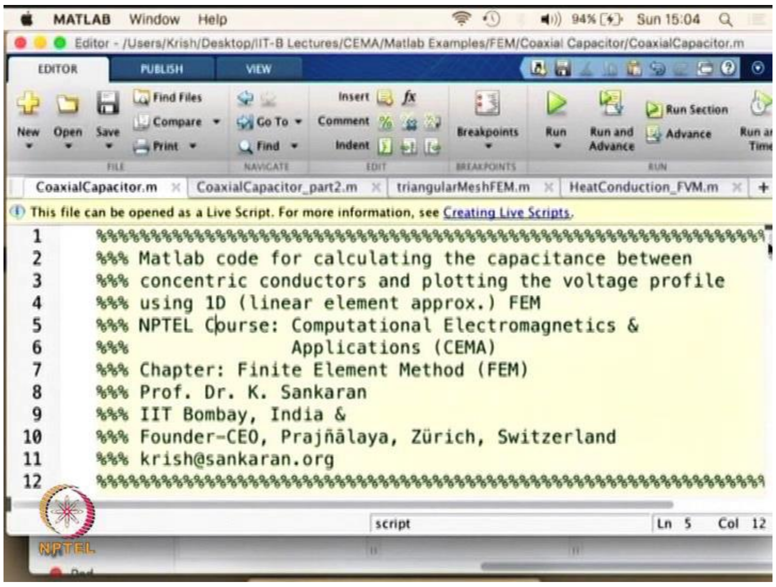
$$\tilde{\varphi}_1 = \varphi_a \left(\frac{c-r}{c-a}\right) + \varphi_c \left(\frac{r-a}{c-a}\right)$$

Region 2 $\quad c \leq r \leq b$

$$\tilde{\varphi}_2 = \varphi_c \left(\frac{b-r}{b-c}\right) + \varphi_b \left(\frac{r-c}{b-c}\right)$$

If it is going to be two points we are just going to join them by 1 line. As you can see here if there are only going to be 2 points I am going to join them by 1 line. If they are going to be 3 points I am going to join it by two lines so on and so forth. But always it is going to be lines that is why we are calling it as linear approximation.

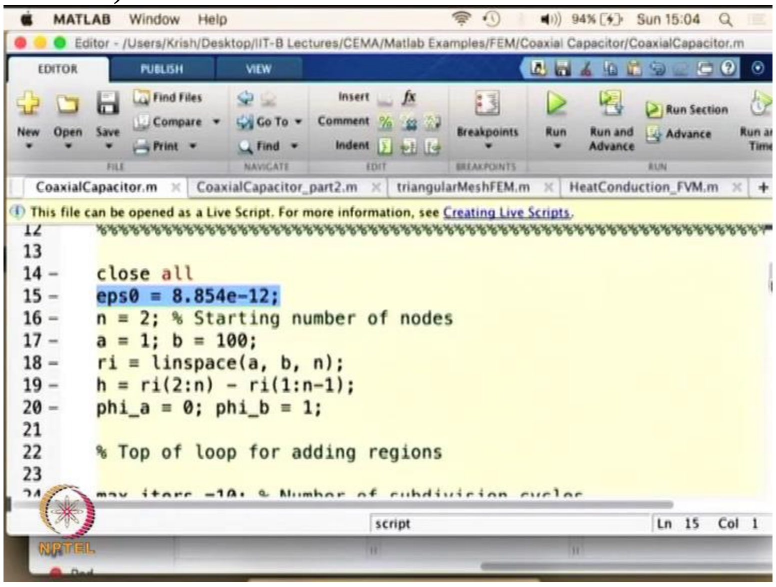
(Refer Slide Time: 09: 02)



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Matlab code for calculating the capacitance between
%%% concentric conductors and plotting the voltage profile
%%% using 1D (linear element approx.) FEM
%%% NPTEL Course: Computational Electromagnetics &
%%%                  Applications (CEMA)
%%% Chapter: Finite Element Method (FEM)
%%% Prof. Dr. K. Sankaran
%%% IIT Bombay, India &
%%% Founder-CEO, Prajñālaya, Zürich, Switzerland
%%% krish@sankaran.org
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

And we are going to have the basis functions as a linear elements so it is going to be of the order of the nature ax plus b. But that is not important for now let us go into the code and see how the proces is setup.

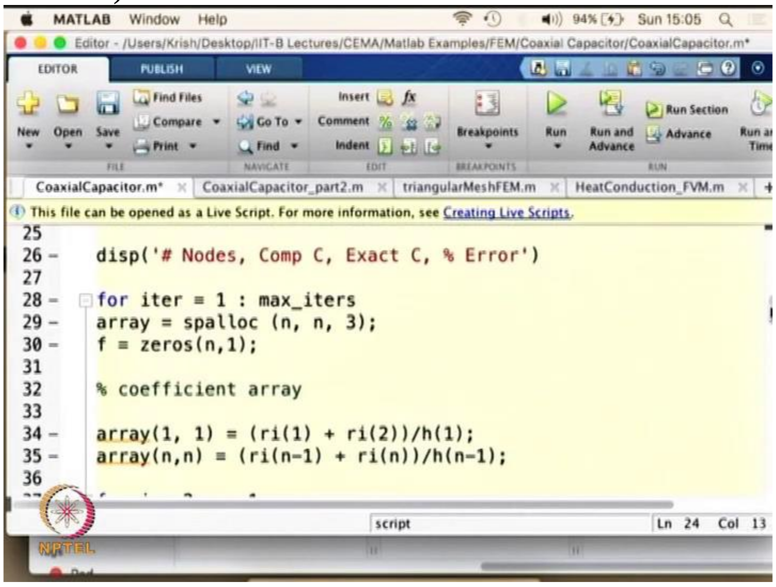
(Refer Slide Time: 09: 15)



```
close all
eps0 = 8.854e-12;
n = 2; % Starting number of nodes
a = 1; b = 100;
ri = linspace(a, b, n);
h = ri(2:n) - ri(1:n-1);
phi_a = 0; phi_b = 1;

% Top of loop for adding regions
```

So we will do this problem for the coaxial capacitor, so we have to set certain parameters so epsilon 0 is set epsilon 0 is going to be 8.854 10 power minus 12. The number of nodes we are starting with 2. The distances are such that a equal to 1 and b equal to 100. So b is 100 times that of a. It is a very big outer circle and very small inner conductor. And the potential Phi a is set to 0 and Phi b is set to 1.

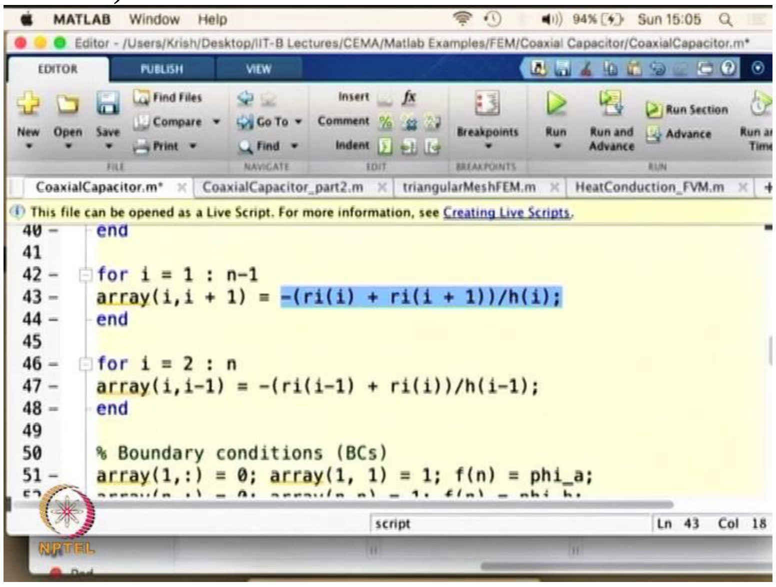
(Refer Slide Time: 09: 50)



```
disp('# Nodes, Comp C, Exact C, % Error')

for iter = 1 : max_iters
array = spalloc (n, n, 3);
f = zeros(n,1);

% coefficient array

array(1, 1) = (ri(1) + ri(2))/h(1);
array(n,n) = (ri(n-1) + ri(n))/h(n-1);
```

And initially we will start with the number of iteration as 1. And iteration is going to have dual function in this code I will explain you that in a bit. For now it is important to know that we are iterating the problem once.

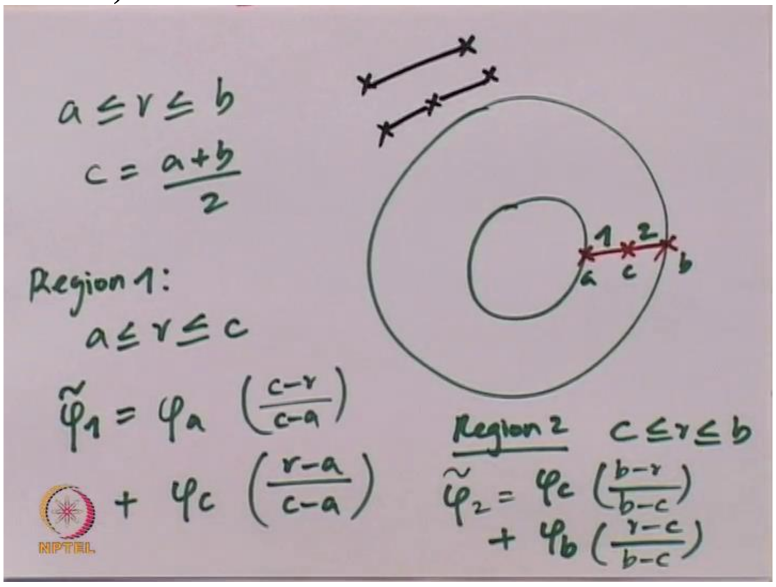
(Refer Slide Time: 10: 12)



```
end

for i = 1 : n-1
array(i,i + 1) = -(ri(i) + ri(i + 1))/h(i);
end

for i = 2 : n
array(i,i-1) = -(ri(i-1) + ri(i))/h(i-1);
end

% Boundary conditions (BCs)
array(1,:) = 0; array(1, 1) = 1; f(n) = phi_a;
```

And we are computing the value of capacitance using finite element method using the weighting function that I have described. So the weighting functions are here. So these are the weighting functions which are geometric in nature.

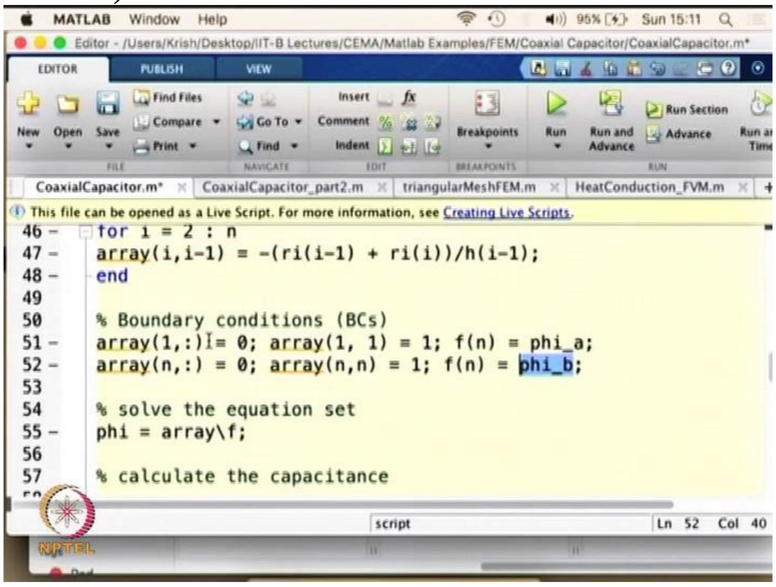
(Refer Slide Time: 10: 30)



$a \le r \le b$
$c = \frac{a+b}{2}$
Region 1:
$a \le r \le c$
$\tilde{\varphi}_1 = \varphi_a \left(\frac{c-r}{c-a}\right) + \varphi_c \left(\frac{r-a}{c-a}\right)$
1
2
a
c
b
Region 2 $c \le r \le b$
$\tilde{\varphi}_2 = \varphi_c \left(\frac{b-r}{b-c}\right) + \varphi_b \left(\frac{r-c}{b-c}\right)$

Let me explain you that here once again in the code. So the weighting functions are the functions that are coming here and those are the values that we have described in the array in the code.

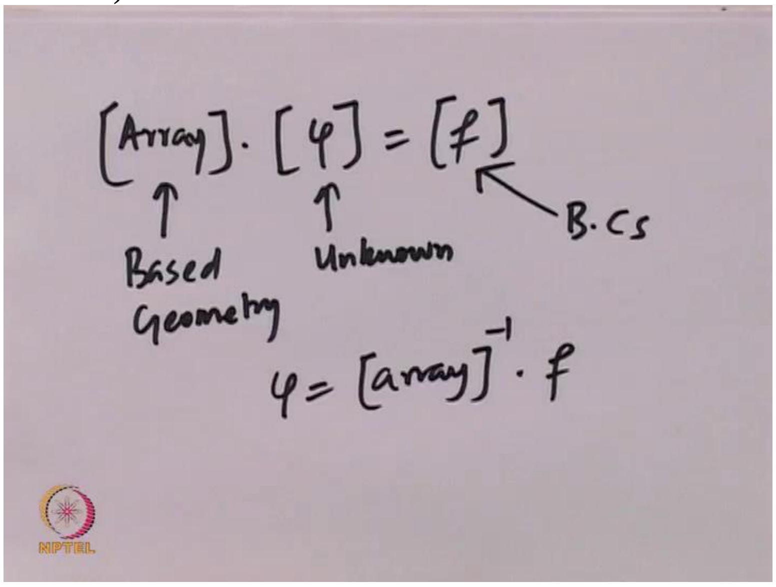
(Refer Slide Time: 10: 40)



```
for i = 2 : n
array(i,i-1) = -(ri(i-1) + ri(i))/h(i-1);
end

% Boundary conditions (BCs)
array(1,:) = 0; array(1, 1) = 1; f(n) = phi_a;
array(n,:) = 0; array(n,n) = 1; f(n) = phi_b;

% solve the equation set
phi = array\f;

% calculate the capacitance
```

And you can see that here, so these are the weighting functions and the boundary conditions are set as you can see the boundary conditions are basically the potentials that we are assigning and once we have that we have array of values. So we have an equation that is basically looking something like this. So we are computing the value of the capacitance using this program while doing that we are setting certain conditions for the boundary conditions.

(Refer Slide Time: 11: 20)

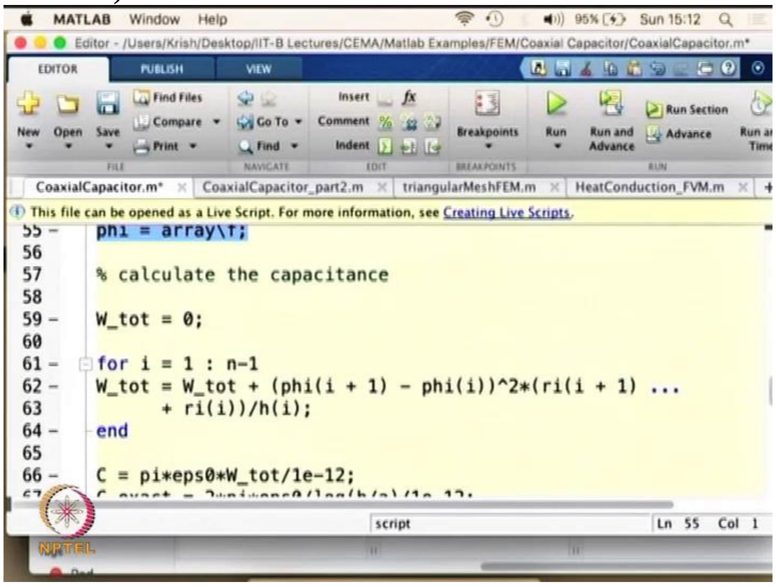


$$[\text{Array}] \cdot [\varphi] = [f]$$

$$\varphi = [\text{array}]^{-1} \cdot f$$

And once we have that what we have is the simple approximation that the array will be the geometrical dependent quantity and Phi are the potential that we are going to compute, and this is going to be equal to F. And as I said this is based on the geometry. And this is the unknown we are computing and these are the boundary conditions we are setting. Once we this one and this one we can compute this one by taking array inverse and multiplying it with F and that is what you see in this code here.

(Refer Slide Time: 12: 16)



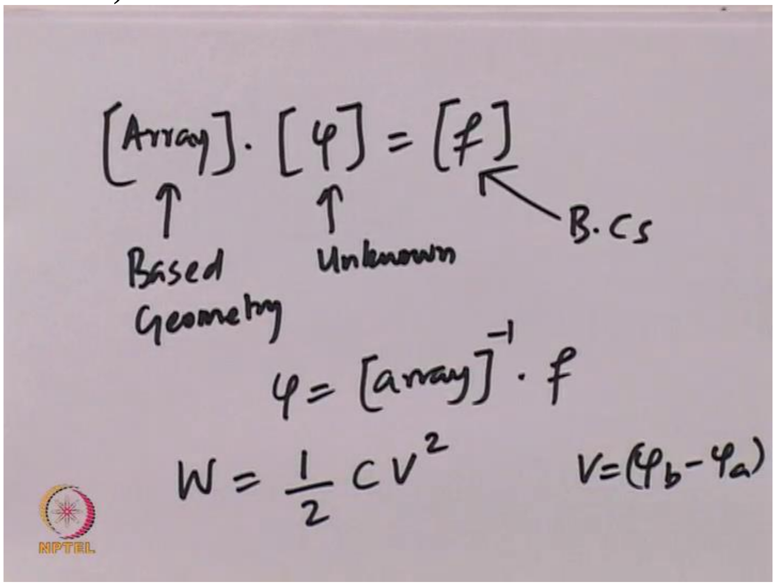
MATLAB Window Help
Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FEM/Coaxial Capacitor/CoaxialCapacitor.m*
EDITOR PUBLISH VIEW
CoaxialCapacitor.m* CoaxialCapacitor_part2.m triangularMeshFEM.m HeatConduction_FVM.m

This file can be opened as a Live Script. For more information, see Creating Live Scripts.

```
phi = array\f;

% calculate the capacitance

W_tot = 0;

for i = 1 : n-1
W_tot = W_tot + (phi(i + 1) - phi(i))^2*(ri(i + 1) ...
      + ri(i))/h(i);
end

C = pi*eps0*W_tot/1e-12;
C_exact = 2*pi*eps0/log(b/a)/1e-12;
```

That is what you see here in this line. And once we have that we are going to compute the value of capacitance in a step by step manner.

(Refer Slide Time: 12: 28)



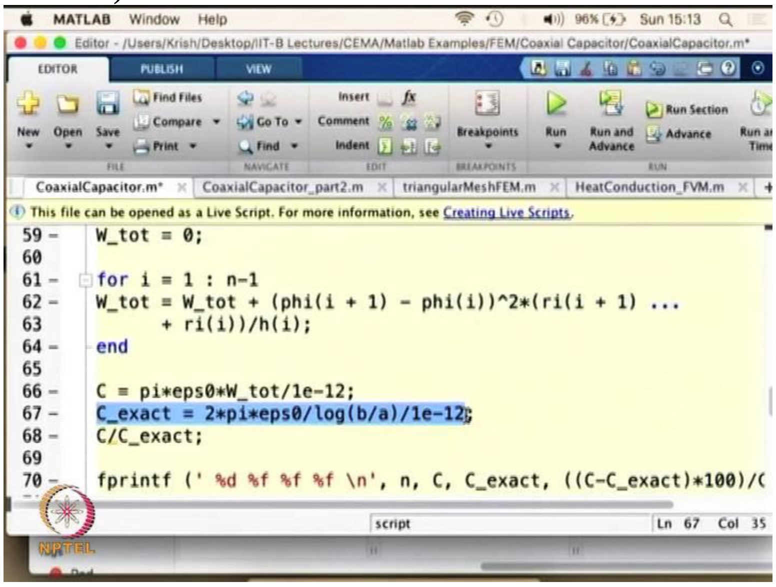
$$\underset{\substack{\uparrow\\ \text{Based}\\ \text{Geometry}}}{[\text{Array}]} \cdot \underset{\substack{\uparrow\\ \text{Unknown}}}{[\varphi]} = [f] \leftarrow \text{B.Cs}$$

$$\varphi = [\text{array}]^{-1} \cdot f$$

$$W = \frac{1}{2} C V^2 \qquad V = (\varphi_b - \varphi_a)$$

So for that we are going to use the equation w is equal to 1 by 2 cv square v equal to Phi b minus Phi a) this is the potential difference. And once we know this for the element we can substitute that here and we can compute the value of the capacitance using the total energy stored.

(Refer Slide Time: 12: 58)



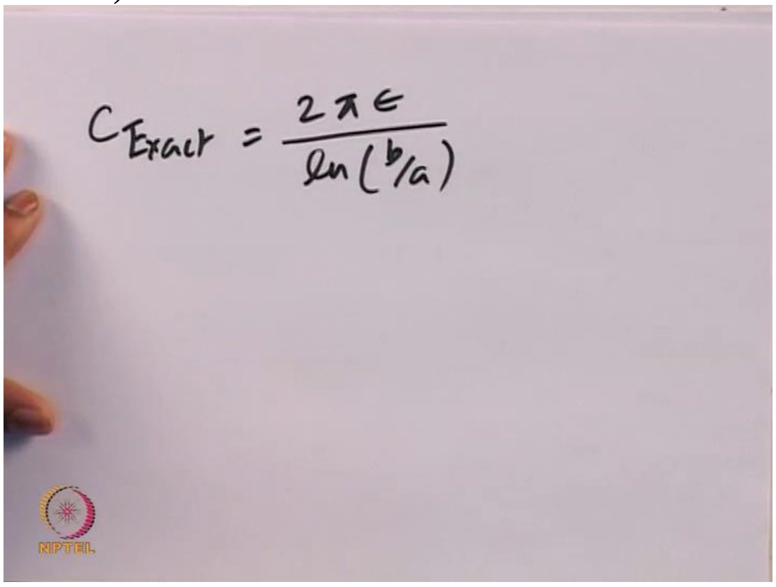
MATLAB Window Help
Editor - /Users/Krish/Desktop/IIT-B Lectures/CEMA/Matlab Examples/FEM/Coaxial Capacitor/CoaxialCapacitor.m*
EDITOR PUBLISH VIEW
CoaxialCapacitor.m* CoaxialCapacitor_part2.m triangularMeshFEM.m HeatConduction_FVM.m
This file can be opened as a Live Script. For more information, see Creating Live Scripts.
script Ln 67 Col 35

```
W_tot = 0;

for i = 1 : n-1
W_tot = W_tot + (phi(i + 1) - phi(i))^2*(ri(i + 1) ...
      + ri(i))/h(i);
end

C = pi*eps0*W_tot/1e-12;
C_exact = 2*pi*eps0/log(b/a)/1e-12;
C/C_exact;

fprintf (' %d %f %f %f \n', n, C, C_exact, ((C-C_exact)*100)/C
```

So the total energy stored is computed in the code in this manner. So initially we set the value to be 0 and then we go through each of the nodes. In our case there are going to be only two nodes. So we will have only one element. So 1, 2 n minus 1 will be 2 minus 1 that will be 1 to 1. So it will be only 1 element. So that will be this element. And once we have that we compute the value of total energy of that particular element. And once we know the total energy we can compute the value of the capacitance from the total energy based on the equation here.

And once we have that we can also know the exact value for the capacitance based on the theoritcal value which we discussed in the earlier module. And once we used that value for this particular geometry we are able to calculate the value for c exact.

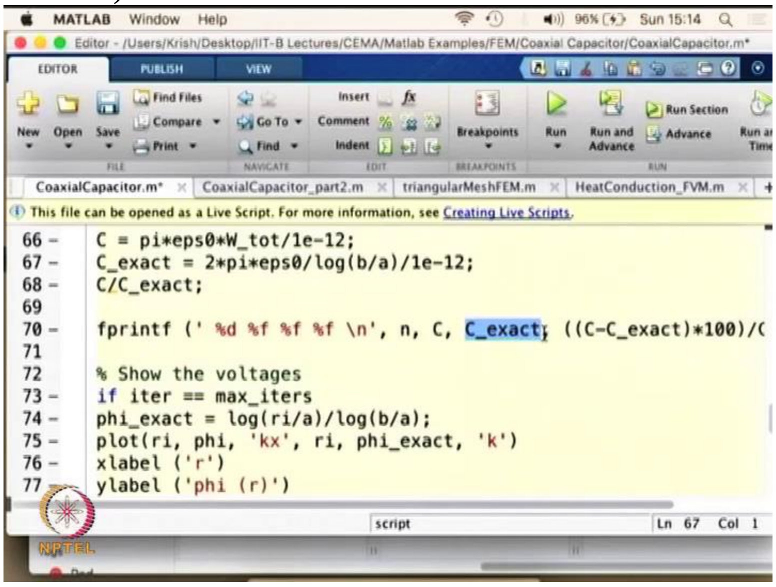
(Refer Slide Time: 14: 02)



$$C_{Exact} = \frac{2\pi\epsilon}{\ln(b/a)}$$

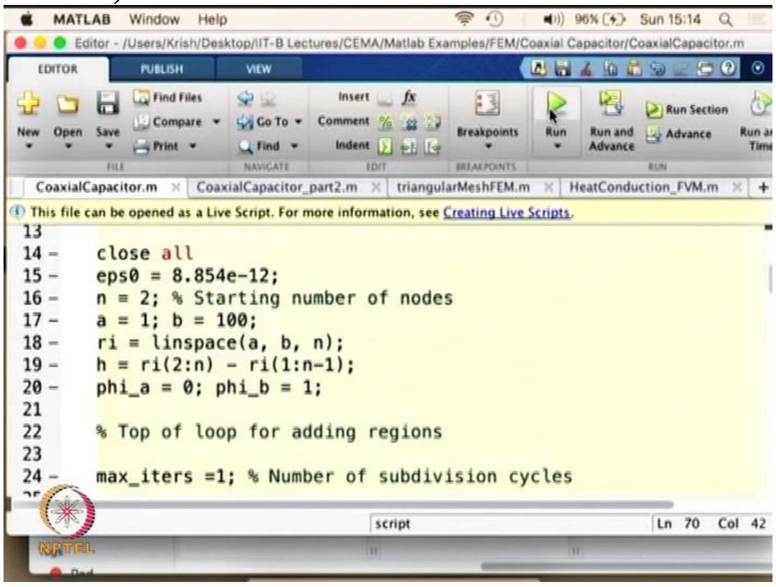
C Exact will be given by 2 pi Epsilon divided by natural logarithm of (b by a) since the unit will be in picofared we have put 1 into 10 power minus 12 here. So this is not important but this is the way we are computing the value of the exact value.

(Refer Slide Time: 14: 29)


```
C = pi*eps0*W_tot/1e-12;
C_exact = 2*pi*eps0/log(b/a)/1e-12;
C/C_exact;

fprintf (' %d %f %f %f \n', n, C, C_exact, ((C-C_exact)*100)/(

% Show the voltages
if iter == max_iters
phi_exact = log(ri/a)/log(b/a);
plot(ri, phi, 'kx', ri, phi_exact, 'k')
xlabel ('r')
ylabel ('phi (r)')
```

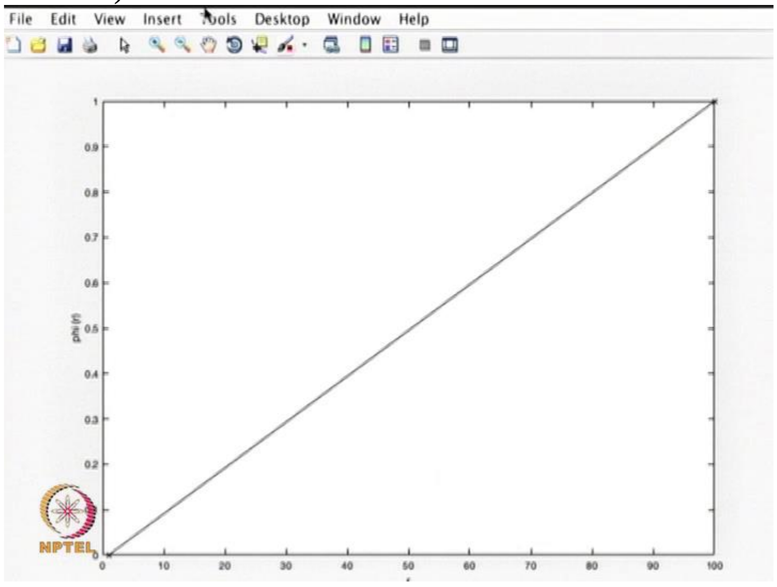
And we can compute the error also here as C by C exact. What we are finally printing is the number of nodes the computed FEM c value the exact c value and the error between the exact value and the computed value.

(Refer Slide Time: 14: 54)


```
close all
eps0 = 8.854e-12;
n = 2; % Starting number of nodes
a = 1; b = 100;
ri = linspace(a, b, n);
h = ri(2:n) - ri(1:n-1);
phi_a = 0; phi_b = 1;

% Top of loop for adding regions

max_iters =1; % Number of subdivision cycles
```

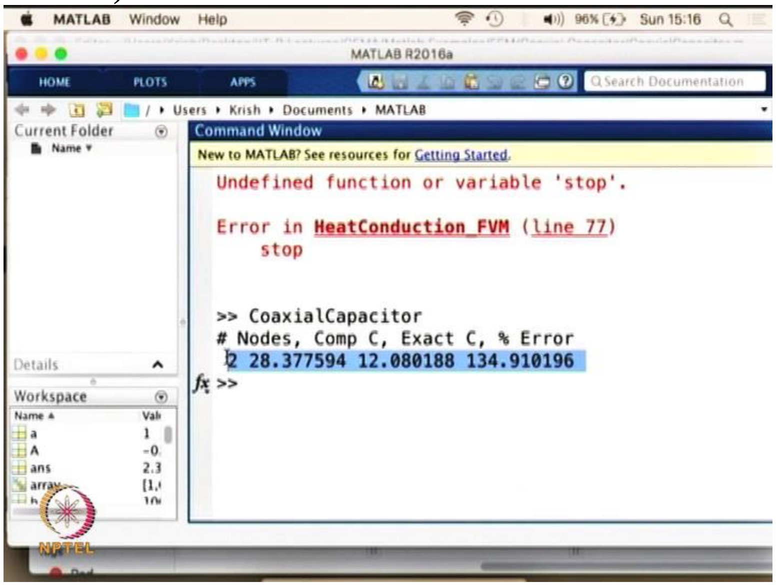
So let us run this program for n equal to 2 and then see the outcome of the problem.

(Refer Slide Time: 15: 03)



So what we are seeing here is we have got only two points and those two points are exactly at r equal to 1 or the radial axis and this is the potential along this axis. And r goes from 1 to 100 so you see that it starts not exactly at 0, slightly away from 0 that is the point number 1. r equal to 1. So it goes from this point to this point and the exact value is basically the value that is between these points. So since there is only one element it has to join the boundaries values. So the potential here and the potential here and you have a straight line. And what you see is nothing but a kind of a parallel plate approximation. If we have only one element it shows as a linear one. And we know that this is not true because the value will follow a kind of a curve like this, and this is a very poor approximation, and this is also reflected in the error that we are computing.

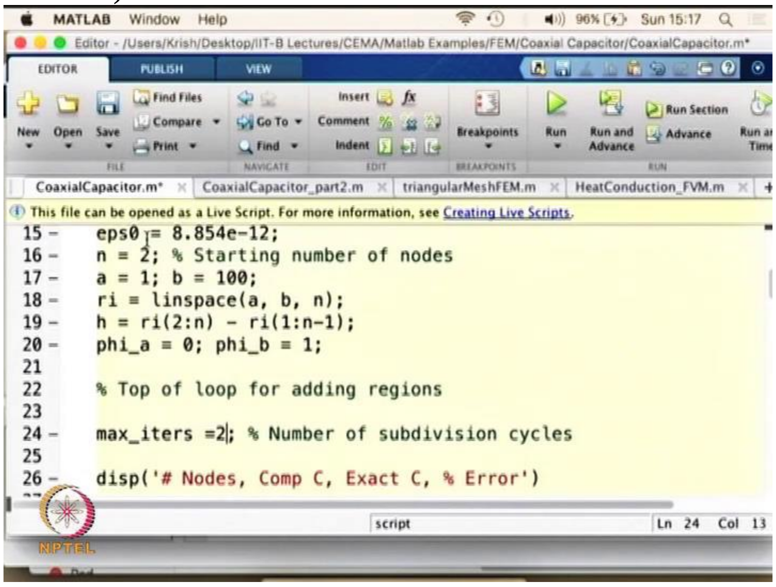
(Refer Slide Time: 16: 13)



```
Undefined function or variable 'stop'.

Error in HeatConduction_FVM (line 77)
    stop

>> CoaxialCapacitor
# Nodes, Comp C, Exact C, % Error
2 28.377594 12.080188 134.910196
>>
```

So if you go to the code and see the error we are computing will be able to see the exact value is 12.08 Pico fared whereas the computed value is 28.37 Pico fared and the error is roughly 134 percent which is quite high.

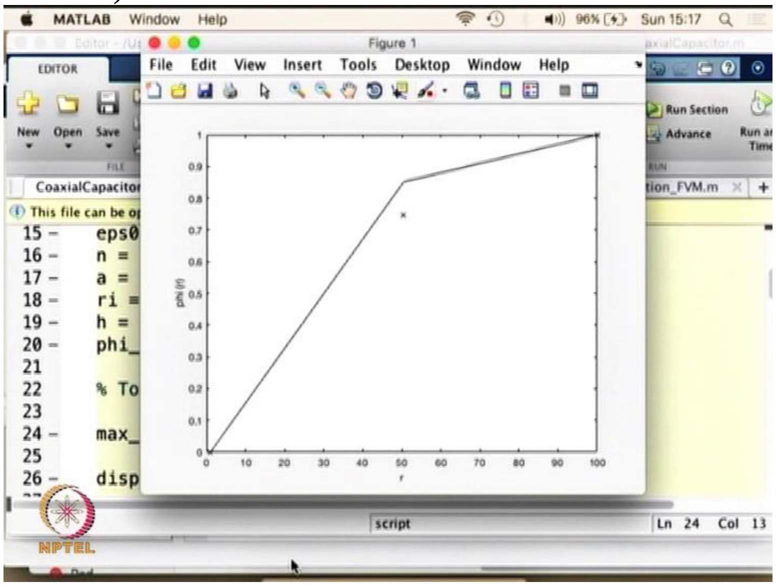
(Refer Slide Time: 16: 40)



```
eps0 = 8.854e-12;
n = 2; % Starting number of nodes
a = 1; b = 100;
ri = linspace(a, b, n);
h = ri(2:n) - ri(1:n-1);
phi_a = 0; phi_b = 1;

% Top of loop for adding regions

max_iters =2; % Number of subdivision cycles

disp('# Nodes, Comp C, Exact C, % Error')
```

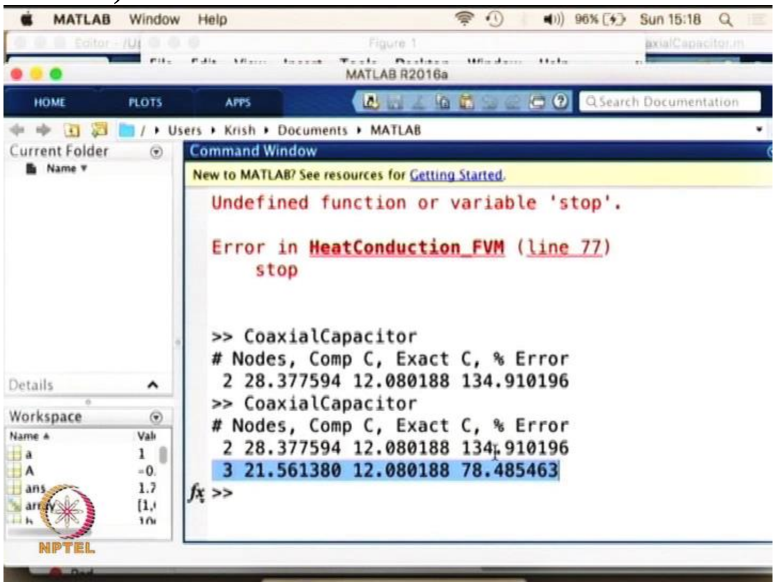
So now let us increase the number of points and we are going to do that in a manner that is easy for us to do. So that is why I said there is going to be another function of this max iters parameter. So if put this value to be 2 what it does is it keeps increasing the number of nodes after each iteration. The first iteration it will have n equal to 2, the second iteration it will increase n by 1, third iteration it will increase by another 1 so on and so forth.

(Refer Slide Time: 17: 09)



So let us iterate it for twice and we will see the results will be clear. So the computed value are the value that we see here. So any way these are boundary values and the computed value is here. And the exact value is there, so the exact value is actually node of this line or this point or this boundary of this line. So you see that this is the difference what we are having. So instead of having let us say 8 point something the value what we see is roughly 7, and this is the difference in the potential that we are computing and similarly the capacitance value will also have its own error. And we can see the error while we are simulating it.

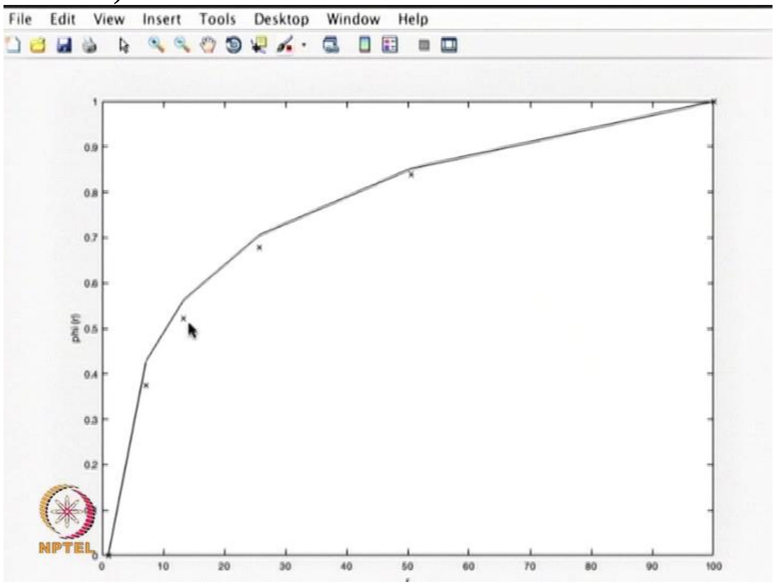
(Refer Slide Time: 17: 54)



```
Undefined function or variable 'stop'.

Error in HeatConduction_FVM (line 77)
    stop


>> CoaxialCapacitor
# Nodes, Comp C, Exact C, % Error
 2 28.377594 12.080188 134.910196
>> CoaxialCapacitor
# Nodes, Comp C, Exact C, % Error
 2 28.377594 12.080188 134.910196
 3 21.561380 12.080188 78.485463
>>
```

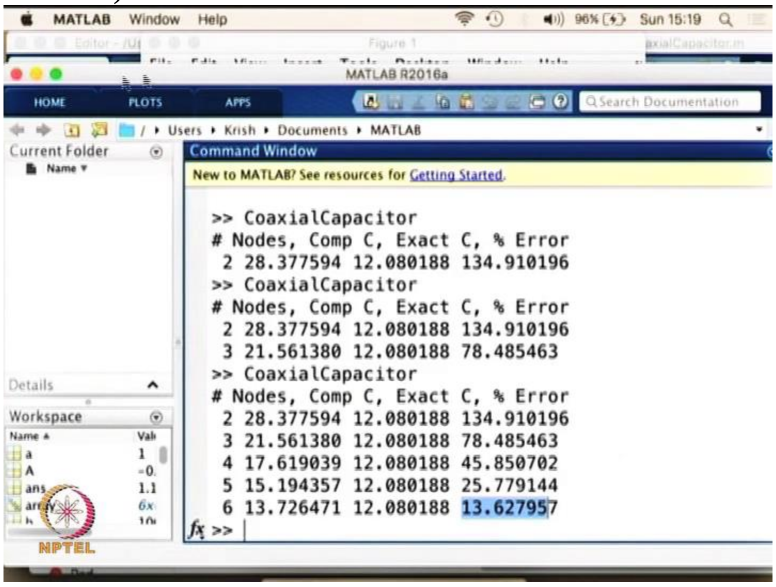
So the first line is exactly the same line as before. And the second line is increasing the number of nodes by 1 and did you see that from 28 which was the earlier computation we are coming to 21 whereas the error is also reducing from 134 percent to 78 percent. We have improved but still huge error.

(Refer Slide Time: 18: 18)



And now we are going to increase the number of iteration let us say by 5. So we will have more points in the simulation. And that is exactly what we see here. Still the exact value and the computed value is having some error so the difference between the tip of this line and the cross mark what we have is the error in the potential computation.

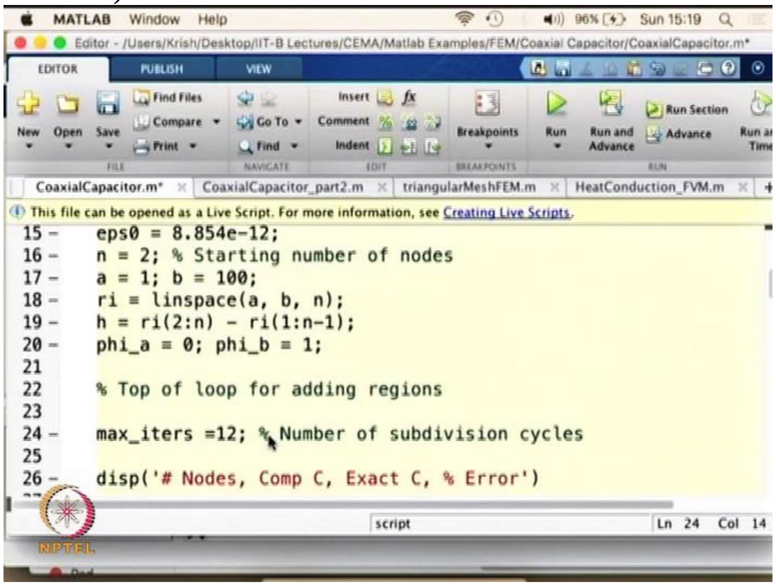
(Refer Slide Time: 18: 49)



```
>> CoaxialCapacitor
# Nodes, Comp C, Exact C, % Error
 2 28.377594 12.080188 134.910196
>> CoaxialCapacitor
# Nodes, Comp C, Exact C, % Error
 2 28.377594 12.080188 134.910196
 3 21.561380 12.080188 78.485463
>> CoaxialCapacitor
# Nodes, Comp C, Exact C, % Error
 2 28.377594 12.080188 134.910196
 3 21.561380 12.080188 78.485463
 4 17.619039 12.080188 45.850702
 5 15.194357 12.080188 25.779144
 6 13.726471 12.080188 13.627957
>>
```

And the error in the computation of capacitance is what we saw or is what we are seeing in the screen here. So the first two lines are the same as this one. And then for four points we have 45 percent error, for 5 points we have 25 percent error, and for 6 points it is 13 percent error. So the error is converging and that is a good sign.

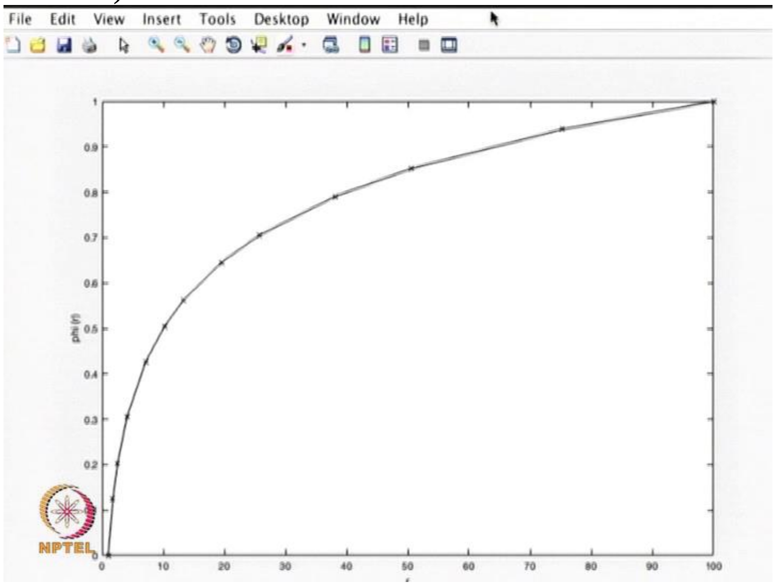
(Refer Slide Time: 19: 09)



```
eps0 = 8.854e-12;
n = 2; % Starting number of nodes
a = 1; b = 100;
ri = linspace(a, b, n);
h = ri(2:n) - ri(1:n-1);
phi_a = 0; phi_b = 1;

% Top of loop for adding regions

max_iters =12; % Number of subdivision cycles

disp('# Nodes, Comp C, Exact C, % Error')
```

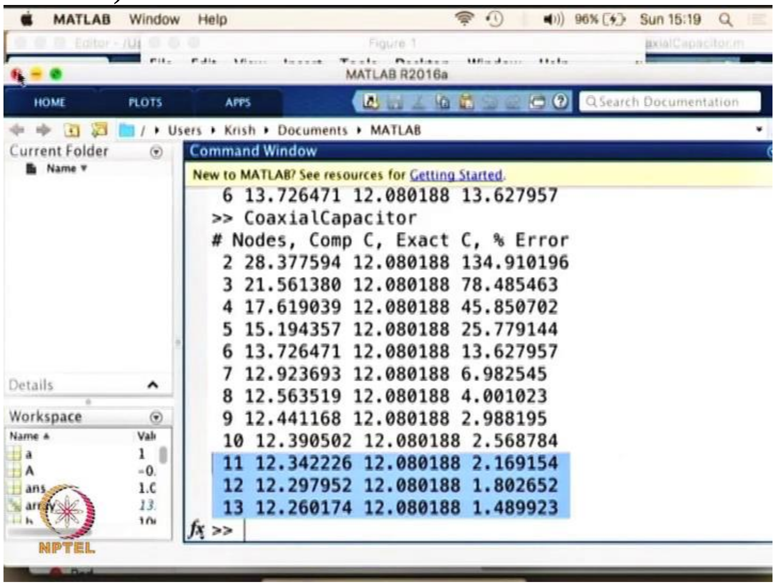
So let us simulate this problem for let us say 10 or may be 12.

(Refer Slide Time: 19: 23)



And now you see the computed value which are the cross marks is exactly sitting on the tip of the line segments which means that the computed value and the exact values are almost the same.

(Refer Slide Time: 19: 34)

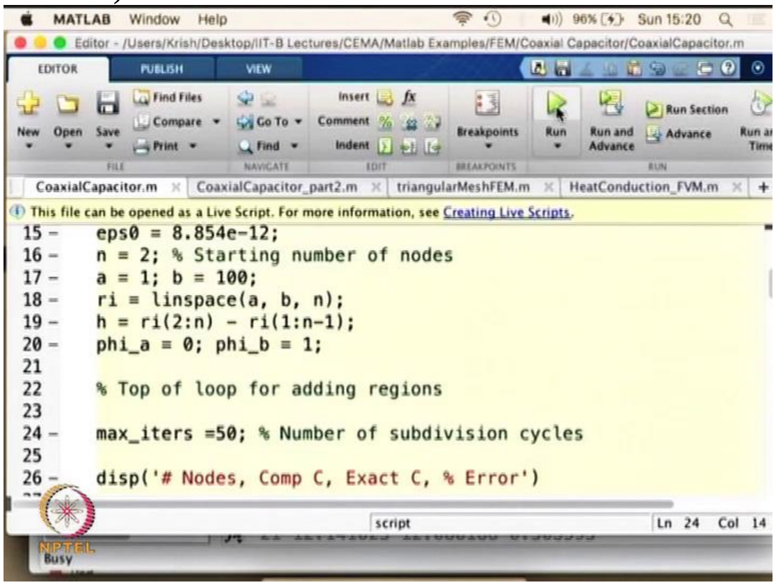


New to MATLAB? See resources for Getting Started.

```
 6 13.726471 12.080188 13.627957
>> CoaxialCapacitor
# Nodes, Comp C, Exact C, % Error
 2 28.377594 12.080188 134.910196
 3 21.561380 12.080188 78.485463
 4 17.619039 12.080188 45.850702
 5 15.194357 12.080188 25.779144
 6 13.726471 12.080188 13.627957
 7 12.923693 12.080188 6.982545
 8 12.563519 12.080188 4.001023
 9 12.441168 12.080188 2.988195
 10 12.390502 12.080188 2.568784
 11 12.342226 12.080188 2.169154
 12 12.297952 12.080188 1.802652
 13 12.260174 12.080188 1.489923
fx >>
```

So this you will also see in the result of the error, so when you have 13 points and 12 points the error is very small you have 1 percent hardly.

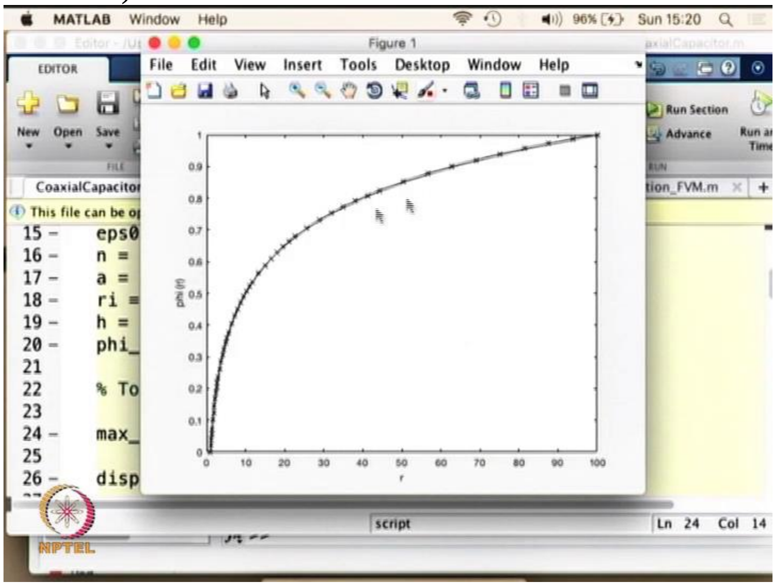
(Refer Slide Time: 19: 48)



```
eps0 = 8.854e-12;
n = 2; % Starting number of nodes
a = 1; b = 100;
ri = linspace(a, b, n);
h = ri(2:n) - ri(1:n-1);
phi_a = 0; phi_b = 1;

% Top of loop for adding regions

max_iters =50; % Number of subdivision cycles

disp('# Nodes, Comp C, Exact C, % Error')
```

And if you a very much on a very high level accurate simulation you can increase the number of nodes at your will and you will see that the errors are going to be quite low for this kind of problem.

(Refer Slide Time: 20: 03)



 you will see there are more number of nodes here and the error is going to be very very low. So what we have done through this program is we have shown how a simple finite element method can teach us quite a lot of aspects about the finite element simulation itself. First of all it tells us the convergence property of the finite element code. And second of all what it teaches is how the structuring of the program is done. And it also tells us how the value that we are computing whether it is going to be energy stored or it is going to be capacitance or it is going to be potential is going to vary based on the approximation that we are using.

So I encourage you to take this code and practice it for yourself. It is a very good example for you to start looking at Finite element method. The source of the code is given here. It is a excellent book which talks a lot about numerical method for electrostatic problems. And we will ask you to practice such kind of codes for you to get understanding of the finite element method in one dimensional case and later on when we do some problems in a two dimensional case it will be of very high value. So I encourage you to do that. And in the next problem what I am going to do is I am going to change the approximation for the finite element. Right now we have said that the points that we are going to join are going to be by lines. So now in the second step of this problem I am going to use a second order basis function which will have a more accurate performance. This is what I am going to teach you in the next module. Thank you!