Electromagnetic Theory, Adv., Report Assignment 2

Dec. 16, 2020

Due date: Jan. 13, 2021

Submit to: PandA

大村研究室

1030-32-5617

Liu Yin

1. One-dimensional electromagnetic finite element analysis using an unequally spaced mesh.

Problem description:

As shown in Fig. 1, there are two parallel plate capacitors with sufficiently large areas. Three dielectrics with permittivity of $\varepsilon_1 = 1, \varepsilon_2 = 2, \varepsilon_3 = 1$ and thickness of $d_1 = 0.4, d_2 = 0.2, d_3 = 0.4$ respectively are existing between the capacitors. In addition, $V_0 = 0, V_1 = 1$. What is the value between V_0 and V_1?

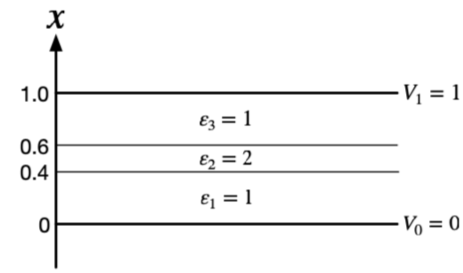


Fig. 1 Problem model

Solution:

One dimensional finite element method is used here to solve electrostatic field equation $-\nabla \cdot \varepsilon \nabla V = \rho$ to obtain value of $V(x)$ between V_0 and V_1. Since $\rho = 0$, we have conditions as follows:

$$-\frac{d}{dx}\left(\varepsilon \frac{dV}{dx}\right) = 0 \begin{cases} V(0) = 0, V(1) = 1 \\ \varepsilon(x) = \begin{cases} 1 \ (0 < x < 0.4) \\ 2 \ (0.4 < x < 0.6) \\ 1 \ (0.6 < x < 1) \end{cases} \end{cases} \tag{1}$$

step 1:

Split the domain $0 < x < 1$ into several nonoverlapping elements. Here, as request, the whole domain is split into unequally space elements(meshes), as shown in Fig. 2.

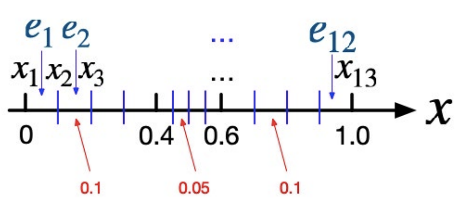


Fig. 2 domain partition

The whole region is split into 12 subregions. Thus, we have 13 nodes ($x_1 \sim x_{13}$) and 12 elements ($e_1 \sim e_{12}$) correspondingly.

Assume we already had the potential value at each point of 13 nodes, using interpolation method, we are able to obtain the potential values between nodes, i.e., elements.

step 2:

The basic linear interpolation method is implemented here. For the value of one element between two nodes, as the diagram shown in Fig. 3, we have the approximate potential value $V^*(x)$ as follows:

$$V^*(x) = V(x_i) + \frac{x-x_i}{x_{i+1}-x_i}[V(x_{i+1}) - V(x_i)] \tag{2}$$
$$= \frac{x_{i+1}-x}{x_{i+1}-x_i}V(x_i) + \frac{x-x_i}{x_{i+1}-x_i}V(x_{i+1})$$

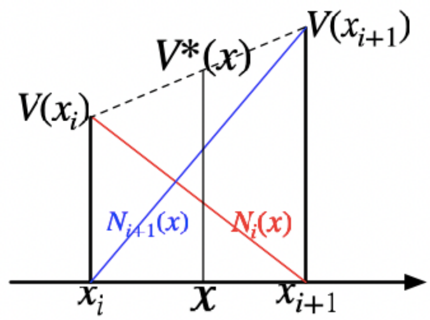


Fig. 3 1D linear interpolation

We define shape function $N_i(x)$ as,

$$N_i(x) = \begin{cases} \frac{x-x_{i-1}}{h_{i-1}} & x_{i-1} \le x \le x_i \\ \frac{x_{i+1}-x}{h_i} & x_i \le x \le x_{i+1}, \\ 0 & Otherwise \end{cases} \tag{3}$$

where i is the label of node, and $h_i = x_{i+1} - x_i$.

Particularly, at the front and end edge,

$$N_1(x) = \begin{cases} \frac{x_2-x}{h_1} & x_1 \le x \le x_2 \\ 0 & Otherwise \end{cases}, \tag{4}$$

and

$$N_{n+1}(x) = \begin{cases} \frac{x-x_n}{h_n} & x_n \le x \le x_{n+1} \\ 0 & Otherwise \end{cases}, \tag{5}$$

where n is the total number of elements.

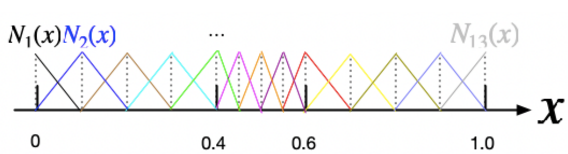


Fig. 4 Shape functions

Fig. 4 shows shape functions for different elements, thus, for the whole region, approximate electric potential $V(x)$ can be represented as:

$$V^*(x) = N_1(x)V(x_1) + N_2(x)V(x_2) + \cdots + N_{13}(x)V(x_{13}) = \sum_{j=1}^{13} N_j(x)V(x_j) \tag{6}$$

step 3:

For the authentic $V(x)$, after taking the integration of Eq. (1) in the whole region, we have

$$\int_{x_1}^{x_{13}} [-\frac{d}{dx}\left(\varepsilon\frac{dV}{dx}\right)]dx = 0. \tag{7}$$

However, it is not suitable for the potential value from Eq. (6). Yet, we can force it satisfiable by adding a weighting function $W(x)$, that is

$$\int_{x_1}^{x_{13}} W(x)[-\frac{d}{dx}\left(\varepsilon\frac{dV^*}{dx}\right)]dx = 0 \tag{8}$$

Although there are several methods for choosing different weighting functions, here, we use *Galerkin* method, which lets the weighting function $W(x)$ be the shape function $N(x)$. Hence, Eq. (8) becomes to

$$\int_{x_1}^{x_{13}} N_i(x)[-\frac{d}{dx}\left(\varepsilon\frac{dV^*}{dx}\right)]dx = 0, \quad (i = 1, \ldots, 13) \tag{9}$$

Through integration by parts, Eq. (9) converts to

$$\int_{x_1}^{x_{13}} \frac{dN_i}{dx}\varepsilon\frac{dV^*}{dx}dx - \left[N_i\varepsilon\frac{dV^*}{dx}\right]_0^1 = 0, \quad (i = 1, \ldots, 13) \tag{10}$$

Substitute V^* by Eq. (6), we obtain

$$\int_{x_1}^{x_{13}} \frac{dN_i}{dx}\varepsilon\sum_{j=1}^{13}[\frac{dN_j}{dx}V\left(x_j\right)]\,dx - \left[N_i\varepsilon\frac{dV^*}{dx}\right]_{x_1}^{x_{13}} = 0, \quad (i = 1, \ldots, 13) \tag{11}$$

step 4:

In order to solve Eq. (11), we need focus on each element. For the first element, the integration region is from x_1 to x_2, and from the shape function, we see only N_1 and N_2 effects on it.

Thus, for element 1, we have

$$\int_{x_1}^{x_2} \frac{dN_i}{dx}\varepsilon\sum_{j=1}^{2}[\frac{dN_j}{dx}V\left(x_j\right)]\,dx - \left[N_i\varepsilon\frac{dV^*}{dx}\right]_{x_1}^{x_2} = 0, \quad (i = 1,2) \tag{12}$$

Define

$$A_{ij} = \int \varepsilon\frac{dN_i}{dx}\frac{dN_j}{dx}dx \tag{13}$$

where ε should be different value for the junction of two dielectrics.

Eq. (12) can be represented as follows:

$$\begin{cases} A_{11}V(x_1) + A_{12}V(x_2) - N_1\left[\varepsilon\frac{dV^*}{dx}\right]_{x_1}^{x_2} = 0 \\ A_{21}V(x_1) + A_{22}V(x_2) - N_2\left[\varepsilon\frac{dV^*}{dx}\right]_{x_1}^{x_2} = 0 \end{cases} \tag{14}$$

In matrix form,

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} V(x_1) \\ V(x_2) \end{bmatrix} + \begin{bmatrix} \left(\varepsilon \frac{dV^*}{dx}\right)_{x=x_1} \\ -\left(\varepsilon \frac{dV^*}{dx}\right)_{x=x_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{15}$$

In the same fashion, for element 2, we have

$$\begin{bmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} V(x_2) \\ V(x_3) \end{bmatrix} + \begin{bmatrix} \left(\varepsilon \frac{dV^*}{dx}\right)_{x=x_2} \\ -\left(\varepsilon \frac{dV^*}{dx}\right)_{x=x_3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{16}$$

and so forth.

Finally, we sum all elements up and obtain the matrix form of whole domain as,

$$\begin{bmatrix} A_{11} & A_{12} & 0 & \cdots & 0 & 0 & 0 \\ A_{21} & 2A_{22} & A_{23} & \cdots & 0 & 0 & 0 \\ 0 & A_{32} & 2A_{33} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2A_{1111} & A_{1112} & 0 \\ 0 & 0 & 0 & \cdots & A_{1211} & 2A_{1212} & A_{1213} \\ 0 & 0 & 0 & & 0 & A_{1312} & A_{1313} \end{bmatrix} \begin{bmatrix} V(x_1) \\ V(x_2) \\ V(x_3) \\ \vdots \\ V(x_{11}) \\ V(x_{12}) \\ V(x_{13}) \end{bmatrix} + \begin{bmatrix} \left(\varepsilon \frac{dV^*}{dx}\right)_{x=x_1} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ -\left(\varepsilon \frac{dV^*}{dx}\right)_{x=x_{13}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{17}$$

From the initial boundary condition, we have $\left(\varepsilon \frac{dV^*}{dx}\right)_{x=x_1} = -\left(\varepsilon \frac{dV^*}{dx}\right)_{x=x_{13}} = 0$, $V(x_1) = 0$ and $V(x_{13}) = 1$. In addition, A_{ij} can be obtained from shape functions which are related to the initial partition of domain.

Therefore, we obtain the final form to calculate potential values of other positions as

$$\begin{bmatrix} 2A_{22} & A_{23} & \cdots & 0 & 0 \\ A_{32} & 2A_{33} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 2A_{1111} & A_{1112} \\ 0 & 0 & \cdots & A_{1211} & 2A_{1212} \end{bmatrix} \begin{bmatrix} V(x_2) \\ V(x_3) \\ \vdots \\ V(x_{11}) \\ V(x_{12}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ A_{1213} \end{bmatrix}. \tag{18}$$

result:

From MATLAB code attached behind, we obtain result as

$[V(x_1), V(x_2), V(x_3), V(x_4), V(x_5), V(x_6), V(x_7), V(x_8), V(x_9), V(x_{10}), V(x_{11}), V(x_{12}), V(x_{13})]^T$ = [0, 0.1111, 0.2222,0.3333, 0.4444, 0.4722, 0.5, 0.5278, 0.5556, 0.6667, 0.7778, 0.8889, 1]^T.

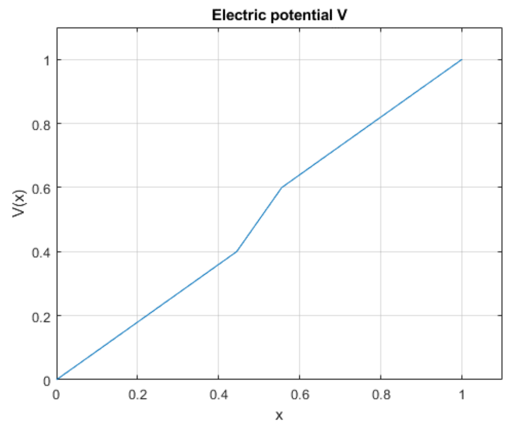


Fig. 5 Result calculated from MATLAB code

MATLAB code:

```
x = [0, 0.1, 0.2, 0.3, 0.4, 0.45, 0.5, 0.55, 0.6, 0.7, 0.8, 0.9, 1]; % partition
epsilon = [1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1]; % permittivity
node_index = [1, 2, 3, 4, 5, 6, 7, 8, 9,  10, 11, 12; % node
              2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13];
A = zeros(13,13);
element_A = zeros(2,2);
node = zeros(2,1);
dndx = zeros(2,1);

% calculate A matrix
for element = 1:12
   node(1) = node_index(1, element);
   node(2) = node_index(2, element);
   width = x(node(2)) - x(node(1));
   dndx(1) = -1/width;
   dndx(2) = 1/width;

   for i = 1:2
       for j= 1:2
           element_A(i,j) = epsilon(element) * dndx(i) * dndx(j) * width;
       end
   end

   for i = 1:2
       for j = 1:2
           A(node(i),node(j)) = A(node(i), node(j)) + element_A(i,j);
       end
   end
end

% calculate result
A_seg = A(2:12,2:12);
B = zeros(11,1);
B(11,1) = -A(12,13);
V = zeros(13,1);
V(13) = 1;
V(2:12) = A_seg\B;

% plot
figure,
plot(V, x)
title("Electric potential V")
axis([ 0.000, 1.100, 0.000, 1.100 ])
grid on
ylabel("V(x)")
xlabel("x")
```

2. Two-dimensional electromagnetic finite element analysis using an unstructured mesh.

<u>Problem description</u>:

As shown in Fig. 6, use 2-D finite element method to calculate electric potential in the area between two bold lines.

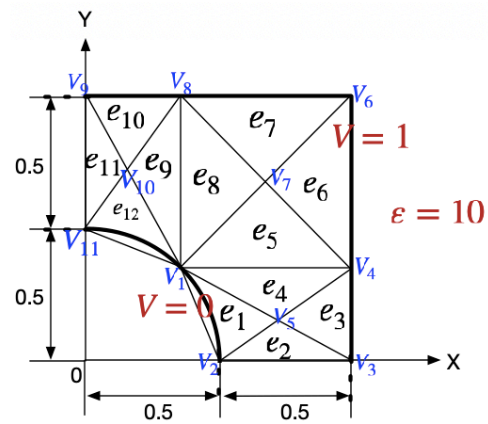


Fig. 6 2-D problem model

<u>Solution:</u>

Similar to Eq. (1), since charge density $\rho = 0$ here, we have electrostatic field equation,

$$-\frac{\partial}{\partial x}\left(\varepsilon\frac{\partial V}{\partial x}\right)-\frac{\partial}{\partial y}\left(\varepsilon\frac{\partial V}{\partial y}\right)=0. \tag{19}$$

In addition, electric potentials of V_1, V_2, V_{11} are 0, potentials at V_3, V_4, V_6, V_8, V_9 are 1, and permittivity in the region $\varepsilon = 10$.

step 1:

The whole domain is split into 12 elements with 11 nodes, as shown in Fig. 6.

step 2:

In 2D condition, there are several interpolation methods, such as using linear shape functions or quadratic functions. Here, we use the simplest linear interpolation method.

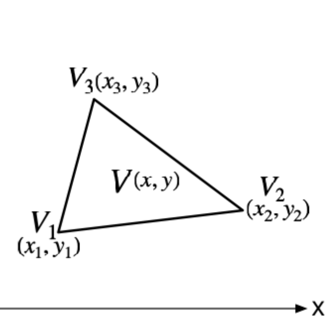


Fig. 7 2-D triangular element

As shown in Fig. 6 and 7, we use triangular element as basic element. Implementing linear interpolation method, we have potential value at inner point of element as,

$$V = \alpha_1 + \alpha_2 x + \alpha_3 y = \begin{bmatrix} 1 & x & y \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} \tag{20}$$

Thus, potential at node 1, 2 and 3 in matrix form represents ad,

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}, \tag{21}$$

hence,

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

$$V = \begin{bmatrix} 1 & x & y \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 1 & x & y \end{bmatrix} \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \tag{22}$$

Similar to 1-D condition, we define shape function as,

$$N(x, y) = \begin{bmatrix} 1 & x & y \end{bmatrix} \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \tag{23}$$

Triangle shown in Fig. 7 has an area expression:

$$S = \frac{1}{2}[(x_1 y_2 - x_2 y_1) + (x_3 y_1 - x_1 y_3) + (x_2 y_3 - x_3 y_2)] \tag{24}$$

Thus, Eq. (23) converts to

$$\begin{cases} N_1(x, y) = \frac{1}{2S}[(x_2 y_3 - x_3 y_2) + (y_2 - y_3)x + (x_3 - x_2)y] \\ N_2(x, y) = \frac{1}{2S}[(x_3 y_1 - x_1 y_3) + (y_3 - y_1)x + (x_1 - x_3)y] \\ N_3(x, y) = \frac{1}{2S}[(x_1 y_2 - x_2 y_1) + (y_1 - y_2)x + (x_2 - x_1)y] \end{cases} \tag{25}$$

Define

$$\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} = \begin{bmatrix} x_2 y_3 - x_3 y_2 & y_2 - y_3 & x_3 - x_2 \\ x_3 y_1 - x_1 y_3 & y_3 - y_1 & x_1 - x_3 \\ x_1 y_2 - x_2 y_1 & y_1 - y_2 & x_2 - x_1 \end{bmatrix}, \tag{26}$$

thus,

$$S = \frac{1}{2}(a_1 + a_2 + a_3) \tag{27}$$

and

$$\begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} = \frac{1}{2S} \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}. \tag{28}$$

Therefore, approximate electric potential at whole domain in Fig. 6 can be expressed as

$$V^*(x,y) = \sum_{j=1}^{11} N_j(x,y)V_j \tag{29}$$

step 3:

Similar to 1-D condition, for the authentic potential value, the integration of Eq. (19) at whole region is 0, yet different for approximate potential in Eq. (29). By *Galerkin* method, using shape function Eq. (28) as weighted function. Thus, we have

$$\iint N_i [-\frac{\partial}{\partial x}\left(\varepsilon \frac{\partial V^*}{\partial x}\right) - \frac{\partial}{\partial y}\left(\varepsilon \frac{\partial V^*}{\partial y}\right)] dxdy = 0 \tag{30}$$

Through integration by parts and Green formula, we have

$$\iint \varepsilon \nabla N_i \cdot \nabla V^* dxdy - \oint \varepsilon N_i \left(\frac{\partial V^*}{\partial x} dy - \frac{\partial V^*}{\partial y} dx\right) = 0 \tag{31}$$

From Eq. (20), we see $\oint \varepsilon N_i \left(\frac{\partial V^*}{\partial x} dy - \frac{\partial V^*}{\partial y} dx\right) = 0$. Substitute V^* by Eq. (29), Eq. (31) converts into

$$\iint \varepsilon \nabla N_i \cdot \sum_{j=1}^{11}\left(\nabla N_j V_j\right) dxdy = 0, \ \ i = 1, \dots, 11 \tag{32}$$

Define

$$A_{ij} = \iint \varepsilon \nabla N_i \cdot \nabla N_j dxdy = \iint \varepsilon \left(\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}\right) dxdy, \ \ i,j = 1, \dots, 11 \tag{33}$$

Thus, Eq. (32) converts into

$$A_{ij} V_j = 0, \ \ i,j = 1, \dots, 11 \tag{34}$$

step 4:

Using Eq. (34) to calculate potential value inside element, take element in Fig. 7 as example, from Eq. (28), we have

$$A = \frac{\varepsilon}{4S} \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = \frac{\varepsilon}{4S} \begin{bmatrix} b_1 b_1 + c_1 c_1 & b_1 b_2 + c_1 c_2 & b_1 b_3 + c_1 c_3 \\ b_1 b_2 + c_1 c_2 & b_2 b_2 + c_2 c_2 & b_2 b_3 + c_2 c_3 \\ b_1 b_3 + c_1 c_3 & b_2 b_3 + c_2 c_3 & b_3 b_3 + c_3 c_3 \end{bmatrix}, \tag{35}$$

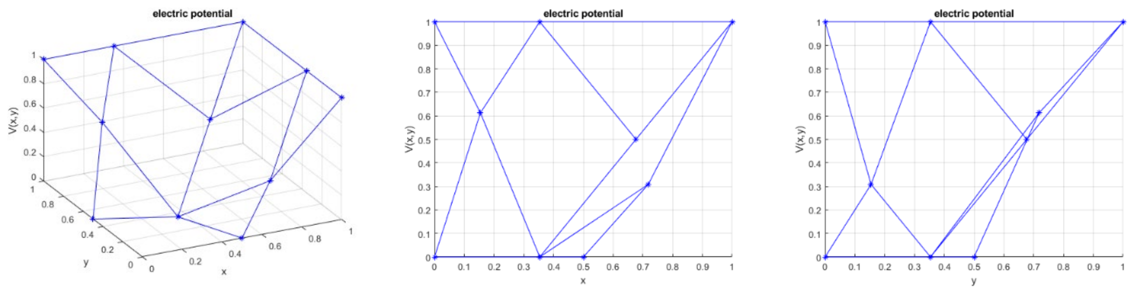
where b and c are defined in Eq. (26).

In the same pattern, we can obtain whole coefficient matrix A of sparse format and consider the initial boundary condition to obtain final result.

From the MATLAB code attached behind, we have the potential values in Fig. 6 as

$[V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V_{10}, V_{11}]^T$ = $[0, 0, 1, 1, 0.3084, 1, 0.5, 1, 1, 0.6126, 0]^T$

Since nodes in Fig. 6 are relatively small, we give their plot as follows:



a) x-y-V   b) x-V   c) y-V

Fig. 8 Electric potential at each node

<u>MATLAB code:</u>

```
epsilon = 10; % permittivity
V_0 = 1;
node_index = [1, 2, 3, 1, 1, 4, 6, 1, 1,  8,  9,  1;
              2, 3, 4, 4, 4, 6, 7, 7, 8,  9,  10, 10;
              5, 5, 5, 5, 7, 7, 8, 8, 10, 10, 11, 11]; % node
% x position
x_1  = 0.5/sqrt(2);
x_2 = 0.5;
x_3 = 1.0;
x_4 = 1.0;
x_5 = (1 - 0.5*x_1)/(1.5 - x_1);
x_6 = 1.0;
x_7 = 0.5 * (x_1 + 1);
x_8 = x_1;
x_9 = 0;
x_10 = x_1/0.5 * ((1 - 0.5*x_1)/(1.5 - x_1) - 0.5);
x_11 = 0;
% y position
y_1 = 0.5/sqrt(2);
y_2 = 0;
y_3 = 0;
y_4 = y_1;
y_5 = y_1/0.5 * ((1 - 0.5*y_1)/(1.5 - y_1) - 0.5);
y_6 = 1.0;
y_7 = 0.5 * (y_1 + 1);
y_8 = 1.0;
y_9 = 1.0;
y_10 = (1 - 0.5*y_1)/(1.5 - y_1);
y_11 = 0.5;

x_node = [x_1, x_2, x_3, x_1, x_1, x_4, x_6, x_1, x_1,  x_8,  x_9,  x_1;
          x_2, x_3, x_4, x_4, x_4, x_6, x_7, x_7, x_8,  x_9,  x_10, x_10;
          x_5, x_5, x_5, x_5, x_7, x_7, x_8, x_8, x_10, x_10, x_11, x_11];
y_node = [y_1, y_2, y_3, y_1, y_1, y_4, y_6, y_1, y_1,  y_8,  y_9,  y_1;
          y_2, y_3, y_4, y_4, y_4, y_6, y_7, y_7, y_8,  y_9,  y_10, y_10;
          y_5, y_5, y_5, y_5, y_7, y_7, y_8, y_8, y_10, y_10, y_11, y_11];

noe = 12; % element number
nond = 11; % node number
imat = ones(9, noe);  jmat = ones(9, noe);  mat = zeros(9, noe); % store
% calculate coefficient matrix
for  i = 1:noe
    x = x_node(:,i);
    y = y_node(:,i);
    a = [ x(2)*y(3)-x(3)*y(2);  x(3)*y(1)-x(1)*y(3);  x(1)*y(2)-x(2)*y(1) ];
    b = [ y(2)-y(3);  y(3)-y(1);  y(1)-y(2) ];
    c = [ x(3)-x(2);  x(1)-x(3);  x(2)-x(1) ];
    S = sum(a) / 2.0;
    A_e = (b * b' + c * c') * epsilon / (4.0 * S) ;
    imat(:, i) = repmat( node_index(:, i), 3, 1 );
    jmat(:, i) = reshape( repmat( node_index(:, i), 1, 3 )' , 9, 1 );
```

```
    mat(:, i) = A_e(:);
end
A = sparse(imat, jmat, mat); % sparse matrix
% boundary condition
earth = [1, 2, 11];
v_0 = [3, 4, 6, 8, 9];
dirichlet = [earth v_0];
A(dirichlet, :) = 0.0;
A(dirichlet, dirichlet) = speye( length( dirichlet ) );
b = zeros(nond, 1);
b(v_0) = V_0;
result = A \ b;
```

Reference:

[1] Heinrich, Juan C., (2017). *The finite element method : basic concepts and applications with MATLAB, MAPLE, and COMSOL.* CRC Press.

[2] 中田高義, 電気工学の有限要素法(第２版),森北出版株式会社, 2004.