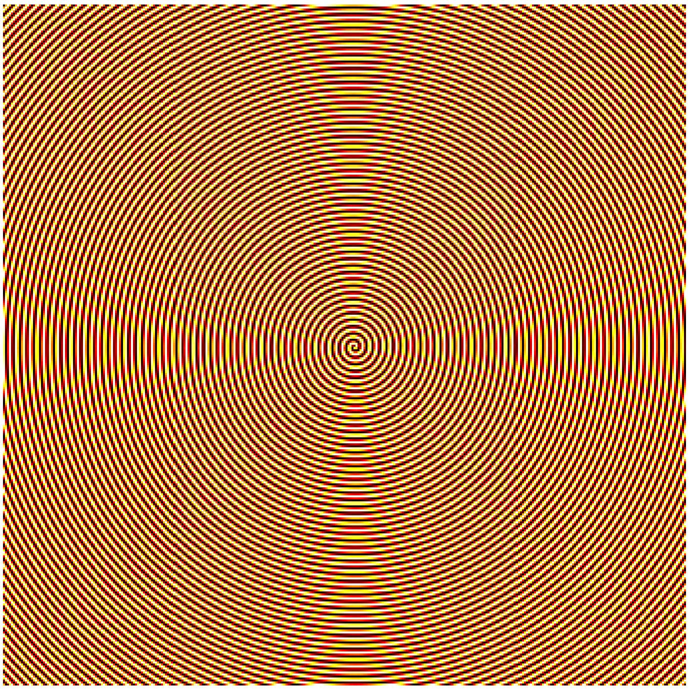
電磁學(上)

Matlab Homework



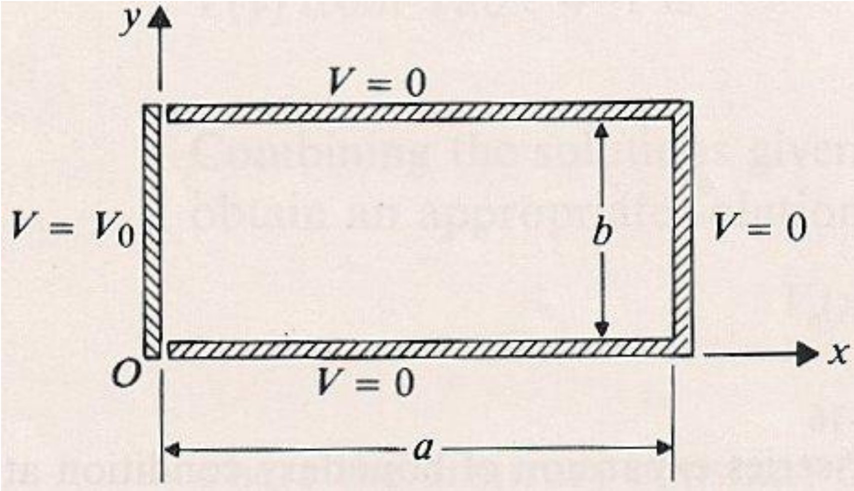
```
[x,y]=meshgrid(-900:1000,-900:1000);
[th,r]=cart2pol(x - 75,y - 50);
Img=sin(r/3 + th);
imagesc(Img);
colormap(hot);
axis equal;
axis off
```

杜銘航, Ming-Hang Du

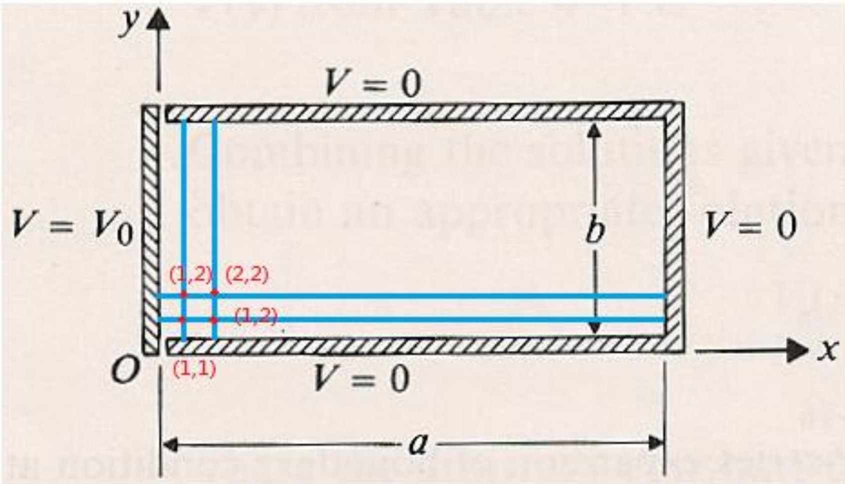
97323020

2010/12/20

Problem：



From desiring, the top, bottem, and right plates are grounded. The plate on the left side is maintained at a constant potential $V_o = 100$.



We first assume all the V to be zero except the points on $x = 1$.

Then, from equation (5), point (1, 1) is known to be $\frac{1}{4}[0 + V_o + 0 + 0]$ $= \frac{1}{4}V_o$. Therefore, point (1, 2)equals to $\frac{1}{4}\left[0 + V_o + \frac{1}{4}V_o + 0\right] = \frac{5}{16}V_o$,

… and son on.

After that, w□ can calculate them again to make them more precisely. This method is called iteraition. That is, we can use Matlab to plot the potential distribution by iteration.

Theoretical analysis：

First Derivative − Central Difference

$$f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x}$$

$$f'(x) = \frac{f\left(x+\frac{\Delta x}{2}\right) - f\left(x-\frac{\Delta x}{2}\right)}{\Delta x}$$

Second Derivative − Central Difference

Using the first derivative central difference approximation

$$f''(x) = \frac{f'\left(x+\frac{\Delta x}{2}\right) - f'\left(x-\frac{\Delta x}{2}\right)}{\Delta x}$$

$$= \frac{\frac{f(x+\Delta x) - f(x)}{\Delta x} - \frac{f(x) - f(x-\Delta x)}{\Delta x}}{\Delta x}$$

$$= \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{\Delta x^2}$$

Laplace equation in rectangular coordinates (2D)

$$\nabla^2 V = \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0 - - - (1)$$

where $x = x_i = i\Delta x \ (i = 1,2 \ldots)$

$y = y_j = j\Delta y \ (j = 1,2 \ldots)$

$$I_m x \ \ V''(x,y) = \frac{V(x+\Delta x, y) - 2V(x,y) + V(x-\Delta x, y)}{\Delta x^2} - - - (2)$$

$$I_m y \ \ V''(x,y) = \frac{V(x, y+\Delta y) - 2V(x,y) + V(x, y-\Delta y)}{\Delta y^2} - - - (3)$$

Substituting (2), (3) into (1)

$$\nabla^2 V = 0$$

$$= \frac{V(x+\Delta x, y) - 2V(x,y) + V(x-\Delta x, y)}{\Delta x^2} + \frac{V(x, y+\Delta y) - 2V(x,y) + V(x, y-\Delta y)}{\Delta y^2} - - - (4)$$

If $\Delta x = \Delta y = \Delta$, then (4) becomes

$$V(x+\Delta, y) + V(x-\Delta, y) + V(x, y+\Delta) + V(x, y-\Delta) = 4\,V(x,y)$$

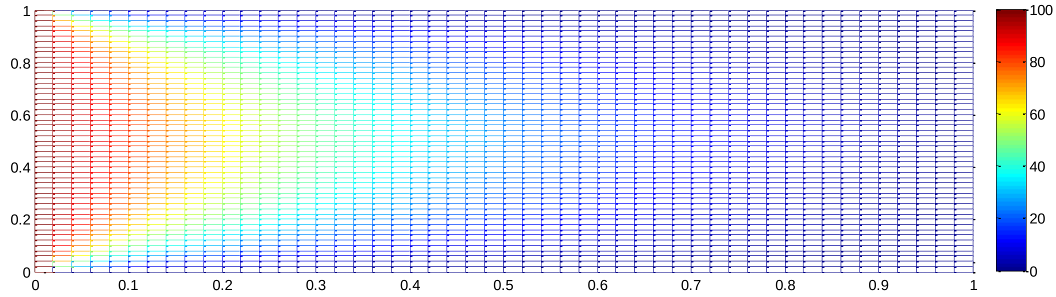
$$\therefore \mathbf{V(x,y) = \frac{1}{4}[V(x+\Delta, y) + V(x-\Delta, y) + V(x, y+\Delta) + V(x, y-\Delta)]} - - - (5)$$

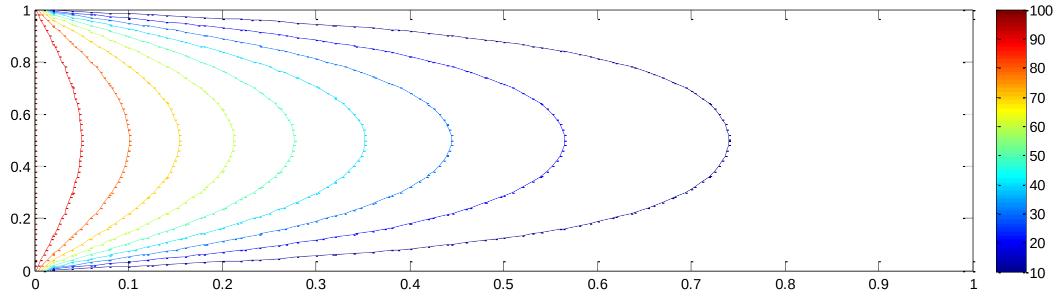
Matlab Code：

```
clear ;clc;

a = input('Please input a:');     % input a
b = input('Please input b:');     % input b
Vo = 100;     % Vo = 100 at x=0

x = 0 : 0.02 : a;     % the interval of x is 0.02
y = 0 : 0.02 : b;     % the interval of y is 0.02

[X,Y] = meshgrid(x,y);     % make x and y line space a two dimention matrix

V(2:length(x), 1:length(y)) = 0;     % the initial value of x>2 is 0
V(1, 1:length(y)) = Vo;     % the initial value of x=1 is Vo

% original equation at p.182 eq.4-114
% V=4*Vo/pi*((sinh((n*pi*(a-X))/b))/n*sinh((n*pi*a)/b)*Y);

for (k = 1 : 1 : 10000)    % number of iteration
    for (i = 2 : 1 : length(x)-1)
    % from x = 2 to margin-1, beacause margin is 0
        for (j = 2 : 1 : length(y)-1)
        % from y = 2 to margin-1 for margin is 0
            V(i, j)=(1/4)*(V(i+1, j)+V(i-1, j)+V(i, j+1)+V(i, j-1));
            % claculate the average V around V
        end
    end
end

subplot(2, 1, 1), mesh(X,Y,V'), colorbar;     % plot the mesh pic
subplot(2, 1, 2), contour(X,Y,V'), colorbar;     % plot the contour pic
% V' is for correction, beacause array claculate from rows to column
```

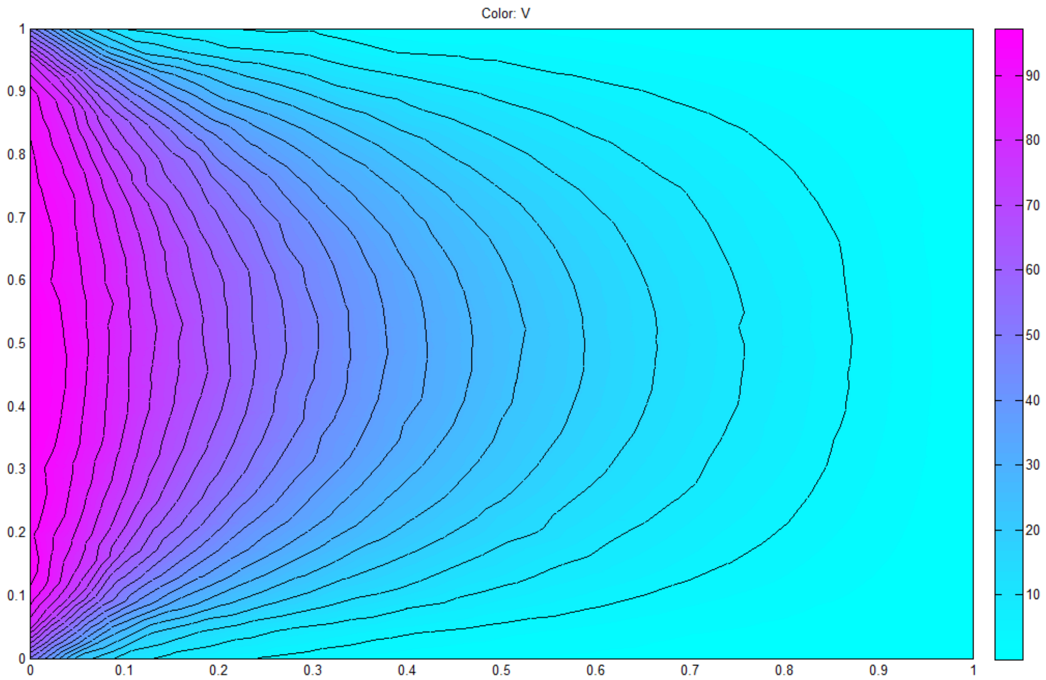
Result：

Mesh & Contour Picture





Use pdetool



Using pdetool toolbox of Matlab, the results is the same.

Improvement： Using matrix

Though the results are right, this program takes too much time. However, there is a better method：matrix.

For example, we can use the command **tic** & **toc** to time the program as said above about how much time we spend.

```
clear all; clc;
a = 5;
b = 2;
Vo = 100;
x = 0 : 0.02 : a;
y = 0 : 0.02 : b;
[X,Y] = meshgrid(x,y);
V(2:length(x), 1:length(y)) = 0;
V(1, 1:length(y)) = Vo;
tic    % time to count
for (k = 1 : 1 : 10000)
    for (i = 2 : 1 : length(x)-1)
        for (j = 2 : 1 : length(y)-1)
            V(i, j)=(1/4)*(V(i+1, j)+V(i-1, j)+V(i, j+1)+V(i, j-1));
        end
    end
end
toc    % end counting
```

Output：

```
Elapsed time is 37.339221 seconds.
```

If we use matrix, in the same condtions, the program will be more effectively.

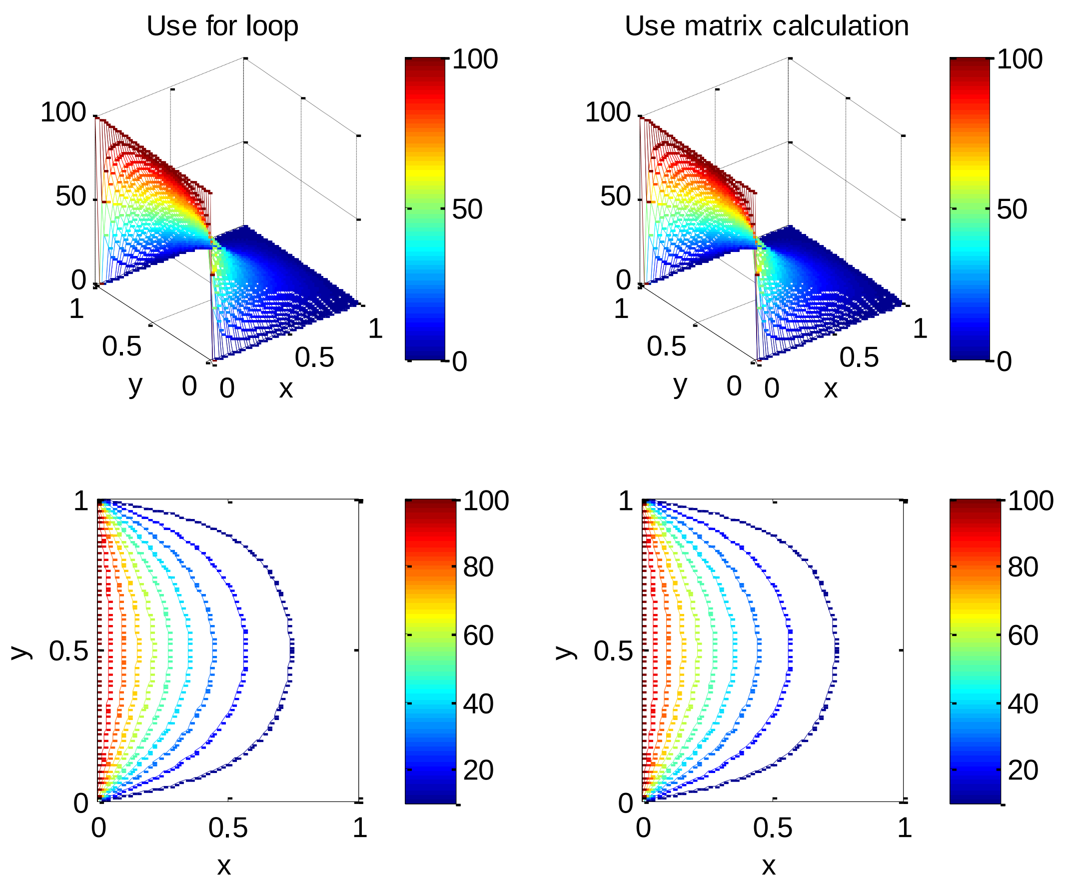
```
clear all; clc;
h = 0.02;
a = 5;
b = 2;
x = 0 : h : a;
y = 0 : h : b;
```

```
n_x = length(x);
n_y = length(y);
V = zeros(length(x), length(y));
% itialize all entry in this matrix to be zero
V(1, 1:length(y)) = 100;
% itialize boundary condition, x=1 -> V=Vo

% matrix calculaition
tic
for (n = 2 : 10000)
  V(2:n_x-1, 2:n_y-1)=(V(1:n_x-2,2:n_y-1)+ V(3:n_x, 2:n_y-1)+ V(2:n_x-1,
1:n_y-2)+V(2:n_x-1, 3:n_y))/4;
end

toc
```

Output：

```
Elapsed time is 5.709953 seconds.
```

According the two outputs, we know matrix calculation is much faster. However, the output figure are the same



Full program code：

```
clear all; clc;

a = input('Please input a:');     % input a
b = input('Please input b:');     % input b
Vo = 100;     % Vo = 100 at x=0

x = 0 : 0.02 : a;     % the interval of x is 0.02
y = 0 : 0.02 : b;     % the interval of y is 0.02

[X,Y] = meshgrid(x,y);     % make x and y line space a two dimention matrix

V(2:length(x), 1:length(y)) = 0;     % the initial value of x>2 is 0
V(1, 1:length(y)) = Vo;     % the initial value of x=1 is Vo

% original equation at p.182 eq.4-114
% V=4*Vo/pi*((sinh((n*pi*(a-X))/b))/(n*sinh(n*pi*a)/b))*sin(n*pi*y /b)

tic    % time to count

for (k = 1 : 1 : 10000)    % number of iteration
    for (i = 2 : 1 : length(x)-1)
    % from x = 2 to margin-1, beacause margin is 0
        for (j = 2 : 1 : length(y)-1)
    % from y = 2 to margin-1 for margin is 0
            V(i, j) = (1/4)*(V(i+1, j)+V(i-1, j)+V(i, j+1)+V(i, j-1));
            % claculate the average V around V
        end
    end
end

toc    % stop counting

subplot(2, 2, 1), mesh(X,Y,V'), xlabel('x'), ylabel('y'), colorbar;
title('Use for loop');
subplot(2, 2, 3), contour(X,Y,V'), xlabel('x'), ylabel('y'), colorbar;
```

```
% Use matrix to improve the program to be effectively

clear all;
h = 0.02;
a = input('\nPlease input a:');    % input a
b = input('Please input b:');    % input b

x = 0 : h : a;
y = 0 : h : b;
n_x = length(x);
n_y = length(y);
V = zeros(length(x), length(y));
% itialize all entry in this matrix to be zero
V(1, 1:length(y)) = 100;

tic   % time to count

for (n = 2 : 10000)
  V(2:n_x-1, 2:n_y-1) = (V(1:n_x-2, 2:n_y-1) + V(3:n_x, 2:n_y-1) +
V(2:n_x-1, 1:n_y-2) + V(2:n_x-1, 3:n_y)) / 4;
end

toc   % stop counting

[X Y] = meshgrid(x, y);
subplot(2, 2, 2), mesh(X,Y,V'), xlabel('x'), ylabel('y'), colorbar;
title('Use matrix calculation');
subplot(2, 2, 4), contour(X,Y,V'), xlabel('x'), ylabel('y'), colorbar
```

Another method：

There is still a method we can use. From our text p. 182 eq. 4 − 114, we know

$$V(x,y) = \frac{4V_o}{\pi}\sum_{n=odd}^{\infty}\frac{\sinh\left[\frac{n\pi(a-x)}{b}\right]}{n\sinh\left(\frac{n\pi a}{b}\right)}\sin\frac{n\pi}{b}y\,,n = 1,3,5,\ldots,a < x < a, 0 < y < b$$

So, we just simply key in the equation, and the result will be there. However, there is a problem with this equation. For there exists sinh which may diverge so quickly in sigma, n must not be too large.

The Matlab code：

```
clear; clc;

a = input('Please input a:');     % input a
b = input('Please input b:');     % input b

Vo = 100;
x = 0.02 : 0.02 : a-0.02;
y = 0.02 : 0.02 : b-0.02;
[X Y] = meshgrid(x, y);
Va = 0;

for (m = 1 : 10)
    n = (2*m-1);
    Va = Va +
4*Vo/pi.*sin(Y.*n*pi./b).*sinh((n*pi.*(a-X)./b))./(n.*sinh(n*pi.*a./b
));
    % V=4*Vo/pi*((sinh((n*pi*(a-X))/b))/(n*sinh(n*pi*a)/b))*sin(n*pi*y
/b)
end

x = 0 : 0.02 : a;
y = 0 : 0.02 : b;
nx = length(x);ny=length(y);
V = zeros(ny,nx);
V(2 : ny-1, 2 : nx-1) = Va;
```

```
V(1 : ny, 1) = 100;

[X Y] = meshgrid(x, y);
subplot(2, 1, 1), mesh (X, Y, V), xlabel('X'), ylabel('Y');
subplot(2, 1, 2), contour(X, Y, V), xlabel('X'),ylabel('Y');
colorbar
```

We let $a = 1$ and $b = 1$, the following is the resulting figure