EE6506 - Computational Electromagnetics
Spring 2016
Lectures-1, 2&3
Basics of differencing

Ananth Krishnan, *Electrical Engineering, IIT Madras*

February 3, 2016

Space for handwritten notes ↓

Learning Objectives

1. To derive finite difference approximations of first and second order derivative of a uniformly sampled function
2. To derive the difference approximations of partial derivatives

Finite difference approximations of first order derivatives or slope of a function

Types of differences - Partial list

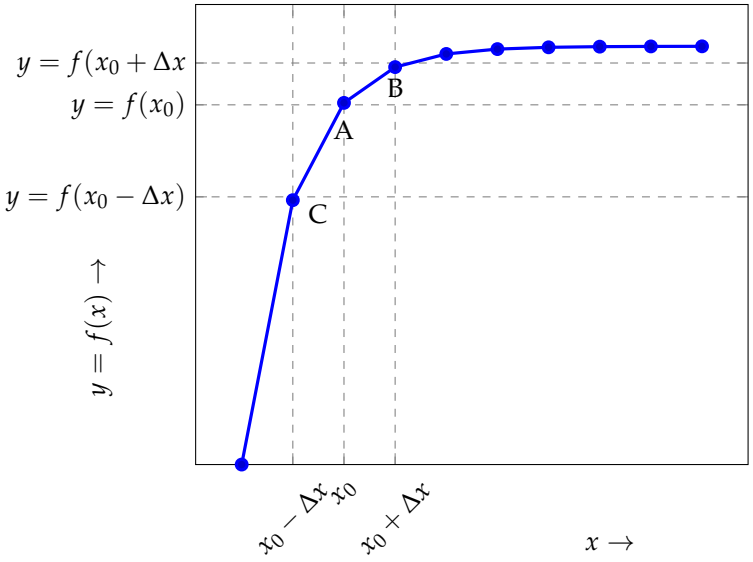

Figure 1: plot of a function y=f(x)

1. We would like to find the slope of the function $f(x)$ at point A. The values of x are uniformly distributed from 0 to 8 in steps of 1.
2. Let the coordinates of point A be (x_0, y_0).
3. The mathematical definition of slope of the curve is

$\boxed{f'(x_0) = \lim_{x\to x_0} \frac{f(x)-f(x_0)}{x-x_0}}$ or $\boxed{f'(x_0) = \lim_{x\to x_0} \frac{f(x_0)-f(x)}{x_0-x}}$ by multiplying the numerator and denominator by -1.

4. The mathematical definition is not computer friendly, since the value of $f(x)$ should be defined at all x, which is typically not the case with data stored on a computer. In a discrete data system (as in the case here), an approximate numerical definition is more appropriate.

5. The numerical definition of derivative (approximate) in discrete dataset is
$\boxed{f'(x_0) \approx \frac{f(x_0+\Delta x)-f(x_0)}{\Delta x}}$ or $\boxed{f'(x_0) \approx \frac{f(x_0)-f(x_0-\Delta x)}{\Delta x}}$

6. The numerical definition is computer friendly, slope at point A can be approximated to slopes of lines AB or AC depending on our choice of the definition.

7. The slope of line AB is known as Forward Difference and the slope of line AC is known as backward difference.

8. There is also another way of writing the slope at point A. The slope could be written as slope of line joining points B and C.
$\boxed{f'(x_0) \approx \frac{f(x_0+\Delta x)-f(x_0-\Delta x)}{2\Delta x}}$
This approximation of slope is called Central Difference.

9. We notice that as $\Delta x \to 0$, all of the above differences will give $f'(x_0)$. But in a discrete data set, which expression then corresponds to the more accurate slope at point A? How much is the error in this numerical approximation compared to mathematical definition of a derivative?

Estimation of errors in Forward Difference

1. Let us look at the Taylor's series expansion of $f(x_0+\Delta x)$

$$f(x_0+\Delta x) = f(x_0) + \frac{\Delta x}{1!}f'(x_0) + \frac{\Delta x^2}{2!}f''(x_0) + \frac{\Delta x^3}{3!}f'''(x_0) + \dots \tag{1}$$

Rearranging the terms,

$$\frac{\Delta x}{1!}f'(x_0) = f(x_0+\Delta x) - f(x_0) - \frac{\Delta x^2}{2!}f''(x_0) - \frac{\Delta x^3}{3!}f'''(x_0) + \dots \tag{2}$$

$$\implies f'(x_0) = \frac{f(x_0+\Delta x) - f(x_0) - \frac{\Delta x^2}{2!}f''(x_0) - \frac{\Delta x^3}{3!}f'''(x_0) + \dots}{\Delta x} \tag{3}$$

$$f'(x_0) \approx \frac{f(x_0+\Delta x) - f(x_0)}{\Delta x} \tag{4}$$

For small values of Δx, the maximum error resulting from the Taylor's series truncation is $\approx \frac{\frac{\Delta x^2}{2!}}{\Delta x} \approx \boxed{\frac{\Delta x}{2}}$

Estimation of errors in Backward Difference

1. Let us look at the Taylor's series expansion of $f(x_0 - \Delta x)$

$$f(x_0 - \Delta x) = f(x_0) - \frac{\Delta x}{1!} f'(x_0) + \frac{\Delta x^2}{2!} f''(x_0) - \frac{\Delta x^3}{3!} f'''(x_0) + \dots \tag{5}$$

Rearranging the terms,

$$\frac{\Delta x}{1!} f'(x_0) = f(x_0) - f(x_0 - \Delta x) + \frac{\Delta x^2}{2!} f''(x_0) - \frac{\Delta x^3}{3!} f'''(x_0) + \dots \tag{6}$$

$$\implies f'(x_0) = \frac{f(x_0) - f(x_0 - \Delta x) + \frac{\Delta x^2}{2!} f''(x_0) - \frac{\Delta x^3}{3!} f'''(x_0) + \dots}{\Delta x} \tag{7}$$

$$f'(x_0) \approx \frac{f(x_0) - f(x_0 - \Delta x)}{\Delta x} \tag{8}$$

For small values of Δx, the maximum error resulting from the Taylor's series truncation is $\approx \frac{\frac{\Delta x^2}{2!}}{\Delta x} \approx \boxed{\frac{\Delta x}{2}}$

Estimation of errors in Central Difference

1. Let us look at both the Taylor's series expansions

$$f(x_0 + \Delta x) = f(x_0) + \frac{\Delta x}{1!} f'(x_0) + \frac{\Delta x^2}{2!} f''(x_0) + \frac{\Delta x^3}{3!} f'''(x_0) + \dots \tag{9}$$

$$f(x_0 - \Delta x) = f(x_0) - \frac{\Delta x}{1!} f'(x_0) + \frac{\Delta x^2}{2!} f''(x_0) - \frac{\Delta x^3}{3!} f'''(x_0) + \dots \tag{10}$$

(1)-(2) gives,
$f(x_0 + \Delta x) - f(x_0 - \Delta x) = \frac{2\Delta x}{1!} f'(x_0) + 2\frac{\Delta x^3}{3!} f'''(x_0) + \dots$

$$\implies f'(x_0) = \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x) - 2\frac{\Delta x^3}{3!} f'''(x_0) + \dots}{2\Delta x} \tag{11}$$

$$f'(x_0) \approx \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x} \tag{12}$$

For small values of Δx, the maximum error resulting from the Taylor's series truncation is $\approx \frac{\frac{2\Delta x^3}{3!}}{2\Delta x} \approx \boxed{\frac{\Delta x^2}{6}}$

2. This method involved finding the derivative or slope of the curve at point A using the slope of the line BC. The errors obtained by truncating the Taylor's series to second term was significantly lower than the forward and backward differences, corresponding to the slopes of lines AB or AC respectively.

Method	Description	Error
Forward Differencing	Slope of AB	$\approx \frac{\Delta x}{2}$
Backward Differencing	Slope of AC	$\approx \frac{\Delta x}{2}$
Central Differencing	Slope of BC	$\approx \frac{\Delta x^2}{6}$

Table 1: Summary of errors in differencing

Numerical form of second order derivatives

Homogeneous second order derivatives

Let us start with the Taylor's series expansions once again

$$f(x_0+\Delta x) = f(x_0) + \frac{\Delta x}{1!}f'(x_0) + \frac{\Delta x^2}{2!}f''(x_0) + \frac{\Delta x^3}{3!}f'''(x_0) + \dots \tag{13}$$

$$f(x_0-\Delta x) = f(x_0) - \frac{\Delta x}{1!}f'(x_0) + \frac{\Delta x^2}{2!}f''(x_0) - \frac{\Delta x^3}{3!}f'''(x_0) + \dots \tag{14}$$

(3)+(4) gives,

$f(x_0+\Delta x) - f(x_0-\Delta x) = 2f(x_0) + \frac{2\Delta x^2}{2!}f''(x_0) + \dots$

$$\implies f''(x_0) = \frac{f(x_0+\Delta x) - f(x_0-\Delta x) - 2f(x_0) - \dots}{\frac{2\Delta x^2}{2!}} \tag{15}$$

$$f''(x_0) \approx \frac{f(x_0+\Delta x) - f(x_0-\Delta x) - 2f(x_0)}{\Delta x^2} \tag{16}$$

For small values of Δx, the maximum error resulting from the Taylor's series truncation is $\approx \frac{\frac{2\Delta x^4}{4!}}{\frac{2\Delta x^2}{2!}} \approx \boxed{\frac{\Delta x^2}{12}}$

Partial second order derivatives

1. Suppose a function $f(x,y)$ is a function of two independent variables x and y, then the second order partial derivatives in numerical representation are

$$\frac{\partial^2 f}{\partial x^2} \approx \frac{f(x_0+\Delta x, y_0) + f(x_0-\Delta x, y_0) - 2f(x_0,y_0)}{\Delta x^2} \tag{17}$$

$$\frac{\partial^2 f}{\partial y^2} \approx \frac{f(x_0, y_0+\Delta y) + f(x_0, y_0-\Delta y) - 2f(x_0,y_0)}{\Delta y^2} \tag{18}$$

Numerical form of Laplace's equation

Using the numerical representation of derivatives discussed above, let us rewrite Laplace's equations in a computer friendly manner. Let us begin with the Laplace's equation for electrostatics

$$\nabla^2\phi = 0 \tag{19}$$

Rewriting this equation in the Cartesian Coordinate system,

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0 \tag{20}$$

where ϕ is the scalar potential.

Rewriting the partial derivatives in the difference form, we get,

$$\frac{\partial^2\phi}{\partial x^2} \approx \frac{\phi(x_0+\Delta x, y_0) + \phi(x_0-\Delta x, y_0) - 2\phi(x_0, y_0)}{\Delta x^2} \tag{21}$$

$$\frac{\partial^2\phi}{\partial y^2} \approx \frac{\phi(x_0, y_0+\Delta y) + \phi(x_0, y_0-\Delta y) - 2\phi(x_0, y_0)}{\Delta y^2} \tag{22}$$

$$\implies \frac{\phi(x_0+\Delta x, y_0) + \phi(x_0-\Delta x, y_0) - 2\phi(x_0, y_0)}{\Delta x^2} + \tag{23}$$

$$\frac{\phi(x_0, y_0+\Delta y) + \phi(x_0, y_0-\Delta y) - 2\phi(x_0, y_0)}{\Delta y^2} \approx 0 \tag{24}$$

Assuming $\Delta x = \Delta y = \Delta h$, we can get rid of the denominators. This would mean that in our discrete data system the x and y data are spaced evenly with same step size.We now re-write our equation as

$$\begin{aligned}&\phi(x_0+\Delta x, y_0) + \phi(x_0-\Delta x, y_0) - 2\phi(x_0, y_0) + \\ &\phi(x_0, y_0+\Delta y) + \phi(x_0, y_0-\Delta y) - 2\phi(x_0, y_0) \approx 0\end{aligned} \tag{25}$$

This is the difference form of Laplace's equation.

Meaning of Laplace's equation explained through differencing

1. The objective of Laplace's equation is to find the potential distribution in a given region, with specified boundary conditions [1].
2. In order to solve for the potentials, we use the difference form of Laplace's equation and rearrange the terms to calculate the potential at each point.

[1] We will study the meaning of boundary conditions later

$$\begin{aligned}&\phi(x_0+\Delta x, y_0) + \phi(x_0-\Delta x, y_0) - 2\phi(x_0, y_0) + \\ &\phi(x_0, y_0+\Delta y) + \phi(x_0, y_0-\Delta y) - 2\phi(x_0, y_0) \approx 0\end{aligned}$$

$$\implies \phi(x_0+\Delta x, y_0)+\phi(x_0-\Delta x, y_0)+$$
$$\phi(x_0, y_0+\Delta y)+\phi(x_0, y_0-\Delta y) \approx 4\phi(x_0, y_0)$$

$$\boxed{\phi(x_0, y_0) \approx \frac{\phi(x_0+\Delta x, y_0)+\phi(x_0-\Delta x, y_0)+\phi(x_0, y_0+\Delta y)+\phi(x_0, y_0-\Delta y)}{4}}$$

NOTE :- A review of uniqueness theorem may be appropriate here.

3. The potential at any point ϕ_0 inside a domain can be approximated to the average potential of the 4 nearest neighboring points $\phi_1, \phi_2, \phi_3 and \phi_4$. Hence

$$\boxed{\phi_0 \approx \frac{\phi_1+\phi_2+\phi_3+\phi_4}{4}}$$

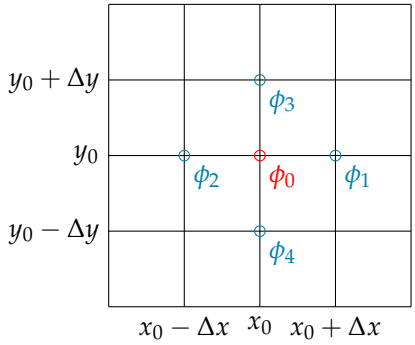

Figure 2: Meaning of Laplace's equation

4. Incidentally, this is normally what we do out of intuition. When we know the potentials of points close by, we take the average of the potentials at known points to approximately calculate the unknown potential in the middle.

Numerical solution to Laplace's equation

1. The objective of solving Laplace's equation using a computer is to find the potential distribution in the region of interest which we shall refer to as Simulation Domain.

2. In other words, there are certain conditions imposed on the periphery of the simulation domain. What would be the potential distribution in the simulation domain?

3. To understand this clearly, let us consider the following example.

4. The top plate is supplied with 10 V voltage and the three metal plates on the sides are connected to the ground (0 V).

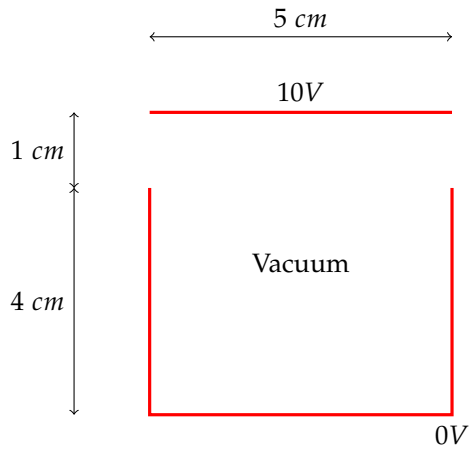

Figure 3: Example 1

5. The question is to find the potential distribution in the 5cm x 5cm region.

6. Solution :-

   (a) Divide the simulation domain into a number of squares as shown below.

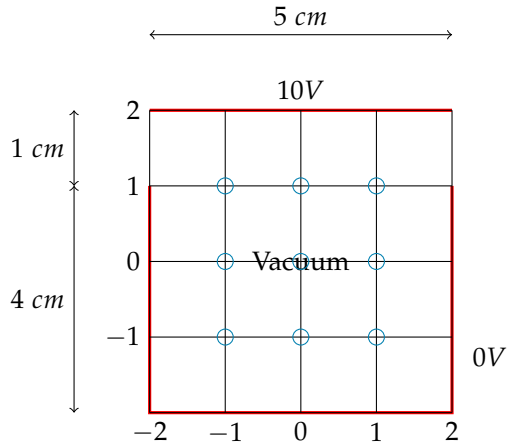

Figure 4: Divide the simulation domain

   (b) Identify the points where the potentials need to be calculated. These are the points marked in blue.

   (c) Assume the potential at all these points to be Zero.

   (d) Start with unknown voltage at Top-left and calculate it as average of its 4 nearest neighbors. The value of the voltage at the point (-1,1) is calculated as $\frac{10+0+0+0}{4} = 2.5V$.

   (e) Now proceed to the point (0,1) and find the potential at that point as $\frac{10+2.5+0+0}{4} = 3.125V$.

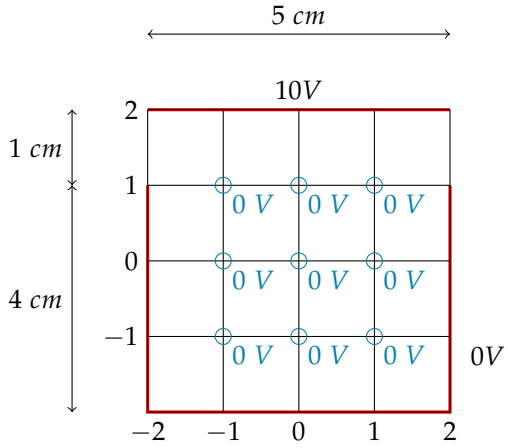

Figure 5: Assume unknown potentials as 0 V

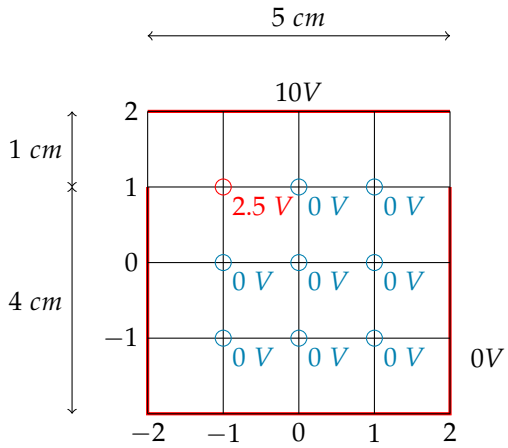

Figure 6: Calculate the potential using nearest neighbors

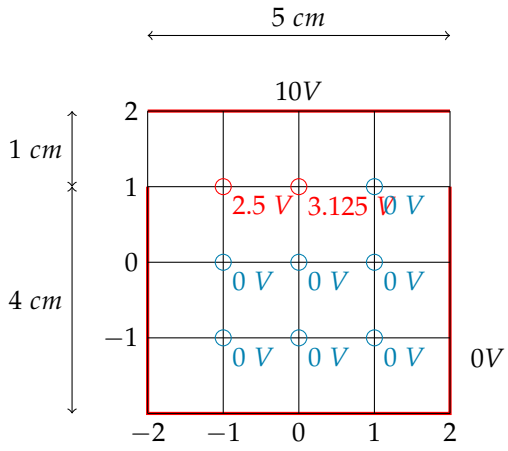

Figure 7: Continue the calculation

(f) After completing the calculation for all unknown potentials, we realize that, for every point where we calculated the potential, the values of the neighboring potentials have changed.

(g) So we once again calculate the potentials at all points using the average of nearest neighbors.

(h) Each time, when we calculate the unknown potentials of all the points, we say that we have complete an Iteration.

(i) The procedure is continued until between successive iterations, the calculated potentials do not differ by more than a threshold value that is decided by the user.

Algorithm

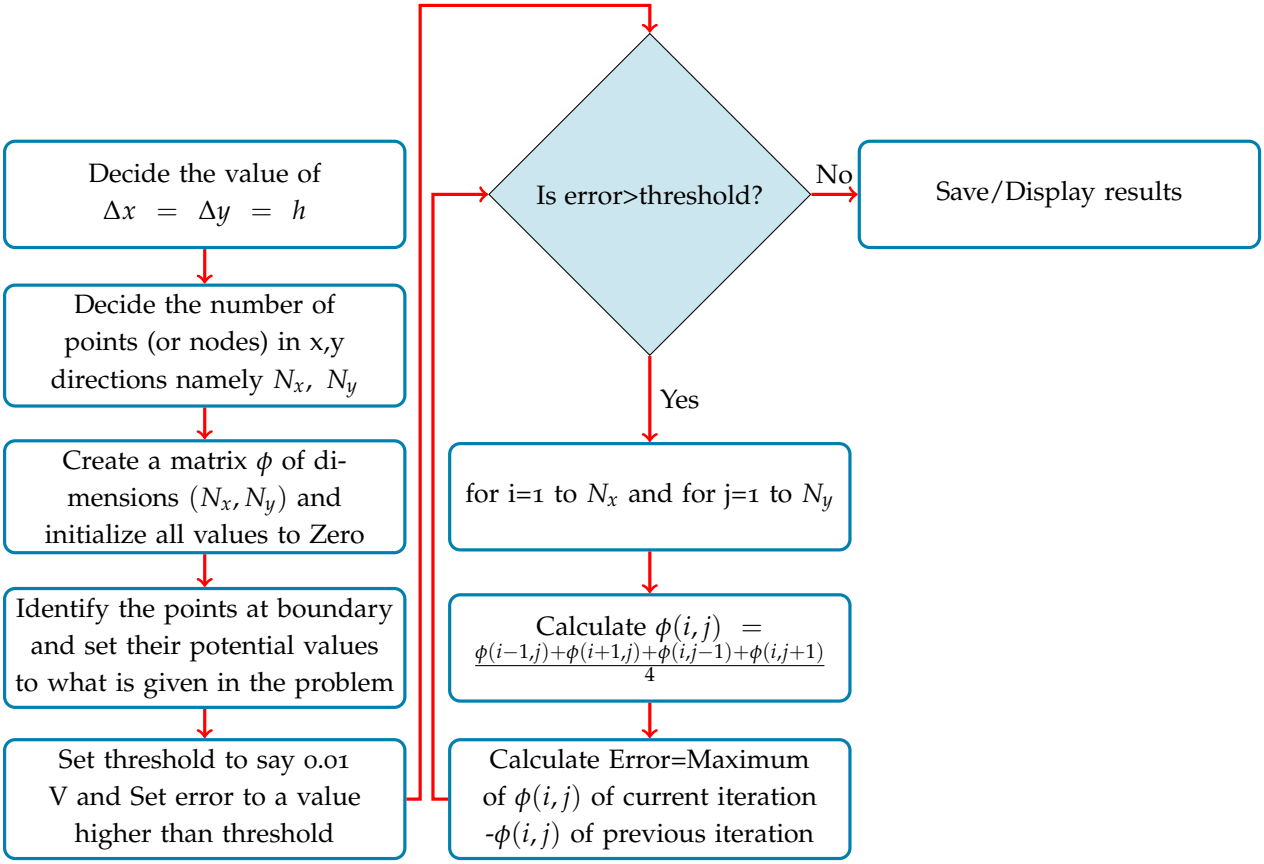

Figure 8: Flowchart for finite difference solution to Laplace's equation

1. The Matlab code for a similar problem is shown below. The problem description can be found in the program comments.
2. The program allows one to view the potential distribution as iterations proceed.
3. At the end, the electric field distribution is calculated as a gradient of the finally calculated potential.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Written for Course :- Computational Electromagnetics, Fall 2011
%                       Department of Electrical Engineering
%                       Indian Institute of Technology Madras
%                       Chennai - 600036, India
%
% Authors            :- Sathya Swaroop Ganta, B.Tech., M.Tech. Electrical Engg.
%                       Kayatri, M.S. Engineering Design
%                       Pankaj, M.S. Electrical Engg.
%                       Sumanthra Chaudhuri, M.S. Electrical Engg.
%                       Projesh Basu, M.S. Electrical Engg.
%                       Nikhil Kumar CS, M.S. Electrical Engg.
%
% Instructor :- Ananth Krishnan
%               Assistant Professor
%               Department of Electrical Engineering
%               Indian Institute of Technology Madras
%
% Any correspondance regarding this program may be addressed to
% Prof. Ananth Krishnan at 'computational.em.at.iit.madras@gmail.com'
%
% Copyright/Licensing :- For educational and research purposes only. No
% part of this program may be used for any financial benefit of any kind
% without the consent of the instructor of the course.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% "Finite Difference Method (FDM) solution to Laplacian"
%
% Objective of the program is to solve for the steady state voltage
% distribution in a region 0<x<30, 0<y<30, given that one of the sides of
% square is excited with a voltage of 10 Volts and all other
% sides are maintained at 0 Volts.
% At any iteration, the value of voltage is updated as average of voltages
% of 4 nearest neighbors, until between consecutive iterations, the error
% is less than 0.01 V.
%
% The tolerance in error between iterations is kept at 0.01 V. This may be
% tweaked to a higher or lower value for lower or higher accuracy
% respectively. Imagesc command by default uses image axis settings, which
% are different from normal plot command and hence x and y axis may look
% flipped. Read Matlab documentation on imagesc for more details.
%
% Program stops when iteration number in plot does not change or can be
```

```
% closed anytime by just closing the plot window. On normal completion, the
% program plots the electric field in a quiver plot.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Clearing variables in memory and Matlab command screen
clear all;
clc;

%Dimensions of the simulation grid in x (xdim) and y (ydim) directions
xdim=30;
ydim=30;

%Initializing previous (V_prev) and present (V_now) voltage matrices
V_now=zeros(xdim+1,ydim+1);
V_prev=zeros(xdim+1,ydim+1);

%Initializing boundary conditions only for V_now
i=1:1:xdim+1;%x-co-ordinates for boundary at y=ydim*grid_size

%A voltage of 10V is applied on one boundary, the remaining
%boundaries are going to remain at zero volts
V_now(i,ydim+1)=10;

%Iteration counter
iter=0;

%Calculation of maximum error between V_now and V_prev at all points
%By setting the applied voltage for only V_now, we have made V_no and
%V_prev different, hence error will be greater than zero and the program
%will enter the while loop following this command.
error=max(max(abs(V_now-V_prev)));
figure;pause;
%Iteration loop
while(error>0.01)%Run this until convergence

    iter=iter+1; % Iteration counter increment

    % Updating present iteration using 4 point Central diffrence form
    % of Laplace equation obtained using Finite Difference method
    for i=2:1:xdim
        for j=2:1:ydim
            V_now(i,j)=(V_now(i-1,j)+V_now(i+1,j)+V_now(i,j-1)+V_now(i,j+1))/4;
        end
    end
```

```
88      error=max(max(abs(V_now-V_prev))); % Calculate the maximum error between previous and
            current iteration at all points
89      V_prev=V_now; % Updating previous iteration matrix to the last iteration performed
90
91      %Movie type colour scaled image plot to see how solution progresses
92      contour(V_now',30);colorbar;colormap('hsv');
93      title(['Voltage distribution on a ',int2str(xdim),' x ',int2str(ydim),' grid at
            iteration no ',int2str(iter)],'Color','k');
94      getframe;
95  end
96
97  %Plot the electric field distribution
98  hold('on');
99  [ex,ey]=gradient(V_now',1,1);
100 sz = sqrt(ex.^2 + ey.^2);
101 exx=ex./sz;
102 eyy=ey./sz;
103 quiver(-exx,-eyy,0.5,'color',[0 0 0]); %Quiver command creates a plot, E=-grad(V), hence
        the negative sign
104
105 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
106 % END OF PROGRAM
107 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

4. The following figure shows the calculated potential

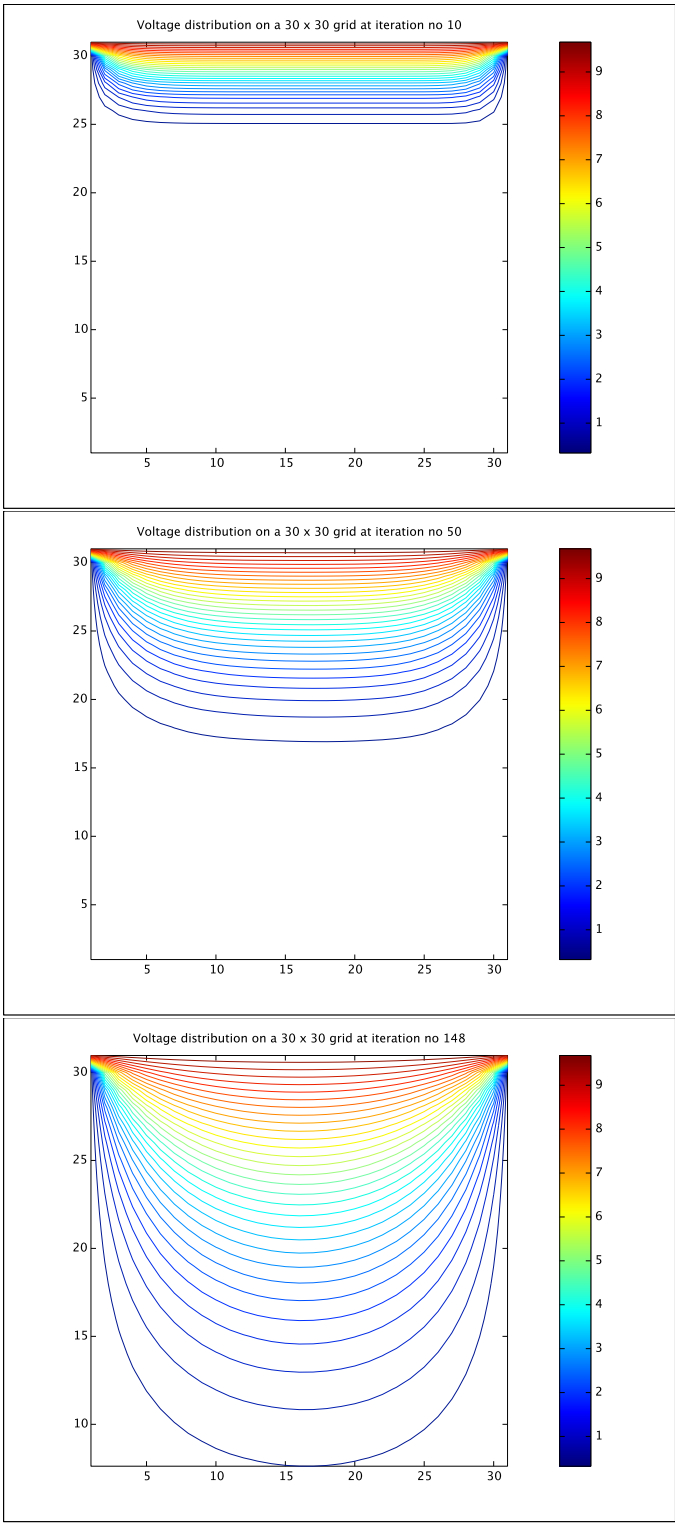

Figure 9: Equipotential Lines at $10^{th}, 50^{th}, 148^{th}$ iterations

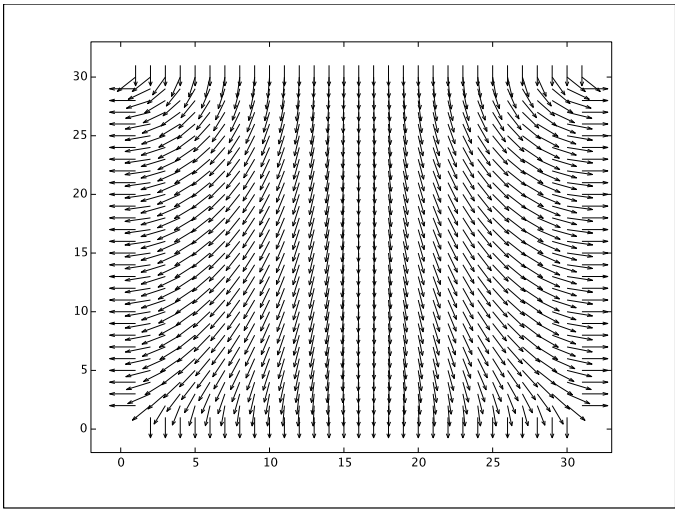

Figure 10: Calculated Electric fields