

Program 20 / Chapter 25

Electric potential

Just as the electric field gives the force on a test charge, the electric potential gives the potential energy on a test charge. The program in this section graphs a given potential and its associated electric field.

Theory Given the electric potential, $V(x,y)$, the electric field is

$$\mathbf{E}(x,y) = -\frac{\partial V}{\partial x} \mathbf{i} - \frac{\partial V}{\partial y} \mathbf{j}$$

(see Section 25–4 in your textbook). The magnitude of the electric field is largest where the derivatives of the potential are maximum. It is often useful to look at contours of constant potential since the electric field points in the direction of the gradient, that is, if the contours were to form a topological map of a mountainous terrain then the electric field at a location would point in the “downhill” direction (see Section 25–3). Hopefully the graphs produced by the program in this section will make this point clear.

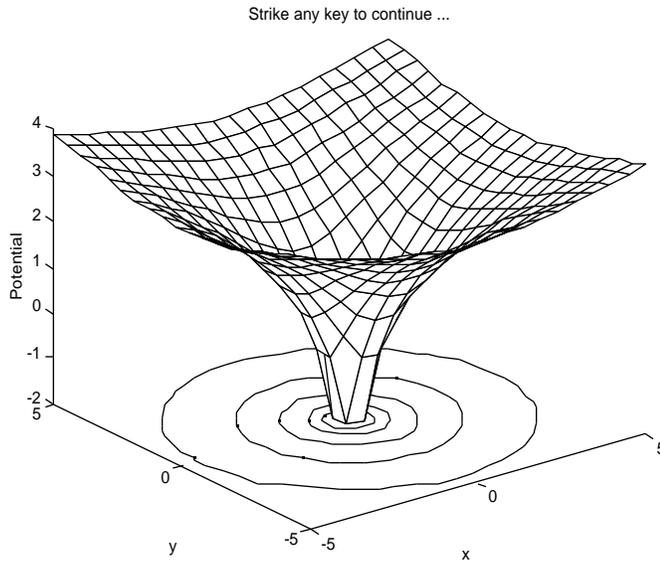
Program The MATLAB program `e_and_v`, which computes the electric field $\mathbf{E}(x,y)$ for a given potential $V(x,y)$ and graphs both in the xy plane, is outlined below:

- Initialize variables (e.g., potential $V(x,y)$, graphics).
- Evaluate electric field as $E_x = -\partial V/\partial x$ and $E_y = -\partial V/\partial y$.
- Loop over all grid points and evaluate $V(x,y)$ and $\mathbf{E}(x,y)$ on grid.
 - Compute potential at the grid point.
 - Compute components of the electric field.
 - Normalize E-field vectors to unit length.
- Plot contours of constant electric potential.
- Add electric field direction to potential contour plot.

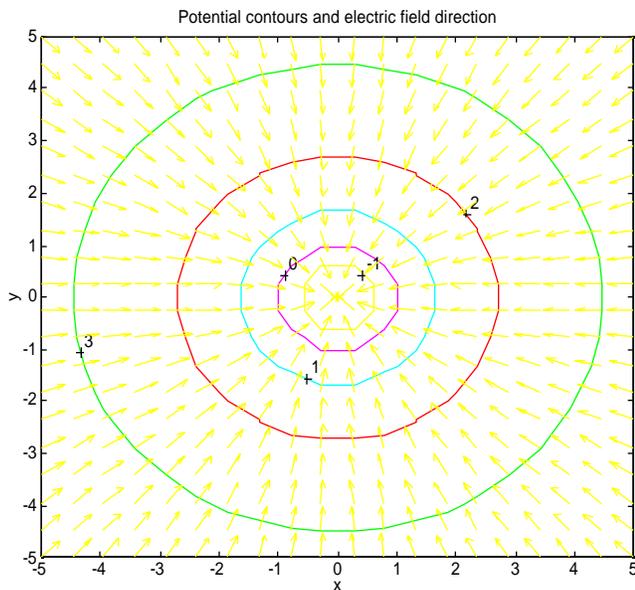
MATLAB is used primarily for numerical calculations and graphics, yet the Student Edition includes symbolic manipulation routines. The x and y components of the electric field are computed symbolically by evaluating the x and y derivatives of the potential using the `diff` function and multiplying the result by (-1) using the `symop` function.

The `e_and_v` program uses several of MATLAB’s advanced graphics routines. The `meshc` function produces a wiremesh and contour plot of a two-dimensional surface. The `contour` function produces just a contour plot; the `clabel` function adds labels to the contour levels. Finally, the `e_and_v` program uses the `quiver` function draws a grid of arrows to indicate the direction of the electric field lines (using `hold` to overlay the quiver plot of electric field direction on the contour plot of potential).

For the potential $V(x,y) = \log(x^2+y^2)$, the program produces the contour and wiremesh plots shown below.



In the graphic above, the wiremesh plot shows this potential surface to be shaped like a wine glass or flower vase with a deep (large negative) center. The contour lines (shown on the xy plane below the wiremesh) are closest together in the center where the gradients are the steepest.



In the plot shown above, notice how the electric field at a location points in the “downhill” direction, as given by the contours of the potential. The program draws all the arrows the same length instead of having the length be proportional to the magnitude of the electric field since doing the latter results in most arrows being too small to see.

Exercises

1. (I) Run the `e_and_v` program and print out the graphs it produces for the potentials listed below.

- | | |
|------------------------------|---------------------------------|
| (a) $V(x,y) = x$ | (b) $V(x,y) = x + y$ |
| (c) $V(x,y) = x^2 + y^2$ | (d) $V(x,y) = 1/(x^2 + y^2)$ |
| (e) $V(x,y) = x/(x^2 + y^2)$ | (f) $V(x,y) = 1/\sqrt{1 + x^2}$ |
| (g) $V(x,y) = \log(x^2)$ | (h) $V(x,y) = \sin(x) \sin(y)$ |

2. (I) For the potentials given in exercise 1, find $\mathbf{E}(x,y)$ by evaluating the x and y derivatives of the potential via pencil and paper calculation. Check your work using the `e_and_v` program's output.

3. (II) Modify the `e_and_v` program to graph the contours of the x and y components of the electric field. Run your program and print the resulting plots for the cases listed in exercise 1.

4. (II) Modify the `e_and_v` program to accept as input $V(r)$ where $r = \sqrt{x^2 + y^2}$. Run your program and print out the resulting graphs for: (a) $V(r) = 1/r$; (b) $V(r) = 1/r^2$; (c) $V(r) = -\ln(r)$ and (d) $V(r) = 75 - r^2$.

5. (III) Modify the `e_and_v` program to accept potentials that are defined piecewise in the form,

$$V(r) = \begin{cases} V_1(r) & r < R \\ V_2(r) & r \geq R \end{cases}$$

Test your program using the following potentials (take $R = 1$),

$$(a) V(r) = \begin{cases} 1/R & r < R \\ 1/r & r \geq R \end{cases} \quad ; \quad (b) V(r) = \begin{cases} 3 - r^2/R^2 & r < R \\ 2/r & r \geq R \end{cases}$$

6. (III) Modify the `e_and_v` program to compute the potential $V(x,y)$ (to within a constant of integration) given the electric field $\mathbf{E}(x,y)$. Check your program using the cases given in exercise 1.

Listing

`e_and_v.m`

```
% e_and_v - Compute electric field from potential
%           and graph potential contours and E-field direction
clear all; help e_and_v; % Clear memory; print header

%@ Initialize variables (e.g., potential V(x,y), graphics)
fprintf('Enter potential V(x,y) as an equation \n');
fprintf('For example: log(x^2 + y^2) \n');
V = input(' : ','s'); % Read in V(x,y) as text string
NGrid = 20; % Number of grid points for plots
xMax = 5; % Values plotted from x= -xMax to x= xMax
yMax = xMax; % Values plotted from y= -yMax to y= yMax
```

```

for i=1:NGrid
    xPlot(i) = -xMax + (i-1)/(NGrid-1)*(2*xMax); % x values to plot
    yPlot(i) = -yMax + (i-1)/(NGrid-1)*(2*yMax); % y values to plot
end

%@ Evaluate electric field as  $E_x = (-1)*dV/dx$  and  $E_y = (-1)*dV/dy$ 
% Note use of symop command to perform symbolic multiplication by -1
Ex = symop( '-1', '*', diff(V,'x') );
Ey = symop( '-1', '*', diff(V,'y') );
fprintf('Electric field components are \n');
disp(['x component : ', Ex]);
disp(['y component : ', Ey]);

%@ Loop over all grid points and evaluate V(x,y) and E(x,y) on grid
for i=1:NGrid
    y = yPlot(i);
    for j=1:NGrid
        x = xPlot(j);

        %@ Compute potential at the grid point
        VPlot(i,j) = eval( V ); % Potential V(x,y)
        %@ Compute components of the electric field
        ExPlot(i,j) = eval( Ex );
        EyPlot(i,j) = eval( Ey );

        %@ Normalize E-field vectors to unit length
        MagnitudeE = sqrt( ExPlot(i,j)^2 + EyPlot(i,j)^2 );
        ExPlot(i,j) = ExPlot(i,j)/MagnitudeE;
        EyPlot(i,j) = EyPlot(i,j)/MagnitudeE;

    end
end

%@ Plot contours of constant electric potential
clf; figure(gcf); % Clear figure; bring figure window forward
meshc(xPlot,yPlot,VPlot); % Plot potential in contour/mesh plot
xlabel('x'); ylabel('y'); zlabel('Potential');
title('Strike any key to continue ...');
pause;
% Specify contour levels used in contour plot
axis([-xMax xMax -yMax yMax]); % Fix the min,max for x,y axes
cs = contour(xPlot,yPlot,VPlot); % Draw contour plot
clabel(cs); % Place contour labels on contour levels
%@ Add electric field direction to potential contour plot
hold on;
quiver(xPlot,yPlot,ExPlot,EyPlot); % Draw arrows for E field
title('Potential contours and electric field direction');
xlabel('x'); ylabel('y');
hold off;

```