

Numerical Solution to an Electric Potential Problem Using Mathematica's NDSolve Function

Adapted from <https://mathematica.stackexchange.com/questions/59441/solve-laplace-equation-using-ndsolve>

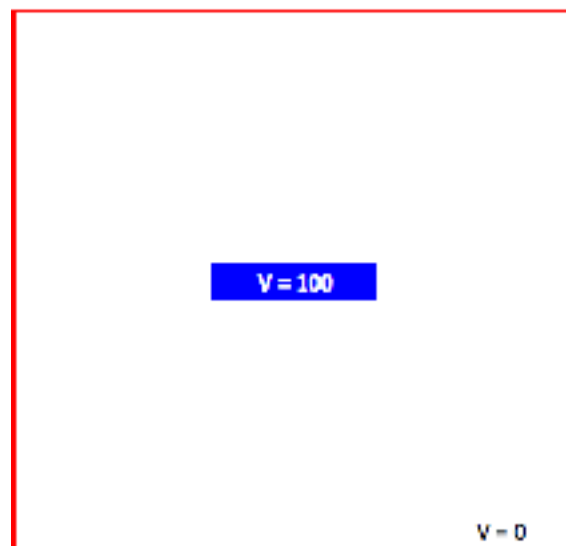
(The solution shown below is based on one by Sjoerd C. deVries.

PROBLEM

Consider a solid rectangular rod electrode of infinite length oriented along the z -axis. A second hollow electrode (with square cross-section), also of infinite length symmetrically surrounds the solid electrode and is oriented along the z -axis.

The inner electrode is at a potential $V = 100$ and the outer electrode is grounded.

This figure shows the 2D representation of the problem; obviously it is a 2D problem because infinite lengths rule out any z -dependence in the potential. Note: Cartesian Coordinates are the natural coordinate system to use here.



(a) First, find numerically the electric potential in the space between the electrodes. I will guide you through this exercise.

The BCs as given are called Fixed BCs or Dirichlet BCs. NDSolve can take BCs as an argument; these are inserted via M 's DirichletCondition function. First we define a region called Ω , where we want to find the potential. I will assume the outer boundary of Ω is 100 units on a side. M defines rectangles by the position (x and y coordinates) of the lower left

corner and the position of the upper right corner. Here `Rectangle[{0,0}, {100,100}]`. The inner boundary of Ω is also a rectangle. I chose the inner boundary to be `Rectangle[{40,48}, {60,52}]`. We want the potential *between* the rectangles. This region is specified by the `RegionDifference[]` command—the big rectangle minus the small one.

In[1]:= (* Study and execute the code below *)

```
ClearAll["`*"]
Ω = RegionDifference[Rectangle[{0, 0}, {100, 100}], Rectangle[{40, 48}, {60, 52}]];
sol = NDSolve[{D[V[x, y], x, x] + D[V[x, y], y, y] == 0,
  DirichletCondition[V[x, y] == 100., x == 40 && 48 ≤ y ≤ 52 ||
  x == 60 && 48 ≤ y ≤ 52 || 40 ≤ x ≤ 60 && y == 48 || 40 ≤ x ≤ 60 && y == 52],
  V[x, 0] == V[x, 100] == V[0, y] == V[100, y] == 0}, V, {x, y} ∈ Ω]
```

Out[3]= $\left\{ \left\{ V \rightarrow \text{InterpolatingFunction} \left[\begin{array}{l} \text{Domain: } \{\{0., 100.\}, \{0., 100.\}\} \\ \text{Output: scalar} \end{array} \right] \right\} \right\}$

The output of M's Numerical Solving routines is a construct that you can treat like a function. To look at Wolfram's description run this cell; click on the >> at the end to get the full Monty.

In[4]:= (* For information on InterpolatingFunctions, execute the code below *)

? InterpolatingFunction

InterpolatingFunction[domain, table] represents
an approximate function whose values are found by interpolation. >>

After extracting our InterprelatingFunction (I call it `VV[x,y].`), we can ask for it's value anywhere it is defined. Note what happens when we go outside Ω .

In[5]:= (* Execute code below *)

```
VV[x_, y_] = V[x, y] /. sol;
(* This code extracts the solution from the InterpolatingFunction*)
VV[25, 25]
VV[300, 300]
```

Out[6]= {21.2786}

 **InterpolatingFunction:** Input value {300., 300.} lies outside the range of data in the interpolating function.

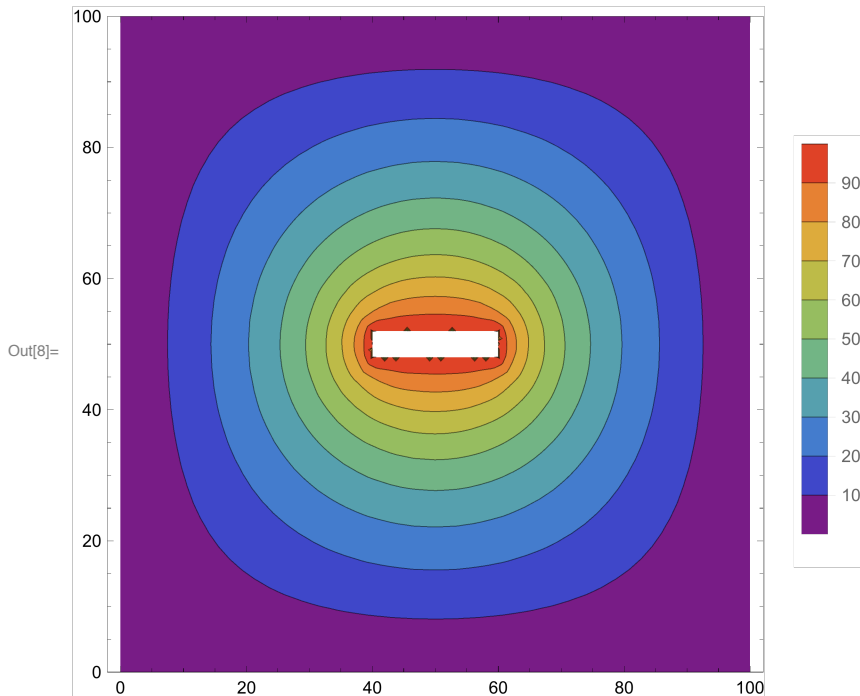
Out[7]= {Indeterminate}

(b) Generate a contour plot of the potential using `ContourPlot`. (Be brave; remember,

VV is an InterpretingFunction.)

In[8]:= (* Input code below - in this input cell *)

```
contoury = ContourPlot[VV[x, y], {x, 0, 100}, {y, 0, 100}, PlotRange -> {0, 100},
  ColorFunction -> "Rainbow", PlotRange -> All, PlotLegends -> Automatic]
```



(c) Comment on your plot.

<Enter comments in this text cell>

First, we again emphasize that we are showing in 2D what really is a 3D problem: the electrodes, both at constant potentials, extend to $\pm\infty$ in the z directions. Consequently the potential does not depend on z and we restrict the potential function to the x - y plane only: $V[x, y]$. Note that if you point your mouse at a contour you can read its magnitude.

The contours obviously fall in magnitude from the value of the central electrode ($V = 100$) to zero (V on the outer electrode). Their shapes evolve from the rectangle outlining the inner electrode to the square corresponding to the outer electrode. Simple results that agree with our intuition.

(d) Again bravely, apply $-\text{Grad}[VV[x, y], \{x, y\}]$ to find a representation of the electric field in the $\{x, y\}$ plane. (You end up with TWO InterpretingFunctions, one for E_x and one for E_y).

It helps to mask the center electrode with a suitably colored rectangle of the appropriate size. This can be created with the Graphics[] command. If you give it a name (I used "pinky", you can superimpose it on your field plot with the Show[] command. For pinky:

```
pinky = Graphics[{Pink, Rectangle[{40,48},{60,52}]}];
```

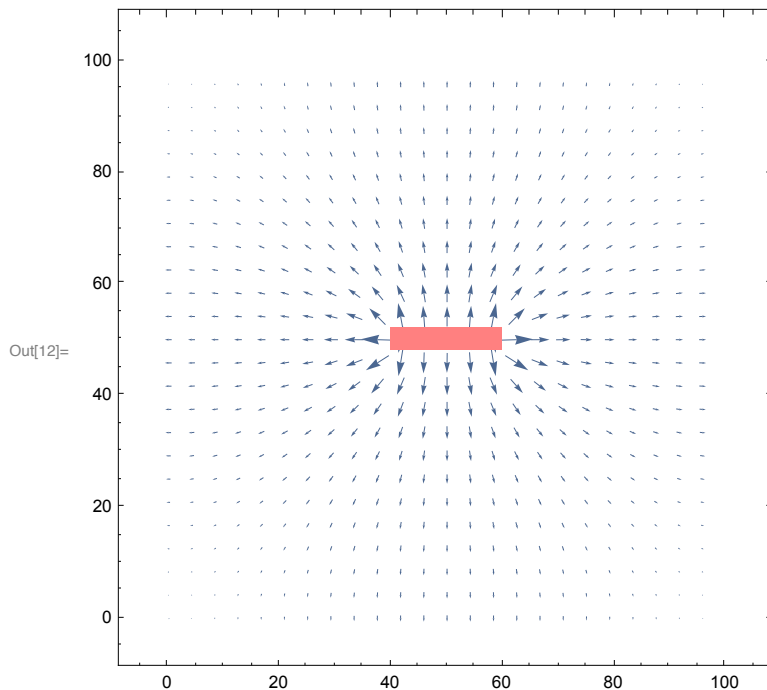
Plot this E field using VectorPlot (Brave! Be Brave!)

In[9]:= (* Input code below -- in this input cell *)

```
EE = -Grad[VV[x, y], {x, y}]
vec = VectorPlot[EE, {x, 0, 100}, {y, 0, 100}, VectorPoints -> Fine] ;
pinky = Graphics[{Pink, Rectangle[{40, 48}, {60, 52}]}];
(* I mask out the inner electrode *)
Show[vec, pinky]
```

Out[9]= $\left\{ \left\{ -\text{InterpolatingFunction} \left[\begin{array}{c} \text{Domain: } \{\{0., 100.\}, \{0., 100.\}\} \\ \text{Output: scalar} \end{array} \right] [x, y], \right. \right.$

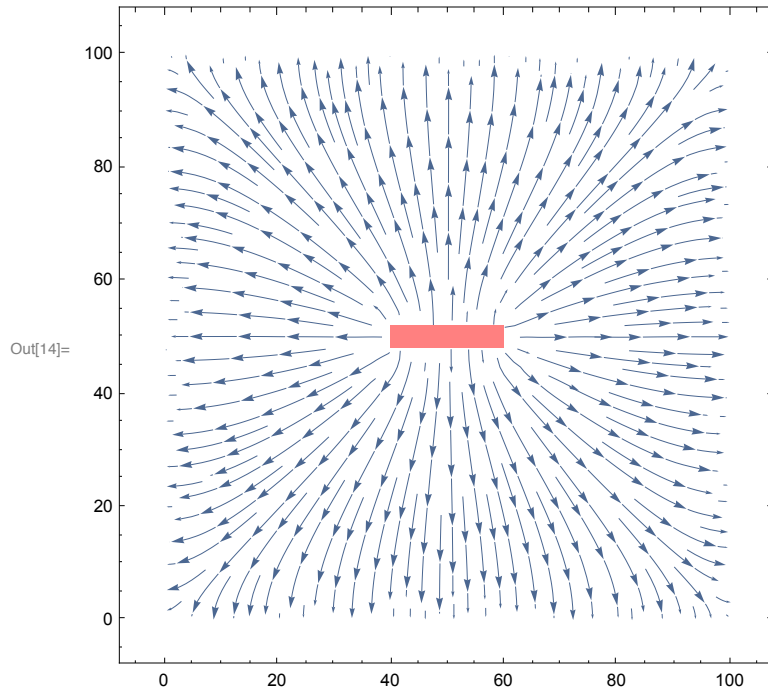
$\left. \left. -\text{InterpolatingFunction} \left[\begin{array}{c} \text{Domain: } \{\{0., 100.\}, \{0., 100.\}\} \\ \text{Output: scalar} \end{array} \right] [x, y] \right\} \right\}$



(e) Generate a StreamPlot of this field.

In[13]:= (* Input code below -- in this input cell*)

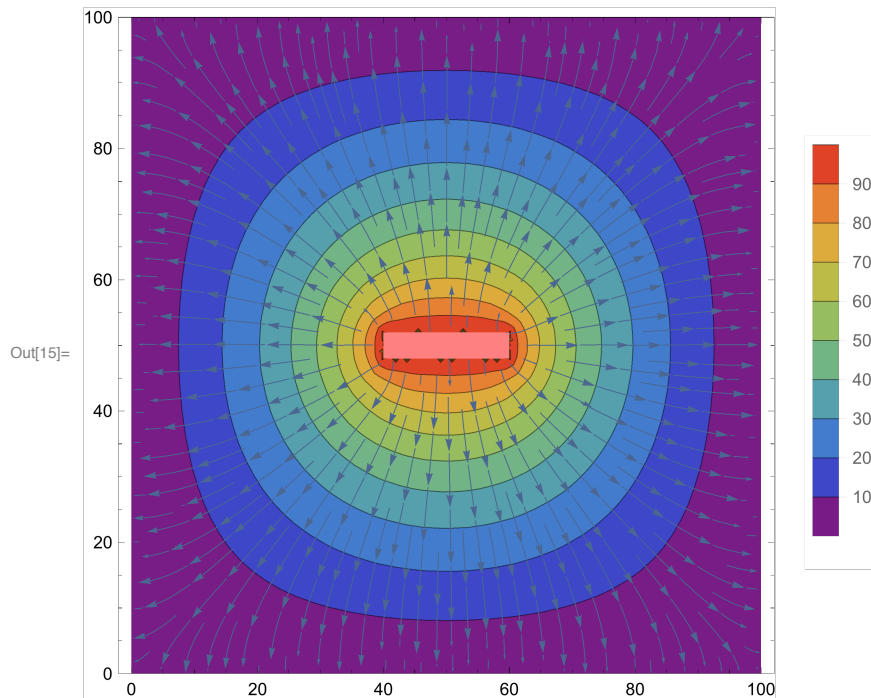
```
streamy = StreamPlot[EE, {x, 0, 100}, {y, 0, 100}, StreamPoints -> Fine];  
Show[streamy, pinky]
```



(e) Use Show, to superimpose this StreamPlot of E onto the ContourPlot of V. If you add pinky to the list of objects to show, the inner rectangle will appear here too.

In[15]:= (* Input code below -- in this input cell*)

Show[contoury, pinky, streamy]



(f) Handshake. State the obvious features if this composite plot.

<Enter your observations in this text cell>

We've already described the contours of the potential above.

The field lines are seen to be normal to the constant potential electrode surfaces AND are normal to the constant potential contours, as they should.

Very complex electrode shapes may need creative programming chops to set up (a) the surfaces of the electrodes and (b) defining Ω . Once done, however, finding V and E are trivial using `NDSolve` and `Grad`.