Laplace's Equation with Boundary Conditions in One Dimension

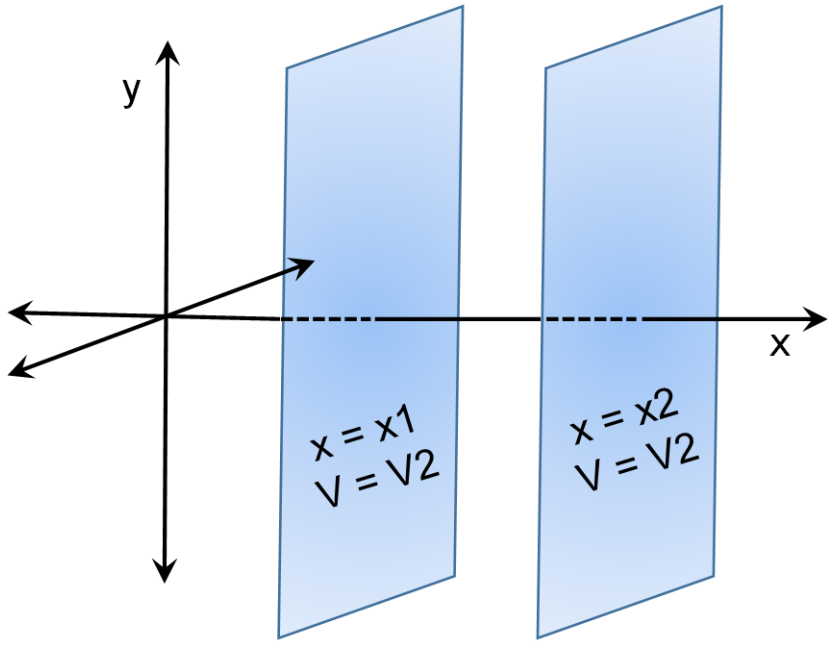
To date we have used Gauss's Law and the Method of Images to find the potential and electric field for rather symmetric geometries. For more complex geometries, V(x,y,z) can often be found by solving Laplace's equation: $\nabla^2 V(x,y,z) = 0$. Although the number of solutions to Laplace's equation is infinite, knowing either the potential or the field along the boundaries of the region of interest ("The Boundary Conditions") normally guarantees a unique solution (the uniqueness theorem).

Solving differential equations can be relatively difficult. In this module, we will restrict ourselves to simple 1-D systems for which solutions are readily derived. After learning what we can from these, we will move on to more complicated systems where we "up the ante".

In this context, a one-dimensional problem does not mean that our fields exist in a one-dimensional vector space. Our electric fields are vectors in 3-space, as they always are. Rather, the direction and magnitude of the electric fields only depend on one variable. In Cartesian space, this could be x. In this case, the direction and magnitude of the electric field would not depend on y or z. Similarly, the value of the potential would depend on x and not y or z.

Laplace's Equation in One Dimension—Infinite Parallel Plates

In the infinite parallel plate geometry, the fields and potentials depend on only one Cartesian variable, say x. The boundaries of the region of interest are planes parallel to the y-z plane, which we will assume intersect the x-axis at points x = x1 and x = x2. We will assume that V(x1, y, z) = V1 (a specific value of potential) for all values of y and z. Similarly V(x2,y,z) = V2. Between x1 and x2, the potential will be some function V(x). This is the geometry of a parallel plate capacitor with plates that extend to infinity in the y and z directions. You will remember from your work with Coulomb's Law and Gauss's Law that V(x) in this system is proportional to x and the E field is constant in magnitude and direction ($\pm \hat{\boldsymbol{x}}$ — in the direction of decreasing potential).



In 1-D Cartesian coordinates, Laplace's equation takes the form

$$\frac{d^2 V[x]}{d x^2} = 0$$

(a) Although you probably know the solution, use M's **DSolve[V''[x] == 0, V[x], x]** command to solve this equation. By NOT inserting the BCs, we will see that we can generate a general solution.

The double prime after the V in the first argument indicates the second derivative. (The double prime is not the quotation mark, ", it is two hits on the apostrophe key, ' '.) We seek V[x], the second argument of the command, and the derivatives are taken with respect to x, the third argument. To refer to the solution later, set the variable *sol* to the result of the DSolve[] command.

```
(* Input code below *)
ClearAll["Global`*"] (* Leave the ClearAll command *)
```

M returns a list of solutions (in the case ONLY ONE!). The entire list is enclosed in brackets, and each solution is enclosed in its own, additional brackets. The solution {V[x] → C[1] + x C[2]} is part of this larger list, so we get nested brackets.

(b) Execute sol[[1]]; this should get you this single bracket element of sol:

```
(* Input code below *)
```

Technically, M treats the solution sol[[1]] as a "rule" that can be applied in various ways. We want to convert this rule to a function. We can use way the "ReplaceAll" command, usually written "/." (without the quotes) to replace V[x] with the rule given by sol[[1]].

(c) Type in and execute V[x]/.sol[[1]]

```
(* Input code below *)
```

This is still not a function that we can plot.

(d) To get a function, we need to set a real function (in x_) equal to V[x]. This adds an extra step. (use % for the above result to achieve this definition, that is, V[x_] = %.)

```
(* Input code below *)
```

(e) Just as a check enter and execute the “new” V[x]

```
(* Input code below *)
```

As expected, this solution is a linear function of x. We started with V’’[x], the second derivative. M had to integrate twice to find the solution. C[1] and C[2] are the two constants of integration.

(f) To meet our boundary conditions, we require V[x1] = V1 and V[x2] = V2. Use these two requirements and M's Solve[] command to determine the values C[1] and C[2]. If the uniqueness theorem applies to this problem (It Does!), only one choice of C[1] and C[2] will work. Like DSolve[], Solve[] returns a nested list of solutions. We must specify which solution we want, even if there is only one. SO: Use Solve and extract the two constants C[1] and C[2] that satisfy our BCs. Then use the ReplaceAll option (or your two hands) to input these values into your V[x] function.

```
(* Input code below *)
```

Handling M’s constants of integration is tricky. Here is another way to do it. Introducing C1 and C2 as temporary constants gets around M’s touchiness regarding the symbol “C”

The ReplaceAll trick works only once per M session *unless you change* the constant names. We can’t change the names of C[1] and C[2]. If you get an error message, quit M and re-execute only one version of of the ReplaceAll command.

Study, but do not execute code below.

```
sol3=Solve[{V[x1] ==V1,V[x2]==V2}, {C[1],C[2]}]
C1= C[1]/.sol3[[1]]
C2 = C[2]/.sol3[[1]]
V[x_]  = C1 + C2 x
```

(g) To verify that this choice of C[1] and C[2] satisfies the boundary conditions, evaluate V[x1] and V[x2] explicitly. [You are trying to show that V[x1] = V1 and V[x2] = V2.

The Simplify[] command may be useful here. For instance: Simplify[V[x1]]. (This also works: V[x1]//Simplify) // = PostFix)

```
(* Input code below *)
```

We see that the BC's are satisfied. If you get something like C[1]+x1 C[2], you need to make sure V[x_] is defined properly in terms of x1, x2, V1 and V2.

(h) Now use DSolve with BCs inserted (notice how it saves a lot of trouble) to find the potential between the plates where again
(code will look something like this: **sol2=DSolve[{V''[x]==0,V[x1] ==V1, V[x2] ==V2}, V[x],x]**
Just a reminder: the plates are located at x = x1 and x = x2. V(x1, y, z) = V1 and V(x2,y,z) = V2.

```
(* Input code below *)
ClearAll["`*"] (* Leave the ClearAll statement.  We are starting over. *)
```

Same as V[x] above. Here we have let M do all the work.

(i) Griffith's (in Section 3.1.2) claims that all solutions V[x] of the 1-D Laplace's equation are feature-less, and the potential at x is the average of the potentials at x+a and x-a; it has no minimum or maximum except at the boundaries. [You can think of *a* being small but it's not necessary].

We therefore expect: $V[xo] = \frac{V[xo-a] + V[xo+a]}{2}$

Use M to see if Wolfram agrees. (Use the test: == ; might want to use Simplify.)

```
(* Input code below *)
```

So the statement that the potential at x is the average of the potentials at x+a and x–a is true.

This general result is important, because it shows that the solutions of Laplace's equation have "no local minima or maxima". All the minima and maxima potential values lie on the boundaries. This is also the basis of the "relaxation method" for numerically solving Laplace's equation in Cartesian coordinates.

(j) For laughs, show that this is true for some consistent (numerical) choices of x1, x2, xo (located between x1 and x2), V1, and V2, and some arbitrary *a*.

I chose:

x1 = 1; x2 = 2;
V1 = 10; V2 = 100;
xo = 1.5; *a* = 0.02;

```
(* Input code below *)
```

(k) Now Plot the Solution for x1 < x < x2.

```
(* Input code below *)
```

(l) Although somewhat mundane, interpret your results. (Is the result consistent with your previous work in this geometry? — think parallel plate capacitor.)

<Enter interpretation in this text cell>

(m) Find the E Field. In 3D, EE = –Grad[V]; This yields a 3D Vector.

The electric field amplitude should be MINUS the slope of the potential plot times (–1), which is about –(90 V)/(1 m) or –90 V/m.

```
(* Input code below *)
Clear [xo, x1, x2, V1, V2, a] (* Leave the Clear statement *)
```

$\overrightarrow{EE}$[x] is CONSTANT VECTOR and points from the more positive plate towards the less positive plate (in the $-\hat{x}$ direction). The E Field is UNIFORM between the plates.

Everything looks consistent with our previous Coulomb's Law/Gauss's Law approach.

A general expression for just the x-component of the E field (the x-component being the only non-zero

component) is given below.

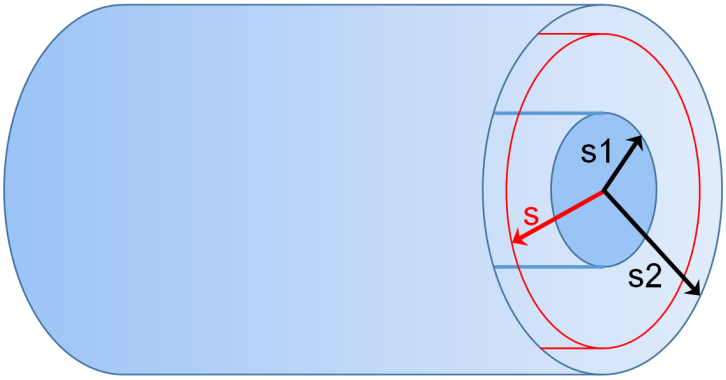
```
(* Execute code below *)

V[x]
EE[x_] = -D[V[x], x] (* note that this is the - of the slope of V[x] *)
```

So lets rewrite: $\overrightarrow{EE}[x] = \frac{V2-V1}{x2-x1}\,\hat{x}$. For V2 > V1 and x2>x1, $\overrightarrow{EE}[x]$ points in the + x direction as it should.

Not much more to say here. BUT: by going through this simple 1D potential problem using Laplace's Equation we get an understanding of the importance of BCs, the nature of the resulting potential, and the simple connection of the E-field to this V.

Laplace's Equation in One Dimension—Concentric Cylinders

In systems with cylindrical symmetry, the potential is a function of the radial coordinate, s, and independent of ϕ and z. We will seek the potential between two circular cylinders which serve as boundaries, where the boundaries are equipotentials. The boundary conditions then take the form V(s1) = V1 and V(s2) = V2.



Ignoring the terms in ϕ and z, Laplace's equation in cylindrical coordinates is given by:

$$\frac{1}{s}\frac{\partial}{\partial s}\left(s\frac{\partial V}{\partial s}\right) = \frac{\partial^2 V}{\partial s^2} + \frac{1}{s}\frac{\partial V}{\partial s} = 0$$

By convention, we multiply each term by s before attempting a solution. Factors of s are often easier to work with than factors of $(1/s)$, that is, s*V''[s] + V'[s] == 0

(n) Use M to find V[s] using : **DSolve[{s*V''[s] + V'[s] == 0, V[s1] ==V1,V[s2]==V2 }, V[s], s]**. WE'RE putting in the BCs into DSolve from the start. This gives us the particular solution. Is your solution consistent with what you found from Gauss's Law earlier in the semester? Comment.

```
(* Input code below *)
ClearAll["Global`*"] (* Leave the ClearAll statement *)
```

<Enter comment in this text cell>

(o) Verify that these expressions satisfy the boundary conditions (i.e., does V[s1] = V1, etc.??) Comment.

```
(* Input code below *)
```

<Enter comment in this text cell>

(p) Plot V[s] for the co-axial cylinders. For plotting, I assumed:

s1 = 1; s2 = 2;

V1 = 10; V2 = 100;

```
(* Input code below *)
```

Note that for the given constants s1, s2, V1, V2, the *slope* of the potential function V[s] is monotonically decreasing; as expected; no min or max in V[s] occurs except at the endpoints (the boundaries).

(q) Clear the constants (s1, s2, V1, V2) and find the ONE DIMENSIONAL E field from V[s].

Use M's Grad function in Cylindrical Coordinates.

Here is some help:

M writes it as: Grad[f[r,θ,z],{r,θ,z},"Cylindrical"], where f is a general, undefined, scalar function in cylindrical coordinates. Evaluate the following:

```
(* Execute code below *)
Clear[s1, s2, V1, V2]
Grad[f[s, ϕ, z], {s, ϕ, z}, "Cylindrical"]
```

M assumes that EACH component of the gradient can have a dependence on all three coordinates. The result is a 3D Vector which is easily translated into:

$$\left\{\frac{\partial}{\partial s}f[s,\phi,z],\ \frac{\frac{\partial}{\partial \phi}f[s,\phi,z]}{s},\ \frac{\partial}{\partial z}f[s,\phi,z]\right\}$$

Note M's notation: $[f^{(1,0,0)}[s,\phi,z] = \frac{\partial}{\partial s}f[s,\phi,z],\ \text{etc.}]$

For our task, I suggest that you simply enter: Ecylindrical[s_] = –Grad[V[s],{s,ϕ,z},"Cylindrical"] and execute. Don't overlook the – sign. The lack of ϕ and z dependence in V does its magic (i.e., the ϕ and z components come out zero.)

```
(* Input code below *)
```

Repeat: Since V[s] is a function of s only, only the s component of the Gradient is non zero $\left(\frac{\partial}{\partial \phi}V[s]\text{ and }\frac{\partial}{\partial z}V[s]\text{ both }=0\right)$.

Given the symmetry of the problem we are not surprised to see that Ecylindrical is:

A function of *s* only

AND

The only non-zero component is the $\hat{s}$ component.

(r) Plot the 2D EE field using Vector Field. (Since the field does not depend on z, consider z = 0).
REMEMBER: M requires that we transform EEcylindrical into EEcartesian for plotting.

```
(* Execute the code below to express the
 field in terms of the Cartesian coordinates x and y *)

EECartesian[s_] = TransformedField[
  "Cylindrical" → "Cartesian", Ecylindrical[s], {s, ϕ, ξ} → {x, y, z}]
(* does not like a common z!!! so we write ξ instead of z in {s,ϕ,ξ} *)
EECartesian2D[x_, y_] = {EECartesian[s][[1]], EECartesian[s][[2]]}
(* selects out the x,y components *)
```

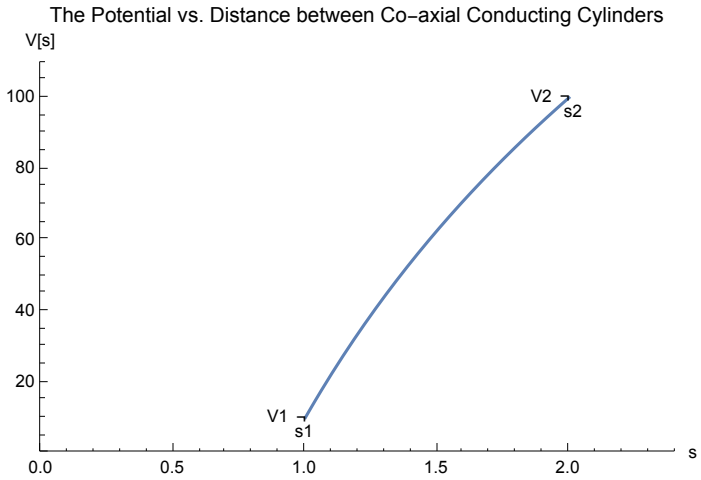
Now put in some values for the constants and plot

```
(* Input code below *)
```

Remember this is a 2D PLOT. It represents a cut through the cylinders normal to the z axis at (any) particular z, say z = 0.

The E vectors are all radial and only are defined between the electrodes. By default, M positions the middle of each arrow at the location of the vector. Therefore the heads and tails will appear to cross the boundaries at points. The middle of each arrow lies between the boundaries.

For our assumed constants, E points from the outer more positive electrode (cylinder), inward. E increases in intensity (the vectors get longer) as one moves from s2 into s1. This is consistent with the SLOPE of V[s] (largest at s = s1). (Again, don't forget the – sign in $\vec{E}$ = – Grad[V] (so the E vectors point towards smaller, decreasing s). Here is a copy/paste of the V[s] plot above.



Just to nail our understanding, we will generate the contours of V[s] (between the electrodes), the stream-line plot of the E field, and their superposition. Note: to plot the contour map, we have to convert V[s] into it's Cartesian form (in this case, it is simply substituting s = $\sqrt{x^2 + y^2}$ into V[s] (from above, V[s] = $\frac{\text{V1 Log[s]–V2 Log[s]+V2 Log[s1]–V1 Log[s2]}}{\text{Log[s1]–Log[s2]}}$)

I use M's TransformedField function to do the work; this time it is transforming a scalar function, V[s].

```
(* Execute code below *)
Clear[s1, s2, V1, V2] (* Leave the Clear statement *)

VV[x_, y_] =
 TransformedField["Cylindrical" → "Cartesian", V[s], {s, θ, ζ} → {x, y, z}]
s1 = 1; s2 = 2;
V1 = 10; V2 = 100;
contoury = ContourPlot[
   If[√(x^2 + y^2) ≤ s1 || √(x^2 + y^2) ≥ s2, Null, VV[x, y]], {x, -3, 3}, {y, -3, 3}] ;
Show[contoury, innerelectrode, outerelectrode]
```

As expected the contours of V[s] are circles (s = constants); in 3D they would be concentric cylinders. If you want to see the magnitudes of V corresponding to each contour, click your mouse inside the plot and point to the contour lines.

Now we plot the streamlines:

```
(* Execute code below *)
streamy =
  StreamPlot[If[Sqrt[x^2 + y^2] ≤ s1 || Sqrt[x^2 + y^2] ≥ s2, {0, 0}, EECartesian2D[x, y]],
   {x, -3, 3}, {y, -3, 3}, StreamPoints → 100];

Show[streamy, innerelectrode, outerelectrode]
```

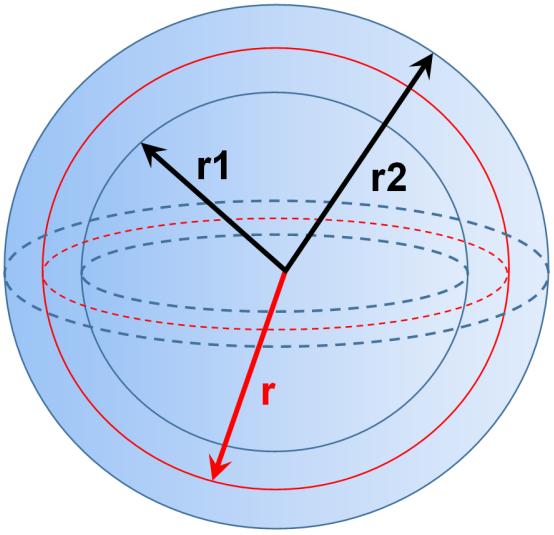
And finely superimpose the streamlines onto the contourplot.

```
(* Execute code below *)
Show[ contoury, innerelectrode, outerelectrode, streamy]
```

One more time — because of the symmetry, this is a ONE D Laplace's Equation problem. (Remember, this is a SLICE through the CYLINDERS, e.g., at z = 0).

Laplace's Equation in One Dimension—Spherical Symmetry

In systems with spherical symmetry, the potential is a function of the radial coordinate, r, and independent of θ and ϕ. We will seek the potential between two concentric spheres which serve as boundaries, where the boundaries are equipotentials. The boundary conditions then take the form V(r1) = V1 and V(r2) = V2.



Laplace's equation for spherical coordinates, ignoring the terms in θ and ϕ, is:

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial V}{\partial r}\right) = \frac{2}{r}\frac{\partial V}{\partial r} + \frac{\partial^2 V}{\partial r^2} = 0$$

It is conventional to multiply each term (including the terms in θ and ϕ, which are not shown) by r^2. This

removes the r's in the denominator of each term (including those not shown).

(s) Use the M DSolve function with the BCs: V[r1] = V1 and V[r2] = V2 to find V[r] between the spherical electrodes. Is your solution consistent with what your found earlier in the semester, using Gauss's Law? Comment.

```
(* Input code below *)
ClearAll["Global`*"] (* Leave the ClearAll statement *)
```

<Enter comment in this text cell>

(t) Check to make sure V[r] does indeed satisfy the boundary conditions, V[r1] = V1 and V[r2] = V2.

```
(* Input code below *)
```

(u) Plot your results for the potential as a function of r for the the concentric spheres. For plotting, I assumed:

r1 = 1; r2 = 2;
V1 = 10; V2 = 100;

```
(* Input code below *)
```

Note that for the given constants r1, r2, V1, V2, the slope of the potential function V[r] is monotonically decreasing; as expected, no min or max in V[r] occurs except at the endpoints (the boundaries).

(v) Clear the constants (r1, r2, V1, V2) and find the ONE DIMENSIONAL E field from V[r].
Use M's Grad function in Spherical Coordinates.

Here is some help:

M writes it as: Grad[f[r,θ,z],{r,θ,z},"Spherical"], where f is the scalar function being operated on. Evaluate the following:

```
(* Execute code below *)
Clear[r1, r2, V1, V2]
Grad[f[r, θ, ϕ], {r, θ, ϕ}, "Spherical"]
```

Here f[r, θ, ϕ] is a general, undefined, scalar function in spherical coordinates; M assumes that EACH component of the gradient can have a dependence on all three coordinates. The result is a 3D

Vector which is easily translated into:

$$\left\{\frac{\partial}{\partial s}f[r,\theta,\phi],\ \frac{\frac{\partial}{\partial\theta}f[r,\theta,\phi]}{r},\ \frac{\text{Csc}[\theta]\ \frac{\partial}{\partial\phi}f[r,\theta,\phi]}{r}\right\},\quad \text{which should look familiar.}$$

Note M's notation: $f^{(0,0,1)}[r,\ \theta,\ \phi] = \frac{\partial}{\partial\phi}f[r,\ \theta,\ \phi]$]

SO: suggest that you simply enter: Espherical[r_] = –Grad[V[r],{s,θ,ϕ},"Spherical"] and execute. Don't overlook the – sign. The lack of ϕ and z dependence in V does its magic (i.e., the ϕ and z components come out zero.)

```
(* Input code below *)
```

Broken Record: This operation yields a 3D vector. Since V[r] is a function of r only, only the r component of the Gradient is non zero $\left(\frac{\partial}{\partial\theta}V[r]\text{ and }\frac{\partial}{\partial\phi}V[r]\text{ both }=0\right)$.

Given the symmetry of the problem we are not surprised to see that Espherical is:

A function of r only

AND

The only non-zero component is the $\hat{r}$ component.

(w) Plot the 2D EE field using Vector Field (there is no z dependence; consider letting z = 0). REMEMBER: M requires that we transform EEspherical into EEcartesian for plotting.

```
(* Execute code below *)
EECartesian[x_, y_, z_] =
 TransformedField["Spherical" → "Cartesian", Espherical[r], {r, θ, ϕ} → {x, y, z}]
EECartesian2D[x_, y_] = {EECartesian[x, y, 0][[1]], EECartesian[x, y, 0][[2]]}
 (* selects out the x,y components *)
```

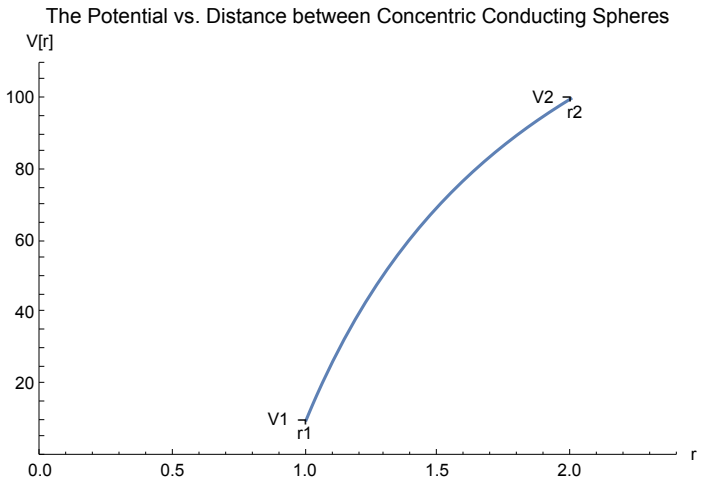
(x) Now put in some values for the constants and make a VectorPlot of the E-Field.

```
(* Input code below *)
```

Remember this is a 2D VectorPlot. It represents a cut through (or cross section of) the spheres. It's easiest to think of a plane passing through the polar (z) axis which of course passes through the origin.

The E vectors are all radial and only are defined between the electrodes. For our assumed constants, E points from the outer more positive sphere, inward. E increases in intensity (the vectors get longer) as

one moves from r2 into r1. This is consistent with the SLOPE of V[r] (largest at r= r1). (Again, don't forget the –sign in $\vec{E}$ = – Grad[V] (so the E vectors point towards smaller, decreasing r.)



Just to nail our understanding, we will generate the contours of V[r] (between the electrodes), the stream-line plot of the E field, and their superposition. Note: to plot the contour map, we have to convert V[r] into it's Cartesian form.

I use M's TransformedField function to do the work; again, it is transforming a scalar function, V[r].

```
(* Execute code below *)
Clear[r1, r2, V1, V2] (* Leave Clear statement *)

VV[x_, y_] =
 TransformedField["Spherical" → "Cartesian", V[r], {r, θ, ϕ} → {x, y, z}] /. z → 0
r1 = 1; r2 = 2;
V1 = 10; V2 = 100;
contoury = ContourPlot[
   If[Sqrt[x^2 + y^2] ≤ r1 || Sqrt[x^2 + y^2] ≥ r2, Null, VV[x, y]], {x, -3, 3}, {y, -3, 3}];
Show[contoury, innerelectrode, outerelectrode]
```

As expected the contours of V[r] are circles (r= constants); in 3D they would be concentric spheres.

Now we plot the streamlines:

```
(* Execute code below *)
streamy =
  StreamPlot[If[Sqrt[x^2 + y^2] ≤ r1 || Sqrt[x^2 + y^2] ≥ r2, {0, 0}, EECartesian2D[x, y]],
   {x, -3, 3}, {y, -3, 3}, StreamPoints → 100];

Show[streamy, innerelectrode, outerelectrode]
```

And finely superimpose the streamlines onto the contour plot.

```
(* Execute code below *)
Show[ contoury, innerelectrode, outerelectrode, streamy]
```

One more time — because of the symmetry, this is a ONE D Laplace's Equation problem (r dependent only). Remember, the plots above are slices through the SPHERES.

Clearly, and not surprising, the basic behavior of the 2D plots for the cylinders and the spheres look somewhat similar — BUT, importantly, they are represented by different functions.