How to plot Electric Field Lines in Maple PhL 11.26.10

Question: Given some explicit expression for the potential $\Phi(\mathbf{r})$, is there a way to write an equation for the electric field lines, perhaps with parameters that distinguish the different lines?

Amazingly, I have never asked this question before as best I can remember, nor have I seen the question asked by any author that I can remember. Authors always draw pictures showing the field lines, but never seem to comment on how they obtained these lines. [Jim says Mathews & Walker and Courant & Hilbert have something to say on this subject, I have not looked.]

1. The General Answer to this Question in 2D 1

2. The General Answer to this Question in 3D 4

3. A Numerical Method of drawing the field lines 7

(a) Wrong path notes. 7

(b) Corrected Notes. 8

4. It is a lot easier to use gradplot3d. 9

5. Maple Success 9

Overview? None needed really, contents I think is sufficient.

1. The General Answer to this Question in 2D

Imagine a 2D curvilinear coordinate system with the following inverse equations

$$\varphi = \Phi(x,y)$$
$$\eta = f(x,y)$$

where Φ is our potential, and f is some function we seek which will cause this to be an orthogonal coordinate system! The curves of constant φ are the equipotential lines we all know and love. The curves of constant η will be orthogonal to these equipotential lines, and will thus be equations for the electric field lines. The catch is that we have to find a function f that makes this work.

One idea is to require that the metric tensor be diagonal. We can blindly do this little computation as follows. Let's regard φ,η as x'_1 and x'_2 and then $x,y = x_1$ and x_2.

$$T^a{}_b = \frac{\partial x'^a}{\partial x^b} \qquad T^1{}_i = \partial_i\varphi \qquad T^2{}_i = \partial_i\eta$$

$$g'^{ab} = T^a{}_c \; T^b{}_c$$

$$g'^{12} = T^1{}_i \; T^2{}_i = \Sigma_i(\partial_i\varphi)(\partial_i\eta) = 0$$

The condition we have found can be written as

$$\nabla\Phi \bullet \nabla f = 0$$

and we see of course that this condition is obvious without thinking about metric tensors. It says that at a point in 2D space (x,y), where two curves intersect, the normals to the curves are at right angles. This just says the curves intersect at right angles, which is what we mean by an orthogonal system.

So how, given Φ, might we find f to make this work? If we write this condition out, it is this:

$$\Phi_x \, \partial f/\partial x + \Phi_y \, \partial f/\partial y = 0$$

This is a first-order partial differential equation in 2 variables x,y for which the coefficient functions are assumed to be something complicated.

How to you solve such an equation?

I see that my friend Polyanin has yet another handbook out, Handbook of First-Order Partial Differential Equations, $160 Amazon. A partial view of that book provides this fascinating information:

2.1.1. Solution Method

1. The characteristic equation. Consider a first order linear homogeneous partial differential equation with two independent variables of the form

$$f(x,y)\frac{\partial w}{\partial x} + g(x,y)\frac{\partial w}{\partial y} = 0. \tag{1}$$

The first order ordinary differential equation

$$\frac{dx}{f(x,y)} = \frac{dy}{g(x,y)} \tag{2}$$

is called the characteristic equation corresponding to the partial differential equation (1). The integral curves of equation (2) are called characteristics.

Remark 1. Suppose the variables x and y belong to a domain G. Let the functions $f(x,y)$ and $g(x,y)$ be continuously differentiable with respect to both x and y in G and let $f^2(x,y)+g^2(x,y)\neq 0$ in G. Then only one characteristic passes through each point of G.

2. Formula for the general solution. Let the general solution of the characteristic equation (2) be given by

$$\Xi(x,y) = C, \tag{3}$$

where C is an arbitrary constant. Then the general solution of equation (1) has the form

$$w = \Phi(\Xi), \tag{4}$$

where $\Phi = \Phi(\Xi)$ is an arbitrary function. The left-hand side of Eq. (3), $\Xi(x,y)$, is referred to as a principal integral of the partial differential equation (1).

Remark 2. For brevity, only principal integrals (3) will often be presented in Chapter 2. The general solution of the equation is given by Eq. (4).

So my PDE is this

$$\Phi_x \, \partial f/\partial x \; + \Phi_y \, \partial f/\partial y \; = 0$$

so we are supposed to go off and ponder this equation

$$dx/\, \Phi_x \; = dy/\, \Phi_y$$

which I can write as an ODE with unknown function y(x)

$$dy/dx - s(x,y) = 0 \qquad\qquad s(x,y) = \Phi_y/\Phi_x$$

In general, this is a *non-linear* first order ODE. Suppose I could solve this for

$$y(x) = F(x)$$

I would then have

$$y - F(x) = 0$$

so I could regard $\Xi(x,y) = y - F(x)$ and $C = 0$. Then one solution of my partial DE would be

$$\Phi_x \, \partial f/\partial x \; + \Phi_y \, \partial f/\partial y \; = 0$$

$$f(x,y) = y - F(x)$$

where I choose the identity as his function Φ.

So how do you solve a first order *nonlinear* differential equation? Here are someone's comments on such things:

First Order Nonlinear Equations

The most general nonlinear first order ordinary differential equation we could imagine would be of the form

$$F(t,y(t),y'(t)) = 0. \qquad (1)$$

In general we would have no hope of solving such an equation. A less general nonlinear equation would be one of the form

$$y'(t) = F(t,y(t)), \qquad (2)$$

but even this more general equation is often too difficult to solve. We will consider then, equations of the form

$$y'(t) = F(y(t)). \qquad (3)$$

Equation (3) is said to be an **autonomous** differential equation, meaning that the nonlinear function F does not depend explicitly on t. The equation (2) is nonautonomous because F does contain explicit t dependence. The equations,

$$y'(t) = y(t)^2 \quad \text{and} \quad y'(t) = y(t)^2 + t^2,$$

So my equation dy/dx – s(x,y) = 0 with s(x,y) = Φ_y/Φ_x falls into the class (2) above with F =s, t = x so he is saying dy/dx = s(x,y).

I agree with this author that there is no "general solution" you can write for my first order nonlinear ODE for "general Φ".

The conclusion then is that for some complicated Φ, you don't stand an ice cube's chance in hell of finding any reasonable equation for the electric field lines. However, in 2D I suppose conformal mapping can in fact solve this problem and give you some horrible equation. But my real interest is 3D.

2. The General Answer to this Question in 3D

Imagine a 3D curvilinear coordinate system as with the following inverse equations

$\varphi = \Phi(x,y,z)$
$\eta = f(x,y,z)$
$\xi = f(x,y,z)$

where Φ is our potential, and f and g are some functions we seek which will cause this to be an orthogonal coordinate system! The surfaces of constant φ are the equipotential surfaces we all know and love. The surfaces of constant η will be orthogonal to these equipotential surfaces. We could regard such a surface as a flat bundle of electric field lines any of which is a legal field line. The same could be said for the ξ surfaces. So if we pick a point in space and have an equipotential surface passing through this point, the field line through that point will lie on the intersection of the unique η and ξ surfaces which

intersect the point. This field line will be perpendicular to the surface. The catch is that we have to find functions f and g that make this work.

One idea is to require that the metric tensor be diagonal. We can blindly do this little computation as follows. Let's regard φ,η,ξ as x'_1 and x'_2 and x_3' and then $x,y,z = x_1$ and x_2 and x_3

$$T^a{}_b = \frac{\partial x'^a}{\partial x^b} \qquad T^1{}_i = \partial_i\varphi \qquad T^2{}_i = \partial_i\eta \qquad T^3{}_i = \partial_i\xi$$

$$g'^{ab} = T^a{}_c \; T^b{}_c$$

$$g'^{12} = T^1{}_i \; T^2{}_i = \Sigma_i(\partial_i\varphi)(\partial_i\eta) = 0 \qquad \nabla\Phi \bullet \nabla f = 0$$
$$g'^{13} = T^1{}_i \; T^3{}_i = \Sigma_i(\partial_i\varphi)(\partial_i\xi) = 0 \qquad \nabla\Phi \bullet \nabla g = 0$$
$$g'^{23} = T^1{}_i \; T^3{}_i = \Sigma_i(\partial_i\eta)(\partial_i\xi) = 0 \qquad \nabla f \bullet \nabla g = 0$$

Any point has three intersecting surfaces, and the normals to these are all perp to each other, and again this just says our system is orthogonal. So now we have a system of two first order PDE's in three variables

$$\Phi_x \, \partial f/\partial x + \Phi_y \, \partial f/\partial y + \Phi_z \, \partial f/\partial z = 0$$
$$\Phi_x \, \partial g/\partial x + \Phi_y \, \partial g/\partial y + \Phi_z \, \partial g/\partial z = 0$$

with a third equation that acts as a condition that must be satisfied, namely $\nabla f \bullet \nabla g = 0$. There are as many solutions to this problem as there are orthogonal systems having $\varphi = \Phi(x,y,z)$ as one coordinate, so nothing will be unique. If we look into the theory of first order PDE's with three variables. Here is what Polyanin has to say about just one PDE of the above form

6.1.1. Solution Methods

1. Characteristic system. Structure of the general solution. Consider a first order linear homogeneous partial differential equation with three independent variables of the form

$$f(x,y,z)\frac{\partial w}{\partial x} + g(x,y,z)\frac{\partial w}{\partial y} + h(x,y,z)\frac{\partial w}{\partial z} = 0. \tag{1}$$

Given an integral basis, two different (functionally independent) integrals,

$$u_1(x,y,z) = C_1, \qquad u_2(x,y,z) = C_2, \tag{2}$$

of the characteristic system

$$\frac{dx}{f(x,y,z)} = \frac{dy}{g(x,y,z)} = \frac{dz}{h(x,y,z)}, \tag{3}$$

the general solution of equation (1) is defined by

$$w = \Phi(u_1, u_2), \tag{4}$$

where Φ is an arbitrary function of two variables.

For the specific equations considered below in Sections 6.2–6.8, only an integral basis will be often presented. The general solution of these equations can be obtained using Eq. (4).

2. The method of solution based on a change of variables. Suppose an integral $u(x,y,z) = C$ of system (3) is known. By switching from x, y, and z to the new variables x, y, and $u = u(x,y,z)$, we arrive at a first order linear homogeneous partial differential equation with two independent variables x and y,

$$\overline{f}(x,y,u)\frac{\partial w}{\partial x} + \overline{g}(x,y,u)\frac{\partial w}{\partial y} = 0, \tag{5}$$

in which u occurs as a parameter. The functions $\overline{f}(x,y,u)$ and $\overline{g}(x,y,u)$ are obtained from $f(x,y,z)$ and $g(x,y,z)$ by passing on to the variables x, y, and u.

For the solution of equation (5), see Section 1.2.

I don't understand this stuff of course, but I can see that this 3D case is even worse than the 2D case, so there is not going to be a magic general solution that you roll out for Φ.

And the answer is: for some general complex Φ of the type we get in our typical "simple" problems like the In-plane Green's Function for the iris, you are not going to find some nice equation that gives you the electric field lines!

3. Reminder of how you do a numerical solution of an ODE

Suppose our ODE is this

y'' + by' + cy = 0 b and c are functions of x

We *don't know* the solution y(x), that is what we are trying to find. So we do this iteration scheme

y(x+dx) = y(x) + dx y'(x)

$y'(x+dx) = y'(x) + dx\ y''(x)$
$y''(x+dx) = -\ b(x)y'(x) - c(x)y(x)$ // ie, this last is from the ODE

You pick some starting point x_0 and you do the above "triple iteration" to the right or left, and "build out" your solution function y(x) one step at a time. If the steps are small enough, your y(x) will be reasonably accurate at least for some distance. This is a whole subject in Scheid's book, including of course PDE's. The key thing above is that the ODE "closes" the series of iterators which would otherwise go on forever. If you had a 10th order equation, you would have 10 iterators plus the closing equation.

3. A Numerical Method of drawing the field lines

(a) Wrong path notes. Let $\mathbf{n}(\mathbf{r}) = \nabla\varphi(\mathbf{r})$. Since we assume we know all about $\varphi(\mathbf{r})$, we can evaluate the vector function $\mathbf{n}(\mathbf{r})$ at any point in space. If we start at point $\mathbf{r}$ and we want to stay "on the field line", we need to move in space by a small amount $d\mathbf{r} = \hat{\mathbf{n}}\ ds$. In general we can say

$$n_i(\mathbf{r}+d\mathbf{r}) \approx n_i(\mathbf{r}) + d\mathbf{r} \bullet \nabla n_i \qquad i = 1,2,3$$

which we can write as

$$\mathbf{n}(\mathbf{r}+d\mathbf{r}) \approx \mathbf{n}(\mathbf{r}) + d\mathbf{r} \bullet \nabla\mathbf{n}(\mathbf{r})$$

with the understanding that $[\nabla\mathbf{n}]_i = \nabla n_i$. So as we march along a field line, we have this going on

$$\mathbf{r}' = \mathbf{r} + \hat{\mathbf{n}}\ ds$$
$$\mathbf{n}(\mathbf{r}') \approx \mathbf{n}(\mathbf{r}) + d\mathbf{r} \bullet \nabla\mathbf{n}(\mathbf{r})$$

We can write this as an iteration scheme in this manner

$$\mathbf{r}_{i+1} = \mathbf{r}_i + \hat{\mathbf{n}}(\mathbf{r}_i)\ ds$$
$$\mathbf{n}(\mathbf{r}_{i+1}) = \mathbf{n}(\mathbf{r}_i) + ds\ \hat{\mathbf{n}}(\mathbf{r}_i) \bullet \nabla\mathbf{n}(\mathbf{r}_i)$$

By the way, this thing $\nabla\mathbf{n}$ really is the "gradient of the gradient" of the potential, it has no other name that I can find. You can write the above as

$$n_k(\mathbf{r}_{i+1}) = n_k(\mathbf{r}_i) + ds\ \hat{n}(\mathbf{r}_i)_j\ \partial_j\partial_k\ \varphi(\mathbf{r}_i)$$

so you could consider this a matrix equation in this sense

$$A_{kj} \equiv \partial_j n_k(\mathbf{r}_i)$$

$$n_k(\mathbf{r}_{i+1}) = n_k(\mathbf{r}_i) + ds\ A_{kj}\ \hat{n}(\mathbf{r}_i)_j$$

$$\mathbf{n}(\mathbf{r}_{i+1}) = \mathbf{n}(\mathbf{r}_i) + ds\ A\ \hat{\mathbf{n}}(\mathbf{r}_i)$$

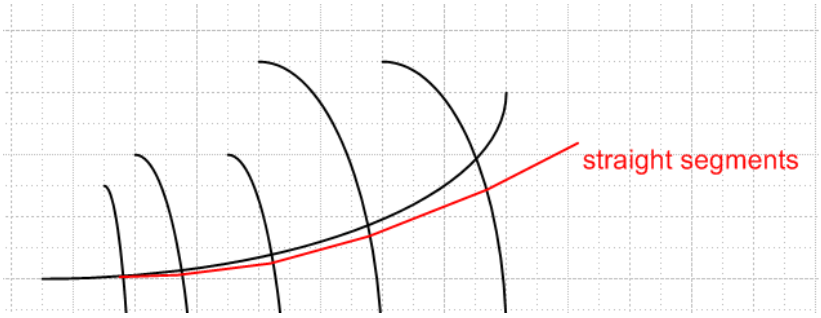
I am claiming that the tensor object $\partial_j \partial_k \varphi(\mathbf{r}_i)$ has no name like "digradient" or some such.

(b) Corrected Notes. Section (a) above is correct, but not really relevant I now realize. The reason is that have a complete formula for computing $\hat{\mathbf{n}}(\mathbf{r})$ at any point in space $\mathbf{r}$ we want, so there is no need to do any kind of iteration with $\mathbf{n}(r)$. I was somehow confusing my ODE solution method with this problem, because in the ODE you have an iterator for the derivative of your function: in one case y'(x), the other case being $\nabla\varphi = \mathbf{n}$ as just a fancier derivative.

Here is the corrected iteration problem. All you need is this

$$\mathbf{r}_{i+1} = \mathbf{r}_i + \hat{\mathbf{n}}(\mathbf{r}_i)\, ds$$

You start at some point, compute $\hat{\mathbf{n}}$ there, then take a small step $d\mathbf{r} = \hat{\mathbf{n}}(\mathbf{r}_i)\, ds$. Then compute $\mathbf{n}$ there, and take another small step. Just keep doing this. If the steps are small, you trace out your electric field lines. If the steps are too large, in each step you wander a bit off your starting field line and start off on some closely adjacent field line. If the true line is black, the red shows what you might get.



So here is a computer program to do the above iteration

```
ds = 0.1          // some small number
for J from 1 to Ncurves do          // do for each curve you want to plot
    r(1) = r_start(J);              // pick your starting point from a pre-made set of them
    for k from 2 to N do            // compute the points along the curve
        n̂ = compute at point r(k-1)
        r(k) = r (k-1) + ds* n̂
    od
    p(J) = create plot structure from the set of r(i) points
od
display all the curves together
```

If we pick ds to be "very small" we might iterate this 100 steps and in that way, we trace out the path of a field line as $\mathbf{r}_0$, $\mathbf{r}_1$, $\mathbf{r}_2$.... We would refer to the continuous field line as $\mathbf{r}(s)$ where s is a distance parameter along the line. We then have Maple plot the sequence of points $\mathbf{r}_i$ and have it connect them with smooth curves or at least straight line segments. In this way, we can generate the unique field line which passes through starting point $\mathbf{r}_0$. We then have to pick some set of starting points to map out our situation, and draw a field line for each starting point. It would be an interesting problem!

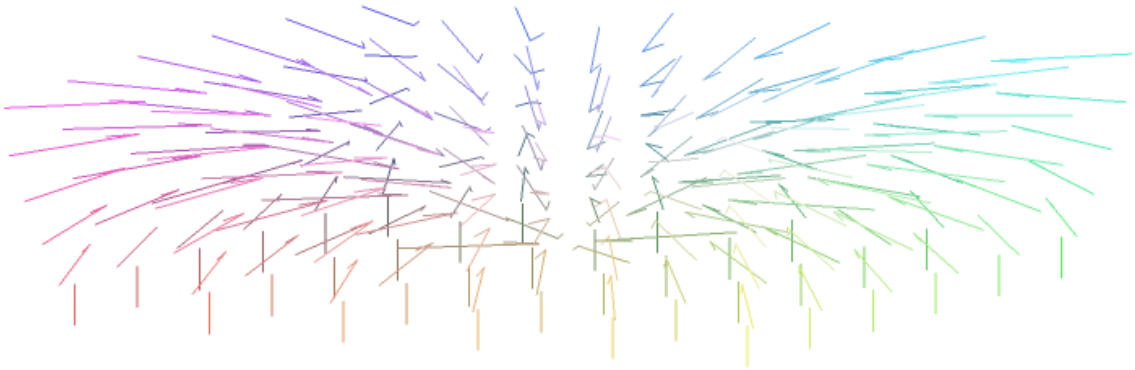
I think the above program could be done in two ways. One way requires making **n** be an actual function of x,y,z and you then just call it **n**(**r**) with the vector argument as needed. Perhaps a simpler way is to have some local x,y,z coordinates inside the inner loop. You would have to do something like this inside the loop

undefine x,y,z; // to allow the gradient to be computed
n = grad(φ,[x,y,z])
x = r[k-1,1]; y = r[k-1,2]; z = r[k-1,2];
n; // at this point, you have n evaluated at point r[k-1]
r(k) = **r** (k-1) + ds* $\hat{\mathbf{n}}$

I will go ahead with the first method because it seems more elegant and uses the D stuff.

4. It is a lot easier to use gradplot3d.

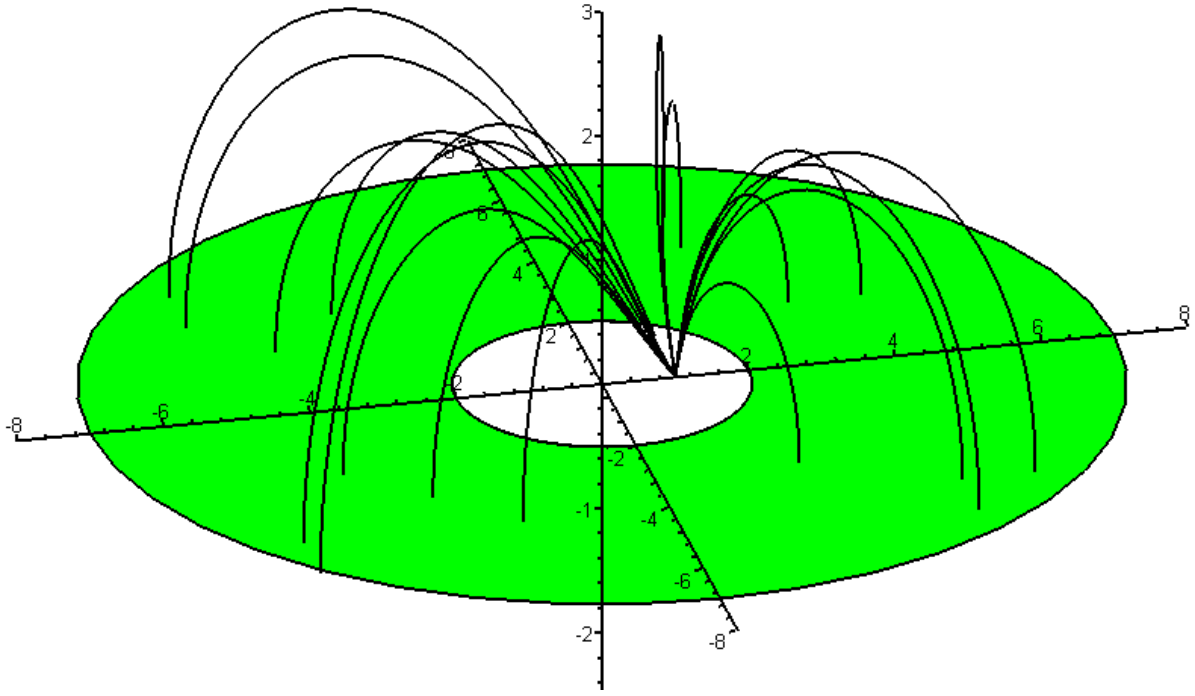
The big problem with gradplot3d is that it makes arrow length be accurate, and then distant arrows are too small to see. When we want to do field lines, we don't care how strong the electric field is along the line, we just want to see where the line is going. You cannot do this with gradplot3d, but you can do it with fieldplot3d, and I did a little of this in "iris plots electric field arrows.mws". Here are some arrows which are supposed to all be the same length which track field lines rising up off an iris (in the iris Green's function problem), and diving into the point charge located in the iris hole



But this kind of plot is just not what you really want. I fiddled a lot and just could not get very happy with the result. If you make a large number of arrows, the picture just turns to mush. You really want those true field lines, and I guess the only way to get them is the iterative method given above.

5. Maple Success

Here is a plot from my implementation of plan 3(b) above:



$\mathbf{r_{k+1}} = \mathbf{r_k} + \hat{\mathbf{n}}(\mathbf{r_k})\ ds$