Fun With Fields

Don Lancaster
Synergetics, Box 809, Thatcher, AZ 85552

http://www.tinaja.com
don@tinaja.com
(928) 428-4073

After all these years, I still have troubles visualizing exactly what **Maxwell's Equations** are and how they **really** work. So, I thought I'd dig back into an ancient fields book (Moon's **Field Theory for Engineers** ferinstance) and see how far we could get by throwing some modern **PostScript** at the problem. And maybe end up with some fundamental field insights and some amazingly powerful results from some astonishingly simple math.

What is a field?

A **field** is some area (2D), volume (3D), epoch (3D+time), or other concept structure in which some measurable parameter assumes some values. More often than not, these values are physically defined, continuously variable, and capable of mathematic analysis. Examples might be the flow of liquid in a pipe; a sponge or concrete drying from the outside in; heat conduction; leakage through a dam; electric voltage and current; a magnetic field; or radio or light em radiation.

At any point in the field the parameter of interest has a **single** measurable value called a **scalar**. Most fields are **continuous** meaning that adjacent measurable values are very close to each other. Although engineers love to draw all sorts of **lines** through fields, it is important to remember that…

> **Most fields are in fact continuous.**
>
> **There are no such things as "field lines".**

For instance, there is no way you can tell if a perfectly uniform magnetic field is stationary or rotating. If these lines existed, some of these might "cut" something else and be detectable. Try rotating one magnet with another for proof. Per **this tutorial**.

Some means is needed to locate the position of a point of interest in a field. This involves choice of a **coordinate system**. Examples of coordinate systems include

Cartesian (x, y, z), **cylindrical** (r, θ, z), **spherical** (r, θ. ϕ), or any of a number of exotic **transformations**. The usual rules of a coordinate system are that there must be **one** variable for **each** dimension and that the variables must be made independent of each other, or **orthogonal**. You usually pick a coordinate system that gives you the easiest math solutions.

Another important rule…

> **The coordinate system used for calculations does not affect the field in any manner.**

There are two factors that uniquely determine the value of any point in any field at any time…

> **DIFFERENTIAL EQUATIONS— Rules for field value change.**
>
> **BOUNDARY CONDITIONS— Unique field edge values.**

Thus, the differential equations give you a **general** solution based upon the expected physical behavior, while the boundary conditions give you a **specific** solution. Examples of boundary conditions might include an impermeable pipe wall and its zero flow rate in **any** direction. Or a perfect electrical conductor that can have no voltage drop across itself.

A regular **differential equation** is simply an expression of what the rules for any **change** are. One very important differential equation is $x = -d^2x/dt^2$.

Which says "the value of x equals minus the rate of change of the rate of change of x, or minus the **second derivative** (or "slope of the slope") of x. One of its solutions is **x = sin(t)**. And, of course, represents a steady state sinewave oscillator or a lossless pendulum.

Differential equations can usually be solved in a number of ways or otherwise approximated. Anything that fits works. The derivative of a sine is a cosine, and the derivitive of a cosine is minus the sine, so minus the second derivative matches what you started out with, balancing and solving the equation.

Fields are typically changing differently in different directions or with time, so you have to go to the more exotic **partial differential equations**. In which you are concerned **only** with how something changes in **one** particular direction or over time. A partial differential equation can be recognized by its "**backwards six**", as we will shortly see.

Any point in a field can only have **one** scalar value at any given time. This might represent the sum of several different activities, such as two or more signals and some background noise.

With fields, we are also likely to be very interested in how things are going to **change** as we leave any given point. We can thus associate a **vector** with a location in a field that points in the direction of maximum change and whose value equals that rate of change.

Such a vector is called a **gradient**…

GRADIENT — The local maximum rate-of-change vector.

$$\nabla\varphi = \frac{\partial\varphi}{\partial x} + \frac{\partial\varphi}{\partial y} + \frac{\partial\varphi}{\partial z}$$

The gradient at any field location will always "point" in the direction of the highest local rate of change. Its absolute value will be the square root of the sums of the squares of the **x**, **y**, and (if 3D) **z** rates of change. Being a vector quantity, the orthogonal **x**, **y**, and **z** change rates apply **only** along their respective axes.

The term $\nabla\varphi$ meaning "gradient of the field phi" is simply an equivalence or a **convenience operator** that greatly simplifies notation for us. One of the problems in reading **Maxwell** is that convenience operators were not invented yet, so there is page upon page of detail tedium to muddle through.

There will always be a direction or a plane or whatever at precisely right angles to any non-zero gradient that will represent a region of **zero** change. This might be called an **equipotential** or an **isotherm** or an **isobar** or some similar name.

A crucial field rule…

Gradients and equipotentials are ALWAYS orthogonal.

Thus, in 2D space, if your gradient points north, there will be no local change in the east west direction. In 3D space, if your gradient points up, there will be no local change in the immediate NESW plane. Which is why you will often see a lot of those **curvilinear squares** in most field sketches. Whose corners **always** are supposed to meet at precisely 90 degrees.

Note in particular that…

Nothing moves or flows or transfers or changes ALONG any equipotential. Changes only occur ACROSS an equipotential.

Nothing moves or flows or transfers or changes ACROSS any gradient. Changes only occur ALONG a gradient.

The Laplacian

Different physical properties will have differing rules for behavior of their fields. Very often these rules can be related to **how fast the changes are allowed to be changing**. When applicable, this property is called the **Laplacian**…

LAPLACIAN — The local rate of change of change vector.

$$\nabla^2\varphi = \frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} + \frac{\partial^2\varphi}{\partial z^2}$$

As with the gradient, the $\nabla^2\varphi$ is also a convenience operator that greatly simplifies notation for us.

Again, **the rules for the Laplacian determine the behavior and all of the physical properties of the field**. A Laplacian of zero widely applies to most electrostatics and to steady state flow of incompressible fluids, heat, or electricity. A Laplacian equal to a constant is called **Poisson's Equation**. A Laplacian equal to a time rate is called a **Diffusion Equation**. Finally, and most importantly, a Laplacian equal to a rate of rate gives us the pair of **Maxwellian wave equations** of electromagnetic radiation.

Let's use a Laplacian of zero and see how far we can get resolving a rather complex field problem with it. We can start off by…

Solving a Zero Laplacian Field

If an incompressible fluid is going to speed up in a chosen direction, it will simultaneously have to slow down in another. Because it is not allowed to "pile up" in any manner. Some intuitive thought or some rather simple differential math can lead us to this astonishingly simple rule…

The value of every point in a zero Laplacian field is simply the average of four adjacent nearby points!

To solve a zero Laplacian field problem, make up a modest sized array of data points. Enter the boundary values in their appropriate positions, and then **guess** what the other values will be. Then replace each non-boundary value with the average of its four adjacent values. Repeat the process a few hundred or a few thousand times, and you will end up very near to a correct solution.

It turns out there is a proof that says **this repeated averaging process will always converge**, although it will take many more repeats for bad guesses. It also turns out that the sequence of averaging does not matter all that much, and a newer

guess can be a mix of adjacent older and newer values. Because a zero Laplacian eventually **must** result.

Simply using **zero** as a bad guess everywhere works just fine and still computes acceptably fast. You can tell how many trips you need by monitoring all of the successive values of a chosen data point. When that point changes less than one in a million every few passes (or meets any other selected criteria), you are done.

If you want to, you can continue for even better accuracy. A differential is **not** a derivative except in the limit. If you have a two dimensional problem, you can double your array size horizontally and vertically by interpolating data points. Then **re-average** a few hundred times. Convergence will be a lot faster since you are starting with far better guesses. It is **much** faster to start with a crude array and then double up because far fewer total trips and many fewer calculations will be needed.

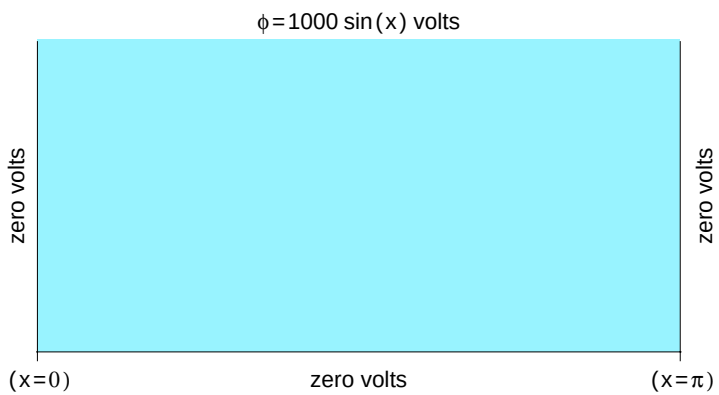
To add frosting to the lily or to gild the cake, a second doubling up can also be done. Which should get you very close to a perfect solution for all points in the field. As a further accuracy check, you can see how much any particular data point field value changes between the first and second doubling. The odds are overwhelming there will be no change.

And we've got a more than good enough solution without every going near hyperbolic trigonometry or anything remotely as fancy.

An Example

All of which begs to get done using **PostScript** with its superb graphics and math capabilities. Let's look at a fairly fancy example. We'll just touch on the highlights here; you can find the detailed code in **FIELDS01.PSL** and its plot in **FIELDS01.PDF**.

Here is the situation…



Assume we have an open trough with perfectly conducting sides and bottom. One-half of a thousand volt sinewave is applied across the top by some means.

Your problem is to accurately show the field and plot some of the gradients and equipotentials. We'll start off with a two-dimensional array that is 30 arrays high by 61 array elements wide…

```
  /field [
[ 0.0000 52.336 ... 998.63 1000.0 998.63 ... 52.336 0.0000 ]
[ 0.0000 0.0000 ... 0.0000 0.0000 0.0000 ... 0.0000 0.0000 ]
[ 0.0000 0.0000 ... 0.0000 0.0000 0.0000 ... 0.0000 0.0000 ]
                ...                         ...
[ 0.0000 0.0000 ... 0.0000 0.0000 0.0000 ... 0.0000 0.0000 ]
[ 0.0000 0.0000 ... 0.0000 0.0000 0.0000 ... 0.0000 0.0000 ]
[ 0.0000 0.0000 ... 0.0000 0.0000 0.0000 ... 0.0000 0.0000 ]
        ] store
```

For this orientation, the position in the array will be the same as the position in the field. We'll ignore the symmetry possibilities since they may cause more grief than they cure.

Our boundary values are shown in red and are **not** allowed to change during averaging. We have (obviously incorrectly) guessed all other tinted potentials within the field to be zero. You then replace each **non-boundary** array value with an average of its four adjacent neighbors, repeating the process many times.

An intermediate 1500 pass result should look something like this…

```
  /field [
[ 0.0000 52.336 ... 998.63 1000.0 998.63 ... 52.336 0.0000 ]
[ 0.0000 49.388 ... 942.38 943.67 942.38 ... 48.388 0.0000 ]
[ 0.0000 46.575 ... 888.71 889.93 888.71 ... 46.575 0.0000 ]
                ...                         ...
[ 0.0000 2.5227 ... 48.141 48.207 48.141 ... 2.5227 0.0000 ]
[ 0.0000 1.2596 ... 24.037 24.076 24.037 ... 1.2596 0.0000 ]
[ 0.0000 0.0000 ... 0.0000 0.0000 0.0000 ... 0.0000 0.0000 ]
        ] store
```

This gives us over 1800 fairly accurate data points and may be good enough for many uses. Since it is fast and easy to improve our results, let's do so. Quadruple the size of the array by inserting a new average value between each existing value. Then insert a new average array between each existing array. This should quadruple the number of data points. Run your average a few hundred times.

If desired, the process can be repeated a second time for a 16X increase in data points. **Points = pixels** is often a good but storage intensive choice.

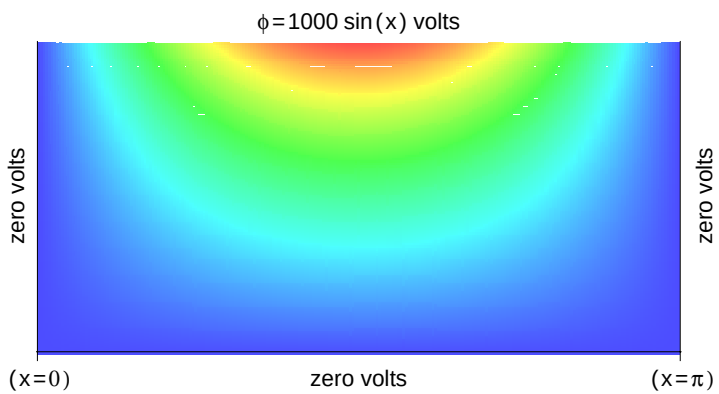
Making it Purty

Our array is now a complete and accurate digital map of our field problem. Should you need values between the array data points, they are easily and linearly interpolated. But since **PostScript** is so good at graphics, lets see if we cannot improve our field visualization.

Our highest voltage is **1000**. We can divide this by **1667** to get a hue of **0.67**. But that would make the "hot" parts of the field blue and the "cold" ones red, so let's reverse things by doing a **1000 sub abs** first. Your field values now should be **0** for hottest red at **1000** volts and **0.667** for coldest blue at **0** volts.

Go through the array and plot a filled unit color unit square for each data point. Should you have **743.66** volts, subtracting **1000** and taking the absolute value should give you **256.33** volts. Dividing by **1667** should yield **0.1537** which, when entered as a **0.1537 0.7 1 sethsbcolor** should give us an orange with slightly eased saturation for less harshness.

Like so…



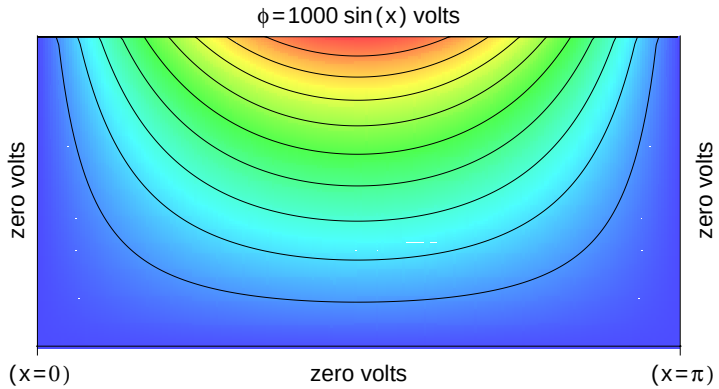
If your plot ends up visually attractive, chances are good you have enough data points and have made enough repeat averaging passes.

Plotting Equipotentials

You can reach down into your field array and extract both equipotentials and gradients. Let's do the equipotentials first. Carefully pick a direction where each equipotential only happens **once**, or in this case **up and down**.

Then go through each vertical column of data to find a pair of values that are just **under** and **over** your current equipotential goal. Interpolate and then do a **lineto** to continue your equipotential line.

Like so…



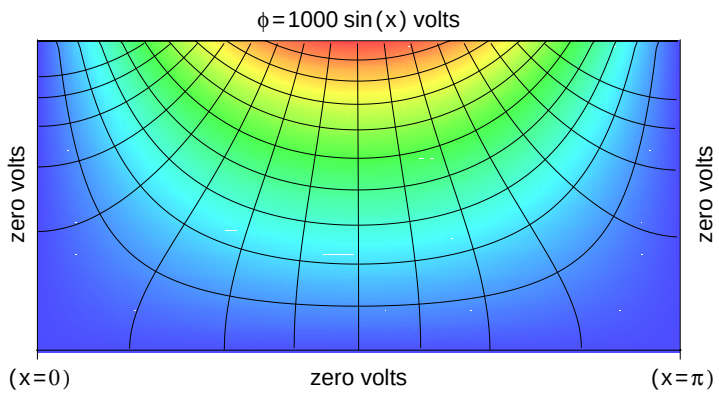
Notice that we are **not** equally spaced down the middle. The further away from the source you go, the further apart the equipotentials. Which is where the nasty hyperbolic trig often rears its ugly head.

Plotting Gradients

Gradients are slightly more subtle but just as easy to plot as the equipotentials. Pick a starting point at the top of the field. Then look **left** and **down** to find how the **current** x and y values **change**. The arctangent or PostScript **atan** operator will give us the direction of the gradient in degrees at this point.

You can convert this to a unit vector **in the array space** simply by using the **sin** for a vertical shift and the **cos** of the angle for a horizontal shift. Continue using the **lineto** operator until you reach the bottom or an edge.

And here is our final plot…



Note that we've worked somewhat backwards from normal, as we have taken the already solved field and derived the gradients and equipotentials from it. Thus some of our curvilinear squares may be somewhat rectangular instead. But they all clearly have the required 90 degree corners and equal diagonals.

For More Help

Consulting services are available per our **Infopack** services and on a contract or an hourly basis. Additional GuruGrams are found **here**, PostScript topics **here**, and math items **here**. Really advanced **PostScript** math problems are found in our **Magic Sinewave** library as well.

Further **GuruGrams** await your ongoing support as a **Synergetics Partner**.