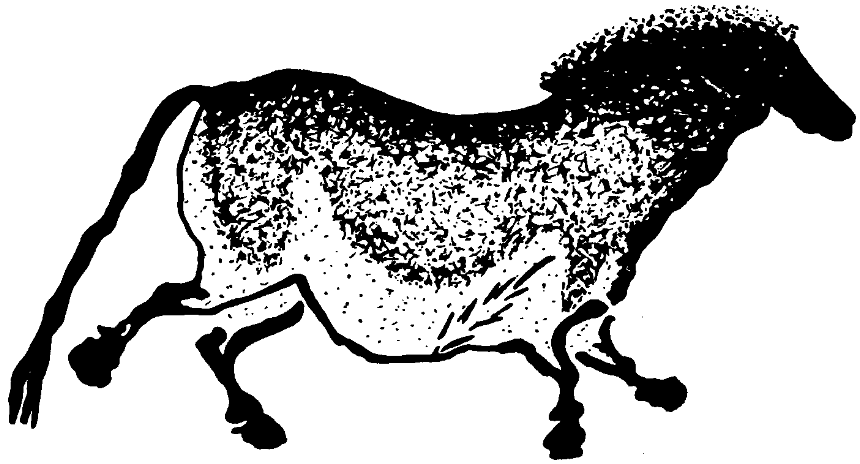


Painted horse 15,000 BC

Lascaux France

This manual, and the accompanying software, are protected by United States and international copyright laws.

Contents

1. Preliminary

Introduction

Fields&Operators is an interactive graphics program that allows you to experiment with curves, surfaces and vector fields in two or three dimensions. Creating graphs, even complex ones with multiple interrelated parts, is simple and intuitive. *No programming is ever required.*

Because curves, surfaces, and vector fields often vary in time, Fields&Operators will let you generate animations to see how your functions evolve. All it takes is a single mouse click.

Part of understanding functions in two or three dimensions is generalizing the one dimensional derivative. In multi-variable calculus this leads to the differential operators Divergence, Gradient, Curl, and Laplacian, as well as vector derivatives and tangent planes. Fields&Operators computes these operators (in closed form) and displays the results as vector fields or surfaces, whichever is appropriate.

New features in Fields&Operators provide tools for exploring the differential geometry of curves: the evolute and osculating circle.

The Touch Panel brings a new level of “hands on” control to mathematical graphics. (Try it!)

You need not know anything about calculus to use and enjoy Fields&Operators. Even without the advanced features there will be enough to delight users with all levels of mathematical experience.

Our intention is to supply a tool for experimentation and, most importantly, for the simple enjoyment of mathematics.

Installing Fields&Operators

Learning to Use Fields&Operators

Although Fields&Operators is a powerful program with many features, we hope that you find its command structure both convenient and natural. After you spend some time with the Guided Tour in the next section, you should be quite comfortable with Fields&Operators.

Fields&Operators includes an extensive Help facility. There you can easily find answers to most of your questions about Fields&Operators. There are many places in this manual that we refer you to Help.

There are many sample graphs on your disk. After you take the Guided Tour, browse through the sample files. Many of these files include brief notes in the User Notes window. Much of the time, you will find that the graphs you want to create are variations on some of the included samples.

Technical Support

If you have any questions, please feel free to call or e-mail. Don't wait until you have a problem to call; we are always eager to hear your comments and suggestions.

(520) 749-0200

support@LascauxSoftware.com

Lascaux Graphics
5354 N. Buckhorn Dr.
Tucson, AZ 85750
USA

http://www.LascauxSoftware.com

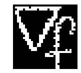
2. Guided Tour

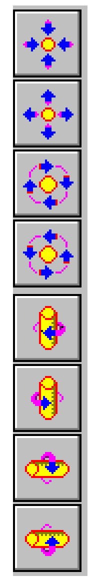
This guided tour is intended to introduce you to the major features of Fields&Operators. Follow along at the computer and feel free to experiment.

The tour will start with a quick look at some features and examples to give you an overview. After that we will look at several specific examples, with step-by-step instructions.

Starting Fields& Operators

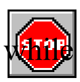
You can start Fields&Operators from the Start Menu.





The graph that appears when you start Fields&Operators depends on how you have configured the startup default. (More on this later.) If you have not made any changes, you will be looking down along the z-axis onto the graph of *z=sin(x+t)*cos(y+t)*. From this point of view the graph is not very interesting.

Click one of the rotation buttons at the left of the window. Hold the mouse down in a button for continuous rotation. You can “lock” the rotation by holding the q key when you click on a rotation button. The graph will continue to rotate in that direction even when you release the mouse. Click

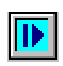


to Stop (but you may want to leave the rotation on while you try time-animation next).

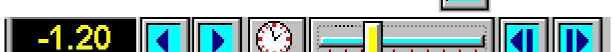
Animation

Animation is one of the most powerful features of Fields&Operators. For now, you should know that both *real time* and *recorded* animation are possible. You will use recorded animation for graphs that take long to draw.

Since the definition of the graph we are working with already contains the *t* variable, you need only click



on the top control bar to start the animation.



There are lots of buttons and controls in this window. For a quick reminder, point to a control with the mouse (without clicking). A "hint" will appear at the bottom right of the main window. After a few seconds, a hint will also appear at the mouse position.

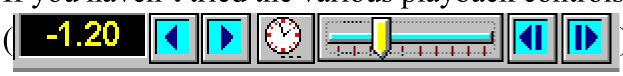
To get an idea of the kinds of graphs you will be able to create with Fields&Operators, let’s take a look at some of the sample graphs that were copied to your hard disk when you installed the program.

There are three ways of opening files. Fields&Operators remembers the latest 4 files you have opened. These will be listed at the bottom of the File menu. There is also a collection of files that will be useful starting points when you make you own graphs. These are available on the QUICK START sub-menu of the File menu. For now, use File-Open and select the file DRUM from the Examples folder.

When you open this file, you will be notified that notes have been attached. To open the Notes window, select it from the View menu. (The notes window is a simple text editor for you to leave notes that are saved as part of the file.)

Recorded Animation

The note for this file informs you that this graph is defined for animation. For this tour, we have purposefully chosen a graph that is slow to draw and will give best results with recorded animation. Select RECORD ANIMATION from the Animation menu. You will be asked to enter a file name. After that Fields&Operators will draw and store a number of frames for the animation. When it is finished, Select PLAY ANIMATION from the Animation menu.

If you haven't tried the various playback controls (

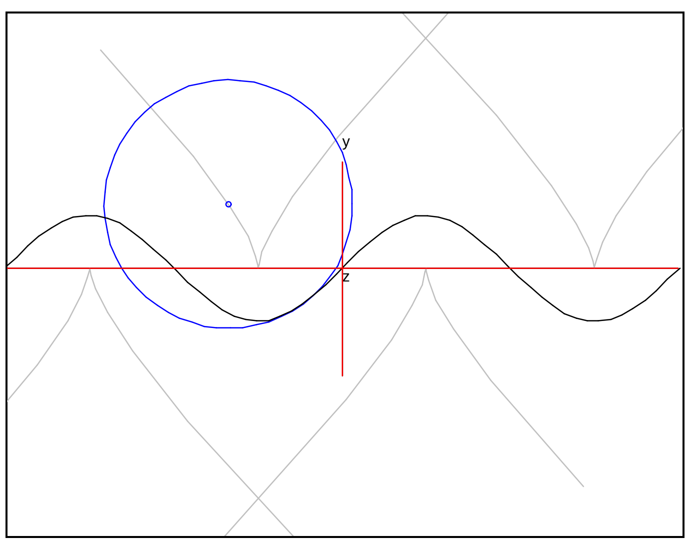
), try them now. Remember that each control has a hint associated with it.

When you are done with this recorded animation, click Close Playback.

Later we will look at the Story Board. This will give you much more control over you animations.

Touch Panel

The next sample graph will introduce the Touch Panel. Open the file TUCHDEMO [Touch Panel Demo]. This file is also available as DIFFERENTIAL GEOMETRY under Quick Start on the File menu.



The graph shows a sine curve, its evolute and the osculating circle at a point. (If you are not familiar with the language of differential geometry keep going anyway. There is a brief description in Help.)

The position at which the osculating circle rests could have been managed with animation, but instead it has been defined to work with the Touch panel. (The file OSCULAT1 [Curve with Osculating Circle 1] contains the animated version.) Click Touch. The simplest way to learn about the purpose of the Touch Panel is to try it. Click on the slider and drag it.

If we were working with a two dimensional surface rather than a one dimensional curve, the Touch Panel would have given two dimensional dragging.

Drawing Quality - Shaded, Solid, and Wire Frame

The Draw Quality menu allows you to select Shaded, Solid, or Wire Frame (transparent) graphs. On some systems there may be a difference in speed among these three styles. (Shaded is slowest; Wire Frame fastest.) Of course, these settings only have an effect with surfaces. If your graph has only curves or vectors, shaded or solid surfaces are meaningless.

With shading you can specify up to 3 light sources (via the Light Sources dialog). For details, see Help.

If you work with very complicated graphs, with many subdivisions in the domain, drawing a graph can take several seconds. This can seem painfully slow while you are rotating the graph, or making other changes. In that case, try using FAST WIRE FRAME which limits the number of points in the graph, overriding your setting in the domain dialogs. You can set an independent Draw Quality for the Draw button. To set the quality for the Draw button, click it with the Secondary (right) mouse button.

Typically you will set the best quality for Draw, and use it on demand.

Exploring on Your Own

Now we ask you to spend some time examining the other example files on your own. But first you should be aware of the general purpose of the main program dialogs.

Coordinates	Allows you to select Rectangular, Cylindrical, or Spherical coordinates.
Layers	Each graph is composed of one or more layers. The Layers dialog controls the mathematical definition and appearance of the layers. *There is much to explore here.*
Domains	A Domain is an array of points distributed in one, two, or three dimensions. Calculations for each layer can depend directly on values from the shared domain, or from a private domain, or from other layers which, in turn, depend on a domain.
Time and Animation	Sets the range of values for *t* during animation. Open the dialog by clicking .
Story Board	Lets you play the film director in controlling your animations.
Layer Summary	Offers a summary of the mathematical definitions in each layer of your graph.
Axes	For defining the appearance of coordinate axes.

Open each of the files in the Examples directory. Most have short notes attached. Feel free to experiment and make changes. Save any changes you make with a different file name so as not to overwrite the examples.

As you explore each graph, open the Layer Summary window.

The tour will continue with a step-by-step walk through in building a number of graphs.

A Surface and Its Gradient

In this part we will begin our detailed look at Fields&Operators by drawing a number of interesting graphs. In passing we will touch all of the major program features and controls.

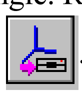
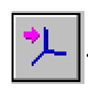
We will focus on the graph of a surface and its gradient. (Even if you are not yet familiar with vector calculus, keep reading.)

We will work with surfaces where *z* is a function of *x* and *y*. $z=F(x,y)$. Other surfaces (where *x, y* and *z* are all functions of other variables) are equally easy to draw. Many examples appear in other sections.

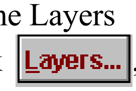
For functions of the form $z=F(x,y)$, the gradient is a vector field in the *x-y* plane. At each point it indicates the direction in which the function changes fastest. (The computation is described in Help.)

Select RESET from the File menu. This will return to the default graph of $sin(x+t)*cos(y+t)$ that we started with. (You can control the graph that appears when you select RESET. See Help.)

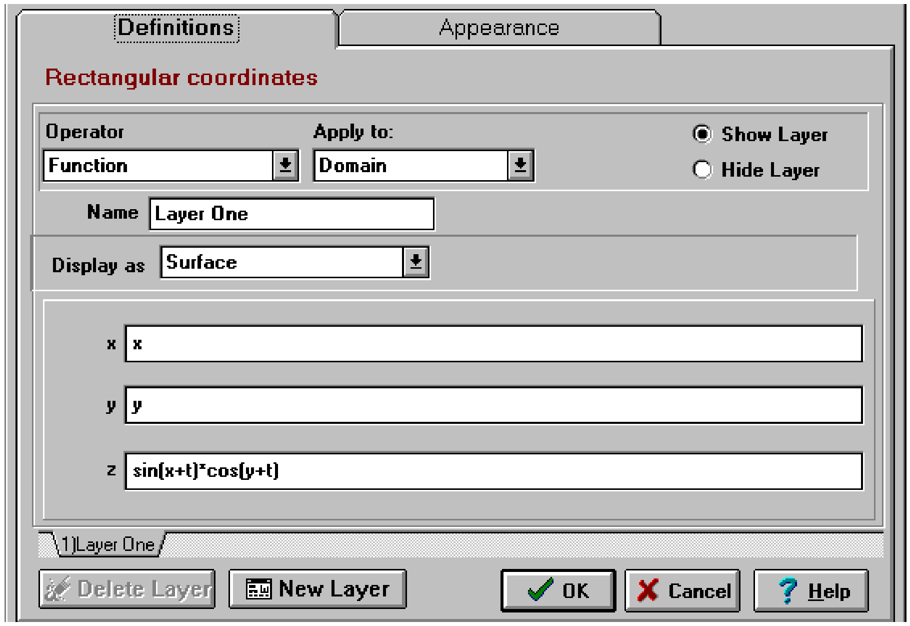
Preferred Viewing Angle

It can become tedious to always start off looking down on the *x-y* plane. Buttons on the rotation palette allow you to specify a preferred viewing angle. Rotate the graph in the usual way to a convenient angle, then click . This will save the angle to your disk. Now, anytime you want to return to this preferred angle, just click .

Layer Dialog

All graphs consist of one or more layers, defined through the Layers dialog. There are several ways of opening this dialog. Click 

, or click in the main program window with the *secondary* (right) mouse button. This is the Pop-Up menu. Select LAYERS from this menu.



The dialog has two pages, Definitions and Appearance. The tabs at the top flip the pages.

As you might expect, this is where you enter the functions that define the layer. Notice that the OPERATOR is set to "Function". Layer 1 is special in that entering functions is the *only* operator available. We will soon see the power that becomes available when you can select an operator.

Coordinate Functions

The dialog shows that the three coordinate functions are currently defined as:

$$F_x = x$$

$$F_y = y$$

$$F_z = sin(x+t)*cos(y+t)$$

Notice that in this case x and y from the domain are not changed. This is the standard format for functions of the form $z=F(x,y)$.

If we had selected Cylindrical or Spherical coordinates, different variables would be used.

Display As

Using the DISPLAY AS drop-down list, you can select if your graph is to be drawn as a surface, vector field, or level curves. (There are restrictions on the use of level curves. See Help. Also, level curves are usually slow to draw because of the extra computations required.)

The choices available here depend on the Operator. Later, for example, we will use the Gradient operator. Since the gradient is a vector field, no other choices would be available.

Feel free to experiment with this setting. You will have to close the Layers dialog to see the results. Don't forget to rotate the graph as needed.

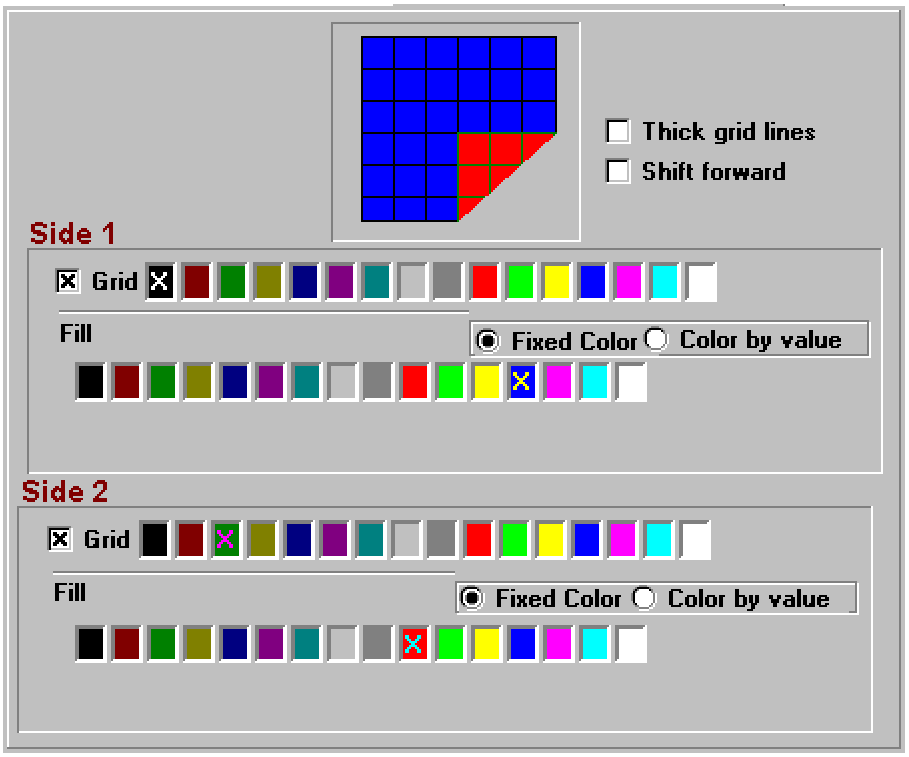
Return to the surface display to continue with this tour.

Apply To

The APPLY TO drop-down list allows you to compose functions by using the output from one layer as the input for another. (We currently have only one layer, so the only choice for the source is the domain.)

Appearance Page

Click on the tab to turn to the Appearance page for layer 1.



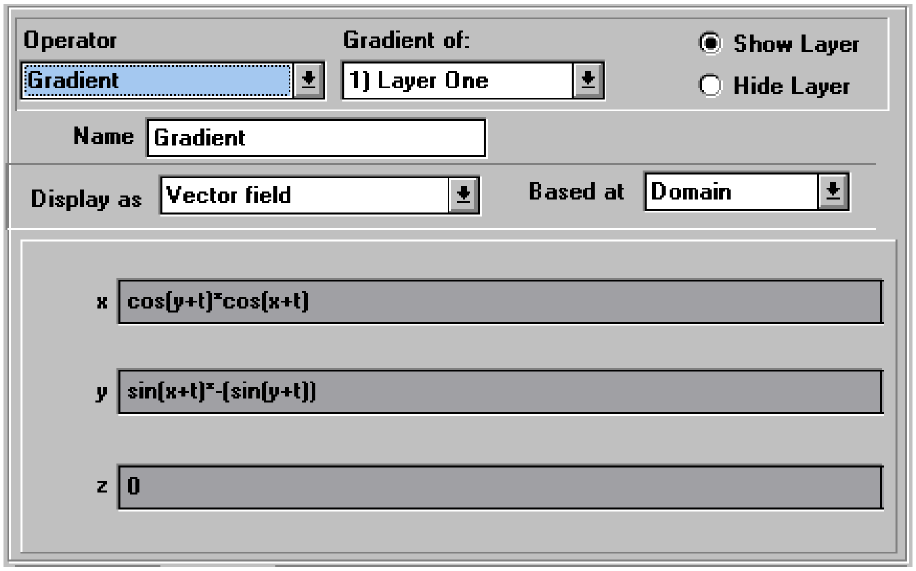
A surface has 2 sides. (Mathematically this is not strictly be true. Can you give an example? Can you draw one? See the KLEIN [Klein bottle] example.) You can set the fill and grid colors for each side here. There are several other options; most will be discussed below. For full details turn to Help.

Return to the Definitions page.

A New Layer

Now we will add the gradient to the graph. Click New Layer. Initially, the new layer is a duplicate of the previous layer, except for its name.

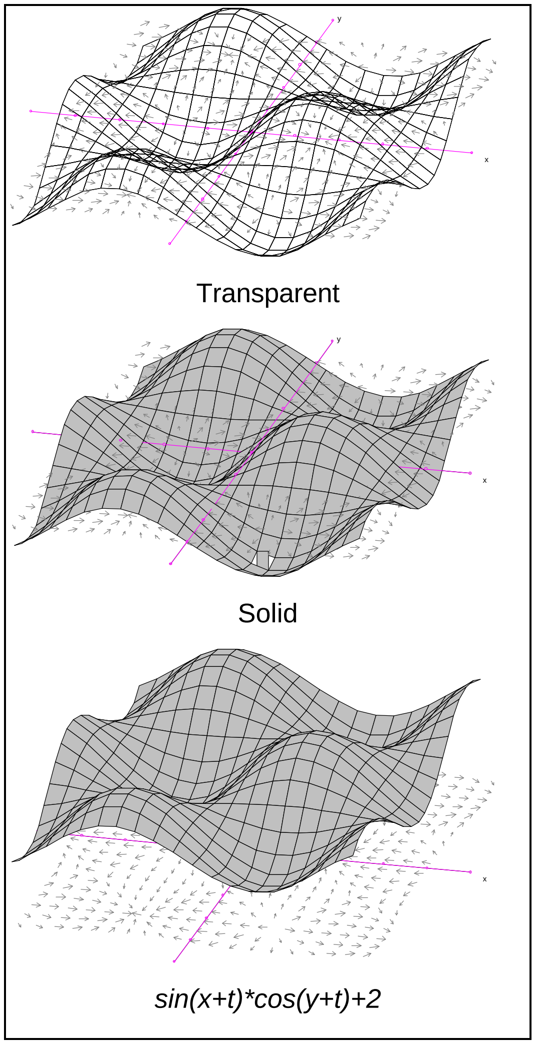
We will change the functions of the new layer to reflect the gradient of layer 1. Select Gradient from the OPERATOR drop-down list.



Fields&Operators will automatically compute new functions. The function definitions cannot be modified. (If you do want to edit the computed functions, change the Operator to FUNCTION.)

Notice that the selection in the DISPLAY AS drop-down list was automatically changed to VECTOR FIELD. (The gradient is applied to a surface (scalar field) and produces a vector field.)

Because the display is a vector field, an additional drop-down list, BASED AT, appears for specifying the placement of the arrows of the field. In this case we want the field to be arrayed throughout the domain, so the default choice is correct.



Close the dialog to draw the newly defined graph. The vector at each point indicates the direction, in the domain, in which the function values change fastest.

Try both solid and wire frame drawing (on the Draw Quality menu). Rotate the graph with each setting.

Don’t forget that this surface, and hence its gradient also, are time dependent. Earlier we saw how to use real time and recorded animation. Watch how the field varies with the surface.

The vector field lies in the *x-y* plane. In this case the field intersects the surface. A simple trick separates the two layers. Change the definition of the surface in layer 1 to *sin(x+t)*cos(y+t)+2.* Adding 2 raises the surface but does not effect the gradient.

Later we will discuss controlling the number of vectors that are drawn. For now, you should take a look at the different controls that become available on the Appearance page of the Layers dialog when the layer is a vector field. Be especially sure to look into vector scaling. (Click Help on the appearance page.)

Save Your Work

Before going on, let's save the current graph in a disk file. In fact, it's a good idea to save work in progress regularly. In the unlikely event of a computer crash, you can at least resume with your last saved version.

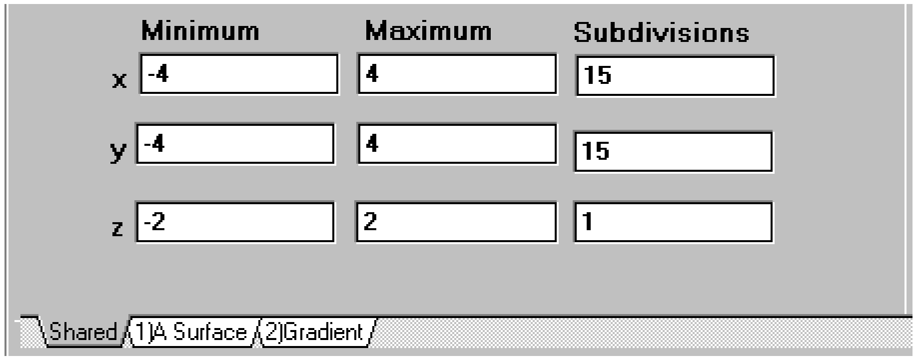
Choose SAVE from the File menu.

The four files you have most recently accessed will appear at the bottom of the File menu for quick retrieval.

Domains

As you know, the trigonometric functions are periodic. Let's look at two full periods by changing the domain. Select DOMAIN from the view menu or click 

.



The tabs at the bottom of the dialog indicate that we are now defining the shared domain. This is the domain accessed by all layers unless a layer specifically requests a private domain. We will look at private domains below.

Notice that the *x* and *y* values of the domain vary between *-4* and *+4*. Change the minimum *x* value to *-2*pi* and the maximum *x* value to *2*pi*. Do the same for the minimum and maximum *y*-coordinates.

> This example illustrates an important feature. The expressions for the minimum and maximum values can include functions; even time dependent ones. The variables *x*, *y*, and *z* (in rectangular coordinates) may *not* be used here. For example, a domain can be made to shrink and expand by entering Minimum: *-2-sin(t),* Maximum: *2+sin(t)*.

Since the *z*-coordinate does not currently appear in any of the function definitions on the Layer dialog, the minimum and maximum *z* values are ignored. However, be sure that the domain has exactly one *z*-subdivision.

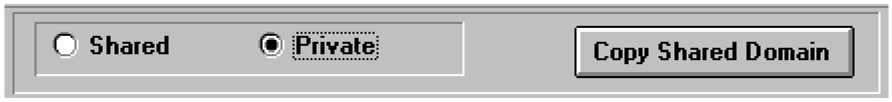
Also experiment with increasing the number of subdivisions in the *x* and *y* dimensions. Higher values give smoother graphs.

A Separate Domain

There are so many vectors in the field that the graph appears cluttered. We could go to the shared domain and reduce the number of subdivisions, but then the surface would not be as smooth. The solution is to introduce a private domain.

On the domain dialog, click on the tab for layer 2.

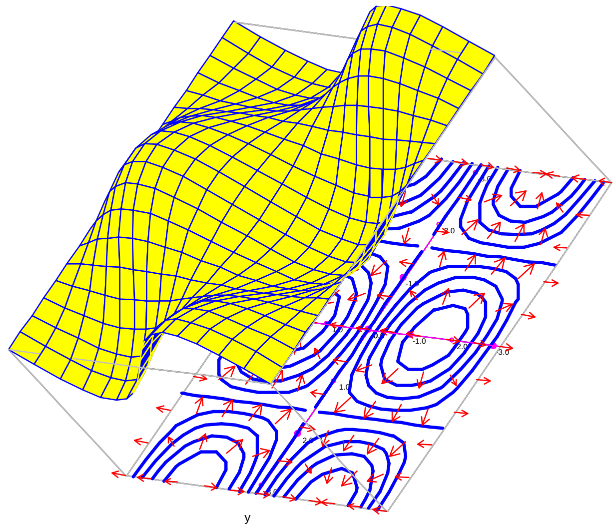
We see that currently this layer is using the shared domain. To obtain a private domain for this layer, click PRIVATE.



To make sure that the minimum and maximum size of this layer match the underlying vector field, begin by making a copy of the shared domain. Click COPY SHARED DOMAIN. Finally, change the number of *x* and *y* subdivisions to 11.

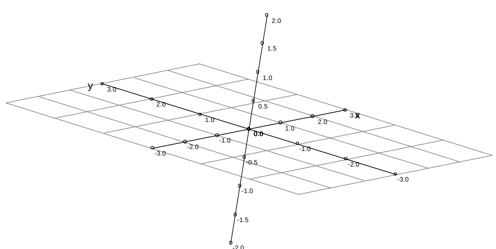
Exercise

Try viewing the surface as Level curves. Better yet, add a new layer with the same definitions as the surface and display it as level curves, keeping the original surface and vector field in the graph as well.



This graph is stored in the file SURFGRAD [Surface With Gradient].

Axes



The coordinate axes help in visualizing the orientation of a graph in three dimensions and in judging its size. Fields&Operators lets you easily control many aspects of the axes appearance.

To open the Axes dialog, select AXES from the View menu or the pop-up menu. Experiment with the settings.

Changing and Updating Functions

Now that you have the domain and viewing angle and other parameters adjusted to give an interesting view of a function and its gradient; you may want to investigate a different function and its gradient, retaining the same design. Save the current graph before making any changes.

We will change the function in layer 1.

Enter the function $F_z = 1/(1+x^\wedge 2+y^\wedge 2)$ in layer 1. Change the name of the layer if you wish. No other changes are necessary. The gradient in layer 2 will automatically be recomputed for this function.

Quick Start Menu

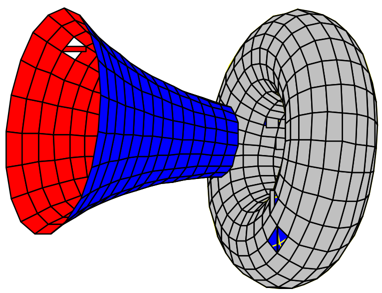
If you develop a way of looking at functions that you would like to use for many different functions, consider placing your graph in the Quick Start menu. The files listed in Quick Start are ordinary Fields&Operators files, but access to them is easy. The idea is for these graphs to serve as templates for constructing graphs with a specific configuration.

Open the Quick Start sub-menu from the File menu and select ADD CURRENT FILE TO QUICK START.

When Fields&Operators was installed, a separate directory was created for Quick Start files. You may first save the file in this directory (or not, as you wish).

Story Board

The Story Board gives you the control of a film director in preparing animations. You can preview your animation; modify the rotation and scaling of any frame; delete frames; and append individual frames or complete sequences.



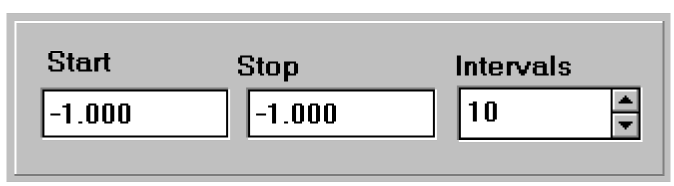
In a later section we will discuss the mathematics behind the animation in which the horn is made to pass though the torus in the example THROUGH [Horn Through Torus]. Here we will simply use that file to demonstrate the Story Board. Open that file now and animate it to see the action before we manipulate it.

Our animation will first rotate the combined figure without the horn moving. After 10 frames we will start the horn moving and change the direction of the rotation.

Rotation Per Frame

Before we even open the Story Board, let's set the amount of rotation per frame. This effects the rotation buttons in the main window as well as the story board. Open the ROTATION INCREMENT dialog from the Environment menu. To have the graph rotate 180° (π radians) in a span of 10 frames, enter an increment of $2\pi/20$.

The horn is made to move by adding *t* to its position. Since we want to hold the horn stationary for the first 10 frames we want to make sure *t* does *not* vary. Open the Time dialog (click ). Set its values as shown.



We are finally ready to open the Story Board. Click 

, or select from the SHOW STORY BOARD from the Story Board menu.

Adding Frames

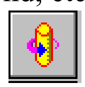
When you open the Story Board window it is initially blank. Click APPEND FROM ANIMATION SETTINGS. (The same option is also available on the Story Board menu.) The 10 frames, all with the same value of *t* and the same viewing angle, will appear.

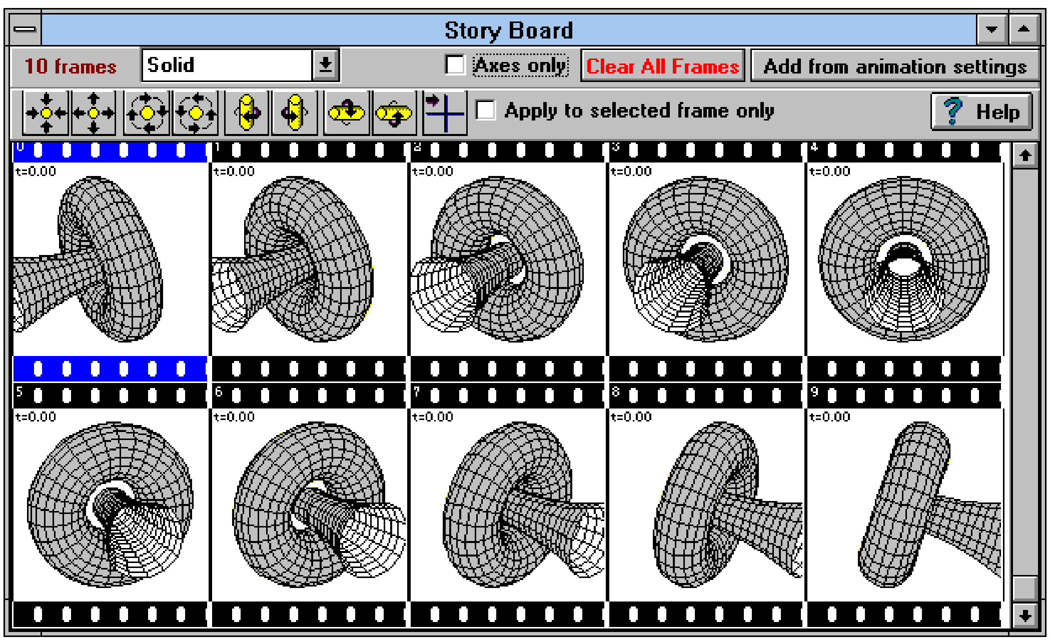
Rotation in the Story Board

Let's rotate all frames starting at the first frame. Click on the first frame to select it. Now click one of the rotation or zoom buttons on the top of the window. These work as they do in the main window. All frames starting at the selected one will rotate in the direction you selected.

If drawing all these frames is too slow on your processor, you can select FAST WIRE FRAME from the Draw Quality drop-down list. For even greater speed, you can check AXES ONLY. This will show only the orientation and size of your graph and so will be very fast. (Recall that axes are defined on the Axes dialog.)

Rotate the image in frame 1 so that it, more or less, matches the orientation in the illustration above. Notice that all frames rotated by the same amount. *This is not the effect we want for animation.*

For animated rotation, we want the second frame to rotate more than the first, and the third more than the second, etc. Make sure frame 1 is selected, and then hold q while you click  one time. Each frame should rotate the way we want.

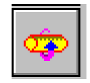
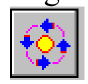


We are now ready to preview our progress so far. Click PLAY FROM STORY BOARD on the Story Board menu. If the drawing in the main window is too slow, you can create a recorded animation instead by selecting RECORD FROM STORY BOARD.

Now would be a good time to save the graph. Select SAVE AS… and give it a new file name.

For the next 20 frames, we do want *t* to vary. Open the Time dialog and let *t* vary from *-1* to *3* in 20 intervals. Click APPEND FROM

ANIMATION SETTINGS, and open the Story Board again to see the total of 30 frames. In the second half the horn should appear to move.

Click in frame 11 to select it. Hold q while you click  one time. Now we will introduce an abrupt change of direction at frame 20. Select frame 20 and click and hold  *without* q.

Play or record the animation from this story board.

Other Options

There are a few Story Board options we haven't looked at here. The Story Board menu has an option for adding a single frame from the main window to the Story Board.

If you click a frame with the secondary (right) mouse button, a menu pops up with additional options. See Help for details.

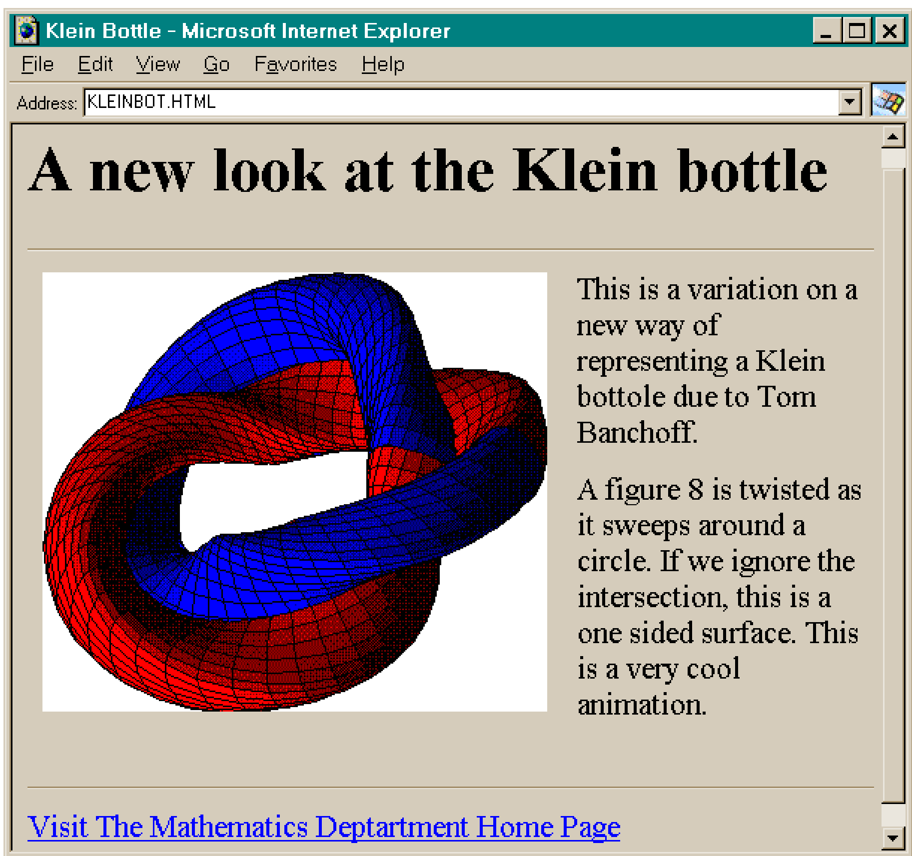
Changing Definitions

The Story Board manages the rotation, scale, and time settings for each frame. It does not store the underlying definitions as set on the Layers dialog. If you were to change the domain and layer definitions (without resetting the Story Board), the sequence of actions would be applied to the new set of functions.

Publish Your Graphs on the World Wide Web

Publishing your graphs on the World Wide Web couldn't be easier. Select SAVE AS HTML from the EXPORT sub-menu on the File menu. This will build and save a Web page consisting of your graph along with the text from the Notes window. The graph will be saved in GIF format in a separate file, and a reference to that GIF file will be included in the Web page.

You will, of course, need a Web browser to see the page.



If you are familiar with HTML (HyperText Markup Language) you can customize the page using any text editor.

Sorry, we cannot give guidance here on posting the page for public, or private, access. Your University or local computer user's group can help.

If you would like to make the graphs you create widely available, please contact us about posting them on our WWW site.

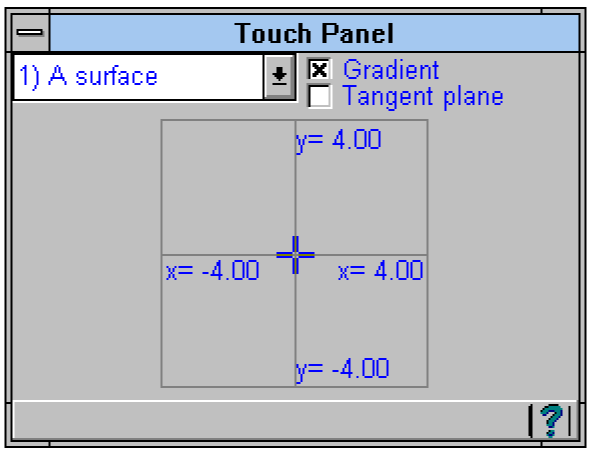
Gradients with the Touch Panel

In this section we present an alternative method for adding a gradient field to a surface.

Select RESET from the File menu, or load any graph defining a surface where *z=F(x,y)*. Rotate the graph to a convenient viewing angle.

Next click  to open the Touch Panel.

The panel represents the domain. As you move the mouse in the Touch Panel, its coordinates in the domain are displayed in the corner of the Touch Panel window, and the coordinates, in 3 dimensions, of the image of that point are shown in the main graph window.



Check GRADIENT or TANGENT PLANE (or both). A new layer will be added for the operator you selected. Now when you move within the touch panel the field or plane will follow your touch on the surface.

For more on the Touch Panel, see Help.

Other Uses of the Touch Panel

Having Fields&Operators add layers automatically is just one way to use the Touch Panel. You may include the expressions *mx* or *my* in any expression in the domain dialog or layer function definitions. These will be interpreted as the *x* and *y* coordinates of the point in the domain pointed to by the mouse.

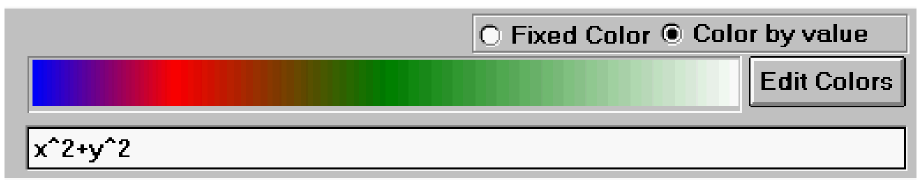
For example, if you have a fast processor, try $F_z = mx*x\char`^2+my*y\char`^2$.

Coloring Surfaces with Functions

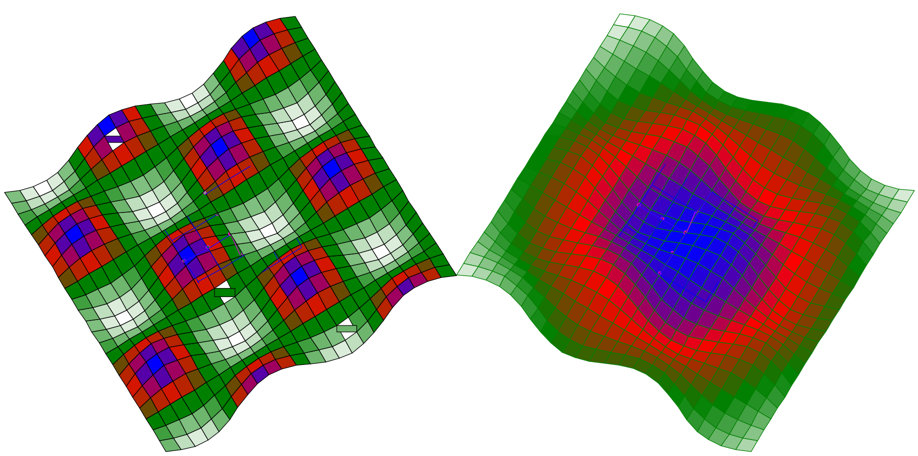
We will next look at a powerful variation of the simple shading we've already seen. Make sure the Draw Quality is set to SHADED before continuing.

Select RESET from the File menu so you will be working with the surface z = *sin(x+t)*cos(y+t).*

Open the Appearance page of the Layer dialog. For Side 1, click COLOR BY VALUE. A continuous color bar appears, as well as an edit box for you to enter a function. The range of colors on the color bar will be mapped to the values of the function you enter. For example, the leftmost color on the bar will be used for points where the minimum function value is found, the rightmost color is used for points where the maximum function value is found. (You can edit the color distribution by clicking EDIT COLORS.)



To illustrate this feature we will color each side of the surface differently. For Side 1, enter the function *sin(x+t)*cos(y+t)* which is the function that defines the surface. For Side 2, enter *x^2+y^2*, to draw a circular pattern. Close the dialog and click  to start animation. If the action is too slow, try a recorded animation.



The technique we used for the circular pattern shows that the coloring needed not be related to the defining function of the surface. Also, the functions may be time dependent.

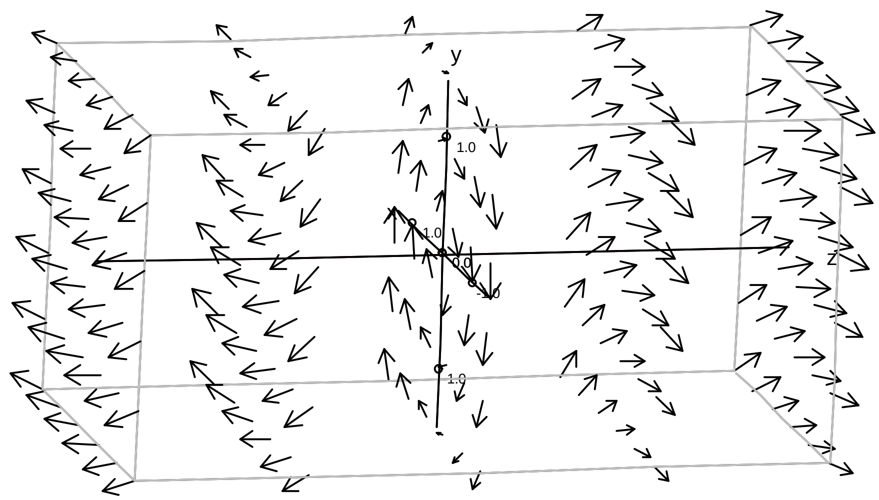
Note: your color depth setting (typically 8 bit (256 colors), 16 bit, or 32 bit) influences the appearance of continuous colors. Because this is Windows (or video driver) configuration, we will not advise you here.

Vector Fields and Integral Flows

An integral flow represents a numerical solution to the differential equation given by a vector field. It describes how a particle would move under the influence of the vector field.

Each point in the domain is an initial point for a solution. Solutions are generated by following the direction of the field at each point.

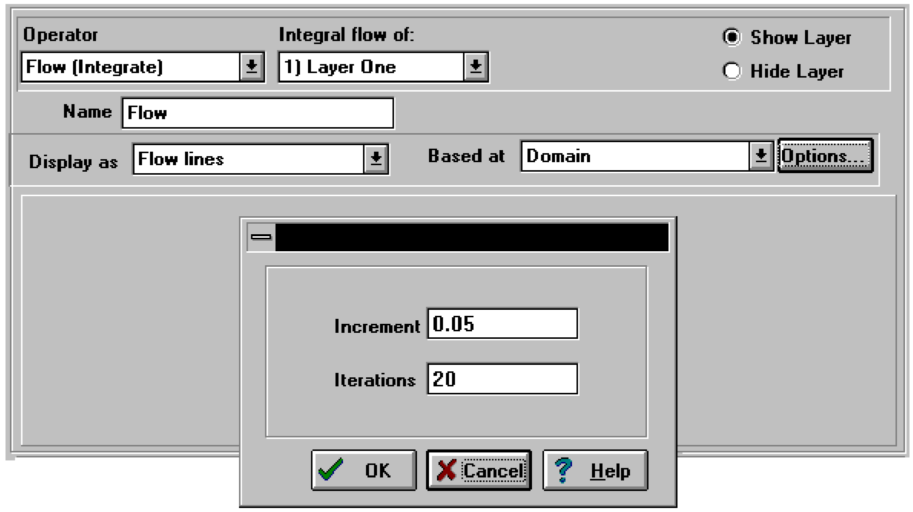
To generate an integral flow we will begin with the sample graph called 3DFIELD [3D Field].



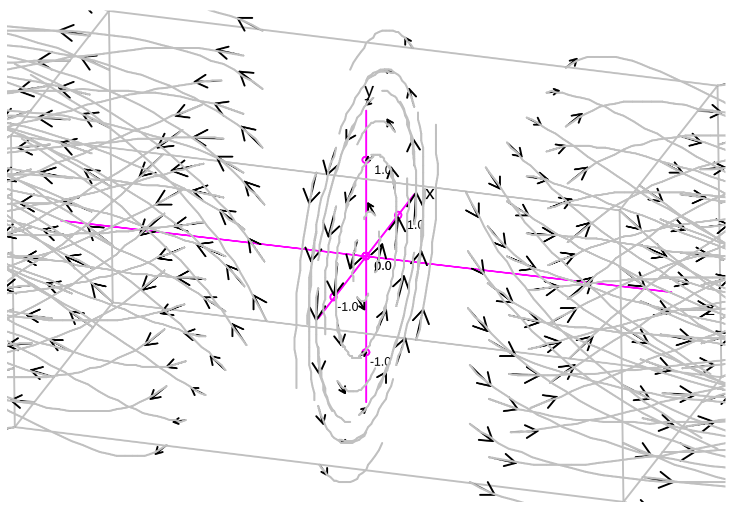
This is the first time we have used a three dimensional domain. You may want to open the Domain dialog to see how it is defined. Although this field appears as several "slices" it is defined in a single layer.

Next add a new layer and choose the Flow operator. The integration is performed numerically; essentially as a Riemann sum. The two parameters that control the summation are entered on the Flow Parameters dialog, The dialog is accessed through the OPTIONS button which appears on the Layer dialog whenever the Flow operator is selected. Open that dialog and enter the values show below.

Smaller values of the increment give more accurate results, but will slow computation..

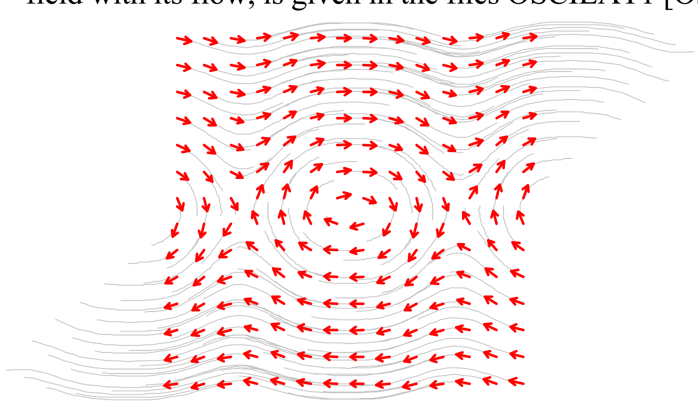


Make sure that the BASED AT control is set to the domain. This determines where each flow line begins.



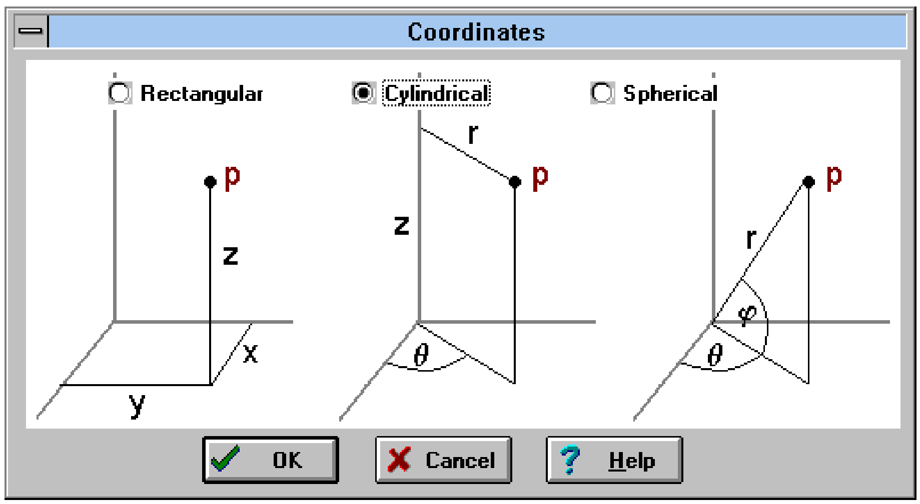
A Related Example

An example representing a two dimensional time dependent vector field with its flow, is given in the files OSCILAT1 [Oscillator]. It is a classic example from the theory of differential equations showing the phase space of a simple pendulum oscillator.



Polar and Cylindrical Coordinates

Fields&Operators can work with three types of coordinate systems: rectangular, cylindrical, and spherical. You select the coordinate system from the Coordinates dialog.



r and *q* are used differently in cylindrical and spherical coordinates.

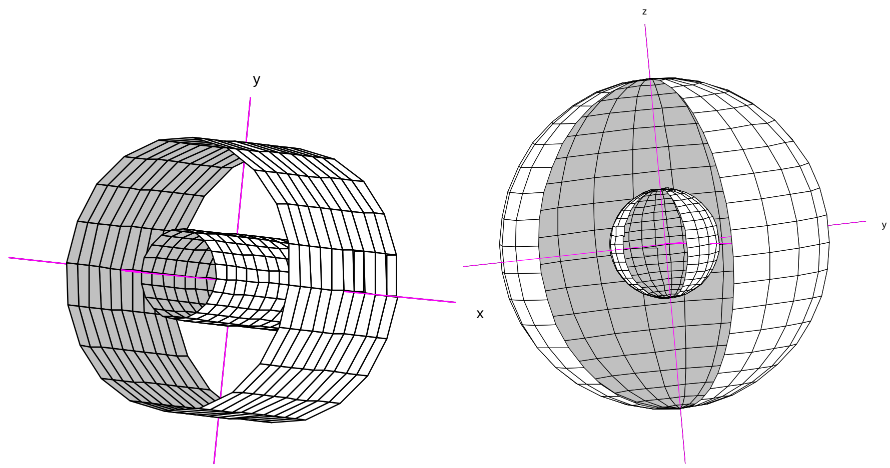
When you change coordinate systems, Fields&Operators will reset all domain, layer, and other definitions to their default values. For this reason, you will be asked to confirm your choice when you change coordinates.

Your choice of coordinate system determines the allowable variables in function definitions on the Layer dialogs. For example, if you are working with spherical coordinates, you cannot use the variables *x*, *y*, or *z*. This would lead to an error message.

When entering functions, you must type out *theta* or *phi*. You cannot enter Greek letters.

Identity Functions

A useful way of thinking about domains in cylindrical or spherical coordinates is to consider the identity functions (These are the defaults in the layer definitions.)



The identity functions with cylindrical and spherical coordinates

In each of these figures, we have defined two *r* subdivisions. The default is one *r*-subdivision giving one cylinder or one sphere.

For the double sphere in the illustration, we have changed the maximum f in the domain to *5*, rather than the default of *2*pi*. This allows us to look inside the spheres.

Operators When you use cylindrical or spherical coordinates the following operators are not available on the Operator pop-up menu on the Layer dialogs:

Dot product	Evolute
Cross product	Osculating Circle
Tangent Plane	Frenet Frame

3. Examples

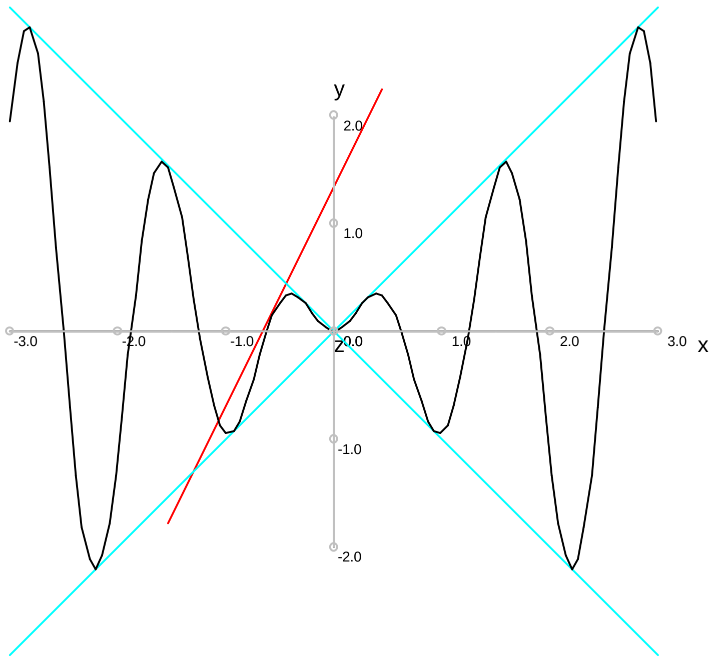
Graphs and Dimension

The first consideration in drawing a graph, (and thinking about functions, and especially their derivatives) is dimension.

Informally, a function is a computation that takes one or more inputs and produces one or more outputs. The set of possible inputs is called the Domain. The number of inputs to the computation is the dimension of the domain. The output of the function is the Range, and the number of outputs is the dimension of the range.

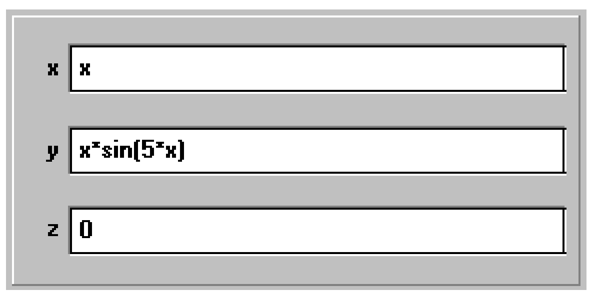
Technically, the graph of a function is the set of "tuples" of the form (in_1, in_2, ... in_n, out_1, out_2, ... out_m) and so the dimension in which we draw the graph is the sum of the dimensions of the domain and range. Fields&Operators can help visualize cases with up to three dimensions in each of the domain and range. (Including animation can effectively add yet another dimension.)

How can we visualize high dimensions in a three dimensional world (or a two dimensional computer screen)? We will illustrate various combinations of dimension through some examples. These examples are stored on your disk. Besides illustrating this discussion, they are useful as starting points for your own graphs.



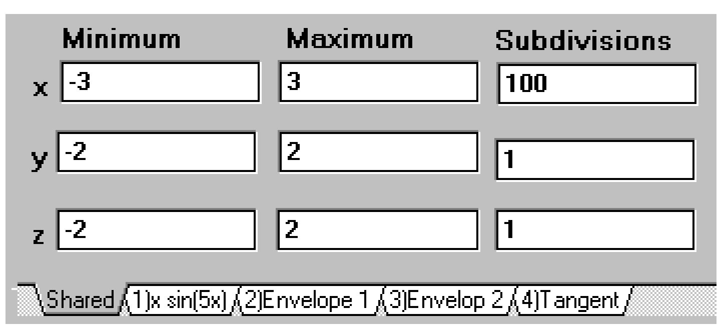
x sin(5x)

The most familiar functions have one input and one output. Our example graphs *f(x) = x*sin(5*x)*.



From the Layers Dialog

The function is defined in layer 1. Notice that f_x is a copy of the domain (*x*). The f_z component is not used since we are working only in the *x*-*y* plane.

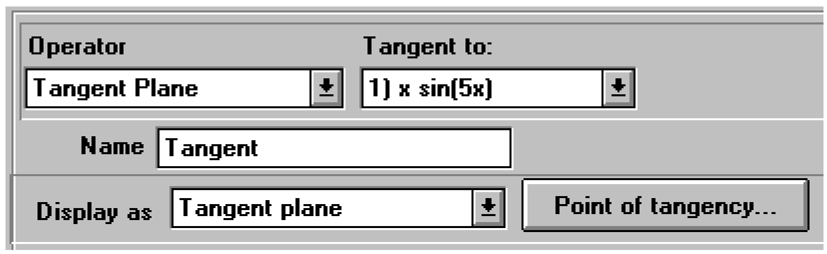


From the Domain Dialog

The Domain dialog shows that the entire domain lies on the x-axis. Notice that *y* and *z* from the domain do not appear as input in any of the function definitions. There is one subdivision in each of those dimensions. If we had defined multiple subdivisions in either the *y* or *z* dimensions, the resulting image would not be changed. However, the same graph would be drawn multiple times. (The *x*-axis would be scanned for each *y* value in the domain.) This is a needless repetition; it would only slow the drawing.

In layers two and three we have added the lines $y=x$ and $y=-x$ to illustrate the "envelope" that results from the multiplication by x.

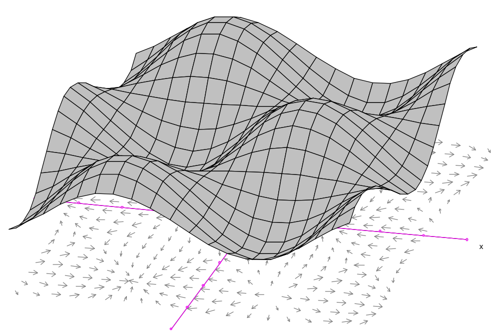
The fourth layer draws a tangent line to the curve. When you animate the graph, the tangent line will move along the graph in each frame. We reserve a discussion of the Tangent operator until later.



When you inspect the layer definition for the tangent operator, pay attention to the TANGENT TO drop-down list. Also check the path for the tangent animation on the Tangent At dialog, accessed through the 

button which appears on the Layer dialog when the operator is Tangent.

R^2 ® R^1(SINCOS)



We have already seen an example of a function $R^2 \rightarrow R^1$.

$$z=F(x,y) = \sin(x)*\cos(y).$$

Note that this is a bona-fide function. Any vertical line intersects the graph at most once. I.e., any point (*x*,*y*) in the domain has only one result (*z*) associated with it.

The notion of derivative should tell us how the output of a function, at a given point, changes when we change the input. What can this mean if the input is two dimensional? In this case we can ask about the directional derivatives. How does the function change at a point*, if we change the input in a specific direction*? In a sense, the Gradient field captures this information. If ***u*** is a unit vector in some direction, the derivative in that direction can be shown to be ***u·grad(f)***. The gradient at each point indicates the direction in the function changes fastest; that is, the direction in which the derivative is largest.

In the example printed above the vectors of the gradient field are drawn normalized; all scaled to same length. This makes for neater drawings, but can be misleading unless understood. The Appearance page of the Layers dialog gives options for controlling vector scaling.

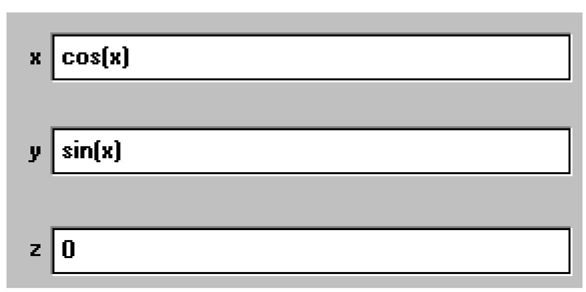
The tangent plane also gives information about the derivative. The tangent at a point is not a line but a two dimensional plane. Examples of tangents to a surface are given *in Derivatives and Tangents*, below.

R^1 ® R^2*(Circle)*

Before looking at this example, let's see how *not* to draw a circle. Theoretically, we can graph the top half of a circle of radius *1* by letting *x* vary from *-1* to *1* and letting y = *Ö(1-x^2)* (On the dialog this would be entered as *(1-x^2)^.5*.) This is unsatisfactory for two reasons. First the spacing between points on the circle is not uniform. (Why? Try it.) Second, it only gives the top half of the circle. (We would have to add a second layer for bottom half, defining *y = - Ö(1-x^2)*.)

A circle is not the graph of any function *y=f(x)* because most *x* values in the domain have two corresponding *y* values.

Instead, we draw a circle parametrically. The angle (from *0* to *2p*) is the one-dimensional parameter. *x* and *y* in the range are both computed from the angle.



Notice that we use *x* as the angle parameter. (We cannot use *theta* when working in rectangular coordinates.) The trick in defining functions parametrically is to not let the terminology determine how you think about functions. *x* is simply the input to the computation.

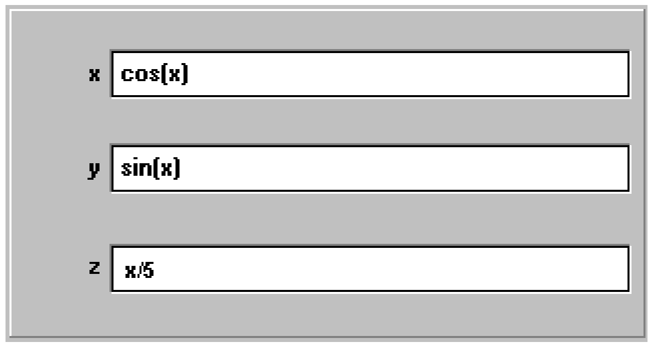
In the domain *x* varies from *0* to*2p*.

The circle we have drawn is not the graph of a function. Recalling the technical definition of a graph, we expect the graph in this case to be three dimensional. Points in the graph are of the form *(x, cos(x), sin(x))*. Can you describe the graph? How can you modify the layer to illustrate it?

Incidentally, drawing a circle is even easier using Cylindrical coordinates.

R^1 ® R^3 (Helix)

This is a simple extension of the Circle example above. Here we take a one dimensional line and lay it down as a path in three dimensions.

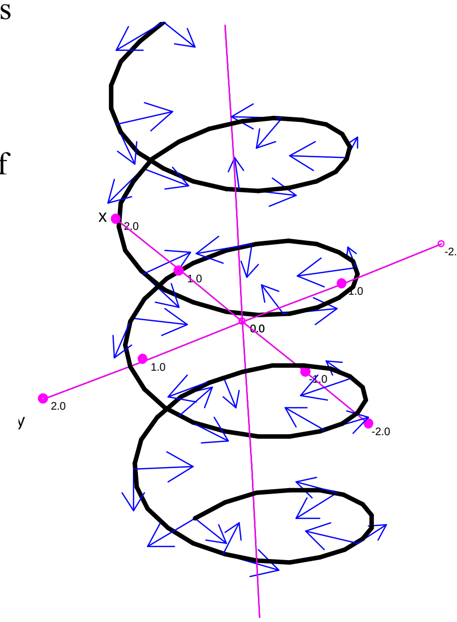


Dividing by *5* in the f_z component compresses the graph.

Derivative and Tangent Vectors

In general a curve in three dimensions is described parametrically by:

$f_x(x),\quad f_y(x),\quad f_z(x)$



How can we present the derivative of this function from R^1 to R^3 ?

As always, the derivative should describe how the output (points on the curve) change as the input (the *x* parameter) changes. This is exactly what the tangent vectors at points along the curve tell us.

Frenet Frame

The Frenet frame is a set of local coordinate axes at each point of a curve. It is composed of the tangent, the normal, and the bi-normal. (See Help for details.) When you select the Frenet frame Operator, you can choose with vectors will appear. In this illustration we included the tangent and normal, but not the bi-normal.

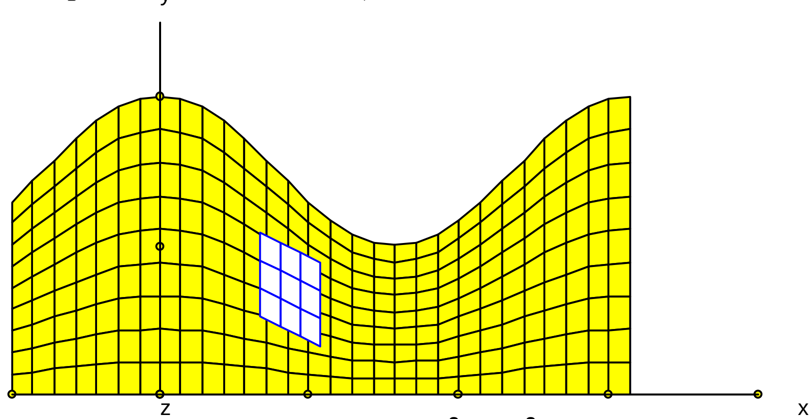
In order for the curve to be smooth, we use many subdivisions in the *x*-dimension on the domain. This would give too many vectors in the Frenet frame layer. Instead, we use a private domain with fewer subdivisions.

R^2 ® R^2 (Jacobian)

This is a distortion of the flat plane.

The Jacobian, at a point, is the linear map that best approximates the underlying map at that point. Again, the Tangent Plane operator performs the computation.

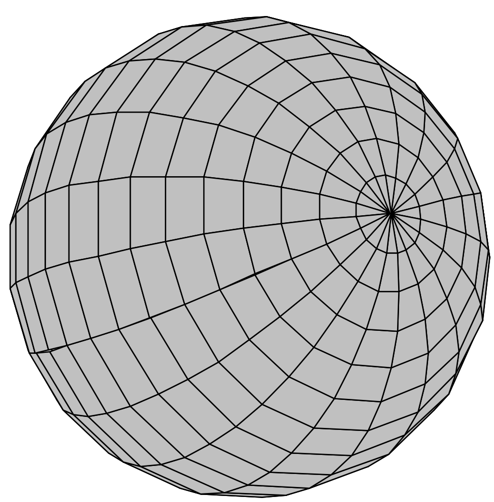
(Linear maps are discussed below.)



Another way to visualize maps from is **R^2** to **R^2** as a two dimensional vector field. We won't cover this here, but see $R^3 \rightarrow R^3$ below.

R^2 ® R^3 (Sphere)

The simplest way to draw a sphere is to use the identity functions in Spherical coordinates. We use Rectangular coordinates here to illustrate defining surfaces parametrically.

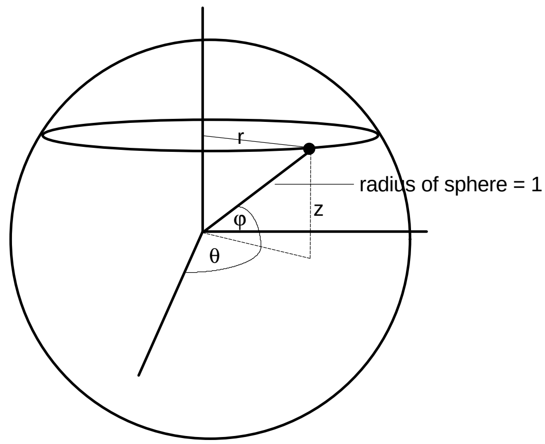


We can think of the $R^2 \rightarrow R^3$ case as taking a two dimensional sheet and distorting it to a surface in three dimensions. This is more general than the surfaces obtainable as *z=F(x,y)* discussed above. A sphere cannot be graphed as *z=F(x,y)* because it would not satisfy the vertical line test.

The Coordinate Functions

For simplicity we'll assume the radius of the sphere is 1.

To see how to draw a sphere, consider the circle shown here. All points on the circle have the same *z*-coordinate, *z=sin(f)*.



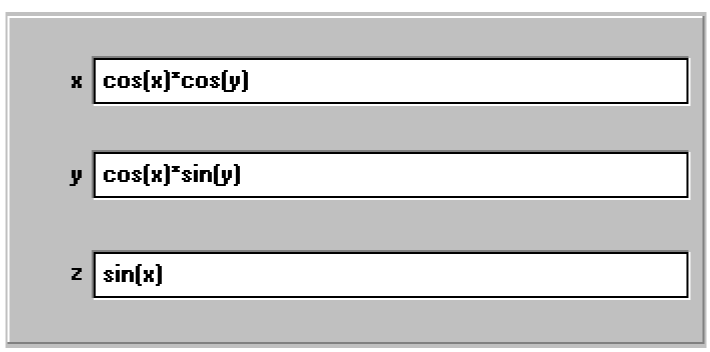
The radius of the circle (which is not the same as the radius of the sphere) is *r=cos(f)*. We get the complete circle if we take points *(r cos(q), r sin(q), z)* where *0£ q <2p*.

Substituting for *r,* this is:

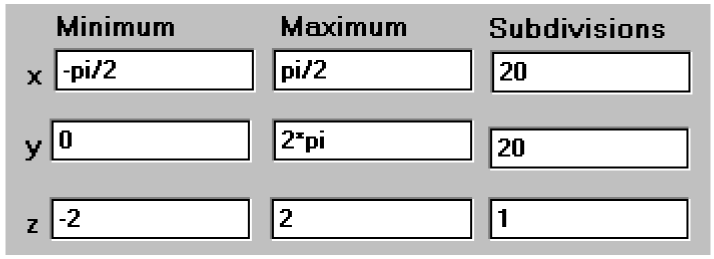
(cos(f) cos(q), cos(f) sin(q), sin(f))

To generate the full sphere we let *f* vary from *-p/2* to *p/2*.

Finally, we recall that in Rectangular coordinates the variables are called *x* and *y*, rather than *f* and *q*.



From the Layers Dialog



From the Domain Dialog

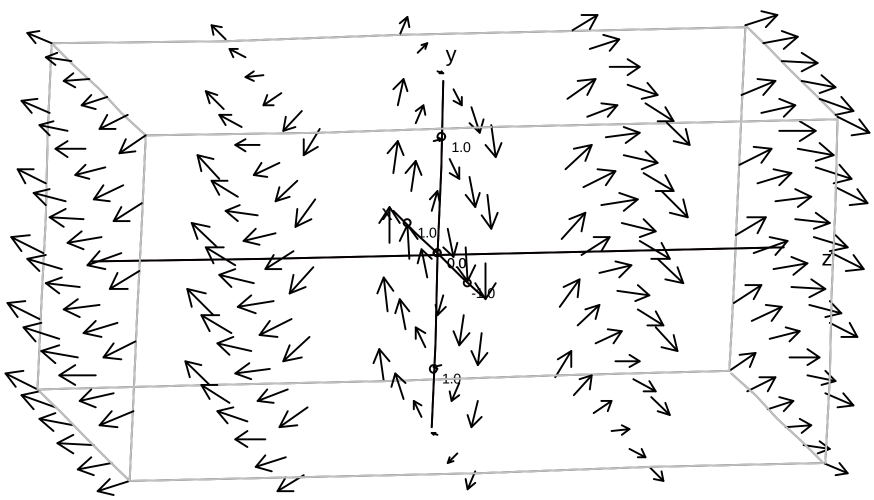
If you find it difficult to visualize the angle parameters, experiment by reducing the size on the domain to yield only a partial sphere.

For other examples of defining surfaces parametrically *see Passing Through a Torus* below.

Another method of visualizing functions from R^2 to R^3 is analogous to the next example. In fact one "slice" from that example serves. The vectors in one slice are based on a two dimensional domain, and point in three dimensional space.

R^3 ® R^3

The vector field in the file 3DFIELD [3D Field] is an example of a function from R^3 to R^3. We used that file when we looked at integral flows above.



(-y,x,z)

At each point in a three dimensional domain we place a three dimensional vector that represents the value of the function at that point. Without use of animation this is the highest dimensional situation that Fields&Operators can handle.

Other Examples

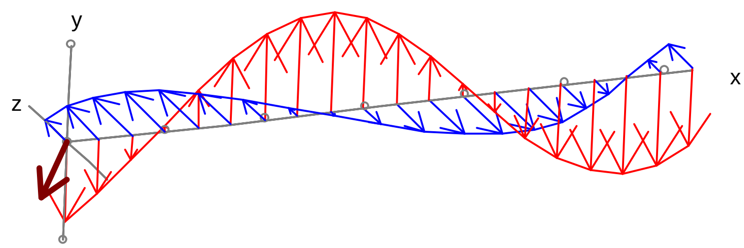
These are a few miscellaneous examples. They are chosen as suggestions for using Fields&Operators.

Each example has an associated file on your disk. Open the files and look at the Layers and Domain dialogs for details on how they are defined.

All of these examples were designed to be animated.

Light Ray (LIGHTRAY)

A light ray is composed of perpendicular oscillating electric and magnetic fields. The changing electric field induces a magnetic field, and the changing magnetic field induces an electric field.



This is complex behavior; even a graph such as that above can be confusing. Animation can help in visualizing the action.

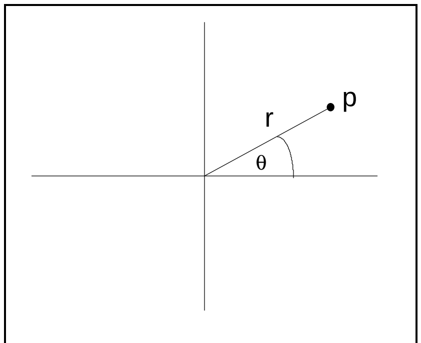
This graph is composed of 5 layers. There is a curve for the outline of each field, as well as the two fields. The fifth layer is a single vector rotating in the *y*-*z* plane.

Normalization is turned off for all vector fields so they are drawn to their correct lengths.

Conic Sections (CONICS)

This example demonstrates the use of Polar coordinates.

You won't find an option for polar coordinates on the Coordinates dialog. Polar coordinates can be though of as the two-dimensional version of either cylindrical or spherical coordinates. (Cylindrical coordinates treat the elevation above the *x*-*y* plane by giving the *z*-coordinate, while spherical coordinates give the angle of elev ation to the point.) In this example we use cylindrical coordinates. In spherical coordinates the angle would be called *f* (phi).



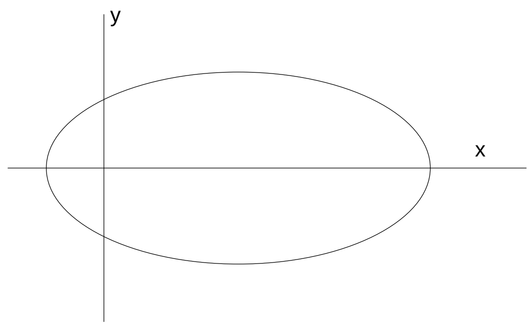
Polar Coordinates

To specify a point in polar coordinates, we imagine a radius, *r*, drawn from the origin to the point. The point is identified by giving the length of r and the angle, *q* (theta), between the *x*-axis and *r*. Notice that any point can be identified in infinitely many ways, because one or more full circles ($n \cdot 2p$) can be added to any angle measurement.

Conics The conic sections - ellipses, parabolas, and hyperbolas - can all be described by a simple class of formulas in polar coordinates.

The general form of a conic section is

$$r = d\, e/(1 - e \cos(q))$$



where *e* is the eccentricity. When $0 < e < 1$ the result is an ellipse. When $e = 1$ the result is a parabola, and when $1 < e$ the result is an hyperbola. (What is the geometric interpretation of *d*? Here we use $d = 1$.)

Any time a family of functions depends on some parameter it is likely that animation can add insight into the functions. (Also consider using animation whenever a constant appears in a function definition.)

We can watch a family of conics evolve by generating an animation in which we use *t* for the eccentricity.

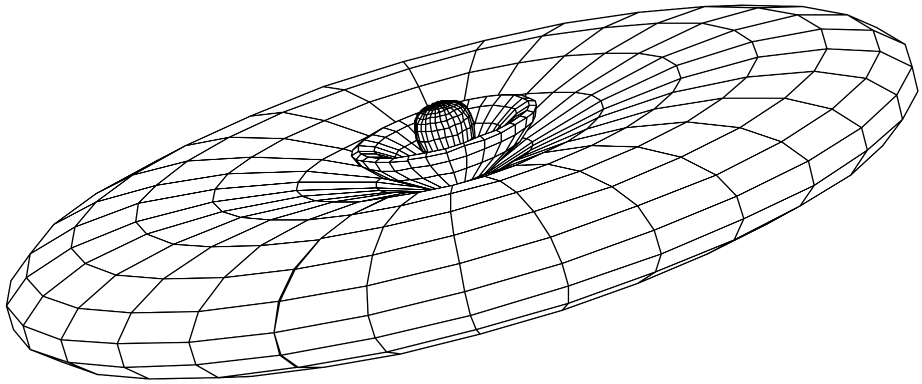
$$f_r = t/(1-t*\cos(theta))$$

In the animation *t* varies from *0* to *2*. The domain has one *z*-subdivision and one *r*-subdivision.

Can you generalize this animation to surfaces?

Antenna

This example is based on graphs presented by Warren Stutzman and John McKeeman of Virginia Tech, Dept. of Engineering at a workshop of Computer Applications in Electromagnetics Education. It represents the antenna pattern for a five element, half-wavelength spaced, uniformly excited linear array.



It is defined in spherical coordinates as:

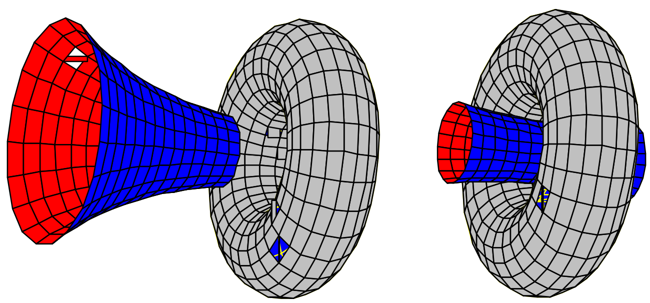
f_r=*abs(sin(t*pi/2*cos(theta)) / (t*sin(pi/2*cos(theta))))*

where *t = 5*.

Notice that if *t=1* this long expression reduces to *+1*, giving a sphere of radius *1* in the graph.

Animation reveals how the lobes evolve as *t* changes from *1* to *5*.

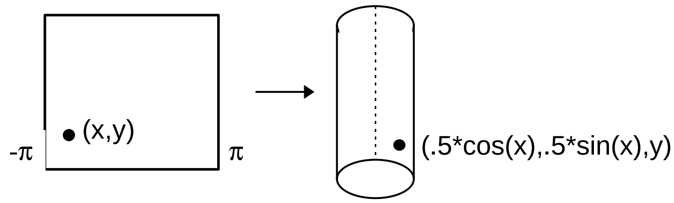
Passing Through a Torus (Through)



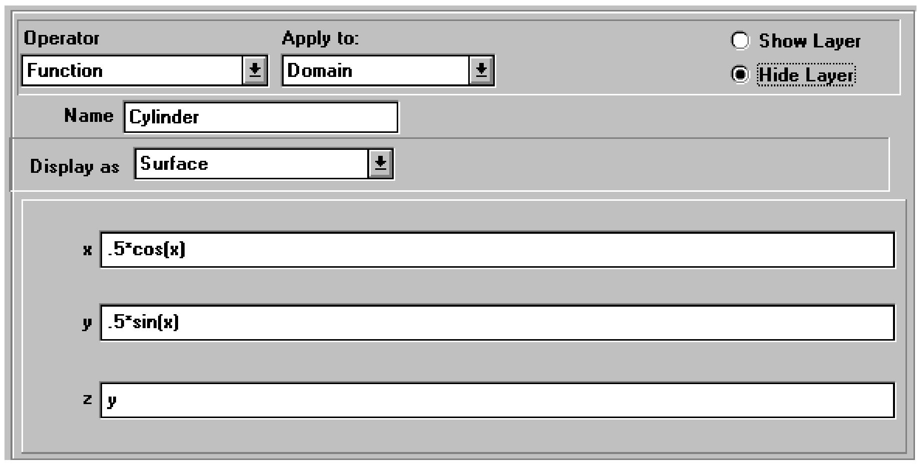
This examples shows a horn passing through a torus. We present it because it involves two separate tricks for generating such animations. The first is in using two layers to draw the torus. The second is how the horn is made to move through the torus.

Drawing a Torus

Writing down a formula that maps a rectangular domain onto a torus can be an interesting, though tedious, exercise. Using two layers makes it simple.



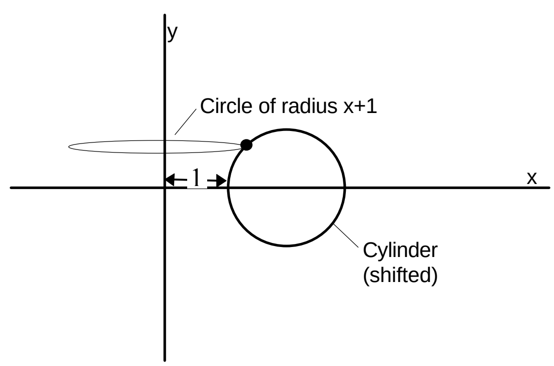
The first step is to "wrap" the rectangle into a cylinder of radius 1/2. The diagram shows the map from two dimensions to three dimensions.

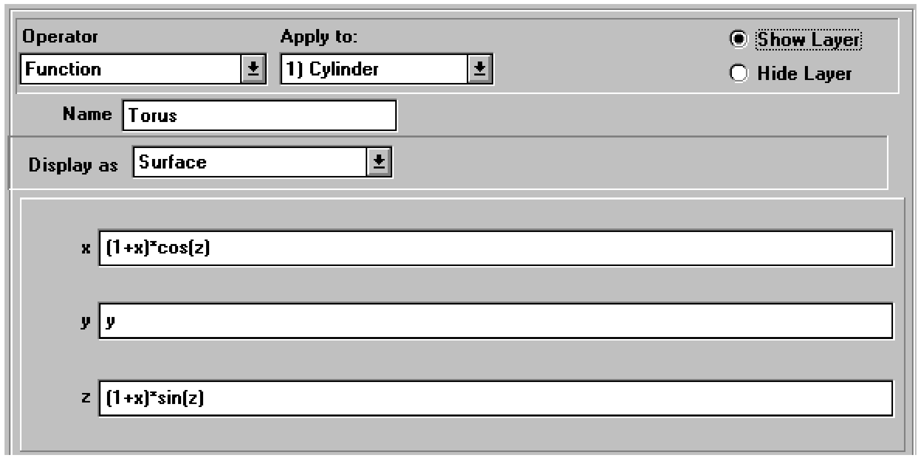


From Layer 1 - Cylinder

Since this is only an intermediate step, the layer is hidden. (Note the radio button at the top of the dialogo.) This means the computations for the layer are available for other layers, but the layer is not drawn as part of the graph.

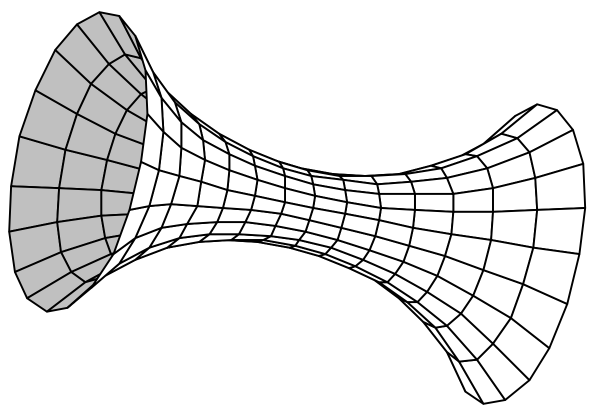
The torus is completed in the second layer by making use of a second wrapping operation. The computation is similar, but this time it is applied to the cylinder of the first layer, and is perpendicular to the first wrap. The axis for the cylinder is thought of as the angle parameter for the torus.





The Horn

The effect of the moving horn is created by taken successive slices from the surface below. It is similar to the cylinder we defined in layer 1, except that its radius changes along its axis. In this case it is centered on the y-axis and the radius is the hyperbolic cosine of y. (*cosh(y)* is close to *1* when y is near *0*, and increases exponentially for large *y*.) We divided by *3* to make it somewhat thinner. The surface is a catenoid.



In the full catenoid above, *y* is between -*2* and *2*. For the animation we take pieces of length *2* and move the pieces by adding *t*. (Look at the Domain and Animation dialogs.)

Derivatives and Tangent Planes

The definition of derivative that is easiest to generalize to high dimensions is that the derivative at a point is the best linear approximation to the given function at that point.

This is illustrated graphically with the Tangent operator.

The next few paragraphs give some background into linear maps and derivatives. Be assured that the graphs we will discuss are generated with only a few mouse clicks or key strokes.

The Tangent operator is not available with cylindrical or spherical coordinates.

Linear and Affine Maps

Linear maps are the simplest functions to understand. In one dimension, a linear function is simply a multiplication. In higher dimensions it is multiplication by a matrix.

For the sake of illustrating linear approximations via tangents, we will use affine maps rather than linear maps. An affine map combines a linear map (stretch) with an addition (shift).

The one dimensional case is most familiar, but it will actually be easier to begin with linear maps applied to a two dimensional domain.

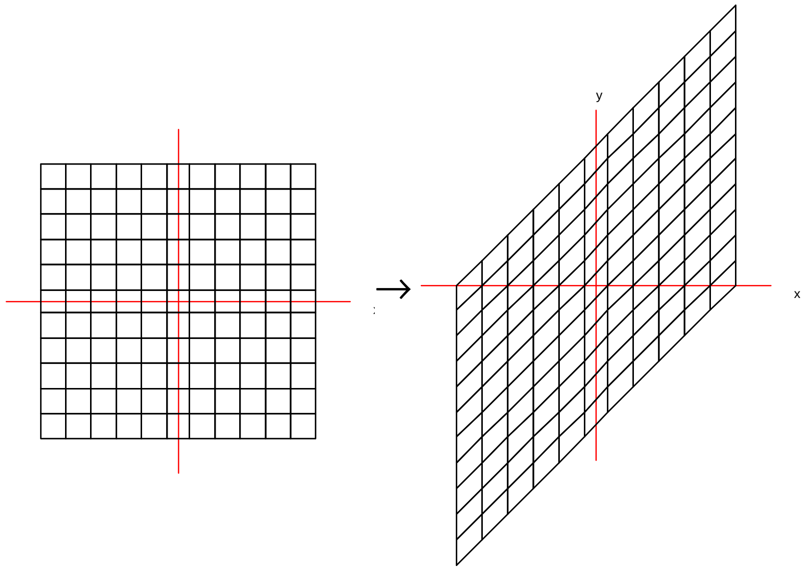
The general form of a linear map from a two dimensional domain to three dimensions is

$$f_x = ax + by$$
$$f_y = cx + dy$$
$$f_z = ex + fy.$$

where a,b,c,d,e, and f are constants. Different constants give different maps. If e and f are both 0, the result is a map from the plane to the plane.

A linear map is usually summarized by writing the coefficients in a matrix.

$$\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}$$



Obviously any linear transformation maps the origin to the origin. Also, lines are mapped to lines, and parallel lines are mapped to parallel lines. A typical linear transformation is illustrated.

The Jacobian Matrix

For a map (not necessarily linear) in rectangular coordinates from R^2 to R^3 the Jacobian matrix is defined as

$$\begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} \\ \frac{\partial f_z}{\partial x} & \frac{\partial f_z}{\partial y} \end{bmatrix}$$

All partial derivatives are evaluated at ***P***

For each point ***P*** in the domain, this is just a matrix of numbers, so it corresponds to a linear map. The linear map it defines is the best linear approximation at ***P***.

As mentioned, linear maps always pass through the origin. To draw the tangent plane, we must use an affine version, shifting the plane to the point of tangency by adding the coordinates of the image of ***P*** in the surface. The component functions for the plane are:

$$\frac{\partial f_x}{\partial x} x + \frac{\partial f_y}{\partial x} y + Q_x$$

$$\frac{\partial f_x}{\partial y} x + \frac{\partial f_y}{\partial y} y + Q_y$$

$$\frac{\partial f_x}{\partial z} x + \frac{\partial f_y}{\partial z} y + Q_z$$

where ***Q*** is the image of ***P***.

Transformations and Equations of a Plane

When you compare these expressions to the equation for the tangent plane found in text books, keep in mind that Fields&Operators works with transformations applied to the domain; it is a computation applied to x and y coordinates producing other coordinates. The equation of a plane, as it appears in most texts, is an equation that is valid for points that lie in the plane. We leave it to you to reconcile these two points of view.

One Dimensional Maps

We have already seen an example of a one dimensional tangent in the XSIN(5X) example. There the function was of the form *y=F(x)*. Exactly the same techniques are used even if the curve defined parametrically with a two dimensional range (e.g. a circle) or a three dimensional range (e.g. a helix).

When drawing tangent lines be sure that the Tangent layer uses a one dimensional domain. This is automatic if it shares the same domain as the curve, but be careful if you define a private domain for it.