Plotting in mathematica

Plot is a simple two-dimensional tracing function in Mathematica. Plot takes two arguments when named and these two arguments can contain numerous parts. This may not sound so simple, but once you see it works it's very straight forward. The first argument for A graphically represented is the function or functions to be plotted. The second argument is a list that contains a variable to be used in the range over which the variable is to be represented. Example: Plot[Cos[x], {x, -2Pi, 2Pi}] The line shown above will draw a simple cosinus curve from -2pi to 2pi. The function, Cos[x], is the first argument for Graphic Representation. This is then followed by a comma and the second argument. The parentheses are used with the second argument because they contain several parts, each of which is separated by a comma. The first part is the variable to be used, followed by the beginning of the interval, and then the end of the interval. The important points to remember about Plot is that two arguments are passed separately by a comma. The arguments themselves can contain numerous parts that would then be enclosed in parentheses and each separated by a comma. Many functions can be plotted at the same time with a single call to Plot. These functions must use the same variable and range, however. Example Plot[{Sin[x], Cos[x]}, {x, -2Pi, 2Pi}] This line will plot a sinus and cosinus function from -2pi to 2pi. Notice that both Sin[x] and Cos[x] are closed by parentheses and separated by a comma. Plot3D Plot3D works in the same way that plot does not, except two variables are used and the function is named with a minimum of three arguments. These arguments consist of the function or functions to be plotted, a variable and its range, as well as a second variable and its range. Again, these arguments are separated by commas. Example Plot3D[Sin[x]+Cos[y], {x, 0, 4Pi}, {y, 0, 4Pi}] This equation will draw a mountainous surface that resembles a box of eggs. Both axes will have an interval of 4pi. Plot3D is able to accept more than three arguments. These arguments generally address the appearance of the final plot of the catchment area and the way in which it is viewed. More plot3D Plot3D examples[Sin[x+Sin[y]], {x, 0, 4Pi}, {y, 0, 4Pi}, PlotPoints->30] Plot3D[Sin[x+Sin[y]], {x, 0, 4Pi}, {y, 0, 4Pi}, PlotPoints->50, Mesh->False] The lines above describe a 3 dimensional plot that has been easily adjusted for viewing. In the first line PlotPoints->30 increases the number of divisions in the plot. By increasing the number of divisions or PlotPoints, as was done in the second example, a surface will essentially become smoother. The second line contains ,kbr> which has the effect of removing gridlines from the surface. Summary The example shown above were taken from Exploring Mathematics with Mathematics and were intended to give a basic start to plot with Mathematica. Also, contain some books that are recommended for people who want to learn the more advanced features of Mathematica. References [ST94] Applied Mathematica William T. Shaw and Jason Tigg Addison-Wesley, Menlo Park CA (1994). [Rob95] Mathematical Engineering with Mathematics John S. Robertson McGraw-Hill, New York NY (1995). [GG91] Exploring mathematics with Mathematics Theodore W. Gray and Jerry Glynn Addison-Wesley, Menlo Park CA (1991). Week 1 Precept Missions Last modified: 97.08.12 Michael Carreno Notification! This document is currently in an archived state. The contents of this document may be incorrect or outdated. Drawing the base outline Draw the density Tracing the 3-D graph Draw the base graph with the Plot command, you can draw a function or several functions. First, it allows you to consider an example of tracing a single function. Take, for example, the equation, x3. To plot a single function, the Graphics command takes the following form: Graphics[function,{variable,xmin,xmax}] For the x3 equation with the limit conditions, -10 < x < 10, enter the Graphic Representation command by follows: When you press the shift and return key at the same time a graph of this function will appear. Then consider the two x2 and x3 equations above the limit conditions, 0 < x < 5. The Graphics command for multiple functions takes the general form: Plot[{function1,function2,...}, {variable,xmin,xmax}] For our two functions, the Plot command becomes: When you press the shift and return key at the same time, a graph of the two functions will appear. Outline Plotting Outline outline is frequently done in science. For example, in the field of atmospheric sciences contouring is done on meteorological maps to show constant lines of pressure, temperature, wind direction, etc. MATHEMATICA can take over functions and contour. For contouring, use the ContourPlot command that takes general form: ContourPlot[function,{variable,xmin,xmax},{variable,ymin,ymax}] For an X2+y2 equation where the limit conditions are, -2 < x < 2 and -2 < y <2, the ContourPlot command takes shape: ContourPlot[x^2+y^2,{x,-2,2},{y,-2.2}] When you press the shift and return keys at the same time a contoured graph should appear. Chart Drawing Density A density plot displays function values in a regular array of points. Lighter regions on a density are larger. To make a density graphic, use the DensityPlot command that has the general shape: DensityPlot[function,{variable,xmin,xmax},{variable,ymin,ymax}] Consider the x2y3 equation with limit conditions, -3 < x < 3 and -3 < y < 3. DensityPlot command will take the form: DensityPlot[x^2*y^2,{x,-3.3},{y,-3.3}] When you press the shift and return key at the same time, it will appear density plot. 3-D Tracing 3-D plot can help visualize physical problems. To make a 3-D graphic, use the Plot3D command that has the general form: general: Consider the Sin(xy) equation at the limit conditions, 0 < x < 3 and 0 < y < 3. The Plot3D command will take the form: Plot3D[Sin[xy],{x,0.3},{y,0.3}] When you tap the shift and return key at the same time, a 3-D graphic will appear. Last modified: 2007/03/07 11:48:6.483000 US/Eastern by Unknown Created: 2007/03/07 11:48:6.483000 US/Eastern by brian.r.brinegar.1. Type Search categories in several keywords that describe what information you're looking for in the text box below. Admin Options: Edit this document Plotting Planes in MathematicaCopyright © 1995, 1997, 2001 by James F. Hurley, University of Connecticut, Department of Mathematics, Unit 3009, Stors CT 06269-3009. All rights reserved. This notebook discusses airplanes in 3-space, and the Mathmatica code to generate plots of them. 1. Normal shape. The best algebraic representation of the planes is the normal shaped equation, the geometric basis of which is simple: a plane is uniquely determined by a P-point and a normal vector n. Definition. The normal shape equation of the plane by P(x0, y0, z0) with the cloud-mal vector n = ai + bj + ck = (a, b, c) is(1) n' (x – x0) = 0, where x = (x, y, z) = xi + yj + zk = OQ for any Point Q(x, y, z) on the plane and x0 = OP = x0i + y0j + z0k =(x0, y0, z0) is the vector from the origin to point P. Equation (1) says that a point Q(x, y, z) is on the plane if and only if the plane's normal vector is perpendicular to the vectorx – x0 from the point givenP toQ. Vectorequation easily leads to a scalar equation. Definition of product point and (1) yes (a i + b j + c k) ? [(x – x0) i + (y – y0Lj + (z – z0Lk] = 0, a(x – x0) + b (y – y0) + c(z – z0) = 0ax + by + cz = ax0 + by0 + cz0(2) ax + by + cz = d , where d = n-x0. By contrast, any linear equation of the shape (2) is the equation of a plane in 3-space, provided that not all three a, b and c are 0. To see why, let P(x0, y0, z0) be any point Planes.nb So requested: For the first example you were solving manually for y and plot Result. For the second problem it wasn't so obvious to do that by hand. Solve can do this for you. Plot[Re[y/. Resolve[y^2/x^2==1-(20/y^2)(y^2-40)/(y^2-60)),y]],{x,0,1}] Re[] displays only the actual solutions. Y/. extract the solutions from the form returned by Solve. Now to answer the question about the ratnz warning. There are often several different algorithms inside the functions, such as Resolution or Reduction. Some of these algorithms work only with exact rational values or integers, others work with decimal approximations. Graphic representation uses decimal values to display graphs. So it is the passage of decimal values to resolve. To find the solutions you're looking for, it seems that Solve believes that using exact values would be the best approach. So it warns you that it is turning the approximate decimal values back to the exact rationals, finding the solution and then moving that back to plot, which then turns the ones back into decimal places and displays the curves. We can avoid this warning with this change solutions = Re [y / . Resolution[y^2/x^2==1-(20/y^2)(y^2-40)/(y^2-60)),y]]; Plot[solutions,{x,0,1}] that first finds the exact solutions and then hands those solutions to Plot. Does that explain enough? fAprox[max_, t_] := (1/2) + Sum[Sin[2 n Pi]/(Pi n) Cos[n Pi t] + (((-1)^n - Cos[2 Pi n])/(Pi n) Sin[n Pi t], {n, 1, max}] f[t_] := Piecewise[{0, 0 < t < 1}, {1, 1 < t < 2}}]; Handling[Plot[{f[t], fApprox[nTerms, t]}, {t, 0, 2}, PlotRange -> Automatic {, {-0.3, 1.3}}, PlotStyle -> {{Thick, Blue}, Red}, Exclusions -> None], {{nTerms, 5, How many terms?}, 1, 30, 1, Layout -> Tagged}, TrackedSimsymbolizes :> {nTerms}] Observe the Gibbs effect on where $f (x)$ is discontinuous. There is 9% overshoot on each side, which cannot be reduced, no matter how high the number of terms is. To plot the extended periodic version: fApprox[max_, t_] := (1/2) + Sum[Sin[2 n Pi]/(Pi n) Cos[n Pi t] + (((-1)^n - Cos[2 Pi n])/(Pi n) Sin[n Pi t], {n, 1, max}] f[t_] := Piecewise[{0, 0 < t < 1}, {1, 1 < t < 2}}]; fExtended[t_]:= If[t < 0 || t > 2, f[Mod[t, 2]], f[t]] Manipulate[Plot[{fExtended[t], fAprox[nTerms, t]}, {t, -4, 4}, PlotRange -> {Automatic, {-0.3, 1.3}}, PlotStyle -> {{Thick, Blue}, Red}, Exclusions -> None], {{nTerms, 5, How many terms?}, 1, 30, 1, Appearance -> Tagged}, TrackedSymbols :> {nTerms}] For more cool Fourier series animations, all made using Mathematica, I found this web page (for some reason, the Mathematica source code used for those is not displayed at this time). Mathematica is probably the best software to make such animations. Animations.