

Mathematica Tutorial – Graphics

Section 1: Introduction.....	2
Section 2: The Plot Function.....	2
2.1 The Basic Plot	2
2.2 Specifying Options.....	2
2.3 Axis Labels.....	3
2.4 Plot Labels in <i>Mathematica</i>	3
Section 3: More Advanced Features	4
3.1 Using Show to Alter a Plot.....	4
3.2 Simple Superscripts.....	4
3.3 Superscripts and Subscripts.....	5
3.4 The Text Function	5
3.5 Using FontForm to Control Text Fonts.....	6
3.6 Color in Mathematica Plots.....	7
3.7 Setting the Default Color.....	7
3.8 Using PlotStyle to Control the Color of Individual Curves	7
3.9 Setting the Axes Color with AxesStyle.....	7
Section 4: Setting Styles and Colors for Tick Markes.....	8
Section 5: Plot Legends	8
5.1 Loading the Legend Package	8
5.2 A Basic Legend	8
5.3 Using Additional Fonts in a Plot Legend.....	9
Section 6: Graphics Arrays	9
Section 7: 2-D Animation.....	10
Section 8: References.....	12

Section 1: Introduction

Mathematica is capable of generating a wide range of graphics, from a simple plot of a function, to three dimensional animations of complex physical systems. This document will concentrate on the most fundamental and most widely used of the *Mathematica* plotting capabilities, the use of the **Plot** function to generate two dimensional graphics.

Many of the functions demonstrated here produce graphical results. These results are not displayed on this page. Hypertext links from the function calls lead to gif images of their results. The images used in this document were obtained by grabbing the screen image of the plot with **XV** and saving it as a gif.

Section 2: The Plot Function

2.1 The Basic Plot

Plot is the simplest and most frequently used of the graphics functions. Calls to the **Plot** function are of the form:

```
Plot[f[var], {var, varmin, varmax}].
```

This generates a plot of the function f as var varies from $varmin$ to $varmax$. For example, to plot **Sin**[$theta$] for $0 < theta < 2\pi$ use:

```
Plot[Sin[theta], {theta, 0, 2 Pi}]
```

Plot is, of course, not limited to operating on functions. As long as it depends on a single variable, almost any expression that *Mathematica* can evaluate can be used. For example:

```
Plot[x^2 + x^5, {x, -10, 10}]
```

2.2 Specifying Options

Optional arguments to the **Plot** command provide a great deal of control over the appearance, layout, and information content of the plot. Options are set by passing :

```
Option -> Value
```

at the end of the argument list. **Options**[*Function*] returns a list of the *Function*'s default options.

```
Options[Plot]
```

returns

```
{AspectRatio -> -----, Axes -> Automatic,
      GoldenRatio

  AxesLabel -> None, AxesOrigin -> Automatic,

  AxesStyle -> Automatic, Background -> Automatic,

  ColorOutput -> Automatic, Compiled -> True,

  DefaultColor -> Automatic, Epilog -> {}, Frame -> False,

  FrameLabel -> None, FrameStyle -> Automatic,

  FrameTicks -> Automatic, GridLines -> None,

  MaxBend -> 10., PlotDivision -> 20., PlotLabel -> None,

  PlotPoints -> 25, PlotRange -> Automatic,

  PlotRegion -> Automatic, PlotStyle -> Automatic,

  Prolog -> {}, RotateLabel -> True, Ticks -> Automatic,

  DefaultFont :> $DefaultFont,

  DisplayFunction :> $DisplayFunction}
```

2.3 Axis Labels

One of the most commonly used **Plot** options is **AxesLabel**. **AxesLabel** attaches labels to the axes. Assigning it a single value places that label on the y axis:

```
Plot[Sin[theta], {theta, 0, 2 Pi}, AxesLabel -> "Sin[theta]"]
```

Assigning **AxesLabel** a list of two values, $\{x\text{label}, y\text{label}\}$, causes *Mathematica* to use these values as labels for the x and y axes respectively.

```
Plot[Sin[theta], {theta, 0, 2 Pi},
      AxesLabel -> {"theta", "Sin[theta]"}]
```

2.4 Plot Labels in *Mathematica*

An option closely related to **AxesLabel** is **PlotLabel**. The **PlotLabel** option associates a label with the entire plot.

```
Plot[Sin[theta], {theta, 0, 2 Pi},
      AxesLabel -> {"theta", "Sin[theta]"},
      PlotLabel -> "Sin[theta] vs theta"]
```

Section 3: More Advanced Features

3.1 Using Show to Alter a Plot

Mathematica graphics, once generated, can be redrawn or altered with the **Show** command. **Show** changes properties such as the aspect ratio, which, if any, grid lines to draw, or which colors are used in the plot. The **Viewpoint** and **PlotRange** attributes can be manipulated to produce panning and zooming effects. **Show** is, however, limited in that it can not produce any new data. The curve is generated by the **Plot** command, **Show** can only alter how it is displayed.

The following discussion assumes that we have an original plot:

```
In[1]:=
Plot[Sin[x^2], {x, 0, 2 Pi}]
```

Several examples will be given where **Show** is used to modify the presentation of this plot.

3.2 Simple Superscripts

As described above, the **AxesLabel** option to **Plot** attaches labels to the axes of a *Mathematica* plot. If the arguments to this option are quoted, they will be placed on the plot exactly as typed. However, if the arguments are not quoted then they are displayed in *Mathematica's* **OutputForm** [*Mathematica*, PP 343-346].

```
Show[%, AxesLabel->{x, Sin[x^2]}]
```

This means that fractions and exponents will be presented in a two-dimensional form. It also means that *Mathematica* will try to evaluate any expression that it is given. In the above case, if x is undefined then:

```
Sin[x^2]
```

becomes

```
Sin[x2]
```

and this will produce the desired label. However, if x is defined, perhaps to be 3, then

```
Sin[x^2]
```

will evaluate to **Sin[9]**, which is not suitable as a label for a plot axis. If a variable's definition interferes with its use in a plot label, remove the definition for the variable with:

```
Remove[variable]
```

Where *variable* will be replaced with the name of the variable to be removed.

3.3 Superscripts and Subscripts

More sophisticated expressions can be generated with *Mathematica*'s **Subscripted** function [*Mathematica*, PP 360-361].

```
Subscripted[f[x,y,z],2,-1]
```

The first argument $f[x,y,z]$, is a variable and an argument list. The second argument to **Subscripted** tells which of the arguments of f should be used as subscripts. In this case 2 means take the first two (x and y) as subscripts. The last argument to **Subscripted** describes the arguments that are to be taken as superscripts. The -1 denotes the last element of the list (z). This produces:

$$f_{x,y}^z$$

Subscripted can be used with **AxesLabel**:

```
Show[%1,AxesLabel->Subscripted[f[x,y,z],2,-1]]
```

3.4 The Text Function

The **Text** function provides much finer control over where the text is placed [*Mathematica*, PP 469-471]. The **Text** function takes two arguments, a text string, and the $\{x,y\}$ coordinates for the center of the string. To place labels on our plot using this function use:

```
Show[%1,Epilog->{Text[Sin[x^2],{-0.355,1.25}],
                  Text[x,{6.9,-0.11}]},
      PlotRange -> All]
```

For 2-D graphics the same coordinate system that is shown on the axes of the plot can be used. However, for 3-D graphics *Mathematica* uses a special coordinate system to position objects. Positioning text in this case will take a bit of trial and error. Some of the front ends to *Mathematica* provide a method to determine the coordinates of an arbitrary point on a plot. For the Macintosh front end, select a plot, then hold down the command key (the one with the apple on it) while the cursor is over the plot, the coordinates of the cursor will be displayed in the lower left corner of the *Mathematica* window. The X interface uses the meta key and displays the coordinates in the lower right corner. Other front ends may provide a similar method to determine coordinates on a plot.

The **Epilog** option causes the text to be drawn after other parts of the plot. This has two advantages. First, the text will be drawn over any other parts of the graph and can not be obscured by them. Further, if we attempt to draw the text as part of the main graph, the axes of the graph will extend to include the labels; we do not want this.

3.5 Using FontForm to Control Text Fonts

The **FontForm** function [see P468 of *Mathematica*] changes the font of the displayed text. The **FontForm** function is inserted around the text argument to the **Text** *Mathematica* function. So that instead of:

```
Text[Sin[x^2],{-0.355,1.25}]
```

one has

```
Text[FontForm[Sin[x^2],
             {"Courier-Oblique",10}],
     {-0.355,1.45}]
```

This generates the string

```
  2
Sin[x ]
```

in the **Courier-Oblique** font of 10pt size and places it at the coordinates $\{-0.355,1.45\}$. A full **Plot** command using this would look like:

```
Show[%1,Epilog->{Text[FontForm[Sin[x^2],
                             {"Courier-Oblique",10}],
                 {-0.355,1.45}],
                 Text[FontForm[x,
                             {"Courier-Oblique",10}],
                 {6.9,-0.11}]},
     PlotRange -> All]
```

These options and functions can be combined to produce many effects, for example Greek letters can be used in plot labels:

```
Show[%1,Epilog -> {Text[FontForm["Sin[",
                                {"Courier-Oblique",10}],
                        {-0.355, 1.45}],
                  Text[FontForm["q",
                                {"Symbol",10}],
                        {0.120, 1.45}],
                  Text[FontForm["2",
                                {"Courier-Oblique",10}],
                        {0.28, 1.55}],
                  Text[FontForm["",
                                {"Courier-Oblique",10}],
                        {0.34, 1.45}],
                  Text[FontForm[q,
                                {"Symbol",10}],
                        {6.9, -0.11}]},
```

```
PlotRange -> All]
```

Far more creative combinations are possible.

3.6 Color in Mathematica Plots

Color adds a great deal to the information content and appearance of most graphs. Color is manipulated in *Mathematica*'s graphics routines by passing various optional arguments to the *Mathematica* graphics functions. **Plot** has optional arguments to set the colors for: the default color for the plot, each curve in the plot, the axes, the frame, the axes labels, the plot title, and even the tick marks on the axes.

3.7 Setting the Default Color

To render the first plot from this note in red use:

```
Plot[Sin[theta], {theta, 0, 2 Pi},  
     DefaultColor -> RGBColor[1, 0, 0]]
```

3.8 Using PlotStyle to Control the Color of Individual Curves

The color of a curve or curves on a plot is set by the **PlotStyle** option.

```
Plot[{Sin[theta], Cos[theta]}, {theta, 0, 2 Pi},  
     PlotStyle -> RGBColor[1, 0, 0]]
```

Providing a list of **RGBColors** for the **PlotStyle** option sets the color for each curve of the plot:

```
Plot[{Sin[theta], Cos[theta]}, {theta, 0, 2 Pi},  
     PlotStyle -> {RGBColor[1, 0, 0],  
                   RGBColor[0, 0, 1]}}
```

This generates the first curve (**Sin[theta]**) with the first color (red), and the second curve (**Cos[theta]**) in the second color (blue). Choosing appropriate colors for each curve in a plot makes it easy to follow the curves even when they cross or become intertwined.

3.9 Setting the Axes Color with AxesStyle

Colors are set for each element of the plot using the optional arguments. **PlotStyle** sets the style, in this case the color, for the curves that are plotted. The tick marks and the text that labels them are usually drawn in the **DefaultColor**. Finally the style for the axes are set with with the **AxesStyle** argument.

```
Plot[{Sin[theta], Cos[theta]},  
     {theta, 0, 2 Pi},  
     PlotStyle -> {RGBColor[0, 1, 0],
```

```
RGBColor[0,1,1]],
DefaultColor -> RGBColor[0,0,1],
AxesStyle -> RGBColor[1,0,0]]
```

Section 4: Setting Styles and Colors for Tick Markes

While an esoteric feature, it is possible to set the style for each individual tick mark. This is documented here primarily because the procedure described in the *Mathematica* Reference Manual is not correct. Note in the example below the values for the **Ticks** argument are of the form:

```
{x, label, len, {style}}
```

where the style argument is in the form of a list.

```
Plot[Sin[theta], {theta, 0, 2 Pi},
  Ticks -> {{{1, "1", .01, {RGBColor[1, 0, 0]}},
            {2, "2", .01, {RGBColor[0, 1, 0]}},
            {3, "3", .01, {RGBColor[0, 0, 1]}},
            {4, "4", .01, {RGBColor[1, 0, 0]}},
            {5, "5", .01, {RGBColor[0, 1, 0]}},
            {6, "6", .01, {RGBColor[0, 0, 1]}},
            {{1, "1", .01, {RGBColor[1, 0, 1]}}}]}
```

Section 5: Plot Legends

Once the elements of a plot have been differentiated with different styles, we can label them with a plot legend.

5.1 Loading the Legend Package

Drawing legends is not a built in part of *Mathematica*. To draw legends it is necessary to read in the standard *Mathematica* package **Graphics`Legend`**, with the command:

```
<< Graphics`Legend`
```

This need only be done once each session before the **Legend** package is used.

5.2 A Basic Legend

```
Plot[{Sin[theta], Cos[theta]}, {theta, 0, 2 Pi},
  PlotStyle -> {RGBColor[1, 0, 0],
              RGBColor[0, 0, 1]},
  PlotLegend -> {"Sin", "Cos"}]
```

5.3 Using Additional Fonts in a Plot Legend

The Legend package only accepts text in a single font for each label within the legend. More complex labels can be built using the technique illustrated above for building axes labels. For example the previous plot can be redisplayed with:

```
Show[%, Epilog -> {Text[FontForm["q",
                        {"Symbol", 10}],
                    Scaled[{0.24, 0.34}]],
      Text[FontForm["q",
                    {"Symbol", 10}],
          Scaled[{0.24, 0.16}]]}]
```

Epilog is used to draw this text on top of all the other elements of the plot, including the legend.

Section 6: Graphics Arrays

Frequently a single plot is not sufficient. One might wish to represent the solution to a differential equation for several values of a parameter, or the value of a series where the series is summed to different orders. This is done with the **GraphicsArray** function. **GraphicsArray** arranges a set of graphics objects into a regular pattern, an array. If we generate:

```
Row1 = Table[Plot[Sum[2^n x^n / (Factorial[n]^2), {n, 0, nmax}], {x, -.5, .5},
                PlotLabel -> StringForm["` ` terms", nmax]],
             {nmax, 0, 2}]
```

and

```
Row2 = Table[Plot[Sum[2^n x^n / (Factorial[n]^2), {n, 0, nmax}], {x, -.5, .5},
                PlotLabel -> StringForm["` ` terms", nmax]],
             {nmax, 3, 5}]
```

These are displayed together with

```
Show[GraphicsArray[{Row1, Row2}]]
```

Alternatively all the frames can be generated with a single call to **Table**:

```
Plots = Table[Plot[Sum[2^n x^n / (Factorial[n]^2), {n, 0, nmax}], {x, -.5, .5},
                  PlotLabel -> StringForm["` ` terms", nmax]],
              {nmax, 0, 5}]
```

An identical **GraphicsArray** to that shown above is created with:

```
Show[GraphicsArray[Partition[Plots, 3]]]
```

Where the **Partition** function breaks the original list of six graphs, into two lists, each containing three graphics objects. The second argument to **Partition** specifies the length of the resulting lists.

Section 7: 2-D Animation

Sometimes, especially if the parameter that varies from frame to frame is time, we will want to display the frames one at a time in series, rather than all at once in a **GraphicsArray**.

Mathematica has a number of ways to generate animations. There are some features that they all have in common. Animation in *Mathematica* is accomplished by first generating a series of images, then displaying them in rapid succession.

Let us start with the solution to a differential equation.

```
In[1]:=
  Solution = DSolve[{ x''[t] + kd x'[t] + x[t] == 0,
                    x[0] == 10,
                    x'[0] == 0},
                    x[t], t]
Out[1]:=
```

$$\left\{ \left\{ x[t] \rightarrow E^{\frac{(-kd - \sqrt{-4 + kd^2}) t}{2}} \left(5 - \frac{5 kd}{\sqrt{-4 + kd^2}} \right) + E^{\frac{(-kd + \sqrt{-4 + kd^2}) t}{2}} \left(5 + \frac{5 kd}{\sqrt{-4 + kd^2}} \right) \right\} \right\}$$

We now have an equation with two free parameters. This allows the generation of a series of graphs of x vs t , each with a differing value of kd . Those graphs can then be displayed as a movie. The first step is accomplished with:

```
In[2]:=
  Table[Plot[x[t] /. Solution, {t,0,20},
          PlotRange -> {-10, 10},
          AxesLabel -> {Time, Amplitude}],
        {kd, 0, 3, .08}];
```

This generates a long series of graphs. The **PlotRange** was specified explicitly to maintain a constant vertical scale from frame to frame. To prepare this series of frames to become a movie double-click on the second cell bracket, the one that covers all the frames of the movie. This will collapse the frames into a single cell.

To play such a collapsed set of frames as a movie, double click on the top image. There may be a short wait before the movie starts, especially if there are a large number of frames.

Another option with the Macintosh version of *Mathematica* is to select **Convert to QuickTime...** from the **Graph** menu while a collapsed set of frames is selected. This will write to disk a QuickTime movie built from the *Mathematica* graphics frames.

In this movie it is not immediately obvious what is varying from frame to frame. It would help if, along with each frame, there was displayed an indicator of the varying quantity. To do this make each frame a **GraphicsArray**; one element of the array will be the same plot that makes up the entire frame in the above movie, the other element of the array will be a bar graph showing the value of *kd*, the parameter that varies from frame to frame.

Like legends, bar charts are not a built in part of *Mathematica*, but are included in the standard *Mathematica* packages. First, read in the package that contains the **BarChart** function.

```
<<Graphics`Graphics`
```

Then generate the movie that contains two images in each frame.

```
Do[Show[GraphicsArray[{Plot[x[t] /. Solution,
                        {t,0,20},
                        PlotRange -> {-10, 10},
                        AxesLabel -> {Time, Amplitude},
                        DisplayFunction -> Identity},
                        BarChart[{kd},
                        PlotRange -> {0, 3},
                        AxesOrigin -> {2,0},
                        AspectRatio -> 3,
                        BarLabels -> "kd",
                        DisplayFunction -> Identity}],
                        GraphicsSpacing -> 0]],
    {kd,0,3,0.08}]
```

This large and complex command requires explanation.

```
Do[Show[...], {kd,0,3,0.08}]
```

Repeats the **Show** command for *kd* varying from 0 to 3 in increments of 0.08. The **Show** command draws the frames of the movie. Each frame of the movie is of the form

```
GraphicsArray[{Left side of the frame,
                Right side of the frame},
                GraphicsSpacing -> 0]
```

The **GraphicsSpacing -> 0** argument reduces as much as possible the separation between the left and right sides of the frame.

The left side of each frame is generated by the same **Plot** function that was used in the previous movie, except that it has the additional argument **DisplayFunction -> Identity**. This prevents the plot from being drawn to the screen at the time of creation. The frames for this

movie are drawn by the **Show** command after they have been built by the **GraphicsArray** function.

The right side of each frame is generated by the **BarChart**[*list*] function. In this case the list consists of one element, the *kd* parameter. Several optional arguments are also passed to **BarChart**: **PlotRange** \rightarrow $\{0, 3\}$ keeps the vertical scale constant from frame to frame, **AxesOrigin** \rightarrow $\{2, 0\}$ shifts the vertical axis to the right, **AspectRatio** \rightarrow 3 sets the height to width ratio, **BarLabels** \rightarrow "kd" places the kd label at the base of the bar, and as before **DisplayFunction** \rightarrow **Identity** suppresses the drawing of the plot until the **Show** function is called to display the entire **GraphicsArray**.

Section 8: References

Mathematica, A System for Doing Mathematics by Computer is the reference manual for *Mathematica*. This book describes all the capabilities of *Mathematica* and gives a large number of simple examples of their use. Copies of this manual are available in the Student Microcomputing Facility, SMF 223, the SMF documentation room, for in lab use.