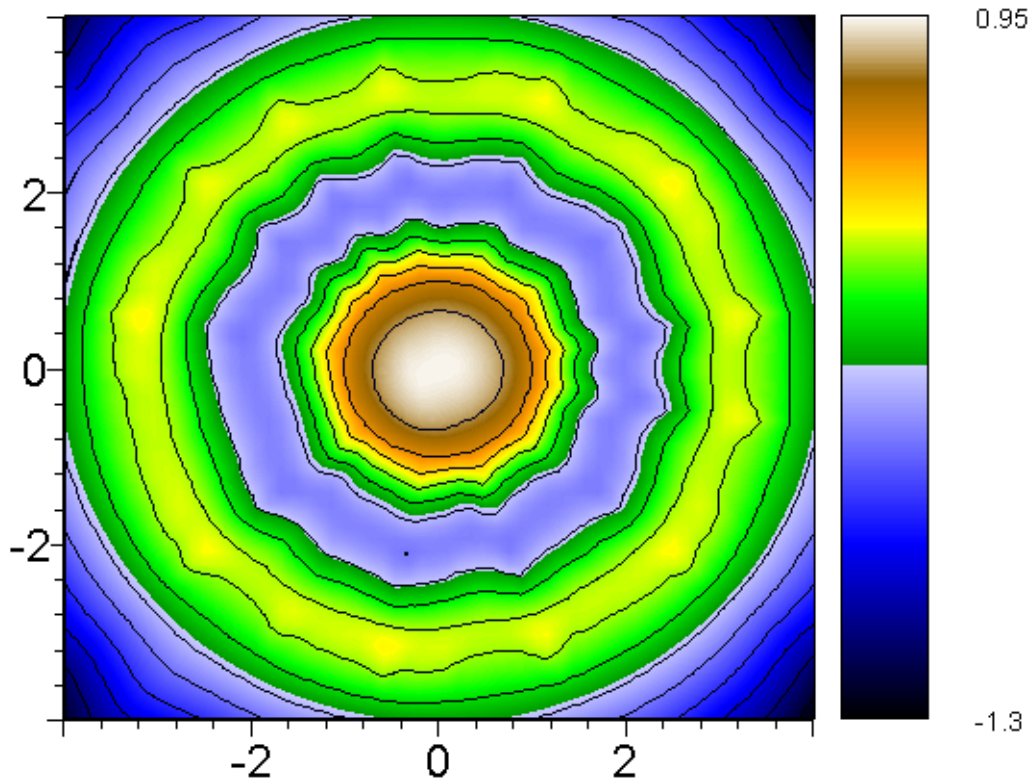


Extrema Tutorial

Using Colormaps and Gradients



INTRODUCTION

Extrema uses an indexed color model. This means that the colors are stored in an array, and referenced by their index number in that array. Since colors are treated as index numbers, you can select colors algorithmically to make sophisticated use of color in your drawings.

Note that indexed color image formats, such as GIF, usually have a maximum number of colors (typically 256, which is why they are called 8-bit color graphics). Extrema will often use colormaps with 256 or fewer colors, which will be compatible with these graphic formats. However, Extrema has no upper limit on the number of colors it can index, and you can devise colormaps with thousands of colors, if you need them. Such images will only reproduce faithfully using image formats and displays that support 16- or 24-bit color, such as JPEG or PNG. (Extrema can export its drawings to JPEG and PNG format.)

Extrema has two groups of colors that it works with; a standard set of colors that are always defined, and a dynamically loaded colormap that can be changed at will. Dynamically-loaded colors use color indexes starting at 1 and increasing to the number of colors in the colormap. Standard colors use index numbers starting at 0 and decreasing (to -22, since there are 23 standard colors).

STANDARD COLORS

The 20 standard colors that are always defined in Extrema are:

white 0	black -1	red -2	green -3	blue -4
purple -5	yellow -6	cyan -7	brown -8	coral -9
salmon -10	sienna -11	tan -12	fuchsia -13	lime -14
maroon -15	navy -16	olive -17	silver -18	teal -19

These standard colors can be referenced by name, or by index number. For instance, if you want to draw in red, select color "RED" or color number -2. This will always give you red, no matter which colormap you are currently using.

MANUAL SETTING OF COLORS

At any time you can set the drawing color using the command:

```
SET COLOR color
```

where `color` is the color index or color name (in the case of a standard color).

You can also predefine the colors that will be used for standard drawing features, such as axes and labels, using the color characteristics for those features:

```
AREAFILLCOLOR, LEGENDTITLECOLOR, CURVECOLOR, PLOTSYMBOLCOLOR, XAXISCOLOR,  
XLABELCOLOR, XNUMBERSCOLOR, YAXISCOLOR, YLABELCOLOR, YNUMBERSCOLOR,  
TEXTCOLOR
```

To set a color characteristic, use the command

```
SET characteristic color
```

Using the Extrema GUI, you can set colors by selecting the color for the drawing object from the displayed palette, when it is provided to you. The displayed palette shows you all of the

currently available colors, which includes the standard colors, plus the currently loaded colormap.

COLORMAPS

In addition to the predefined standard colors, there are a number of predefined colormaps. If none of these is satisfactory, you can load up your own colors using a colormap file.

A colormap is a set of RGB (Red, Green, Blue) values -- the red, green, and blue components combine to make a particular color on the actual drawing, according to the principles of additive color theory. An individual RGB value is expressed using hexadecimal notation (values from 00 to FF). For instance, this is the RGB value for white:

```
FF FF FF
```

Here is the RGB value for yellow (i.e., a combination of red and green):

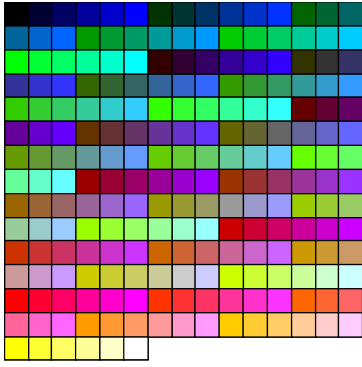
```
FF FF 00
```

Predefined Colormaps

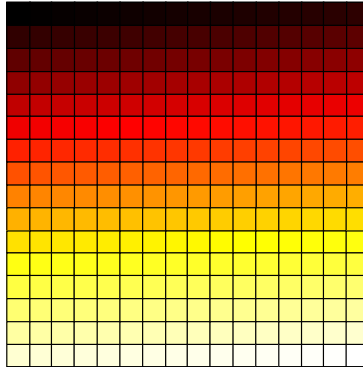
Extrema has a number of predefined colormaps, which are chosen by name using the following command

```
SET COLORMAP colormapname
```

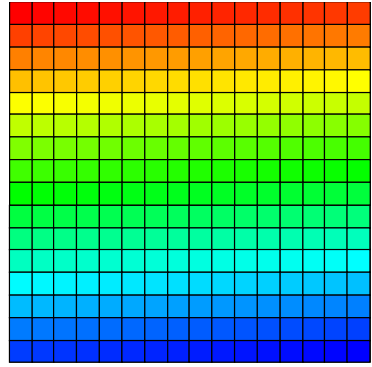
The colormap called `DEFAULT` is pre-loaded whenever you start Extrema. The predefined colormaps are displayed below.



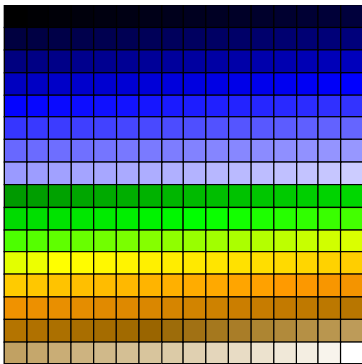
DEFAULT



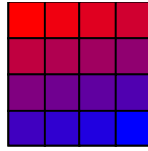
THERMAL



RAINBOW



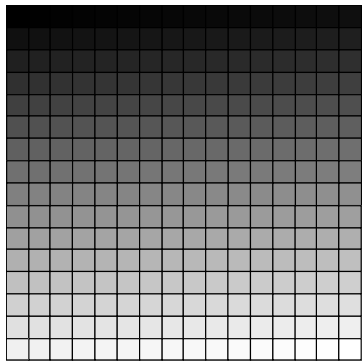
TOPO256



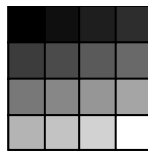
REDBLUE16



CLASSIC8



GRAY256



GRAY16

The `DEFAULT` colormap is the so-called “Netscape palette,” which is the set of all colors made by combining red, green, and blue at the following intensities: 0, 20, 40, 60, 80, and 100%. With these 6 color levels in each of the 3 RGB color channels, this gives a set of 216 colors to work with, which is enough for most colored plots. This is also the most standardized set of colors for faithful reproduction on the web. Other colors might not

reproduce well on 8-bit color displays. (Note that Extrema's standard colors also conform to the Netscape palette, and can be safely used for web graphics.)

The colors in the default colormap are produced by cycling upward through the 6 values in each of red, green, and blue. Blue cycles the fastest and red the slowest. In other words, the first 8 colors of the default color map are:

```
00 00 00
00 00 33
00 00 66
00 00 99
00 00 CC
00 00 FF
00 33 00
00 33 33
...
```

and the last 8 colors are:

```
...
FF CC CC
FF CC FF
FF FF 00
FF FF 33
FF FF 66
FF FF 99
FF FF CC
FF FF FF
```

Note for legacy users: The colormap called CLASSIC8 reproduces the old pen-plotter palette of 8 colors. Users of the Physica package can reproduce their old color selections by loading this colormap before doing any drawing. This may be useful for comparison with old drawings, but it is rather limited if you are not working with pen plotters.

User defined colormaps

Extrema allows you to create your own colormaps. A user created colormap has the name USERDEFINED. There are two ways to generate your own colormap. The simplest method is to create three vectors, say *rvec*, *gvec*, and *bvec*, containing the R, G, B color values and enter

```
SET COLORMAP rvec gvec bvec
```

You can also write your R, G, B color values in hexadecimal format to a plain text file and enter

```
SET COLORMAPFILE filename
```

A colormap file provides a sequence of RGB values, one per line. These are read into Extrema's internal colormap, where they are indexed in the same order. That is, line number 1 in the file is color number 1 in the colormap, and so on. If you load a new colormap, it replaces all the indices of the previous one. It does not replace the standard colors, since those use negative index numbers

Color order is not important in a regular colormap, but you may want to make note of which color numbers are which if you will ever need to invoke particular colors on demand from a script or command. Using the GUI, you will always be able to browse all available colors and select them using the mouse.

Colormap Example #1

Say you need to plot 50 different sets of X-Y data points against each other on the same scatterplot, for comparison of data sets. Each data set needs to be visually distinct, but using different shapes (plotting symbols) for each one is problematic, since the symbols have to be large enough to read and yet small enough that they don't overlap and interfere with each other. Using colors to distinguish between data sets is a better option, since it keeps the points small and simple.

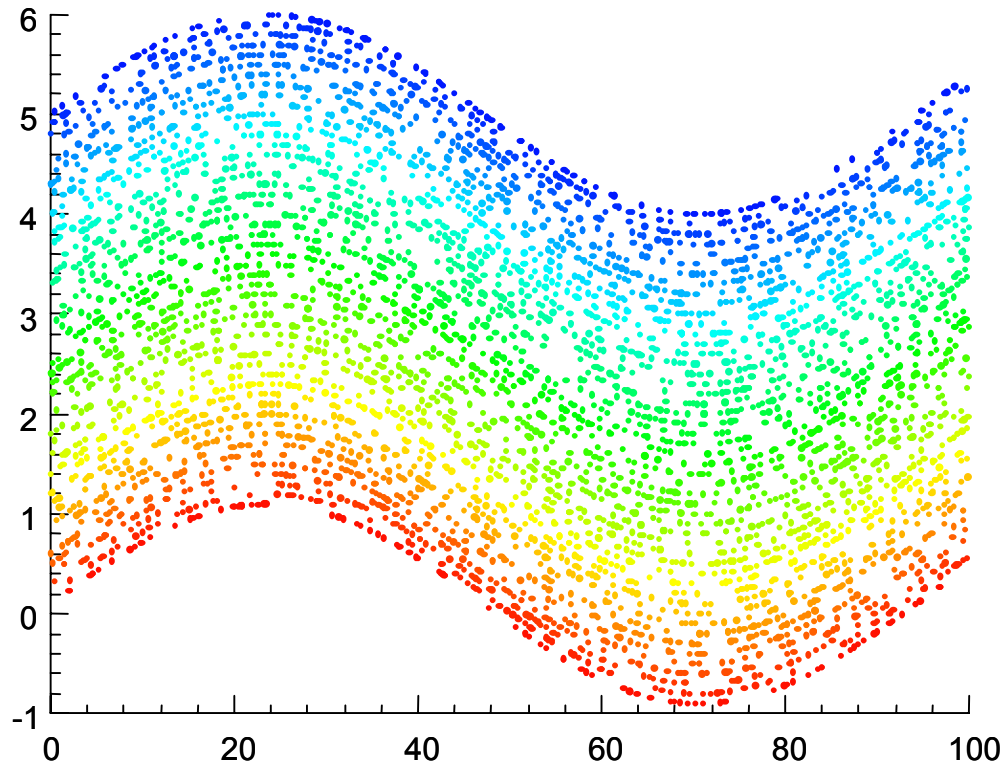
If your data exists in separate files, you'll need to load the data from each file, increment the color number, and then plot the X-Y pairs. If the data file names are stored in a text array, FILES, you could do something like this:

```
SET PLOTSYMBOL -11          ! disconnected points
DO I = [1:50]
  READ FILES[I] X Y
  SET PLOTSYMBOLCOLOR I
  GRAPH X Y
ENDDO
REPLOT                      ! to plot all points on common scale
```

If the data is organized into X and Y matrices, where the columns correspond to the data sets, then you could do something like this:

```
SET PLOTSYMBOL -11
DO I = [1:50]
  SET PLOTSYMBOLCOLOR I
  GRAPH X[* , I] Y[* , I]
ENDDO
REPLOT
```

These examples will use the first 50 colors in whatever colormap you are using. If the first 50 colors of the default colormap are not to your satisfaction, you could load your own colormap with your 50 favorite colors in it, before executing the above commands.



GRADIENTS

A gradient is just a colormap where the ordering of colors is meaningful. This is another way of saying that colors close together in the colormap are visually similar, while those that are widely separated are visually different. This is not necessarily true of the Netscape palette. For instance, in the `DEFAULT` colormap, color number 210 (light pink) has no obvious visual relationship to color 211 (bright yellow), even though the colors are derived in a logical order.

A good example of a gradient is the white light spectrum, or `RAINBOW` gradient. By setting deep red to color number 1 and violet to color number 256, you can fill in the remaining values by traversing the spectrum in small steps. Each increment produces a color closely related to those beside it, but distantly related to those further away.

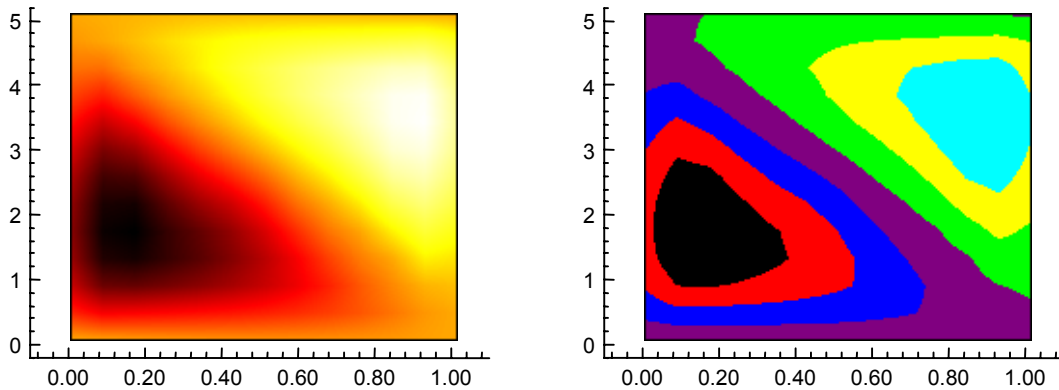
There are many ways to produce an ordered set of colors that will serve as a gradient. `Extrema` comes with 7 predefined gradient colormaps (in addition to the `DEFAULT`):

<code>RAINBOW</code>	256 colors covering the white light spectrum from red to violet.
<code>THERMAL</code>	256 colors ranging from black to red to yellow to white.
<code>TOPO256</code>	256 colors emulating topographic map colors
<code>REDBLUE16</code>	16 colors ranging from pure red to pure blue
<code>CLASSIC8</code>	8 colors reproducing the old pen-plotter palette

GRAY256 256 greyscale shades ranging from black to white

GRAY16 16 greyscale shades ranging from black to white

Note that while many of these gradients use more than 250 colors, others use only 16. In practice, your gradients can have any number of colors, from 2 to a thousand or more. A low number of colors will mean that your color scale is more granular, and may have more of a stepped or contoured appearance. A large number of colors will make transitions appear smoother. Much more than 256 colors will usually not have much noticeable effect, unless most of your graph activity occurs in a very narrow subset of the gradient.

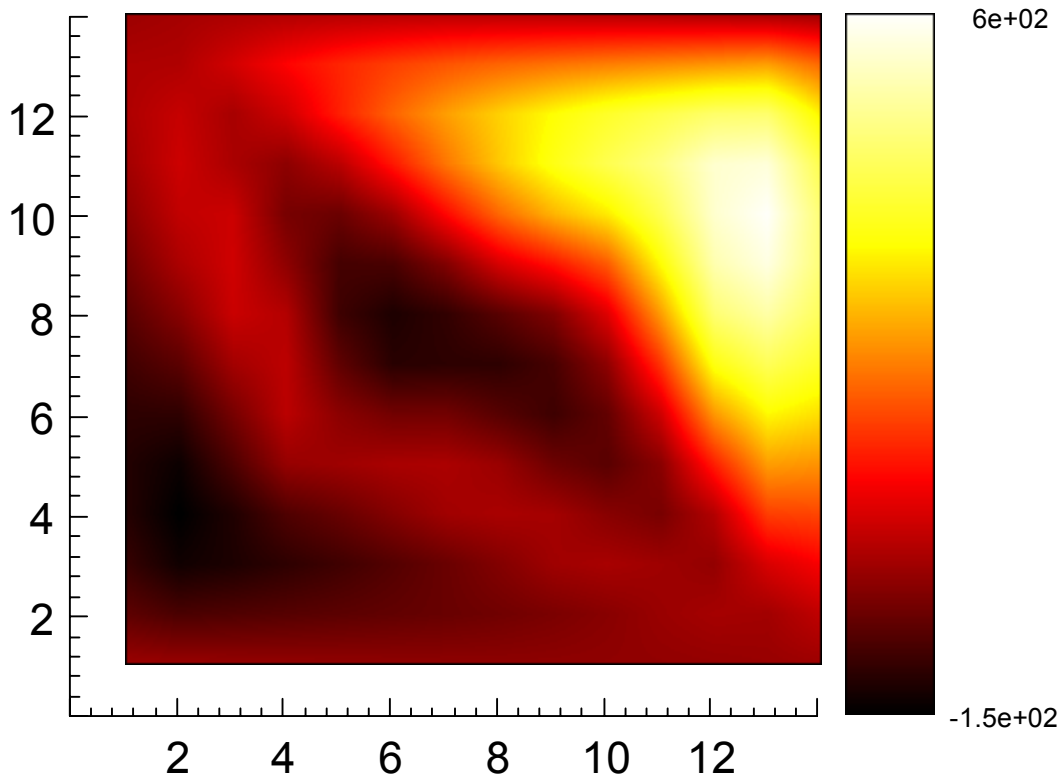


Gradient example #1

The graph type called a density plot uses full-color gradients by default. The `DENSITY` command automatically scales the graph to use the full color range of the gradient.

To make a density plot, you simply need to provide it with a matrix of data:

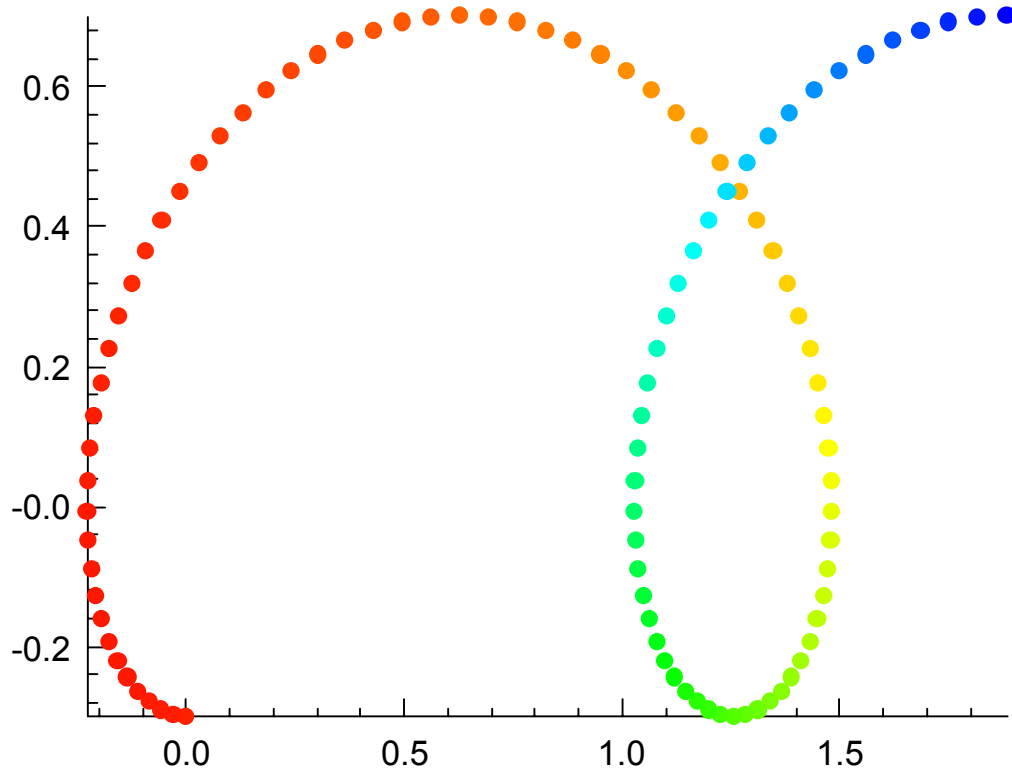
```
DENSITY myMatrix
```



Note that there are other types of density plots, which do not use color gradients. You might want to consider these for drawings that will be reproduced in a monochrome print medium.

Gradient example #2

You can also use color gradients to represent extra dimensions of data in other types of plots. For instance, say you have three vectors of data, X , Y , and T , representing X - Y coordinate pairs, and the time T at which each pair was measured. By converting the time T to a color number in a gradient, you can display all three dimensions in a simple 2-D graph:



The above drawing was produced using these commands:

```

DEFINE\CONSTANTS
GENERATE PHI 0,,3*PI 100
A=0.2
B=0.5
X=A*PHI-B*SIN(PHI)
Y=A-B*COS(PHI)
F=2*PHI^2-3*PHI+5
SET COLORMAP RAINBOW
GET COLORMAPSIZE CSIZE
C=INT(F*CSIZE/MAX(F)+0.99)
SCALES\-MESSAGES MIN(X) MAX(X) MIN(Y) MAX(Y)
SET
  PLOTSYMBOL -17
  PLOTSYMBOLCOLOR C

GRAPH X Y

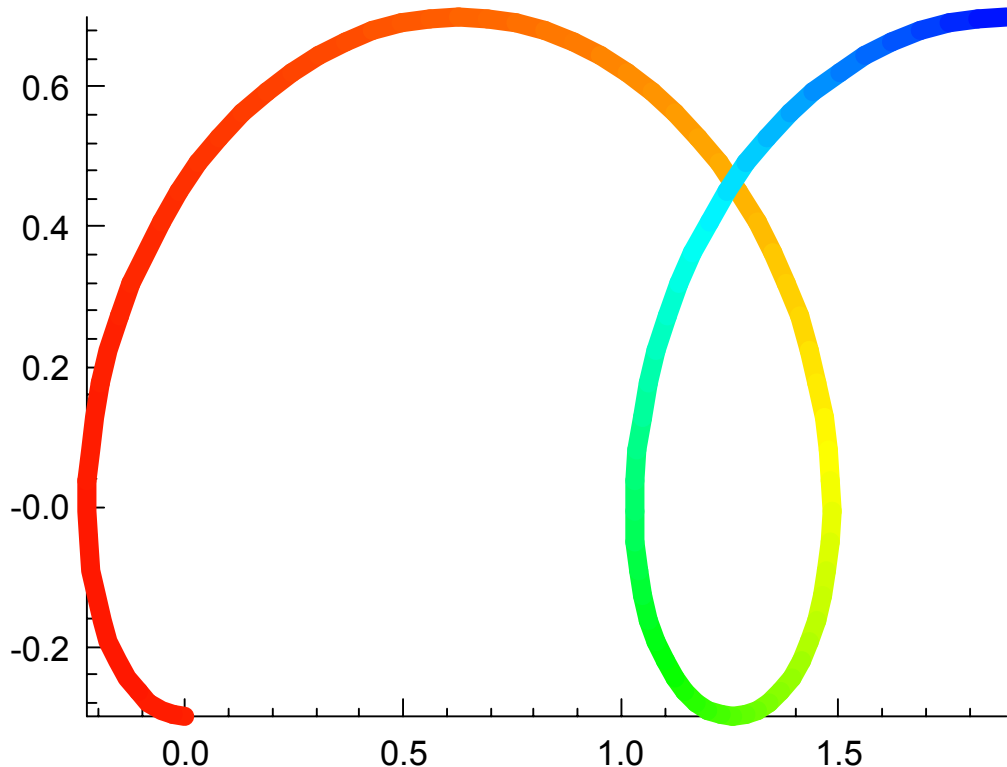
```

Gradient example #3

Consider a parametric path, $X(t)$ and $Y(t)$, and a function that assumes different values along this path, $F(t)$. You could display F using a color gradient, using a similar approach to the above. The only difference is that you may want to represent the path as a line instead of a scatterplot. Here is one way you could accomplish this:

```
DEFINE\CONSTANTS
GENERATE PHI 0,,3*PI 100
A=0.2
B=0.5
X=A*PHI-B*SIN(PHI)
Y=A-B*COS(PHI)
F=2*PHI^2-3*PHI+5
SET COLORMAP RAINBOW
GET COLORMAPSIZE CSIZE
C=INT(F*CSIZE/MAX(F)+0.99)
SCALES\-MESSAGES MIN(X) MAX(X) MIN(Y) MAX(Y)
SET
  PLOTSYMBOL 0
  CURVELINEWIDTH 10

GRAPH\AXESONLY
DO I = [1:LEN(PHI)-1]
  SET CURVECOLOR C[I]
  GRAPH\OVERLAY [X[I];X[I+1]] [Y[I];Y[I+1]]
ENDDO
```



HOW TO GENERATE YOUR OWN GRADIENTS

If you want a special color effect that you cannot accomplish with Extrema's predefined gradients, you can easily generate your own. It helps to have some basic knowledge of additive color theory to know how to combine red, green, and blue values to produce the colors you want, and how to blend them with adjacent colors.

For instance, say you need to represent the state of health of some botanical samples, and you would like a gradient that runs from brown through yellow to green. Start by noting the R, G, B values for the key colors in this sequence:

```

brown = 66 33 00
yellow = FF FF 00
green = 00 FF 00

```

Now decide how many colors you want in your gradient. Let's say it is 100. You have two color ranges to span in this example gradient (brown to yellow, and yellow to green), so you will need 50 colors in each sequence if you want the colors evenly balanced between these two ranges.

For the first sequence, brown to yellow, you need to shift the red component from hex 66 (decimal 102) to hex FF (decimal 255) in 50 steps, and the green component from hex 33 (decimal 51) to hex FF (decimal 255) in the same 50 steps. The blue component does not change.

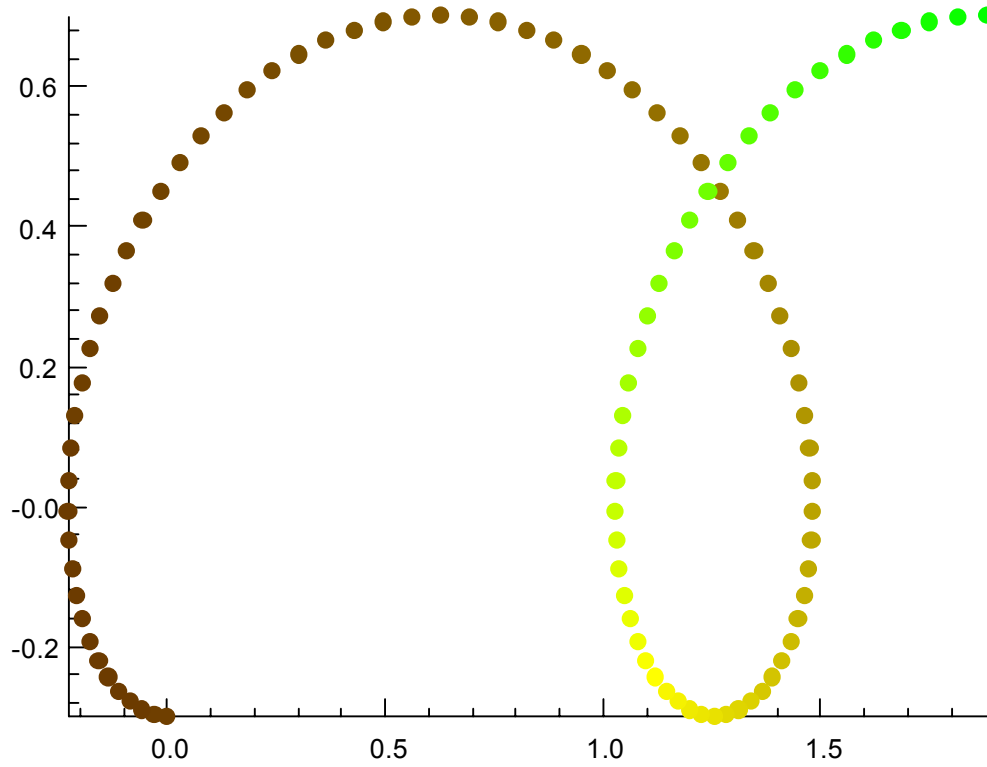
For the second sequence, yellow to green, you need to shift the red component from hex FF (decimal 255) to hex 00 (decimal 0) in 50 steps. The green and blue components do not change.

By appending the two lists that result, you get your desired colormap. Note that because the first sequence ends at yellow, and the second sequence begins at yellow, pure yellow will occur twice in the sequence (at color numbers 50 and 51). That may not be important to you, but if you need absolute precision, then simply make the second sequence progress in 51 steps, and drop the first element (yellow), since you already have it from the first sequence.

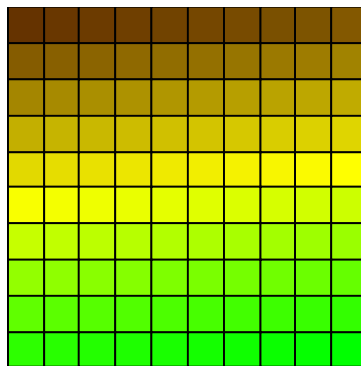
Here is an Extrema script that creates this gradient colormap and draws the graph:

```
PI=ACOS(-1)                                ! generate some "data"
GENERATE X 0,,2*PI,100
Y=SIN(X)
GENERATE T 1,,3600,100
!
GENERATE R1 102,,255 50                    ! make the R G B vectors
GENERATE G1 51,,255 50
B1[1:50] = 0
GENERATE R2 255,,0 51
G2[1:51] = 255
B2[1:51] = 0
R = INT(R1//R2[2:51])
G = INT(G1//G2[2:51])
B = INT(B1//B2[2:51])
SET COLORMAP R G B                        ! make the colormap
GET COLORMAPSIZE CSIZE
C=INT(T*CSIZE/MAX(T)+.99)
SET
  PLOTSYMBOL -17
  PLOTSYMBOLCOLOR C

GRAPH X Y
```



Here's what the user defined colormap looks like:



SUMMARY

Here are the commands you will need to use to achieve these color effects in your drawings. These may be looked up in the Extrema Command Reference for more details on their use.

Manually setting colors:

```
SET COLOR  
SET AREAFILLCOLOR  
SET LEGENDTITLECOLOR  
SET PLOTSYMBOLCOLOR  
SET CURVECOLOR  
SET XAXISCOLOR  
SET XLABELCOLOR  
SET XNUMBERSCOLOR  
SET YAXISCOLOR  
SET YLABELCOLOR  
SET YNUMBERSCOLOR  
SET TEXTCOLOR
```

Loading colormaps:

```
SET COLORMAP  
SET COLORMAPFILE
```

Density plots:

```
DENSITY
```