Neptune & Triton
Version 1.0 User Manual

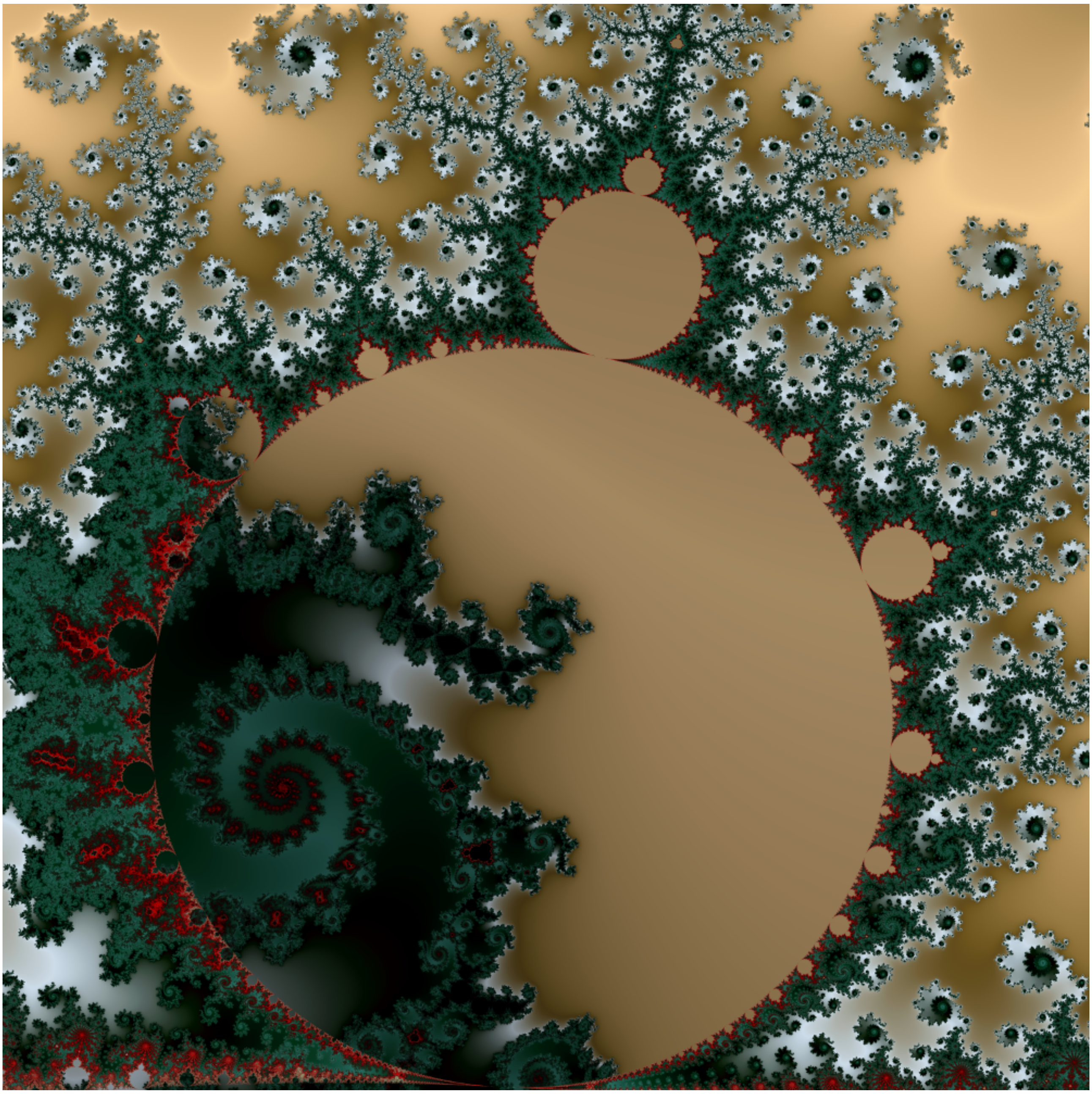


Table of Contents

1 Introduction

I am not a technical author so I apologise in advance if you find this manual confusing, I've endeavoured to make Neptune and Triton easy to use so you may find it easier to just use the software and dis-regard this manual.

Neptune and its sister program Triton are programs for generating multiple critical point fractal images. Neptune is used for exploring fractals and Triton is used for expanding images saved from Neptune.

Neptune and Triton are based on Saturn and Titan (version 4.1.0) and are named based on the same scheme for my earlier fractal programs.

Now, what is a "multiple critical point fractal"? Fractals of the Mandelbrot sort have critical points which can be used as the initial value of each location calculated using the Mandelbrot algorithm, if a value that is not a critical point is used the resulting image is distorted and often looks like it has had chunks bitten out of it such fractals is called perturbed. The Mandelbrot itself only has one critical point which is zero. The pictures produced using different critical points mostly different from each other, there can be fwer pictures produced than there are critical points, sometimes all the critical points produce the same picture. Multiple critical point fractals are produced by combining all the different pictures together in one image.

So, how are critical points determined? Critical points are found by solving the derivative of the fractal's function equated to zero.

The Mandelbrot formula is:

$$z_{n+1} = z_n^2 + c$$

the function is

$$f(z) = z^2 + c$$

its derivative

$$f'(z) = 2z$$

so the critical point is solution of

$$2z = 0$$

so the critical point is when $z = 0$. This also shows that the standard Mandelbrot set only has one critical point.

The standard cubic Mandelbrot also only has one critical point:

$$z_{n+1} = z_n^3 + c$$

the function is

$$f(z) = z^3 + c$$

its derivative

$$f'(z) = 3z^2$$

so the critical point is solution of

$$3z^2 = 0$$

so the critical point is when $z = 0$. To get more than one critical point the final equation to be solved must equal something other than zero.

So adding a z term to the cubic Mandelbrot formula:

$$z_{n+1} = z_n^3 + z + c$$

the function is

$$f(z) = z^3 + z + c$$

its derivative

$$f'(z) = 3z^2 + 1$$

so the critical points are the solutions of

$$3z^2 + 1 = 0$$

which is

$$z^2 = -1/3$$

so the critical values are the square roots of -1/3 or approximately 0.5774 and -0.5744. Each critical point produces a different picture, the fractals produced look oddly incomplete parts of them have normal Mandelbrot type buds in some areas but other areas look barren when the both pictures are overlaid on each other does the fractal become complete.

Adding a z squared term produces different critical points:

$$z_{n+1} = z_n^3 + z^2 + c$$

the function is

$$f(z) = z^3 + z^2 + c$$

its derivative

$$f'(z) = 3z^2 + 2z$$

so the critical points are the solutions of

$$3z^2 + 2z = 0$$

which is

$$z(3z + 2) = 0$$

so the solutions are 0 and -2/3.

Fortunately you do not need to work out the critical points, Neptune handles that for you.

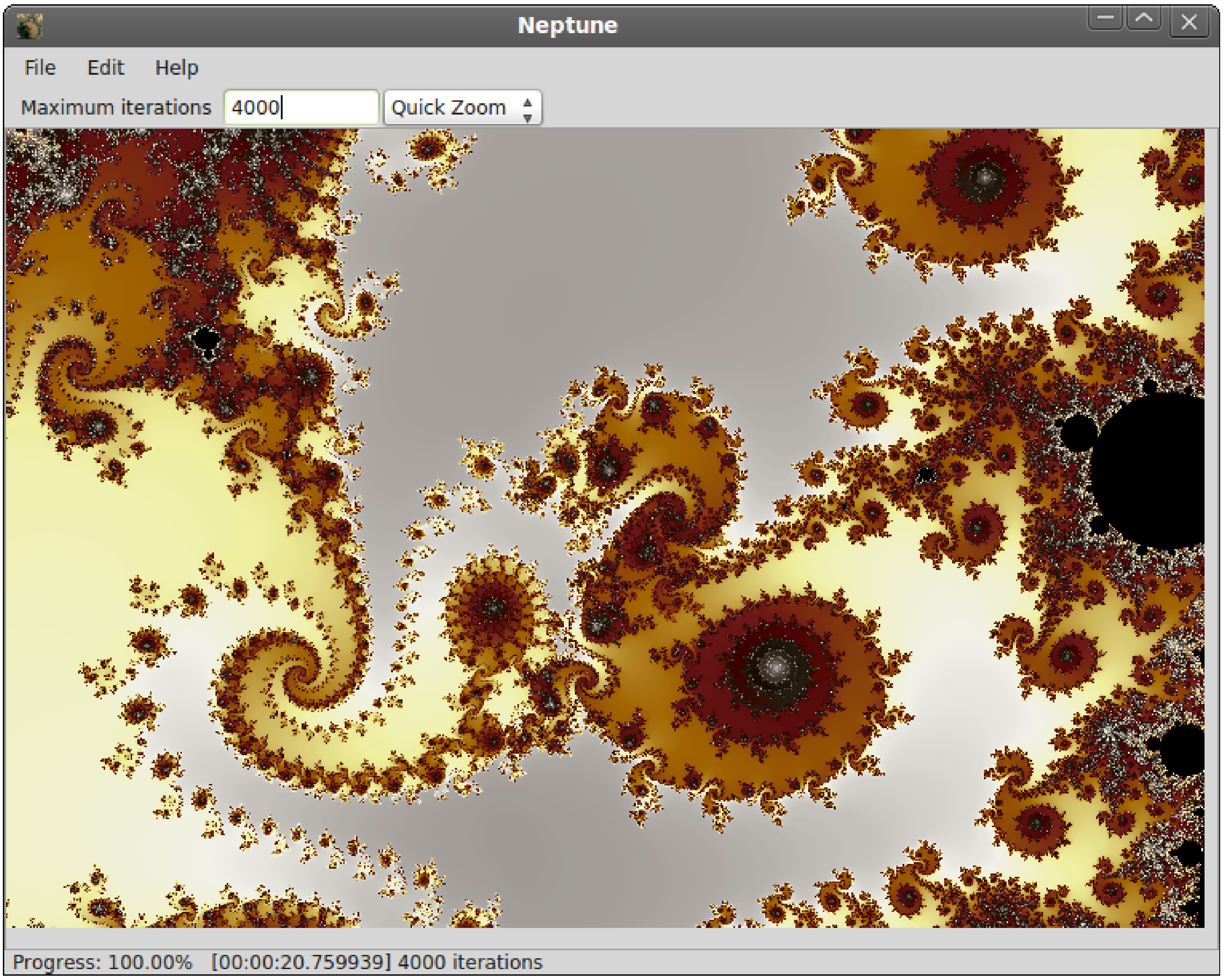
2 Definitions

seed file	a portable network graphics file (PNG) with all the necessary fractal parameters embedded in it so that it can be reproduced when it is opened by either Neptune or Triton. WARNING: if this file is processed in any way by a program other than Neptune or Triton the embedded data will be lost.
parameter file	a file containing all the parameters necessary to recreate a fractal. The data is in XML and the files have an npf extension (for Neptune Parameter File).
colour map	a colour map contains the colour definition use by the colouring method, there are three types “auto”, “manual” and component. Colour maps are embedded in seed files and included in parameter files, a maximum of 512 distinct colours can be defined in a colour map.
colour map file	Neptune can import colours from 3 different file types “.map” (FractInt), “.ugr” (Ultra Fractal) and “.scm” (Saturn Colour Map). Saturn can also export colour maps as either “.ugr” or “.scm” files.
z	the iterating complex variable
i	The imaginary unit number such that i^2 is equal to -1.
Greek letters	The Greek lower case letters α, β,γ, δ, ε, ζ, η, θ these are the variables used as parameters in many fractal formulae in the formula section of the fractal tab, they are also used with .r and .i suffixes. In the summary tab of Titan however their names are used instead alpha, beta, gamma etc.
abs()	when applied to a complex number z the result is the magnitude, i.e. the square root of z.r*z.r + z.i*z.i. when applied to a real number it forces a negative value into a positive.
norm()	the norm of a complex number is z.r*z.r + z.i*z.i

3 Neptune

3.1 Main Window

Here is an example of the main window:



3.1.1 Menu

The menu consists of, File, Edit, View and Help.

File sub menu:

- Open – this can be used to open a seed or parameter file. The file contains the definition of the fractal and the colour map used.
- Save image … – save the currently displayed fractal image as a seed file.
- Save parameters … – save the definition of the currently displayed image as a parameter file.
- Import Colour Maps … – this option displays the import colour map window.
- Export Colour Maps ... – this option displays the colour map export window.
- Quit – exit Neptune.

Save image and save parameters both display a file save dialogue, the name used to save the file will have either .png or .npf appended if the extension is missing from the file name. The file saved by the image option is a seed file which can be loaded in Neptune or to create a much larger image using Titan. The parameter file is a text file containing the parameters to recreate the fractal and are defined in XML.

Parameters files are much smaller than seed files and are easier to share, any processing of a seed file will result in the parameter data being lost resulting in an ordinary png file. Parameter files can only be loaded into Neptune, if you you have an npf file which you want to create a large version of the fractal you must use Neptune to save a seed file which can then be used by Triton.

Edit sub menu:

- Fractal Settings … - this option will display the fractal settings window.
- Revert – revert to the starting complex plane, see the position tab of the fractal settings window.

The help menu just provides an about option which displays version information.

3.1.2 Above the Fractal Picture

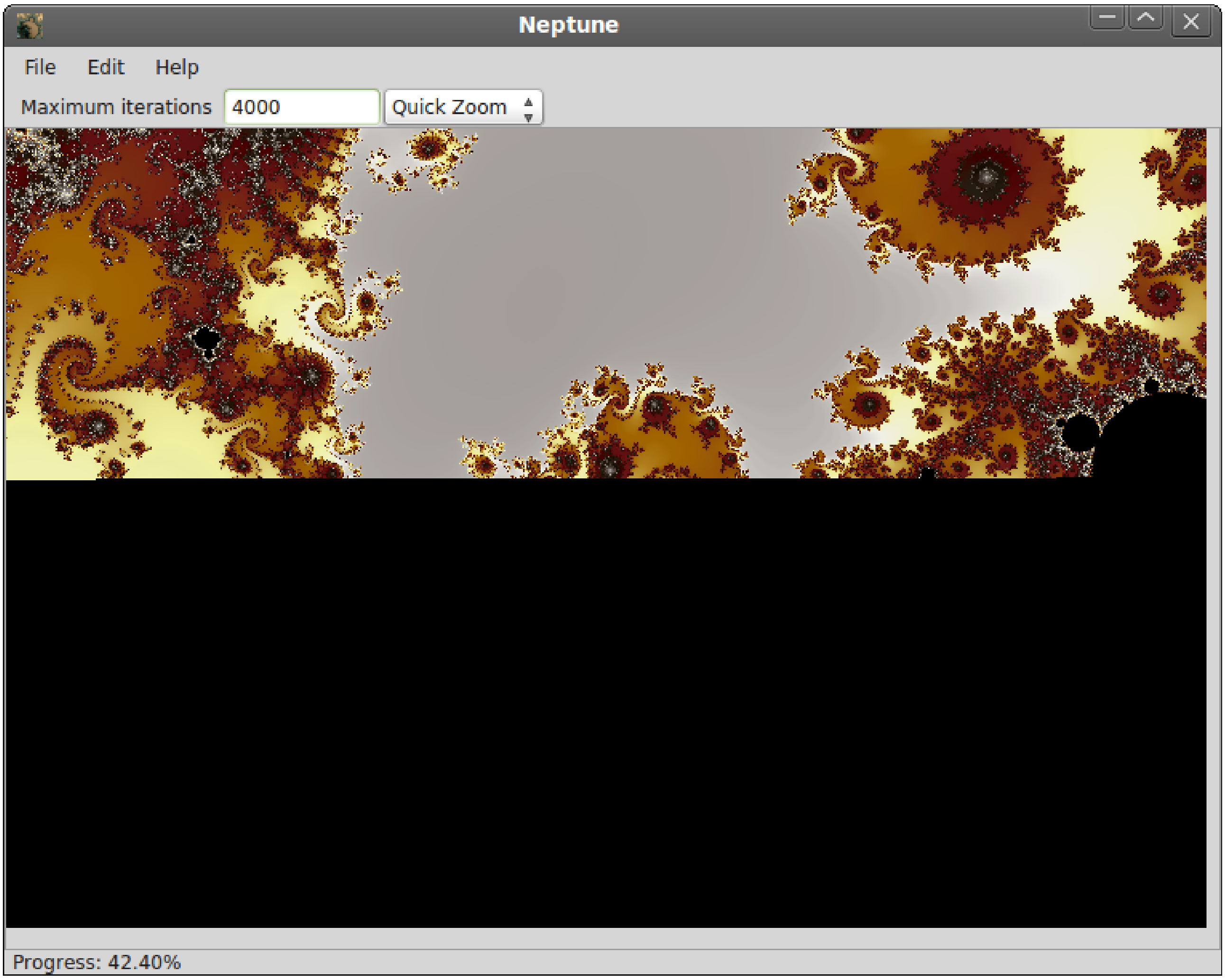
There are several items above the fractal picture and are context sensitive. The items are:

1. One of “Maximum iterations” or “Orbit Length”.
2. An entry box showing the maximum number of iterations, orbit length or number of calculating cycles.
3. Zoom type selection: “Quick Zoom”, “Zoom In”, “Centre In”, “Centre Out” and “Zoom Off”.
4. Zoom button, displayed when the zoom type is “Zoom In”, “Centre In” or “Centre Out”.
5. Centre button, displayed when the zoom type is “Centre In” or “Centre Out”.

The value in the entry box can be changed at any time and fractal calculation will restart.

3.1.3 Image

The fractal image is drawn “row-by-row”.

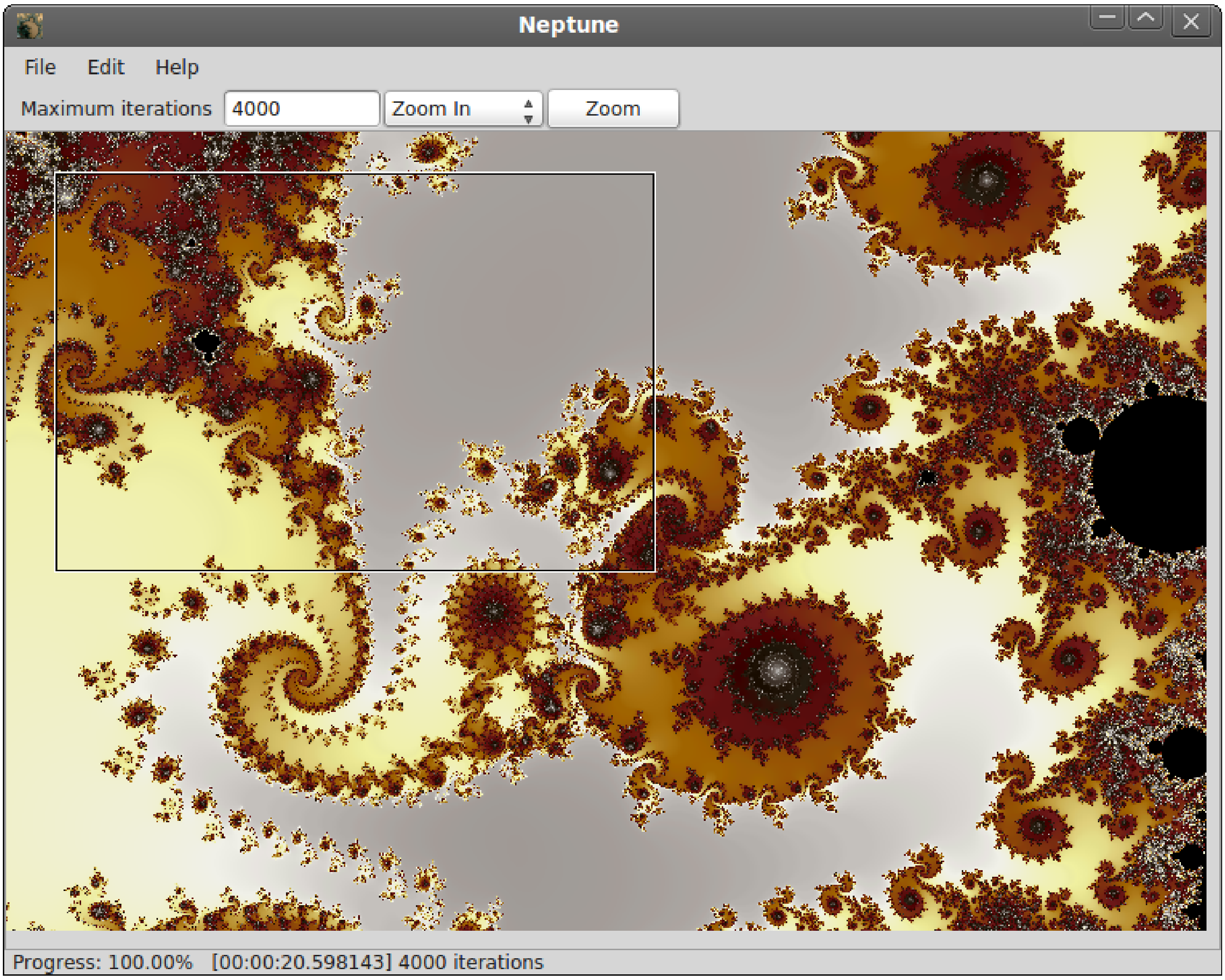


The area can also be used as a target for drag and drop of a seed or parameter file, the file will be opened and the fractal reproduced. If a png file which does not contain the embedded fractal data is dropped on the area a dialogue box will displayed informing the user that it does not contain the required fractal data. Files with extensions other than png or npf will be ignored.

Example seed files can be downloaded from http://element90.wordpress.com.

3.1.4 Zooming

Zoom rectangles are drawn as two rectangles, a white lined rectangle with a blacked lined rectangle within so that it can always be seen regardless of background.



3.1.4.1 Quick Zoom

A zoom rectangle is drawn while the left mouse button is clicked an held down, the rectangle changes its dimensions as the mouse cursor is moved (it will disappear if it is too small). When the mouse button is released the picture is redrawn. This zoom type also allows a quick zoom out by clicking on the right mouse button, the image is recentred on the position of the cursor when the mouse button was clicked and the magnification is reduced by a factor of 4 (i.e. the dimensions of the complex plane are doubled).

3.1.4.2 Zoom In

When the left mouse button is clicked a zoom rectangle half the dimensions of the picture is displayed centred on the position of the the mouse cursor.

The rectangle can be grabbed when the cursor changes to an open hand inside the zoom rectangle, clicking the left mouse button grabs the rectangle which can then be moved while the mouse button is held down. While the rectangle is grabbed the mouse cursor is a closed hand. The rectangle is released when the the mouse button is released.

The size of the zoom rectangle can be changed “grabbing” one of the sides, when the cursor is near a side a double headed cursor will be displayed, the size of the rectangle can be altered by clicking the left mouse button and holding it down and moving the mouse. Releasing the left mouse button releases control of the zoom rectangle's size.

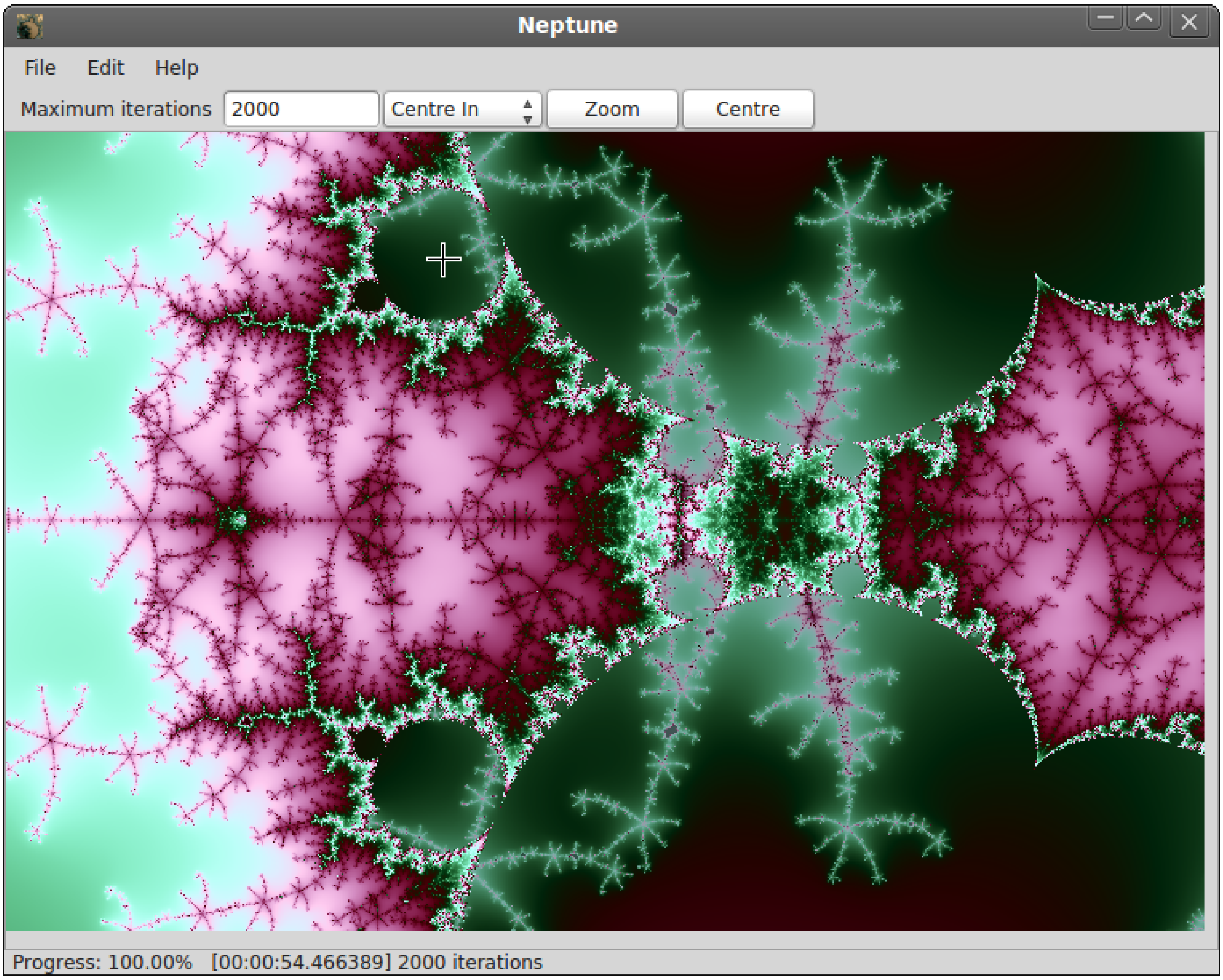
Calculation of the new zoomed image will only start when the zoom button is pressed.

The zoom rectangle can be dismissed by clicking the right mouse button or by changing the zoom type.

3.1.4.3 Centre In and Centre Out

These zoom modes are for recentring the image and for recentring the image and zooming in or out. The zoom factor is always 4 i.e. doubling or halving the complex plane dimensions.

Clicking the left mouse button displays a cross.



The cross can be grabbed when the mouse cursor changes to an open hand (over the cross), clicking the left button grabs the cross and it can be moved while the button is held down, the cursor changes to a closed hand while it's grabbed, the cross is released by releasing the mouse button and the cursor changes.

Calculation of the new zoomed image is started when either the zoom button is pressed. To just recentre the image the centre button is pressed.

The centring cross can be dismissed by pressing the right mouse button or changing the zoom type.

3.1.5 Status Line

The status line is used to show how the generation of a fractal has progressed and consists of several items:

1. Percentage progress.
2. Precision, displayed only when multi-precision calculation is being used.
3. Duration enclosed in square brackets and is displayed when calculation is complete.
4. Maximum number of iterations reached.

3.2 Fractal Settings

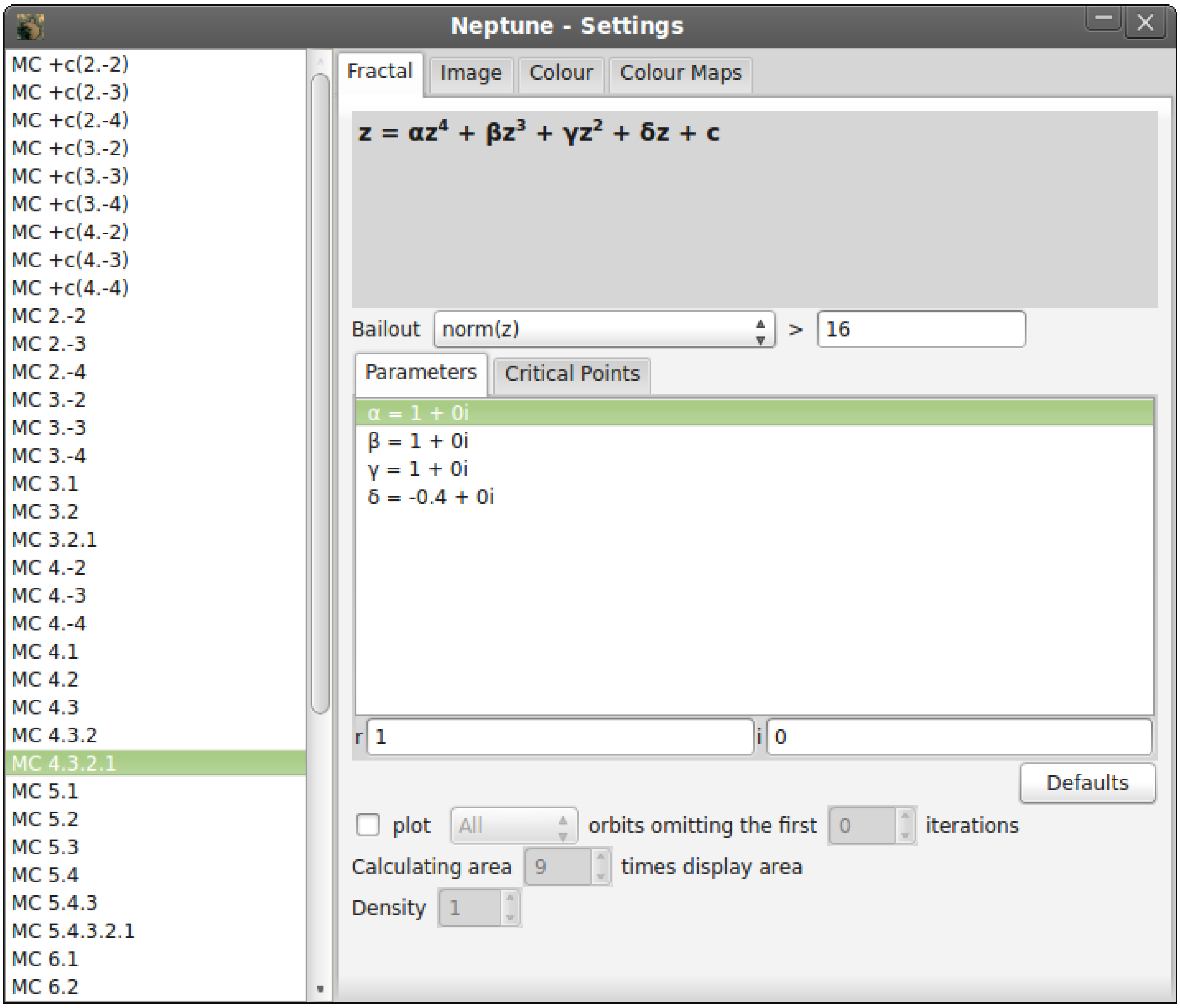
The fractal settings window consists of two parts, on the left there is a list and on the right there are tabs, the list contains the names of fractal types for all tabs except for the Colour and Colour Maps tabs when it contains the names of colour maps.

When it is a fractal type list the cursor will be displayed on the current fractal type i.e. the type currently used to generate the image in the main window. Changing the cursor with change the current fractal being displayed.

When it is a colour map list the cursor will be on the current colour map to be edited, or the colour map used by the fractal's colour methods. The cursor is removed if the colour type is a single colour.

3.2.1 Fractal Tab

The fractal tab showing the fractals's parameter list:



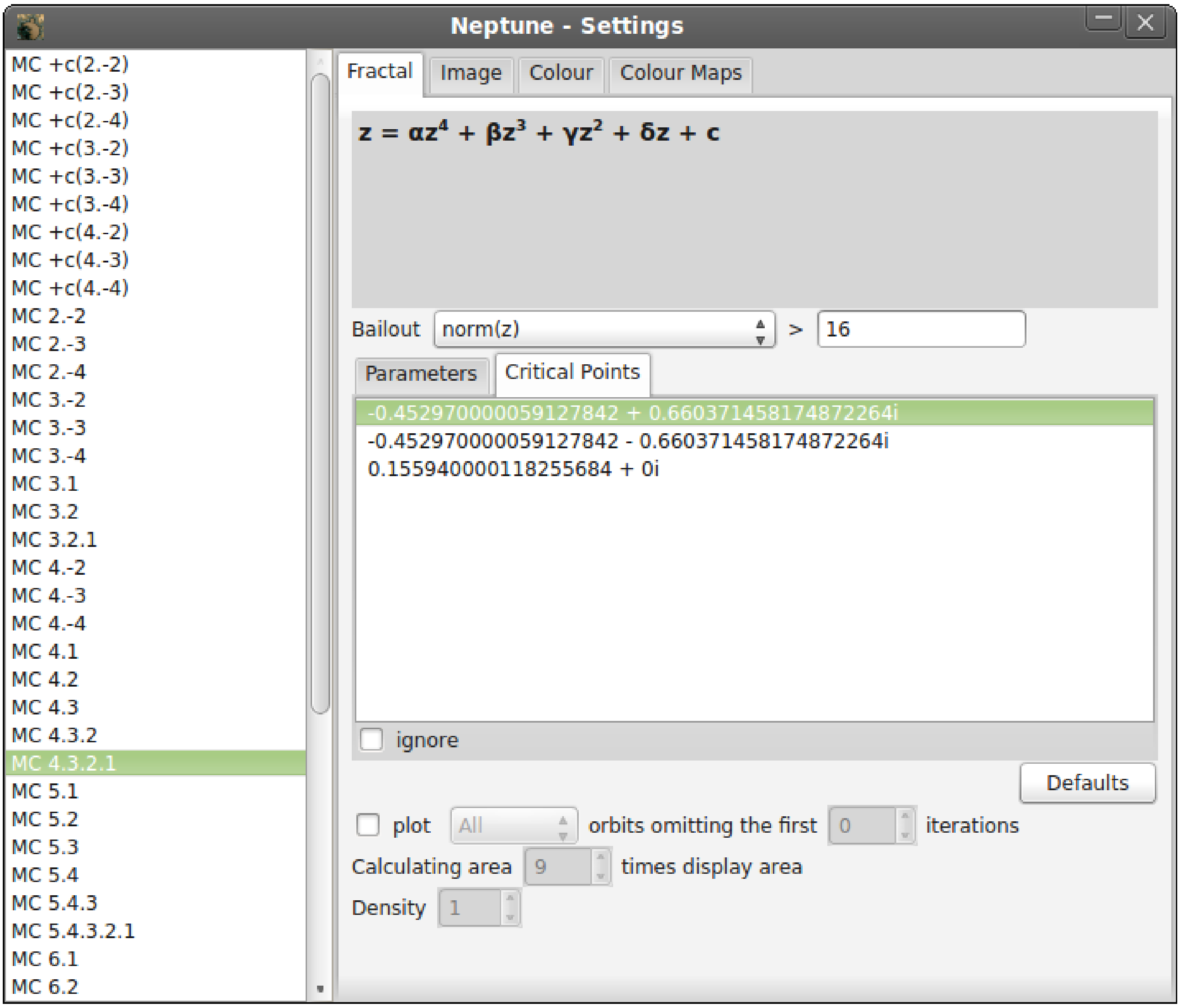
The formula

The formula used to generate the fractal, showing the parameters used. Parameters are identified by Greek lower case letters.

Parameter Tab

A list of the parameters defined for a fractal. The cursor is used to select the parameter to be changed and the new real and/or imaginary values are entered in the r or i entry boxes. When the ne value has been accepted calculation of the fractal will be restarted and the list updated.

Critical Points tab



The critical points tab lists all the critical points for the fractal. Each critical point produces a picture which up the final image, however the same image can be produced by more than one critical point when this happens more calculation than are necessary are performed. The ignore option can be set so that fewer calculations are performed. Of course an “incomplete” final image can be produced using fewer critical points if desired. At least one critical point must be defined so it is not possible to ignore all the critical points.

The critical points are recalculated whenever the fractal's parameters are changed.

Plot orbits

The orbits of the fractal are plotted when this option is used. There are three plot variations, all orbits, escaped orbits and captive orbits, a number of iterations at the start of each of the orbits can be omitted from being plotted.

The nature of orbit plotted fractals means that detail is lost as you zoom into the picture as orbits that would form part of the picture are never calculated. To reduce the effect, the area used for calculating the fractal can be increased up to a maximum of 1000 times the display area. The larger the calculating area the greater the time required to generate the fractal picture.

Increasing the density value increases the number of orbits calculated for the calculating area, e.g. fo a density 100 times more orbits are plotted than when the density is 1.

The orbit plot option is disabled if there are complex plane transformations defined for the fractal.

Bailout Condition

The bailout condition used for the fractal. The available bailout conditions are as follows:

- abs(z)
- norm(z)
- imag(z)
- imag(z) squared
- real(z)
- real(z) squared
- abs(real(z)*imag(z))
- abs(real(z)) + abs(imag(z))
- real(z) + imag(z)
- abs(real(z) + imag(z))
- max(real(z), imag(z)) squared. See Note 1
- min(real(z), imag(z)) squared. See Note 1
- abs(real(z) – imag(z))

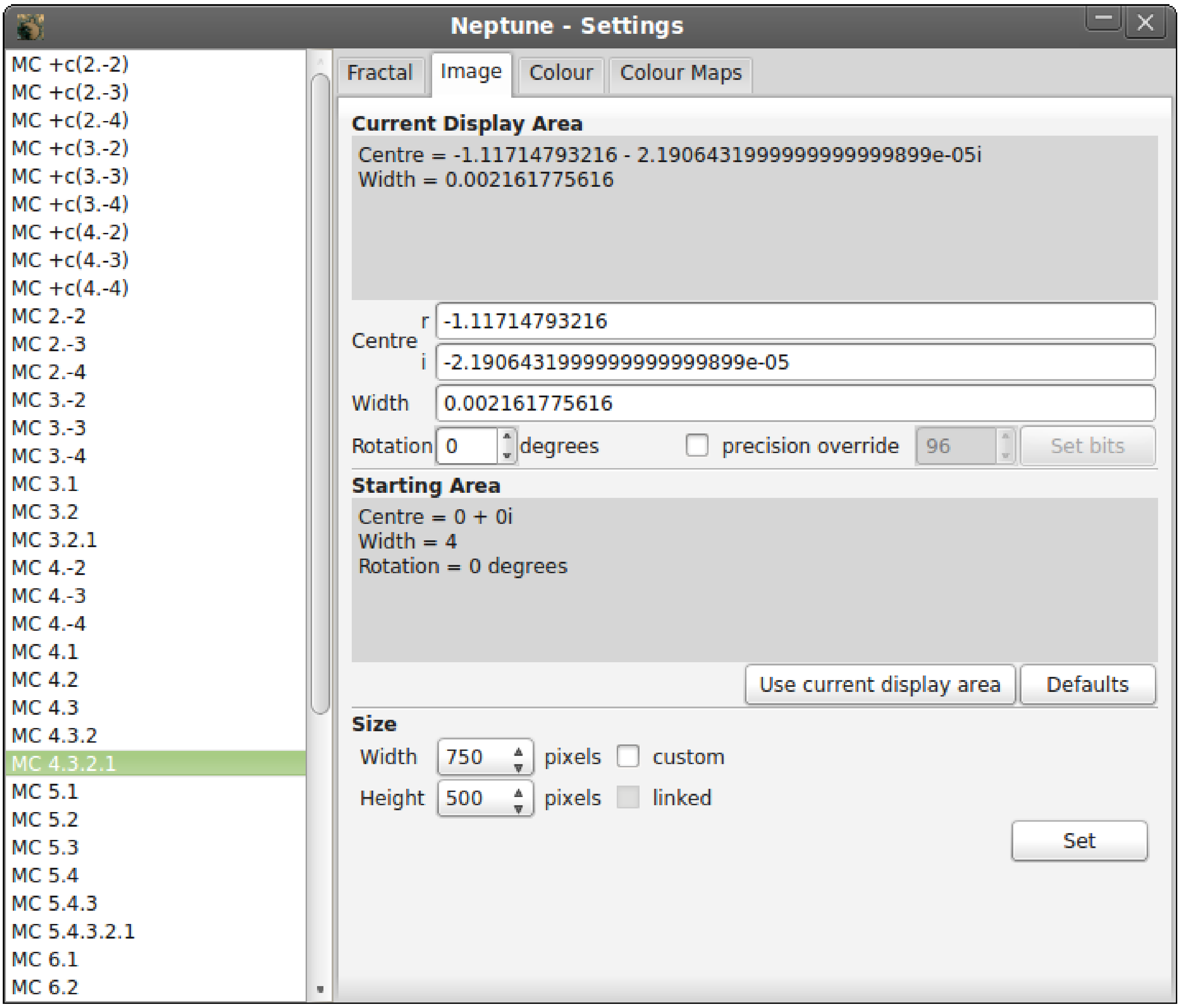
So, for example if the bailout condition is norm(z) and the limit is 16 the following is evaluated:

norm(z) > 16

if the result is true then calculation for the location has completed.

Note 1: the component parts are squared BEFORE determining maximum or minimum.

3.2.2 Image Tab



Current Display Area

There is a read only text area which displays the location of the fractal in terms of the centre of the image and the width. All available digits are displayed, the number of digits will increase as the precision used to calculate the fractal is increased. The text area will include a vertical scroll bar when the text becomes too large for the display area.

Below the text area are the entry boxes for setting the location of the fractal manually. The mechanism for determining the required precision breaks down for some fractals, the symptoms are blocky or fuzzy images, the precision can bet set manually using the precision override option, the values are in intervals of 16 bits starting at 96 (80 on Windows).

The data in the the Current Display Area are automatically updated when zooming and centring the image in the main window.

Starting Area

The starting area displays the initial default location and width of the fractal. The current location and width are set to the starting area values when the revert option in the edit menu is used. The location to revert to can be changed by setting it to the current display location so that whenever the revert option is used the fractal will be calculated for that “starting position”, the starting position can be changed back to the system default at any time.

Size

The size defines the dimensions of the picture in pixels, there are a number of pre-defined widths and heights. In addition custom values can be used, setting the custom option changs the combboxes to spin boxes, the values can be changed independently or linked so that the aspect ratio is maintained.

The new size is only used when the set button is pressed.

3.2.3 Colour Tab

The appearance of the colour tab is dependent on whether or not the fractal is orbit plotted. The list on the left hand side displays the names of colour maps instead of fractal types.

3.2.3.1 Colour – Common Settings

The colour tabs have some common settings relating to the use of colour maps.

Colour offsets

With the exception of fixed colour all colouring methods have colour offset spin buttons. There are three offsets one for each component, the values can be varied independently or in lock-step (i.e. if one is changed the other two are also changed). The colour offsets affect the starting positions in the map for each colour component, the effective range of colours is changed and the new set of colours is displayed in one of the colour map rectangles at the bottom of the tab.

Colour Component Order

Each colour in a colour map consists of three values assigned to red, green and blue (RGB). The values can be assigned to different colours for example GRB uses the red value for green and the green value for red. The are six different component order combinations which effectively increases the number of colour maps available. This may not result in six different sets of colours, for example greyscale colour maps with remain the same regardless of the colour component order,

Colour Map Rectangles

At the bottom of the colour tab there are either one (for Orbit Plot colouring) or two colour map rectangles, the upper colour map is for outer or positive (Lyapunov) colouring.

The colour offsets and component order affect how colours are extracted from the map, the colour map rectangle is 512 pixels wide and each pixel column represents a colour in the map starting at index 0 and finishing at index 511. The colours in the colour map are adjusted according to the colour offsets and component map and the resulting sequence of colours is displayed in the colour map rectangle.

Scale

The frequency in which the colours change can be altered by changing the scale value.

Smoothing

Most colour methods convert a number into a colour, the number is used to look up a colour from a colour map where 512 individual colours are defined. The number does not map precisely on to a colour in the colour map, instead it will fall between one colour and the next, if the smooth option is enabled the colour used is a blend of the colour below and the colour above the number, if smooth is not enabled the lower colour only is used. When blending colours more weight is given to the closer colour, for example if the first colour is yellow and the second is red the result will be an orange, if the number is closer to yellow then a yellower shade will be used.

For iteration colouring there is a direct match to individual colours, enabling smooth alters the number used to find the colours such that it falls between colours in the colour map and a blended colour is used. The alteration to iteration is detailed in the section on iteration colour.

Absolute log

When this option is enabled the absolute log of the value is used to determine the index of the colour in the colour map. If the logarithm of the value is negative the sign is removed. If the logarithm of the value is complex then magnitude of that complex number is used.

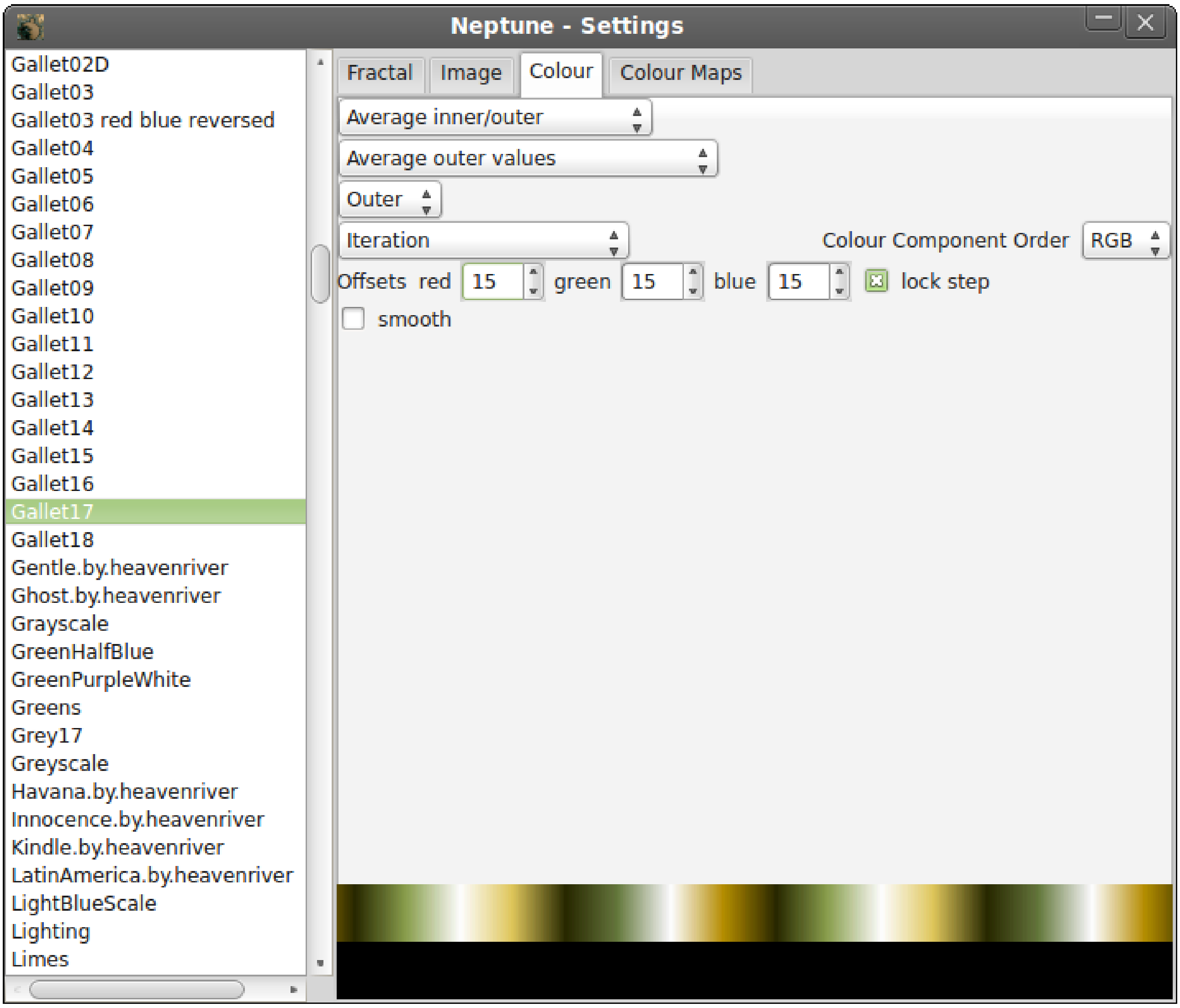
Result Statistics

For many of the colour methods a set of statistics is collected, the colours used are dependent of the statistic used. The statistics are:

1. Minimum
2. Iteration @ Minimum
3. Maximum.
4. Iteration @ Maximum.
5. Range.
6. Average.
7. Variance.
8. Standard Deviation.
9. Coefficient of Variation.
10. Fractal Dimension.
11. Exponential Sum.
12. Exponential Inverse Change Sum.

A different set of statistics is used for Gaussian Integer.

3.2.3.2 Colour – Escape Time Fractals



There are two comboboxes controlling how the different critical point pictures are combined. One for combining areas that are coloured as “outer” and the the other for combining areas where “outer” and “inner” colouring overlap. Areas that are only coloured as inner the colouring data is averaged.

Handling inner/outer overlaps:

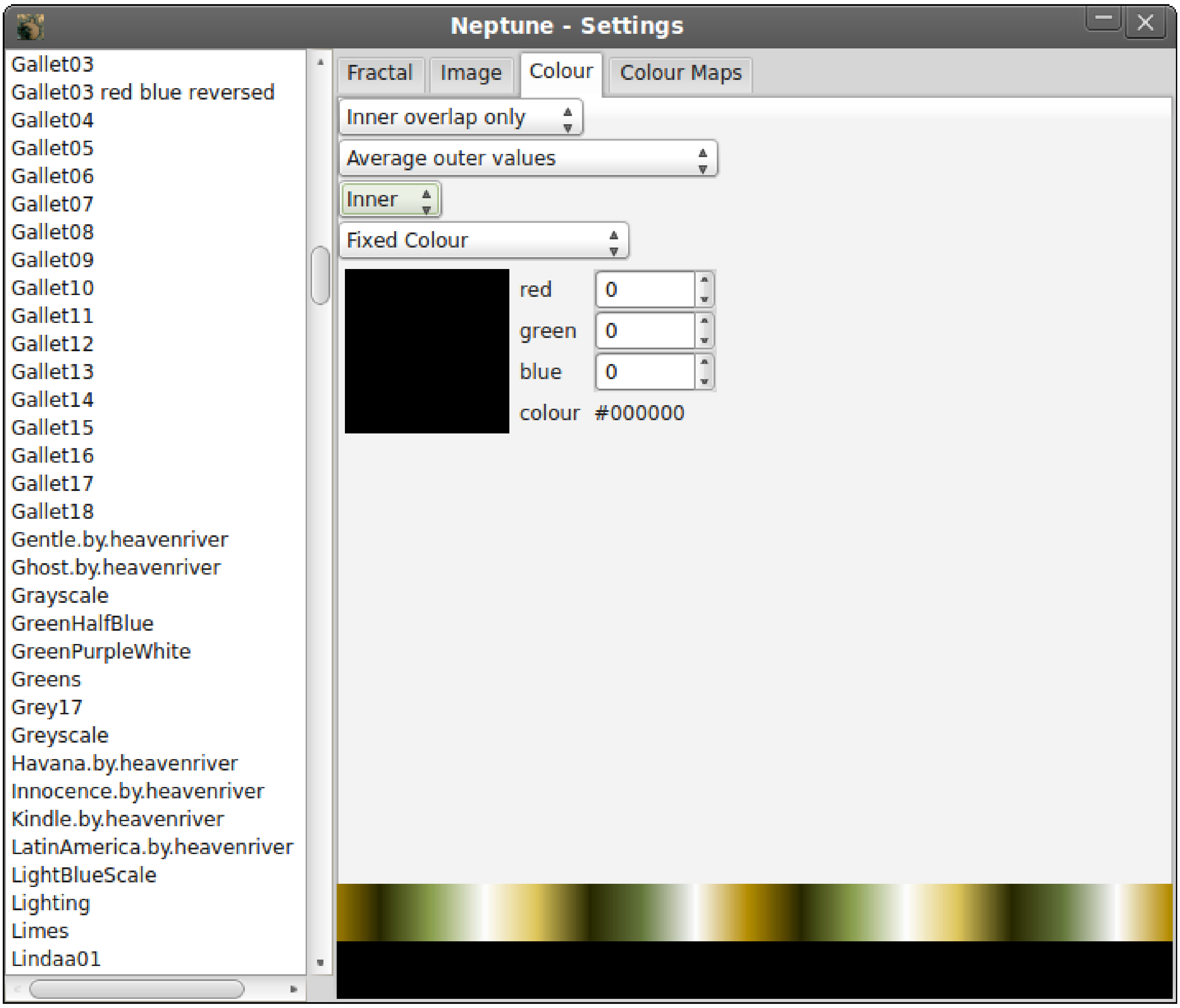
- Average inner/outer
- Outer overlap only
- Inner overlap only

Handling outer/outer:

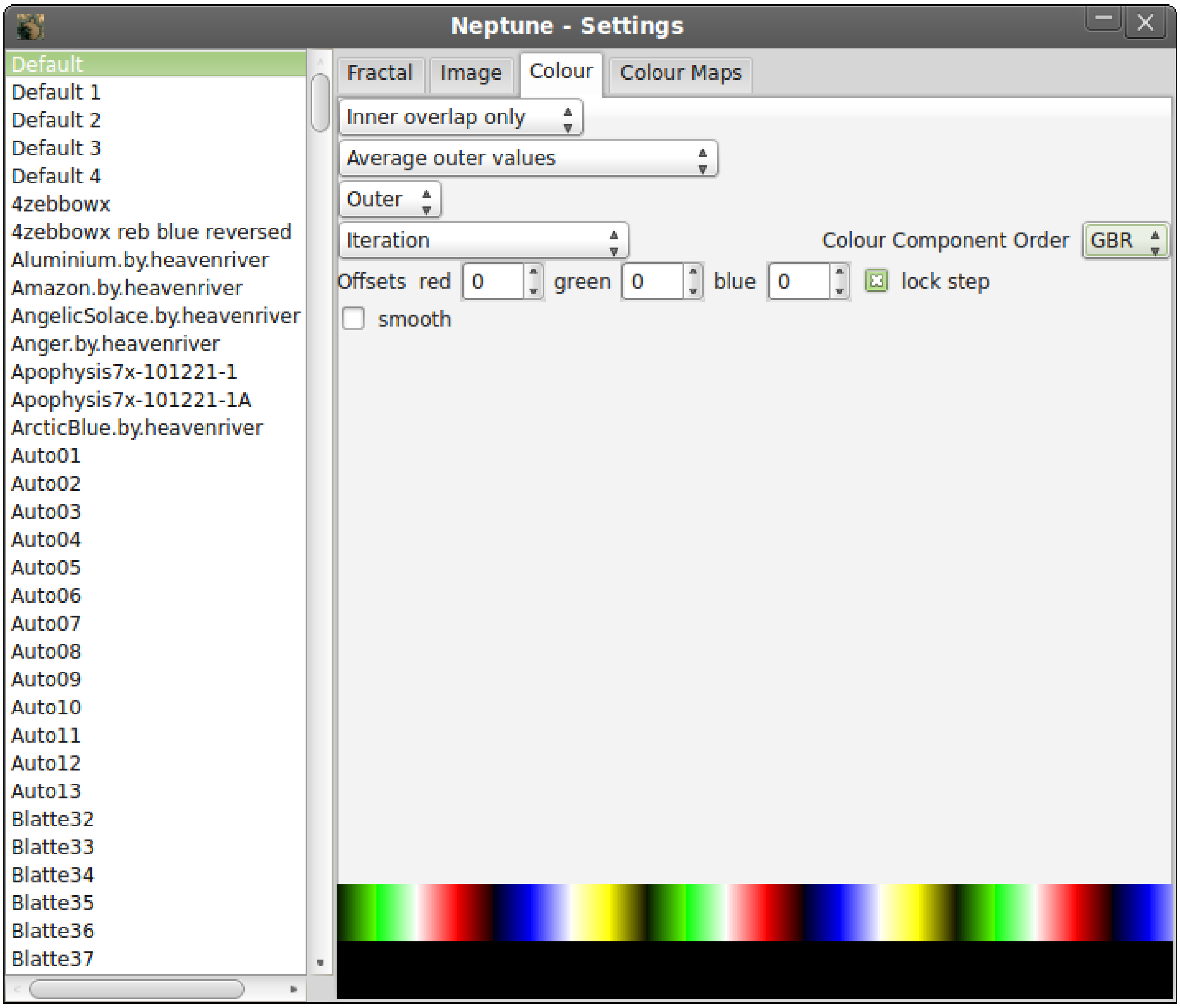
- Average outer values
- Use highest iteration outer values

Fixed Colour

This method colours all the selected points the same colour and is usually used for inner colouring. The default is black.



Iteration



The iteration colour method is used for outer colouring, the iteration used is the iteration where the bailout condition was met. The iteration number is used as an index into the colour map.

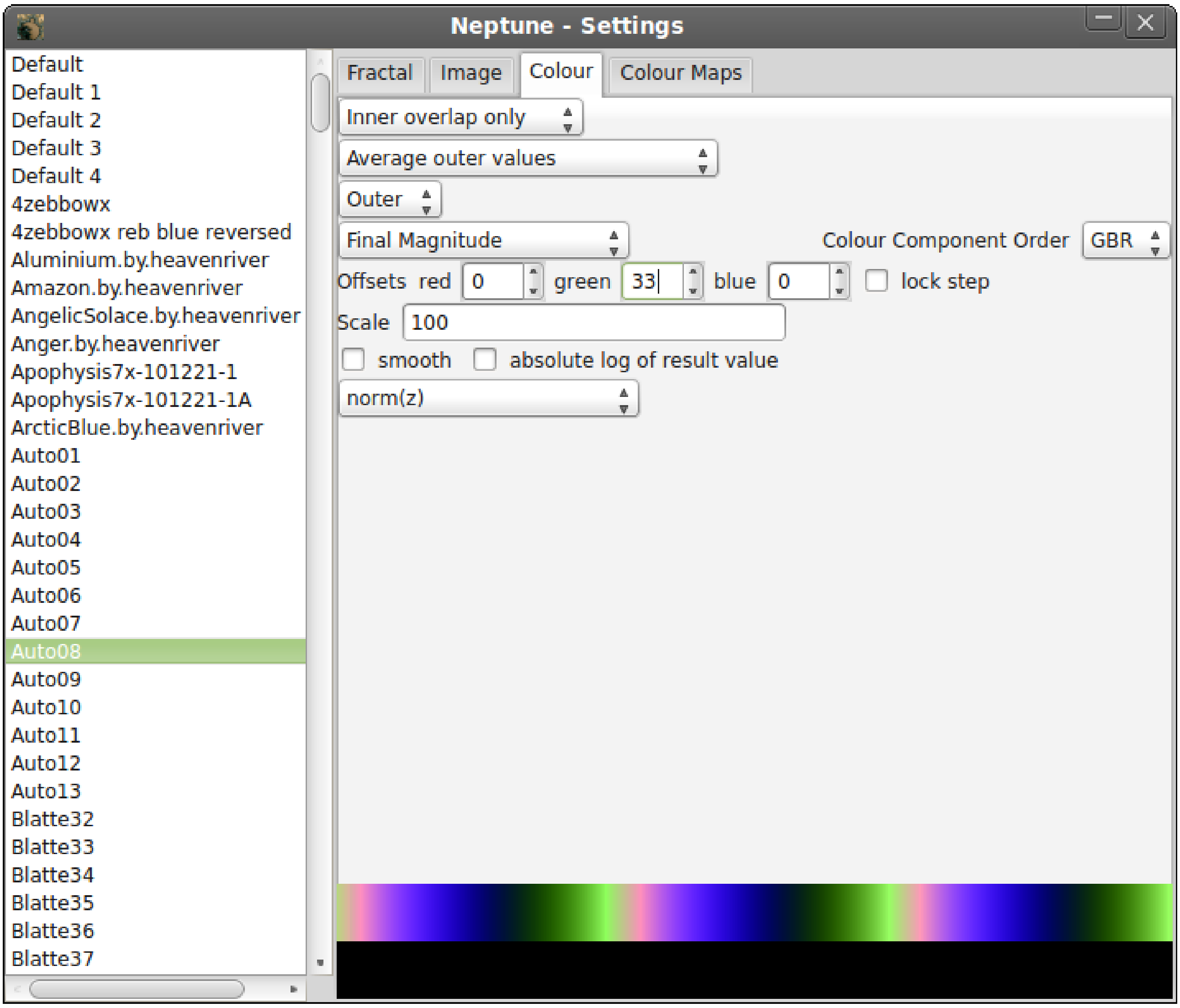
The smooth option can be enabled which modifies the index used by adding:

bailout limit ÷ (abs(z) + 0.000000001)

to the iteration number, so a blended colour is used.

This modified method is also known as “Continuous Potential”.

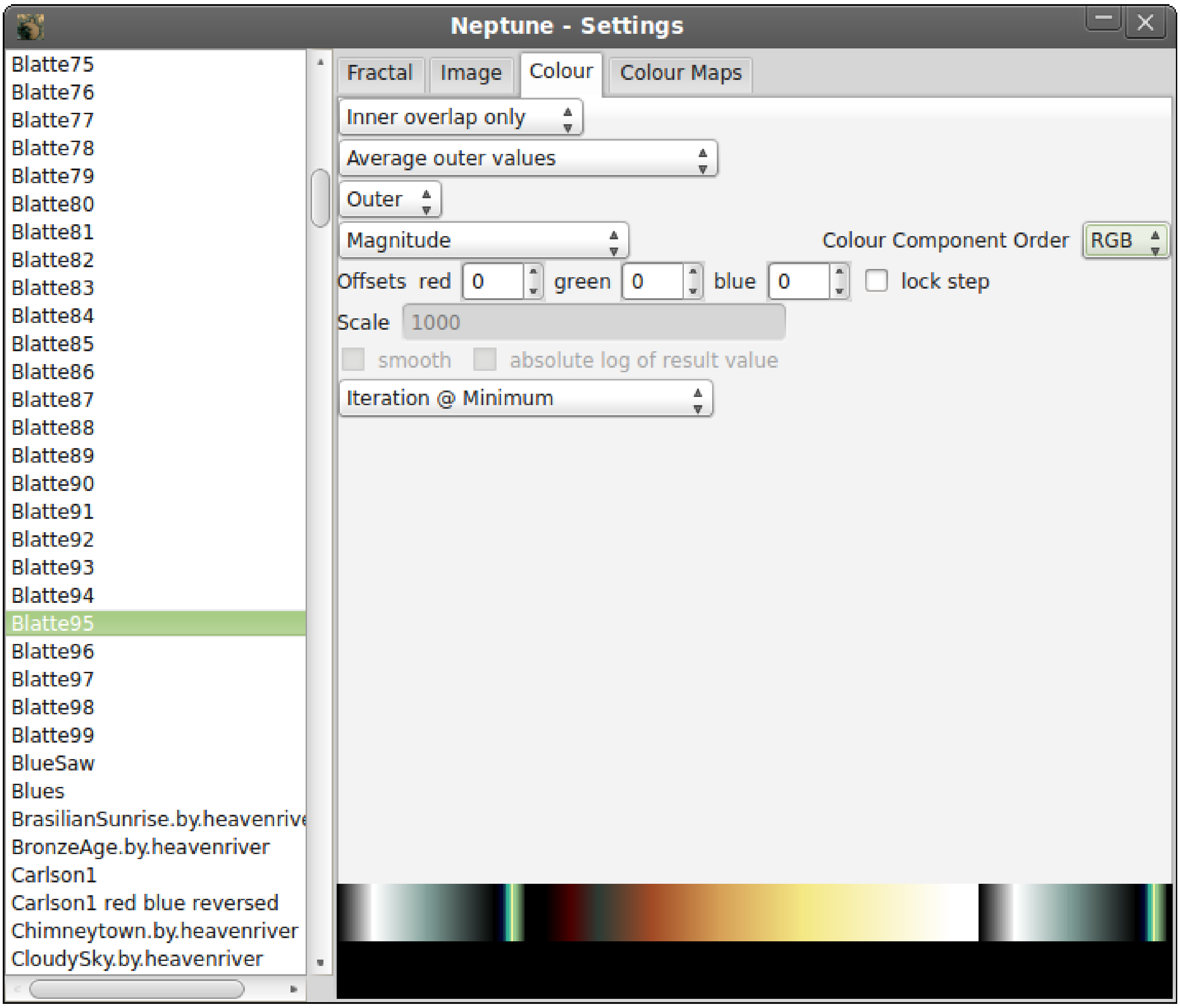
Final Magnitude



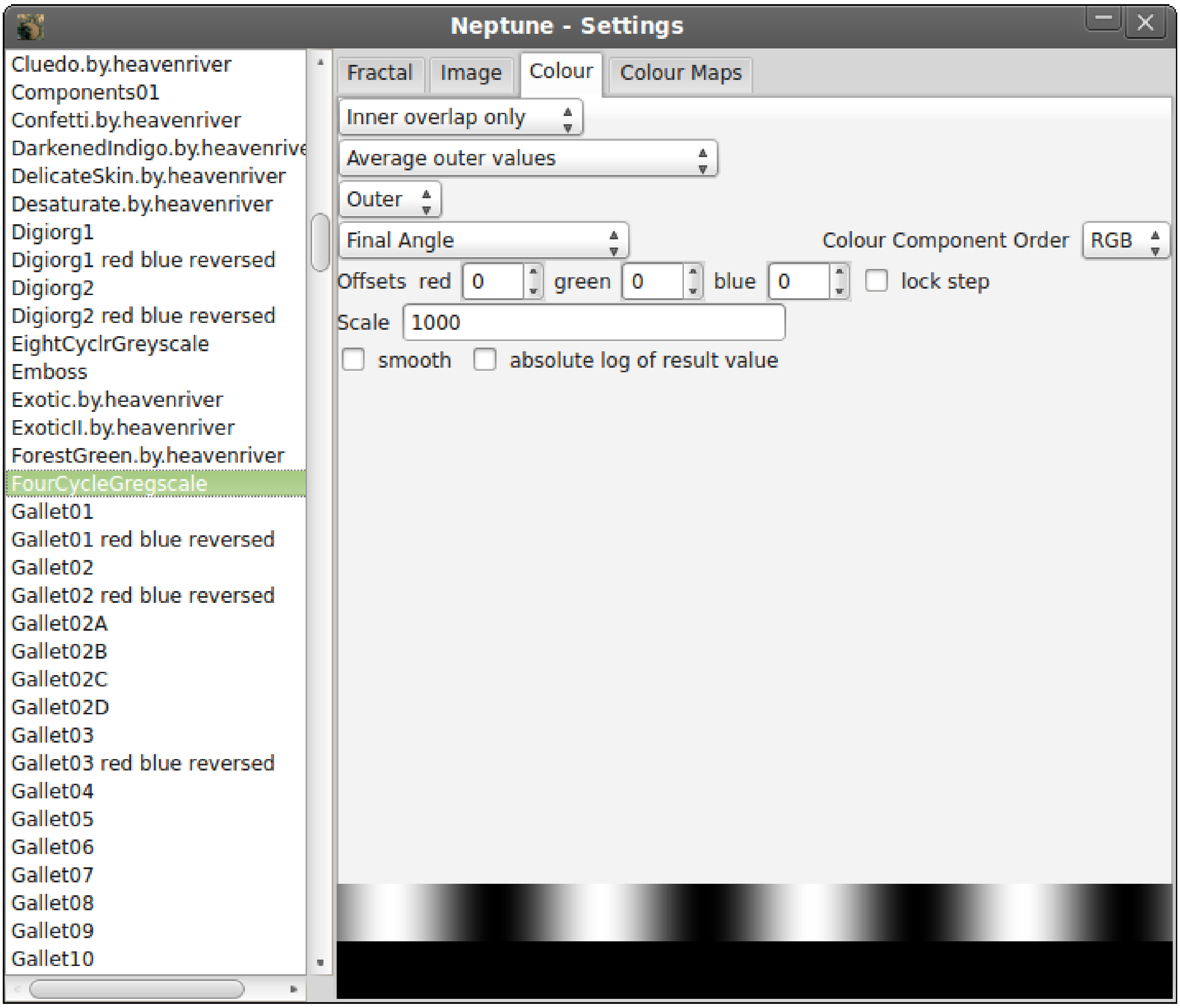
The options combo box provide the following modification of the final value to be used:

1. norm(z)
2. smaller of real(z) and imag(z)
3. larger or real(z) and imag(z)
4. real(z)
5. abs(real(z))
6. imag(z)
7. abs(imag(z))
8. real(z) + imag(z)
9. abs(real(z) + imag(z))
10. real(z)*imag(z)
11. abs(real(z)*imag(z))

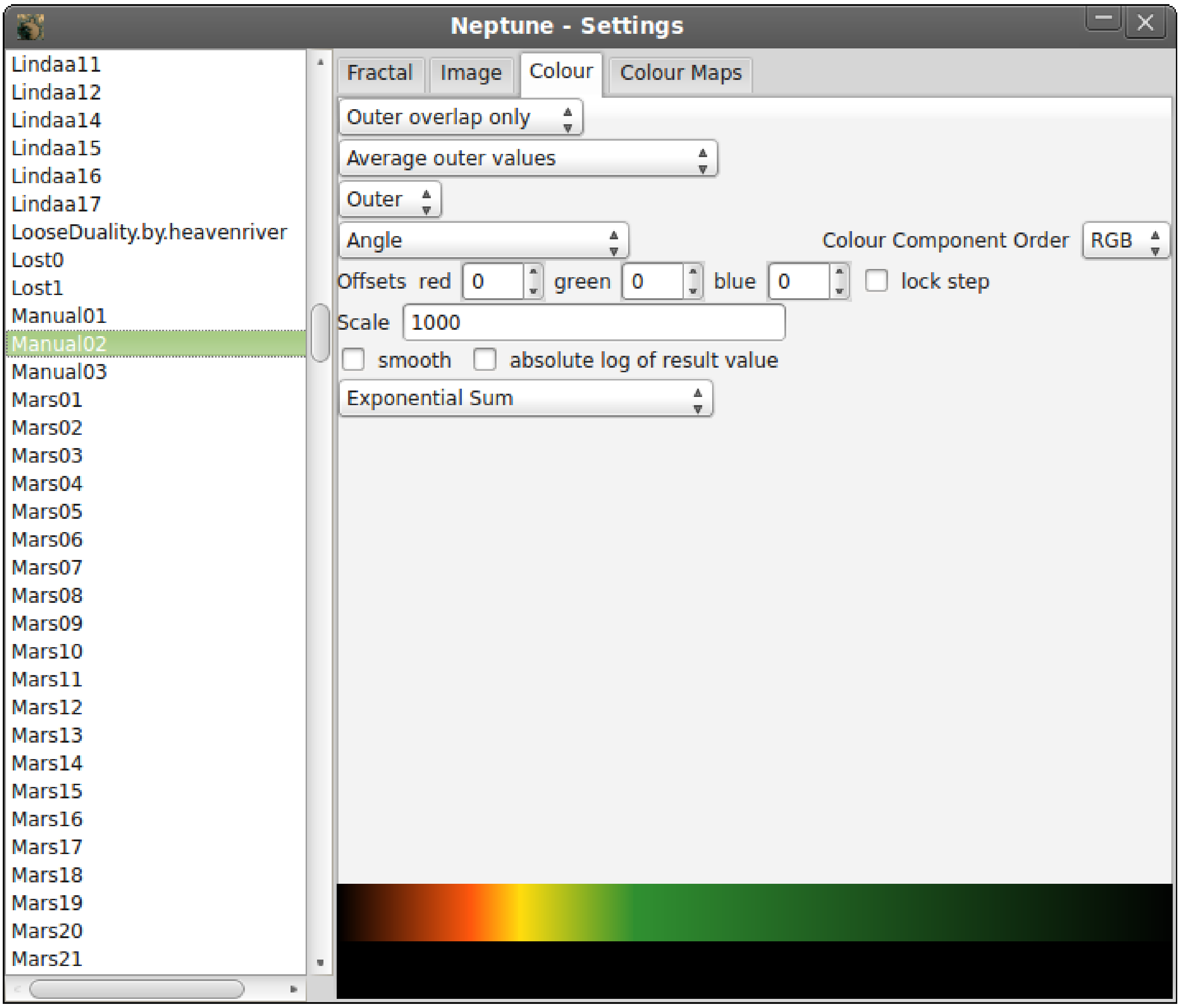
Magnitude and Change in Magnitude



Final Angle

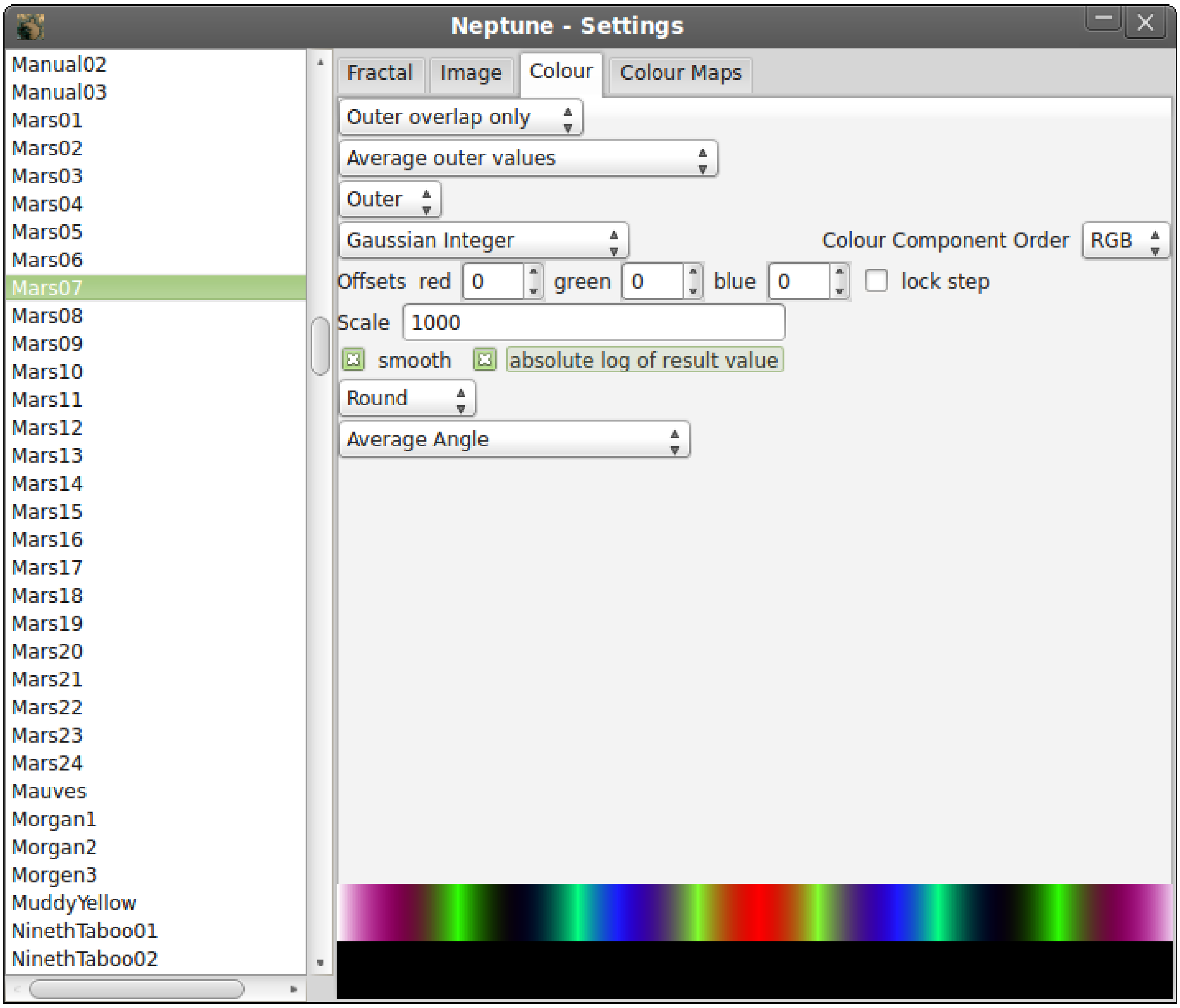

Neptune - Settings
Cluedo.by.heavenriver
Components01
Confetti.by.heavenriver
DarkenedIndigo.by.heavenrive
DelicateSkin.by.heavenriver
Desaturate.by.heavenriver
Digiorg1
Digiorg1 red blue reversed
Digiorg2
Digiorg2 red blue reversed
EightCyclrGreyscale
Emboss
Exotic.by.heavenriver
ExoticII.by.heavenriver
ForestGreen.by.heavenriver
FourCycleGregscale
Gallet01
Gallet01 red blue reversed
Gallet02
Gallet02 red blue reversed
Gallet02A
Gallet02B
Gallet02C
Gallet02D
Gallet03
Gallet03 red blue reversed
Gallet04
Gallet05
Gallet06
Gallet07
Gallet08
Gallet09
Gallet10
Fractal
Image
Colour
Colour Maps
Inner overlap only
Average outer values
Outer
Final Angle
Colour Component Order
RGB
Offsets red
0
green
0
blue
0
lock step
Scale
1000
smooth
absolute log of result value

Angle and Change in Angle



The argument of z after each iteration is found which is the angle of the vector from the origin to the complex number on the complex plane, the value is in the range -π to π radians, if the value is negative then 2π radians is added to the value. Change in value is the absolute difference between the current value and the value for the previous iteration, initial value is zero. The value in radians is used to accumulate statistics.

Gaussian Integer



The Gaussian Integer used for collecting statistics is one of four different types:

- Round
- Ceiling
- Floor
- Truncate

The value used to determine the colour is one of these statistics:

1. Minimum Distance
2. Iteration @ Minimum Distance
3. Angle @ Minimum Distance
4. Maximum Distance
5. Iteration @ Maximum Distance
6. Angle @ Maximum Distance
7. Average Distance
8. Minimum Angle
9. Average Angle

10. Maximum Angle
11. Maximum Distance/Minimum Distance
12. Range
13. Variance
14. Standard Deviation
15. Coefficient of Variation

Trap Distance and Change in Trap Distance

Most orbit traps have common parameters, some have an option for a extra point at the centre of the trap and the Steiner Chain trap has options controlling how the trap is constructed.

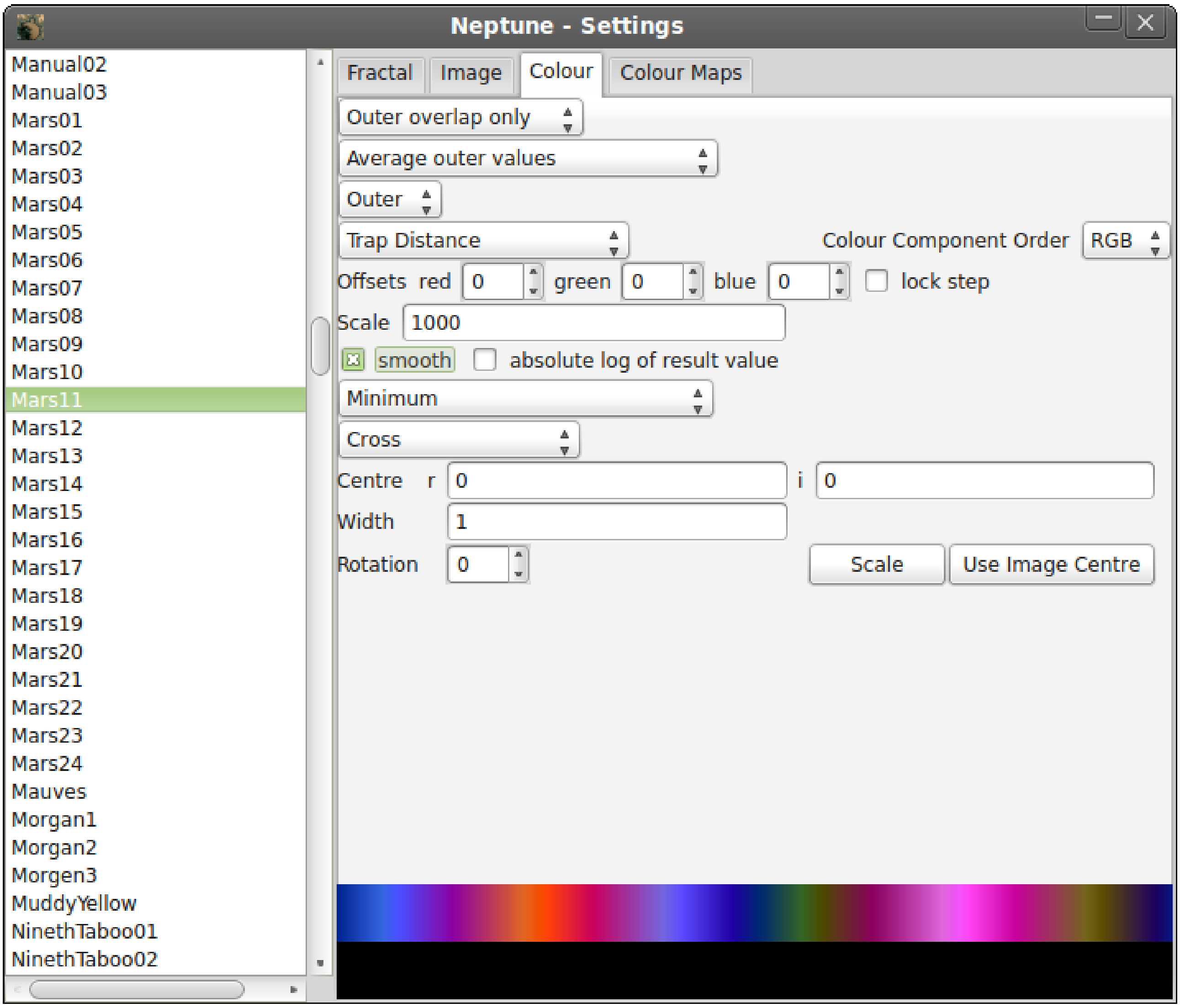
Statistics of the distance or change in distance to the orbit trap are accumulated and the statistic is used to determine the colour.

There are a number Orbit Trap shapes, the appearance of these orbit traps are shown in the Orbit Trap Appendix.

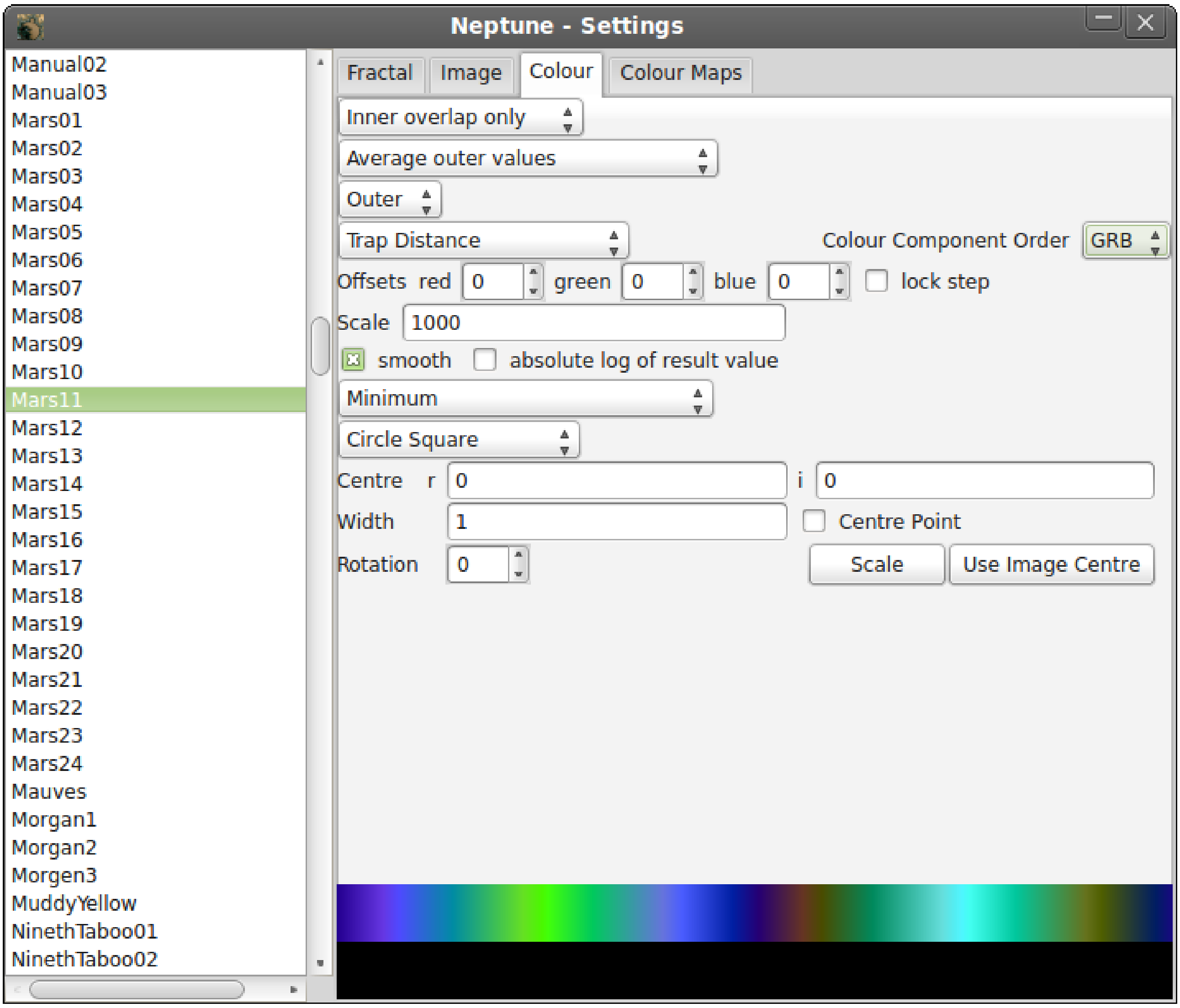
The size and position of the orbit trap are specified using the position and size entry boxes, the size is the width of the orbit trap e.g. for triangles it is the length of the base and for circles it is the diameter. The position of the trap can be set to the current image centre using the button below the position entry boxes and the size can be scaled to the image by pressing the button below the size entry box. The size set for scale to image is half the current width of the image. The size is disabled for the Point orbit trap as its size is always zero.

The orientation of the trap can be altered by rotating it about its centre, the spin button is disabled for all traps that are unaltered by rotation such as point and circle.

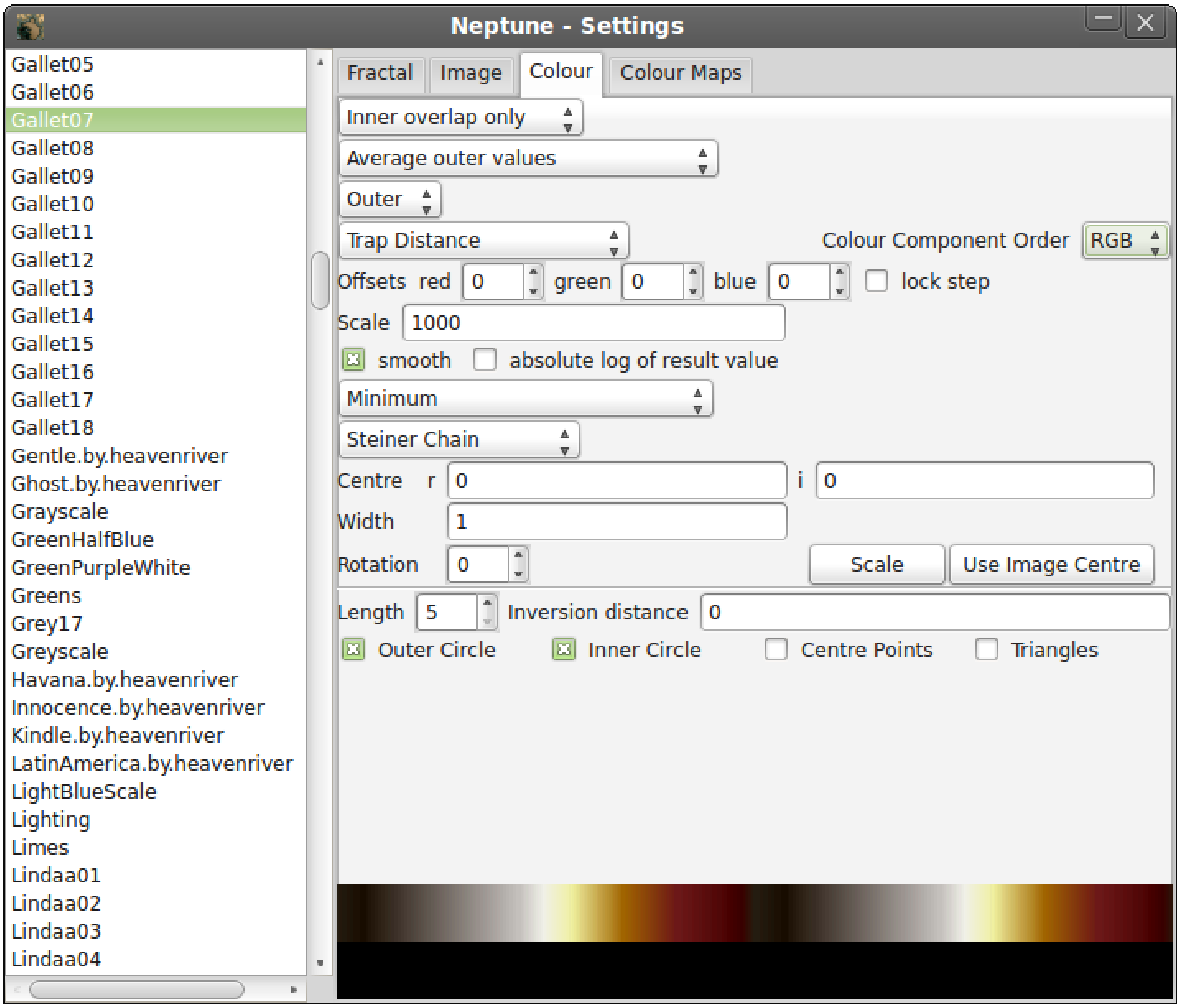
Note: the more complicated the orbit trap the greater the number of calculations required to find the distance to the orbit trap. There will be a noticeable delay before the first image and current number of iterations are displayed.



An orbit trap without an optional central point.



An orbit trap with an optional central point.

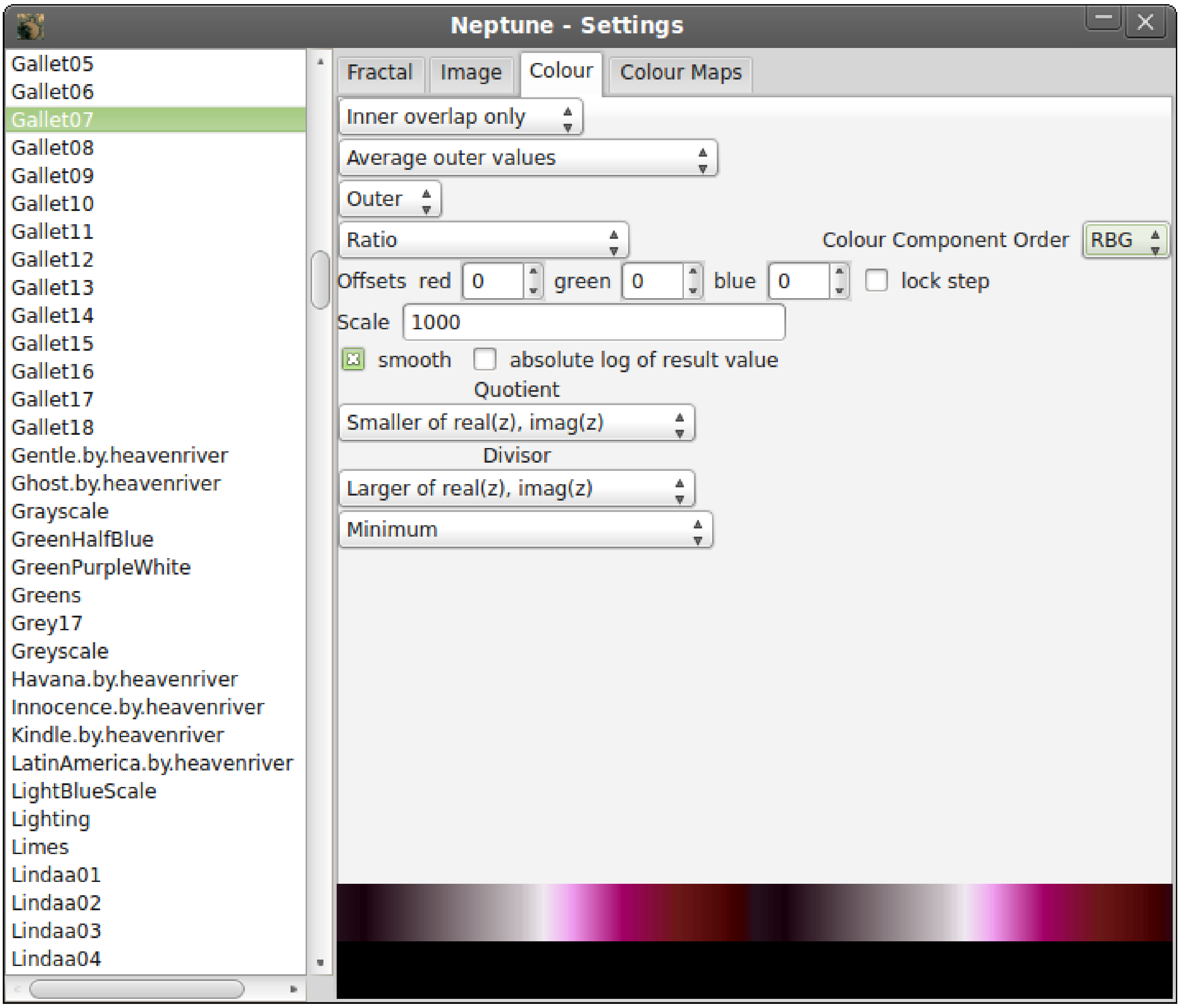


The extra options for the Steiner Chain Orbit Trap control how the chain is constructed:

1. Length, the number of circles enclosed within the inner and outer circles. The values are restricted to the range 5 to 64.
2. Inversion distance, a value of zero produces a ring of circls of equal sizes, non-zero values distort the chain producing a chain of circles of varying sizes. The values are restricted to the range -0.9 to 0.9.
3. Outer Circle, display of the outer enclosing circle is optional.
4. Inner Circle, display of the inner enclosing circle is optional.
5. Centre Points, an extra option not usually associated with Steiner Chains, if this option is set a centre point will be displayed for the chain circles.
6. Triangles, an extra an extra option not usually associated with Steiner Chains, if this option is set an inscribed triangle will be used in the chain circles. One of the corners of the inscribed triangles touches the inner enclosing circle.

Examples of Steiner Chain orbit traps can be found in the orbit trap appendix which show the shapes of the various orbit traps.

Ratio and Change in Ratio



These colour methods select colour based on statistics gathered of the ratio of values, for change in ratio the values used are the differences between the current ratio and the previous ratio.

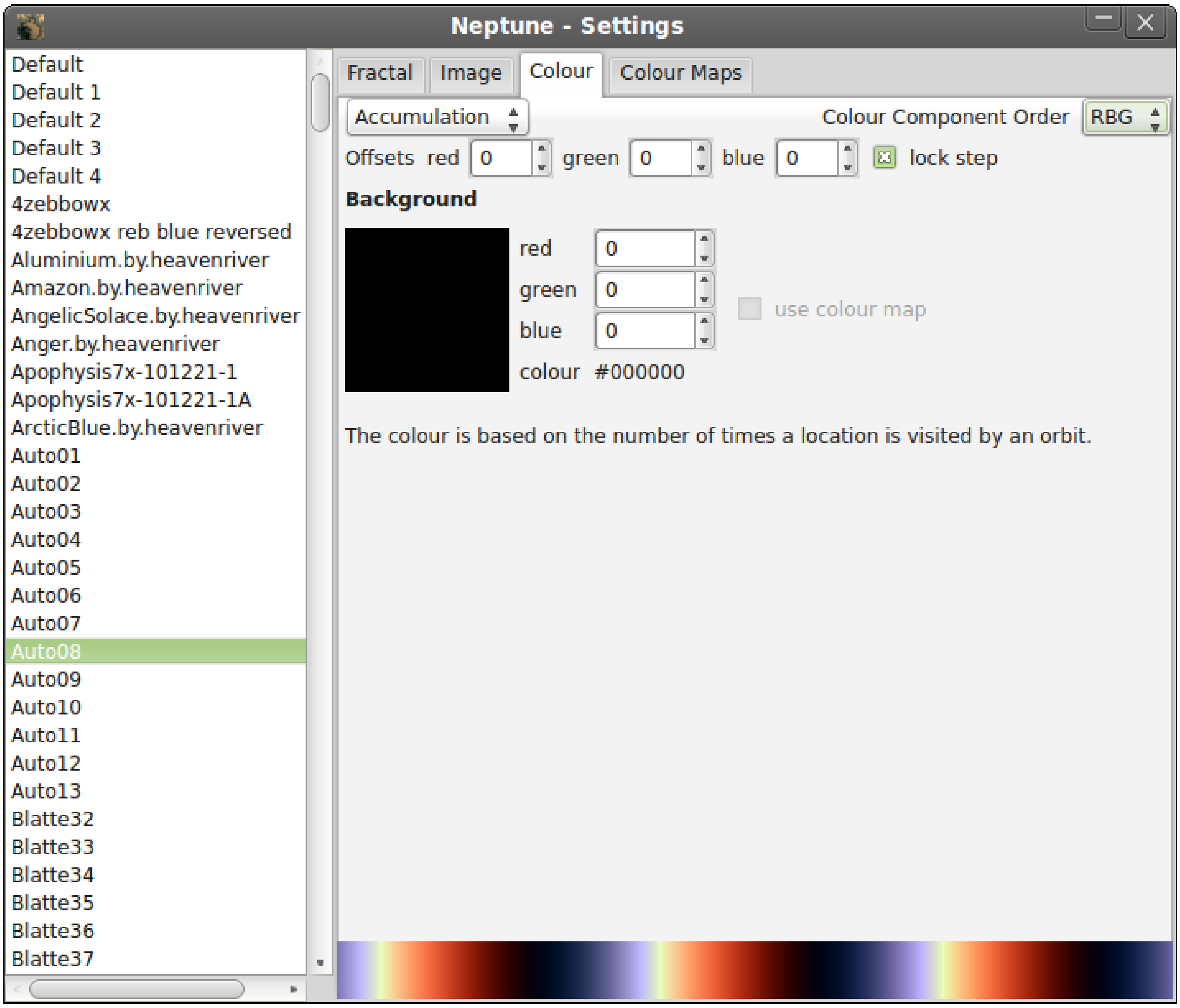
The quotient and divisor of the ratio can be selected from these values:

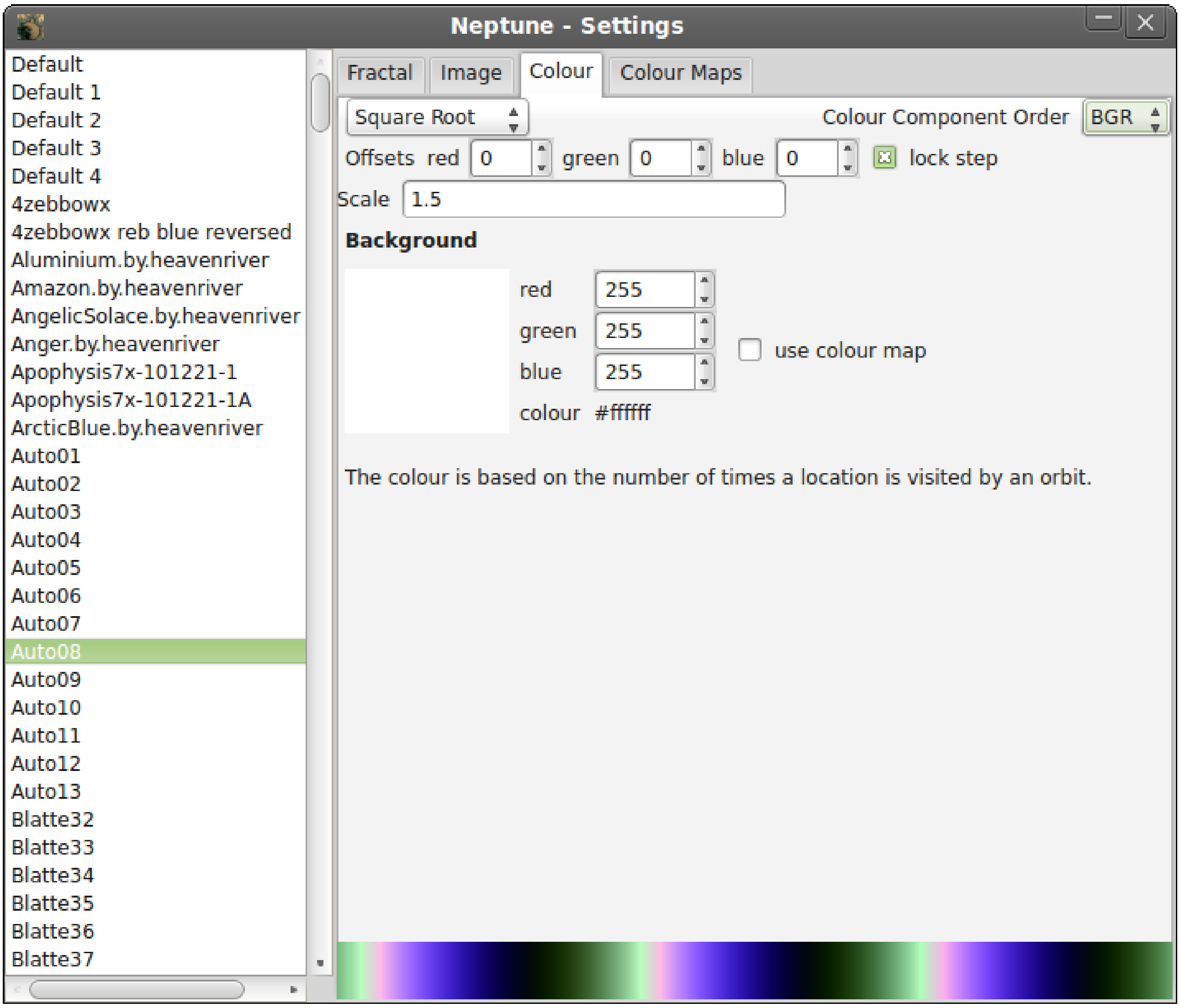
1. Smaller of real(z), imag(z)
2. Larger of real(z), imag(z)
3. abs(real(z))
4. abs(imag(z))
5. abs(real(z) + imag(z))
6. abs(real(z) – imag(z))
7. abs(real(z)) + abs(imag(z))
8. abs(abs(real(z)) – abs(imag(z)))
9. abs(real(z))*abs(imag(z))
10. abs(z)
11. norm(z)

Use of the same values for quotient and divisor will result in a uniform colour.

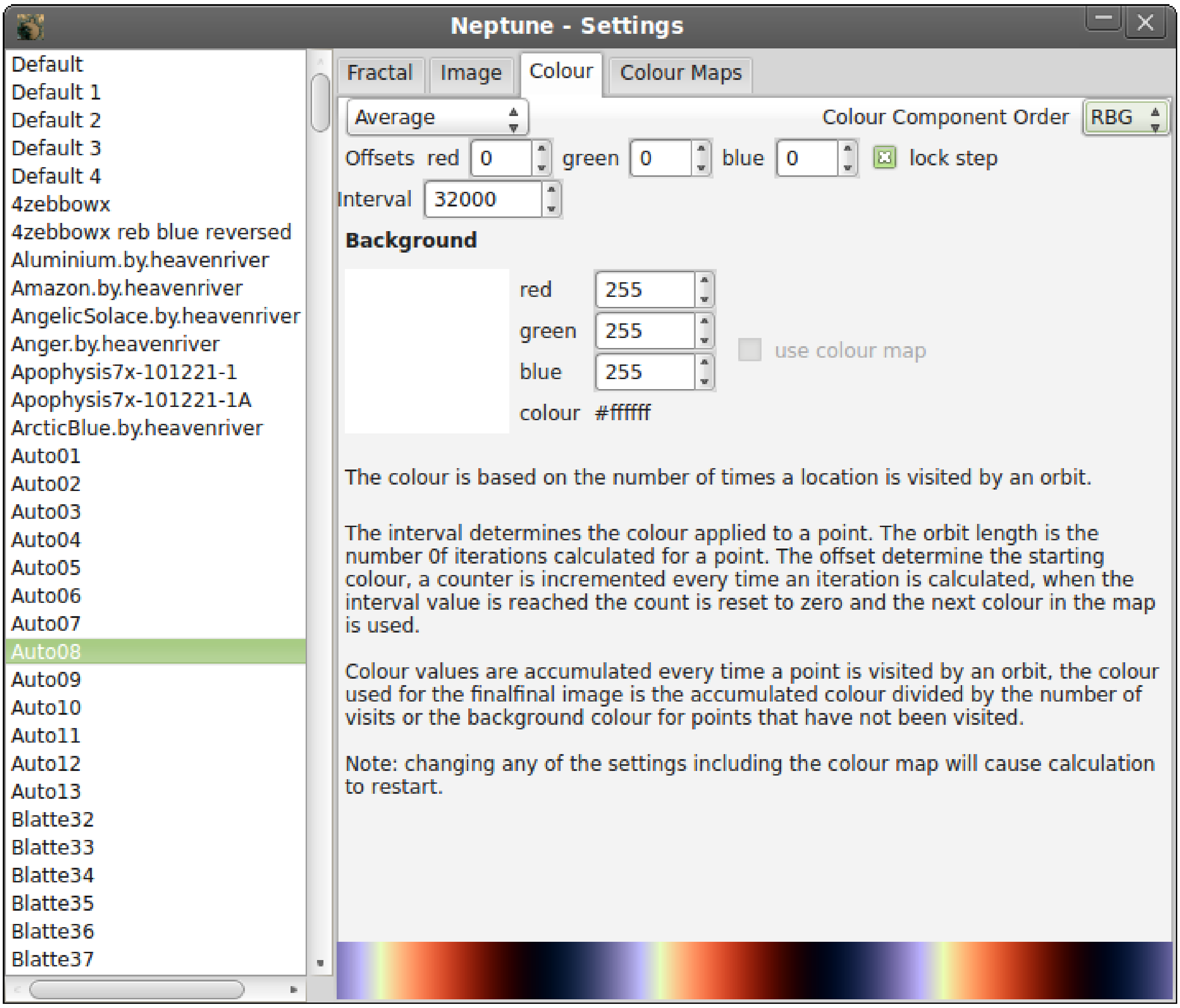
3.2.3.3 Colour - Orbit Plotted Fractals

The colour methods are based on the number of times a location is visited by an orbit (hits). Accumulation simply uses the hits value, square root and logarithm either take the square root or logarithm of the hits. A scale factor can be specified for both Square Root and Logarithm methods.





The layout the logarithm option is the same as for the square root option.



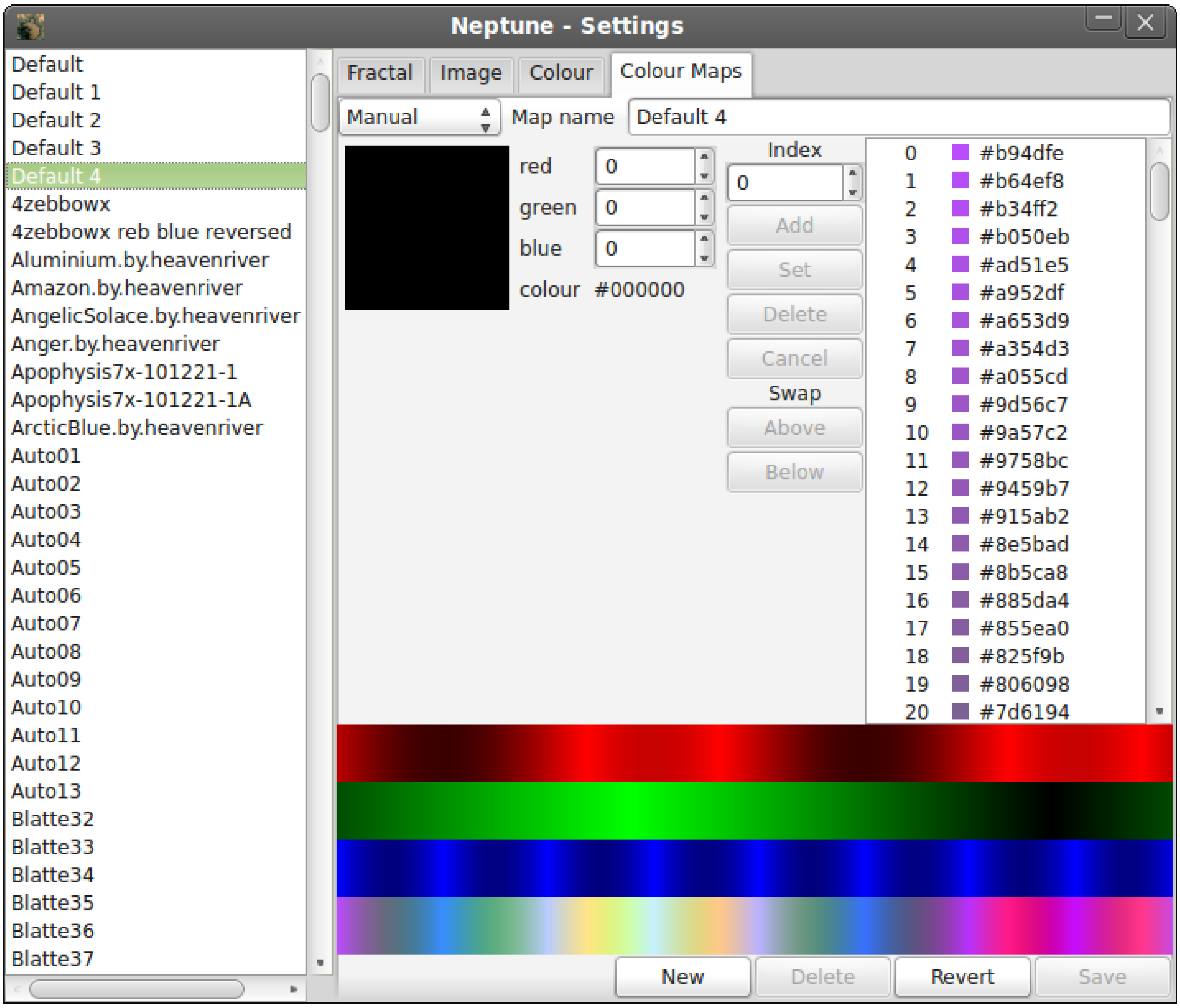
Note: this colour method is inherited from Saturn where it only appeared to work well for fractals that were originally called Pickover Popcorn and are now produced using Pickover Popcorn 4F & 6F fractals calculated using the Julia algorithm. It might produce interesting results.

3.2.4 Colour Maps Tab

This tab handles the management of colour maps. Maps can be added, edited, renamed and deleted. The editing of a map is dependent of whether it is a manually edited map or an auto generated map and the appearance of the tab varies accordingly.

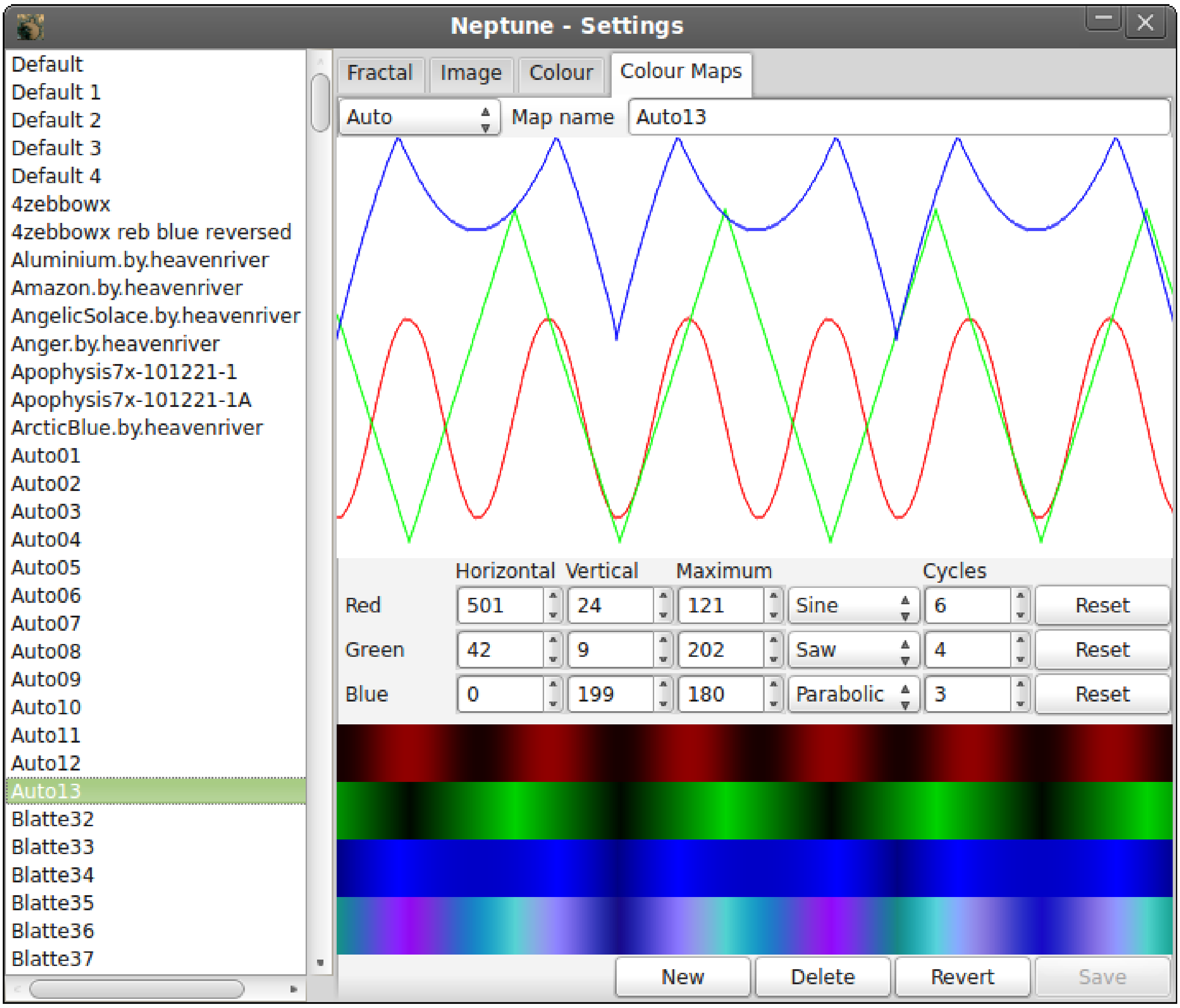
There are four colour map rectangles one for each component and one for the components combined.

Manual

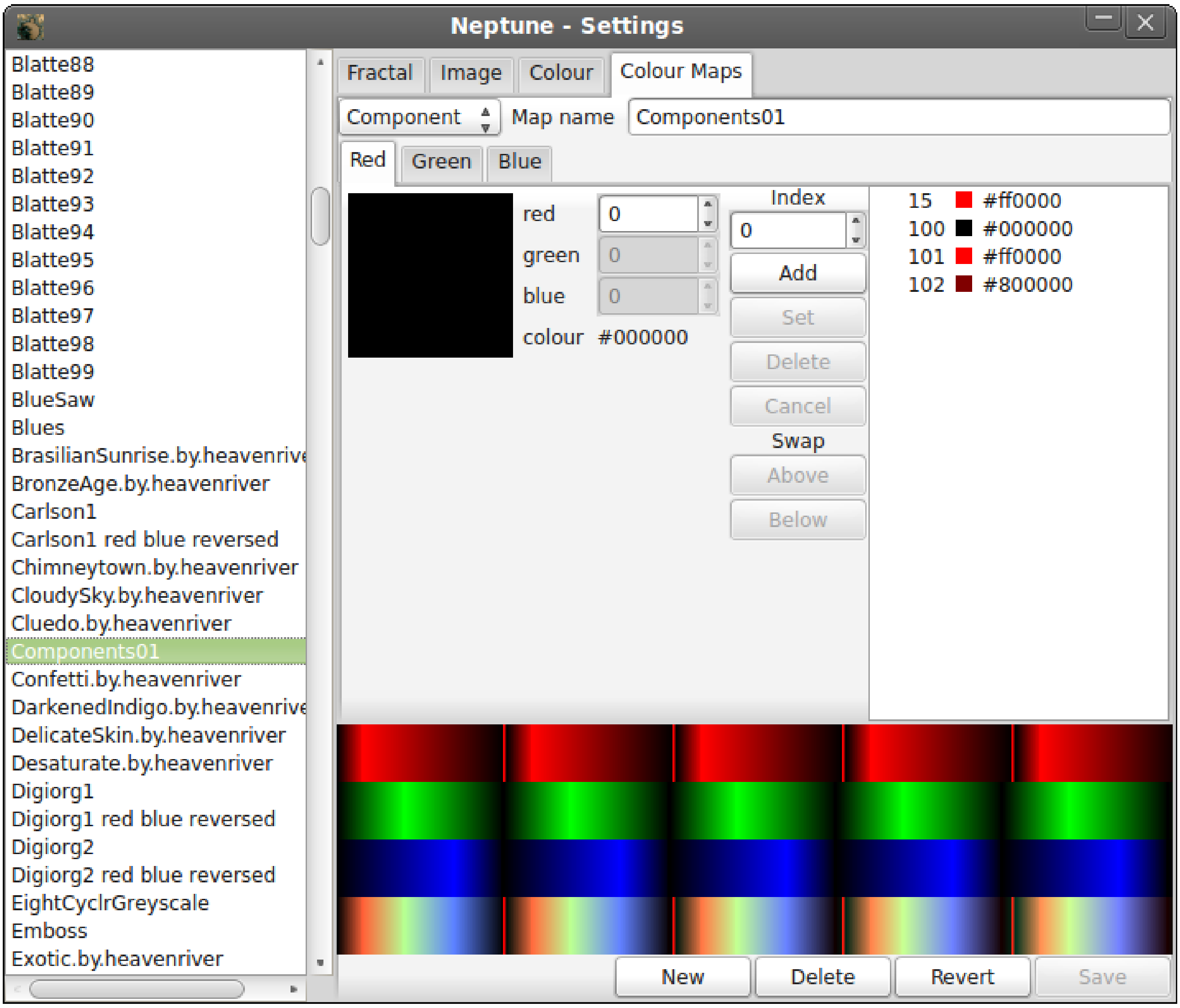


Auto

Auto colour maps are generated based on waveforms, the amplitude, horizontal and vertical positions can be adjusted. The type and number of cycles of the waveform can also be changed. If the part of the waveform is greater than 255 or less than 0 then that part of the waveform will be reflected.

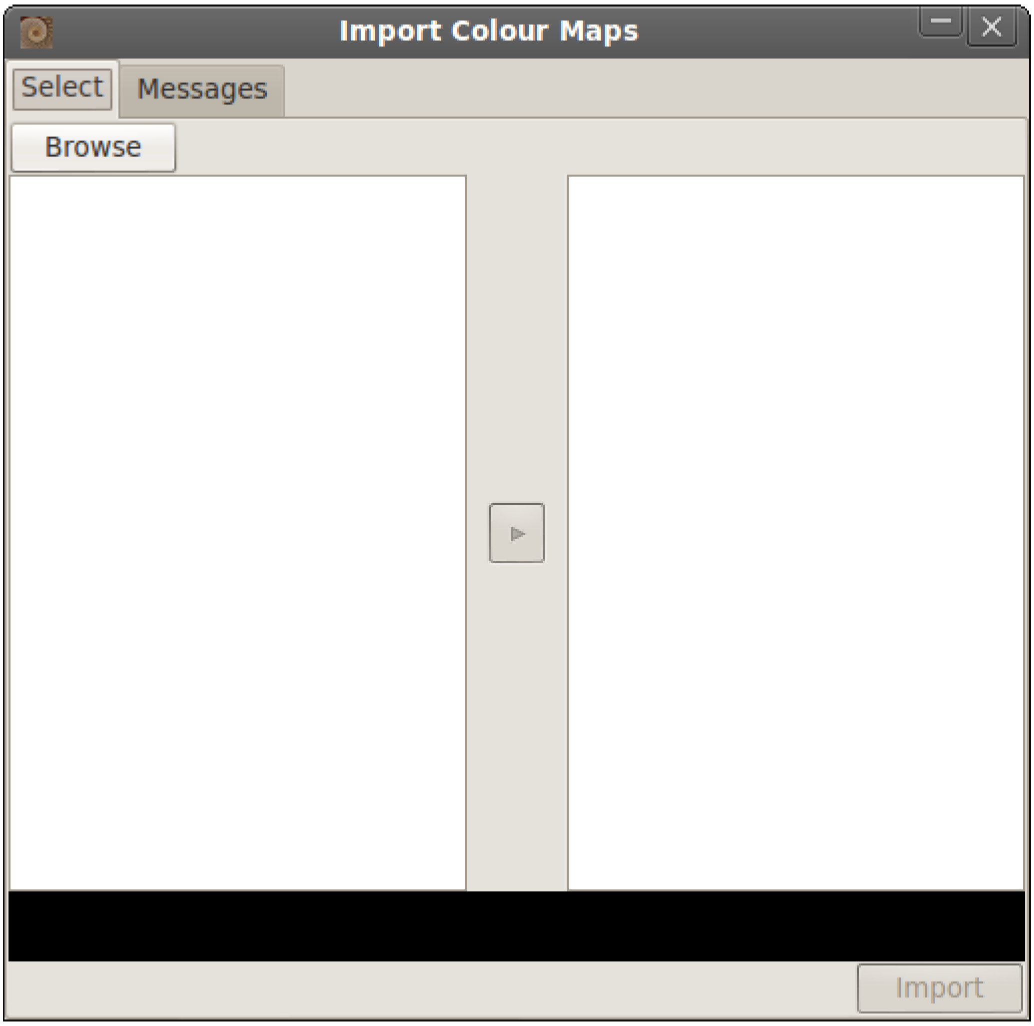


Component


Neptune - Settings
Blatte88
Blatte89
Blatte90
Blatte91
Blatte92
Blatte93
Blatte94
Blatte95
Blatte96
Blatte97
Blatte98
Blatte99
BlueSaw
Blues
BrasilianSunrise.by.heavenrive
BronzeAge.by.heavenriver
Carlson1
Carlson1 red blue reversed
Chimneytown.by.heavenriver
CloudySky.by.heavenriver
Cluedo.by.heavenriver
Components01
Confetti.by.heavenriver
DarkenedIndigo.by.heavenrive
DelicateSkin.by.heavenriver
Desaturate.by.heavenriver
Digiorg1
Digiorg1 red blue reversed
Digiorg2
Digiorg2 red blue reversed
EightCyclrGreyscale
Emboss
Exotic.by.heavenriver
Fractal
Image
Colour
Colour Maps
Component
Map name
Components01
Red
Green
Blue
red
0
green
0
blue
0
colour #000000
Index
0
Add
Set
Delete
Cancel
Swap
Above
Below
15 #ff0000
100 #000000
101 #ff0000
102 #800000
New
Delete
Revert
Save

3.3 Import Colour Maps Window

The initial import colour maps window is shown below:



Colour map files can be added to the tree on the left hand side (the source tree) using the browse button, three types of colour map can be imported, .map, .scm and .ugr. Several colour maps can be defined in .ugr and .scm files, only one colour map is defined in a map file.

All the colour maps in a file can be copied to the list on the right (importing list) using the arrow button or individual colour maps in a file can be copied. If the source tree cursor is on a file name it is disabled if all its colour maps have been copied to the importing list, if the cursor is on a colour map name the arrow button is disabled if the colour map is also in the importing list.

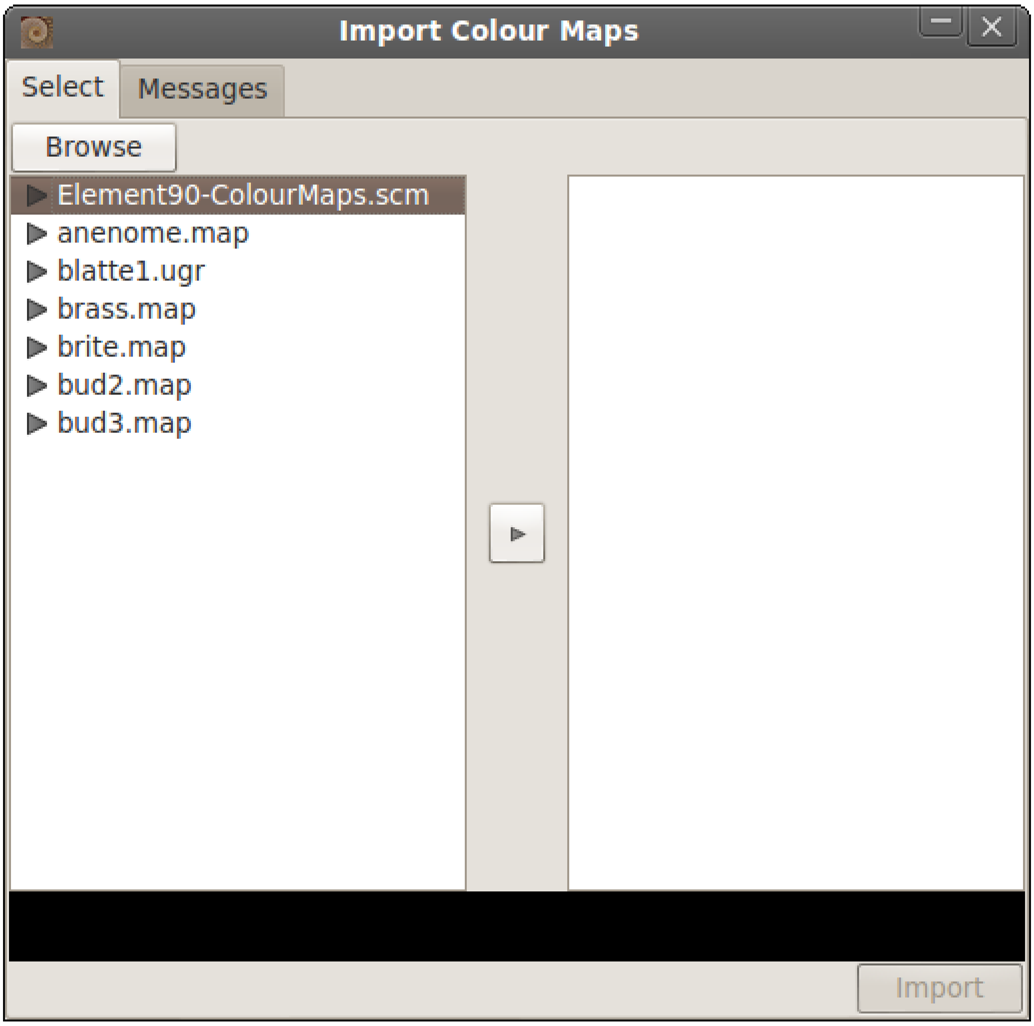
If the source tree cursor is on a file name there is no cursor displayed on the importing list, if the cursor is on a colour map name and that colour map is in the importing list the importing list cursor is displayed on the corresponding colour map name.

The names in the importing list can be different from the colour map names in the source tree, this is because the names in the import file may already exist in the Neptune's collection of colour maps, when the colour map is copied to the importing list the name will be altered if it already exists by the addition "~*n*" where *n* is a number, for example if a colour map is called "Reds" and "Reds" already exists the name become "Reds~0" if that name also already exists then "Reds~1" will be tried and so on until an unused name is found. The altered names are intended to be temporary and can changed once the maps have been imported..

The black rectangle above the import button will display the current colour map only when the source tree cursor is on a colour map name (the cursor may be hidden), the rectangle will be black when the cursor is on a file name.

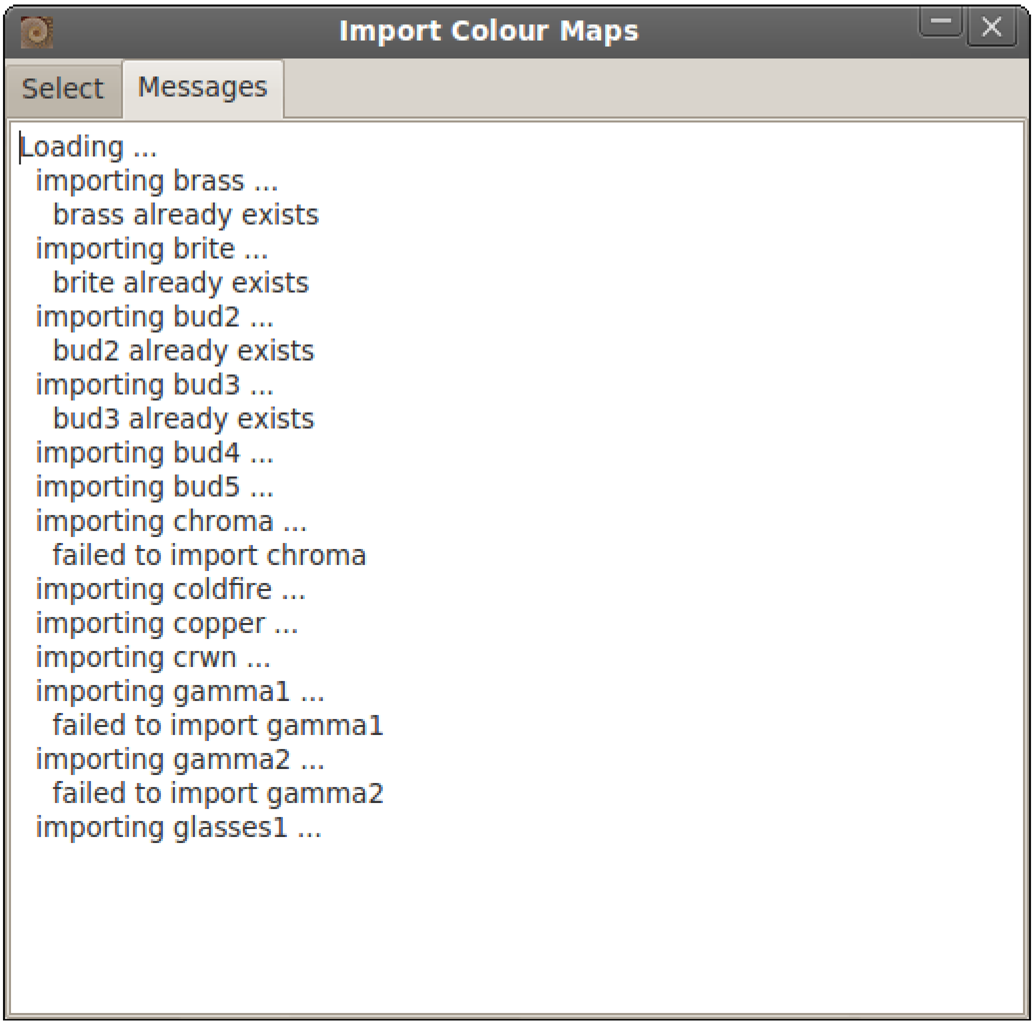
The import button is enabled when there are colour maps in the importing list, when the button is pressed the colour maps are added to Neptune's collection, the importing list is cleared and the imported colour maps are also removed from the source tree, if all the maps in a file are imported the file name is also removed. On completion of the import the import button is disabled and no source cursor is displayed.

The window below shows two files that have been added to the source tree.



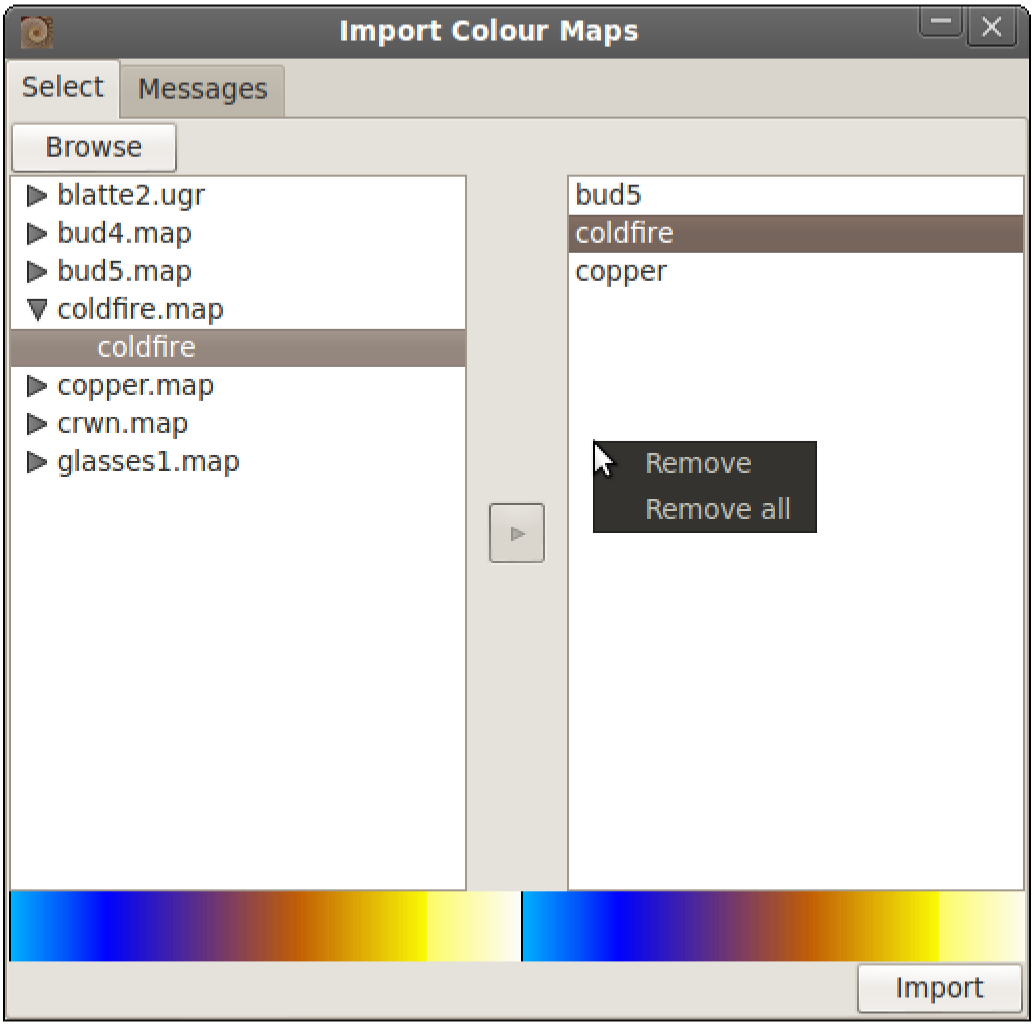
When files are added to the source tree a check is made for each map to determine whether it is already in Neptune's collection of colour maps if the map is already in the collection it is disregarded, the file name is only added to the list if it contains new colour maps. The message tab is used to report duplicate colour maps and any errors that occurred while adding an import file to the source tree. If Neptune fails to read a colour map file it is not added to the source tree.

An example of the messages tab:



The import of .map files can often fail because they contain text on lines after the triplet of numbers representing a colour, removing the text will allow the .map file to be imported.

The window below shows the colour maps defined in a file, colour maps in the importing list and a pop up menu.



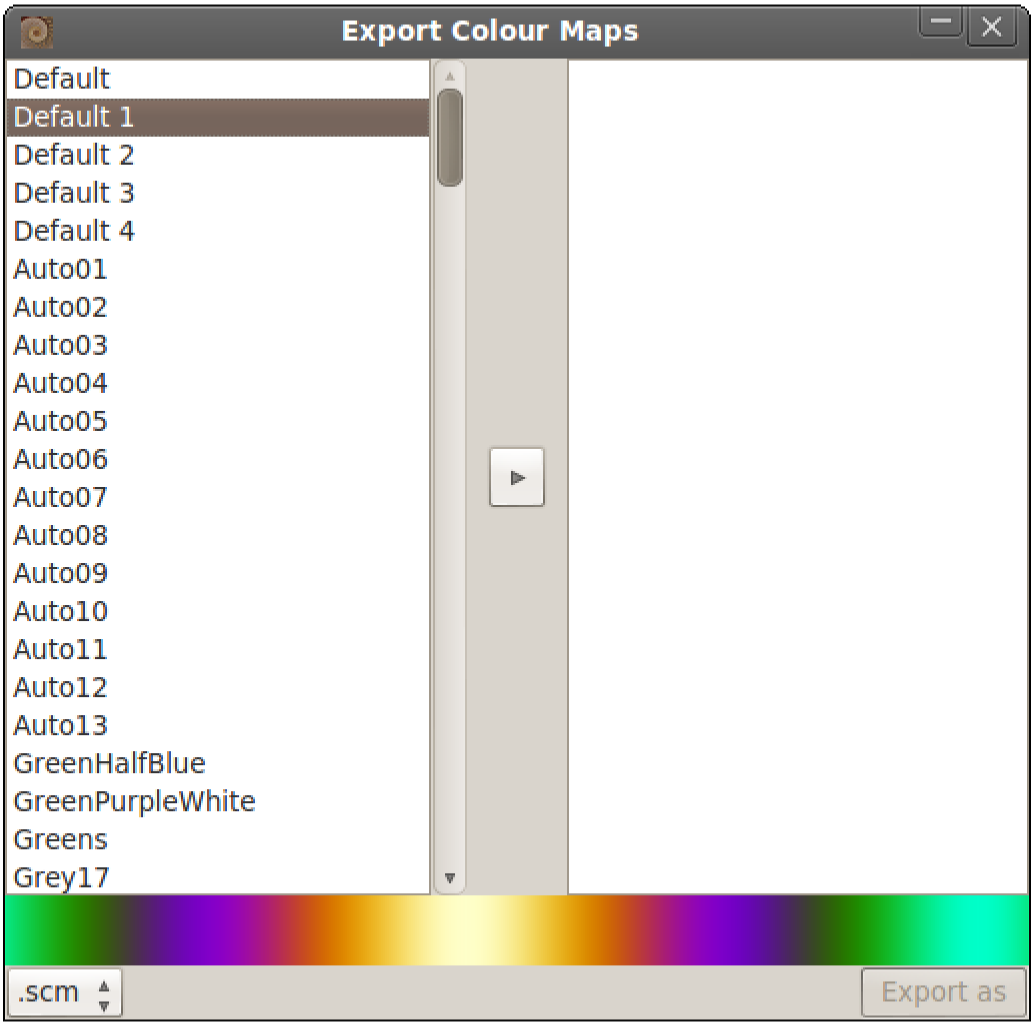
The pop up menu shown allows all the colour maps in the importing list or an individual colour map to be removed.

The pop up menu can also be used to delete all the source tree entries, individual colour maps or a file of colour maps.

The pop up menus are displayed when the right mouse button is clicked on the source tree or the importing list.

3.4 Export Colour Maps Window

The export colour maps window when it is first started:



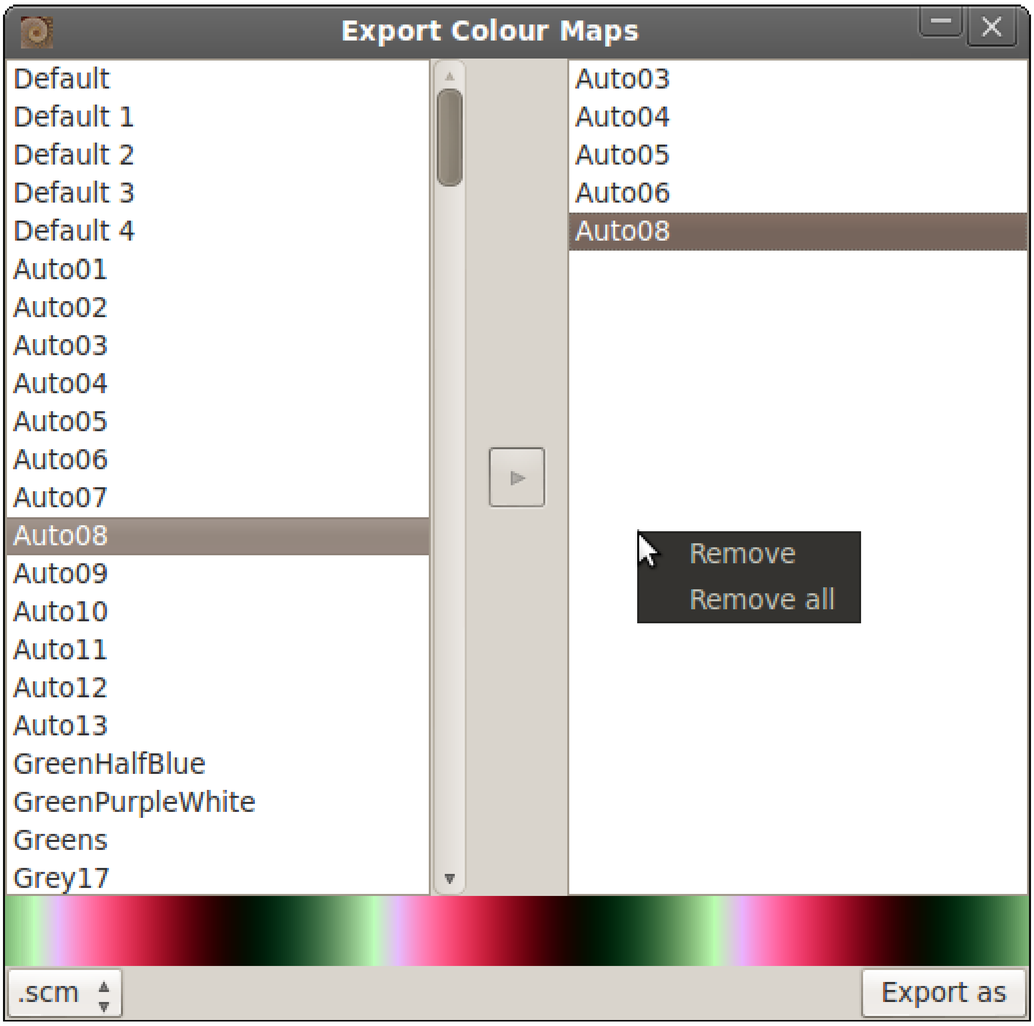
A list of all the colour maps defined in Saturn is on the left hand side, on the right hand side is a list of the colours to be exported, initially this empty and as there are no colour maps to export and the export as button is disabled.

Colour maps are copied into the right hand side (the exporting list) using the arrow button which copies the colour map at the current cursor, the cursor then moves down one position. The cursor on the left hand list (the source list) can be moved up and down, if the colour map at the cursor is also in the exporting list then the arrow button is disabled and a cursor is also displayed on the exporting list. The colours in the colour map are displayed at the bottom of the window.

The cursor can also be moved on the exporting list, the arrow button will be disabled and a cursor will be displayed on the corresponding colour map in the source list.

The “export as” button is enabled as soon as there is a colour map in the exporting list, when the “export as” button is pressed you will be asked for a file name, if the file extension is omitted the appropriate extension will be added. The are two two type of file that can be exported, .scm or .ugr. When the colour maps have been exported the exporting list is not cleared as a second export of the same colour maps in a different format may be required.

The exporting list can have individual colour map or all colour maps removed using a pop up menu. To remove an individual colour map select it using the cursor and then right click on the colour map, a pop up menu will appear with two items: Remove and Remove All.



4 Fractal Types

4.1 Escape Time

All of the fractals implemented in Neptune and Triton are so called escape time fractals which are iterated a maximum number of times or until a bailout condition is met when processing for a given point stops.

4.1.1 Mandelbrot Algorithm

The Mandelbrot algorithm is used when at least one of the complex parameters is set to use the location in the complex plane.

The algorithm builds a picture by iterating the formula until the maximum number of iterations has been reached or the escape (or bailout) condition has been met. The value of the complex parameters set to use the complex plane is varied and is different for all locations calculated. Usually the initial value of z is a fixed value commonly zero except where the use of zero produces infinity or an undefined value (e.g. log of zero), Saturn also provides for the use of the location in the complex plane, any complex plane transforms can be ignored.

4.2 Orbit Plotting

Orbit fractals consist of all the points calculated using the orbit fractal's formula. The initial value is a point on a plane when the formula is evaluated a new point on the plane is plotted which is then used to determine the next point to be plotted. The number of points plotted is specified by the orbit length, the set of plotted points being the orbit. A grid is overlaid on the plane with a number of columns and rows, the position on the plane of the starting value corresponds with the row and column position on the plane, the value calculated is a position on the plane and plotted position is found by converting to the nearest row and column on the grid.

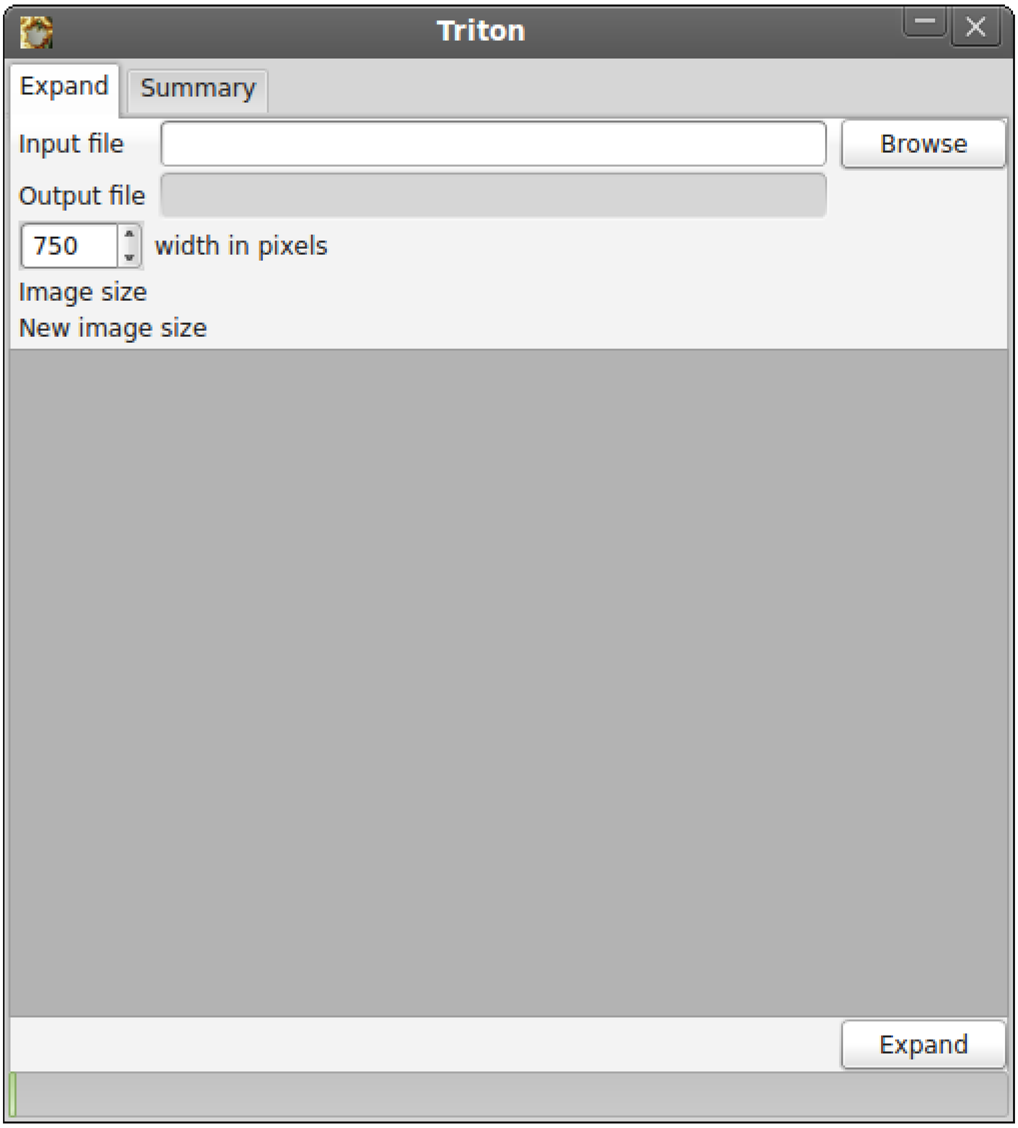
The grid used to calculate the the fractals, the calculating area, is larger than the display area and only the points that are in the display area are displayed. The orbit fractals plotted by Saturn consist of the visible parts of the orbits plotted for every row and column in the calculating area grid. The size of the calculating area can be up to 1000 times the display area, the number of columns and rows in the calculating can also be increased by altering the density.

5 Triton

Titan is a fractal expander and is the sister program to Neptune. The size of the pictures that can be saved by Saturn is restricted, however, seed files have all the necessary parameters for recreating the image, these file can be read by Titan and much larger images can be created.

That's is Triton's job in a nutshell, without Neptune it's useless.

This is what Triton looks like when it is first started.



The area bounded by the the horizontal lines separating the width in pixels and the expand button is the display area, once a file has been loaded an image is displayed and it will scaled and scroll bars may appear.

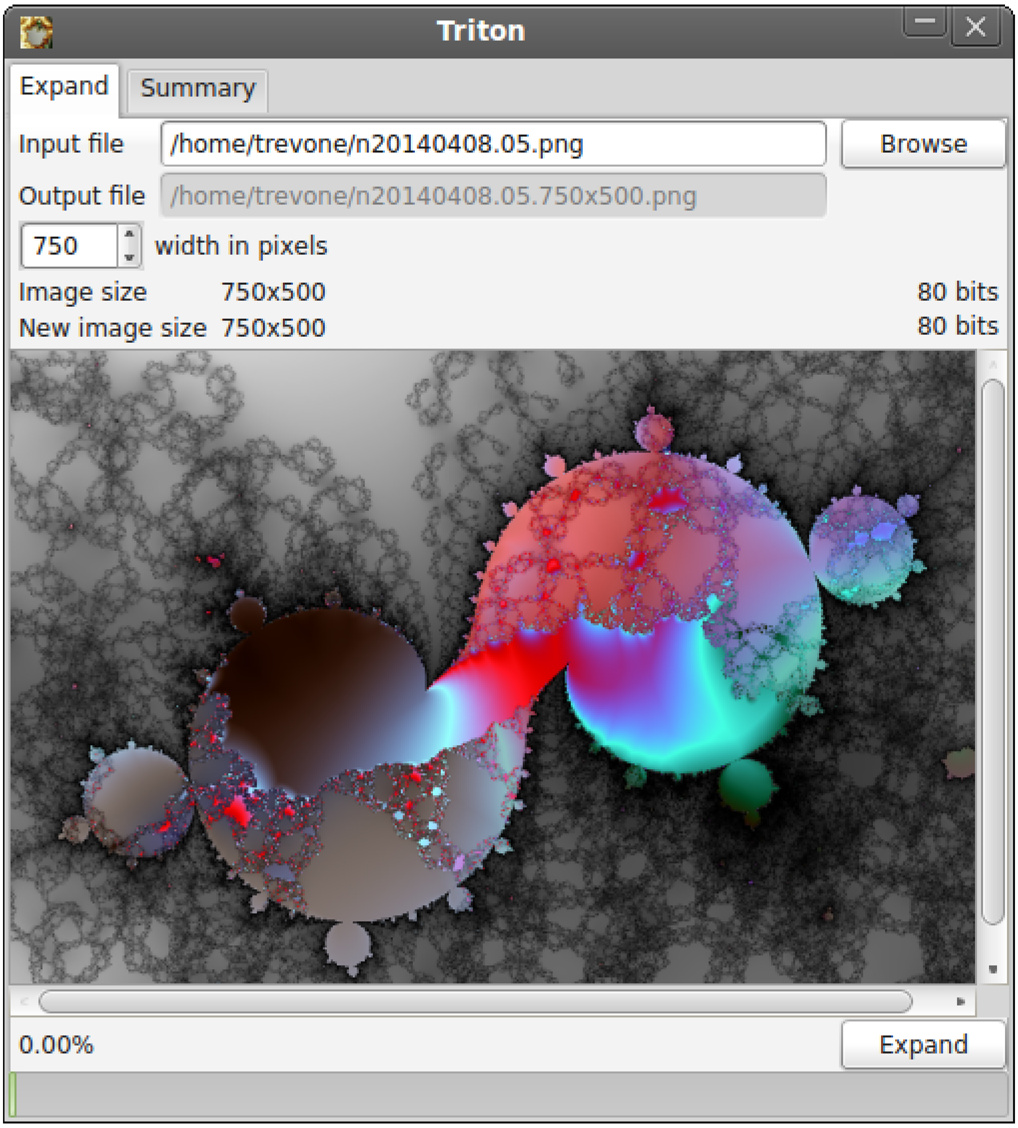
5.1 Loading a Seed File

A seed file can be loaded either by using the browse button or by dragging a seed file and dropping it on the display area. The input file entry box is used to display the input file name and can not be edited.

When a seed file has been successfully loaded its image will be displayed in the display area, its size and the expansion size are displayed along with input and output file names.

The output file name is automatically generated by inserting the size of the output image before the extension ".png".

Triton with a seed file loaded:

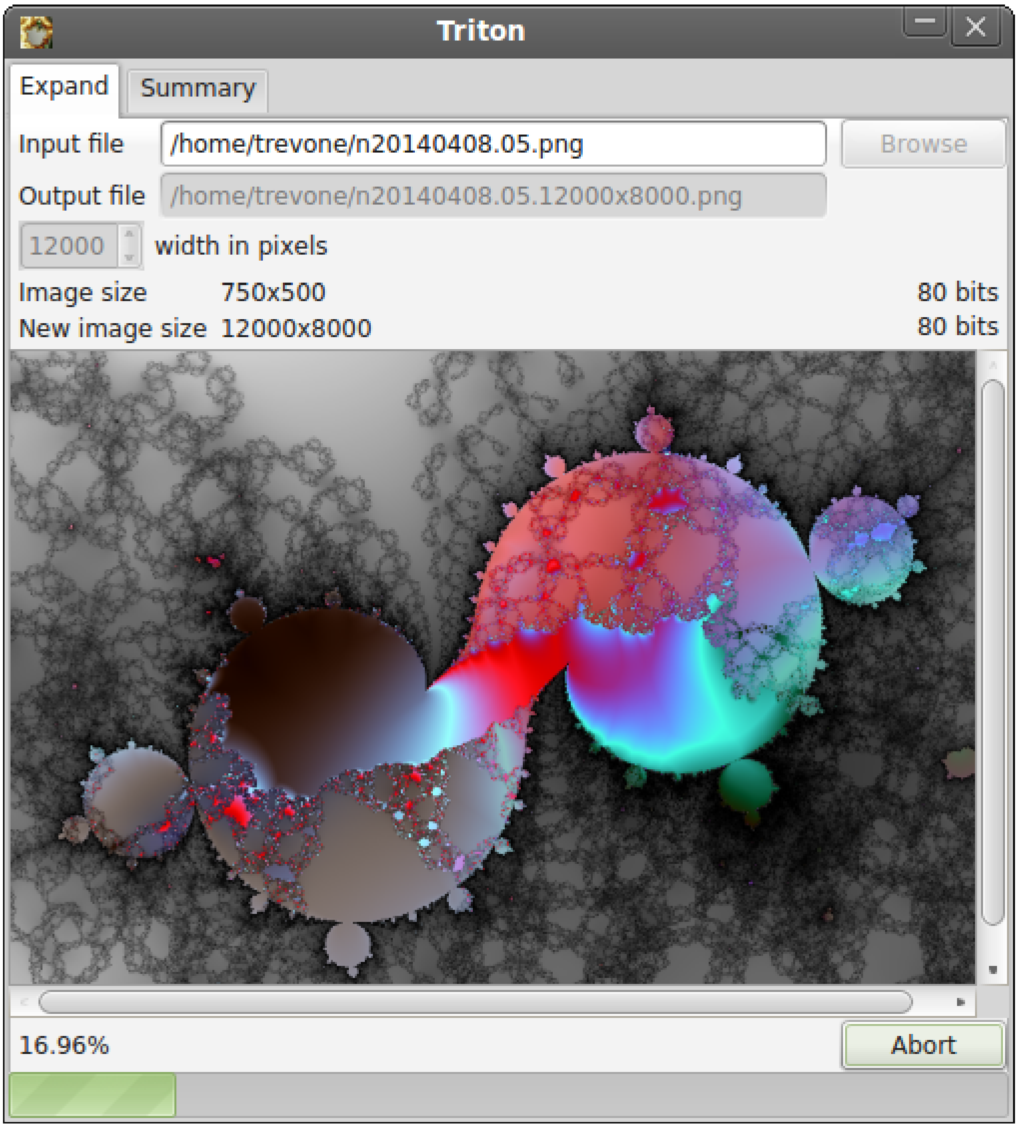


Once the seed file has been loaded the size of the required image can be set by setting the width in pixels. Number of precision bits required to calculate the image is displayed for the original sized image and for the new size.

5.2 Expanding the Image

To expand to image press the “Expand” button. Expansion will be started and progress shown in the progress bar. The “Expand” button turns into the “Abort” button which can be used to abandon the expansion.

Triton expanding an image:

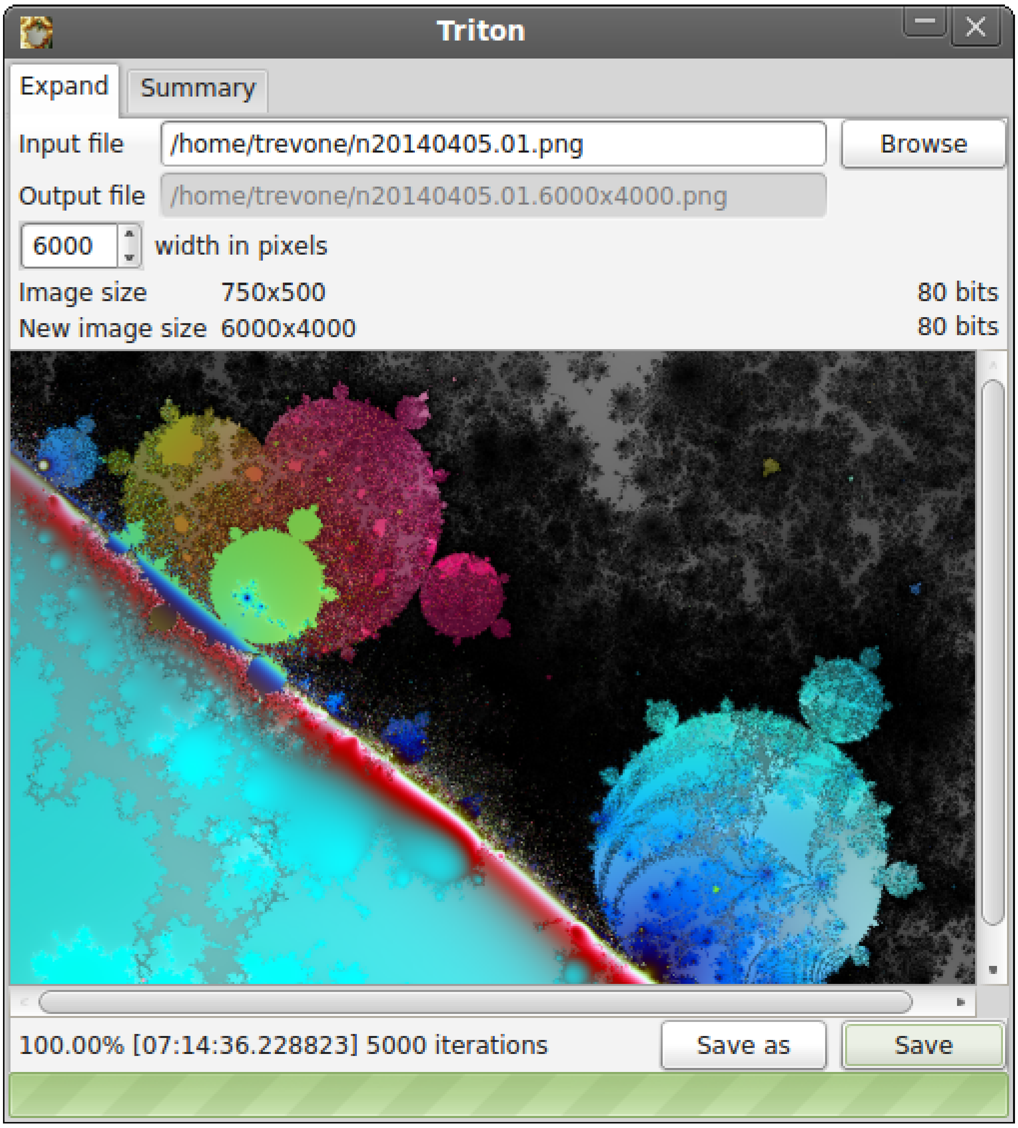


If the “Abort” button is pressed the button reverts to “Expand”. While expansion is in progress loading seed files and changing the output image size are disabled.

5.3 Saving the Image

When the expansion has completed two buttons are displayed, “Save” and “Save as”. Pressing the “Save” will save the newly generated image using the automatically generated file name. Pressing the “Save as” leads to a dialogue where the location and file name of the newly generated image can be specified. If the file of the same name already exists a dialogue message will be displayed and you will not be able to save the image.

Triton ready to save the expanded image.



Once the file has been saved the “Save” and “Save as” buttons will be replaced with the “Expand” button.

When expansion has completed loading of seed files and changing the output image size are enabled. ***CAUTION - Loading a seed file or changing the output image before saving the image will cause the expansion to be lost as the buttons will be replaced with the “Expand” button.***

The final image should be examined at full resolution as it is possible that the required precision wasn't correctly determined. If the image is clearly blocky or fuzzy then the only rememdy is to load the seed file back into Neptune so that the precision can be set manually, a new seed file should be saved which can then be used by Triton to produce a higher quality picture.

5.4 Expansion Time

The time taken to expand a fractal can be considerable, for a rough estimate of the time required to complete the expansion it is best to determine how long Tritan takes to produces a fractal image that is the same size as the image produced by Neptune. When the input file has been loaded just press the expand button, when the fractal is complete the duration will be displayed above the progress bar. To find the rough duration for the expansion: find the expansion factor and multiply by the duration.

expansion factor = new image size / image size

So to expand a 750x500 image to 7500x5000 image the expansion factor is 100, to expand it to 12000x8000 the expansion factor is 250.

For a duration of 10 seconds the rough expansion times would be 16 minutes 40 seconds for a 7500x5000 and 42 minutes 40 seconds for a 12000x8000 image.

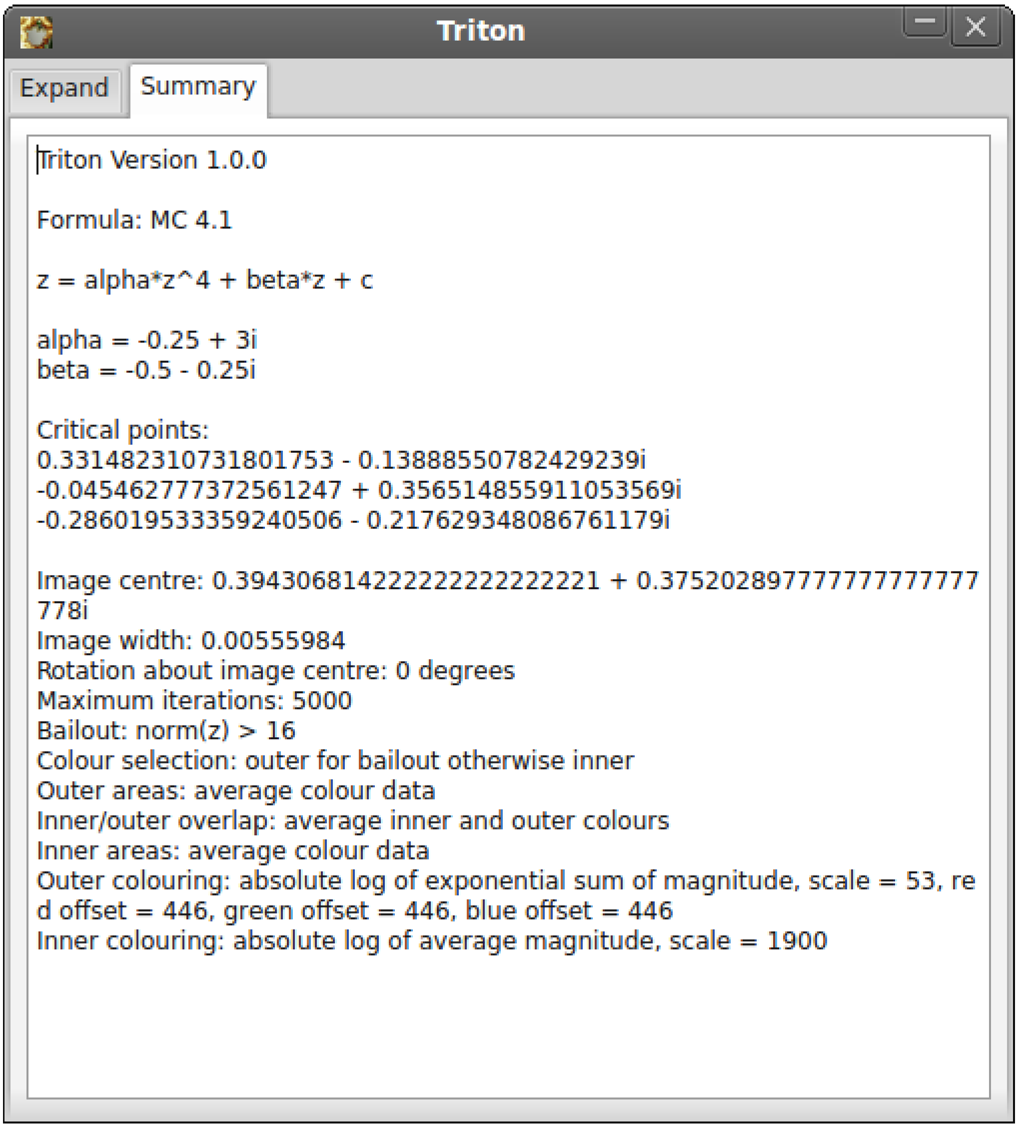
If the the number of bits for the new image is greater than that for the original image the time required for expansion will be greatly increased.

5.5 Memory Use

The maximum size of an expanded image is around 700 megapixels. Escape time fractals can be calculated on a point by point basis so only memory for the picture buffer is of any great size. Orbit fractals require a working buffer with up to 16 bytes per pixel for each pixel in the final image, so the memory requirement can be very large. If the required memory can not be allocated a message will be displayed saying so.

5.6 The Summary Tab

The summary tab displays a summary of the parameters used to generate the fractal image. The summary is displayed in a text box so it can copied and pasted as required.



5.7 Post Triton Processing

The files produced by Triton are PNG files, they can easily be loaded into other image manipulation programs for adding:

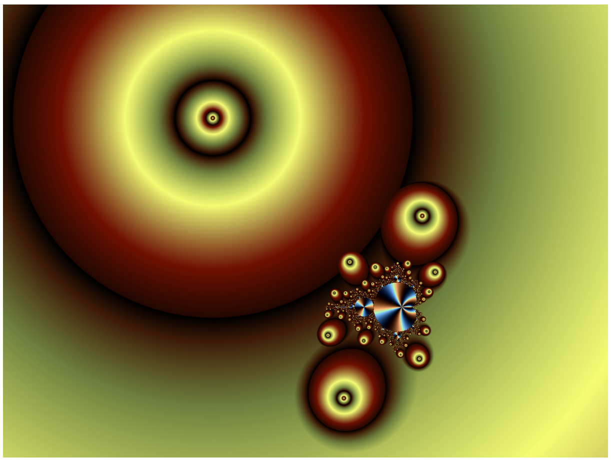
1. Text such as a title, creator's name or copyright notice.
2. Rotation from landscape to portrait.
3. Conversion to a different format such as JPEG.

Neither Saturn or Titan provide for anti-aliasing: to produce a higher quality image for a given picture size, a much larger image can be produced and then scaled down to the required size using an image manipulation program, the program you use should have a quality option such as interpolation, GIMP provides, Linear, Cubic and Lancz03.

6 Appendix – Orbit Trap Types

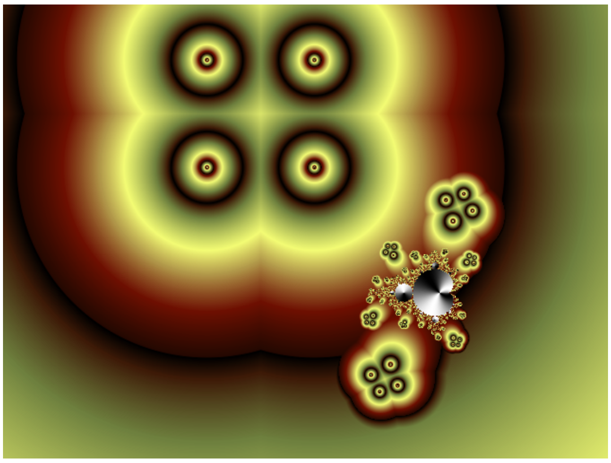
This appendix shows what the orbit trap shapes look like, some are obvious others are not for completeness pictures of all the orbit trap shapes are shown with any optional variations.

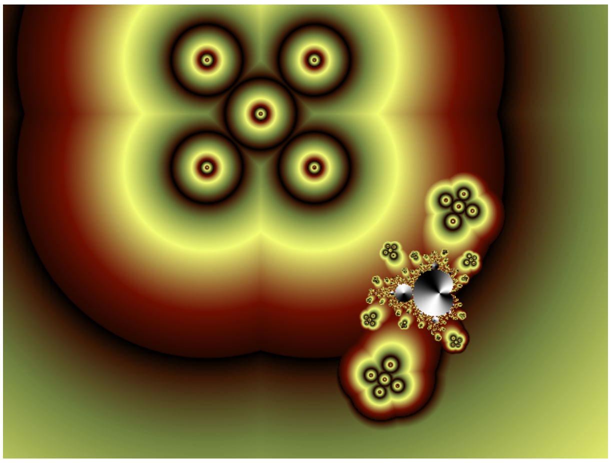
6.1 Point



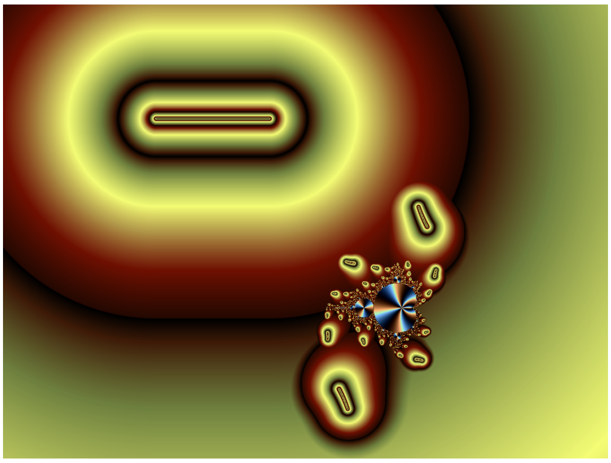
6.2 Four Points

Four point orbit trap, without and without central point.

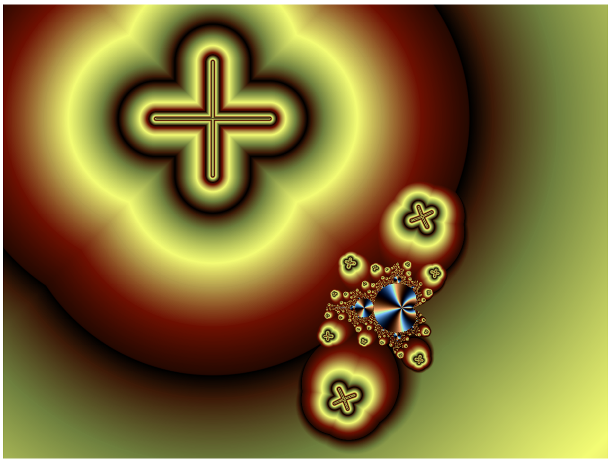




6.3 Line

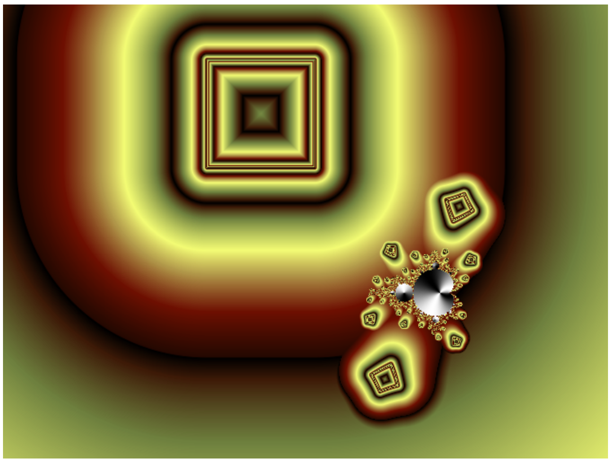


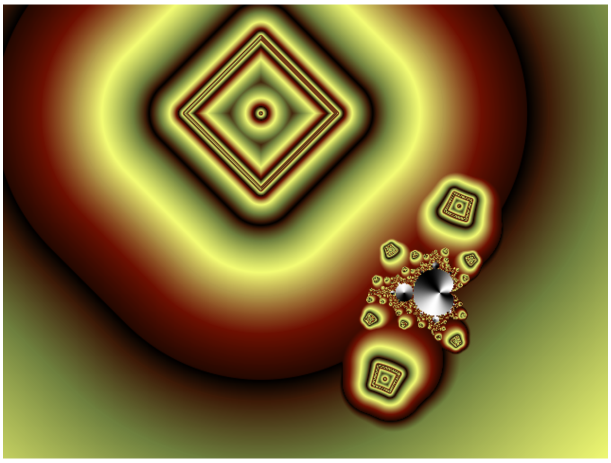
6.4 Cross



6.5 Square

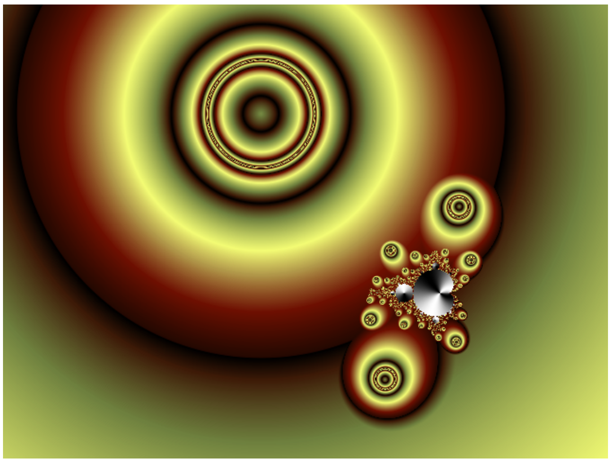
Square orbit trap, without the central point and with central point and rotated 45 degrees.

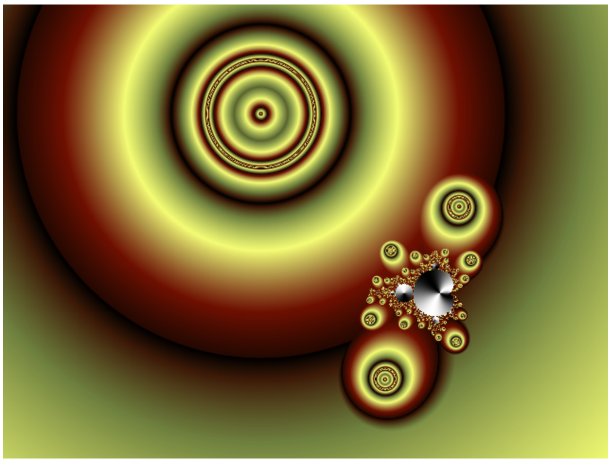




6.6 Circle

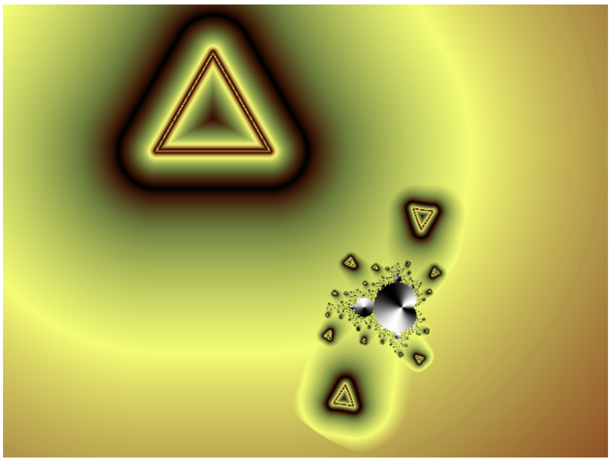
Circle orbit trap without and with central point.

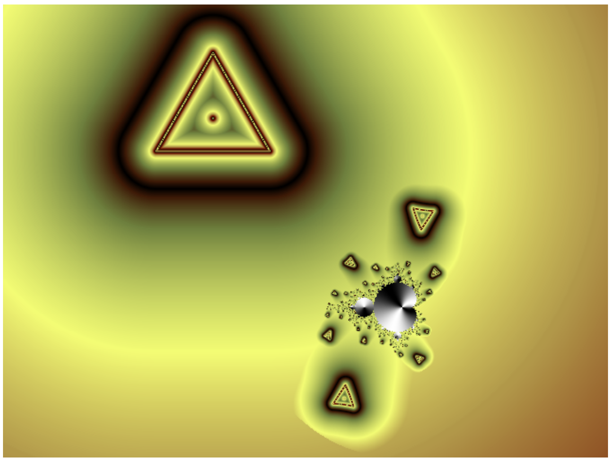




6.7 Triangle

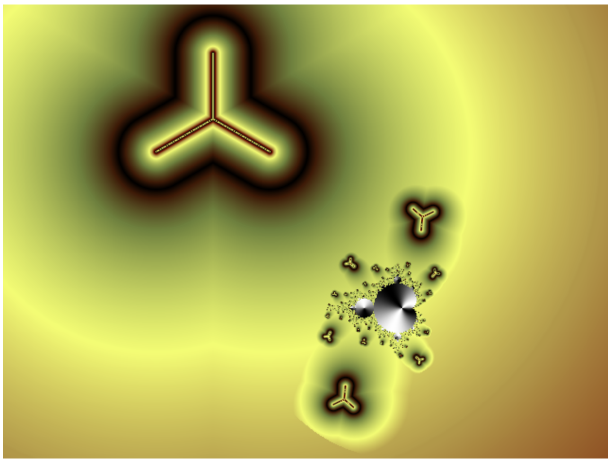
Triangle orbit trap without and with central point.





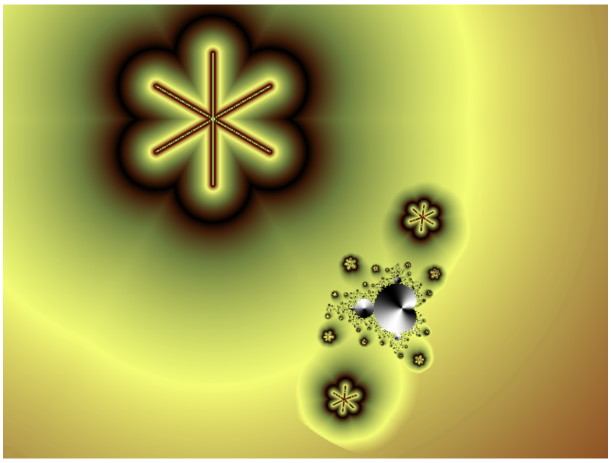
6.8 Triform

Triform orbit trap, no optional central point.

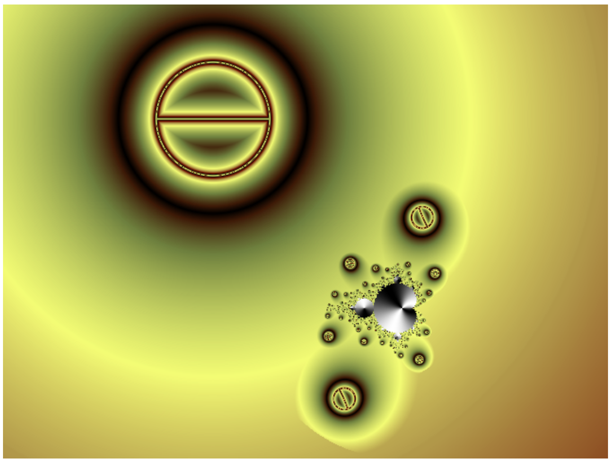


6.9 Asterisk

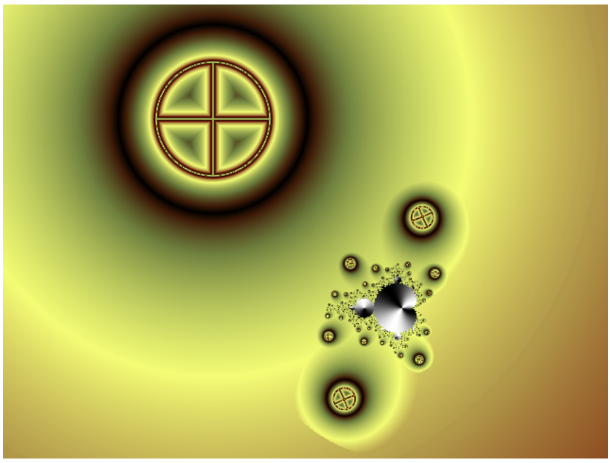
The asterisk orbit trap, again, no optional central point.



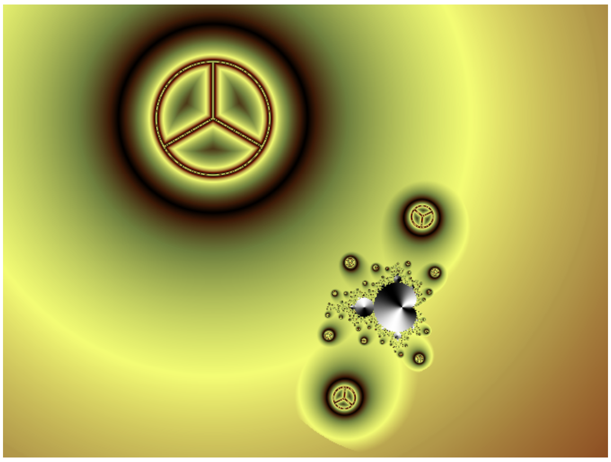
6.10 Circle Line



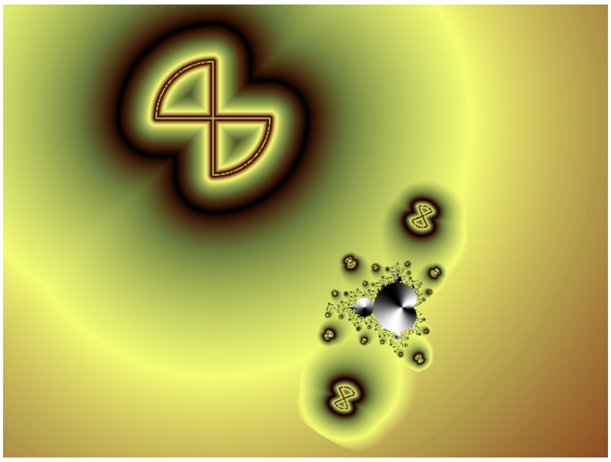
6.11 Circle Cross



6.12 Circle Triform

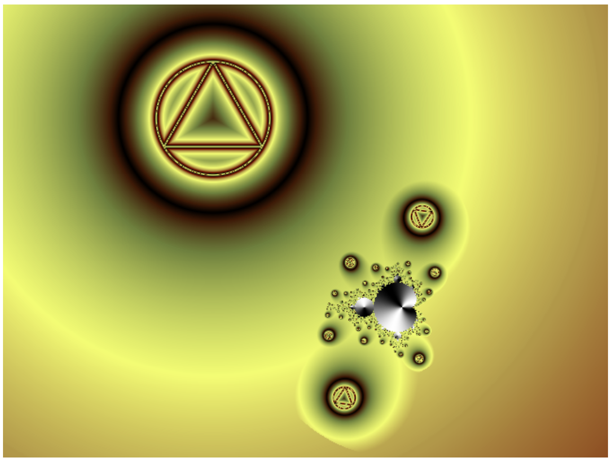


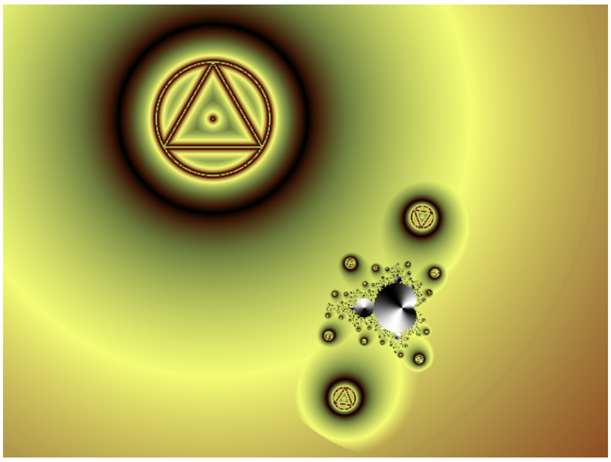
6.13 Two Quarter Circles



6.14 Circle Triangle

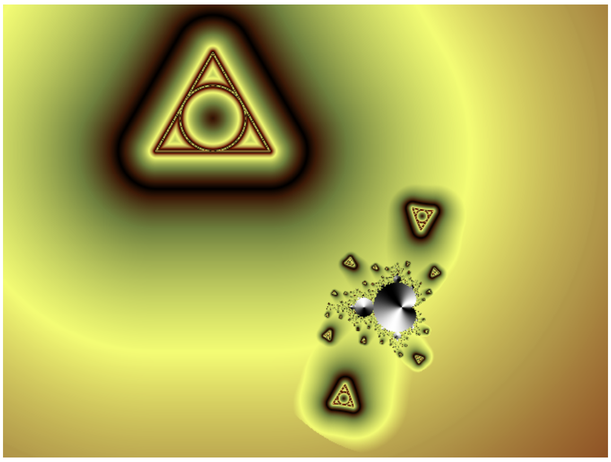
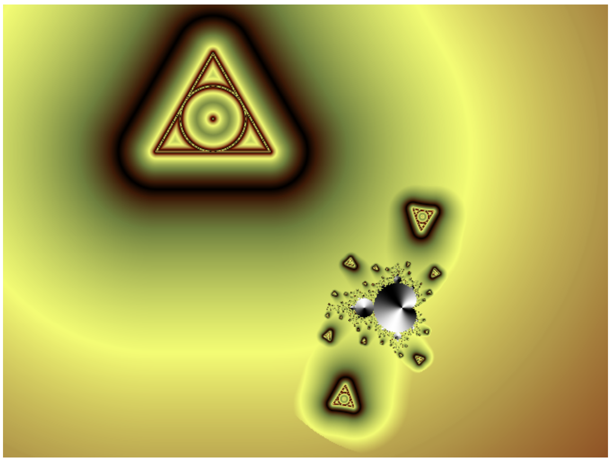
Circle Triangle orbit trap without and with optional central point.





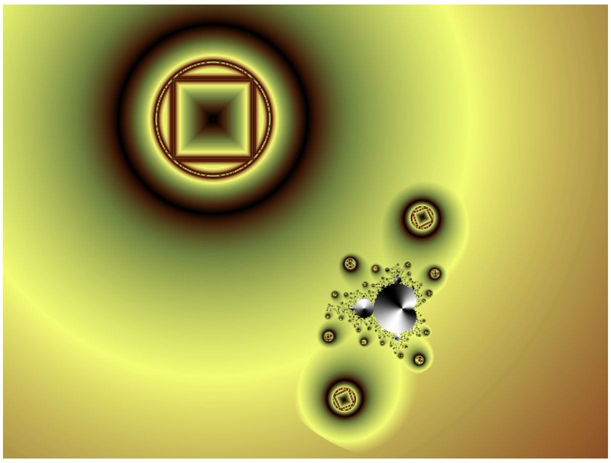
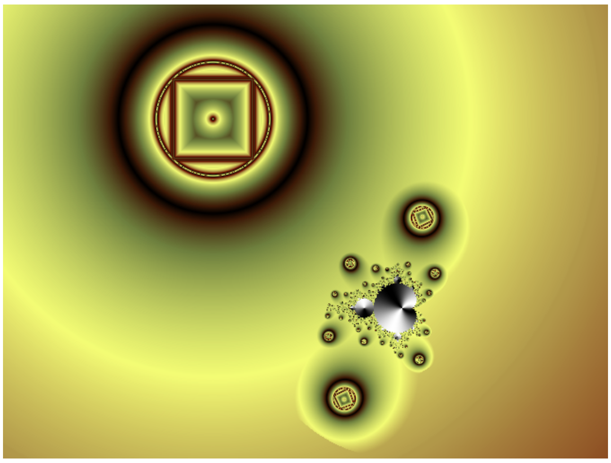
6.15 Triangle Circle

Triangle circle obit trap without and with central point.

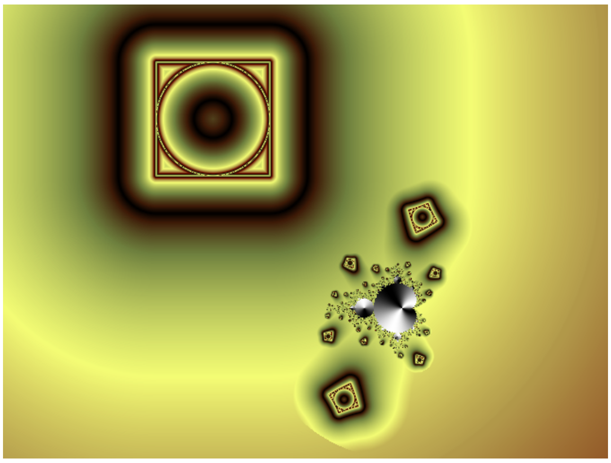
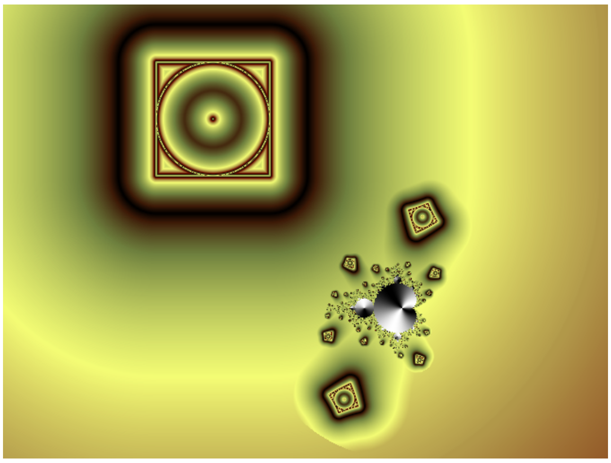
6.16 Circle Square

Circle square orbit trap without and with central point.

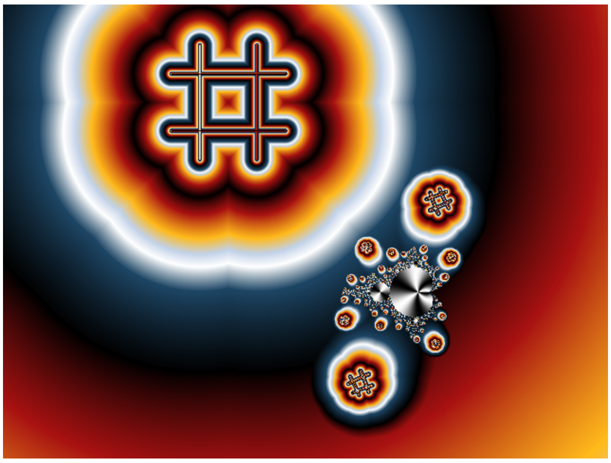
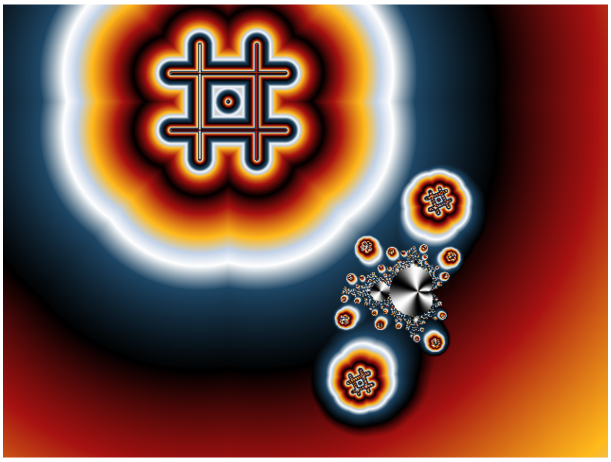
6.17 Square Circle

Square circle orbit without and with central point.

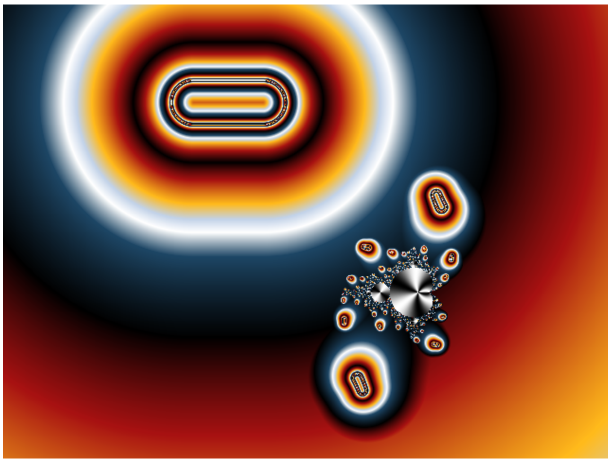
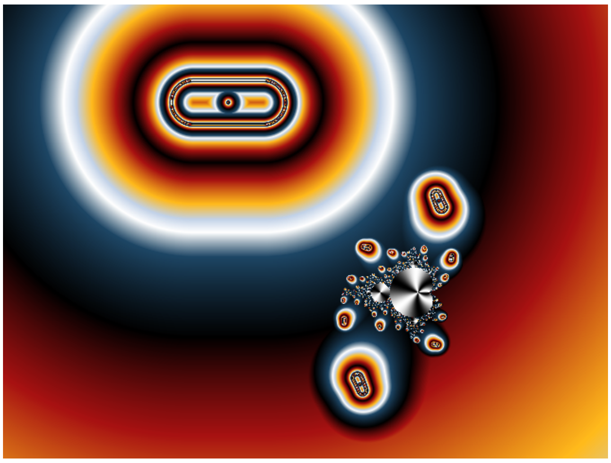
6.18 Octothorpe

Octothorpe (or pound, hash, sharp) orbit trap without and with central point.

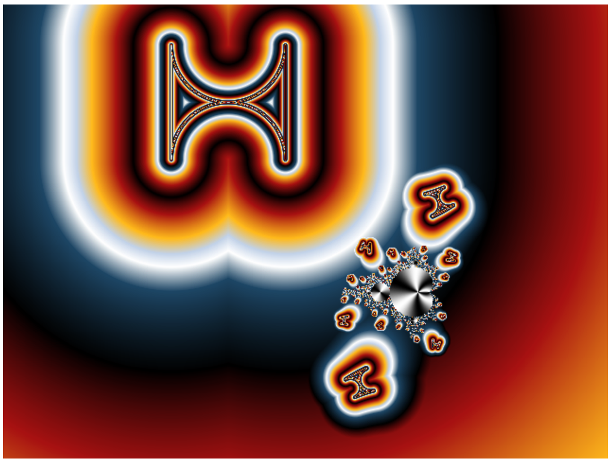
 

6.19 Running Track

Running track orbit trap without and with central point. The length of the straight lines and the semi-circles are the same.

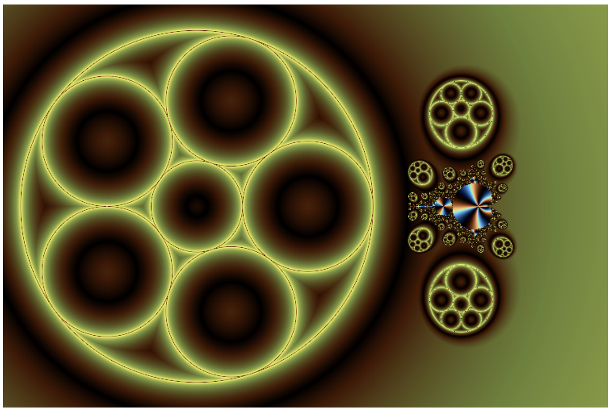
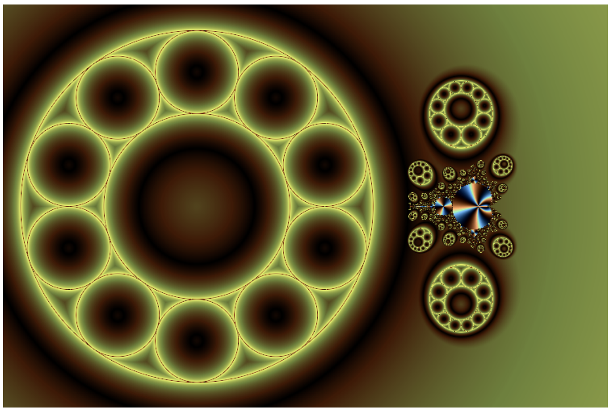
 

6.20 Pinch

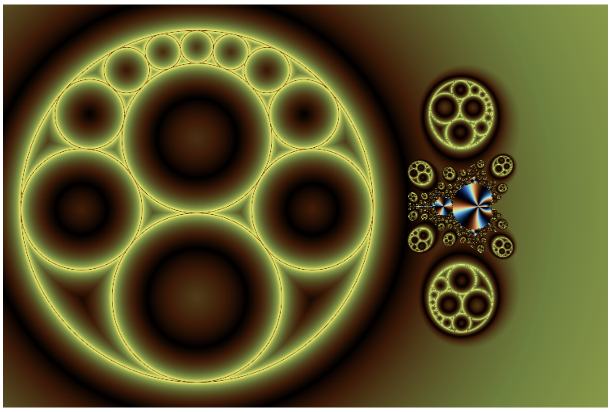
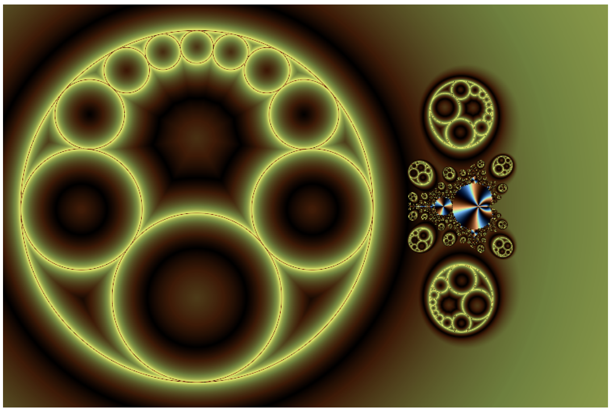


6.21 Steiner Chain

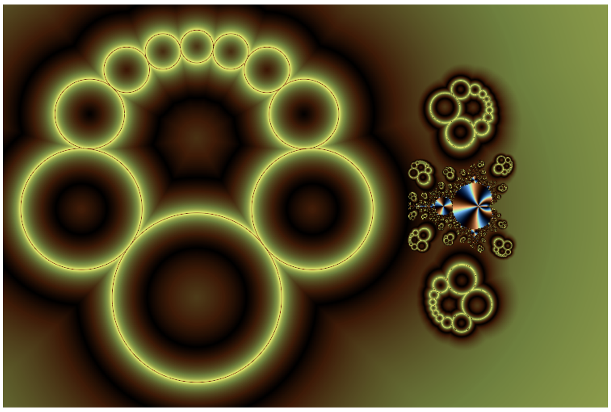
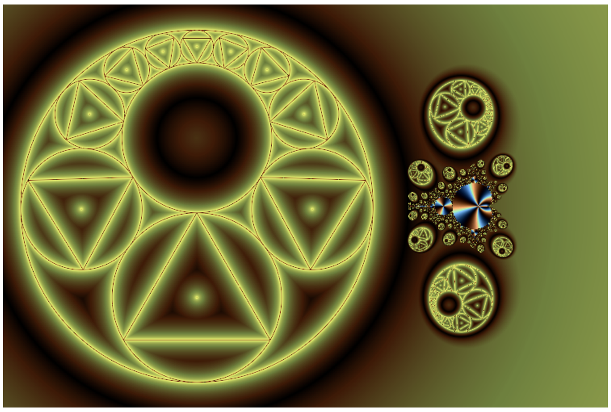
The Steiner Chain used an orbit can be configured so that its appearance can be varied widely.

Two annular Steiner Chains one with a chain length of 5 and the other with a chain length of 10.

Two Steiner Chains of length 10, inversion distance of -0.5 and rotation of 90 degrees, the second one has its inner circle omitted.

The same Steiner Chain tfirstly with both the inner and outer circles omitted and secondly with all options enabled.

7 Appendix - Statistics

Several of the colouring methods collect statistics for use in determining the value to use to select the colour. This appendix contains the statistic formulae for the non-obvious statistic values.

7.1 Range

Simple the difference between the maximum and minimum values.

range = maximum – minimum

7.2 Variance

The variance is calculated from the sum of the results of all the iterations (sum) and the sum of the squares of the results of all the iterations (sum2) and the number of iterations using this formula:

variance = (sum2 – ((sum*sum) ÷ iterations)) ÷ iterations

This is the final stage of the "naïve algorithm" which just requires that sums of the values and squares of the values are accumulated. I've checked the variance obtained using this algorithm on a small set of data and calculated the actual variance and the result using the "naïve algorithm" is just wrong. However as we using the values just to select colour it really doesn't matter.

7.3 Standard Deviation

The standard deviation is the square root of the variance. This uses the the variance obtained using the "naïve algorithm" so its values are also wrong.

7.4 Co-efficient of Variation

The co-efficient of variation is the standard deviation divided by the mean (average). This uses the the variance obtained using the "naïve algorithm" so its values are also wrong.

7.5 Fractal Dimension

The formula I'm using for this is:

fractal dimension = (sum2 – ((sum*sum) ÷ iterations)) ÷ range

I don't know whether this formula means any thing at all, although in all probability it does not calculate the fractal dimension. The sum2 and sum values are the same as for variance.

7.6 Exponential Sum

The exponential sum is an accumulation of the exponential of the result of each iteration, so e^z is added to the running total. The total is used to determine the colour.

7.7 Exponential Inverse Change Sum

The exponential inverse change sum is an accumulation of the negative inverse of the difference in magnitude between the result of the each iteration and the corresponding previous iteration, so if z is the result of an iteration and old_z is the result of the previous iteration then the following is added to the running total:

$e^{(-1 \div \mathrm{abs}(z - \mathrm{old\_z}))}$

the total is used to determine the colour.

8 Appendix – Neptune Memory Use

In order for Neptune to display a full fractal image every few iterations or so and to allow for colour map changes without re-calculation it reserves a significant amount of memory. The memory allocated is dependent on the size of the image and the type of fractal. The example memory allocated is for the largest screen size of 1280x1280.

8.1 Escape Time Fractal

737.5 Megabytes of working memory is allocated. This is very large so the more memory your computer has the better.

8.2 Orbit Fractals

1.5 Megabytes of working memory is allocated. There are three types of or orbit plots an orbit may not be plotted and that can only be determined after the orbit has been calculated so a buffer must be allocated for the orbit, (20 bytes times the orbit length). An orbit buffer is allocated for each thread used to calculate the fractal. On a single processor one thread is used, dual core uses 2, quad core uses 4 etc., the processors may also support 2 hardware threads per core so those vales will be doubled.