

FRACTALS IN POLAR COORDINATES

I have written two programs to calculate Mandelbrot and Julia-sets in polar coordinates. The first program called [pfRACT](#) calculates the actual iterations into a special file (see [specs](#)), and the other program is then used for [colorizing](#) the results to get a PPM-image. This can be converted to other formats by [PBMPLUS](#)-library.

These programs really are quite simple and they should compile with minimal effort. Check the macros UINT16 and UINT32 so that they are defined as a 16-bit unsigned integer and a 32-bit unsigned integer on your machine. Check also the macros for float and double versions of the math functions. Both of the programs can be compiled by using double precision arithmetics (default) or single precision floating point arithmetics by defining NODOUBLE -flag. The *pfRACT* program can be compiled either for 16-bit integers (default) or 32-bit integers (by BIG_ITERATIONS -flag). The *colorize* program can read both of these 16- and 32-bit files.

Below are the explanations of these programs and their command-line options. Take a look at [some examples](#) too.

PFRACCT

This program does the actual calculation of the fractals. It's output is a raw-file where the iterations are stored. The coordinate system is such that the angular component goes to the y-axis, and the radius goes to the x-axis of the resulting image. In addition, the x-axis (radius) is normalized with exp-function so that angles are preserved in this mapping. Here is the list of available options:

```
> pfRACT -h
```

```
-----
| Mandelbrot- and Julia-set fractals in polar coordinates. |
| (C) Mika Seppa 1996 (email: my.email.address)           |
|-----|
```

```
Usage: pfRACT [options] [outfile.raw]
```

```
-cx ...      : Start-point x-coordinate (0)
-cy ...      : Start-point y-coordinate (0)
-help       : This help
-iterations ... : Maximum iterations (255)
-julia      : Calculate Julia-set
-rad ...    : Minimum radius (-5)
-size ...   : Size of the image (256x256)
-x ...      : Origin x-coordinate (0)
-y ...      : Origin y-coordinate (0)
```

-cx,-cy:

The seed-point from Mandelbrot-set for the Julia-set.

-help,-h:

The help shown above.

-iterations:

The maximum iterations value.

-julia:

Calculate Julia-set instead of Mandelbrot-set.

-rad:

Minimum radius. This is the value before the exponential mapping, so negative values just mean radiuses below 1. The maximum radius is calculated from this value and the size of the target image.

-size:

Size of the target image. Use very wide images for background purposes.

-x,-y:

The origin of the polar-coordinates. If you find a nice region from the Mandelbrot-set, use these values for -cx,-cy -options and try to calculate a Julia-set.

HOWTO

I use xfractint and [MandelSpawn](#) for looking Mandelbrot and Julia-sets. When I find a nice region, I pick the coordinates from these programs and run the *pfRACT* program. You might notice that the *pfRACT*-options are similar to *MandelSpawn*-options, so the coordinates can be simply cut-n-pasted. Then I just try to find nice minimum radius, maximum iteration value and the image size. After I get the composition right, I test for good colors.

COLORIZE

This program reads the raw-output from the *pfra*ct program and colorizes it into a PPM-file. With the help of options shown below, one has a simple control over the colorizing process.

```
> colorize -h
```

```
-----
|           Iteration file colorizer           |
| (C) Mika Seppa 1996 (email: my.email.address) |
|-----|
```

```
Usage: colorize [options] [infile.raw] [outfile.ppm]
```

```
-fadeout ...      : Fades out to white (0)
-fadedepth ...    : Depth of fading (20)
-help            : This help
-info           : Show information about file
-inset .. .. .   : Color in the set (0,0,0)
-scale ...       : Color scale (1)
-spawn ...       : Color angle to spawn (360)
-start ...       : Starting color angle (0)
--              : Use stdin instead of file
```

-fadeout:

Iteration count below which the color goes to white.

-fadedepth:

How many steps (iterations) to use for fading (saturating) the white to a color.

-help,-h:

The help shown above.

-info:

Some information about the raw-file.

-inset:

What color to use for pixels inside the Mandelbrot or Julia-set. Default value is black.

-scale:

Determines how fast the color changes in the low and high-iteration areas of the image. A kind of gamma correction for iteration values.

-spawn:

The color-angle to spawn in degrees.

-start:

The offset for color-angle in degrees.

--:

Use -- in place of inputfile if you want to specify an outfile, but the input comes through a pipe.

HOWTO

Just try different options to get a pleasing image. It seems to be good to increase the *-scale* option if the maximum iterations in the raw-file is high.