FRACTAL GENERATOR

January 18, 2025

Contents

1 Introduction:

This implementation of a fractal generation algorithm, while not unique, does represent an attempt to keep it simple and efficient. The focus is on computation using complex data types and parallelization, along with an embedded macro language, to provide a simple but flexible capability to produce fractal images. Output is provided in a simple, standard portable bitmap format that can be piped to appropriate translation utilities to produce images files, avoiding the need to implement support for any specific output format.

The algorithm used in this implementation is focused on the Mandelbrot set and related Julia sets arising from the iterative solution of the simple equation $z = z^2 + c$ where z and c are complex numbers ($c = [a + bi]$ where $i^2 = -1$). The Mandelbrot set is defined as the set of complex numbers c for which the function $z = z^2 + c$ does not diverge to infinity with an initial value $z_0 = 0$. The rate at which values of z diverge can be visualized by assigning a color, based

on the iteration count n that occurs when the absolute value of z exceeds the escape value 2 (i.e., $|z_n| > 2$), to an image pixel placed at a coordinate derived from the real and imaginary parts of c. An iteration count that exceeds a threshold, without $|z|$ exceeding the escape value, implies that the assigned value of c is a "likely" candidate for inclusion in the Mandelbrot set. The resulting "fractal" visualization is in fact a graphical depiction of the rate of divergence of z for all values of c in the defined view, that are by definition, not part of the Mandelbrot set.

In contrast, a Julia set is defined as the set of initial values of z_0 for which the function $z = z^2 + c$ does not diverge to infinity for a constant static value of c. For initial values of z that diverge, the resulting iteration count is visualized using a image color value assigned to the pixel at the coordinate derived from the real and imaginary parts of z_0.

If the specified number of threads defined is greater than 1 then image computation will be allocated to each thread in column sequence for the most equitable and efficient division of computational resources. Hopefully, it is obvious that the number of threads specified should be equal to or less than the available hardware cores or hyper threads.

2 Usage:

From command shell type:

-> fractal [rexx script file]

2.1 Embedded REXX Macro Language

This version of Fractal registers itself as an embedded Rexx interpreter which will process commands not directly understood by the Rexx macro processor. This effectively adds all the capabilities of the Rexx macro language to Fractal program macros.

3 Command Reference:

3.1 reset(XRES, YRES, MAXITER, NTHREAD)

This command resets the image resolution, maximum number of iterations, and number of allocated threads. The resolution (XRES, YRES) defines the screen resolution of the generated image.

An example of this command is:

```
reset 1600 1200 4000 4
```

3.2 spectrum(SHIFT, [INDEX, COLOR]...)

This command sets the palette of color spectrum to the defined set of colors, with the interpolation between color values defined by the difference in index values between entries. The SHIFT value defines the starting color index. If the iteration value exceeds the maximum color index then the modulus of the iteration (with SHIFT applied) value against the last color index will be used to select the color interval to interpolate between.

An example of this command is:

```
palette 0 25 000764 50 206bdd 100 edffff 150 ffaa00 215 310230
```

3.3 view(XC, YC, SIZE, SCALE)

This command defines the fractal view center coordinates and size. If the size parameter value divided by the horizontal resolution (XRES) is less than double epsilon ($2.220446E-16$) then arbitrary precision math will be used, which will be a much higher cost in computation, but allows for deep zoom of the Mandelbrot set. The SCALE argument (default 0) is an integer that sets the grid size to 2^{SCALE} allowing for reduced computations with a sacrifice in resolution. For a view size, that triggers arbitrary precision math, this scale value can be useful to obtain a generated fractal in shorter times, with subsequent generations set to lower scale value for greater resolution.

An example of this command is:

```
view 0.0 0.0 2.6
```

3.4 fractal(TYPE, [CX, CY,] CTYPE)

This command initiates execution of a fractal image defined by the specified type (TYPE which currently only supports a value of 0 indicating the Mandelbrot set), escape type (CTYPE which may be the default 0 for $re(z)^2 + im(z)^2$, 1 for $re(z^2)$, or 2 for $im(z^2)$), and, if specified, the Julia set defined at CX, CY.

An example of this command is:

```
fractal
fractal 0 0.355534 -0.337292
```

3.5 save(FILENAME)

This command will save a generated fractal solution to a file.

The FILENAME argument specifies a file to save the resulting fractal solution to. The file output will be in an interchange file format (IFF) format with binary chunks specific to fractal solution data elements. If the file name begins with a pipe ('|') then the output will be provided as a standard input stream to the specified process. An ancillary program 'ffftoppm' is provided for converting a fractal solution IFF file into a portable pixmap (PPM).

An example of this command is:

```
save(|ffftoppm|pnmtojpeg -quality 90 > output.jpg)
```

3.6 read(FILENAME)

This command will read in a previous generated fractal solution, including all parameters used. These parameters may be accessed in a Rexx macro from the following stem variables:

- display.xres
- display.yres
- display.scale
- fractal.xc
- fractal.yc
- fractal.size

An example of this command is:

```
read fractal.fff
```

3.7 display

An example of this command is:

```
display
```

4 Example

```
/* macro for Fractal */
numeric digits 96
xc = -0.17064378523731846796177582519278165990750358885420006719704 17
yc = -1.08968429519467513226778417588392559760815172194470415367575 00
w = 1.25E-55
reset 640 640 50000 16
spectrum 0 25 000764 50 206bdd 100 edffff 150 ffaa00 215 310230
view xc yc w
fractal
'save(|ffftoppm|pnmtopng -compression 9 > fractal.png)'
exit
```