

programas de trazado de rayos: *ix*

M. Fernández-Guasti 

March 25, 2016

e-mail: mfg@xanum.uam.mx, url: <http://luz.izt.uam.mx>

1 introducción

Se pueden realizar visualizaciones (en 2D) de la estructura tridimensional mediante trazado de rayos. Es esencial utilizar programas de código abierto para poder modificarlos. Más aún, los programas de código abierto, que además son gratuitos, son la forma de colaborar y avanzar en el conocimiento de la humanidad.

2 programa para *ix*

Para incluir el *fractal con escatores imaginarios bajo iteración cuadrática*, codificado *ix*, como abreviación de *ixbalanqué*, se debe modificar la relación de recurrencia y la magnitud de acuerdo al álgebra de escatores imaginarios.

La iteración, ahora de tres funciones, se realiza con un lenguaje eficiente y rápido, por ejemplo C++. Entre corchetes aquí abajo, las relaciones recurrentes. Sería preferible la siguiente notación, donde el escalar es **s**, y las componentes directoras **x,y**:

```
{
    double s2 = z.s * z.s;
    double x2 = z.x * z.x;
    double y2 = z.y * z.y;

    double news = s2 - x2 - y2 + (x2 * y2) / s2;
    double newx = 2.0 * z.s * z.x * (1 - y2 / s2 );
    double newy = 2.0 * z.s * z.y * (1 - x2 / s2 );

    z.s = news;
```

```

        z.x = newx;
        z.y = newy;
    }

```

3 Mandelbulber modificado para incluir *ix*

En este ejemplo, se utilizó el programa Mandelbulber v 2.07-1, desarrollado por el director del proyecto Krzysztof Marczak y los programadores Krzysztof Marczak, Sebastian Jennen, Graeme McLaren, Bernardo Martelli.

- añadir en Mandelbulber, v 2.07-1 en el archivo /src/fractal__formulas.cpp

```

/** quadratic iteration in imaginary scator algebra */
void ImaginaryscatorPower2Iteration(CVector3 &z)
{
    double x2 = z.x * z.x;
    double y2 = z.y * z.y;
    double z2 = z.z * z.z;

    double newx = x2 - y2 - z2 + (y2 * z2) / x2;
    double newy = 2.0 * z.x * z.y * (1 - z2 / x2 );
    double newz = 2.0 * z.x * z.z * (1 - y2 / x2 );

    z.x = newx;
    z.y = newy;
    z.z = newz;
}

```

- añadir en fractal__formulas.hpp

```
void ImaginaryscatorPower2Iteration(CVector3 &z);
```

- añadir en fractal__list.cpp

```

fractalList->append(sFractalDescription("Imaginary scator Power 2",
                                         "imagsca_power_2",
                                         fast_imagsca_power2,
                                         analyticDEType,
                                         logarithmicDEFunction,
                                         cpixelEnabledByDefault,
                                         10));

```

- añadir en fractal__list.hpp

```
fast_imagsca_power2 = 152,
```

- añadir en compute_fractal.cpp

```
case fast_imgsca_power2:
{
    ImaginaryscatorPower2Iteration(z);
    break;
}
```

- tambien modificar y añadir en linea 711

```
// r calculation
// r = sqrt(z.x * z.x + z.y * z.y + z.z * z.z + w * w);
switch(fractal->formula)
{
    default:
    {
        r = sqrt(z.x * z.x + z.y * z.y + z.z * z.z + w * w);
        break;
    }
    //scator magnitudes
    // magnitude in imaginary scator algebra
    case fast_imgsca_power2:
    {
        r = sqrt(z.x * z.x + z.y * z.y + z.z * z.z
            + (z.y * z.y * z.z * z.z) / (z.x * z.x) );
        break;
    }
}
```

- tambien añadir: en switch (formula) línea 836

```
case fast_imgsca_power2:
```

- añadir en /usr/share/mandelbulber2/language/ qt_data_en.ts (en dos lugares) justo despues de "../qt_data/fractal_mandelbulb_power_2.ui"

```
<location filename="../qt_data/fractal_imgsca_power_2.ui" line="14"/>
```

```
<location filename="../qt_data/fractal_imgsca_power_2.ui" line="20"/>
```

- copiar archivo fractal_mandelbulb_power_2.ui en /usr/share/mandelbulber2/qt_data con nombre

```
fractal_imgsca_power_2.ui
```

- para compilar e instalar

```
cd makefiles
qmake mandelbulber.pro
make all
cd ..
./install
```

4 parámetros

Los parámetros importantes son:

1. que se detenga en el numero maximo de iteraciones
2. la estimación de distancia realizarla con el método Delta DE
3. deseable, aunque no esencial, poner un número grande como condición para salir de las iteraciones

En las líneas aqui abajo, se muestran los parámetros típicos de un archivo `.fract` de Mandelbulber

```
# Mandelbulber settings file
# version 2.07
# only modified parameters
[main_parameters]
aux_light_enabled_1 true;
aux_light_intensity_1 1;
aux_light_position_1 1 -3 -2;
bailout 100000;
camera 0.1 -3 0;
camera_distance_to_target 3.001666203960727;
camera_rotation 1.909152432996377 0 0;
camera_top 0 0 1;
delta_DE_method 1;
flight_last_to_render 0;
formula_1 152;
iteration_threshold_mode true;
keyframe_last_to_render 0;
use_default_bailout false;
```