Fractals on non-euclidean metric

Yeray Cachón Santana

April 1, 2018

As far as I know, there is no a study on fractals on non euclidean metrics.This paper proposes a first approach method about generating fractals on a non-euclidean metric. The idea is to extend the calculus of fractals on non-euclidean metrics. Using the Riemann metric, there will be defined a non-euclidean modulo of a complex number in order to check the divergence of the series generated by the Mandelbrot set. It also shown that the fractals are not invariant versus rotations. The study will be extended to the quaternions, where is shown that the study of fractals might not be extended to quaternions with a general metric because of the high divergence of the series (a condition in order to generate a fractal is selecting bounded operators). Finally, a Java program will be found as example to show those kind of fractals, where any metric can be defined, so it will be helpful to study those properties.

Contents

1 Riemann distance

As an introduction, let's using the length of the arc on a Riemann geometry:

$L = \int_a^b \sqrt{g_{ij} \frac{dx^i}{dt} \frac{dx^j}{dt}} dt$

Once chosen a base, the metric tensor will be represented by a matrix G.

$g = \sum_{i,j=1}^{n} g_{ij} dx^i \otimes dx^j$

$$G = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & \ddots & & \\ \vdots & & \ddots & \\ g_{m1} & & & g_{mn} \end{pmatrix}$$

For example, using the euclidean metric $g_{ij} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, the length of the arc will be the euclidean metric:

$L = \int_a^b \sqrt{\frac{dx}{dt}\frac{dx}{dt} + \frac{dy}{dt}\frac{dy}{dt}} dt = \int_a^b \sqrt{dx^2 + dy^2}$

When considering the Mandelbrot set:

$\begin{cases} z_0 = 0 \\ z_{n+1} = z_n^2 + c \end{cases}$,where c is a complex constant

This set is constructed following those steps:

1) Get a point of the complex plane

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

3) If the succession diverges, this element doesn't belong to a Mandelbrot set. If the succession converges, the element belongs.

Lemma: The distance on a euclidean metric is the the modulo of the complex number.

$z_n = x_n + iy_n$

$|z| = L = \sqrt{|x_n|^2 + |y_n|^2}$

Proof:

Taking the parametric equation:

$x_n = t$

$y_n = at$, the distance will be:

$L = \int_0^{x_n} dt \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} = \int_0^x \sqrt{1 + a^2} dt = x\sqrt{1 + a^2} = \sqrt{x^2 + (ax)^2} = \sqrt{x^2 + y^2} \blacksquare$

A compact topological space X is self-similar if there exists a finite set S indexing a set of non-surjective homeomorphisms $\{f_s : s \in S\}$for which $X = \bigcup_{s \in S} f_s(X)$. If $X \subset Y$. we call X self-similar if it's only a non-empty subset of Y such the previous equations for $\{f_s : s \in S\}$.

Let's call $\mathcal{L} = (X, S, \{f_s : s \in S\})$a self-similar structure.

The Mandelbrot set is self-similar around Misiurewicz points. A parameter c is a Misiurewicz point if $M_{k,n}$if it satisfies the equations:

$f_c^{(k)}(z_{cr}) = f_c^{(k+n)}(z_{cr})$ and

$f_c^{(k-1)}(z_{cr}) \neq f_c^{(k+n-1)}(z_{cr})$

so

$M_{k,n} = c : f_c^{(k)}(z_{cr}) = f_c^{(k+n)}(z_{cr})$, where

z_{cr} is a critical point

k and n are positive integers

f_c^kdenotes the k-th iterate of f_c

It doesn't depend on the metric chosen so, in a non-euclidean metric, the properties of the fractal may be conserved. This allows to extend the study of the fractals to a non-euclidean geometry.

2 Non-euclidean metric

In this example, let's use a non-euclidean metric, as it was shown previously:

$g = \sum_{i,h=1}^{n} g_{ij} dx^i \otimes dx^j$

$= \begin{pmatrix} cxy & 0 \\ 0 & 1 \end{pmatrix},$

Firstly, let's calculate the term:

$\sum_{ij} g_{ij} \frac{dx^i}{dt}\frac{dx^j}{dt} = cxy\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2$

$L = \int_0^{x_n} \sqrt{cxy\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} dt$

Using the parametric equation:

$x = Vt$

$y = Vat$, considering the complex number $z_n = x_n + iy_n$, the distance will be:

$L = \int_0^{t_n} \sqrt{cxy\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} dt = V\int_0^{t_n} \sqrt{cV^2at^2 + a^2}dt = aV\int_0^{t_n} \sqrt{1 + \frac{cV^2}{a}t^2}dt = aV\int_0^{t_n} \sqrt{1 + \left(tV\sqrt{\frac{c}{a}}\right)^2}dt$

With the substitution $u = tV\sqrt{\frac{c}{a}}$

$L = aV\int_0^{u_n} \frac{1}{V}\sqrt{\frac{a}{c}}\sqrt{1+u^2}du = a\sqrt{\frac{a}{c}}\int_0^{u_n} \sqrt{1+u^2}du$

$\int \sqrt{u^2+1}du = \frac{u\sqrt{u^2+1}}{2} + \frac{1}{2}ln\left(u + \sqrt{1+u^2}\right)$

So, $L = a\sqrt{\frac{a}{c}}\left\{\frac{u_n\sqrt{u_n^2+1}}{2} + \frac{1}{2}ln\left(u_n + \sqrt{1+u_n^2}\right)\right\}$

Now, taking in consideration:

$a = \frac{y_n}{Vt} = \frac{y_n}{x_n}$and $u = tV\sqrt{\frac{c}{a}} = x_n\sqrt{\frac{cx_n}{y_n}}$:

$L = \frac{y_m}{x_n}\sqrt{\frac{y_n}{cx_n}}\left\{\frac{1}{2}\frac{x_n}{y_n}\sqrt{c^2x_n^4 + cx_ny_n} + \frac{1}{2}ln\left(x_n\sqrt{\frac{cx_n}{y_n}} + \sqrt{1+\frac{cx_n^3}{y_n}}\right)\right\}$

So, the "modulo" on a complex number x+iy considering a non-euclidean metric is done by:

$L = \frac{y_m}{x_n}\sqrt{\frac{y_n}{cx_n}}\left\{\frac{1}{2}\frac{x_n}{y_n}\sqrt{c^2x_n^4 + cx_ny_n} + \frac{1}{2}ln\left(x_n\sqrt{\frac{cx_n}{y_n}} + \sqrt{1+\frac{cx_n^3}{y_n}}\right)\right\}$

Once the metric is defined, the Mandelbrot set:

$\begin{cases} z_0 = 0 \\ z_{n+1} = z_n^2 + c \end{cases},$

The divergence of the succession will be according to the new metric defined. So, according to the new modulo defined, the succession will converge or the will diverge. On a practice way, that it's done is set a threshold value so, once the modulo is greater than the threshold, the number of iterations needed to reach the threshold will define the color of the pixel. This allows to create computer programs in order to graph those sets.

2.1 Invariants

In this part, let's check if the modulo and the fractals are invariant versus rotations. Taking an euclidean metric:

$z_n = x_n + iy_n$

$|z| = L = \sqrt{|x_n|^2 + |y_n|^2}$

Considering a rotation of angle φ, $w = ze^{i\varphi}$, following the same method:

$\begin{cases} z_0 = 0 \\ z_{n+1} = e^{i\varphi}\left(z_n^2 + c\right) \end{cases}$,where c is a complex constant

This set is constructed following those steps:

1) Get a point of the complex plane

2) Iterate $z_{n+1} = e^{i\varphi}\left(z_n^2 + c\right) = e^{i\varphi}z_n^2 + c_\varphi$

3) If the succession diverges, this element doesn't belong to a Mandelbrot set. If the succession converges, the element belongs.

So, the rotation won't change the nature of divergence or convergence. Now, the question is if the number of iterations will be the same after a rotation.

$|z_{n+1}| = \left|z_n^2 + c_n\right| \le \left|z_n^2\right| + |c_n| < L$

$|w_{n+1}| = \left|e^{i\varphi}z_n^2 + c_n\right| \le \left|e^{i\varphi}z_n^2\right| + |c_n| = \left|z_n^2\right| + |c_n| < L$

The number of iterations is not invariant versus rotations. In order to see this, let's check this table, which summarizes the process:

z_n	w_n
z_o	$w_0 = z_o e^{i\varphi}$
$z_1 = z_o^2 + c$	$w_1 = w_0^2 + c = z_0^2 e^{2i\varphi} + c = z_1 e^{2i\varphi} + c - ce^{-2i\varphi}$
$z_2 = z_1^2 + c$ $= (z_o^2 + c)^2 + c$ $= (z_o^4 + 2z_0 c + c^2) + c$	$w_2 = w_1^2 + c = \left(z_1 e^{2i\varphi} + c - ce^{-2i\varphi}\right)^2 + c$ $= \left(\left(z_0^2 + c\right) e^{2i\varphi} + c - ce^{-2i\varphi}\right)^2 + c$ $= \left(z_0^2 + c\right)^2 e^{4i\varphi} + \left(c - ce^{-2i\varphi}\right)^2 + 2\left(z_0^2 + c\right) e^{2i\varphi}\left(c - ce^{-2i\varphi}\right) + c$ $= (z_o^4 + 2z_0 c + c^2) e^{4i\varphi} + \underbrace{c^2 + c^2 e^{-4i\varphi} - 2c^2 e^{-2i\varphi} + 2(z_0^2 + c)\left(ce^{2i\varphi} - c\right) + c}_{I}$ $= z_2 e^{4i\varphi} - ce^{4i\varphi} + I$

$|w_2| = \left|z_2 e^{4i\varphi} - ce^{4i\varphi} + I\right| \leq \left|z_2 e^{4i\varphi} - ce^{4i\varphi}\right| + |I| \leq \left|z_2 e^{4i\varphi}\right| + \left|ce^{4i\varphi}\right| + |I| = |z_2| + |c| + |I|$

So, after 2 iterations, the modulo won't match. As the threshold L is defined for the complex $|z_2|$, so the threshold is not respected, the thresholds differ by $|c| + |I|$. So the fractals are not invariant versus rotation.

3 Quaternions

This section will show a study about the possibility of using quaternions to generate fractals. Firstly, let's define the quaternion:

$q = a + b\hat{i} + c\hat{j} + d\hat{k}$

The conjugate: $\bar{q} = a - b\hat{i} - c\hat{j} - d\hat{k}$, so the norm n(a) of a quaternion is defined by $n(q) = \sqrt{q\bar{q}} = \sqrt{a^2+b^2+c^2+d^2}$. Defining the metric tensor G as $G = \begin{pmatrix} 1 & & & \\ & -1 & & \\ & & -1 & \\ & & & -1 \end{pmatrix}$, the modulo of a quaternion can be represented as:

$|q|^2 = q^\alpha q_\alpha = q^\alpha G_\alpha^\beta q_\beta = \begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} 1 & & & \\ & -1 & & \\ & & -1 & \\ & & & -1 \end{pmatrix} \begin{pmatrix} a \\ -b \\ -c \\ -d \end{pmatrix} = a^2 + b^2 + c^2 + d^2$, as $q_\beta = G_{\beta\alpha} q^\alpha = \bar{q}$

In a particular case, considering the complex number $z = a + bi$, this can be represented on a tensor way as z^α, so the square modulo will be defined as:

$G = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix}$, $|z|^2 = z\bar{z} = z^\alpha G_\alpha^\beta z_\beta = \begin{pmatrix} a & b \end{pmatrix} \begin{pmatrix} 1 & \\ & -1 \end{pmatrix} \begin{pmatrix} a \\ -b \end{pmatrix} = a^2 + b^2$, as $z_\beta = G_{\beta\alpha} z^\alpha = \bar{z}$

In this case, a fractal set on a quaternion case will be like this:

$\begin{cases} q_0 = 0 \\ q_{n+1} = q_n^2 + q_c \end{cases}$,where q_c is a quaternion constant

This set is constructed following those steps:

1) Get a point of the quaternion space

2) Iterate $q_{n+1} = q_n^2 + q_c$

3) If the succession diverges, this element doesn't belong to a this set. If the succession converges, the element belongs.

Bearing in mind the quaternion $q_n = a_n + b_n\hat{i} + c_n\hat{j} + d_n\hat{k}$, let's calculate q_n^2. In order to do this, take in consideration the table of quaternions multiplication:

x	1	i	j	k
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

so ij=k, ji=-k,jk=i,kj=-i,ki=j,ik=-j

So, $q_n^2 = \left(a_n + b_n\hat{i} + c_n\hat{j} + d_n\hat{k}\right) * \left(a_n + b_n\hat{i} + c_n\hat{j} + d_n\hat{k}\right) = \left(a_n^2 - b_n^2 - c_n^2 - d_n^2\right) + 2a_nb_n\hat{i} + 2a_nc_n\hat{j} + 2a_nd_n\hat{k}$,

Let's check the iteration with several values:

iteration	a_n	b_n	c_n	d_n
0	0,194	0,06291959	0,050000249	0,977704
1	-0,924728011	0,024412801	0,019400097	0,379349152
2	0,710243767	-0,045150402	-0,035879626	-0,701589574
3	0,008892372	-0,064135583	-0,050966561	-0,996599244
4	-0,999841942	-0,001140635	-0,000906427	-0,017724263
5	0,999367637	0,002280909	0,001812568	0,035442924
6	0,997470985	0,004558934	0,003622844	0,070841022
7	0,989896006	0,009094809	0,007227363	0,141323727
8	0,959786757	0,01800583	0,014308675	0,279791586
9	0,84237834	0,034563514	0,027466554	0,537080518
10	0,419196736	0,058231111	0,04627446	0,904849991
11	-0,648559791	0,048820584	0,038796205	0,758620325
12	-0,158763591	-0,063326135	-0,050323318	-0,984021279
13	-0,949634635	0,020107769	0,015979021	0,312453504
14	0,803519096	-0,038190068	-0,030348464	-0,593433337
15	0,291100302	-0,061372898	-0,048771141	-0,953670038
16	-0,830892412	-0,035731338	-0,028394588	-0,555227271
17	0,380021897	0,059377796	0,047185695	0,922668253
18	-0,712652276	0,045129725	0,035863194	0,70126828
19	0,012773205	-0,064323603	-0,051115974	-0,999520872
20	-1,005629188	-0,001643237	-0,00130583	-0,02553417
21	1,010633664	0,003304974	0,002626361	0,051355813
22	1,018725162	0,006680237	0,005308577	0,103803826
23	1,026952915	0,01361065	0,010815962	0,211495139
24	1,009599861	0,027954994	0,022214968	0,434391098
25	0,829321266	0,056446717	0,044856457	0,877122385
26	-0,08676825	0,093624925	0,074400828	1,454832493
27	-2,123309963	-0,016247342	-0,012911259	-0,25246654
28	4,44427517	0,068996286	0,054829211	1,072129438
29	18,59435353	0,61327696	0,487352202	9,529676477
30	254,3216285	22,8069772	18,12399826	354,3963468

As shown on the table, the succession diverges very quickly.

$|q_n^2| = \sqrt{(a_n^2 - b_n^2 - c_n^2 - d_n^2)^2 + 4a_n^2 b_n^2 + 4a_n^2 c_n^2 + 4a_n^2 d_n^2}$

At the same way that on the Mandelbrot set, it need to be checked if the succession converges or diverges.

As a particular case, where $c_n = d_n = 0$, $|q_n^2| = \sqrt{a_n^2 + b_n^2} = |z_n^2|$

Following up the same method as the section 2, let's try to define a non-euclidean metric in 4 dimensions.

Prior to this, let's consider the quaternion $q = \begin{pmatrix} u \\ x \\ y \\ z \end{pmatrix}$ and let's consider the matrix G giving by:

$$G = \begin{pmatrix} 1 & & & \\ & -1 & & \\ & & -1 & \\ & & & -1 \end{pmatrix}$$

In a particular case, considering the complex number $z = a + bi$, this can be represented on a tensor way as z^α, so the square modulo will be defined as:

$G = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix}$, $|z|^2 = z\bar{z} = z^\alpha G_\alpha^\beta z_\beta = \begin{pmatrix} a & b \end{pmatrix} \begin{pmatrix} 1 & \\ & -1 \end{pmatrix} \begin{pmatrix} a \\ -b \end{pmatrix} = a^2 + b^2$, as $z_\beta = G_{\beta\alpha} z^\alpha = \bar{z}$

$$g = \sum_{i,h=1}^{n} g_{ij} dx^i \otimes dx^j$$

$$= \begin{pmatrix} g_{00} & & & \\ & -g_{11} & & \\ & & -g_{22} & \\ & & & -g_{33} \end{pmatrix}$$

As this operator is bounded, the metric will be bounded. Let's check this following the same example as previously:

$$|q|^2 = q^\alpha q_\alpha = q^\alpha G_\alpha^\beta q_\beta = \begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} g_{00} & & & \\ & -g_{11} & & \\ & & -g_{22} & \\ & & & -g_{33} \end{pmatrix} \begin{pmatrix} g_{00}a \\ -g_{11}b \\ -g_{22}c \\ -g_{33}d \end{pmatrix} = g_{00}^2 a^2 + g_{11}^2 b^2 +$$

$$g_{22}^2 c^2 + g_{33}^2 d^2, \text{ as } q_\beta = \bar{q} = G_{\beta\alpha} q^\alpha = \begin{pmatrix} g_{00} & & & \\ & -g_{11} & & \\ & & -g_{22} & \\ & & & -g_{33} \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} g_{00}a \\ -g_{11}b \\ -g_{22}c \\ -g_{33}d \end{pmatrix}$$

Let's define this metric:

$$g = \sum_{i,h=1}^{n} g_{ij} dx^i \otimes dx^j$$

$$= \begin{pmatrix} a & & & \\ & -bxy & & \\ & & -cxz & \\ & & & -dyz \end{pmatrix},$$

Because of the matrix G is symmetric, it can be found a base $\{B_n\}$where the matrix is diagonal.

Firstly, let's calculate the term$|q_n^2|$:

$$|q_n^2| = q^\alpha G_\alpha^\beta q_\beta = q^\alpha G_\alpha^\beta (G_{\beta\alpha} q^\alpha) = \sum_{ij} g_{ij} \frac{dx^i}{dt}\frac{dx_j}{dt} = a^2 \left(\frac{du}{dt}\right)^2 + b^2x^2y^2 \left(\frac{dx}{dt}\right)^2 + c^2x^2z^2 \left(\frac{dy}{dt}\right)^2 + d^2y^2z^2 \left(\frac{dz}{dt}\right)^2$$

$$L = \int_0^{t_n} \sqrt{a^2 \left(\frac{du}{dt}\right)^2 + b^2x^2y^2 \left(\frac{dx}{dt}\right)^2 + c^2x^2z^2 \left(\frac{dy}{dt}\right)^2 + d^2y^2z^2 \left(\frac{dz}{dt}\right)^2} dt$$

Using the parametric equations:

$u = Vt$

$x = Vp_1t$

$y = Vp_2t$

$z = Vp_3t$

considering the quaternion $q_n = a_n + b_n\hat{i} + c_n\hat{j} + d_n\hat{k}$,

$$a^2 \left(\frac{du}{dt}\right)^2 + b^2x^2y^2 \left(\frac{dx}{dt}\right)^2 + c^2x^2z^2 \left(\frac{dy}{dt}\right)^2 + d^2y^2z^2 \left(\frac{dz}{dt}\right)^2 = a^2V^2 + b^2p_1^2p_2^2V^4t^4V^2p_1^2 + c^2V^2p_1^2V^2p_3^2t^4V^2p_2^2 +$$

$$d^2V^4p_2^2p_3^2t^4V^2p_3^2 = a^2V^2 + \left\{b^2p_1^4p_2^2V^6 + c^2p_1^2p_2^2p_3^2V^6 + d^2p_2^2p_3^4V^6\right\} t^4$$

the distance will be:

$$L = \int_0^{t_n} \sqrt{a^2V^2 + \{b^2p_1^4p_2^2V^6 + c^2p_1^2p_2^2p_3^2V^6 + d^2p_2^2p_3^4V^6\} t^4} dt$$

$$L = aV \int_0^{t_n} \sqrt{1 + \frac{b^2p_1^4p_2^2 + c^2p_1^2p_2^2p_3^2 + d^2p_2^2p_3^4}{a^2} V^4t^4} dt$$

Let's call $M^4 = \frac{b^2p_1^4p_2^2 + c^2p_1^2p_2^2p_3^2 + d^2p_2^2p_3^4}{a^2} V^4$.

$$L = aV \int_0^{t_n} \sqrt{1 + M^4t^4} dt$$

This integral doesn't have an antiderivate, so let's try to do a serial development:

$\sqrt{1 + M^4t^4} \approx 1 + \frac{M^4t^4}{2} - \frac{M^8t^8}{8} + \frac{M^{12}t^{12}}{16} + ...$, where $M^2t^4 \le 1$

So,

$$L = aV \int_0^{t_n} \sqrt{1 + M^4t^4} dt \approx aV \int_0^{t_n} \left\{1 + \frac{M^4t^4}{2} - \frac{M^8t^8}{8} + \frac{M^{12}t^{12}}{16} + ...\right\} dt = aV \left\{t_n + \frac{M^4t_n^5}{10} - \frac{M^8t_n^9}{72} + \frac{M^{12}t^{13}}{16*13} + ...\right\}$$

This formula needs to be set in function of (u_n, x_n, y_n, z_n), so

$t = V^{-1}u_n$

$p_1 = \frac{x_n}{u_n}$

$p_2 = \frac{y_n}{u_n}$

$p_3 = \frac{z_n}{u_n}$

So, making those substitutions, the "distance" of a 4d point will be:

$$L \approx a \left\{u_n + \frac{1}{10}\frac{b^2p_1^4p_2^2 + c^2p_1^2p_2^2p_3^2 + d^2p_2^2p_3^4}{a^2} u_n^5 - \frac{1}{72}\left(\frac{b^2p_1^4p_2^2 + c^2p_1^2p_2^2p_3^2 + d^2p_2^2p_3^4}{a^2}\right)^2 u_n^9 + \frac{1}{16*13}\left(\frac{b^2p_1^4p_2^2 + c^2p_1^2p_2^2p_3^2 + d^2p_2^2p_3^4}{a^2}\right)^3 u_n^{13} + ...\right\}$$

$$L \approx a \left\{u_n + \frac{1}{10}\frac{b^2x_n^4y_n^2 + c^2x_n^2y_n^2z_n^2 + d^2y_n^2z_n^4}{a^2} u_n^{-1} - \frac{1}{72}\left(\frac{b^2x_n^4y_n^2 + c^2x_n^2y_n^2z_n^2 + d^2y_n^2z_n^4}{a^2}\right)^2 u_n^{-3}\right\}$$

$$+a \left\{\frac{1}{16*13}\left(\frac{b^2x_n^4y_n^2 + c^2x_n^2y_n^2z_n^2 + d^2x_n^2y_n^4}{a^2}\right)^3 u_n^{-5} + ...\right\}$$

As the succession diverges very quickly, could it be possible to calculate the modulo of the succession in order to create a fractal? Is there a practical way to create those fractals? As the metrics depends on the coordinates, the same criteria cannot be applied on a general case.

The fractal set on a quaternion case will be like this:

$$\begin{cases} q_0 = 0 \\ q_{n+1} = q_n^2 + q_c \end{cases} \text{,where } q_c \text{ is a quaternion constant}$$

$q_n^k(0) = q_n\left(\underset{k\,times}{q_n\,(...)}\right)$

In the last example, it shown that, in a general case, the succession diverges very quickly, so there is no way to calculate the modulo of the succession. Let's see what are the conditions where the norm can converge or diverge according to the point.

Considering the matrix g as an operator, the succession will be bounded if the operator is bounded. Let's see on the previous examples, if the operators are bounded or not.

Ex 1: $g = \sum_{i,h=1}^{n} g_{ij}dx^i \otimes dx^j$

$$= \begin{pmatrix} a & & & \\ & -b & & \\ & & -c & \\ & & & -d \end{pmatrix}$$

An operator is bounded if, and only if, $\|Lv\|_{\psi} \leq M\|v\|_{\chi}$. A trace class operator is a compact operator for which a trace may be defined, such that the trace is finite and independent of the choice of basis.

$\|g\| = Tr\,|g| = a - b - c - d$

So, as the trace is absolutely convergent, the operator g is bounded and, hence, the succession of the several point can be calculated to create a fractal.

Ex 2: $g = \sum_{i,h=1}^{n} g_{ij}dx^i \otimes dx^j$

$$= \begin{pmatrix} a & & & \\ & -bxy & & \\ & & -cxz & \\ & & & -dyz \end{pmatrix}$$

The trace$\|g\| = Tr\,|g| = a - bxy - cxz - dyz$, is not bounded, so cannot be calculate a fractal because the succession will diverge in every point.

Having seen this, a condition in order to generate a fractal is selecting bounded operators.

Ex 3:$g = \sum_{i,h=1}^{n} g_{ij}dx^i \otimes dx^j$

$$= \begin{pmatrix} a & & & \\ & -bcos(x) & & \\ & & -ccos(y) & \\ & & & -dcos(z) \end{pmatrix}$$

This metric is bounded, $\|g\| = Tr\,|g| = a - bcos(x) - ccos(y) - dcos(z)$, $a - b - c - d \leq \|g\| \leq a + b + c + d$. Also, let's define a positive metric, so a > b+c+d. Following the same procedure:

$\left|q_n^2\right| = q^{\alpha}G_{\alpha}^{\beta}q_{\beta} = \sum_{ij} g_{ij}\frac{dx^i}{dt}\frac{dx_j}{dt} = a^2\left(\frac{du}{dt}\right)^2 + b^2cos^2(x)\left(\frac{dx}{dt}\right)^2 + c^2cos^2(y)\left(\frac{dy}{dt}\right)^2 + d^2cos^2(z)\left(\frac{dz}{dt}\right)^2$

$L = \int_0^{t_n}\sqrt{a^2\left(\frac{du}{dt}\right)^2 + b^2cos^2(x)\left(\frac{dx}{dt}\right)^2 + c^2cos^2(y)\left(\frac{dy}{dt}\right)^2 + d^2cos^2(z)\left(\frac{dz}{dt}\right)^2}dt$

Using the parametric equations:

$u = Vt$

$x = Vp_1t$

$y = Vp_2t$

$z = Vp_3t$

considering the quaternion $q_n = a_n + b_n\hat{i} + c_n\hat{j} + d_n\hat{k}$,

$L = \int_0^{t_n}\sqrt{a^2V^2 + b^2cos^2(Vp_1t)V^2p_1^2 + c^2cos^2(Vp_2t)V^2p_2^2 + d^2cos^2(Vp_3t)V^2p_3^2}dt$

In fact, we're interested to know if the length has a finite value. In this case, it's known that $m(b-a) \leq \int_a^b f(x) \leq M(b-a)$ where m = min f(x) and M = max f(x).

$L = V\int_0^{t_n}\sqrt{a + b^2cos^2(Vp_1t)p_1^2 + c^2cos^2(Vp_2t)p_2^2 + d^2cos^2(Vp_3t)p_3^2}dt$

Let's call $f(x) = \sqrt{a^2 + b^2cos^2(Vp_1t)p_1^2 + c^2cos^2(Vp_2t)p_2^2 + d^2cos^2(Vp_3t)p_3^2}$, so, in order to check if this function is bounded.

$f_E = f^2 = a^2 + b^2cos^2(Vp_1t)p_1^2 + c^2cos^2(Vp_2t)p_2^2 + d^2cos^2(Vp_3t)p_3^2$

$f'_E = -Vb^2p_1^3sin(2Vp_1t) - Vc^2p_2^3sin(2Vp_2t) - Vd^2p_3^3sin(2Vp_3t)$

$f_E" = -2V^2b^2p_1^4cos(2Vp_1t) - 2V^2c^2p_2^4cos(2Vp_2t) - 2V^2d^2p_3^4cos(2Vp_3t)$

t=0 -> f'_E=0 and $f"_E$<0, so at t=0 is the maximum of the function, $f_{max} = \sqrt{a^2 + b^2p_1^2 + c^2p_2^2 + d^2p_3^2}$, so the function has a maximum (it's bounded).

4 Conclusions

This paper proposes a method about create fractals uses non-euclidean metric, which allows to extend the possibilities of study of fractals. It was shown how to build a distance on a Riemann geometry in order define the "modulo" of a complex number. Then, it was extended the study to the quaternions, where it was shown some conditions where the fractals are strictly divergent, so it cannot be included. I hope that this paper form the basis for extending the study for those objects. At the end of this paper you can find a Java program, which allows to test several formulas in order to see the images generated. Feel free to send me any comments to ycachon@gmail.com about this paper.

5 Java program

This section shows an example of Java which allows to show fractals using whatever modulo metric you want to set:

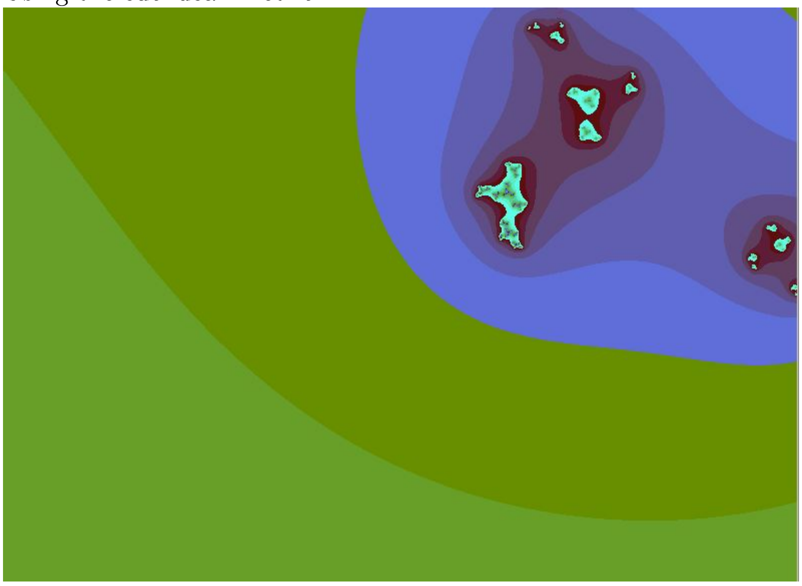
```
package fractal2d;
import java.math.*;
import java.awt.Graphics;
import java.awt.image.BufferedImage;
import javax.swing.JFrame;
public class Fractal2d extends JFrame{
private final int MAX_ITER = 80;
private final double ZOOM = 3000;
private BufferedImage I;
private double zx, zy, cX, cY, new_zx;
private int cr,cg,cb; // Color RGB
private double r,theta,val;
final private double centerX = -1000;
final private double centerY = 50;
double modulo (double zx,double zy)
{
// Here you set the modulo you want to define
return zx * zx + zy * zy;
}
public fractal_Set()
{
setBounds(100, 100, 800, 600);
setResizable(false);
setDefaultCloseOperation(EXIT_ON_CLOSE);
I = new BufferedImage(getWidth(), getHeight(), BufferedImage.TYPE_INT_RGB);
val = 2;
for (int y = 1; y < getHeight(); y++) {
for (int x = 1; x < getWidth(); x++) {
zx = zy = 0;
cX = (x - centerX) / ZOOM;
cY = (y - centerY) / ZOOM;
int iter = MAX_ITER;
while (modulo(zx,zy) < 100 && iter > 0)
{
new_zx = zx * zx - zy * zy + cX;
zy = 2.0 * zx * zy + cY;
zx = new_zx;
iter–;
}
cr=iter;
cg=iter*20;
cb=iter*40;
I.setRGB(x, y, (cr<<16) | (cg<<8) | cb); }
}
}
public void paint(Graphics g)
```

```
{
g.drawImage(I, 0, 0, this);
}
public static void main(String[] args) {
new Fractal2d().setVisible(true);
}
}
```

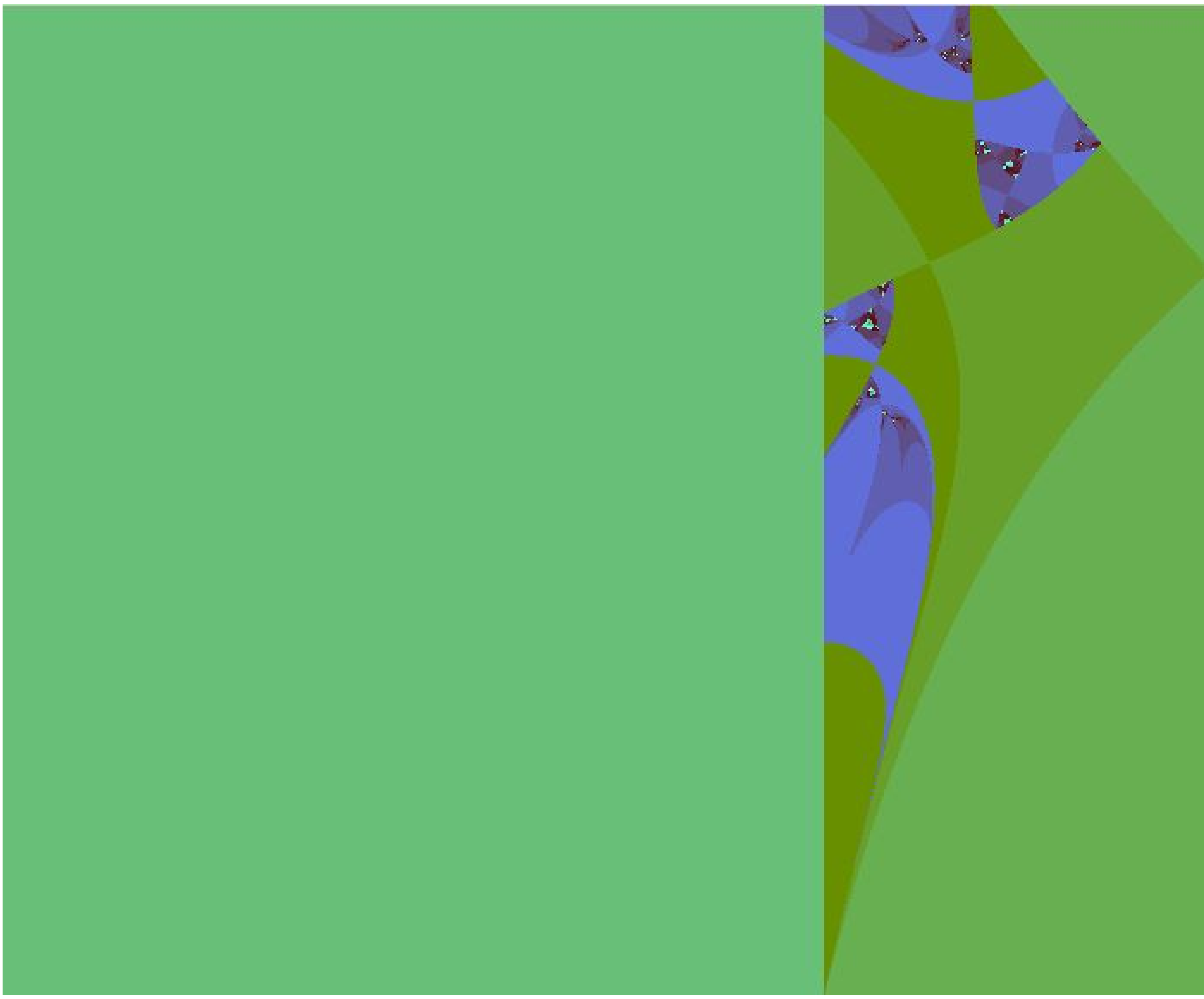
Here there is an example as (center 480,600), zoom 400 iter 100 using the modulo
$L = \frac{y_m}{x_n}\sqrt{\frac{y_n}{cx_n}}\left\{\frac{1}{2}\frac{x_n}{y_n}\sqrt{c^2x_n^4 + cx_ny_n} + \frac{1}{2}ln\left(x_n\sqrt{\frac{cx_n}{y_n}} + \sqrt{1 + \frac{cx_n^3}{y_n}}\right)\right\}.$
On the previous program, it should be use this function:

```
double modulo (double zx,double zy,double c)
{
return zy/zx*Math.sqrt(zy/zx/c)*(0.5*zx/zy*Math.sqrt(c*c*Math.pow(zx, 4)+c*zx*zy) +
0.5*Math.log(zx*Math.sqrt(c*zx/zy)+Math.sqrt(1+c*Math.pow(zx, 3)/zy)));
}
```

Using the euclidean metric:



Using the non-euclidean metric:



References

[https://en.wikipedia.org/wiki/Mandelbrot_set]

[https://en.wikipedia.org/wiki/Misiurewicz_point]

[https://en.wikipedia.org/wiki/Self-similarity]

[https://en.wikipedia.org/wiki/Compact_operator_on_Hilbert_space]

[1] https://en.wikipedia.org/wiki/Bounded_operator

[2] https://en.wikipedia.org/wiki/Trace_class