An Algorithm Since the Vector Approach in Mathematica to Generate the Von Koch Curve and Islands in a General Way

Segundo B. Correa-Erazo[1], Ricardo Velezmoro-León[1] and Robert Ipanaqué-Chero[1]

[1]*Universidad Nacional de Piura, Perú*

Abstract

In this paper, the Von Koch curve is generated in a general way from a vector approach without using the classic system of iterated functions. This approach also allows constructing different Koch islands based on any regular polygon; to do this, a Koch curve is drawn on each of the sides of the regular polygon, taking into account that if the vertices of the polygon are traversed counterclockwise, An island will be generated inside the polygon, but if the vertices are traversed clockwise, an island outside the polygon will be developed. Given the versatility of the method, an interior island and an exterior island are also united, giving origin to a new fractal. To visualize the results, Mathematica scientific software was used.

Keywords

Von Koch curve, Von Koch Island, algorithm, new focus

1. Introduction

The system of iterated functions that generates the classical Von Koch curve is given by:

$$
\begin{cases}
S_1(x,y) = (\frac{1}{3}x, \frac{1}{3}y) \\[2mm]
S_2(x,y) = (\frac{1}{6}x - \frac{\sqrt{3}}{6}y + \frac{1}{3}, \frac{\sqrt{3}}{6}x + \frac{1}{6}y) \\[2mm]
S_3(x,y) = (\frac{1}{6}x + \frac{\sqrt{3}}{6}y + \frac{1}{2}, -\frac{\sqrt{3}}{6}x + \frac{1}{6}y + \frac{\sqrt{3}}{6}) \\[2mm]
S_4(x,y) = (\frac{1}{3}x + \frac{2}{3}, \frac{1}{3}y)
\end{cases}
$$

It has the limitation of using the segment whose ends are $\{(0,0), (1,0)\}$ and, when applied to any pair of points in the plane, converges towards the classical Koch curve [1]. This is due to the action of Banach's fixed point theorem, since the operator $S(A) = \bigcup\limits_{i=1}^{4} S_i(A)$ is a

(a) Start.

(b) First iteration.

(c) second iteration.
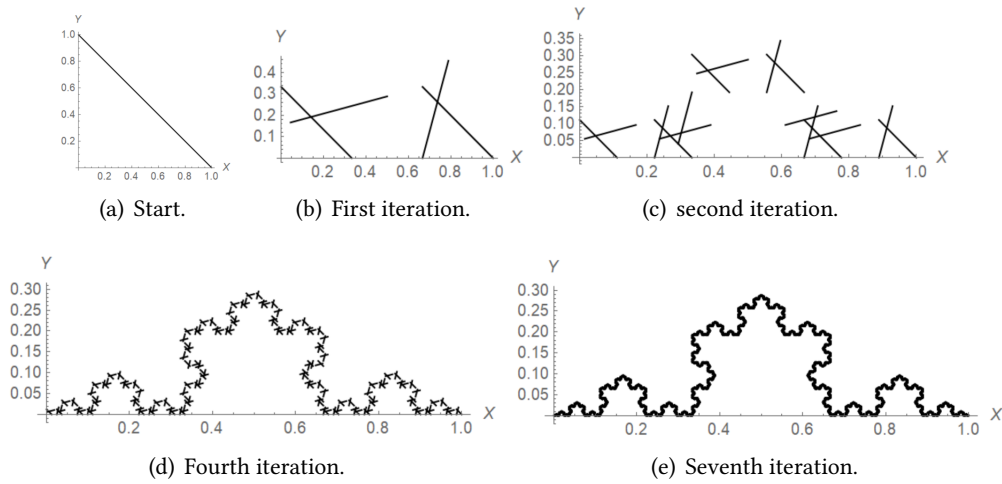
(d) Fourth iteration.

(e) Seventh iteration.

Figure 1: Convergence towards the operator's fixed point.

contractive operator with the Hausdorff metric and has a single fixed point [2].
This means that if we take any arbitrary ends segment of the plane and apply the above IFS to it, is obtained as a final result the classic Von Koch curve as can be seen in the figure 1.

As described above, numerically, it can be observed that effectively when evaluating the transformations S_1, S_2, S_3 and S_4 starting at any pair of points in the plane, the values converge to the points of the classical Von Koch curve defined in the interval $[0, 1]$. In table 1 it can be seen that if we recursively evaluate the transformations S_1 and S_4 at the point $(2, 3)$ using ten iterations, the first transformation converges to $(0, 0)$, while S_4 does so towards $(1, 0)$ which are the extremes of the classical Von Koch curve.

The big question is how to generate the Von Koch curve by taking a segment with arbitrary ends $\{P, Q\}$ of the plane. We want the Koch curve obtained to be located on the segment $\overline{PQ}$. The interest in searching for different variants of the Von-Koch curve is because there are exciting applications in the field of communications because, due to its shape, multiple resonances can be obtained when the angle is modified from 60^0 to a different one [3].
In this article, an algorithm is built to generate the Von Koch curve generally based on three operations, obtaining the five necessary points to develop the first iteration. This algorithm has the power to draw the Koch curve on any segment of the plane that is taken; in addition, it allows the generation of Koch islands based on any regular polygon, and the versatility of the algorithm offers the possibility of generating an island inside the polygon and another outside the polygon depending on whether the vertices are taken in a counterclockwise and clockwise direction respectively.

Table 1

Convergence of the evaluation of the transformations S_1 and S_4 of the classical Con Koch curve.

Iteration	S_1	S_4
Star	$(2, 3)$	$(2, 3)$
1	$(0 \cdot 666667, 1.)$	$(1 \cdot 66667, 0 \cdot 666667)$
2	$(0 \cdot 222222, 0 \cdot 333333)$	$(1 \cdot 22222, 0 \cdot 222222)$
3	$(0 \cdot 0740741, 0 \cdot 111111)$	$(1 \cdot 07407, 0 \cdot 0740741)$
4	$(0 \cdot 0246914, 0 \cdot 037037)$	$(1 \cdot 02469, 0 \cdot 0246914)$
5	$(0 \cdot 00823045, 0 \cdot 0123457)$	$(1 \cdot 00823, 0 \cdot 00823045)$
6	$(0 \cdot 00274348, 0 \cdot 00411523)$	$(1 \cdot 00274, 0 \cdot 00274348)$
7	$(0 \cdot 000914495, 0 \cdot 00137174)$	$(1 \cdot 00091, 0 \cdot 000914495)$
8	$(0 \cdot 000304832, 0 \cdot 000457247)$	$(1 \cdot 0003, 0 \cdot 000304832)$
9	$(0 \cdot 000101611, 0 \cdot 000152416)$	$(1 \cdot 0001, 0 \cdot 000101611)$
10	$(0 \cdot 0000338702, 0 \cdot 0000508053)$	$(1 \cdot 00003, 0 \cdot 0000338702)$

Note. S_1 converges toward the origin, while S_4 converges toward $(1, 0)$.

2. Mathematical and computational foundations

2.1. About the Von Koch curve classical

The Von Koch curve classical is obtained from the following description algorithm [4]:

- `step 1`: In the first iteration, we divide the unit segment into three equal parts;
- `Step 2`: Above the central part, we build an equilateral triangle;
- `Step 3`: Now, we eliminate the base of the triangle, thus obtaining four segments of length $\frac{1}{3}$; that is, the length of the curve is now $\frac{4}{3}$;
- `Step 4`: For the second iteration, we construct equilateral triangles in each of the four segments and apply step 3.

From this algorithm, the four transformations are obtained using the scheme in figure 2:
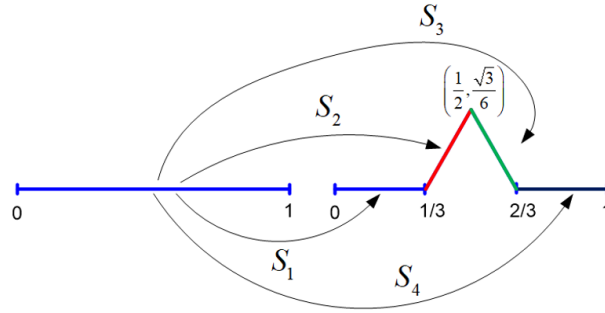
Figure 2: Transformations that generate the Von Koch curve.

The first transformation contracts the segment of length one to its third part; that is to say, $S_1(x, y) = \left(\frac{1}{3}x, \frac{1}{3}y\right)$. To obtain the second transformation rotate S_1 an angle of 60^0 and we move it to the point $\left(\frac{1}{3}, 0\right)$. Therefore:

$$S_2(x, y) = \begin{pmatrix} \cos(60^0) & -\sin(60^0) \\ \sin(60^0) & \cos(60^0) \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{3}x \\ \frac{1}{3}y \end{pmatrix} + \begin{pmatrix} \frac{1}{3} \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{6}x - \frac{\sqrt{3}}{6}y + \frac{1}{3} \\ \frac{\sqrt{3}}{6}x + \frac{1}{6}y \end{pmatrix}$$

Similarly, to determine the transformation S_3 we must rotate S_1 an angle of -60^0 and translate it to the point $\left(\frac{1}{2}, \frac{\sqrt{3}}{6}\right)$. Therefore:

$$S_3(x, y) = \begin{pmatrix} \cos(-60^0) & -\sin(-60^0) \\ \sin(-60^0) & \cos(-60^0) \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{3}x \\ \frac{1}{3}y \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{6} \end{pmatrix} = \begin{pmatrix} \frac{1}{6}x + \frac{\sqrt{3}}{6}y + \frac{1}{2} \\ -\frac{\sqrt{3}}{6}x + \frac{1}{6}y + \frac{\sqrt{3}}{6} \end{pmatrix}$$

Finally the transformation S_4 is a translation from S_1 to the point $\left(\frac{2}{3}, 0\right)$; that is to say $S_4(x, y) = \left(\frac{1}{3}x + \frac{2}{3}, \frac{1}{3}y\right)$.

Among the peculiarities of this curve is that it has infinite length because if we calculate its length in stages, it has that in the first stage, its length will be $L_1 = 4(\frac{1}{3})$, in the second stage is have $L_2 = 16(\frac{1}{9})$; and in the $n-$th iteration it result $L_n = \left(\frac{4}{3}\right)^n$. Therefore, the length of the Von Koch curve will be:

$$\lim_{n \to \infty} \left(\frac{4}{3}\right)^n = \infty$$

On the other hand, it is a continuous curve everywhere but differentiable in none of them [5] since it has a corner at each of its points and can be used as an example in the classroom of the differential calculus course to explain the characteristics of a curve that is not differentiable at any of its points. When a cognitive conflict is created when explaining differentiability, a test is offered in [6]. Its dimension is calculated using the formula:
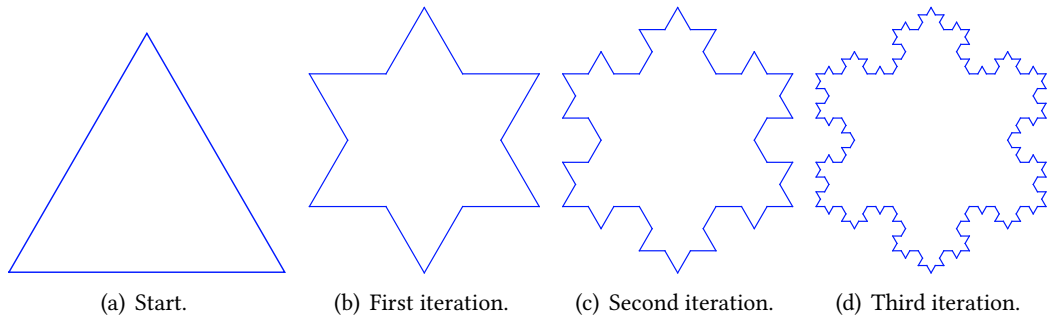
$$\dim = \frac{\log n}{\log\left(\frac{1}{s}\right)}$$

(a) Start. (b) First iteration. (c) Second iteration. (d) Third iteration.

Figure 3: Construction of the Koch snowflake.

Where n is the number of boxes with which the curve is covered and s is the scale that, in this case, will be $\frac{1}{3}$. Therefore the dimension of the Von Koch curve will be $\dim = \frac{\log 4}{\log\left(\frac{1}{\frac{1}{3}}\right)} = 1 \cdot 26186$ [3] and can be interpreted as a dimension of capacity; That is, the Koch curve has a capacity of $1 \cdot 26186$ to occupy the space $\mathbb{R}^2$.

One of the variants offered by the Koch curve is that a significant mathematical curve called a snowflake or Koch island can be constructed from it. This curve has acquired a lot of importance because it is being used as an antenna in wireless communications [7] in antennas for satellites [8]. It is constructed by starting with an equilateral triangle, removing the inner third of each side, making another equilateral triangle where the side was removed, and then repeating the process indefinitely [9]. The different stages of the construction are shown in the figure 3.

In the same way, the snowflake can be generated starting with a square and, in general, from any polygon. This range of possibilities leads us to think of an algorithm that is simple to handle, given that if we use a system of iterated functions, twelve transformations would be needed when we start with a triangle. In the case of starting with a square sixteen and generally for a polygon with n sides, $4n$ transformations with $n \geq 3$ are required.

2.2. Standard Mathematica tools for generating the Von Koch curve classical

Mathematica incorporates the **KochCurve** function to generate the Von Koch curve classical. The variants of this function are:

$$\texttt{KochCurve}[n]$$

which gives the line segments representing the $n^{(th)}$-step Koch curve,

$$\texttt{KochCurve}[n, \{\theta_1, \theta_2, \ldots\}]$$

which takes a series of steps of unit length at successive relative angles θ_i, and

$$\texttt{KochCurve}[n, \{\{r_1, \theta_1\}, \{r_2, \theta_2\}, \ldots\}]$$

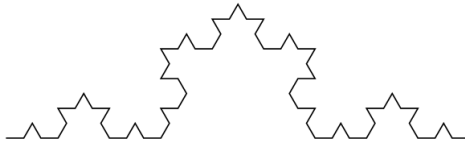which takes successive steps of lengths proportional to r_i.
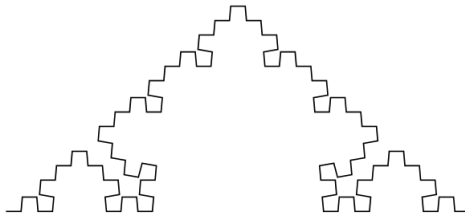
Figure 4: Graph obtained with the first variant.

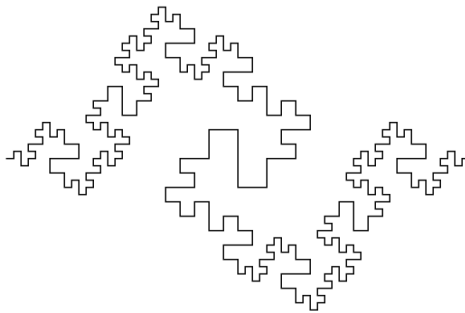**Figure 5:** Graph obtained with the second variant.

Figure 6: Graph obtained with the third variant.

Below are examples for each variant.

```
In[1]:= Graphics[KochCurve[3]]
Out[1]:= See figure 4.

In[2]:= Graphics[KochCurve[3,{0,85 Degree,-85 Degree,
          -85 Degree,85 Degree}]]
Out[2]:= See figure 5.

In[3]:= Graphics[
          KochCurve[3,{{1,0},{1,90 Degree},{1,-90 Degree},
           {2,-90 Degree},{1,90 Degree},{1,90 Degree},
           {1,-90 Degree}}]]
Out[3]:= See figure 6.
```

It is worth mentioning that the standard Mathematica function does not obtain the Von Koch curve in a general way as proposed in this paper.
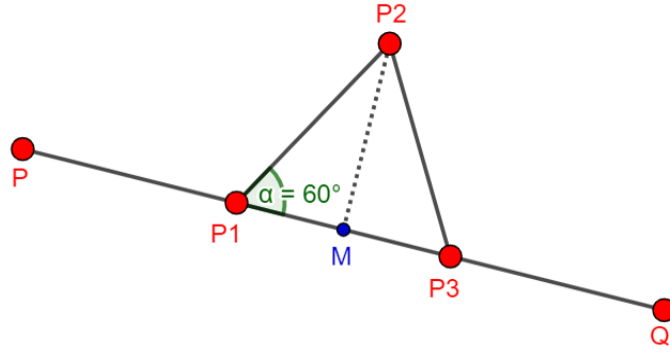
Figure 7: Graphic scheme for the construction of the algorithm.

2.3. Algorithm

It is known that the IFS that generates the Koch curve has four transformations, and if we start with a segment of extremes $P(p_1, p_2)$ and $Q(q_1, q_2)$ when evaluating the IFS in the first iteration will be obtained 8 points; for the second 32 points; for the third 128 and so on, which would require more computational time.

This algorithm consists of three operations for which 5 points are obtained in the first iteration, 20 points for the second, 80 for the third, and so on. The tools that will be used to describe the algorithm are obtained from the following analysis.

Consider the scheme proposed in figure 7. The points P and Q remain fixed, and the only thing we would need are the points P_1, P_2 and P_3. Indeed:

P_1 is obtained through:

$$P_1 = P + \frac{1}{3}(Q - P) \Rightarrow P_1 = \frac{2P + Q}{3}$$

To get to P_2 we use the complex transformation $J : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ defined by $J(p_1, p_2) = (-p_2, p_1)$ [10], given that the segment $\overline{MP_2}$ is perpendicular to the segment $\overline{PQ}$, where M is the midpoint of the segment $\overline{PQ}$. Therefore:

$$P_2 = \frac{P + Q}{2} + \frac{\sqrt{3}}{6}\|Q - P\|\frac{J(Q - P)}{\|J(Q - P)\|} \Rightarrow P_2 = \frac{P + Q}{2} + \frac{\sqrt{3}}{6}J(Q - P)$$

Finally:

$$P_3 = P + \frac{2}{3}(Q - P) \Rightarrow P_3 = \frac{P + 2Q}{3}$$

From this, the algorithm consists of the following steps:

- `step 1`: To consider two arbitrarily different points of the plane;
- `Step 2`: To calculate the points $\{P_1, P_2, P_3\}$ and add the points P y Q, obtaining the list $\{P, P_1, P_2, P_3, Q\}$, which corresponds to the first iteration;

- **Step 3:** Partition the previous list two by two, leaving:

$$\{\{P, P_1\}, \{P_1, P_2\}, \{P_2, P_3\}, \{P_3, Q\}\}$$

And to apply step 2 to each of the partitions;
- **Step 4:** Re-partition the resulting list and apply step 2 again, as often as the desired iteration indicates;
- **Step 5:** Draw the traverse from the points obtained in the required iteration; this traverse will be the Von Koch curve at the desired stage.

2.4. About code in Mathematica

The coding of the above algorithm in the scientific software Mathematica is based on simple commands offered by the software [11], which are detailed below:

The Flatten command removes the grouping signs corresponding to the braces from any list. For example:

```
In[1]:= Flatten[{{2,4,6},{2,1},{8}}]
Out[1]:= {2,4,6,2,1,8}
```

The Partition command partitions a list into a set of lists according to the user's convenience. It can be two by two, three by three, etc. For example:

```
In[2]:= Partition[{2,4,6,2,1,8},2]
Out[2]:= {{2,4},{6,2},{1,8}}
```

If in the previous entry after option 2, we add option 1; it results in a consecutive partition increasing the number of lists. This is:

```
In[3]:= Partition[{2,4,6,2,1,8},2,1]
Out[3]:= {{2,4},{4,6},{6,2},{2,1},{1,8}}
```

Using the previous commands defined the command **IniKoch** as follows:
————————————————————Command1————————————————————————

```
In[4]:= IniKoch[P:{_,_},Q:{_,_}]:=Module[{J,PuntP1,PuntP2,PuntP3};
          J[x1_,x2_]:={-x2,x1};
          PuntP1[{P_,Q_}]:=(2P+Q)/3;
          PuntP2[{P_,Q_}]:=(P+Q)/2+(sqrt3/6)J[Q-P];
          PuntP3[{P_,Q_}]:=(P+2Q)/3;
         {P,PuntP1[{P,Q}],PuntP2[{P,Q}],PuntP3[{P,Q}],Q}]
```
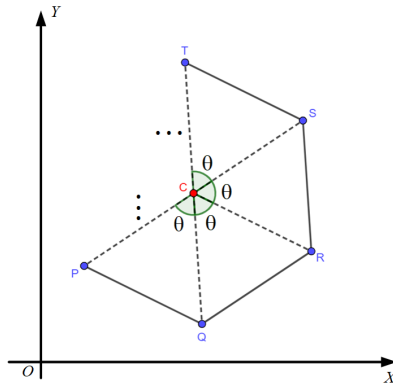
Figure 8: Graphic scheme for the construction of a regular polygon from the center and an arbitrary point.

The command **IniKoch** defined above has the mission of generating the five points $\{P, P_1, P_2, P_3, Q\}$ corresponding to the first iteration of the curve Von Koch. It should be noted that the command input corresponds to any two points of the plane $\{P, Q\}$ and the points $\{P_1, P_2, P_3\}$ are calculated based on as mentioned in the description of the algorithm. Using the **IniKoch** command, the **KochCurveGeneral** command is defined that will generate the Von Koch curve in a general way as follows:

—————————————————Command2—————————————————

```
In[5]:= CurvaVonKochGeneral[P:{_,_},Q:{_,_},n_,Opts___]:=Module[{},
        vk[1]:={P,Q};
        vk[2]:=N[IniKoch[vk[1],6];
        vk[3]:=IniKoch/@Partition[vk[2],2,1];
        vk[i_]:=IniKoch/@Partition[Partition[Flatten[vk[i-1]],2],2,1];
        Graphics[{Hue[color],Line[vk[n]]},Opts]
```

To increase the variants of the Koch curve, we can think about applying the previous commands to the sides of a regular polygon; however, determining the vertices of a regular polygon without computational help is very laborious. This difficulty is resolved by the command in Mathematica **PoliRG2P** detailed below. Let's think of a point C on the plane as the center of the polygon; then, from any other point P on the plane, we can construct any regular polygon using a rotation matrix as indicated in the figure 8. From this idea, the command that will generate the vertices of the regular polygon is:

—————————————————Command3—————————————————

```
In[6]:= PoliRG2P[Cen:{_,_},P:{_,},nlados_]:=Module[{a,pun,Rotxy},
        Rotxy[t_]:={{Cos[t],-Sin[t]},{Sin[t],Cos[t]}};
        a=P-Cen;
        Tras[R_]:=R+Cen;
        pun=Table[Rotxy[t].a,{t,0,2Pi-2Pi/nlados,2Pi/nlados}];
        Tras/@pun]
```
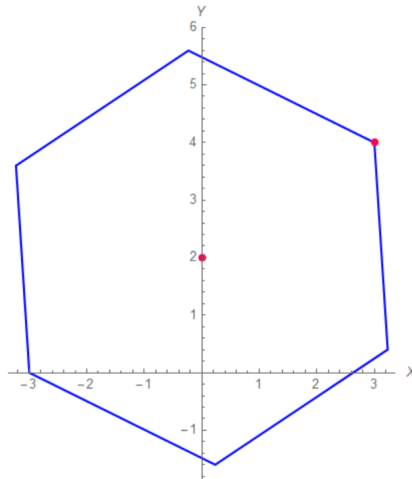
Figure 9: Regular hexagon formed from the center $C(0, 2)$ and a vertex $P(3, 4)$.

If we execute the previous command for the points $C(0, 2)$ and $P(3, 4)$ for $n = 6$ is obtained:

```
Out[6]:= {{3,4},{3/2-sqrt[3],3/2(2+sqrt[3])},
         {-3/2-sqrt[3],1+3sqrt[3]/2},{-3,0},{-3/2+sqrt[3],
         1-3sqrt[3]/2},{3/2+sqrt[3],3-3sqrt[3]/2}}
```

the which are the vertices of a regular hexagon, see figure 9.

By combining commands 2 and 3, we can create a new command **IslaKochGeneralInt** that will draw the Koch curve internally on each side of the regular polygon. The command is defined as follows:

──────────────────────────────Command4──────────────────────────────

```
Out[7]:= IslaKochGeneralInt[Cen:{_,_},P:{_, _},nlados_,iter_,
         Opts___]:= Module[{vv,ll},
         vv=PoliRG2P[Cen,P,nlados];
         ll=Append[vv,vv[[1]]];
         Show[Table[CurvaVonKochGeneral[ll[[i]], ll[[i + 1]],
         iter, 0.65], i, 1,Length[ll] - 1], Axes -> True,
         AxesLabel -> {X,Y}] ]
```

To draw the Koch curve externally on the sides of the polygon, we reverse the list of vertices using the Mathematica command **Reverse**; this will cause the command to traverse the vertices of the polygon clockwise; thus, command 4 is slightly modified by internally adding the option to invert the list, being defined as follows:

```
Out[8]:= IslaKochGeneralExt[Cen:{_,_},P:{_, _},nlados_,iter_,
            Opts___]:= Module[{vv,ll},
            vv=PoliRG2P[Cen,P,nlados];
            vvr= Reverse[vv];
            ll=Append[vvr,vvr[[1]]];
            Show[Table[CurvaVonKochGeneral[ll[[i]], ll[[i + 1]],
            iter, 0.65], i, 1,Length[ll] - 1], Axes -> True,
            AxesLabel -> {X,Y}] ]
```

Another advantage of this method is that the input list of command 2 consists of only two points, namely P and Q. However, it can be generalized so that the input is more than two points so that Koch curves can be graphed in one polygonal. Having said this, the command will be:

```
In[9]:= CurvaVonKochPoligonal[list_,n_,Opts___]:= Module[{Newlist,Pol},
            Newlist=Partition[list,2,1];
            Pol[i_] := CurvaVonKochGeneral[Newlist[[i]], n, color];
            Table[Pol[i], {i, 1, Length[Newlist]}];
            Show[Table[Pol[i], {i, 1, Length[Newlist]}], Opts]  ]
```

3. Results

From the code proposed above, the following results are obtained:

Example 1. for the segment whose ends are $\{(0,1),(2,0)\}$, is obtain the results of figure 10.

Example 2. If we want the rotation to be downwards, we change the order of the points; that is, we take in the order $\{(2,0),(0,1)\}$ and obtain the graphic of the figure 11.

Example 3. For the segment of extremes $\{(3,1),(6,5)\}$, is obtained (see figure 12):
Example 4. If we apply command 4 to the points $C(0,2)$ y $(3,4)$ for an equilateral triangle is obtain (see figure 13):
Example 5. If in the previous example we apply command 5 we obtain the island in the figure 14.
Example 6. If we apply commands 5 and 6 for the same center and vertex and then join them, we obtain a fascinating island, the same one formed by drawing the internal and external Koch curves on the sides of a polygon. For $C(2,3)$ y $P(4,1)$, is obtained (see figure 15):
Ejemplo 7. For the comannd 6, when entering points $\{\{0,0\},\{2,2\},\{4,2\},\{5,0\}\}$ is obtained the images of the figure 16:
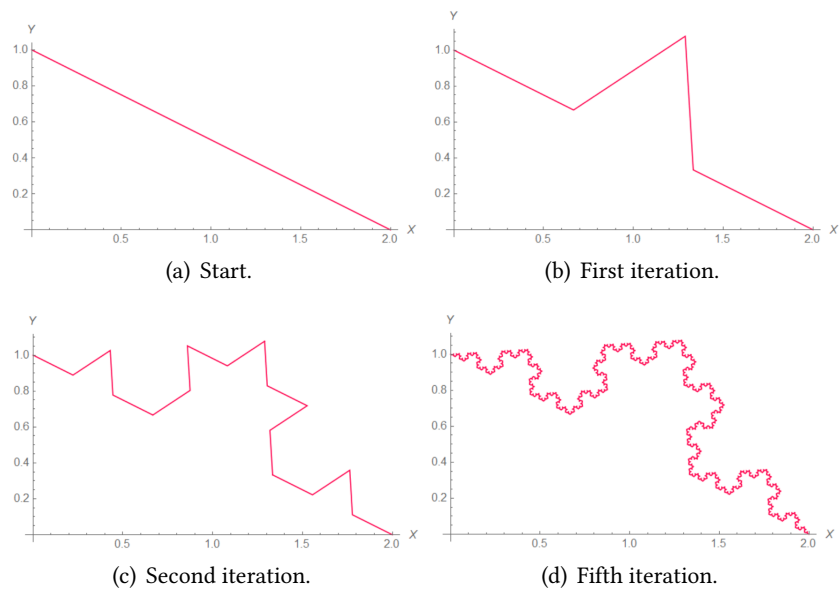
(a) Start.

(b) First iteration.

(c) Second iteration.

(d) Fifth iteration.

Figure 10: Von Koch curve with initial segment of points $P(0,1)$, $Q(2,0)$.

(a) First iteration.

(b) Second iteration.
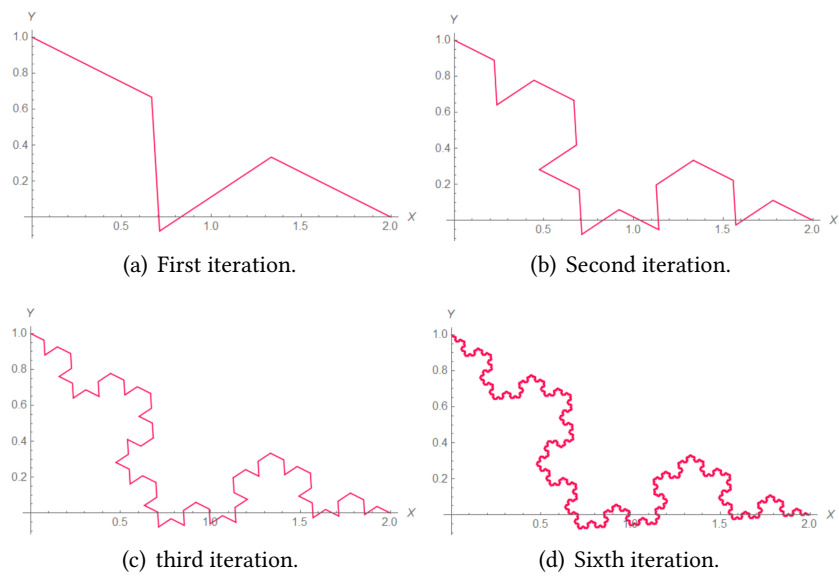

(c) third iteration.

(d) Sixth iteration.

Figure 11: Von Koch curve with initial segment of points $P(2,0)$, $Q(0,1)$.

Ejemplo 8. If we apply command 6 to the list of points $\{\{0,0\}, \{2,2\}, \{4,0\}, \{0,0\}\}$, That is, to the vertices of a triangle traveled counterclockwise, is obtained the images in the figure 17:

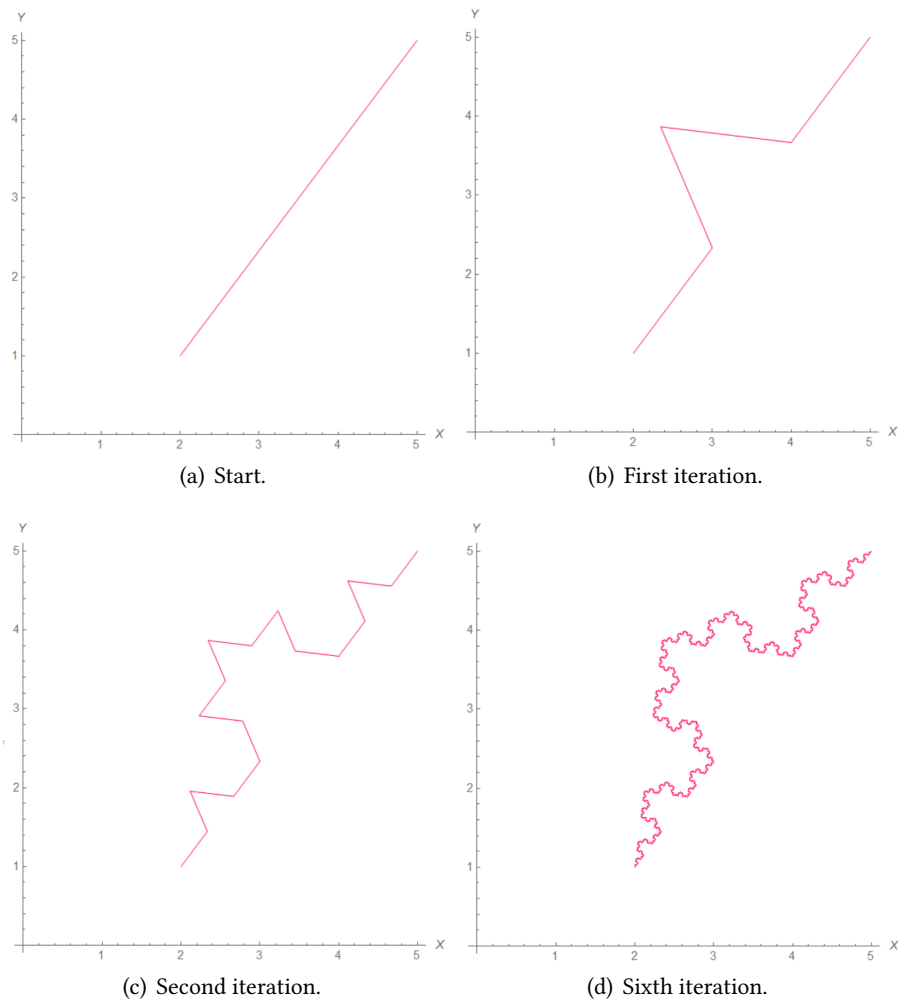(a) Start.

(b) First iteration.

(c) Second iteration.

(d) Sixth iteration.

Figure 12: Von Koch curve with initial segment of points $P(2, 1)$, $Q(5, 5)$.

4. Conclusions

This article shows an algorithm described based on simple computational operations extracted from a vector point of view. The importance lies in the versatility of the algorithm since it is possible to construct a Koch curve simply by taking any initial segment without the need to build a system of iterated functions for each case. It was also shown that from this vector approach, it is possible to construct two Koch islands, one interior, and one exterior, from any regular polygon and generate a new fractal as the union of the two islands inherent to each polygon. In addition to this, a command was also implemented with the power to graph a Koch curve in each of the segments of a polygonal, giving rise to a new generalized Koch curve; it is recommended to computationally implement the approach proposed in this article for when the points belong to a sphere given that the world in which we live has a spherical shape.
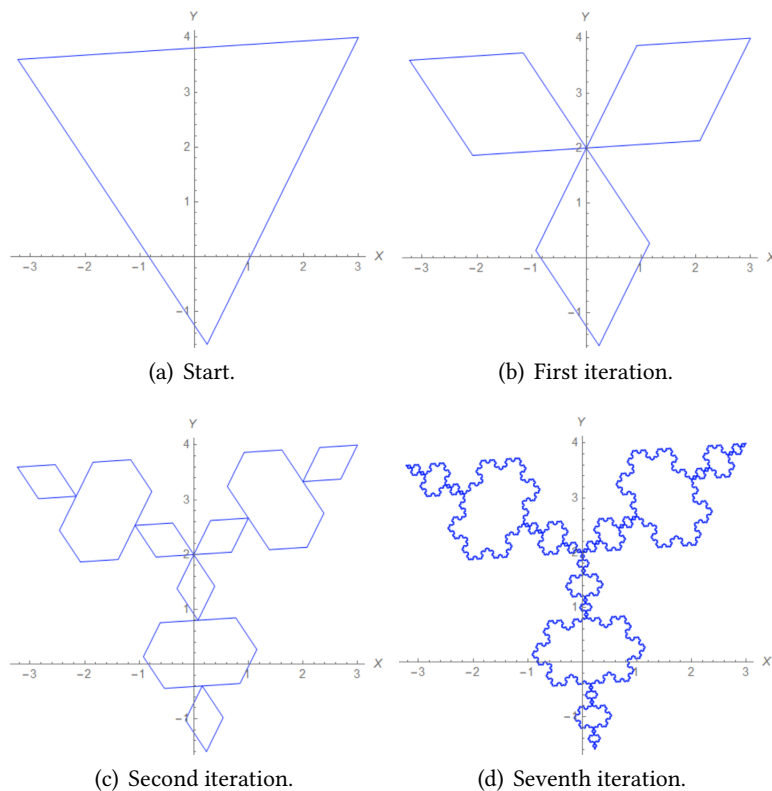
(a) Start.

(b) First iteration.

(c) Second iteration.

(d) Seventh iteration.

Figure 13: Interior Koch Island with an equilateral triangle with center at $C(0,2)$ and vertex at $P(3,4)$.

Acknowledgments

References

[1] S. Plaza, Fractales y generación computacional de imágenes, Pontificia Universidad Católica del Perú, 2000.

[2] J. E. Hutchinson, Fractals and self similarity, Indiana University Mathematics Journal 30 (1981) 713–747.

[3] K. Vinoy, J. K. Abraham, V. K. Varadan, On the relationship between fractal dimension and the performance of multi-resonant dipole antennas using koch curves, IEEE Transactions on Antennas and Propagation 51 (2003) 2296–2303. doi:10.1109/tap.2003.816352.

[4] S. Pool-Domínguez, Introducción a la geometría fractal y su relación con la naturaleza, Abstraction and Application Magazine 7 (2014).

[5] T. Keleti, E. Paquette, The trouble with von koch curves built from n gons, The American Mathematical Monthly 117 (2010) 124–137. doi:10.4169/000298910x476040.
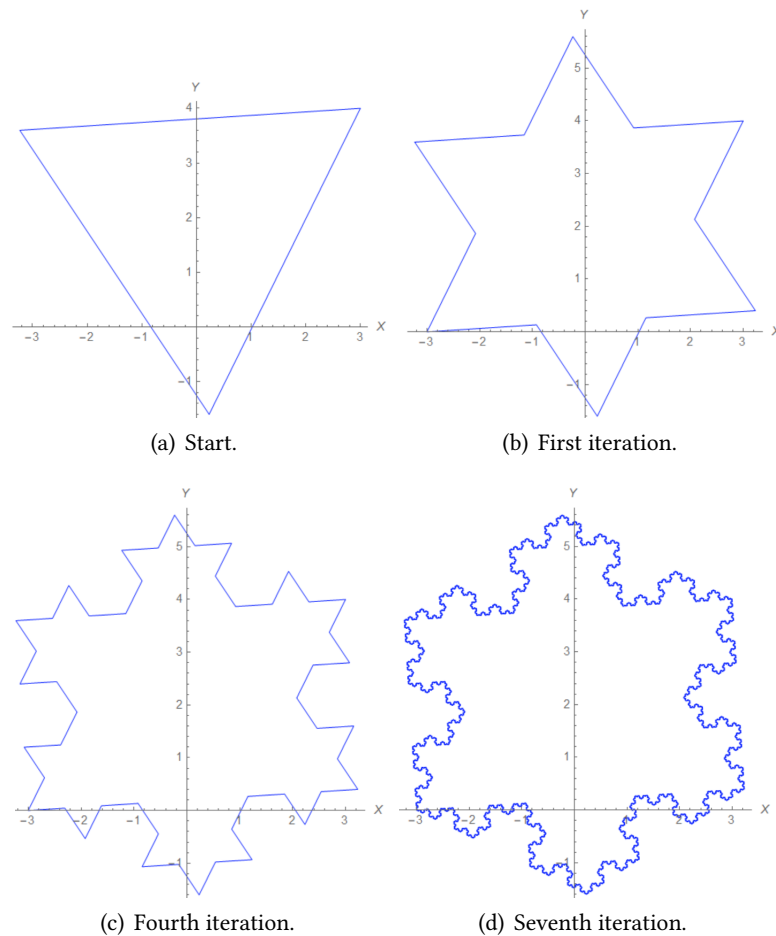
(a) Start.

(b) First iteration.

(c) Fourth iteration.

(d) Seventh iteration.

Figure 14: Exterior Koch Island with an equilateral triangle with center at $C(0,2)$ and vertex at $P(3,4)$.

[6] v. Ungar, The koch curve: A geometric proof, The American Mathematical Monthly 114 (2007) 61–66. URL: http://dx.doi.org/10.1080/00029890.2007.11920392. doi:10.1080/00029890.2007.11920392.

[7] M. Rani, R. U. Haq, D. K. Verma, Variants of koch curve: A review, Int. J. Comput. Appl 2 (2012) 20–24.

[8] O. F. G. Palacios, R. E. D. Vargas, J. A. H. Perez, S. B. C. Erazo, S-band koch snowflake fractal antenna for cubesats, in: 2016 IEEE ANDESCON, IEEE, 2016, pp. 1–4. doi:10.1109/andescon.2016.7836227.

[9] E. W. Weisstein, Koch snowflake, https://mathworld. wolfram. com/ (2008).

[10] E. Abbena, S. Salamon, A. Gray, Modern differential geometry of curves and surfaces with Mathematica, CRC press, 2017.

[11] J. W. Gray, Mastering Mathematica®: Programming Methods and Applications, Academic Press, 2014.

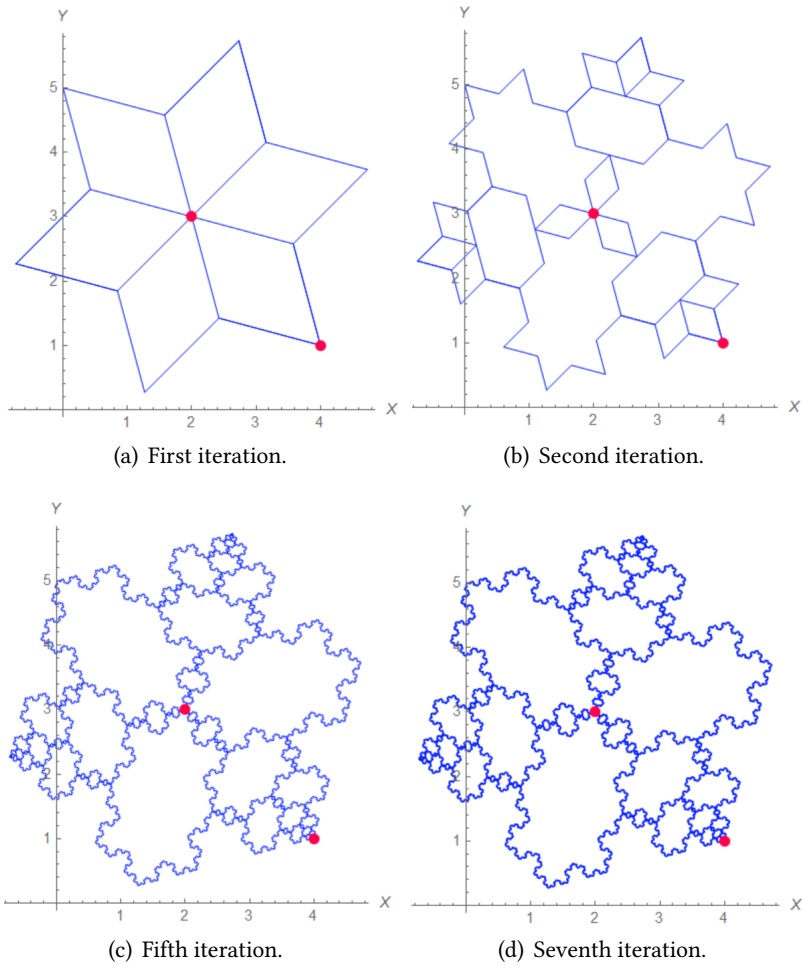(a) First iteration.

(b) Second iteration.

(c) Fifth iteration.

(d) Seventh iteration.

Figure 15: Union of the interior and exterior Koch curve to an equilateral triangle with a center at $C(2,3)$ and a vertex at $P(4,1)$.
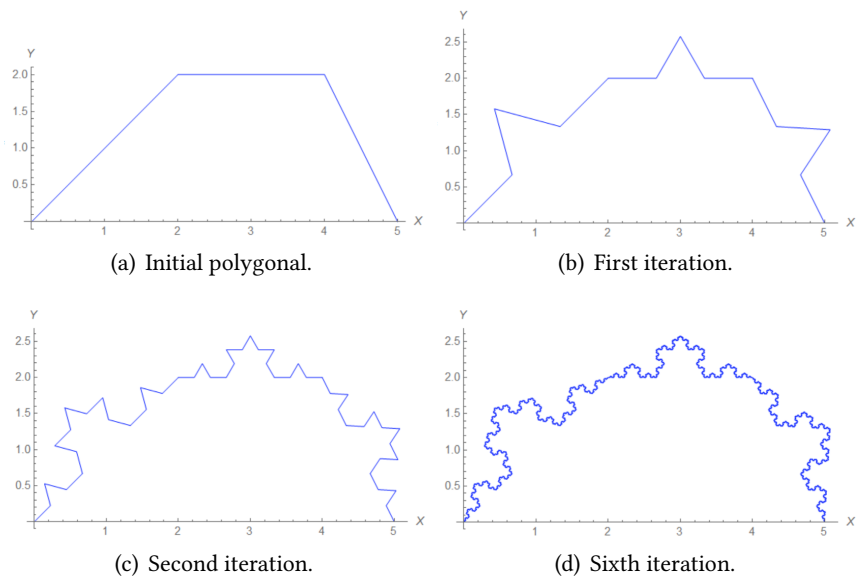
(a) Initial polygonal.

(b) First iteration.

(c) Second iteration.

(d) Sixth iteration.

Figure 16: The Koch curve in a polygonal.

(a) Initial polygonal.

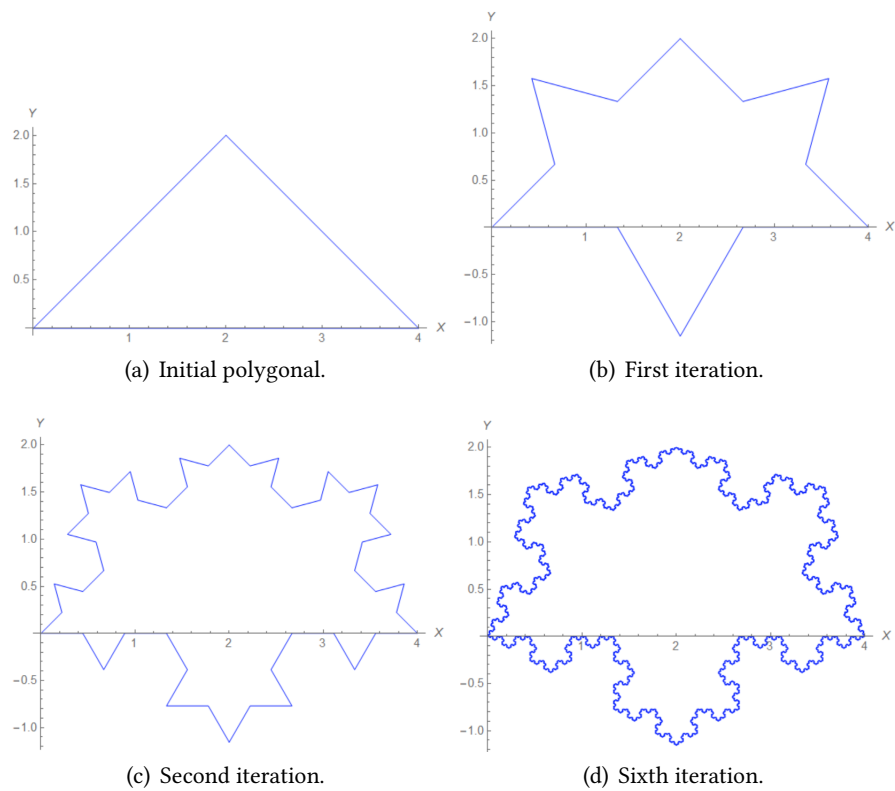(b) First iteration.

(c) Second iteration.

(d) Sixth iteration.

Figure 17: When the polygonal is a triangle, a Koch island is obtained.