L2 - Langage de programmation - Géométrie

Jean-Paul CHEHAB

LAMFA UMR CNRS 7352, Univ. Picardie Jules Verne

Confinement, an 1

Plan

Plan

Plan

Plan

Scilab offre un ensemble de possibilités graphiques pour représenter des objets géométriques :

- En les construisant "à la main", par morceaux, y compris à l'aide d'un algorithme (polygones)
- En représentant des ensembles de points, telles des surfaces (fractals, surfaces)

Ce représentations seront un moyen d'illustrer des résultats mathématiques mais aussi de faire des expériences numériques et se poser des questions.

```
clf ; X=[1 , 2 ,3 ,1] ;
Y= [ 3 , 4 , 8 , 3 ] ;
plot (X,Y) ;
a=gca ( ) ;
a.axes_visible =["off","off","off"] ; // Les axes sont effaces .
a.isoview="on" ; // Le repère sera orthonormé .
```

```
clf ; X=[1 , 2 ,3 ,1] ;
Y= [ 3 , 4 , 8 , 3 ] ;
plot (X,Y) ;
a=gca ( ) ;
a.axes_visible =["off","off","off"] ; // Les axes sont effaces .
a.isoview="on" ; // Le repère sera orthonormé .
```

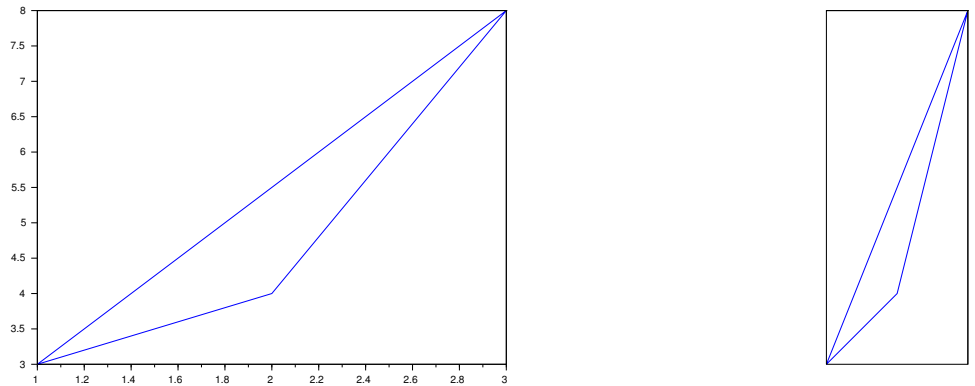


Figure – *Triangle*

CNS de points alignés dans le plan

3 points $(A = (x_1, y_1), B = (x_2, y_2), C = (x_3, y_3)$ ne sont pas alignés (i.e. définissent un triangle non dégénéré) si et seulement si

$$det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix} \neq 0$$

CNS de points alignés dans le plan

3 points $(A = (x_1, y_1), B = (x_2, y_2), C = (x_3, y_3)$ ne sont pas alignés (i.e. définissent un triangle non dégénéré) si et seulement si

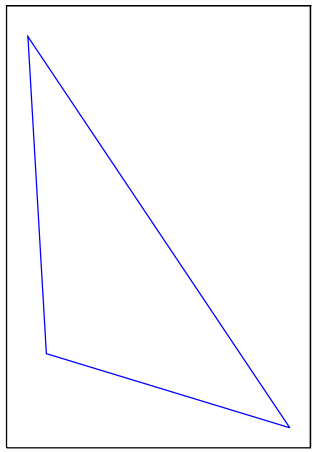
$$det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix} \neq 0$$

Exemple

```
A=[0.9125331,0.0362982];
B=[0.4720888,0.1703793];
C=[ 0.4387004,0.7439883];
det(M)=-0.2481661
les points (A,B,C) ne sont pas alignés
```

Programme

```
A=1-2*rand(2,1);B=1-2*rand(2,1);C=1-2*rand(2,1);un=ones(3,1);
M=[[A(1) B(1) C(1)]; [A(2) B(2) C(2)];un'];
disp("det(M)=",det(M))
if det(M)<>0
disp(["les points (A,B,C) ne sont pas alignés"])
else
disp(["le triangle (A,B,C) est dégénéré"])
end
//dessin du triangle
X=[A(1) B(1) C(1) A(1)];Y=[A(2) B(2) C(2) A(2)];plot (X,Y) ;
a=gca ( ) ; a.axesvisible =["off","off","off"] ; a.isoview="on" ;
```



Coordonnées barycentriques

Theorem

Soient 3 points $A_1 = (a_1, b_1), A_2 = (a_2, b_2), A_3 = (a_3, b_3)$ *non alignés. Pout tout point* $M = (x, y)$ *du plan il existe un unique triplet* $(\lambda_1, \lambda_2, \lambda_3)$ *tel que*

$$M = \sum_{i=1}^{3} \lambda_i A_i \text{ avec } \sum_{i=1}^{3} \lambda_i = 1$$

Les $\lambda_i = \lambda_i(M)$ *sont appelées coordonnées barycentriques de* M*. On a de plus*

- $\lambda_i(Aj) = \delta_{i,j}$
- M *est un point du triangle* (A_1, A_2, Q_3) *si et seulement si* $0 \leq \lambda_i(M) \leq 1$

Preuve

Les conditions $M = \sum_{i=1}^{3} \lambda_i A_i$ avec $\sum_{i=1}^{3} \lambda_i = 1$ se réécrivent

$$\left\{ \begin{array}{l} x = \lambda_1 a_1 + \lambda_2 a_2 + \lambda_3 a_3, \\ y = \lambda_1 b_1 + \lambda_2 b_2 + \lambda_3 b_3, \\ 1 = \lambda_1 + \lambda_2 + \lambda_3 \end{array} \right. \iff \left(\begin{array}{ccc} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ 1 & 1 & 1 \end{array} \right) \left(\begin{array}{c} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{array} \right) = \left(\begin{array}{c} x \\ y \\ 1 \end{array} \right)$$

qui admet un unique triplet solution $(\lambda_1, \lambda_2, \lambda_3)$ puisque par hypothèse

$$det \left(\begin{array}{ccc} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ 1 & 1 & 1 \end{array} \right) \neq 0$$

Enfin, $\lambda_i(A_j) = \delta_{i,j}$ s'obtient en remarquant qu'il est (l'unique) solution. La dernière condition revient à définir M comme barycentre des A_j.

Programme

```
//Point choisi au hasard
Pt=1-2*rand(2,1);
//calcul des coordonnées barycentriques de M
scmb=[Pt(1);Pt(2);1];//le second membre du système
lambda=M\ scmb; nc=string(Pt');
if lambda(1)>0  lambda(2) >0  lambda(3) >0 then
disp(["le point",nc," est interne au triangle"])
else
disp(["le point",nc," est externe au triangle"])
enddisp(lambda')//coordonnées barycentriques de Pt
plot(Pt(1),Pt(2),'r*')//
```

Programme

```
//Point choisi au hasard
Pt=1-2*rand(2,1);
//calcul des coordonnées barycentriques de M
scmb=[Pt(1);Pt(2);1];//le second membre du système
lambda=M\ scmb; nc=string(Pt');
if lambda(1)>0  lambda(2) >0  lambda(3) >0 then
disp(["le point",nc," est interne au triangle"])
else
disp(["le point",nc," est externe au triangle"])
enddisp(lambda')//coordonnées barycentriques de Pt
plot(Pt(1),Pt(2),'r*')//
```

Résultat

```
column 1 to 3
!le point  0.4720888 0.1703793 !
column 4
! est externe au triangle !
-> disp(lambda')
2.735333 -1.0977964 -0.6375366
```

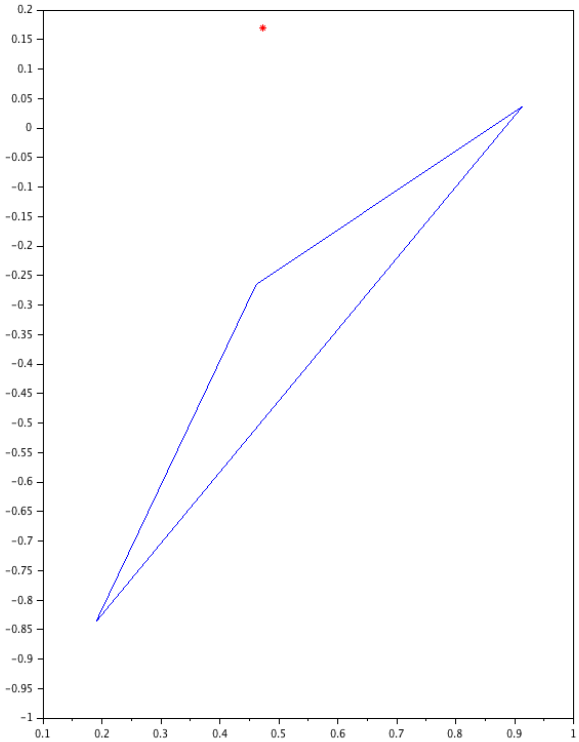


Figure – *Triangle et point*

Polygone

```
c l f ;
n=input ( " n= " ) ;
X=cos (2* %pi/n * [ 0 : n ] ) ;
Y=sin (2* %pi/n * [ 0 : n ] ) ;
plot (X,Y) ;
```

Polygone

```
c l f ;
n=input ( " n= " ) ;
X=cos (2* %pi/n * [ 0 : n ] ) ;
Y=sin (2* %pi/n * [ 0 : n ] ) ;
plot (X,Y) ;
```

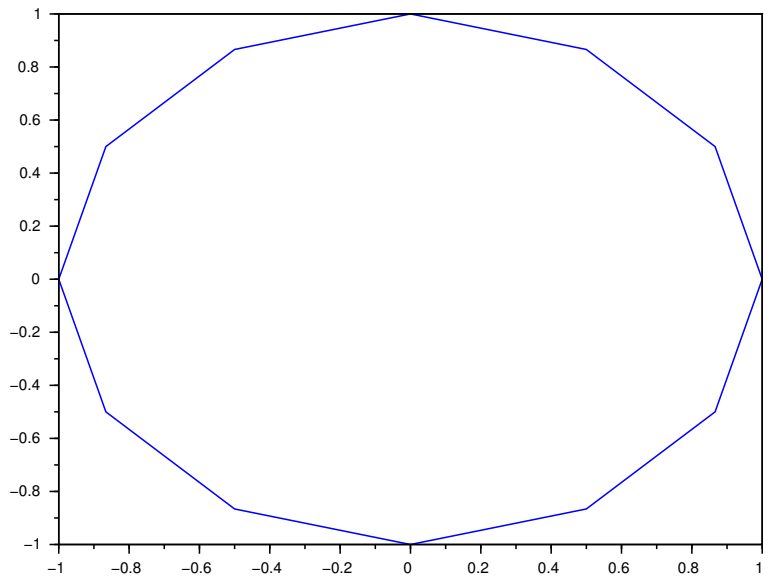
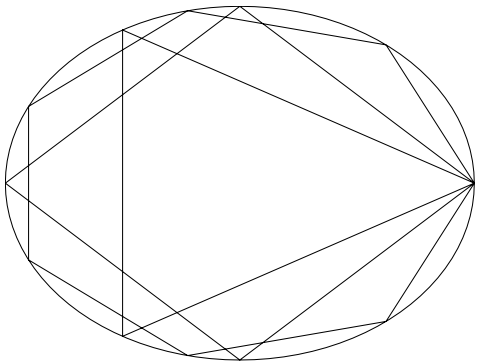
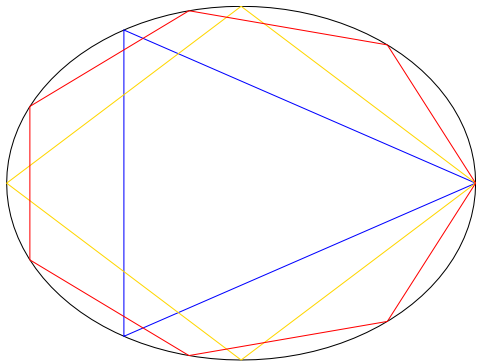


Figure – *Polygone régulier à. 12 côtés*

Plusieurs polygones inscrits

```
clf; X100=cos (2*%pi /100* [ 0 : 100] )';
Y100=sin (2* %pi /100* [ 0 : 100] )';
plot2d ( X100 , Y100 , style=1,axesflag=0) ;
X3=cos (2* %pi / 3 * [ 0 : 3 ] ) ;
Y3=sin (2* %pi / 3 * [ 0 : 3 ] ) ;
plot2d (X3 , Y3 ,style=2,axesflag=0) ;
X4=cos (2* %pi / 4 * [ 0 : 4 ] ) ;
Y4=sin (2* %pi / 4 * [ 0 : 4 ] ) ;
plot2d (X4 , Y4 , style=32,axesflag=0) ;
X7=cos (2* %pi / 7 * [ 0 : 7 ] ) ;
Y7=sin (2* %pi / 7 * [ 0 : 7 ] ) ;
plot2d (X7 , Y7 , style=5,axesflag=0) ;
```





Les commandes

- `xpoly(x,y,"lines",0)` si le polygone est ouvert
- `xpoly(x,y,"lines",1)` si le polygone est fermé
- `xpolys(X,Y,[2,-5])` pour des contours avec lignes ou marques
- `xfpolys(X,Y,[2,-5])` pour un remplissage avec couleurs
- `xrpoly([5,5],6,4)` permet de tracer un polygone régulier, en donnant : centre, nombe de côtés, rayon du cercle dans lequel il est inscrit

Les commandes

- `xpoly(x,y,"lines",0)` si le polygone est ouvert
- `xpoly(x,y,"lines",1)` si le polygone est fermé
- `xpolys(X,Y,[2,-5])` pour des contours avec lignes ou marques
- `xfpolys(X,Y,[2,-5])` pour un remplissage avec couleurs
- `xrpoly([5,5],6,4)` permet de tracer un polygone régulier, en donnant : centre, nombe de côtés, rayon du cercle dans lequel il est inscrit

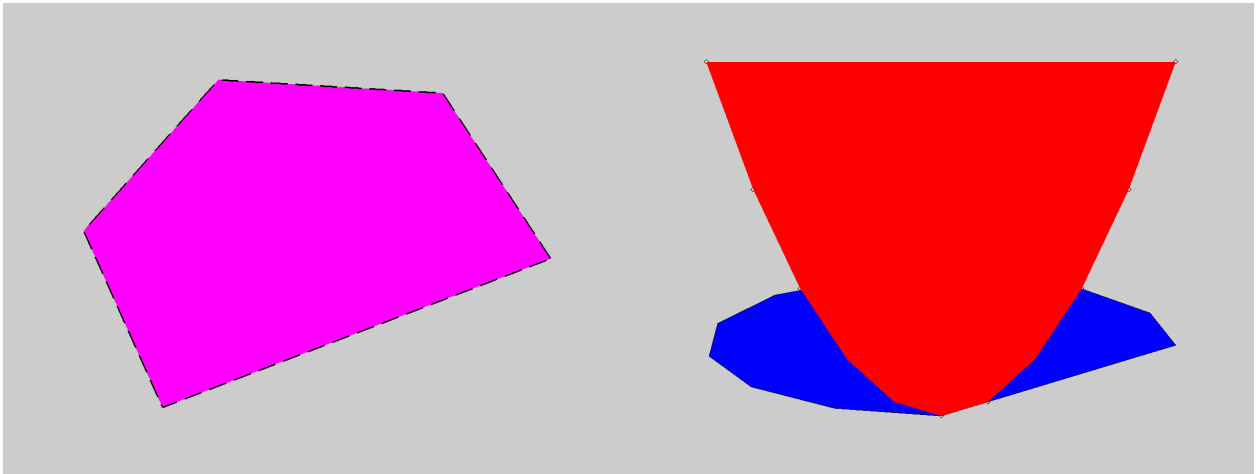


Figure – *Polygones remplis avec* `xfpolys`

La commande permettant de représenter des surfaces dans l'espace est `plot3d`. On définit la surface par exemple sous la forme `z=f(x,y)` ou bien la paramétrant par un angle, lorsqu'elle est de révolution (sphère, tore)

Sphère $\begin{cases} x = R\cos\phi\cos\theta, \\ y = R\cos\phi\sin\theta, \\ z = R\sin\phi \end{cases}$

```
function
[x,y,z]=sphere(theta,phi)
R=0.8
x=R*cos(phi).*cos(theta)
y=R*cos(phi).*sin(theta)
z=R*sin(phi)
endfunction
```

Tore

$$\begin{cases} x = (R + r\cos\phi)\cos\theta, \\ y = (R + r\cos\phi)\sin\theta, \\ z = r\sin\phi \end{cases}$$

```
function
[x,y,z]=tore(theta,phi)
R=1,r=0.2
x=(R+r*cos(phi)).*cos(theta)
y=(R+r*cos(phi)).*sin(theta)
z=r*sin(phi)
endfunction
```

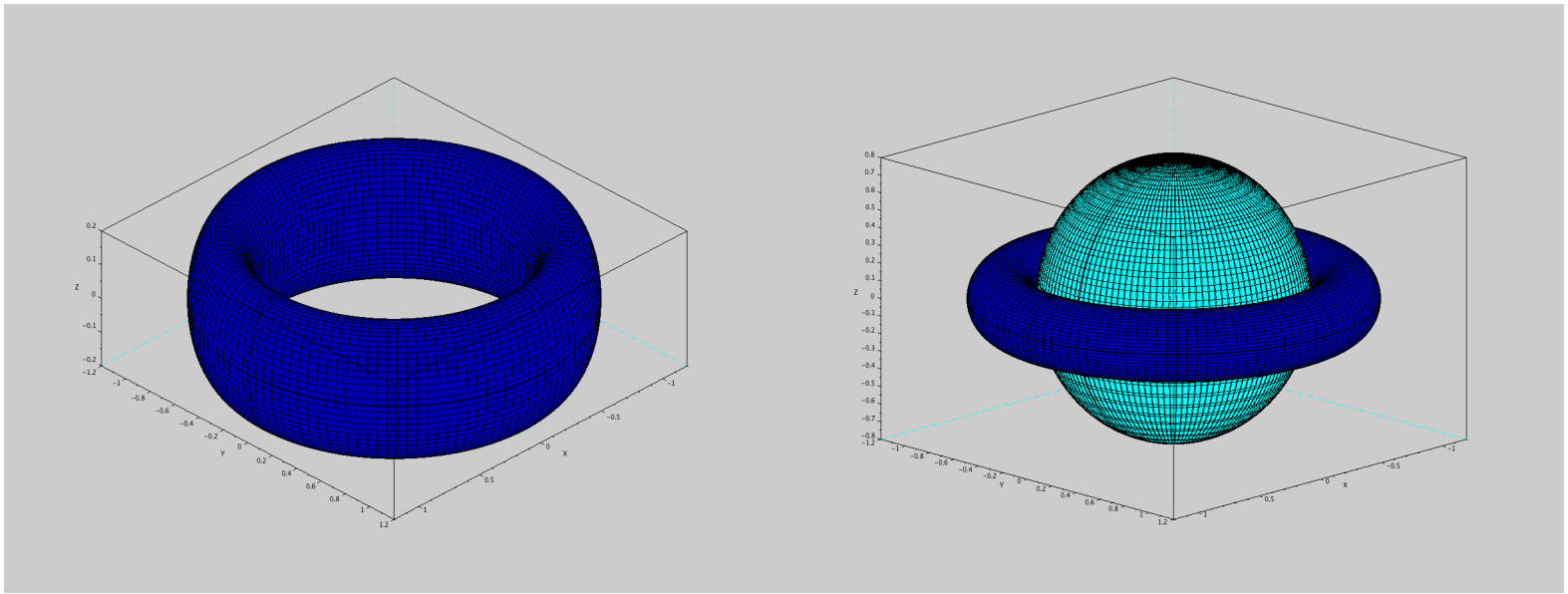


Figure – *Tore et sphère*

Selle à cheval $z = x^2 - y^2$

```
function z=f(x,y)
z=x^2-y^2
endfunction
```

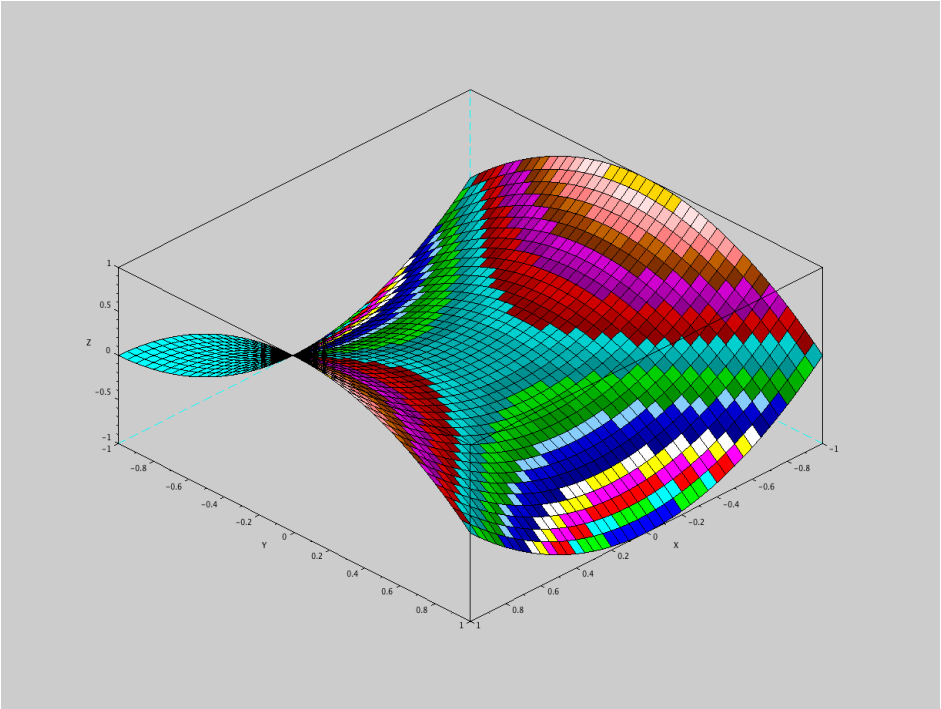


Figure – *Selle à cheval*

Les fractales sont des objets géométriques (courbes ou surfaces) invariants par changement d'échelle (par zoom).

Ensemble de Cantor

Le premier exemple que l'on rencontre est l'ensemble de Cantor : c'est un sous-ensemble fermé de $[0, 1]$ et d'interieur vide.

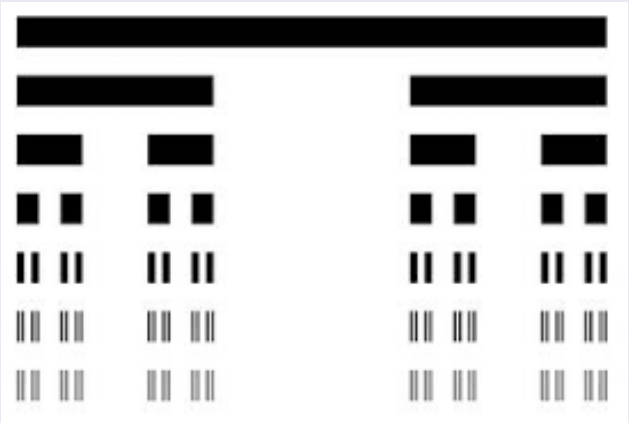


Figure – *Premières itérations de l'ensemble de Cantor*

La courbe

Commençons par la **courbe** de Von Koch , fractale avant l'heure (1904).

Construction en 3 étapes

- On divise le segment de droite en trois segments de longueurs égales.
- On construit un triangle équilatéral ayant pour base le segment médian de la première étape.
- On supprime le segment de droite qui était la base du triangle de la deuxième étape.

On répète (on itère) ces étapes sur chaque arète de la figure obtenue

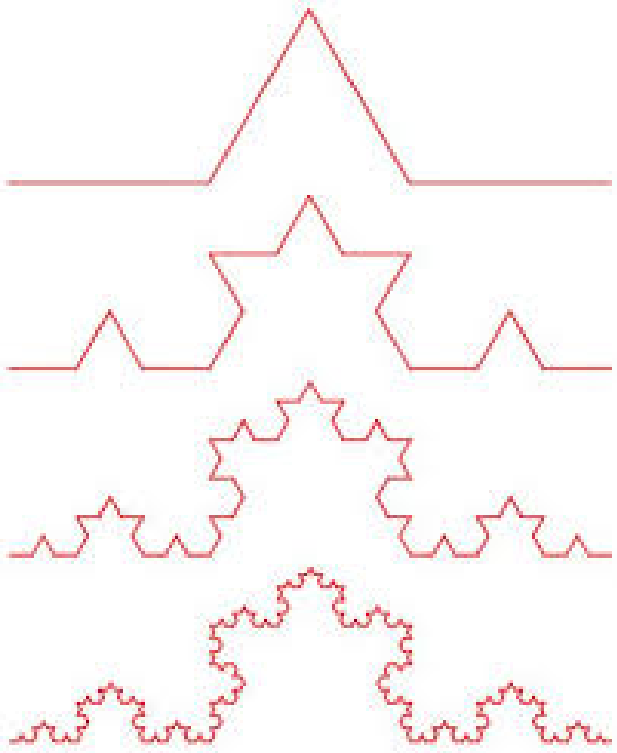


Figure – *Premières itérations de la courbe de von Koch*

Le flocon de Von Koch

Construction

On procè de de la même manière mais on part d'un triangle équilatéral au lieu d'un segment.
On répète (on itère) les étapes sur chaque arète de la figure obtenue

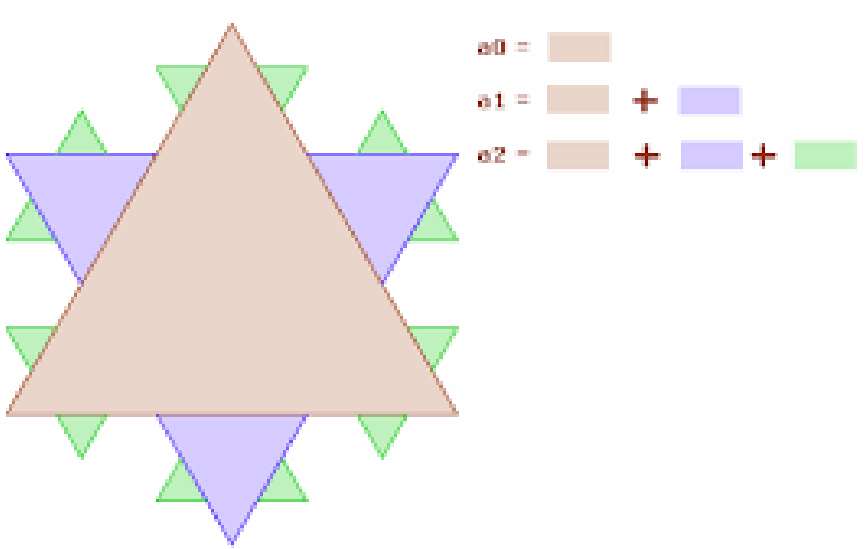


Figure – *Premières itérations du flocon de Von Koch*

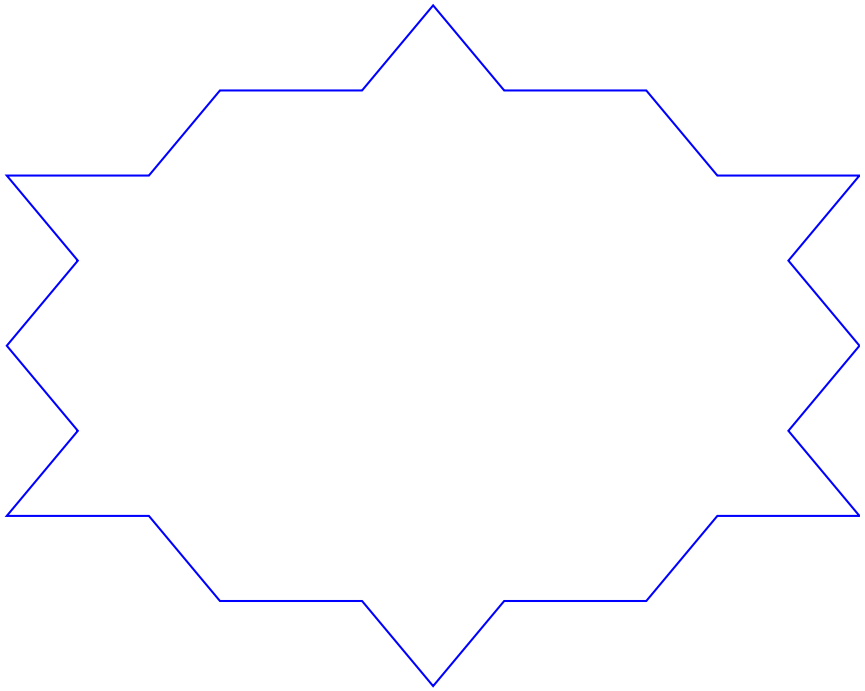


Figure – *Première itération du flocon de Von Koch*

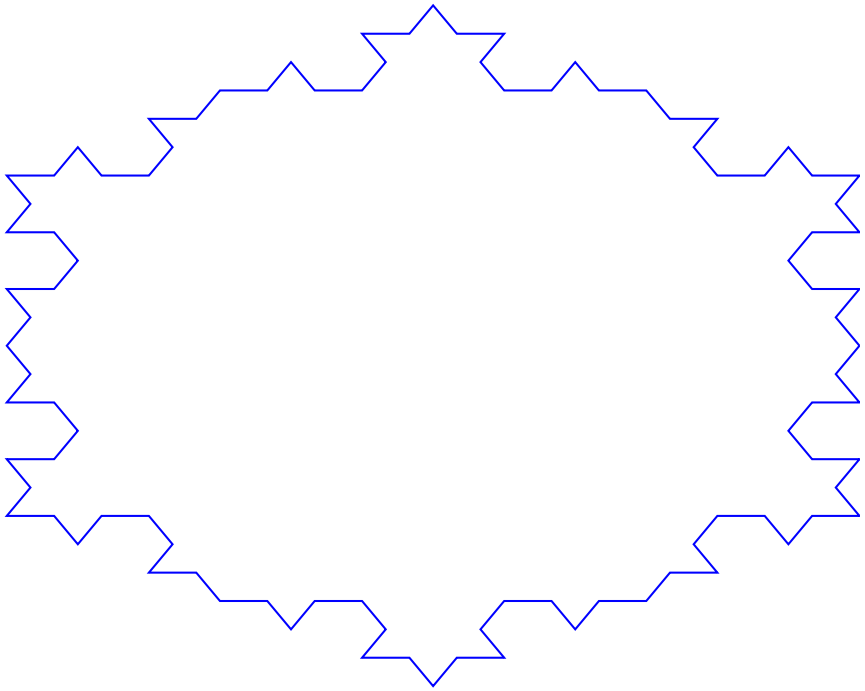


Figure – *deuxième itération du flocon de Von Koch*

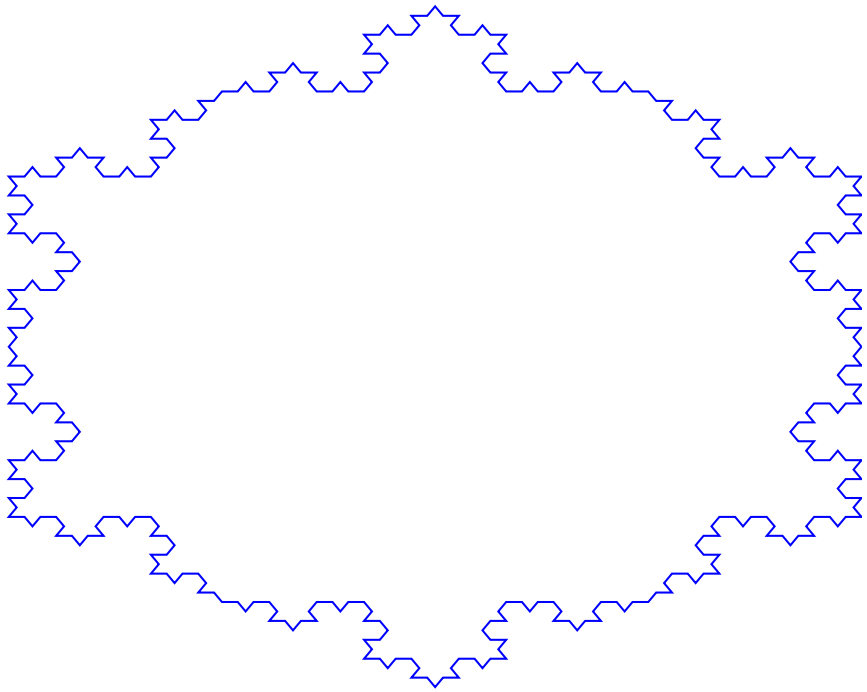


Figure – *troisième itération du flocon de Von Koch*

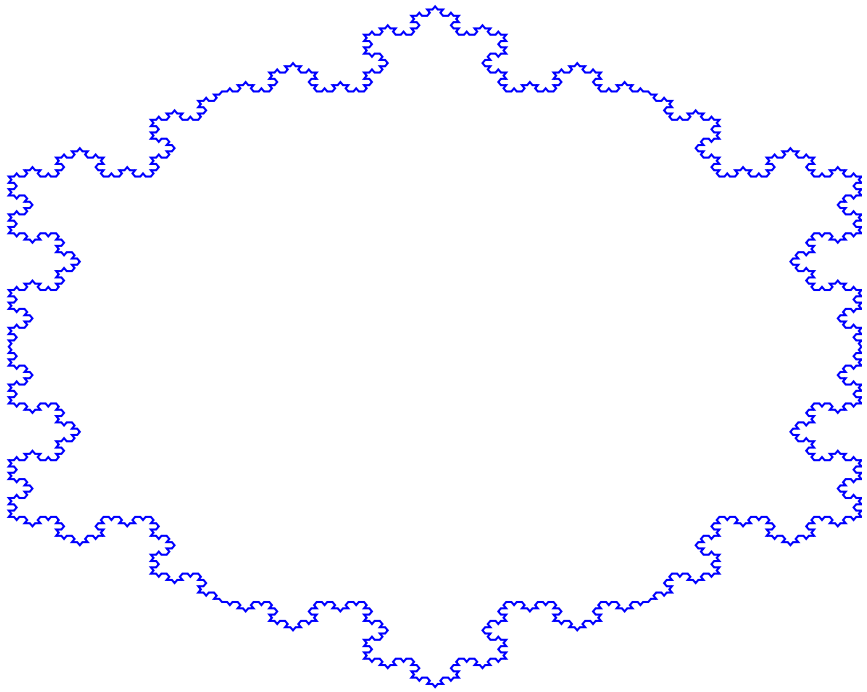


Figure – *quatrième itération du flocon de Von Koch*

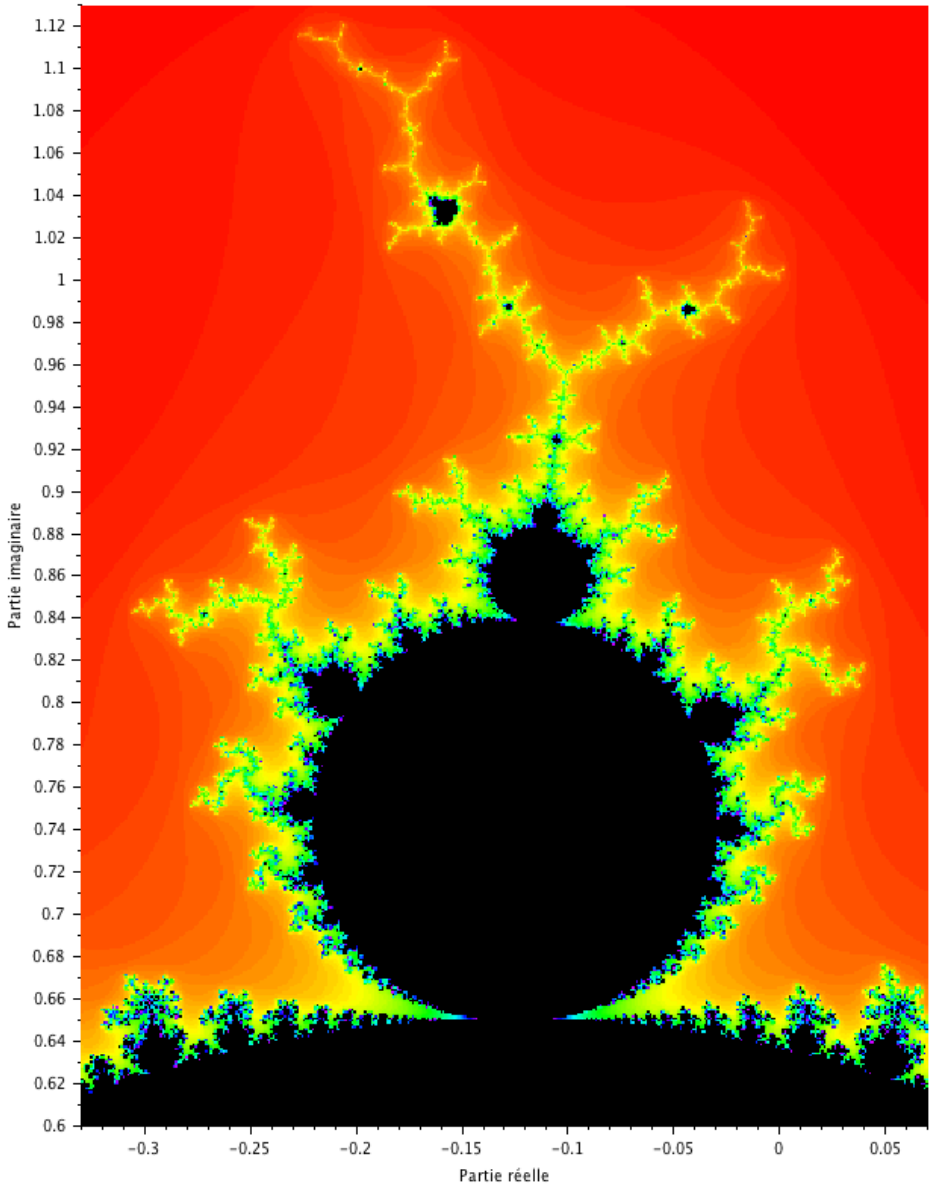
Ce type d'ensemble a été découvert par Gaston Julia et Pierre Fatou avant 1914 et popularisé par Benoit Mandelbrot dans les années 1980, grâce à l'essort des illustrations par ordinateur. Il s'agit d'un ensemble fractal défini par l'ensemble des points c du plan complexe pour lesquels la suite

$$\begin{aligned} z_0 &= 0, \\ z_{k+1} &= z_k^2 + c \end{aligned}$$

est bornée. On le note $\mathcal{M}$. On peut également le définir par son complémentaire :

$$\mathbb{C} \setminus \mathcal{M} = \left\{ c \in \mathbb{C} / \lim_{n \to +\infty} |z_n(c)| = +\infty \right\}$$

autrement dit $\mathcal{M}$ est l'ensemble des valeurs de c pour lesquelles le module de $z_n(c)$ ne tend pas vers l'infini.


1.12
1.1
1.08
1.06
1.04
1.02
1
0.98
0.96
0.94
0.92
0.9
0.88
0.86
0.84
0.82
0.8
0.78
0.76
0.74
0.72
0.7
0.68
0.66
0.64
0.62
0.6
Partie imaginaire
−0.3
−0.25
−0.2
−0.15
−0.1
−0.05
0
0.05
Partie réelle

Une condition nécessaire pour $c \in \mathcal{M}$

On montre une condition suffisante pour $c \notin \mathcal{M}$

Lemma

i. *Si* $|c| > 2$ *alors* $\lim_{n \to +\infty} |z_n(c)| = +\infty$ *(i.e.* $c \notin \mathcal{M}$*)*

ii. *On se donne* $c \in \mathbb{C}$*. S'il existe* $n_0 \in \mathbb{N}$ *tel que* $|z_{n0}(c)| > 2$ *alors* $\lim_{n \to +\infty} |z_n(c)| = +\infty$ *(i.e.* $c \notin \mathcal{M}$*). La réciproque est vraie.*

Conséquence

L'ensemble $\mathcal{M}$ est contenu dans le disque de centre 0 et de rayon 2.

Preuve

On établit d'abord i. Soit $|c| > 2$. On montrer par récurrence la propriété

$$\mathcal{H}_n \qquad |z_n = z_n(c)| \geq |c|\left(|c| - 1\right)^{n-1}, \forall n \geq 1.$$

Pour $n = 1$, on a $z_1 = c$, relation est vérifiée.
Soit maintenant $n \geq 1$. Supposons que $|z_n| \geq |c|\left(|c| - 1\right)^{n-1}$. On a

$$\begin{aligned}
|z_{n+1}| &= |z_n^2 + c| \\
&\geq ||z_n|^2 - |c|| \\
&\geq \left(|c|(|c| - 1)^{n-1}\right)^2 - |c| \text{ (par hypothèse de récurrence)} \\
&\geq |c|^2(|c| - 1)^{n-1} - |c|(|c| - 1)^{n-1} \text{ (car } |c| > 2) \\
&\geq |c|(|c| - 1)^{n-1}(|c| - 1) = |c|(|c| - 1)^n
\end{aligned}$$

On se donne une fenêtre graphique

$$\{x_{min} \leq \Re(c) \leq x_{max}, y_{min} \leq \Im(c) \leq y_{max}\}$$

On se donne $N \in \mathbb{N}^*$ et construit alors le réseau de points $c_{i,j}$ tels que

$$\Re(c_{i,j}) = x_{min} + (i-1)\frac{x_{max} - x_{min}}{N}, \Im(c_{i,j}) = y_{min} + (j-1)\frac{y_{max} - y_{min}}{N}, i, j = 1, \cdots, N+1$$

On pose `hx=(xmax-xmin)/N` et `hy=(ymax-ymin)/N`

```
N=1000,kmax=100,xmin=-2;ymin=-2;xmax=2;ymax=2;
for i=1:N+1
for j=1:N+1
c = xmin + (i-1)*hx+ %i * (ymin + (j-1)*hy)
k=0;z=0;
while (k < kmax & abs(z) <= 2)
z = z∧2+c;
k = k+1
end
if (abs(z) <= 2) // on stocke Re(c) et Im(c) des c sélectionnés
RM=[RM real(c)]; RI=[RI imag(c)];
end
end
```

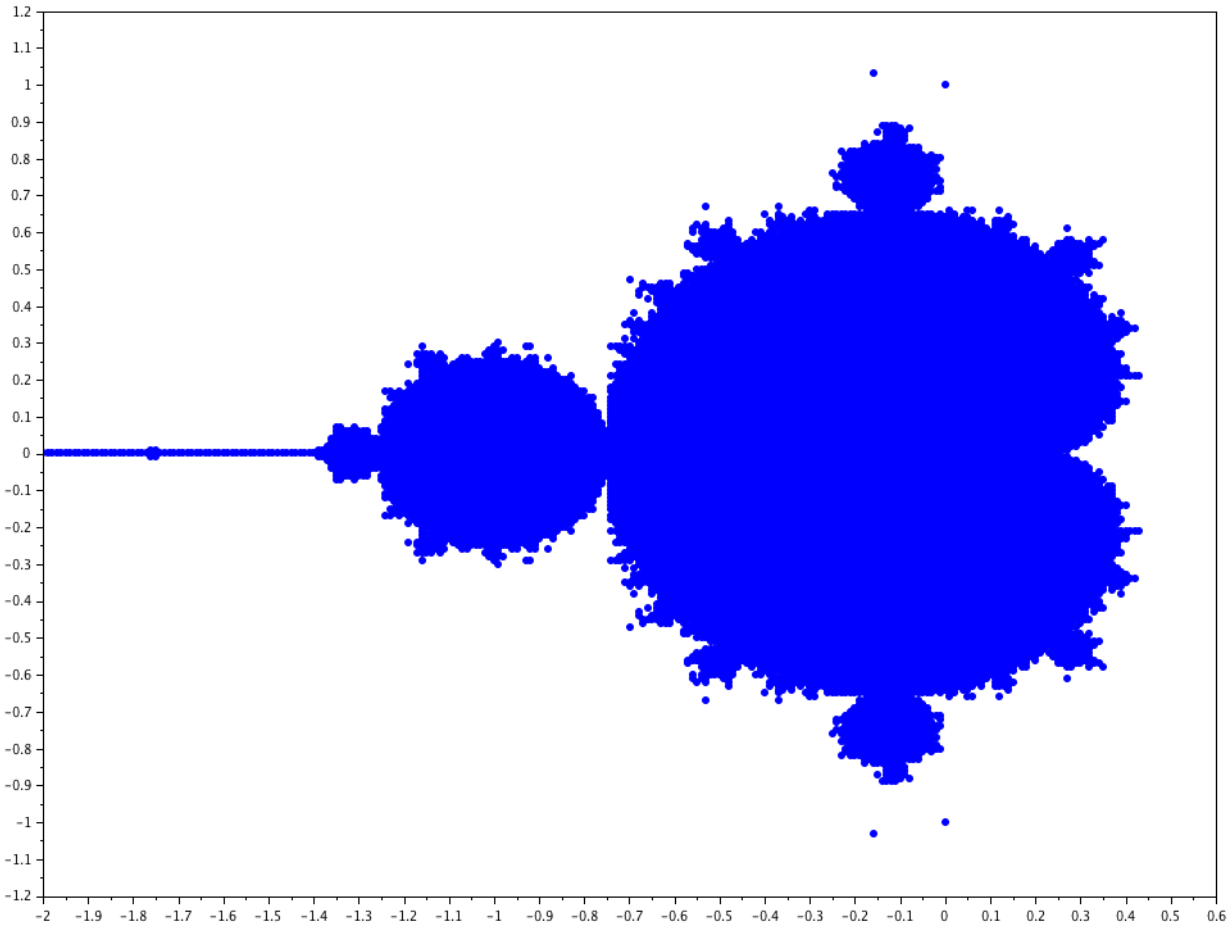


Figure – *Ensemble de Mandelbrot*, $N = 400$, $Kmax = 30$

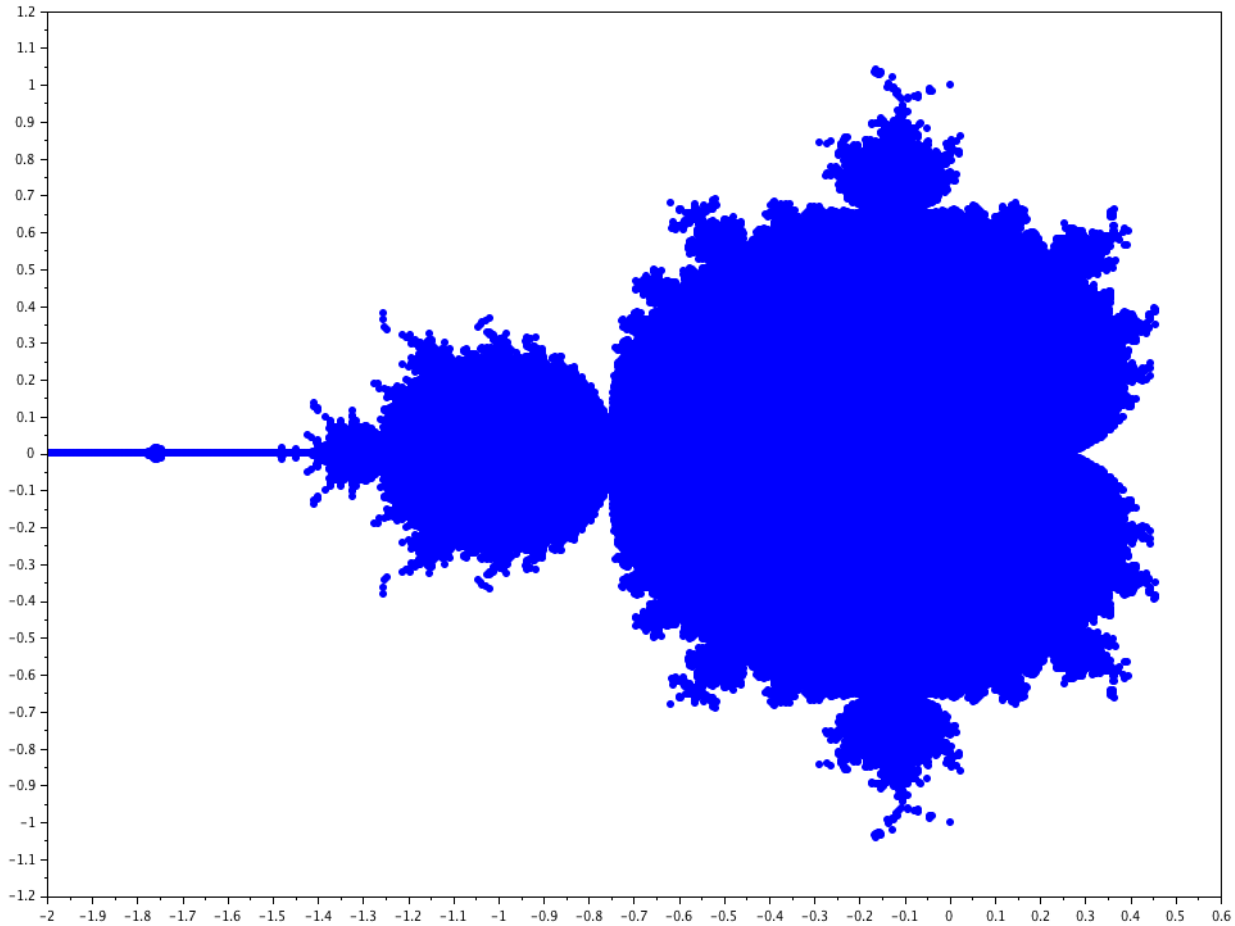


Figure – *Ensemble de Mandelbrot*, $N = 1000, Kmax = 30$

En 2D

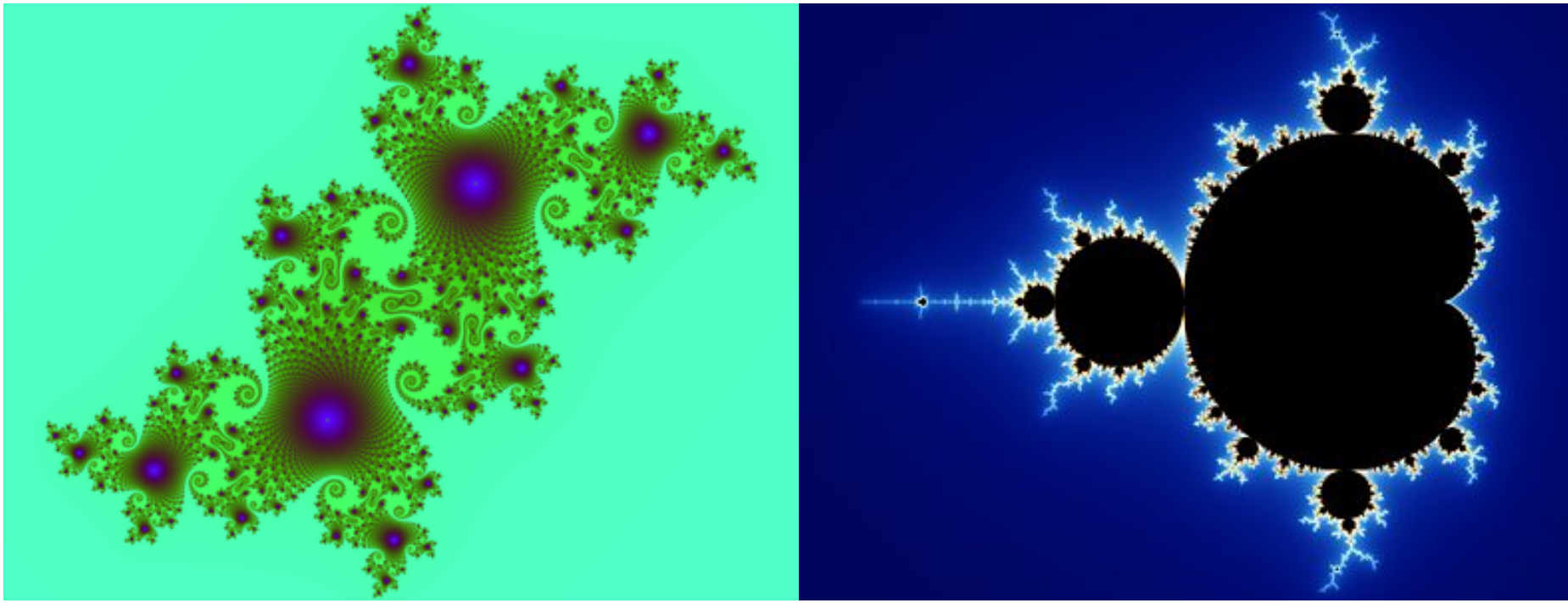


Figure – *Ensembles de Julia et de Mandelbrot*

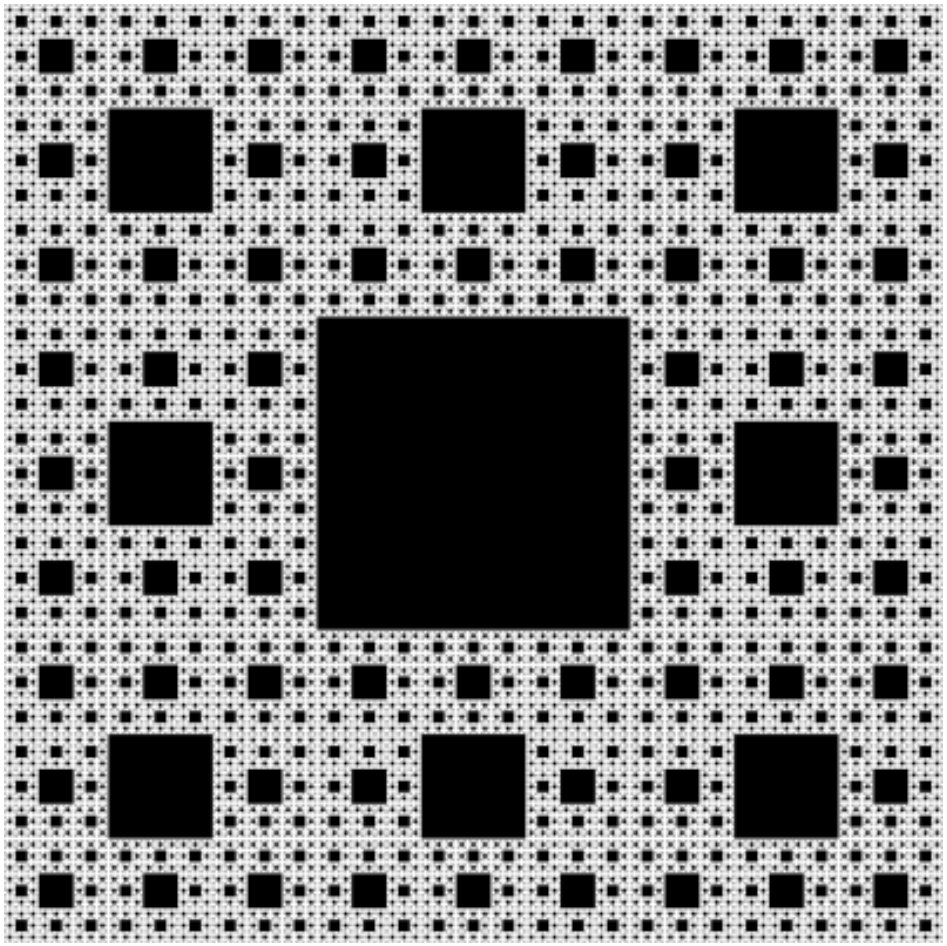


Figure – *Tapis de Sierpinski (version 2D de l'ensemble de Cantor)*

Dans la nature

Figure – *Vous voulez des broccolis ? et des cristaux ?*

En 3D

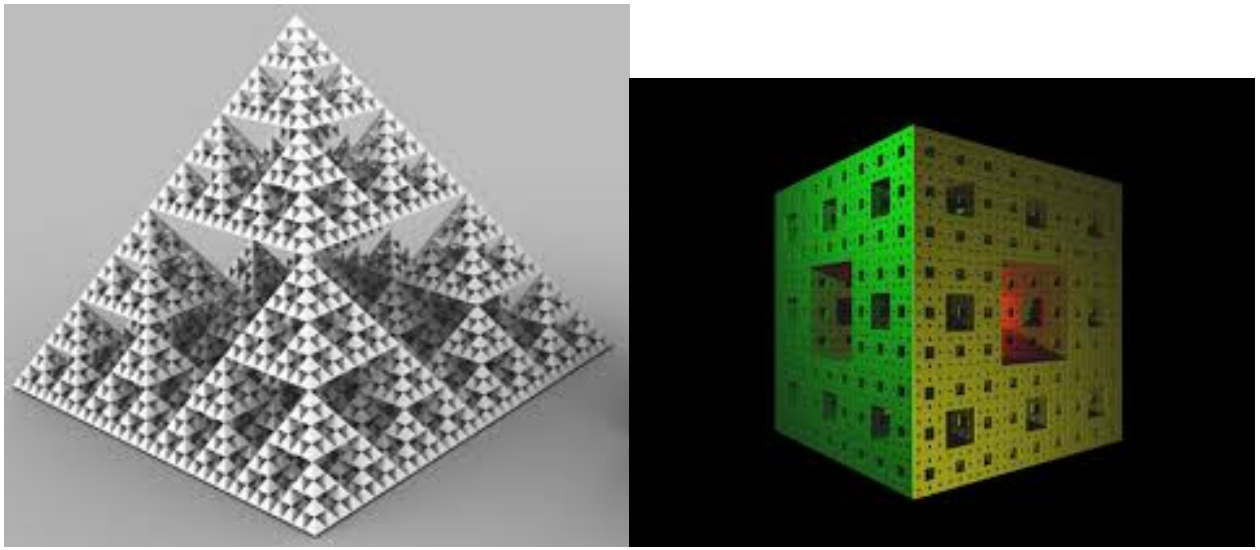


Figure – *Sierpinski 3D*

- Représentations 3D
  `https://complexe.jimdofree.com/les-fractales/galerie-images/fractales-en-3d-existe/`

- Animation 3D (youtube)
  ▸ Link