

## II. Etirer et plier : le chaos du malaxage

### Attracteurs étranges feuilletés

Nous allons passer des récurrences en une dimension, comme celle de l'application logistique, aux récurrences en deux dimensions. Cela permettra de préciser les notions d'attracteurs chaotiques, d'attracteurs étranges, de bassins d'attraction. Mais comment créer du désordre par des phénomènes simples et répétitifs? Le pétrissage de la pâte à pain ou le malaxage dans une bétonnière sont des exemples familiers. Commençons par là.

#### 1. La transformation du boulanger

Prenons un carré, censé représenter de la pâte à pain, puis aplatissons-le de façon que sa longueur double et sa largeur soit divisée par deux. Ensuite coupons-le au milieu et mettons le deuxième morceau par-dessus le premier, ce qui redonne un carré (*figure 1*). En prenant comme repère deux bordures du carré, celui-ci ayant un côté de longueur 1, un point  $(x, y)$  de la pâte à pain est d'abord transformé en  $(2x, y/2)$ , puis il reste là si  $x < 0,5$  ou sinon il devient  $(2x - 1, y/2 + 0,5)$ .

Ecrivons cette transformation en binaire. Par exemple le point  $x = 0,1011\dots, y = 0,0110\dots$ , avec  $x > 0,5$ , est transformé en  $x' = 0,011\dots, y' = 0,10110\dots$ . Pour simplifier, on condense l'écriture en plaçant  $x$  à droite de la virgule et  $y$  à l'envers à gauche de la virgule. Le point  $(x, y)$  s'écrit  $\dots 0110,1011\dots$ , et il devient  $\dots 01101,011\dots$ . Autrement dit, le nouveau point s'obtient en décalant d'un cran la virgule vers la droite. Il en est de même lorsque  $x < 0,5$ . Par exemple, le point  $x = 0,0111\dots, y = 0,0110\dots$  devient  $x' = 0,111\dots, y' = 0,110$ , soit en écriture condensée :  $\dots 0110,0111\dots$  devient  $\dots 0110,0111\dots$ , ce qui revient aussi à un décalage d'un cran à droite de la virgule. Cela se vérifie dans le cas général.

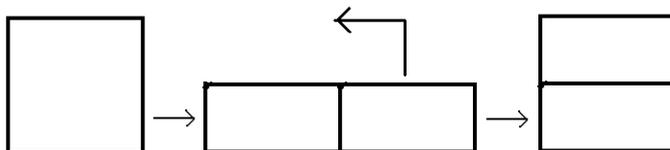


Figure 1 : La transformation du boulanger.

Répetons maintenant cette opération, comme le fait le boulanger qui pétrit sa pâte à pain. Pour simplifier plaçons-nous en une dimension, en étudiant le mouvement de l'abscisse  $x$  (le comportement de  $y$  est analogue). La fonction agissante s'appelle l'application décalage. Supposons d'abord que  $x$  soit un nombre rationnel, c'est-à-dire une fraction d'entiers. Son écriture en binaire est toujours finalement périodique, comme par exemple  $5 / 110 = 0,0 \underline{45} \underline{45} \underline{45}\dots$ . Par le jeu des décalages successifs à droite,  $x$  va être transformé en un nombre à l'écriture périodique, et la trajectoire devient périodique. Par contre si  $x$  est un nombre quelconque, en ce sens qu'il ne présente aucune périodicité dans son écriture (il est irrationnel) sa trajectoire ne finira pas par devenir périodique, et poursuivra son cheminement erratique. Si l'on ajoute le fait que l'exposant de Liapounov vaut  $\log 2 > 0$ , puisque la dérivée de la fonction est 2, il s'ensuit un phénomène de sensibilité aux conditions initiales. Comme les nombres irrationnels sont infiniment plus nombreux que les nombres rationnels, leur comportement désordonné va prévaloir, et il se produit un phénomène de mixage dans la pâte à pain, comme le boulanger le sait depuis longtemps, sans rien y voir d'irrationnel. Cela nous permet de définir ce que l'on appelle une trajectoire chaotique (en une dimension, mais cela s'étend aux dimensions supérieures).

Trajectoire chaotique : Il s'agit d'une trajectoire qui n'est pas asymptotiquement périodique, et telle que l'exposant de Liapounov est supérieur à 0.

Remarquons la nécessité de ces deux conditions, puisqu'un nombre rationnel a une trajectoire non chaotique alors que l'exposant de Liapounov est positif. La notion d'asymptotiquement périodique indique la tendance à devenir périodique lorsque le nombre d'itérations augmente à l'infini. Une trajectoire finalement périodique est asymptotiquement périodique, mais l'inverse n'est pas vrai.

### Exercice 1 : Fonction décalage et fonction tente

1) L'application décalage en une dimension fait passer de  $x$  à  $x'$ . Tracer sa courbe représentative.

L'application faisant passer de  $x$  à  $x'$  est telle que  $x' = 2x$  si  $x < 0,5$  ou  $2x - 1$  si  $x > 0,5$ , soit

$x' = 2x$  ramené modulo 1 (à la limite pour  $x = 0,5$  ou pour  $x = 1$ , on fera  $x' = 0$ ).

Sa courbe se situe dans le carré de 1 sur 1, elle est formée de deux segments de pente 2 (figure 2)

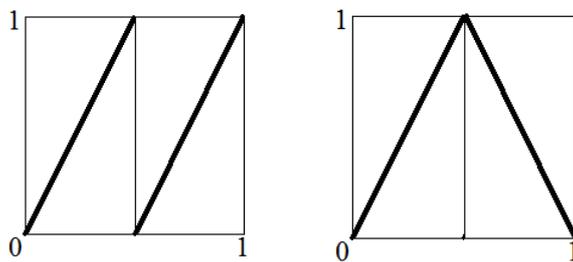


Figure 2 : Application décalage en une dimension à gauche, application tente à droite.

2) Modifions maintenant l'expérience précédente, en nous rapprochant du geste du boulanger. Plutôt que de couper la pâte qu'il a étalée, il la plie pour replacer le deuxième morceau sur le premier. Le changement est minime par rapport au cas précédent. Au lieu de l'application décalage, on a l'application « tente », à cause de la forme de la courbe représentative de  $x'$  en fonction de  $x$ . Tracer cette courbe et donner son équation.

L'application « tente » s'écrit, en une dimension :  $x' = 2x$  pour  $x < 0,5$  et  $x' = 2(1 - x)$  pour  $x > 0,5$ . La courbe représentative a la forme d'un  $\Lambda$  (figure 2 à droite).

3) Ecrire le mouvement d'un point lors d'un étirement-plier en deux dimensions pour le carré de pâte à pain. Programmer ce mouvement en le répétant un certain nombre de fois. Constaté que les trajectoires, d'où qu'elles partent, convergent vers le point  $(0,0)$ . Pourquoi l'ordinateur livre-t-il ce résultat complètement faux ?

Prenons un point  $M(x, y)$  dans le carré. Lors de l'étirement,  $M$  devient  $M_1$  tel que :  $x_1 = 2x$ ,  $y_1 = y/2$ . Puis effectuons le pliage, le point  $M_1$  devenant  $M_2$  confondu avec  $M_1$  si  $x_1 < 1$ , ou tel que  $[M_1M_2]$  a pour milieu le point  $(1, 1/2)$  si  $x_1 > 1$ , soit  $x_2 = 2 - x_1$ ,  $y_2 = 1 - y_1$  (figure 3).

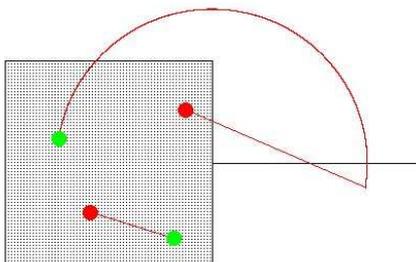


Figure 3 : A partir d'un point initial en rouge, on arrive au point en vert après l'étirement-plier.

Le programme suivant permet de répéter ce mouvement une soixantaine de fois :

```

rectangle(xorig+zoom,yorig-zoom/2,xorig+2*zoom,yorig,black);
x0=0.87,y0=0.76; /* point initial */
x=x0;y=y0; filldisc(xorig+zoom*x,yorig-zoom*y,7,red);
for(i=0;i<60;i++) /* boucle des étirements-pliages */
{ x1=2.*x; y1=y/2.;
  line(xorig+zoom*x,yorig-zoom*y,xorig+zoom*x1,yorig-zoom*y1,red);
  if(x1>1.) {x2=2.-x1; y2=1.-y1;}
  else {x2=x1;y2=y1;}
  if(x1>1.) demicercle(x1,y1);1
  filldisc(xorig+zoom*x2,yorig-zoom*y2,2,red);
  x=x2;y=y2;
}
filldisc(xorig+zoom*x2,yorig-zoom*y2,7,green); /* point final */

```

On constate que le point final est en  $O$ , quel que soit le point de départ que l'on prenne (*figure 4*), comme si toute la matière se concentrait en  $O$ , ce qui est complètement contraire à la réalité. Si ce résultat est faux, c'est parce que l'ordinateur travaille sur du fini. On a vu que le mouvement d'un point se traduisait par un décalage de la virgule pour un nombre écrit en binaire. Lors de ce décalage à droite, seul  $x$  est concerné mais pas  $y$ . Au bout d'un certain nombre de décalages, quand on atteint les limites de la machine, car le nombre de chiffres des nombres flottants est limité, tout se passe au-delà comme si l'on n'avait plus que des 0.

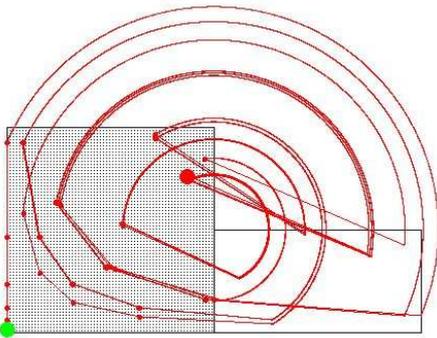


Figure 4 : Mouvement d'un point (rouge) lors du pétrissage, et son arrivée finale en  $O$  (point vert).

4) Reprendre l'application décalage en une dimension, et faire des corrections pour que l'on retrouve la réalité du mixage. La correction consiste à utiliser des nombres vraiment quelconques. Le meilleur moyen est de fabriquer une suite de chiffres 0 ou 1 au hasard, comme si l'on jouait à pile ou face. Il s'ensuivra une trajectoire chaotique à partir de là. Dans l'ordinateur, c'est grâce au hasard, en fait une suite de nombres pseudo-aléatoires, obéissant à un phénomène déterministe mais lui aussi chaotique, que l'on obtient du chaos !

Commençons par faire le diagramme en toile d'araignée correspondant à l'application décalage, faisant passer de  $x$  à  $x' = 2x$  [modulo 1]. Comme précédemment, on constate qu'au bout de quelques dizaines d'itérations, on tombe définitivement sur 0, puis que l'on y reste indéfiniment. C'est ce que donne le programme suivant, avec son résultat sur la *figure 6 à gauche*.

<sup>1</sup> A défaut d'utiliser une fonction `arc()`, on peut faire ce petit programme du demi-cercle :

```

void demicercle(float xx,float yy)
{ double rr,x,y,m;
  rr=(xx-1.)*(xx-1.)+(yy-0.5)*(yy-0.5); /* rayon au carré du cercle */
  m=(yy-0.5)/(xx-1.); /* pente du diamètre M1M2 */
  for(x=-0.2;x<2.2;x+=0.001) for(y=-0.7;y<1.7;y+=0.001)
  if (fabs((x-1.)*(x-1.)+(y-0.5)*(y-0.5)-rr)<0.001 && y>m*(x-1.)+0.5)
  putpixel(xorig+zoom*x,yorig-zoom*y,red);
}

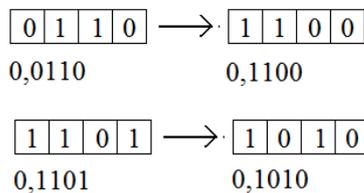
```

```

x0=0.87; x=x0;filldisc(xorig+zoom*x0,yorig,5,red); /* point initial */
x1=2*x; if (x>0.5) x1=-1.; line(xorig+zoom*x,yorig,xorig+zoom*x,yorig-zoom*x1,black);
for(i=1;i<60;i++) /* même avec des milliers d'itérations on reste sur le point final 0 */
{ line(xorig+zoom*x,yorig-zoom*x1,xorig+zoom*x1,yorig-zoom*x1,black);
  xx=2*x1; if (x1>0.5) xx=-1.; if (x1==1.) xx=0.;
  line(xorig+zoom*x1,yorig-zoom*x1,xorig+zoom*x1,yorig-zoom*xx,black);
  x=x1;x1=xx;
}
line(xorig+zoom*x,yorig,xorig+zoom*x,yorig-zoom*x1,black); filldisc(xorig+zoom*x,yorig,5,green);

```

Maintenant pour partir d'un nombre "quelconque", voici une façon de procéder. Dans un tableau de longueur  $N$  très grande (quelques dizaines de milliers de cases), mettons au hasard des 0 et des 1. Cette succession de chiffres constitue la partie derrière la virgule d'un nombre en binaire. L'application décalage  $x' = 2x [1]$  revient exactement à décaler d'un cran à gauche tous les chiffres en binaire du tableau, ce qui écrase le chiffre en position 0, qu'il soit 0 ou bien 1. Par la même occasion, on place un 0 dans la dernière case qui était devenue vide lors du décalage. Deux exemples sont donnés sur la *figure 5*.



*Figure 5* : L'application décalage, sur deux exemples en binaire, avec  $N = 4$ .

En faisant cela de façon répétée, le nombre se termine par une quantité de plus en plus grande de 0, et il n'a rien d'irrationnel. Mais si l'on se contente de faire quelques milliers d'itérations sur un nombre qui a une longueur dix fois plus grande, c'est la partie initiale du nombre qui joue, avec son aspect quelconque. Du même coup, le diagramme en toile d'araignée occupe de façon dense la zone concernée, comme on le constate sur la *figure 6* à droite, grâce au programme suivant.

```

for(i=0;i<N;i++) a[i]=rand()%2; /* remplissage au hasard du tableau a[] */
x0=nombre(); x=x0;filldisc(xorig+zoom*x0,yorig,5,red); /* point initial x0 après conversion du nombre */
for(i=1;i<=N-1;i++) a[i-1]=a[i]; /* décalage de la virgule d'un cran */
a[N-1]=0; /* ajout d'un 0 à droite */
x1=nombre(); /* ici x1 est le transformé de x0 */
line(xorig+zoom*x,yorig,xorig+zoom*x,yorig-zoom*x1,black);
for(etape=1;etape<2000;etape++) /* boucle de quelques milliers d'étapes */
{ line(xorig+zoom*x,yorig-zoom*x1,xorig+zoom*x1,yorig-zoom*x1,black);
  for(i=1;i<=N-1;i++) a[i-1]=a[i];
  a[N-1]=0;
  xx=nombre(); /* transformé de x1, converti en décimal */
  line(xorig+zoom*x1,yorig-zoom*x1,xorig+zoom*x1,yorig-zoom*xx,black);
  x=x1;x1=xx;
}
line(xorig+zoom*x,yorig,xorig+zoom*x,yorig-zoom*x1,black);
filldisc(xorig+zoom*x,yorig,5,green);

double nombre(void) /* conversion du binaire en décimal, seulement pour pouvoir faire le dessin */
{ double div,n; int i;
  div=2.; n=0.;
  for(i=0;i<1000;i++) { n+=a[i]/div; div=2.*div; } /* inutile d'avoir une trop grande précision */
  return n;
}

```

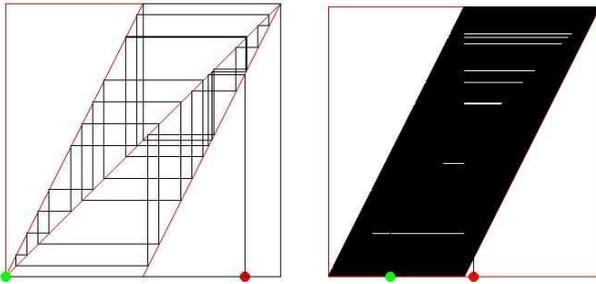


Figure 6 : Application décalage, avec le point initial en rouge, et le point final en vert. A gauche, le résultat donné par l'ordinateur est faux, à droite on retrouve le malaxage recherché.

### Exercice 2: De l'application tente à l'application logistique

Reprendre l'application logistique  $L$  lorsque  $c = 4$ , pour ce que l'on a appelé le chaos ultime, tel qu'on l'a observé expérimentalement. Par le biais de la parabole d'équation :  $y' = 4y(1 - y)$  dans le repère  $O y y'$ , l'intervalle de départ  $[0, 1]$  est transformé en lui-même, mais après avoir doublé de longueur et avoir été plié. De son côté, l'application tente  $T$ , avec sa courbe en forme de  $\Lambda$  dans le repère  $O x x'$ , s'écrit  $x' = 2x$  pour  $0 \leq x \leq 0,5$  et  $x' = 2 - 2x$  pour  $0,5 < x < 1$ . Son lien avec l'application logistique peut être établi en pratiquant le changement de variable  $y = \sin^2(\pi x / 2)$ , cette transformation  $U$  étant due à S. Ulam et J. Von Neumann. Vérifier qu'il en est bien ainsi. Cela démontrera le caractère chaotique de l'application logistique pour  $c = 4$ , avec une trajectoire de points qui parcourt de façon dense l'intervalle  $[0, 1]$  d'une façon d'apparence imprévisible. Précisons que pour d'autres valeurs de  $c$  le comportement est tout aussi chaotique d'après l'expérience, mais on ne sait pas démontrer pourquoi.

Prenons un point  $x$  et son correspondant  $y = \sin^2 \pi x$ .<sup>3</sup>

Puis appliquons  $T$  à  $x$  et  $L$  à  $y$  :

$x \mapsto x' = 2x$  ou  $2 - 2x$  selon que  $x$  est plus petit ou plus grand que  $0,5$ .

$y \mapsto y' = 4y(1 - y)$ .

Il s'agit de prouver que  $y' = \sin^2 \pi x'$ .

Or  $y' = 4\sin^2 \pi x (1 - \sin^2 \pi x) = 4\sin^2 \pi x \cos^2 \pi x = (2\sin \pi x \cos \pi x)^2 = \sin^2 2\pi x$

Calculons maintenant  $\sin^2 \pi x'$  en distinguant deux cas :

\* si  $0 \leq x \leq 0,5$ ,  $\sin^2 \pi x' = \sin^2 2\pi x$

\* si  $0,5 < x < 1$ ,  $\sin^2 \pi x' = \sin^2 (2\pi - 2\pi x) = \sin^2 2\pi x$

Finalement, avec  $y = U(x)$ , on a aussi  $y' = U(x')$  lorsque  $x' = T(x)$  et  $y' = L(y)$ .

Les deux suites liées l'une à l'application tente et l'autre à l'application logistique, démarrant avec  $y_0 = U(x_0)$ , sont ensuite liées pas à pas par la fonction  $U$  :

$$\begin{array}{ccccccc}
 x_0 & \xrightarrow{T} & x_1 & \xrightarrow{T} & x_2 & \xrightarrow{T} & \dots \\
 \downarrow U & & \downarrow U & & \downarrow U & & \\
 y_0 & \xrightarrow{L} & y_1 & \xrightarrow{L} & y_2 & \xrightarrow{L} & \dots
 \end{array}$$

<sup>2</sup> La suite de points associée à l'application tente que l'on répète, a le même caractère chaotique que l'application décalage. Pour faire le lien avec l'application logistique, il est plus simple de prendre l'application tente plutôt que l'application logistique, car la courbe en  $\Lambda$  de l'application tente est assez proche de la parabole de l'équation logistique.

<sup>3</sup> La fonction  $U$  fait passer de  $x$  sur  $[0, 1[$  à  $y$  sur  $[0, 1[$ . Elle est aussi bijective, son inverse étant  $x = (2/\pi) \text{Arcsin} \sqrt{y}$ .

Cette méthode d'étirement-plier est à la base des techniques de malaxage et de feuilletage. Elles donnent lieu à des phénomènes chaotiques, deux points très proches au départ finissant par se séparer irrémédiablement. En reprenant cette idée et en la prolongeant, Michel Hénon, astronome français, a développé en 1976 un modèle où les trajectoires convergent vers ce que l'on appelle un attracteur étrange et chaotique.

## 2. L'attracteur de Hénon

Considérons la transformation qui fait passer d'un point du plan  $(x, y)$  au point  $(x', y')$  avec

$$\begin{cases} x' = 1 + y - ax^2 \\ y' = bx \end{cases} \text{ en prenant } b = 0,3 \text{ comme l'a fait M. Hénon.}$$

Cette transformation est non-linéaire mais seulement du second degré, elle est inversible, et elle contracte les surfaces, ce qui est indispensable pour que, lorsque l'on répète cette transformation, les trajectoires convergent vers un ensemble limite.

Pour comprendre son lien avec la méthode d'étirement-plier, il suffit de la décomposer, en faisant :

1) la transformation  $x_1 = x, y_1 = 1 + y - ax^2$ , qui correspond à un étirement-plier donnant une parabole d'axe vertical à partir d'un segment horizontal

2) la transformation  $x_2 = bx_1, y_2 = y_1$ , qui correspond à un serrage de la parabole ( $b = 0,3 < 1$ )

3)  $x' = y_2, y' = x_2$ , qui intervertit les axes, et met la parabole selon un axe horizontal (figure 7). On retrouve bien la relation entre le point initial  $(x, y)$  et le point final  $(x', y')$  de la transformation de Hénon.

Il s'agit exactement du phénomène d'étirement-plier, par le biais de courbes lisses (continues et sans rupture de pente), à la différence de l'application du boulanger. L'action répétée de cette transformation crée un feuilletage. Signalons l'importance de la troisième transformation, car sans elle il n'y aurait pas de feuilletage.



Figure 7 : A gauche la transformation de Hénon, décomposée en trois étapes, avec au début un rectangle aplati, et à la fin un pliage avec étirement. Les paramètres sont  $a = 1,4$  et  $b = 0,3$ . A droite, l'attracteur de Hénon, c'est-à-dire la forme obtenue après répétitions multiples de la transformation.

### Exercice 3 : Attracteur étrange et bassin d'attraction

1) Avec  $b = 0.3$  et  $a = 1,4$ , tracer sur ordinateur la trajectoire d'un point  $(x_0, y_0)$  comme par exemple  $(0, 0)$  en pratiquant de façon répétée la transformation précédente. Constaté qu'elle converge vers un ensemble de points en forme de  $\supset$ , qui n'est pas vraiment une courbe mais plutôt une sorte de feuilletage. Changer, sans exagérer, le point de départ de la trajectoire et constater qu'elle vient se coller sur le même ensemble limite, que l'on appelle l'attracteur.

Le programme est particulièrement simple, et le résultat obtenu est donné sur la figure 7 à droite.

$x_0=0.; y_0=0.; x=x_0;y=y_0;$

```

for(i=0;i<10000000;i++)
{ newx=y+1-a*x*x; newy=b*x; x=newx; y=newy;
  if ( i>10000)
    { xe=xorig+zoom*x; ye=yorig-zoom*y; putpixel(xe,ye,black);
    }
}

```

2) Faire des zooms sur de petites parties de l'ensemble limite pour observer le feuilletage.

On prend une fenêtre carrée de centre  $(x_c, y_c)$  et de longueur  $l$  (par exemple  $l = 0,05$ ) dans la zone de calcul, qu'il s'agit de grossir sur l'écran avec une longueur  $L$  (par exemple  $L = 500$ ), soit un zoom de  $L/l$ . Puis on prend la trajectoire du point  $(x_0, y_0)$ , et l'on ne garde que les points qui sont dans la fenêtre de longueur  $l$  (figure 8).

```

L=500.; l=0.08; xc=0.3;yc=0.2;
zoom=L/l;
x0=0.;y0=0.; x=x0;y=y0;
for(i=0;i<100000000;i++)
{ newx=y+1-a*x*x; newy=b*x; x=newx; y=newy;
  if (fabs(x-xc)<1/2 && fabs(y-yc)<1/2 && i>10000)
    { xe=L/2+zoom*(x-xc); ye=L/2-zoom*(y-yc); putpixel(xe,ye,black);
    }
}

```

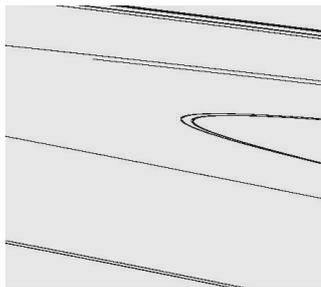


Figure 8 : Zoom sur une petite zone, ici de centre  $(0,3, 0,2)$ .

3) Prendre maintenant tous les points sur l'écran dont les coordonnées  $x$  et  $y$  dans le domaine du calcul sont comprises entre  $-2$  et  $2$ . Ils se séparent en deux catégories: ceux dont la trajectoire converge vers l'ensemble limite précédent et ceux dont la trajectoire s'en va à l'infini. La première catégorie est appelée le bassin d'attraction de l'attracteur, et l'autre le bassin d'attraction de l'infini. Pour améliorer la visualisation, colorier de façon différente les points du bassin de l'infini selon leur vitesse d'échappée (leur trajectoire traverse plus ou moins vite un grand carré tel que  $|x|=1000$  ou  $|y|=1000$  par exemple).

Les points  $(x_e, y_e)$  de la zone écran sont convertis en  $(x, y)$  de la zone calcul, et à partir de ces derniers on lance leur trajectoire. Celle-ci a deux possibilités, soit s'échapper à l'infini, soit rester dans une zone finie. On considère que le fait de sortir d'un carré centré en  $O$  et de longueur  $20$  indique que la trajectoire part à l'infini. La vitesse d'échappée vers l'infini est prise en compte grâce à une palette cyclique `coul[ ]` à base de 4 couleurs (figure 9).

```

for(xe=xorig-2.3*zoom;xe<xorig+2.3*zoom;xe++)
for(ye=yorig-2.3*zoom;ye<yorig+2.3*zoom;ye++)
{ x0=(xe-xorig)/zoom; y0=(ye-yorig)/zoom;
  x=x0;y=y0;
  for(i=0;i<100;i++)
    { newx=y+1-a*x*x; newy=b*x; x=newx; y=newy;
      if (fabs(x)>10. && fabs(y)>10.) { putpixel(xe,ye,coul[i%4]);break; }
    }
}

```

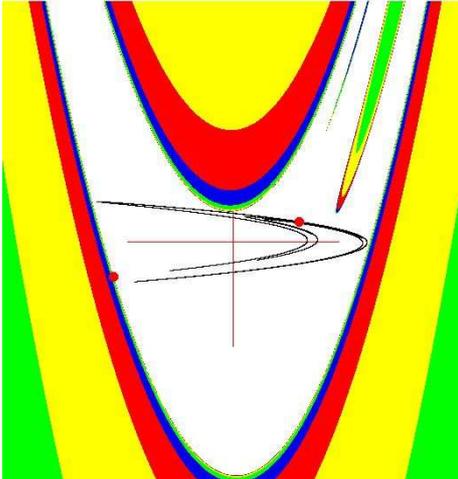


Figure 9 : L'attracteur en noir avec son bassin d'attraction en blanc. Le bassin d'attraction vers l'infini est colorié en vert, jaune, rouge, bleu, suivant la vitesse d'échappée. Les deux points fixes de l'application de Hénon sont représentés par deux points rouges.

4) Augmenter maintenant  $a$ , et constater que l'attracteur disparaît. Il ne reste plus que le bassin d'attraction de l'infini. Colorier les points selon la vitesse d'échappée de leur trajectoire, pour  $a = 1,8$  par exemple.

Lorsque le paramètre  $a$  atteint 1,43, l'attracteur disparaît brutalement, et par la même occasion son bassin d'attraction est remplacé par une zone de points dont la trajectoire part à l'infini, avec une grande dispersion des vitesses d'échappée (figure 10 à gauche). Lorsque  $a$  continue d'augmenter, les vitesses d'échappée s'organisent selon des formes fluides (figure 10 à droite).

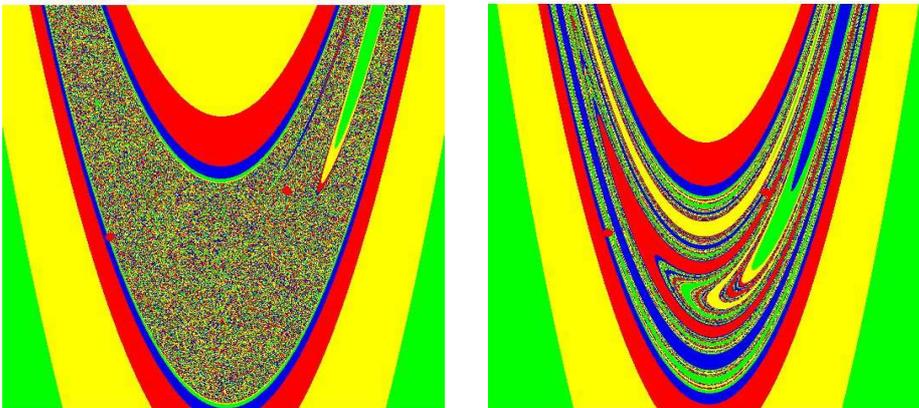


Figure 10 : A gauche le bassin d'attraction vers l'infini occupe tout pour  $a = 1,43$ . A droite, la même situation pour  $a = 1,8$ , avec des vitesses d'échappée mieux organisées.

On sait maintenant ce qui se passe pour  $a = 1,4$ , et au-delà avec la disparition de l'attracteur feuilleté, mais qu'arrive-t-il avant ? Tout comme l'application logistique, la transformation de Hénon présente un schéma de bifurcations lorsque l'on fait varier le paramètre  $a$  de 0 jusqu'à 1,4 (en gardant  $b = 0,3$ ). Commençons par chercher les points fixes de l'application de Hénon. Ceux-ci vérifient :

$$x = 1 + y - ax^2, y = bx, \text{ ce qui donne}$$

$$x = (b - 1 \pm \sqrt{(b-1)^2 + 4a}) / 2a, y = bx$$

d'où deux points fixes dès que  $a$  est supérieur à  $a_0 = -(b - 1)^2/4 = -0,1225$ . L'un est d'abord stable (attracteur puits) et l'autre instable (point hyperbolique), comme on peut le constater par le calcul, puis le point fixe stable devient à son tour instable, et donne naissance à deux nouveaux points

fixes stables (dès que  $a = 0.37$ ), et ainsi de suite comme pour l'application logistique. Un se divise en deux, des bifurcations apparaissent, jusqu'à l'apparition de l'attracteur chaotique qui se développe jusqu'à  $a = 1.42$ . Pour le constater, il suffit de tracer la courbe donnant les abscisses  $x$  des fins de trajectoire à partir d'un point initial  $(x_0, y_0)$ , en fonction de  $a$  (figure 11)<sup>4</sup>.

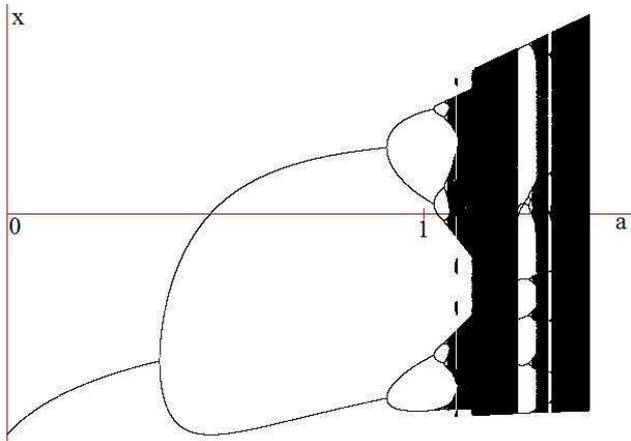


Figure 11 : Schéma de bifurcations lors de la transition vers le chaos de l'application de Hénon.

### 3. Notions d'attracteur étrange et d'attracteur chaotique

Lorsque la trajectoire d'un point  $x_0$ , sous l'effet répété d'une application, finit par se coller sur un ensemble de points qui peut être réduit à un point (un point fixe stable) ou à un cycle de points, ou à un ensemble infini et dense de points situés dans une zone finie, comme pour la transformation de Hénon, cet ensemble est appelé ensemble limite pour le point  $x_0$ . Plus précisément, la trajectoire d'un point  $x_0$  possède un ensemble limite si chaque point de cet ensemble limite est approché d'aussi près qu'on veut par des points de la trajectoire, et cela infiniment souvent. Cet ensemble est globalement invariant sous l'effet de la transformation. Il est appelé attracteur quand il existe toute une surface dont les points ont leur trajectoire convergeant vers ce même ensemble limite. La surface en question est le bassin d'attraction de l'attracteur. Lorsque l'attracteur présente une structure fractale, notamment feuilletée comme l'attracteur de Hénon, il est dit attracteur étrange. Quand la trajectoire issue d'un point de l'attracteur est chaotique (présentant des phénomènes de mixage et de sensibilité aux conditions initiales), il est dit attracteur chaotique.

Encore convient-il qu'un attracteur existe. Une condition nécessaire pour qu'il en soit ainsi est que l'application faisant passer de  $(x, y)$  à  $(x', y')$  contracte les surfaces. Ainsi, une surface initiale, soumise aux effets répétés de l'application, va voir son aire diminuer à chaque étape, pour venir finalement se coller sur l'attracteur. Au contraire une application dilatante provoquerait un grossissement infini d'une surface au départ finie, empêchant toute formation d'un attracteur.

Comment savoir si l'application est contractante ? Prenons un point  $(x_0, y_0)$  et plaçons-nous dans un petit voisinage de ce point avec un point  $(x, y)$ . Comme  $x - x_0$  et  $y - y_0$  sont proches de 0, on peut négliger les termes de la forme  $(x - x_0)^n$  et  $(y - y_0)^n$  lorsque  $n > 1$  dans le développement en série de l'application. Celle-ci, qui s'écrit  $x' = f(x, y)$ ,  $y' = g(x, y)$ , peut être linéarisée, en ce sens que l'on ne conserve d'elle que les termes du premier degré en  $x - x_0$  et  $y - y_0$ . L'application s'écrit alors sous forme d'une matrice :

<sup>4</sup> Cela signifie qu'en faisant quelques milliers d'itérations pour  $(x, y)$  on ne garde que le dernier millier de points. On va ainsi trouver, lorsque  $a$  augmente à partir de 0, la présence d'un point fixe (un millier de valeurs de  $x$  identiques), puis d'un cycle fixe de deux points, jusqu'au développement final de la zone chaotique.

$$\begin{pmatrix} \frac{\partial f}{\partial x}(x_0, y_0) & \frac{\partial f}{\partial y}(x_0, y_0) \\ \frac{\partial g}{\partial x}(x_0, y_0) & \frac{\partial g}{\partial y}(x_0, y_0) \end{pmatrix}$$

Son déterminant en valeur absolue est égal à l'aire obtenue après l'action de l'application sur une aire unité. Par exemple, pour la transformation de Hénon, si l'on prend un point au voisinage de l'origine  $O$ , le déterminant s'écrit :

$$\begin{vmatrix} 0 & 1 \\ b & 0 \end{vmatrix} = -b$$

et il y a bien contraction pour  $b = 0,3$ . Toute surface située dans le bassin d'attraction au voisinage de  $O$  va au fil des itérations voir son aire tendre vers 0, et elle va venir se coller sur l'attracteur. Mais remarquons qu'une surface située dans le bassin d'attraction de l'infini devient elle aussi nulle, sans qu'il y ait un attracteur dans le fini.

#### **Exercice 4 : Un attracteur chaotique mais pas du tout étrange**

Prendre l'application en polaires faisant passer du point  $(r, \theta)$  au point  $(r', \theta')$  par

$$\begin{cases} r' = \sqrt{r} \\ \theta' = 2\theta \end{cases}$$

Montrer qu'il admet un attracteur simple, quel que soit le point initial  $(r_0, \theta_0)$  autre que  $O$ .

Supposons d'abord  $r_0 > 1$ , et montrons que la suite est décroissante, soit  $r_{n+1} < r_n$  pour tout  $n$ , et que l'on a aussi  $r_n > 1$ . Pour cela, faisons un raisonnement par récurrence.

\*  $r_0 > 1$  et  $r_1 = \sqrt{r_0} < r_0$ ,  $r_1 - r_0 < 0$ , la formule est vraie au départ.

\* Supposons que l'on ait  $r_{n+1} - r_n < 0$  à un certain rang  $n$ , ainsi que  $r_n > 1$ , et montrons que cela reste vrai au rang suivant :

$$r_{n+2} - r_{n+1} = \sqrt{r_{n+1}} - \sqrt{r_n} = \frac{r_{n+1} - r_n}{\sqrt{r_{n+1}} + \sqrt{r_n}} < 0 \text{ et d'autre part } r_{n+1} = \sqrt{r_n} > 1 \text{ car } r_n > 1.$$

La suite  $(r)$  étant décroissante et minorée par 1, elle admet une limite  $R$ , et cette limite vérifie la relation de récurrence, soit  $R = \sqrt{R}$  d'où  $R^2 = R$ . Comme  $R$  ne peut pas être nul, il reste  $R = 1$ . Les points de la suite convergent sur le cercle de centre  $O$  et de rayon 1.

Si l'on prend maintenant  $r_0 < 1$ , le même raisonnement conduit au fait que la suite  $(r)$  est croissante avec  $r_n < 1$ , et que les points de la suite convergent aussi vers le même cercle que précédemment.

La suite des points  $(r, \theta)$  admet ce cercle ou une partie de ce cercle comme attracteur. Si l'on prend en plus  $\theta$  qui double à chaque étape modulo  $2\pi$ , et que  $\theta$  est de la forme  $a 2\pi$ , où  $a$  n'est pas une fraction d'entiers, les points  $(r, \theta)$  décrivent de façon dense le cercle, avec une évolution chaotique sur ce cercle, car le fait de se ramener modulo  $2\pi$  provoque une évolution erratique de la courbe, cela étant confirmé par la sensibilité aux conditions initiales, deux points voisins ayant leur distance qui double à chaque étape. On obtient le cercle comme attracteur chaotique, mais pas du tout étrange.

#### **Exercice 5 : Bassins d'attraction à géométrie complexe**

Reprendre la transformation de Hénon, mais en prenant  $b = -0.3$ . En faisant varier le paramètre  $a$  de 1 à 3, constater que la frontière du bassin d'attraction de l'attracteur passe d'une structure lisse à une structure fractale.

On constate que pour les valeurs choisies du paramètre  $a$ , l'attracteur commence par être un point, avant de se dédoubler en deux points puis subir le schéma classique des bifurcations, et finir par

disparaître. Lorsque l'on est dans la zone où il existe un cycle de deux points attracteur, son bassin d'attraction passe d'une frontière lisse à une structure fractale, avec l'apparition d'une sorte de feuilletage où s'intercalent des zones d'attraction (figure 12).

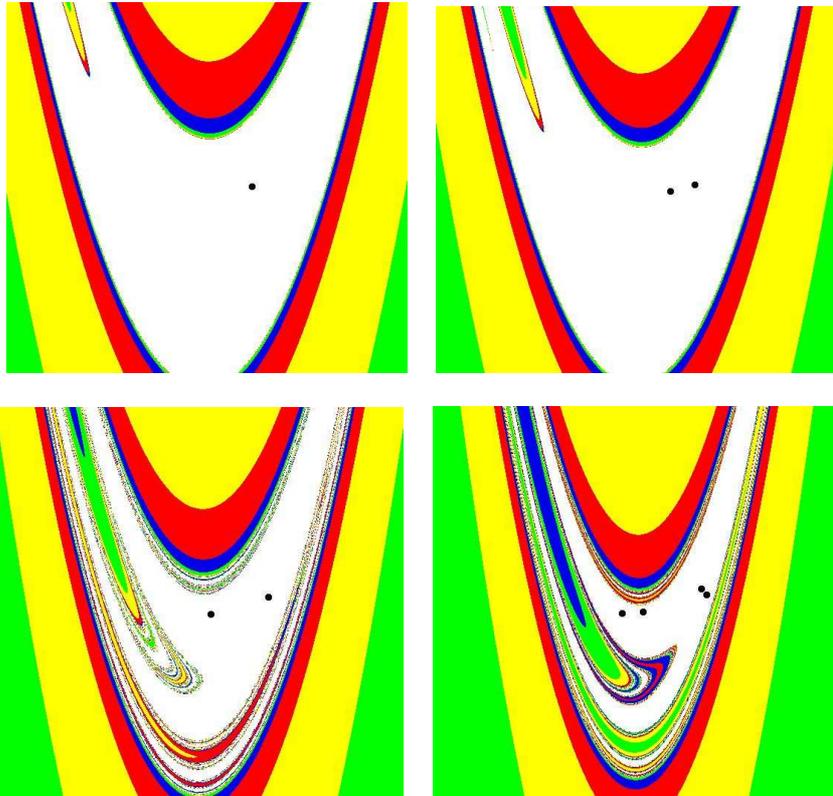


Figure 12 : Attracteur et son bassin d'attraction en blanc, lorsque  $a$  augmente. Un cycle de deux points apparaît pour  $a = 1,3$ , avec un bassin d'attraction à frontière lisse, celle-ci devenant fractale pour  $a = 1,5$ , et qui le reste lorsque le cycle passe à 4 points pour  $a = 1,9$ . L'attracteur et son bassin disparaissent pour  $a = 2,2$ .

## 4. Exemples d'attracteurs étranges et de bassins d'attraction étranges

### 4.1. L'attracteur d'Ikeda<sup>5</sup>

Considérons la fonction faisant passer du point  $(x, y)$  au point  $(x', y')$  par

$$\begin{cases} x' = K + C_2(x \cos t - y \sin t) \\ y' = C_2(x \sin t + y \cos t) \end{cases} \text{ avec } t = C_1 - C_3 / (1 + x^2 + y^2)$$

#### Exercice 6 : Etude de l'attracteur d'Ikeda

1) Prendre comme valeur des paramètres  $R = 0,9$ ,  $C_1 = 0,4$ ,  $C_2 = 0,9$ ,  $C_3 = 6$ . Programmer pour avoir la trajectoire du point  $(1,0)$ , obtenue par l'action répétée de la fonction précédente. On constate que la trajectoire vient s'enrouler sur un attracteur étrange, lui aussi plié et feuilleté -l'attracteur d'Ikeda.

```
x=1.;y=0.;
for(i=0;i<1000000;i++)
{ t=C1-C3/(1.+x*x+y*y);
```

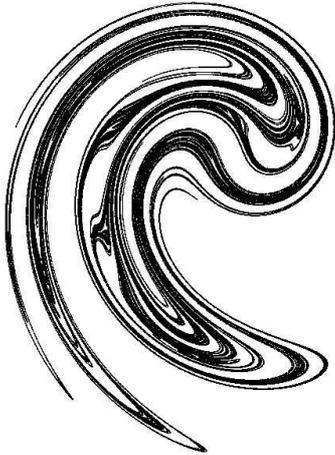
<sup>5</sup> En 1980, K. Ikeda propose une fonction qui s'écrit en complexes  $z' = K + C_2 \exp(i(C_1 - C_3 / (1 + |z|^2)))$  pour modéliser des phénomènes optiques non-linéaires, comme des impulsions lumineuses émises par un laser et circulant dans une cavité en anneau non-linéaire.

```

oldx=x; x=K+C2*(x*cos(t)-y*sin(t)); y=C2*(oldx*sin(t)+y*cos(t));
putpixel(xorig+zoom*x,yorig-zoom*y,black);
}

```

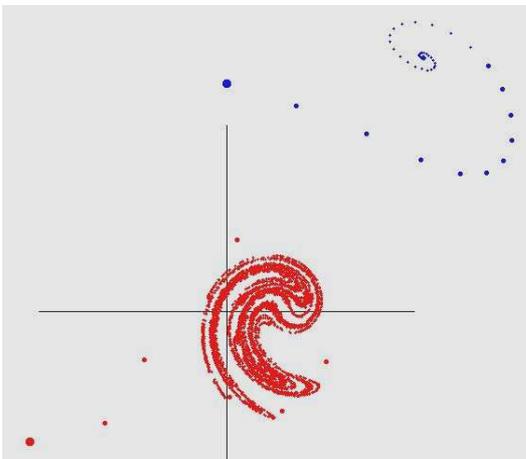
La *figure 13* donne le résultat du programme.



*Figure 13* : Attracteur d'Ikeda.

2) Constaté, en changeant les points de départ des trajectoires, que certaines vont vers l'attracteur, et que d'autres convergent vers un point fixe (puits).

On a pris une trajectoire partant du point (-3, -2) et convergeant vers l'attracteur d'Ikeda, et une autre partant de (0, 3,5) convergeant vers un point fixe (*figure 14*).



*Figure 14* : En rouge, une trajectoire convergeant vers l'attracteur, en bleu une trajectoire convergeant vers un point fixe.

3) En se plaçant dans un disque de rayon 10 autour de l'origine, classer les points selon que leur trajectoire est attirée vers le point fixe ou vers l'attracteur étrange. Cela donne un dessin en deux couleurs, représentant l'un le bassin d'attraction du puits, l'autre celui de l'attracteur étrange.

```

zoom=L/20.;xo=L/2.; yo=L/2.; /* on prend un carré d'écran, avec l'origine au centre */
for(xe=0;xe<L;xe++) for(ye=0;ye<L;ye++)
{ x=(double)(xe-xorig)/zoom; y=(double)(yorig-ye)/zoom; /* passage de la zone écran à la zone calcul */
for(i=0;i<500;i++)
{ d2=x*x+y*y; if(d2>100.) break; /* on ne garde que les points d'un cercle de rayon 10 */
t=C1-C3/(1.+d2); oldx=x;oldy=y;
x=K+C2*(x*cos(t)-y*sin(t)); y=C2*(oldx*sin(t)+y*cos(t));
/* il y a convergence vers le point fixe dès que deux points successifs de la trajectoire
sont très proches */
}
}

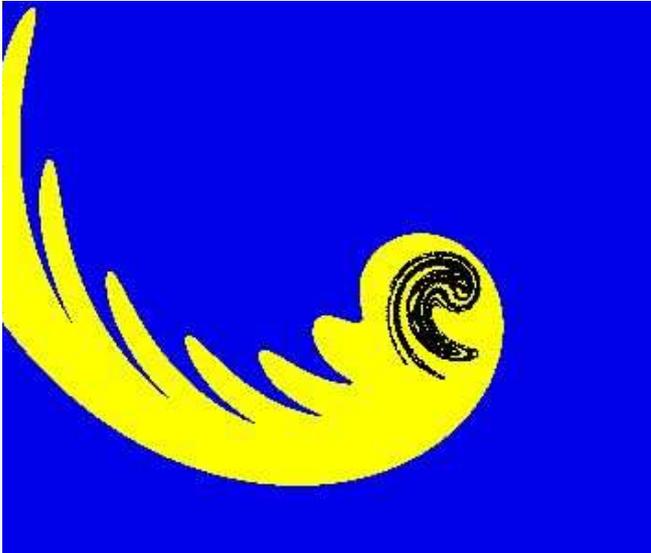
```

```

    if (fabs(x-oldx)<0.0001 && fabs(y-oldy)<0.0001) { putpixel(xe,ye,13); break;}
  }
  /* sinon, on considère que la trajectoire se colle sur l'attracteur chaotique et étrange */
  if (i ==500) putpixel(xe,ye,14);
}

```

On constate que le bassin de l'attracteur d'Ikeda a une forme en spirale (*figure 15*).



*Figure 15* : Bassin d'attraction de l'attracteur d'Ikeda en jaune, bassin d'attraction du point fixe en bleu.

#### 4.2. Les bassins d'attraction entremêlés<sup>6</sup>

Il existe aussi des bassins d'attraction complexes, complètement imbriqués les uns dans les autres, comme pour l'application qui s'écrit en nombres complexes complexes :

$$z' = z^2 - (1 + a i) \bar{z}, \text{ ou en réels :}$$

$$\begin{cases} x' = x^2 - y^2 - x - ay \\ y' = 2xy - ax = y \end{cases}$$

$$\text{avec } a = 1,0287$$

Sous l'effet de cette application, chaque point a une trajectoire qui soit s'en va à l'infini, soit converge vers trois attracteurs possibles en forme de morceaux de droites (*figure 16*). Le programme suivant donne l'exemple de trois trajectoires convergeant vers chacun de ces attracteurs.

<sup>6</sup> Appelé en anglais *riddled basins* (bassins criblés de trous). Cela signifie que tout point situé dans un bassin a dans son voisinage, si petit soit-il, un point d'un autre bassin. L'exemple ici traité est dû à J. Alexander en 1992.

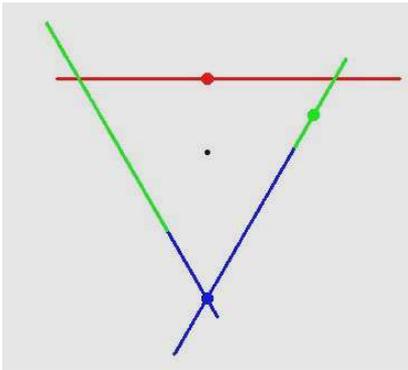


Figure 16 : Les trois attracteurs, en rouge, en bleu et en vert, l'ensemble ayant une forme de triangle équilatéral, dont l'origine du repère est le centre (en noir). Les trois gros points ajoutés, situés sur chacun des attracteurs, vont permettre de savoir sur quel attracteur on se trouve, lors de la convergence des trajectoires vers ces attracteurs.

```
a=1.0287137;
xx0=0.;yy0=0.5; trajectoire(0); /* trois trajectoires convergent vers chacun des attracteurs */
xx0=0.2;yy0=-0.5;trajectoire(1);
xx0=0.3;yy0=-0.5;trajectoire(2);
filldisc(xorig,yorig-zoom*a/2.,5,red);xx1=0;yy1=a/2.; /* trois points choisis sur chaque attracteur */
filldisc(xorig,yorig+a*zoom,5,coul[1]); xx2=0;yy2=-a;
filldisc(xorig+zoom*(5.*a/12.)*sqrt(3.),yorig-zoom*(a/4.),5,green);xx3=(5.*a/12.)*sqrt(3.);yy3=a/4.;
```

Fonction associée :

```
void trajectoire(int i) /* i est l'indice de la couleur attribuée à chacun des trois attracteurs */
{ int j;
  x=xx0,y=yy0; /* point initial, ces variables ayant été déclarées en global */
  for(j=1;j<11000;j++)
  { newx=x*x-y*y-x-a*y; newy=2.*x*y-a*x+y;
    x=newx;y=newy;
    if (x*x+y*y>10000.) break;
    if (j>8000) filldisc(xorig+zoom*x,yorig-zoom*y,1,coul[i]); /* coul[] tableau de trois couleurs */
  }
}
```

Connaissant trois points sur chacun des attracteurs, on a les trois bassins d'attraction correspondants en distinguant les points de départ des trajectoires selon qu'elles arrivent au voisinage de l'un de ces points. Le programme s'ensuit, avec son résultat sur la figure 17.

```
for(xx0=-2;xx0<2.5;xx0+=0.005) for(yy0=-2.4;yy0<2.;yy0+=0.005) /* parcours d'un rectangle */
trajectoire2();
```

Fonction associée :

```
void trajectoire2(void)
{ int j;
  x=xx0,y=yy0;
  for(j=1;j<1000;j++)
  { newx=x*x-y*y-x-a*y; newy=2.*x*y-a*x+y; x=newx;y=newy;
    if (x*x+y*y>10000.) { color=black; break;}
    if (j>500 && fabs(x-xx1)<0.05 && fabs(y-yy1)<0.05) { color=coul[0];break;}
    if (j>500 && fabs(x-xx2)<0.05 && fabs(y-yy2)<0.05) { color=coul[1];break;}
    if (j>500 && fabs(x-xx3)<0.05 && fabs(y-yy3)<0.05) { color=coul[2];break;}
    putpixel(xorig+zoom*xx0,yorig-zoom*yy0,color);
  }
}
```

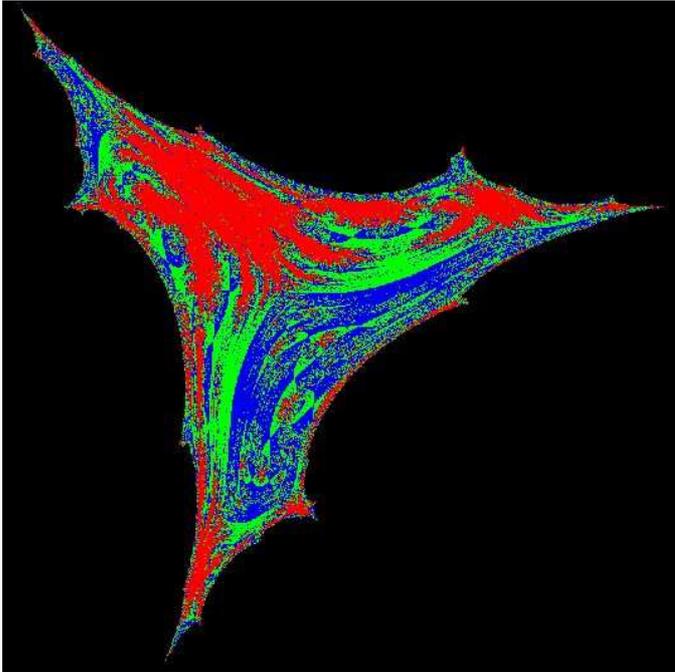


Figure 17 : Les trois bassins d'attraction imbriqués avec leurs trois couleurs respectives.

## 5. Symétrie et chaos

En 1992, M. Field et M. Golubitsky<sup>7</sup> ont proposé l'application  $F$  s'écrivant en complexes :

$$F(z) = (k + az\bar{z} + b\operatorname{Re}(z^n) + oi)z + c\bar{z}^{n-1}$$

Pour certaines valeurs bien choisies des paramètres  $a$ ,  $b$ ,  $o$  et  $c$ , l'action répétée de cette application fait converger les trajectoires vers un attracteur étrange. Plus précisément, lorsque  $o = 0$ , cet attracteur présente les  $2n$  symétries de ce que l'on appelle le groupe diédral  $D_{2n}$ , par exemple pour  $n = 4$  les huit symétries du cube. Par symétrie, l'on entend les isométries<sup>8</sup> qui laissent la figure invariante. Ainsi le carré possède quatre rotations qui le laissent invariant, ainsi que quatre réflexions. Si l'on prend  $o$  différent de 0, l'attracteur présente seulement les  $n$  symétries de rotation

M. Field et M. Golubitsky ont démontré que  $F(z)$  doit être de la forme précédemment indiquée, parmi les polynômes en  $z$  et  $\bar{z}$ , afin de donner des trajectoires symétriques dans  $D_{2n}$ . Dans l'exercice suivant, nous démontrons cette formule dans le cas particulier où  $n = 3$ .

### ***Exercice 7 : Attracteur ayant les symétries du triangle équilatéral***

*Une symétrie du triangle équilatéral  $ABC$  (figure 18 à gauche) étant une isométrie qui laisse globalement invariant le triangle, il existe exactement 6 symétries de ce genre. En notant  $O$  le centre de gravité du triangle (et l'origine du repère), il s'agit de l'identité  $I$ , de la rotation  $R$  de centre  $O$  et d'angle  $2\pi/3$ , de la rotation  $R^2$  de centre  $O$  et d'angle  $4\pi/3$ , et des trois réflexions autour des axes passant par  $O$  et par l'un des sommets du triangle. Appelons  $T$  la réflexion autour de  $(OA)$ . Les deux autres sont  $TR$  et  $TR^2$ . Ces six isométries forment le groupe diédral  $D_6$ . Les trois rotations  $I$ ,  $R$ ,  $R^2$*

<sup>7</sup> voir le livre Symétrie et chaos, de Michael Field et Martin Golubitsky (Oxford University Press, 1992).

<sup>8</sup> Par définition une isométrie est une transformation conservant les distances. Elle a comme propriétés de conserver les angles non orientés, les barycentres, la forme des figures. En deux dimensions, les seules isométries sont les translations, les rotations, les réflexions autour d'un axe, et les réflexions-glissemens (ou *loopings*, c'est-à-dire réflexions suivies d'une translation parallèle à l'axe de la réflexion).

forment un sous-groupe noté  $Z_3$  (isomorphe à  $Z/3Z$  muni de l'addition). Le groupe  $Z_3$  est engendré par  $R$ , le groupe  $D_6$  est engendré par  $R$  et  $T$ , toutes les symétries étant obtenues par composition de  $R$  et  $T$ .

L'objectif est de trouver des trajectoires de points ayant ces symétries sous l'action répétée d'une fonction  $F$ . Il suffit de prendre les deux symétries  $T$  et  $R$ , et d'imposer que

$$F(T(z)) = T(F(z)) \text{ et } F(R(z)) = R(F(z)).$$

En effet sous l'effet de  $F$ , un point  $M(z)$  devient  $M'$  d'affixe  $F(z)$ , et le symétrique  $M_1$  de  $M$  par  $T$ , d'affixe  $T(z)$ , devient  $M'_1$  d'affixe  $F(T(z))$ , d'où la nécessité  $T(F(z))=F(T(z))$  (figure 18). De même avec  $R$ .

Ce qui vaut pour les deux symétries  $R$  et  $T$ , convient aussi pour les autres symétries car la fonction  $F$  commute avec  $R$  et  $T$ . Par exemple  $F(TR(z)) = F(T(R(z))) = T(F(R(z))) = T(R(F(z))) = TR(F(z))$ . Les trajectoires qui en découlent à partir de deux points symétriques vont conserver cette symétrie, et pour peu qu'elles convergent vers un attracteur, celui-ci héritera de cette symétrie.

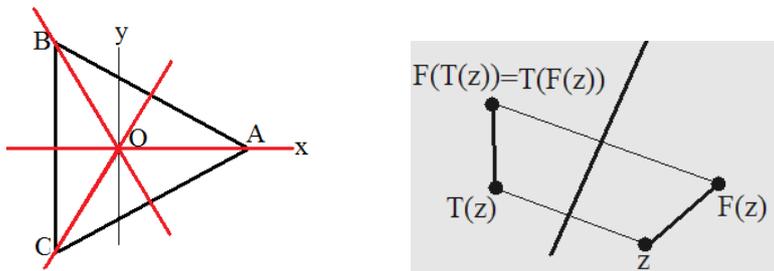


Figure 18 : Triangle équilatéral  $ABC$  à gauche, et à droite deux points symétriques avec conservation de la symétrie pour les trajectoires.

1) Montrer que  $F(z) = \bar{z}^2$  convient (et non pas  $F(z) = z^2$ ). Puis programmer : partir d'un point  $M$  de module inférieur à 1, prendre ses six symétriques, et constater que les six trajectoires à partir de ces points sont symétriques. A cause de leur convergence rapide vers  $O$ , le résultat demeure rudimentaire.

Constatons d'abord que  $T(z) = \bar{z}$  et que  $R(z) = \exp(i 2\pi / 3) z$ . Puis vérifions que  $FT = TF$  et  $FR = RF$  :

$$\begin{aligned} z \xrightarrow{T} \bar{z} \xrightarrow{F} \bar{z}^2 = z^2 & \quad z \xrightarrow{R} e^{i2\pi/3} z \xrightarrow{F} (e^{-i2\pi/3} \bar{z})^2 = e^{i2\pi/3} \bar{z}^2 \\ z \xrightarrow{F} \bar{z}^2 \xrightarrow{T} \overline{\bar{z}^2} = z^2 & \quad z \xrightarrow{F} \bar{z}^2 \xrightarrow{R} e^{i2\pi/3} \bar{z}^2 \end{aligned}$$

Cela se vérifie aussi par programme (figure 19).

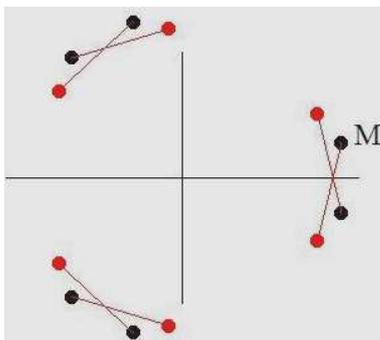


Figure 19 : Un point  $M$  et ses symétriques en noir, avec leurs transformés en rouge, eux aussi présentant les mêmes symétries.

2) Montrer que  $F(z) = \operatorname{Re}(z^3) z$  convient (et non pas  $F(z) = \operatorname{Re}(z) z$  ni  $F(z) = \operatorname{Re}(z^2) z$  ni  $F(z) = \operatorname{Im}(z^3) z$ ). Ce résultat est lui aussi rudimentaire. A partir de d'un point  $M$  de module inférieur à 1, la trajectoire tend vers  $O$  en ligne droite.

$$\begin{array}{l} z \xrightarrow{T} \bar{z} \xrightarrow{F} \operatorname{Re}(z^3) \bar{z} \\ z \xrightarrow{F} \operatorname{Re}(z^3) z \xrightarrow{T} \operatorname{Re}(z^3) \bar{z} \end{array} \quad \text{et} \quad \begin{array}{l} z \xrightarrow{R} e^{i2\pi/3} z \xrightarrow{F} \operatorname{Re}(z^3) e^{i2\pi/3} z \\ z \xrightarrow{F} \operatorname{Re}(z^3) z \xrightarrow{R} \operatorname{Re}(z^3) e^{i2\pi/3} z \end{array}$$

On a bien  $FT = TF$  et  $FR = RF$ .

3) Montrer que  $F(z) = f(z\bar{z}) z$  convient, où  $f$  est une fonction quelconque de  $z\bar{z}$ . Comme  $z\bar{z}$  est le module de  $z$  au carré,  $M$  est transformé en  $M'$  avec  $O, M, M'$  alignés, et la transformation accepte toutes les isométries laissant le point  $O$  invariant. Pourquoi prendre une classe aussi vaste de transformations? Parce qu'elle va provoquer une dispersion et une sortie des points du voisinage de l'origine  $O$ , là où les entraînaient les autres transformations. C'est elle qui va ajouter la pincée de chaos indispensable pour avoir des attracteurs étranges. Pour simplifier on choisira ensuite :

$$f(z\bar{z}) = k + a z\bar{z}.$$

$$\begin{array}{l} z \xrightarrow{T} \bar{z} \xrightarrow{F} f(z\bar{z}) \bar{z} \\ z \xrightarrow{F} f(z\bar{z}) z \xrightarrow{T} f(z\bar{z}) \bar{z} \end{array} \quad \text{et} \quad \begin{array}{l} z \xrightarrow{R} e^{i2\pi/3} z \xrightarrow{F} f(z\bar{z}) e^{i2\pi/3} z \\ z \xrightarrow{F} f(z\bar{z}) z \xrightarrow{R} f(z\bar{z}) e^{i2\pi/3} z \end{array}$$

4) Prendre finalement  $F(z) = (k + a z\bar{z} + b \operatorname{Re}(z^3)) z + c \bar{z}^2$ . Vérifier que toute combinaison linéaire des solutions précédemment trouvées est aussi solution,  $a, b, c$  étant réels.

Le fait de multiplier l'une quelconque des fonctions précédentes par un nombre réel préserve les égalités  $TF = FT$  et  $RF = FR$ .

Prenons maintenant deux solutions précédemment trouvées pour  $F$ , soit  $F_1$  et  $F_2$ , vérifiant  $F_1 T(z) = T F_1(z)$  et  $F_2 T(z) = T F_2(z)$ , et de même avec  $R$ . Montrons que leur somme vérifie aussi ces relations.

$$T(F_1+F_2)(z) = T((F_1+F_2)(z)) = T(F_1(z)+F_2(z)) = \overline{F_1(z)+F_2(z)} = \overline{F_1(z)} + \overline{F_2(z)}$$

$$(F_1+F_2) T(z) = (F_1+F_2)(T(z)) = (F_1+F_2)(\bar{z}) = F_1(\bar{z}) + F_2(\bar{z})$$

Il ne reste plus qu'à vérifier que pour chacune des fonctions  $F$  trouvées précédemment, on a bien  $F(z) = \overline{F(\bar{z})}$ , ce que l'on constate aisément.

5) Programmer l'exemple où  $k = 1,56, a = -1, b = 0,1, c = -0,8$ . Si l'on ne dispose pas de bibliothèque complexe, on convertira  $z' = F(z)$  en réels, soit :

$$\begin{cases} x' = (k + a(x^2 + y^2) + b(x^3 - 3xy^2))x + c(x^2 - y^2) \\ y' = (k + a(x^2 + y^2) + b(x^3 - 3xy^2))y - 2cxy \end{cases}$$

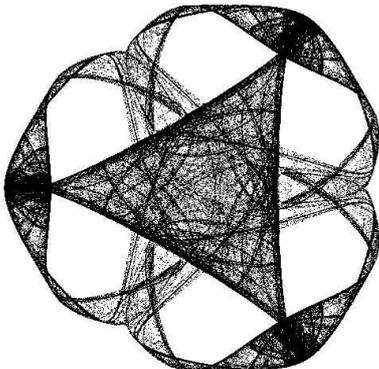


Figure 20 : Attracteur possédant les symétries du triangle équilatéral.

6) Montrer que  $G(z) = oiz$  avec  $o$  nombre réel est une similitude de centre  $O$  et d'angle droit., et qu'elle commute avec les rotations de centre  $O$  mais pas avec les réflexions. Le fait d'ajouter  $G(z)$  à  $F(z)$  fait que les trajectoires sont symétriques pour les rotations seulement.

$z \mapsto iz$  est une rotation de centre  $O$  et d'angle  $\pi/2$ . Le fait de multiplier par  $o$  donne une similitude de rapport  $|o|$  et d'angle  $\pi/2$  si  $o > 0$  ou  $-\pi/2$  si  $o < 0$ .

Lorsqu'elles ont le même centre, deux similitudes commutent, en particulier si l'une est une rotation. Ainsi  $GR(z) = RG(z)$ . Par contre :

$$\begin{aligned} z &\xrightarrow{G} ioz \xrightarrow{T} -io\bar{z} \\ z &\xrightarrow{T} \bar{z} \xrightarrow{G} io\bar{z} \end{aligned} \quad \text{et } GT(z) \neq TG(z)$$

Ce que nous avons fait pour  $n = 3$  se généralise à  $n$  quelconque. Par exemple, pour  $n = 4$ , la fonction  $F$  s'écrit en réels :

$$\begin{cases} x' = qx + c(x^3 - 3xy^2) - oy \\ y' = qy + c(y^3 - 3x^2y) + ox \end{cases} \quad \text{avec } q = k + a(x^2 + y^2) + b(x^4 - 6x^2y^2 + y^4)$$

Selon que l'on prend  $c$  nul ou pas, l'attracteur a les huit symétries du carré ou seulement les quatre symétries de rotation. Le programme suivant permet de le constater (figure 21) :

```
a = 2.; b = 0.; c = 1.;
o=0.; /* prendre c entre -0.2 et 0.2 */
for(k=-1.91;k<-1.88;k+=0.01) /* plusieurs valeurs de k pour voir l'évolution de l'attracteur */
{ x=0.1;y=0.5; /* point de départ de la trajectoire */
  for(i=0;i<1100000;i++)
  { q=k+a*(x*x+y*y)+b*( x*x*x*x-6.*x*x*y*y+y*y*y*y); /*fonction F */
    newx=q*x+c*( x*x*x-3.*x*y*y)-o*y;
    newy=q*y+c*(-3.*x*x*y+y*y*y)+o*x;
    x=newx;y=newy;
    if (i>900000) putpixel(xorig+zoom*x,yorig-zoom*y,black);
  }
}
```

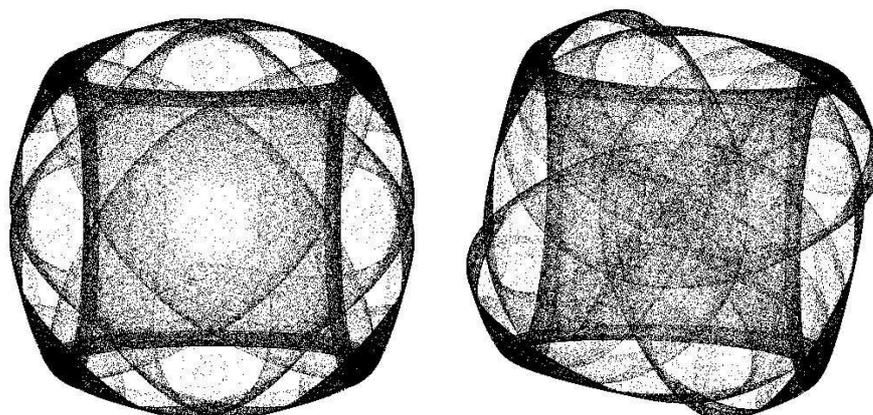


Figure 21 : Attracteur présentant les huit symétries du carré ( $o = 0$ ) à gauche, ou seulement les quatre symétries de rotation ( $o = 0,1$ ) à droite. Les paramètres sont  $k = -1,85$ ,  $a = 2$ ,  $b = 0$ ,  $c = 1$ .

On peut aussi observer l'évolution de l'attracteur lorsque l'on fait varier un paramètre. Par exemple pour  $n = 3$ , avec  $a = 2$ ,  $b = 0$ ,  $c = -0,5$ ;  $o = 0$ , et en faisant varier  $k$  de 1.7 à -2.4. On voit d'abord apparaître trois cycles de deux points (il faudra prendre plusieurs trajectoires pour les trouver), puis un phénomène de bifurcations se développe, jusqu'à l'apparition de trois attracteurs

étranges distincts, qui finalement fusionnent pour n'en donner qu'un (figure 22). Dans le cas des trois attracteurs, la figure 23 indique comment sont disposés leurs bassins d'attraction.

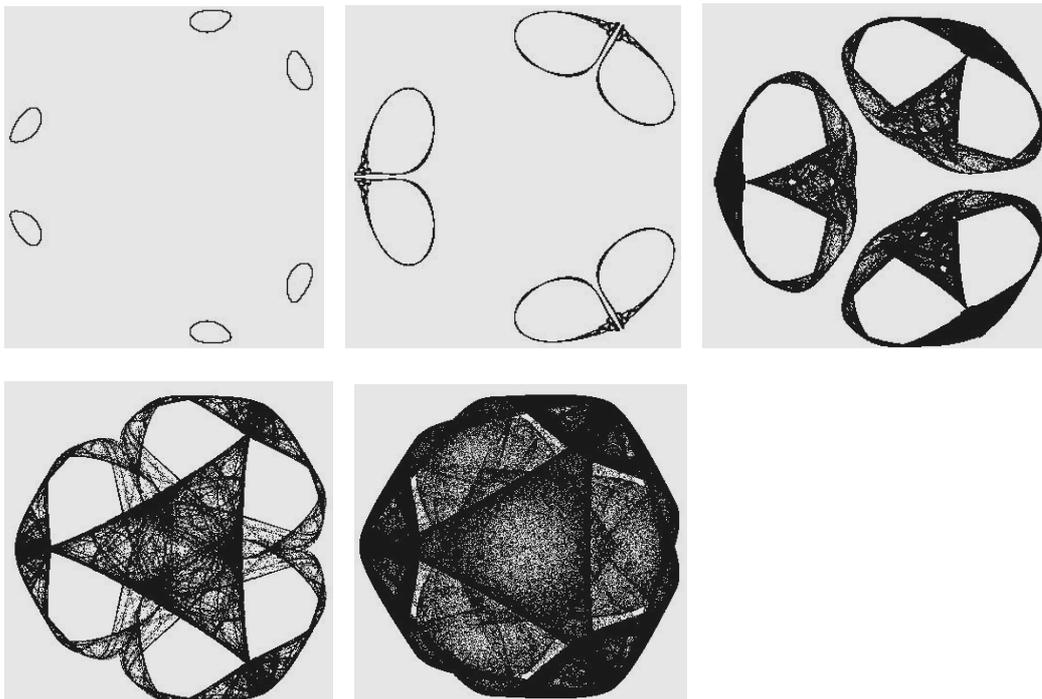


Figure 22 : Evolution de l'attracteur lorsqu'un paramètre varie.

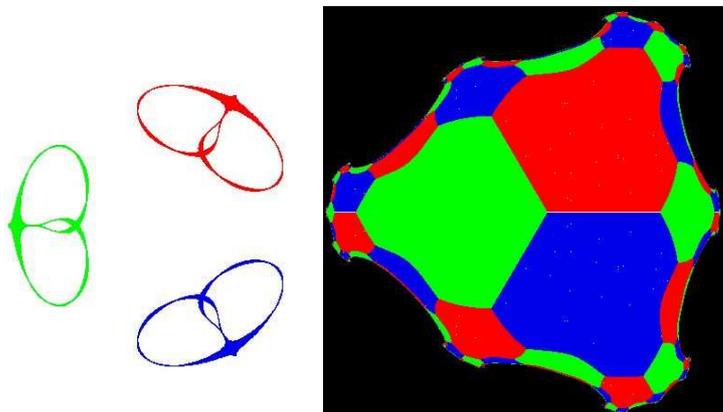


Figure 23 : Cycles de trois attracteurs et leurs bassins d'attraction.

### Exercice 8 : Bassin d'attraction présentant des symétries

Découper le plan en six secteurs angulaires autour de  $O$ , en les coloriant de façon différente (rouge, vert, bleu, ... mais pas noir). Puis reprendre la fonction  $F(z)$  pour  $n = 3$  avec  $k = 1,43$ ,  $a = -1$ ,  $b = 0,1$ ,  $c = -0,8$ ,  $o = 0$ .

Pour chaque point  $z_0$  d'un carré d'écran autour de  $O$ , pratiquer  $z_1 = F(z_0)$ , et colorier le point de départ  $z_0$  avec la couleur de la zone dans laquelle tombe  $z_1$ . Continuer en prenant  $z_2 = F^2(z_0)$ , ou encore  $z_2 = F(z_1)$ , et colorier le point de départ  $z_0$  avec la couleur de la zone où tombe  $z_2$ . Poursuivre de la même façon en prenant  $F^3, F^4 \dots$ . S'arranger pour colorier en noir les points dont l'image par  $F^k$  s'en va hors de certaines limites (la trajectoire ira à l'infini si on la continuait). La zone des points de départ coloriés avec les six couleurs correspond finalement au bassin d'attraction de l'attracteur, et elle présente les mêmes symétries.

Le programme suit, ainsi que ses résultats (figure 24).

```

L=390; zoom=(float)L/3.; xorig=L/2.; yorig=L/2.; /* carré de côté L, en haut à gauche de l'écran */
for(xe=0;xe<=L;xe++) for(ye=0;ye<=L;ye++) /* parcours du carré */
{ x=(xe-xorig)/zoom;y=(yorig-ye)/zoom;
  if (x>=0 && y>=0 && y<x) putpixel(xe,ye,red); /* coloriage initial avec six couleurs */
  else if (x>=0 && y >=0 && y>=x)putpixel(xe,ye,blue);
  else if (x<0 && y >=0 && y>=-x)putpixel(xe,ye,green);
  else if (x<0 && y >=0 && y<-x)putpixel(xe,ye,yellow);
  else if (x>=0 && y<0 && y<-x) putpixel(xe,ye,greenn);
  else if (x>=0 && y <0 && y>-x)putpixel(xe,ye,yelloww);
  else if (x<0 && y <0 && y>x)putpixel(xe,ye,redd);
  else if (x<0 && y <0 && y<x)putpixel(xe,ye,bluee);
}
SDL_Flip(screen);pause();

for(etape=1;etape<7;etape++) /* transformés successifs des points du carré */
{ for(xe=0;xe<=L;xe++) for(ye=0;ye<=L;ye++)
  { x=(xe-xorig)/zoom;y=(yorig-ye)/zoom;
    newx=FX(x,y); newy=FY(x,y); /* fonctions donnant x' et y' en fonction de x et y */
    newxe=xorig+zoom*newx;newye=yorig-zoom*newy;
    if (newxe>0 && newxe<L && newye>0 && newye<L)
      { color=getpixel(xorig+zoom*newx,yorig-zoom*newy);
        putpixel(xe+L+10,ye,color); /* dessin fait dans un autre carré situé à droite du carré initial */
      }
    else putpixel(xe+L+10,ye,black); /* en noir les points dont la trajectoire s'en va à 'l'infini */
  }
  for(xe=0;xe<=L;xe++) for(ye=0;ye<=L;ye++) /* renvoi du carré des transformés dans le carré initial */
  { color=getpixel(xe+L+10,ye); putpixel(xe,ye,color); }
  for(xe=0;xe<=L;xe++) for(ye=0;ye<=L;ye++) putpixel(xe+L+10,ye,white);
  SDL_Flip(screen);pause();
}

float FX(float x,float y) { q=k+a*(x*x+y*y)+b*(x*x*x-3.*x*y*y); return (q*x+c*(x*x-y*y)-o*y); }
float FY(float x,float y) { q=k+a*(x*x+y*y)+b*(x*x*x-3.*x*y*y); return (q*y-2.*c*x*y+o*x); }

```

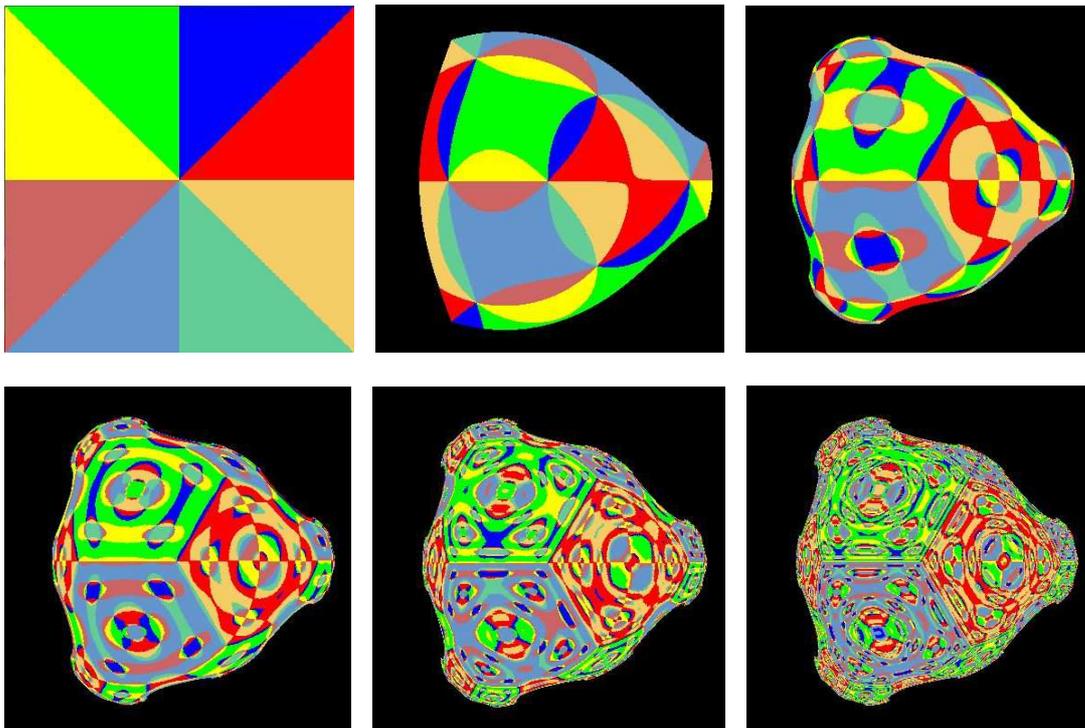


Figure 24 : Evolution colorée vers l'attracteur.

## 6. Les diverses formes d'attracteurs en deux dimensions

Commençons par le cas simple des applications linéaires, de la forme

$$\begin{cases} x' = a_{11}x + a_{12}y \\ y' = a_{21}x + a_{22}y \end{cases} \text{ ou encore sous forme matricielle } \mathbf{x}' = M \mathbf{x}.$$

Elles admettent l'origine  $O$  du repère  $x, y$  comme point fixe. Ce point fixe peut être stable ou instable. Quand il est stable, il s'agit soit d'un puits, les trajectoires allant vers lui en ligne droite ou en spirale, soit d'un point elliptique (ou centre, ou tourbillon), les trajectoires tournant autour de lui. Quand il est instable, il s'agit soit d'une source, toutes les trajectoires s'en éloignant en ligne droite ou en spirale, soit d'un point hyperbolique (ou selle, ou col), où seuls les points situés sur l'une des deux asymptotes ont des trajectoires qui convergent vers lui, les autres s'en éloignant irrémédiablement.

Pour une application linéaire, on montre que le point  $O$  est un puits lorsque les valeurs propres de la matrice  $M$  sont inférieures à 1 en valeur absolue, qu'il s'agit d'une source lorsque les valeurs propres sont supérieures à 1 en valeur absolue, et qu'il s'agit d'une selle lorsqu'une valeur propre est supérieure à 1 en valeur absolue et l'autre inférieure à 1 en valeur absolue. Cela est vrai aussi bien pour des valeurs propres réelles que complexes (la valeur absolue étant alors le module).

Un point fixe est dit attracteur lorsque tout point du voisinage a sa trajectoire qui converge vers lui. Il est dit repousseur lorsque tout point du voisinage a une trajectoire qui sort de ce voisinage. Pour les applications linéaires, les seuls points attracteurs sont les puits, et les seuls points repousseurs sont les sources, les cas des tourbillons et des selles étant différents.

### Exercice 9 : Point hyperbolique d'une application linéaire

Prendre  $\begin{cases} x' = 1,5x + 0,5y \\ y' = x + y \end{cases}$ . Vérifier que les valeurs propres sont 2 et 0.5, et que des vecteurs

propres associés sont  $(1, 1)$  et  $(-0,5, 1)$ . Puis programmer le tracé de plusieurs trajectoires de points situés à proximité, en joignant les points successifs de chaque trajectoire par une ligne.

Les valeurs propres  $\lambda$  vérifient  $\begin{vmatrix} 1,5 - \lambda & 0,5 \\ 1 & 1 - \lambda \end{vmatrix} = 0$  ou  $\lambda^2 - (5/2)\lambda + 1 = 0$ , ce qui donne  $\lambda = 2$  ou  $1/2$ .

Les vecteurs propres associés à 2 vérifient  $x - y = 0$ , d'où  $(1, 1)$  notamment, ceux associés à  $1/2$  vérifient  $x + 0,5y = 0$ , d'où  $(-0,5, 1)$ . Les deux vecteurs propres forment les asymptotes du point asymptotique  $O$ . Pour le tracé des trajectoires, on a choisi un carré de centre  $O$  et de côté 10. Les points de départ sont pris sur certaines parties de la bordure du carré (figure 25).

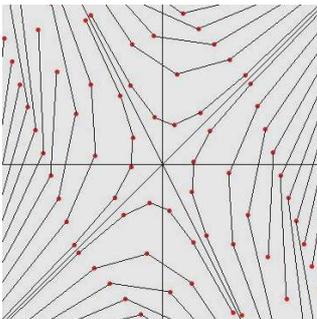


Figure 25 : Quelques trajectoires autour du point hyperbolique.

Les résultats précédents à propos des applications linéaires se généralisent aux points fixes éventuels des applications non-linéaires  $f$ . Il suffit de faire une approximation linéaire au voisinage du point fixe, en prenant ce que l'on appelle la matrice jacobienne. Il en est de même pour les cycles fixes de  $k$  points, car ces  $k$  points sont des points fixes pour l'application  $f^k$ .

Mais quels sont les divers types d'attracteurs ? En deux dimensions, un attracteur peut être un point source, ou un cycle de points sources, mais il peut aussi être un attracteur étrange et chaotique, à l'instar de la courbe pliée et feuilletée de la transformation de Hénon.