

# I. Relations de récurrence et évolutions de populations. De l'ordre au chaos

Une suite de nombres s'écrit  $u_0, u_1, u_2, u_3, \dots$ , où chaque nombre est numéroté, c'est-à-dire indexé par un entier naturel. Elle s'écrit aussi  $(u_n)$  sous forme abrégée. Elle peut être fabriquée à partir d'une relation de récurrence liant chaque terme à celui ou à ceux qui le précèdent, avec en plus des conditions initiales données. Par exemple la suite  $(u_n)$  obéissant à la relation de récurrence  $u_{n+1} = 2 u_n$ , avec en condition initiale  $u_0 = 1$ , est parfaitement déterminée, et l'on aboutit à la succession de nombres 1, 2, 4, 8, 16, etc. Une telle suite tend manifestement vers l'infini. Mais il existe d'autres suites qui admettent une limite finie lorsque  $n$  tend vers l'infini, ou qui finissent par osciller sur deux ou plusieurs valeurs. Ce sont là des cas classiques. Nous allons voir qu'en compliquant légèrement la relation de récurrence, l'évolution prend soudain une allure désordonnée, d'apparence imprévisible.

## 1. Le contexte général des évolutions de populations

Considérons une certaine population dont le nombre est  $u_n$  à l'étape de temps  $n$ , une étape de temps pouvant s'étendre sur un an, ou une saison, ou une seconde selon les cas. La connaissance de la population au départ, soit  $u_0$ , ainsi que d'une relation de récurrence liant la population à un certain moment et ce qu'elle était avant, permet de déterminer  $u_n$  quel que soit  $n$ , autrement de prévoir son évolution à long terme.

Le cas le plus simple est celui d'une relation de récurrence linéaire de la forme  $u_{n+1} = k u_n$ , avec  $u_0$  donné. Le taux de variation est constant dans le temps. Par exemple une population qui croît de 10% par an obéit à la récurrence  $u_{n+1} = 1,1 u_n$ <sup>1</sup> (figure 1). Comme  $k$  est supérieur à 1, cette évolution classique est du type exponentiel, avec  $u_n = u_0 k^n$ <sup>2</sup>. L'augmentation est de plus en plus grande et possède finalement un caractère relativement explosif. Une autre évolution exponentielle est celle de la suite de Fibonacci, que nous allons voir.

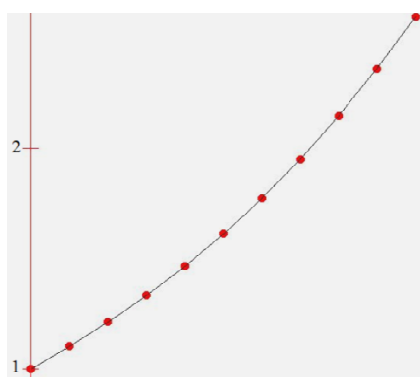


Figure 1 : Evolution exponentielle de la suite vérifiant  $u_{n+1} = 1,1 u_n$  avec  $u_0 = 1$ , représentée par les points d'abscisse  $n$  et d'ordonnée  $u_n$ , ici de  $n = 0$  à  $n = 10$ .

<sup>1</sup> 10% signifie  $10 / 100 = 0,1$ . Une augmentation de 10% par an signifie que la population augmente de  $0,1 u_n$  d'une année à la suivante, soit  $u_{n+1} = u_n + 0,1 u_n = 1,1 u_n$ .

<sup>2</sup> On a  $u_n = k u_{n-1}$ , puis  $u_{n-1} = k u_{n-2}$ , et ainsi de suite jusqu'à  $u_1 = k u_0$ , ce qui donne après simplifications  $u_n = u_0 k^n$ .

## 2. Les lapins de Fibonacci

Commençons par la suite de Fibonacci.<sup>3</sup> Celle-ci obéit à la relation de récurrence  $u_{n+1} = u_n + u_{n-1}$ , avec au départ  $u_0 = 0$  et  $u_1 = 1$ . Chaque terme est la somme des deux termes qui le précèdent. Ainsi définie, la suite est parfaitement déterminée. On obtient : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, etc.<sup>4</sup>

### Programmation

Pour calculer les termes successifs de la suite, nous avons besoin de trois cases mémoires, notées  $u$ ,  $v$ ,  $w$ . La suite peut être représentée graphiquement par des points  $(n, u_n)$  que nous joignons par des segments (figure 2).

```
u=0.; v=1.; /* conditions initiales : on met u_0 dans u et u_1 dans v */
filldisc(xorig,yorig-zoomv*u,2,red);filldisc(xorig+zoomh,yorig-zoomv*v,2,red);
line( xorig,yorig-zoomv*u,xorig+zoomh,yorig-zoomv*v,black); /* dessin des deux points et jonction */
for(i=2;i<14;i++)
{ w=v+u; /* terme w qui est la somme des deux qui le précèdent */
  filldisc(xorig+zoomh*i,yorig-zoomv*w,3,red);
  line( xorig+zoomh*(i-1),yorig-zoomv*v,xorig+zoomh*i,yorig-zoomv*w,black); /* dessin */
  u=v; /* actualisation avant de passer à l'étape suivante */
  v=w;
}
```

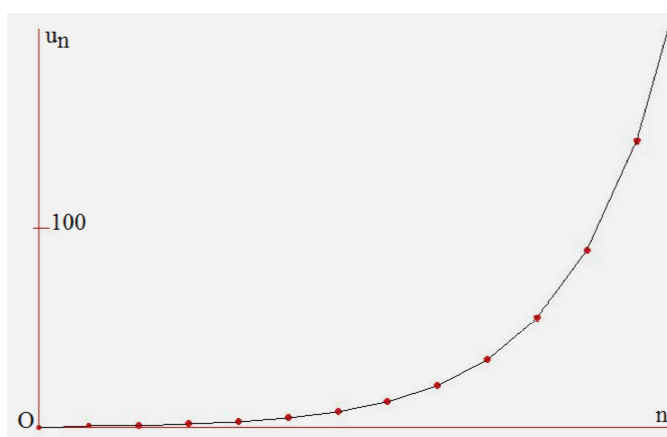


Figure 2 : Premiers termes de la suite de Fibonacci.

<sup>3</sup> Fibonacci, ou Léonard de Pise, dont le père résidait à Béjaia (Bougie, en Algérie), est un mathématicien qui fut un passeur des mathématiques arabes vers l'Italie et l'Europe, vers 1200. C'est en partie grâce à lui qu'a été introduite la numération des chiffres arabes, qui remplaça avantageusement, c'est le moins qu'on puisse dire, les chiffres latins. Pourtant, ce grand mathématicien était souvent considéré comme un traducteur ou un commentateur de textes, voire un plaisantin avec ses problèmes de lapins, jusque dans les années 1970, grâce à nos « grands » mathématiciens français. C'est lui qui a posé, dans un de ses livres ce problème, qui est à l'origine de la suite qui porte son nom : « Combien de couples de lapins obtiendrait-on à la fin de l'année si, commençant avec un couple, chacun des couples produisait chaque mois un nouveau couple, et que celui-ci pourrait à son tour procréer au second mois de son existence ? » Dernier outrage : au lieu de « second mois », c'est « troisième mois » qui est indiqué, par exemple dans *Wikipedia* (en 2015) ou *Wolfram Institute*, mais dans ce cas de troisième mois, on n'aurait plus la suite de Fibonacci !

<sup>4</sup> On a les points de repère suivants :  $u_5 = 5$  et  $u_{12} = 144 = 12^2$ .

### Exercice 1 : Forme explicite de $u_n$ dans la suite de Fibonacci

Il s'agit de prouver que  $u_n = \frac{1}{\sqrt{5}} \left( \left( \frac{1+\sqrt{5}}{2} \right)^n - \left( \frac{1-\sqrt{5}}{2} \right)^n \right)$

1) Prendre la relation de récurrence  $u_{n+1} = u_n + u_{n-1}$ . Il existe une infinité de suites vérifiant cette relation, selon les conditions initiales que l'on prend. Parmi toutes ces suites, trouver deux suites où  $u_n$  est de la forme  $r^n$  avec  $r$  différent de 0.

Imposons que  $r^n$  vérifie la relation de récurrence :  $r^{n+1} = r^n + r^{n-1}$ , et comme  $r \neq 0$ , cela se réduit à :  $r^2 - r - 1 = 0$ . Cette équation admet deux solutions  $r = \frac{1 \pm \sqrt{5}}{2}$ . Nous venons de trouver deux suites vérifiant la relation de récurrence.

2) On appelle  $r_1$  et  $r_2$  les deux valeurs de  $r$  trouvées. Montrer qu'il existe une infinité de suites de la forme  $K_1 r_1^n + K_2 r_2^n$  vérifiant la relation de récurrence,  $K_1$  et  $K_2$  étant des constantes quelconques.

Il suffit de former  $u_{n+1} - u_n - u_{n-1}$  en remplaçant  $u_n$  par  $K_1 r_1^n + K_2 r_2^n$  et de constater que le résultat est bien 0.

3) Parmi l'infinité de suites obtenues, trouver la suite de Fibonacci.

Il suffit d'imposer que  $u_0 = 0$  et  $u_1 = 1$ , ce qui donne un système dont les inconnues sont  $K_1$  et  $K_2$  :

$$\begin{cases} K_1 + K_2 = 0 \\ K_1 r_1 + K_2 r_2 = 1 \end{cases}$$

Ce système admet une solution unique<sup>5</sup> :  $K_1 = 1/\sqrt{5}, K_2 = -1/\sqrt{5}$ . D'où  $u_n = \frac{1}{\sqrt{5}} (r_1^n - r_2^n)$ .

Mais quel est le lien entre la suite de Fibonacci et l'évolution d'une population de lapins, c'est-à-dire d'une population qui est censée proliférer ? On utilise pour cela une modélisation simplifiée, où les lapins sont considérés comme éternels : à chaque étape de temps, que nous appellerons une saison, un couple de lapins nouveaux-nés passe du stade enfant au stade adulte, et un couple adulte reste adulte tout en donnant naissance à un couple enfant. En notant 0 un couple enfant, et 1 un couple adulte, ces deux règles sont appliquées d'une saison à la suivante :

0 → 1 passage d'enfant à adulte  
1 → 10 couple adulte restant adulte et qui donne naissance à un couple enfant

Cela donne l'évolution suivante, saison après saison, à partir d'un couple enfant (figure 3):

1 → 10 → 101 → 10110 → 10110101 → ...

Il en ressort que le nombre de couples, noté  $v_n$  à l'étape de temps  $n$ , commence par  $v_0 = 1, v_1 = 2, v_2 = 3, v_4 = 5, v_5 = 8$ .<sup>6</sup> Cela correspond à la suite de Fibonacci  $u_n$ , mais décalée d'un cran, à partir de

<sup>5</sup> Quelles que soient les conditions initiales choisies, on trouvera toujours une solution unique. Cela prouve qu'il n'y a pas d'autres solutions à la relation de récurrence que celles de la forme  $K_1 r_1^n + K_2 r_2^n$ . Précisons qu'avec  $u_0$  et  $u_1$  donnés, on trouve  $K_1 = (u_1 - u_0 r_2) / (r_1 - r_2)$  et  $K_2 = (u_0 r_1 - u_1) / (r_1 - r_2)$

<sup>6</sup> Il s'agit de la longueur des mots formés de 0 et de 1. Mais la construction de ces mots donne plus d'indications que leur simple longueur. Ils contiennent un ordre interne, où les couples-enfants notés 0 sont placés juste après ceux qui les ont procréés.

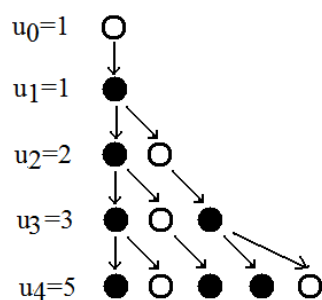


Figure 3 : Evolution des couples de lapins, avec *en blanc* un couple enfant et *en noir* un couple adulte.

$v_0 = u_1 = 1$ . Pour le démontrer, appelons  $a_n$  le nombre de couples enfants et  $b_n$  le nombre de couples adultes à l'étape  $n$ . Les règles d'évolution imposent que :

$$a_{n+1} = b_n, \text{ et } b_{n+1} = a_n + b_n.$$

Il s'ensuit que  $b_{n+1} = b_n + b_{n-1}$  avec  $b_0 = 0$  et  $b_1 = 1$ . C'est exactement la suite de Fibonacci ( $u_n$ ), soit  $b_n = u_n$ . D'autre part,  $a_{n+1} = b_n = u_n$ , c'est la suite de Fibonacci décalée d'un cran. Finalement :

$v_n = a_n + b_n = u_n + u_{n-1} = u_{n+1}$ . On retrouve bien la suite de Fibonacci décalée d'un cran.

### Exercice 2 : Simulation d'un élevage de lapins

Une personne achète 10 couples de lapins nouveaux-nés, dans le but d'en faire l'élevage, et de mesurer la rentabilité de l'opération. Le scénario d'évolution est le suivant : les lapins nouveaux-nés ont l'âge 0, puis ils passent à l'âge 1 au bout d'un saison, puis à l'âge 2 la saison suivante, période durant laquelle ils peuvent procréer, en donnant naissance à un nouveau couple d'âge 0, puis à l'âge 3 la saison suivante, où ils donnent aussi naissance à un nouveau couple, et au-delà ils sont vendus. Précisons qu'il s'agit d'une évolution de style Fibonacci, mais plus réaliste, puisque les lapins les plus âgés sont vendus. L'éleveur est en mesure de construire un clapier où il pourra mettre environ 3000 couples de lapins. Il désire savoir combien de saisons seront nécessaires pour remplir son clapier, et combien il aura alors vendu de lapins. Pour cela nous allons pratiquer une simulation sur ordinateur.

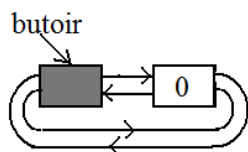
1) Au lieu de partir de 10 couples, on va partir d'un seul couple, il suffira à la fin de multiplier les résultats par 10. Ecrire l'évolution de la population des lapins de la saison 0 jusqu'à la saison 6, chaque couple étant identifié par son âge. On écrira cela comme dans l'évolution des lapins de Fibonacci, en plaçant les enfants juste à droite de leurs parents.

étape      couples de lapins nommés selon leur âge

0	0
1	1
2	2 0
3	3 0 1
4	1 2 0 et un couple enlevé
5	2 0 3 0 1
6	3 0 1 1 2 0 et un couple enlevé

2) Pour le programme, on va utiliser une liste doublement chaînée circulaire, où chaque cellule représente un couple de lapins. Comme des suppressions de cellules vont arriver, pour les couple ayant atteint l'âge 3, on ajoute une cellule butoir factice, qui ne pourra jamais être supprimée, où vont s'accrocher les cellules-lapins. Ecrire la structure de base de la cellule représentant un couple de lapins, puis les conditions initiales, avec le couple d'âge 0, accroché à la cellule butoir.

La cellule de base, appelée *lapin*, a une structure contenant trois composantes : l'âge, un pointeur vers la cellule suivante, et un pointeur vers la cellule précédente. Les conditions initiales correspondent au dessin de la *figure 4* :



*Figure 4* : Mise en place de la liste doublement chaînée circulaire initiale, à l'étape 0, avec la cellule factice *butoir*, et la cellule correspondant au couple de nouveaux-nés.

3) Après la construction de la liste chaînée initiale, faire la boucle des étapes. A chaque étape de temps, un pointeur courant *ptr* va faire le tour de la boucle. Au cours d'un premier tour, il va supprimer les couples de lapins d'âge 3, et pour les autres, il augmente leur âge de 1. Puis au cours d'un deuxième tour, il va ajouter un couple nouveau-né à la droite de chaque couple ayant atteint l'âge 2 ou 3. Faire le programme.

On trouvera ci-dessous le programme complet :

```
#include <stdio.h>
#include <stdlib.h>
struct lapin { int age; struct lapin * suivant; struct lapin * precedent; };
struct lapin *ptr, *butoir, *lapin1, *newl, *ptravant, *ptrapres;

int main()
{ int etape, nbenleves, nombrecouple, cumul;
  /****** conditions initiales (etape 0) *****/
  butoir=(struct lapin *) malloc(sizeof (struct lapin));
  lapin1=(struct lapin *) malloc(sizeof (struct lapin));
  butoir->age=-1; butoir->suivant=lapin1; butoir->precedent=lapin1;
  lapin1->age=0; lapin1->suivant=butoir; lapin1->precedent=butoir;
  printf("\netape %d\n", 0); /* affichage */
  ptr=butoir->suivant; printf("%d ", ptr->age); nombrecouple=1;
  /****** boucle des étapes *****/
  cumul=0; /* premier parcours pour supprimer les lapins d'age 3 et augmenter
  for(etape=1; etape <=20; etape++) de 1 l'age des autres */
  { ptr=butoir->suivant; nbenleves=0;
    do
    { if (ptr->age==3)
      { nbenleves++; nombrecouple--;
        ptravant=ptr->precedent; ptrapres=ptr->suivant;
        ptravant->suivant=ptrapres; ptrapres->precedent=ptravant;
        free(ptr);
        ptr=ptrapres;
      }
      else { (ptr->age)++; ptr=ptr->suivant; }
    }
    while(ptr!=butoir);
    cumul+=nbenleves;

    ptr=butoir->suivant; /* deuxième parcours pour ajouter les nouveaux-nés à droite de leurs parents */
    do
    { if(ptr->age==2 || ptr->age==3)
      { newl=(struct lapin *) malloc(sizeof (struct lapin)); /* nouvelle cellule */
        ptrapres=ptr->suivant;
        newl->age=0; newl->suivant=ptrapres; newl->precedent=ptr;
        ptr->suivant=newl; ptrapres->precedent=newl;
        ptr=ptrapres;
      }
    }
  }
}
```

```

        nombrecouple++;
    }
    else if (ptr->age==1) ptr=ptr->suivant;
}
while(ptr!=butoir);
/***** affichage *****/
ptr=butoir->suivant;
do
{ printf("%d ",ptr->age);
  ptr=ptr->suivant;
}
while(ptr!=butoir);
printf("\netape %d nombre de couples : %d",etape,nbreCouple);
printf("\nNombre de couples de lapins enlevés durant l'etape : %d ",nbenlevés);
printf("\nNombre cumulé de couples de lapins enlevés : %d\n",cumul);
}
getchar();return 0;
}

```

4) Voir expérimentalement, grâce au programme, à quel moment l'élevage atteint approximativement 3000 têtes. Et indiquer les résultats obtenus qui permettront à l'éleveur de savoir si le remplissage complet de ses clapiers à lapins est acceptable, étant entendu qu'ensuite, il pourra gérer facilement ses 3000 couples en vendant tous ceux qui sont en sus.

Le programme indique, en multipliant ses résultats par 10, qu'à l'étape 20, le nombre des couples atteint 3140, et que jusque là 1500 couples ont pu être vendus. A l'éleveur de juger si le jeu en vaut la chandelle.

### Exercice 3 : Lapins de Fibonacci avec mortalité

Comme on l'a déjà fait, distinguons la population juvénile et la population adulte, en nombre  $a_n$  et  $b_n$  respectivement à l'étape de temps  $n$ . Mais ajoutons un effet limitatif dû à la mortalité, grâce à un coefficient  $1 - d$ , avec  $d$  nombre réel tel que  $0 < d < 1$ . Dans ces conditions, chaque couple adulte produit à l'étape suivante  $k$  couples nouveaux-nés ( $k$  étant éventuellement un nombre positif à virgule), mais avec l'effet limitatif  $1 - d$ , et la population adulte à l'étape  $n + 1$  est égale à la population adulte à l'étape  $n$  augmentée des  $a_n$  jeunes qui sont passés à l'âge adulte, toujours avec ce même effet limitatif, ce qui donne le système linéaire<sup>7</sup> :

$$\begin{cases} a_{n+1} = k(1-d)b_n \\ b_{n+1} = (1-d)(a_n + b_n) \end{cases}$$

1) Déterminer la relation de récurrence vérifiée par la suite  $(u_n)$  correspondant à la population totale.

$$\begin{aligned} u_{n+1} &= a_{n+1} + b_{n+1} = k(1-d)b_n + (1-d)(a_n + b_n) = k(1-d)^2(a_{n-1} + b_{n-1}) + (1-d)(a_n + b_n) \\ &= (1-d)u_n + k(1-d)^2u_{n-1} \end{aligned}$$

2) En s'aidant de l'exercice 1, déterminer les solutions de cette relation de récurrence.

Lorsque l'on cherche des solutions sous la forme  $r^n$ , on trouve que  $r$  doit vérifier :

$$r^2 = (1-d)r + k(1-d)^2 \text{ ou } r^2 - (1-d)r - k(1-d)^2 = 0. \text{ Cette équation admet deux solutions :}$$

<sup>7</sup> Ce modèle est proposé par O. Arino, dans *Mathématiques appliquées à la dynamique de population*.

$r_1 = \frac{1-d}{2}(1+\sqrt{1+4k})$  et  $r_2 = \frac{1-d}{2}(1-\sqrt{1+4k})$ . On constate que  $r_1 > 0$ ,  $r_2 < 0$  avec  $|r_2| < r_1$ .

La solution générale est de la forme  $K_1 r_1^n + K_2 r_2^n$  ou encore  $K_1 (r_1^n - r_2^n)$  si l'on prend  $u_0 = 0$ , la population démarrant à  $u_1$ , nombre positif que l'on se donne, ce qui permet de déterminer  $K_1$ . On aura alors  $u_2 = (1-d)u_1$ .

3) On pose  $f(x) = x^2 - (1-d)x - k(1-d)^2$ . La courbe représentative de  $f$  est une parabole dont le sommet correspond à un minimum, coupant l'axe des  $x$  en  $r_1$  et  $r_2$ . Calculer  $f(1)$ , et selon les cas, en déduire le comportement de la suite  $(u_n)$  lorsque  $n$  augmente indéfiniment.

On obtient  $f(1) = d - k(1-d)^2$ , et trois cas se présentent :

- \* si  $f(1) > 0$ , comme sur la figure 5, cela prouve que  $0 < |r_2| < r_1 < 1$ . Dans ce cas,  $r_1^n$  et  $r_2^n$  tendent vers 0, et la population tend à disparaître.
- \* si  $f(1) < 0$ ,  $r_1 > 1$ , et la population tend à augmenter à l'infini, tout en oscillant.
- \* si  $f(1) = 0$ ,  $r_1 = 1$  et  $|r_2| < 1$ , la population subit des oscillations mais tend vers un équilibre à 1.

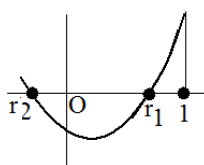


Figure 5 : Cas où  $f(1) > 0$ , d'où  $r_1 < 1$  et  $|r_2| < 1$ .

4) Vérifier expérimentalement les résultats obtenus au 3°.

Si l'on prend  $k = 1$ , on a  $f(1) = 0$  pour  $d = (3 - \sqrt{5}) / 2 \approx 0,385$ , seule valeur avec  $0 < d < 1$ , ce qui donne une suite constante, en prenant comme conditions initiales  $u_0 = u_1 = 50$ . Pour une valeur inférieure  $d = 0,37 < 0,385$ ,  $f(1) < 0$ , la suite est croissante et part à l'infini. Pour une valeur supérieure  $d = 0,41$ ,  $f(1) > 0$ , et la suite décroît vers 0 (figure 6). Les conditions initiales choisies évitent les oscillations.

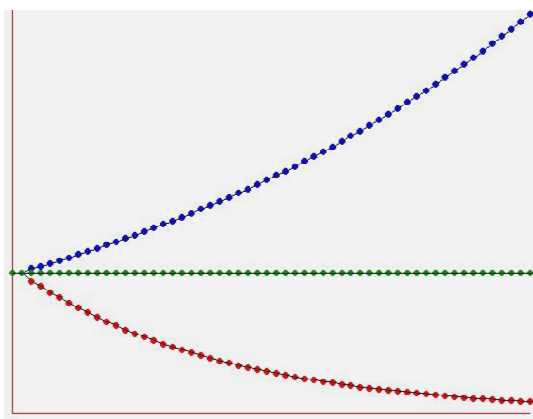


Figure 6 : Suite en bleu pour  $f(1) < 0$ , en rouge pour  $f(1) > 0$ , et en vert pour  $f(1) = 0$ .

### 3. Evolution de style Fibonacci, mais avec un effet de retard et des freins

La croissance exponentielle manque de réalisme à long terme, dans la mesure où des phénomènes de régulation se produisent, empêchant une croissance infinie. Une autre façon de ralentir la croissance exponentielle, mais à court terme cette fois, consiste à faire dépendre l'évolution de la population non pas de ce qu'elle était juste avant, mais aussi de ce qu'elle était bien avant. Ce qui nous amène à modifier les relations de récurrence précédentes.

### 3.1. Evolution proliférante, avec un frein supplémentaire

Une évolution exponentielle correspond à une récurrence de la forme  $u_{n+1} = k u_n$  avec  $k > 1$ . Ajoutons un frein de la forme  $-u_n^2$ , d'où la récurrence  $u_{n+1} = k u_n - u_n^2$ . La relation n'est plus linéaire, car elle fait intervenir un terme du second degré. Si l'on part d'une valeur initiale  $u_0$  proche de 0, le terme  $-u_n^2$  est négligeable et la population a d'abord tendance à augmenter, mais en grossissant, elle subit de plus en plus le frein provoqué par le terme négatif  $-u_n^2$ . Comme  $u_n^2$  est approximativement le nombre de façons de coupler deux à deux les individus de la population, et que par exemple la transmission de maladies se fait par couplages d'individus, le terme  $-u_n^2$  représente la mortalité provoquée par la propagation d'épidémies. Il peut aussi représenter une lutte pour la survie dans le cas de ressources limitées en nourriture.

#### **Exercice 5 : Etude de la récurrence $u_{n+1} = 2 u_n - u_n^2$**

1) Etudier rapidement la fonction  $f(x) = x(2 - x)$  et montrer que tout  $x$  compris entre 0 et 1 est tel que  $f(x)$  est aussi entre 0 et 1.

La courbe représentative de  $f$  est une parabole coupant l'axe des  $x$  en 0 et 2 et de sommet (1, 1) correspondant au maximum de la fonction. La fonction est strictement croissante sur  $[0, 1]$ . Il en découle que pour tout  $x$  compris entre 0 et 1,  $f(x)$  est entre 0 et 1.<sup>8</sup>

2) En constatant que  $u_{n+1} = f(u_n)$ , montrer que si  $0 \leq u_0 \leq 1$ , alors  $0 \leq u_n \leq 1$  quel que soit  $n \geq 0$ .

$$u_{n+1} = 2 u_n - u_n^2 = u_n (2 - u_n) = f(u_n).$$

Pour montrer que  $0 \leq u_n \leq 1$  pour tout  $n$ , faisons un raisonnement par récurrence :

\* la formule est vraie au départ, puisque  $0 \leq u_0 \leq 1$ .

\* Supposons la formule vraie à un certain rang  $n$ , et montrons qu'elle reste vraie au rang  $n + 1$  :

Comme  $0 \leq u_n \leq 1$ , on a aussi  $0 \leq f(u_n) \leq 1$  grâce au 1°, soit  $0 \leq u_{n+1} \leq 1$ .

3) Montrer que 0 et 1 sont deux points fixes, autrement dit qu'en partant de 0 ou de 1 la suite reste constante.

On constate que  $f(0) = 0$  et  $f(1) = 1$ . En partant de  $u_0 = 1$ , on a aussi  $u_1 = f(0) = 0$ , et ainsi de suite par récurrence évidente, soit  $u_n = 0$  pour tout  $n$ . De même la suite est constante à partir de  $u_0 = 1$ .

4) Montrer que la suite est strictement croissante lorsque  $0 < u_0 < 1$ .

Commençons par montrer que  $u_1 > u_0$ . Pour cela formons  $f(x) - x = x - x^2$ . Ce trinôme du second degré, de racines 0 et 1, est positif entre les racines, soit  $f(x) > x$  sur  $]0, 1[$ . Avec  $0 < u_0 < 1$ , on en déduit que  $u_1 = f(u_0) > u_0$ . Puis on fait un raisonnement par récurrence pour montrer que  $u_{n+1} > u_n$  pour tout  $n$ .

\* On a vu que la formule est vraie au départ.

\* Supposons qu'elle soit vraie à un certain rang  $n$ , soit  $u_n > u_{n-1}$ , et montrons qu'elle reste vraie au rang  $n+1$  : Puisque  $u_n > u_{n-1}$ , on a aussi  $f(u_n) > f(u_{n-1})$ , la fonction  $f$  étant strictement croissante, soit  $u_{n+1} > u_n$ .

5) Avec  $0 < u_0 < 1$ , en déduire que le point 0 est un point fixe instable (on l'appelle une source). Autrement dit, si l'on part d'un point voisin si près soit-il de 0, on va s'éloigner de 0. Et montrer que

<sup>8</sup> En toute rigueur, on applique le théorème de la bijection :  $f$  étant continue et strictement croissante sur  $[0, 1]$ , elle réalise une bijection de  $[0, 1]$  sur  $[f(0), f(1)] = [0, 1]$ .



le point fixe 1 est un point fixe stable et attracteur, ou puits, en ce sens que  $u_n$  tend à se rapprocher de 1 et de l'admettre comme limite dès que  $0 < u_0 < 1$ .

Pour  $0 < u_0 < 1$ , la suite est strictement croissante à partir de  $u_0$ , et ses successeurs  $u_1, u_2$ , etc., ne cessent de s'éloigner de  $u_0$ , si près soit-il de 0.

Comme la suite est croissante et majorée par 1, elle admet une limite, et celle-ci est forcément un point fixe. Comme ce n'est pas 0, ce ne peut être que 1. Le point fixe 1 est stable et attracteur, puisque toute suite a pour limite ce point.

6) Montrer que si l'on part de  $u_0$  avec  $1 < u_0 \leq 2$ , on retombe sur le cas précédent à partir de  $u_1$ . Les autres possibilités avec  $u_0 > 2$  sont hors de propos, puisque  $u_1$  est négatif.

La fonction  $f$  étant continue et décroissante sur  $[1, 2]$ , elle réalise une bijection de  $[1, 2]$  sur  $[f(1), f(2)] = [0, 1]$ . Lorsque  $u_0$  est entre 1 et 2,  $u_1 = f(u_0)$  est compris entre 0 et 1. Tout se passe ensuite comme précédemment.

7) En prenant successivement plusieurs conditions initiales, faire le programme qui trace les trajectoires de ces évolutions, dans le repère  $n, u_n$ , en joignant les points successifs obtenus.

Les résultats sont donnés sur la figure 7.

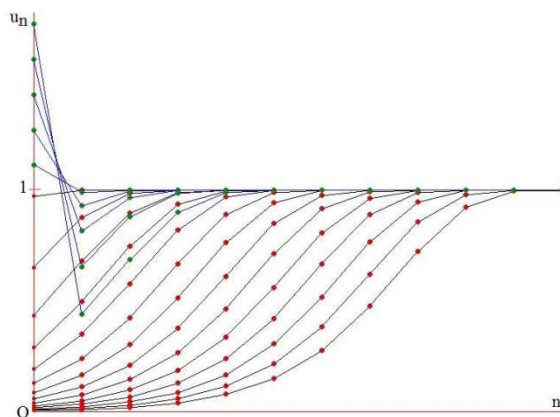


Figure 7 : Evolutions qui convergent vers 1.

### 3.2. Evolution de style Fibonacci, avec un effet de retard

Il s'agit d'une suite dont la relation de récurrence est  $u_{n+1} = u_n + u_{n-T}$ , où  $T$  est supérieur à 1.<sup>9</sup> L'influence de la population  $T$  intervalles de temps auparavant, se traduit encore par une évolution exponentielle, mais précédée d'une longue période de croissance très lente, comme on le constate expérimentalement (figure 8). Ce type d'évolution ressemble par exemple à celui d'une invasion de sauterelles après une longue phase de gestation.

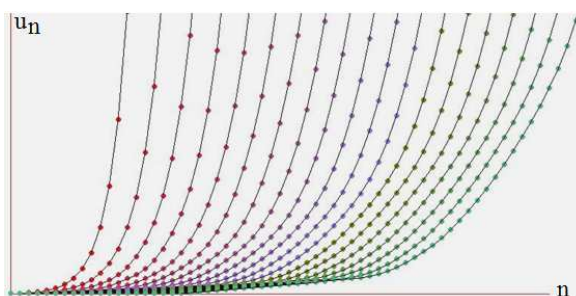


Figure 8 : Suites de Fibonacci avec retard, pour  $T$  allant de 1 à 19, de gauche à droite.

<sup>9</sup> Remarquons que pour  $T = 1$ , on retrouve la suite de Fibonacci.

### 3.3. Evolution avec frein et retard

Considérons la relation de récurrence  $u_{n+1} = u_n + f(u_n, u_{n-T})$ , où  $f$  est une fonction donnée, qui va intégrer un effet de retard et un frein. Donnons quelques exemples :

- $u_{n+1} = u_n + u_{n-T} - u_{n-T}^2$  avec  $f(u_n, u_{n-T}) = u_{n-T} - u_{n-T}^2$
- $u_{n+1} = u_n + u_{n-T} - u_n^2$  avec  $f(u_n, u_{n-T}) = u_{n-T} - u_n^2$
- $u_{n+1} = u_n + u_{n-T} - u_n^2 - u_{n-T}^2$  avec  $f(u_n, u_{n-T}) = u_{n-T} - u_n^2 - u_{n-T}^2$
- $u_{n+1} = 2u_n + u_{n-T} - u_n^2 - u_{n-T}^2$  avec  $f(u_n, u_{n-T}) = u_n + u_{n-T} - u_n^2 - u_{n-T}^2$

Dans les deux premiers cas, la population finit par disparaître après une phase de croissance (*figure 9*).

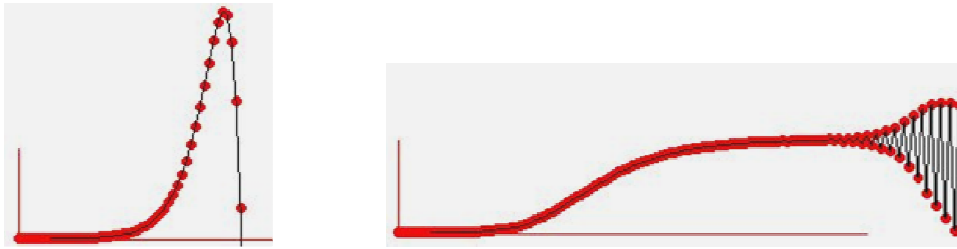


Figure 9 : Evolution avec retard et un frein bien avant à gauche et juste avant à droite.

Dans les deux derniers cas, la population tend à converger, soit très lentement vers 1 avec des oscillations dans le troisième cas (*figure 10 en bleu*), soit directement vers 1/2 dans le quatrième cas (*figure 10 en rouge*).

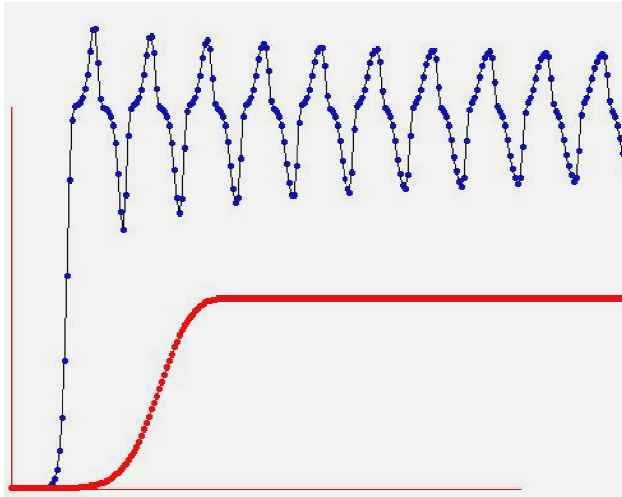


Figure 10 : Evolution vers un équilibre, *en bleu* très lentement avec de fortes oscillations autour de 1, *en rouge* avec retard mais sans oscillations vers 1/2.

Pour programmer ces évolutions, on commence par les conditions initiales qui nécessitent de se donner  $u_0, u_1, \dots, u_T$ , occupant  $T + 1$  cases mémoires, placées dans un tableau  $u[]$ . Par exemple pour  $T = 4$ , le tableau  $u[]$  a 5 cases, remplies par  $u_0, u_1, \dots, u_4$ , avec une variable courante  $k$  mise à 0, c'est-à-dire placée sur la case 0 contenant  $u_0$ . Cette variable  $k$  va parcourir le tableau de façon cyclique.

Pour calculer  $u_5$ , on a besoin de  $u_4$  et de  $u_0$ . Cela étant fait, comme l'on n'aura plus besoin de  $u_0$  par la suite, on met  $u_5$  dans la case 0, et l'on avance  $k$  d'un cran, soit  $k = 1$  (*figure 11 à gauche*). Puis  $u_6$  est obtenu grâce à  $u_5$  et  $u_1$ , et l'on écrase  $u_1$  par  $u_6$ , avant de mettre  $k$  sur la case 2 suivante. Et ainsi de suite. Lorsque  $k$  arrive sur la dernière case, on calcule  $u_9$  grâce à  $u_8$  et  $u_4$ , puis  $u_9$  est mis à la place de

$u_4$ , et  $k$  avance d'un cran de façon cyclique, ce qui signifie qu'il revient à 0 (figure 11 à droite). On constate que chaque nouveau terme ( $u_{n+1}$ ), avant d'être placé dans la case où pointe  $k$ , est calculé à partir du terme ( $u_n$ ) que nous appelons  $pja$  parce qu'il est situé juste avant dans le tableau<sup>10</sup>, et du terme correspondant à  $u_{n-T}$ , que nous appelons  $pba$ , et qui n'est autre que le terme sur qui pointe  $k$ . On en déduit le programme.

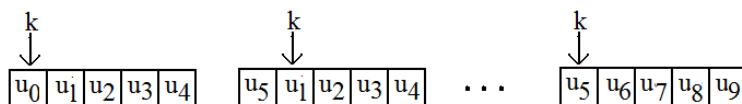


Figure 11 : Evolution du tableau  $u[]$ , parcouru de façon cyclique par la variable  $k$ .

```

On se donne T
for(i=0;i<=T;i++)
{ u[i]=0.001; /* conditions initiales */
  filldisc(xorig+zoomh*(i),yorig-zoomv*u[i],3,red); /* dessin des premiers points (n, un) de la suite */
}
k=0;
for(i=T+1;i<150;i++) /* boucle calculant les termes successifs de la suite */
{ pja=u[(k+T)%(T+1)]; /* population juste avant */
  pba=u[k]; /* population bien avant */
  u[k]=pja+f(pja,pba); /* nouveau terme de la suite */
  xe=xorig+zoomh*i; ye=yorig-zoomv*u[k]; /* coordonnées écran du nouveau point */
  if (xe<800 && ye>0 && ye<600) /* pour rester dans les limites de l'écran */
  { filldisc(xe,ye,3,red);
    line(xorig+zoomh*(i-1),yorig-zoomv*u[(k-1+T+1)%(T+1)], xe,ye,black);
  }
  k++; if (k==T+1) k=0; /* parcours cyclique du tableau par k */
}

```

### Exercice 5 : Evolution d'une population sous l'effet d'une récurrence complexe

Prendre comme relation de récurrence  $u_{n+1} = 0,9 u_n + \frac{0,2 u_{n-T}}{1 + (0,1 u_{n-T})^{20}}$ . Faire le programme, et observer l'évolution de la suite dans le repère  $(u_{n-T}, u_n)$  pour des valeurs croissantes de  $T$ .

La relation de récurrence s'écrit aussi :  $u_{n+1} = u_n + \frac{0,2 u_{n-T}}{1 + (0,1 u_{n-T})^{20}} - 0,1 u_n$ , de la forme

$u_{n+1} = u_n + f(u_n, u_{n-T})$  avec  $f(u_n, u_{n-T}) = \frac{0,2 u_{n-T}}{1 + (0,1 u_{n-T})^{20}} - 0,1 u_n$ , ce qui permet d'utiliser le programme

précédent.

Expérimentalement, le fait de se placer dans le repère  $(u_{n-T}, u_n)$  a pour avantage de localiser la trajectoire dans un domaine limité, à la différence du repère  $(n, u_n)$ , donnant par exemple une trajectoire circulaire au lieu d'oscillations périodiques. On constate qu'après avoir obtenu un point fixe pour  $T = 1$ , signifiant la convergence de  $(u_n)$  vers une limite, la trajectoire converge vers une courbe fermée de plus en plus complexe lorsque  $T$  augmente, et que l'on appelle attracteur, dans la mesure où pour un certain nombre de valeurs initiales, on tombe finalement sur cette même forme. Tant qu'il en est ainsi, les oscillations de la suite sont périodiques, même si sur une même période les mouvements sont de plus en plus compliqués. On voit que l'attracteur se déforme, zigzague, se dédouble à plusieurs reprises, donnant ce que l'on appelle un attracteur étrange. Puis, à partir de  $T = 30$ , l'attracteur éclate

<sup>10</sup> Lorsque  $k$  vaut 0, le terme juste avant appelé  $pja$  se trouve dans la case 4, soit  $-1$  ramené modulo 5. Mais les termes négatifs modulo ne sont pas acceptés dans un langage informatique comme le langage C. Au lieu de  $-1$ , on fait donc  $(-1 + 5)$  ramené modulo 5, soit 4.

en un nuage de points, signe que les oscillations ne sont plus périodiques, et l'on assiste à un phénomène chaotique.

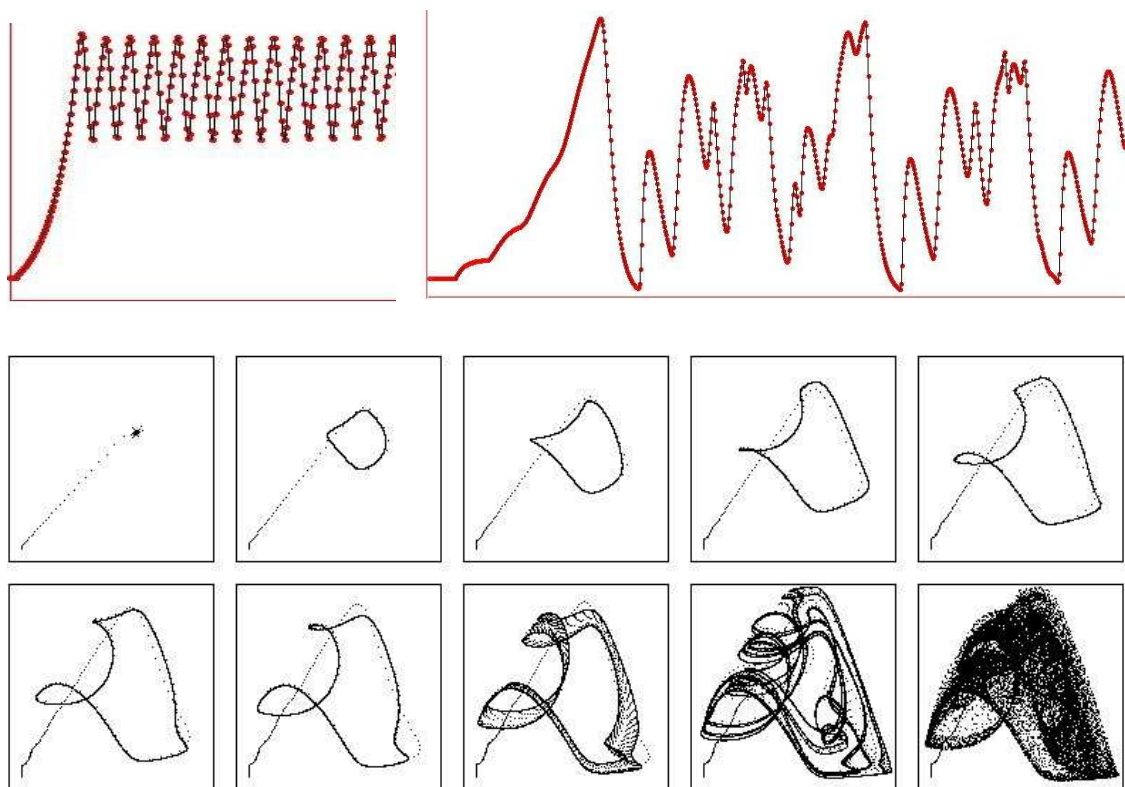


Figure 12 : En haut, évolution de la suite dans le repère  $(n, u_n)$ , à gauche pour  $T = 4$  avec des oscillations périodiques, à droite pour  $T = 34$  avec des oscillations chaotiques. En bas, même évolution de la suite mais dans le repère  $u_{n-T}, u_n$ , pour  $T$  allant de 1 à 28. Les oscillations périodiques donnent maintenant une courbe fermée de plus en plus complexe, tandis que pour  $T$  proche de 30 le nuage de points obtenu indique l'apparition d'un phénomène chaotique.

Cette évolution des attracteurs de l'ordre au chaos, que nous venons de rencontrer pour la première fois, est typique de nombreux phénomènes. Le cas générique est celui de l'application logistique, que nous allons voir maintenant.

## 4. L'application logistique

Il s'agit de la suite obéissant à la relation de récurrence

$u_{n+1} = c u_n (1 - u_n) = c u_n - c u_n^2$ , soit  $u_{n+1} = f(u_n)$  où  $f$  est l'application  $f(x) = c x (1 - x)$ , avec  $u_0$  donné, pris entre 0 et 1. Le paramètre  $c$  est choisi entre 2 et 4.

Ici encore, il s'agit de l'évolution d'une population qui aurait un essor exponentiel ( $c u_n$ ) si celui-ci n'était contrecarré par diverses tueries réciproques ou propagation d'épidémies, ce qui nécessite des contacts deux à deux, d'où le terme  $-c u_n^2$ . Vue sous un autre angle, la fonction sous-jacente  $f(x) = c x (1 - x)$  peut être considérée comme la plus simple des fonctions non linéaires, à cause de la présence d'un terme du second degré. La parabole représentative de cette fonction s'insère dans un carré  $1 \times 1$ , les coordonnées liées à  $u_n$  étant positives.

### 4.1. Diagramme en toile d'araignée de $(u_n)$

On veut voir comment s'effectue la succession  $u_0, u_1, u_2, \dots$ , ce que l'on appelle la trajectoire de  $u_0$ . Pour cela commençons par tracer la parabole d'équation  $y = f(x)$ , soit  $y = c x (1 - x)$ , ainsi que la

première bissectrice du repère. On constate l'existence de deux points fixes, c'est-à-dire de nombres  $x$  vérifiant  $f(x) = x$ , ici  $x = 0$  et  $x = 1 - 1/c$ , correspondant à l'intersection de la courbe et de la bissectrice. Plaçons le point  $u_0$  sur l'axe des  $x$ , puis montons verticalement jusqu'à la parabole. Le point obtenu a pour ordonnée  $u_1 = f(u_0)$ , et l'on obtient un point  $u_1$  sur l'axe des  $y$ . Mais comment avoir  $u_2$  ? Il faut d'abord rabattre le point  $u_1$  sur l'axe des  $x$  : pour cela on mène une horizontale jusqu'à la première bissectrice, puis on descend verticalement jusqu'à l'axe des  $x$ . Maintenant  $u_1$  est sur l'axe des  $x$ , puis en montant jusqu'à la parabole, on obtient le point d'ordonnée  $u_2$ . Et l'on recommence comme avant... Cela étant bien compris, il suffit de tracer une succession de marches d'escalier pour visualiser l'évolution de la suite. Le dessin obtenu est appelé diagramme en toile d'araignée. Un exemple est traité sur la figure 13.

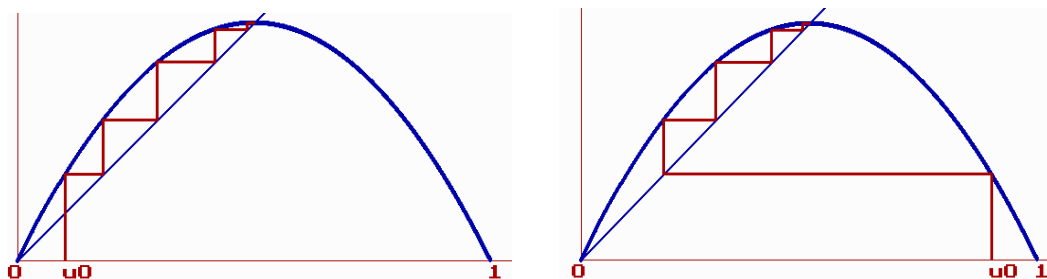


Figure 13 : Visualisation de la suite vérifiant  $u_{n+1} = f(u_n)$  avec  $f(x) = 2x(1-x)$ , à partir de  $u_0 = 0,1$  puis de  $u_0 = 1,9$ . On constate dans les deux cas que la population finit par se stabiliser à 0,5.

Pour faire le programme correspondant, on commence par tracer le trait vertical initial, joignant le point d'abscisse  $u_0$  sur l'axe horizontal au point correspondant  $(u_0, f(u_0))$  sur la courbe. Puis on entre dans une boucle répétitive, avec à chaque fois le dessin d'un trait horizontal suivi d'un trait vertical. Plus précisément, il s'agit de passer de façon répétée d'un point  $(x_1, y_1)$  de la courbe à un point  $(x_2, y_2)$  de la première bissectrice, puis de revenir sur la courbe. Le point  $(x_1, y_1)$  de la courbe est tel que  $y_1 = f(x_1)$ , et l'on va vers la première bissectrice par un trait horizontal, d'où  $x_2 = y_1$  et  $y_2 = y_1$ . Puis on revient sur la courbe par un trait vertical, d'où les nouvelles valeurs de  $x_1$  et  $y_1$  :  $x_1 = x_2$  et  $y_1 = f(x_2)$  (figure 14). Et l'on recommence à partir de ce point ... D'où le programme :

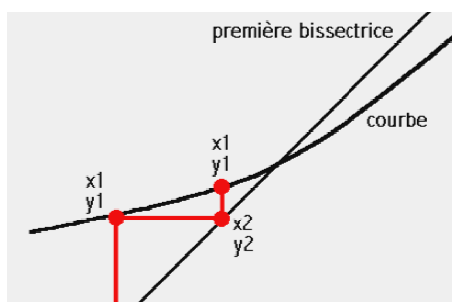


Figure 14 : Marche d'escalier pour tracer le diagramme en toile d'araignée.

```

u0=0.1; u1=f(u0); line(xorig+zoom*u0,yorig,xorig+zoom*u0,yorig-zoom*u1);
x1=u0;y1=u1;
for(i=0;i<100;i++)
{ x2=y1;y2=y1; line(xorig+zoom*x1,yorig-zoom*y1,xorig+zoom*x2,yorig-zoom*y2);
  x1=x2;y1=f(x2); line(xorig+zoom*x2,yorig-zoom*y2,xorig+zoom*x1,yorig-zoom*y1);
}

float f(float x) /* la fonction f */
{ float y;      y=c*x*(1.-x); return y; }

```

### Exercice 6 : Etude théorique de la suite vérifiant $u_{n+1} = f(u_n)$ avec $f(x) = 2x(1-x)$

1) On suppose d'abord que  $u_0$  appartient à l'intervalle  $I = ]0, 1/2]$ . Montrer que  $f(I) = I$ , puis que  $u_n$  appartient à  $I$  pour tout  $n$ .

L'équation  $y = 2x(1-x)$  de la courbe de  $f$  est un trinôme du second degré. La courbe est une parabole d'axe vertical. Elle passe par les deux points  $(0,0)$  et  $(1,0)$ . Son axe vertical a donc pour équation  $x = 1/2$ , et son sommet est  $(1/2, f(1/2)=1/2)$ .

Strictement croissante et continue sur  $]0, 0,5] = I$ , la fonction  $f$  est une bijection de  $I$  sur  $]f(0), f(0,5)]=]0, 0,5]=I$ , d'où  $f(I)=I$ .

Faisons un raisonnement par récurrence pour montrer que  $u_n$  appartient à  $I$ .

- $u_0$  appartient à  $I$ .
- Supposons que pour un certain rang  $n$   $u_n$  appartienne à  $I$ , et montrons qu'il en est de même pour  $u_{n+1}$  :  $u_{n+1} = f(u_n)$  avec  $u_n$  dans  $I$ , d'où  $f(u_n)$  appartient à  $I$  aussi puisque  $f(I)=I$ .

2) Montrer que  $(u_n)$  est croissante.

Formons  $u_{n+1} - u_n = 2u_n(1-u_n) - u_n = u_n - 2u_n^2 = u_n(1-2u_n) \geq 0$  car  $u_n > 0$  et que  $1-2u_n \geq 0$  sachant que  $u_n \leq 0,5$ . La suite est croissante.

3) En déduire que la suite  $(u_n)$  converge vers une limite que l'on précisera.

Croissante et majorée par  $1/2$ , la suite converge vers une limite  $\leq 1/2$ . Cette limite  $L$  est un point fixe, elle est telle que  $f(L) = L$ ,  $f$  n'ayant aucun problème de continuité, d'où  $L = 0$  ou  $L = 1/2$ . Mais  $L = 0$  est impossible puisque  $u_0 > 0$  et  $(u_n)$  croissante. La suite converge donc vers  $1/2$ .

4) Que se passe-t-il si l'on prend maintenant  $u_0$  dans  $[0,5, 1[$  ?

Sur cet intervalle, la fonction  $f$  est strictement décroissante et continue. Elle réalise une bijection de  $[0,5, 1[$  sur  $[0,5, 0[ = I$ . Alors  $u_1 = f(u_0)$  appartient à  $I$ . Il suffit alors de reprendre à partir de  $u_1$  ce que l'on a fait précédemment à partir de  $u_0$  sur  $I$ . La suite converge encore vers  $1/2$  (cf. figure 13).

## 4.2. Stabilité des points fixes

Rappelons que pour une fonction  $f$  un point fixe vérifie  $f(x) = x$ . Il s'agit de l'abscisse (et de l'ordonnée) du point d'intersection de la courbe de  $f$  et de la première bissectrice du repère. Si la suite vérifiant  $u_{n+1} = f(u_n)$  admet une limite  $L$ , celle-ci doit vérifier la relation de récurrence, soit  $f(L) = L$ , et la limite est un des points fixes. Un point fixe  $x^*$  est considéré comme stable si, sous l'effet de  $f$ , il attire vers lui les points de son voisinage, et cela se produit si la valeur absolue de la dérivée en  $x^*$  est inférieure à 1. Si la dérivée est positive, le diagramme en toile d'araignée de la suite est en marche d'escalier au voisinage du point (cf. figure 13). Si elle est négative, le diagramme est en colimaçon. Cela se comprend aisément puisque la dérivée est la pente de la tangente en  $x^*$ . Par contre si  $|f'(x^*)| > 1$ , le point est instable, en ce sens qu'il repousse les points de son voisinage.

Pour l'application logistique, la dérivée de  $f(x)$  est  $f'(x) = c - 2cx$ . Le point fixe 0 est instable puisque  $f'(0) = c$ , et que l'on suppose  $c$  compris entre 2 et 4. Le deuxième point fixe est  $x^* = 1 - 1/c$ , avec  $f'(x^*) = 2 - c$ . Lorsque  $c$  est compris entre 2 et 3, le point  $x^*$  est stable, et la suite va converger vers lui en colimaçon, puisque la dérivée est négative<sup>11</sup>.

Mais que se passe-t-il lorsque  $c$  traverse la frontière 3 ? Le point  $x^*$  devient instable et repousseur, mais il apparaît deux points  $x_1^*$  et  $x_2^*$  formant un cycle fixe, vers lesquels la suite va converger en oscillant de l'un à l'autre. Ces points sont tels que  $f(x_1^*) = x_2^*$ , et  $f(x_2^*) = x_1^*$ , ou encore  $f^2(x_1^*) = x_1^*$  et  $f^2(x_2^*) = x_2^*$ ,  $f^2$  étant la fonction  $f$  répétée deux fois en succession. Ainsi la fonction  $f^2$  admet un cycle

<sup>11</sup> Le fait que le point fixe soit stable ne suffit pas pour montrer que la suite converge bien vers lui. Dans le cas présent de la suite logistique, on peut démontrer que la suite converge bien vers ce point quel que soit  $u_0$  dans  $]0, 1[$ . Cet intervalle est appelé le bassin d'attraction de  $x^*$ .



fixe stable de deux points  $x_1^*$  et  $x_2^*$  tout en gardant le point fixe  $x^*$  qui, lui, est devenu instable. On constate par le calcul que ces deux points restent stables pour  $c$  compris entre 3 et  $1 + \sqrt{6} \approx 3,45$  (cf. *exercice 7 ci-dessous*). Au delà de  $c = 3,45$ , ces deux points deviennent instables à leur tour, et ils se dédoublent pour donner chacun deux nouveaux points fixes stables, d'où un cycle fixe de 4 points sur lequel la suite va s'enrouler. Puis ces quatre points se dédoublent à leur tour, et ainsi de suite. Il se produit ce que l'on appelle une cascade de bifurcations. Les valeurs de  $c$  pour lesquelles se produisent ces dédoublements sont  $c_1 = 3$ ,  $c_2 = 3,45$  comme on l'a vu, puis  $c_3$ ,  $c_4$ , etc. cette suite ayant une limite  $c_\infty = 3,57$ .

Jusque-là, nous sommes restés dans une zone où la suite converge, même si cela se fait sur un cycle de points. Mais que se passe-t-il au delà de  $c = 3,57$  ? Les éléments de la suite s'étalent maintenant sur une zone dense de points, ce qui marque la fin des phénomènes périodiques précédents, annonçant la présence d'un certain chaos. Lorsque  $c$  augmente, cette zone grandit pour occuper tout le segment  $]0, 1[$  lorsque  $c = 4$ , ce que l'on appelle le chaos ultime. Toutefois dans un certain nombre de fenêtres à l'intérieur de la zone chaotique, on retrouve un certain ordre, notamment la présence de cycles fixes de trois points, ainsi que de nouvelles bifurcations. Un certain nombre de ces résultats sont indiqués sur la *figure 15*.

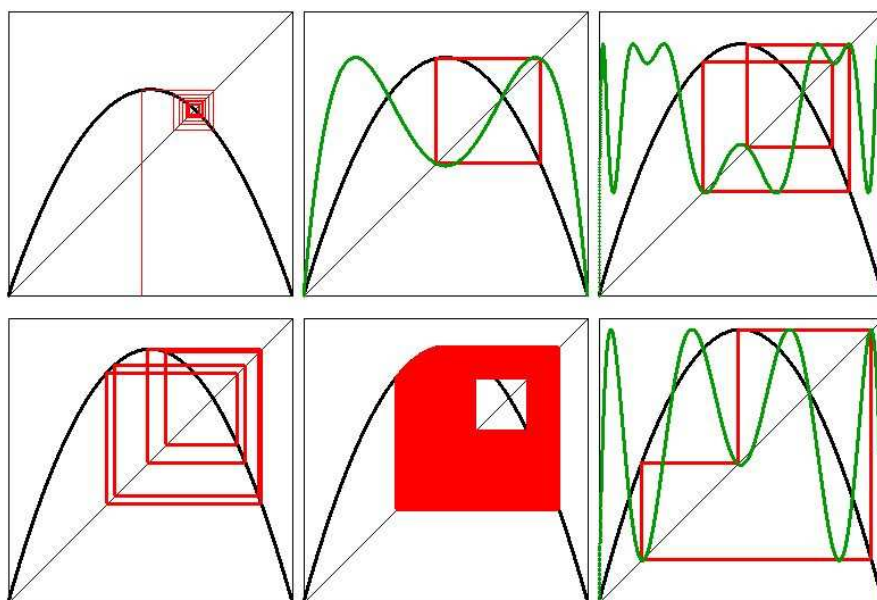


Figure 15 : Evolution de la suite logistique pour des valeurs croissantes de  $c$ . De gauche à droite et de haut en bas :

- $c = 2,9$  avec le diagramme en colimaçon vers le point fixe  $x^*$ ,
- $c = 3,35$  où l'on ne montre que la fin de la trajectoire, avec un cycle fixe de deux points qui sont aussi des points fixes pour la fonction  $f^2$ , dont la courbe est tracée en vert.
- $c = 3,54$  avec le cycle fixe de quatre points, qui sont aussi les points fixes pour la courbe de  $f^4$ .
- $c = 3,564$  avec un cycle fixe de huit points.
- $c = 3,6$  dans la zone chaotique, avec la trajectoire occupant une zone dense.
- $c = 3,84$  où apparaît un cycle fixe de trois points, qui sont aussi points fixes de  $f^3$  (courbe en vert).

### Exercice 7 : Etude du cycle fixe de deux points, pour $c > 3$

1) Calculer  $f^2(x) = f(f(x))$ .

$$x \xrightarrow{f} y = c x(1-x) \xrightarrow{f} z = f^2(x) = c y(1-y) = c^2 x(1-x)(1-cx+cx^2)$$

$$\text{Finalement, } f^2(x) = -c^3 x^4 + 2c^3 x^3 - (c^3 + c^2)x^2 + c^2 x.$$

2) En utilisant le fait que les points de  $f$  sont aussi des points fixes de  $f^2$ , déterminer les deux nouveaux points fixes de  $f^2$ .

Les points fixes vérifient  $-c^3 x^4 + 2c^3 x^3 - (c^3 + c^2)x^2 + c^2 x = x$ , soit :  
 $c^3 x^3 - 2c^3 x^2 + (c^3 + c^2)x - c^2 + 1 = 0$ , après avoir divisé par  $x$ , le point  $x = 0$  étant un point fixe.

$$x^3 - 2x^2 + \frac{c+1}{c}x - \frac{c^2-1}{c^2} = 0.$$

On sait que le point fixe  $x^*$  de  $f$  est aussi un point fixe pour  $f^2$ , ce qui permet de factoriser :

$$\left(x - \frac{c-1}{c}\right)\left(x^2 - \frac{c+1}{c}x + \frac{c+1}{c^2}\right) = 0$$

Les deux nouveaux points fixes sont les racines de

$$x^2 - \frac{c+1}{c}x + \frac{c+1}{c^2} = 0, \quad \text{soit} \quad x_1^* = \frac{c+1 + \sqrt{c^2 - 2c - 3}}{2c} \quad \text{et} \quad x_2^* = \frac{c+1 - \sqrt{c^2 - 2c - 3}}{2c}, \quad \text{car le}$$

discriminant  $c^2 - 2c - 3$  est positif pour  $c > 3$ .

3) Vérifier que les points 0 et  $x^*$  sont instables pour  $f^2$ . Utiliser le fait que  $f^2(x) = f(f(x))$  pour dériver.

La dérivée de  $f^2$  est  $f'(f(x)) \times f'(x)$ , avec  $f'(x) = c - 2cx$ .

En 0,  $(f^2(0))' = c^2$  puisque  $f(0) = 0$ . Le point 0 reste instable.

En  $x^*$ , avec  $f(x^*) = x^*$ ,  $(f^2(x^*))' = (c - 2cx^*)^2 = (c - 2(c-1))^2 = (-c+2)^2 < -1$  pour  $c > 3$ . Le point est devenu instable.

4) Montrer que les deux nouveaux points fixes de  $f^2$  sont stables pour  $c > 3$  jusqu'à une valeur de  $c$  que l'on précisera..Constater que la dérivée de  $f^2$  est la même en ces deux points.

En  $x_1^*$ , la dérivée de  $f^2$  est  $f'(f(x_1^*)) \times f'(x_1^*) = f'(x_2^*) \times f'(x_1^*)$ , et l'on a le même résultat en  $x_2^*$ .

$$\begin{aligned} &= (c - 2cx_2^*)(c - 2cx_1^*) \\ &= c^2 - 2c^2(x_1^* + x_2^*) + 4c^2x_1^*x_2^* \\ &= c^2 - 2c(c+1) + 4(c+1) = -c^2 + 2c + 4. \end{aligned}$$

Pour  $c$  supérieur à 3, cette fonction de  $c$  est strictement décroissante. En 3, elle vaut 1, et quand  $c$  augmente, elle décroît jusqu'à  $-1$  pour une valeur de  $c$  telle que  $-c^2 + 2c + 4 = -1$ ,  $-c^2 + 2c + 5 = 0$ . Cette équation admet une seule solution supérieure à 3, il s'agit de  $1 + \sqrt{6} \approx 3,45$ .

Ainsi pour  $c$  allant de 3 à 3,45, le point fixe  $x_1^*$  est stable, et il en est de même pour  $x_2^*$ . Au-delà, les deux points vont devenir instables.

### 4.3. Diagramme des bifurcations, de l'ordre au chaos

Une autre façon de voir l'évolution de la suite logistique suivant les valeurs du paramètre  $c$  consiste à se placer dans le repère  $(c, u_n)$ . Par  $u_n$ , nous entendons les valeurs de la suite pour  $n$  grand, quand celle-ci a atteint un certain équilibre. Par exemple en itérant  $n$  jusqu'à 1000, on ne dessine que les points obtenus au-delà de 800. Par exemple, pour chaque valeur de  $c$  inférieure à 3, on obtiendra un seul point, à cause de la convergence de la suite  $(u_n)$ . On verra ensuite apparaître les bifurcations, comme signe annonciateur du chaos, puis le chaos lui-même, jusqu'au chaos ultime pour  $c = 4$ .<sup>12</sup> On s'aperçoit alors qu'à l'intérieur de la zone chaotique surgissent des fenêtres de réordonnement, par exemple avec l'apparition d'un cycle fixe de trois points en  $c = 3,84$ , lui-même soumis ensuite à des bifurcations (figure 16).

<sup>12</sup> Remarquons que dans l'exercice 5, avec son contexte plus complexe, on avait assisté à une évolution analogue.



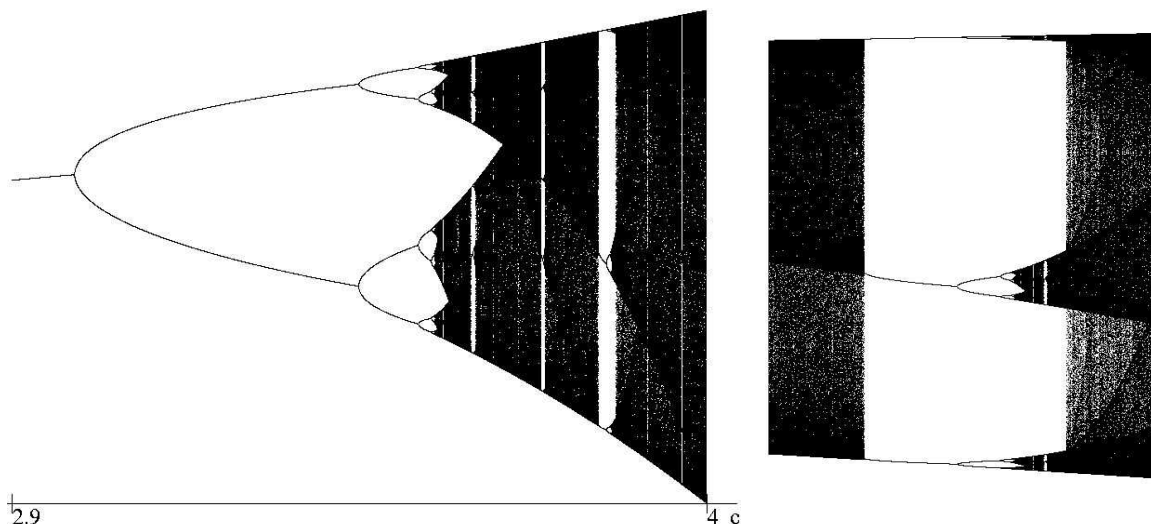


Figure 16 : A gauche, diagramme des bifurcations et zone chaotique avec ses fenêtres dans le repère  $c, u_n$ . A droite, grossissement de la plus grande fenêtre et de son voisinage, entre  $c = 3,815$  et  $c = 3,87$ , pour voir le cycle fixe de trois points, qui se dédoublent ensuite, avec de nouvelles fenêtres dans la fenêtre..

### Exercice 8 : Chaos ultime et fréquences

Pour  $c = 4$ , dans la phase de chaos ultime, avec  $f(x) = 4x(1-x)$  représentée graphiquement par la parabole dont le sommet est en  $(1/2, 1)$ , les points successifs  $(u_n, u_{n+1})$  de la suite logistique parcourent toute la parabole lorsque l'on prend une trajectoire suffisamment longue, en partant de  $u_0$  quelconque (tout en évitant certaines valeurs particulières comme 0,5 ou 0,25 où le chaos ultime ne se produit pas). En découpant l'intervalle  $[0, 1]$  de l'axe des  $x$  en 100 petits intervalles, faire le programme qui compte combien de fois chacun de ces petits intervalles est touché par la trajectoire. Tracer la courbe des fréquences, afin de connaître la distribution des touchers de la suite sur la parabole. Par la même occasion déterminer la valeur moyenne des valeurs absolues des dérivées de  $f$  prises en chaque point de la trajectoire.

Le programme s'écrit simplement, ici en mode texte :

```
#include <stdio.h>
#include <stdlib.h>
int f[1000]; double cumulderivee;

int main()
{ double x,fprime; int i,k;
  x=0.31; /* valeur initiale */
  for(i=0;i<500000;i++) /* boucle de la trajectoire */
  { x=4.*x-4.*x*x; /* passage d'un terme au suivant de la suite logistique */
    fprime=4.-8.*x; /* calcul de la dérivée en chaque point */
    k=(int) (100.*x); /* k est le numéro de l'intervalle où tombe x */
    f[k]++; /* f[k] augmente de 1 chaque fois qu'un point tombe dans l'intervalle k */
    cumulderivee+=labs(fprime); /* cumul des valeurs absolues des dérivées */
  }
  for(i=0;i<100;i++) printf("(%d %d) ",i,f[i] );
  printf("\n\n dérivée moyenne: %f ",cumulderivee/500000. );
  getchar();return 0;
}
```

L'histogramme des fréquences est donné sur la figure 17, et l'on trouve une valeur moyenne de la dérivée en valeur absolue égale à 1,97.

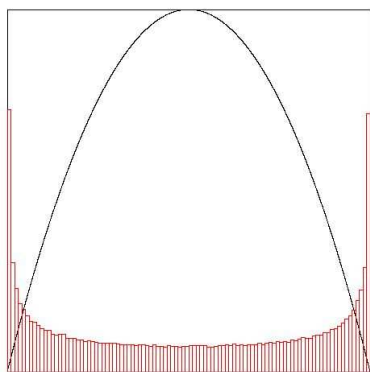


Figure 17 : Fréquences des touches de la parabole par les points de la suite logistique pour  $c = 4$ .

## 5. Exposant de Liapounov et mesure du chaos

Partons d'un point  $x_0$ , en fait un nombre, et soumettons-le à l'action répétée d'une fonction  $f$ . La trajectoire obtenue est  $x_0, x_1 = f(x_0), x_2 = f(x_1)$ , etc. Par définition, le nombre de Liapounov est :

$$\lambda = \lim_{n \rightarrow \infty} \left( \left| \frac{df}{dx}(x_0) \right| \left| \frac{df}{dx}(x_1) \right| \dots \left| \frac{df}{dx}(x_{n-1}) \right| \right)^{\frac{1}{n}}$$

Il s'agit de la valeur moyenne de la dérivée en valeur absolue, en tous les points de la trajectoire, ou encore de la valeur moyenne de la pente de la tangente (en valeur absolue) en ces points sur la courbe de  $f$ . Quand la trajectoire converge vers un point, ce nombre est inférieur à 1 car la pente de la tangente au voisinage des points fixes est inférieure à 1 en valeur absolue, et c'est là que se trouvent quasiment tous les points de la trajectoire. De même lorsque la trajectoire converge vers un cycle fixe de points, où le produit des valeurs absolues des dérivées sur un cycle est inférieur à 1.<sup>13</sup> Par contre, quand ce nombre devient supérieur à 1, cela indique qu'il n'y a plus de cycles de points stables et que l'on est entré dans un mouvement chaotique. Par exemple pour la fonction  $f$  de la suite logistique et sa parabole lorsque  $c = 4$ , correspondant au chaos ultime, tous les points de cette courbe sont touchés et l'on a constaté expérimentalement que la moyenne des valeurs absolues des dérivées est proche de 2 (cf. exercice 8).

En pratique, on préfère prendre ce qui s'appelle l'exposant de Liapounov  $\mu$  tel que  $\mu = \log(\lambda)$ , ce qui remplace les multiplications par des additions et 1 par 0 :

$$\mu = \lim_{n \rightarrow \infty} \left( \frac{1}{n} \sum_{k=0}^{n-1} \log \left| \frac{df}{dx}(x_k) \right| \right)$$

Pour  $\mu$  négatif, le comportement de la suite est cyclique avec convergence vers un ou des points, et pour  $\mu$  positif, le comportement est chaotique, plus ou moins suivant la valeur de  $\mu$ .

### Exercice 9 : Exposant de Liapounov pour l'application logistique

Reprendre l'application logistique  $f(x) = c x (1 - x)$  et faire le programme qui dessine la courbe donnant l'exposant de Liapounov  $\mu$  en fonction de  $c$  pour  $c$  variant de 2 à 4.

<sup>13</sup> On l'a vu pour un cycle de deux points  $x_1$  et  $x_2$ , avec  $f^2(x_1) = f(f(x_1)) = f'(f(x_1)) f'(x_1) = f'(x_2) f'(x_1)$ . Un point fixe comme  $x_1$  est stable pour  $f^2$  si  $|f^2(x_1)| < 1$ , ce qui implique que le produit des dérivées sur un cycle fixe de points pour  $f$  est aussi inférieur à 1. Ce qui est valable pour un cycle fixe de deux points le reste pour un cycle fixe de  $n$  points.

Dans le programme, on a mis les logarithmes en base 2, ce qui revient à diviser les logarithmes népériens  $\log$  par  $\log(2)$ , et dans ces conditions le chaos ultime est obtenu pour  $\mu$  de l'ordre de 1. On constate notamment sur le dessin (*figure 18*) des variations verticales dans la zone de stabilité, pour  $\mu$  négatif. Les points les plus bas sont obtenus lorsque la tangente aux points fixes est horizontale, les plus hauts (avec  $\mu$  nul) correspondent aux bifurcations. On constate aussi, dans le zone de chaos avec  $\mu$  positif, de nombreuses descentes dans la zone négative correspondant aux fenêtres de réordonnement.

```
line(xorig,yorig, xorig+700,yorig,black); /* axe horizontal du repère */
for (c=2.7;c<4.;c+=0.00005)
{ x=0.4; cumulog=0.;
  for (i=0;i<400;i++) /* on pratique 400 itérations */
  { der=c-2*c*x; absder= fabs(der);
    if(absder>0.000001) cumulog+=log(absder)/log(2.);
    x=c*x*(1.-x);
  }
  mu=cumulog/400.; putpixel(xorig+400.*(c-2.7),yorig-50.*mu,black);
}
```

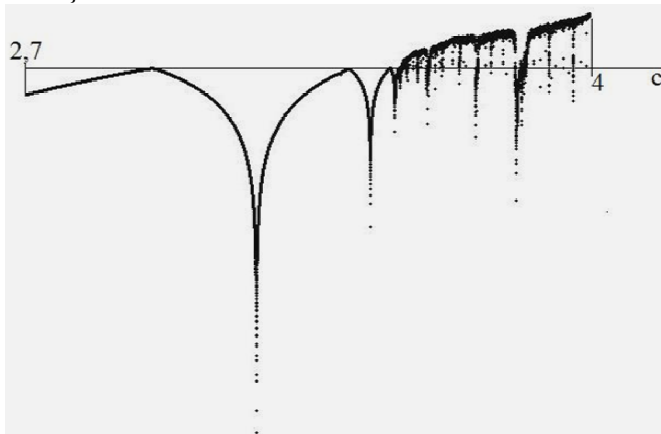


Figure 18 : Exposant de Liapounov  $\mu$  en fonction de  $c$

L'exposant de Liapounov est très précisément lié à la notion de sensibilité aux conditions initiales, c'est-à-dire à la tendance pour deux points très proches de voir leurs trajectoires se séparer de façon exponentielle. Quand elle a lieu cette sensibilité aux conditions initiales est un facteur majeur de comportement chaotique.

Prenons deux points initiaux  $x_0$  et  $y_0$  très proches. Par itérations successives, on arrive à  $x_n = f^n(x_0)$  et  $y_n = f^n(y_0)$ . Exprimons qu'il y a divergence exponentielle :

$|y_n - x_n| = |y_0 - x_0| \exp(\alpha n)$  avec  $\alpha > 0$ , ce qui signifie que :

$$\frac{1}{n} \log \left| \frac{y_n - x_n}{y_0 - x_0} \right| \text{ tend vers } \alpha.$$

Mais souvent, comme dans le cas de l'application logistique, le mouvement se fait dans une région limitée, et il ne peut pas y avoir de divergence exponentielle infinie. Alors, au lieu de pratiquer une évolution du petit à l'infini, on la fait de l'infiniment petit au fini, en faisant tendre  $y_0 - x_0$  vers 0, d'où :

$$\begin{aligned} \alpha &= \lim_{n \rightarrow \infty} \left( \frac{1}{n} \left( \lim_{y_0 - x_0 \rightarrow 0} \log \left| \frac{y_n - x_n}{y_0 - x_0} \right| \right) \right) = \lim_{n \rightarrow \infty} \left( \frac{1}{n} \log \left| \frac{df^n}{dx}(x_0) \right| \right) \\ &= \lim_{n \rightarrow \infty} \left( \frac{1}{n} \sum_{k=0}^{n-1} \left| \frac{df}{dx}(x_k) \right| \right) \end{aligned}$$

car la dérivée  $n^{\text{ème}}$  s'obtient en faisant le produit des dérivées sur les points successifs de la trajectoire, par généralisation de ce que l'on a fait précédemment pour  $f^2(x)$ . On retrouve ainsi l'exposant de Liapounov, et  $\mu = \alpha$ .

### Exercice 10 : Espace de Liapounov

Compliquons le phénomène en faisant agir en alternance deux fonctions logistiques, l'une avec le paramètre  $c$ , l'autre avec le paramètre  $c'$ , à partir d'un point  $x_0$ . Puis plaçons-nous dans le repère  $(c, c')$ , les deux paramètres étant compris entre 0 et 4. Pour chaque point  $(c, c')$ , on lance la trajectoire du point  $x_0$  et l'on détermine l'exposant de Liapounov  $\mu$  correspondant. En associant une couleur qui est fonction de  $\mu$ , chaque point  $(c, c')$  se trouve ainsi colorié. Plus précisément on choisira un dégradé d'une certaine couleur pour la zone chaotique ( $\mu > 0$ ) et un dégradé d'une autre couleur dans la zone d'ordre ( $\mu < 0$ ). Faire le programme, en pratiquant des zooms sur des zones de l'écran. On constatera que des dessins curieux surgissent. Surtout si l'on introduit des variations. Au lieu de répéter l'alternance 12 de la fonction logistique 1 et de la fonction logistique 2, on peut choisir d'autres motifs répétitifs, au gré de son imagination, comme l'ont fait Mario Markus et Benno Hess vers 1990.

On sait déjà calculer l'exposant de Liapounov. Le plus délicat est de passer de la zone calcul, carré de longueur 4 avec une fenêtre où  $c$  va de  $c_1$  à  $c_2$  horizontalement, et de  $c'_1$  à  $c'_2$  verticalement, à la même fenêtre, mais grossie, sur l'écran. Le rectangle écran ayant comme dimensions  $L$  et  $L'$ , on a aussitôt le zoom horizontal  $\text{zoomx} = L/(c_2 - c_1)$  et  $\text{zoomy} = L'/(c'_2 - c'_1)$ . On en déduit les formules de passage du point calcul  $(c, c')$  au point écran  $(ce, c'e)$  (figure 19) :

$$ce = x_{\text{orig}} + \text{zoomx} (c - c_1)$$

$$c'e = y_{\text{orig}} - \text{zoomy} (c' - c'_1)$$

En inversant :

$$c = (ce - x_{\text{orig}} + \text{zoomx} \times c_1) / \text{zoomx}$$

$$c' = (-c'e + y_{\text{orig}} + \text{zoomy} \times c'_1) / \text{zoomy}$$

On en déduit le programme, des résultats étant présentés sur la figure 20.

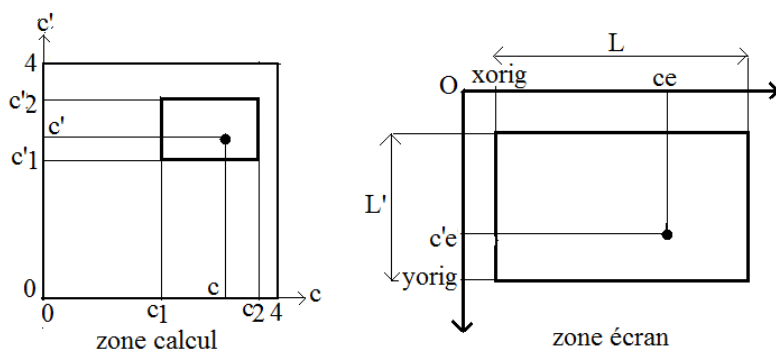


Figure 19 : Passage de la zone calcul à la zone écran.

On se donne  $L$  et  $LL$ , longueur et largeur du rectangle de l'image sur l'écran, par exemple 400 et 650

$c1 = 2.76; c2 = 4;$  /\* dimensions de la fenêtre image dans la zone calcul \*/

$cc1 = 2.2; cc2 = 4;$

$\text{zoomx} = L/(c2 - c1); \text{zoomy} = LL/(cc2 - cc1);$

for( $ce = x_{\text{orig}}; ce < x_{\text{orig}} + L; ce++$ ) for( $cce = y_{\text{orig}}; cce > y_{\text{orig}} - LL; cce--$ ) /\* parcours de la zone écran \*/

{  $c = ((\text{double})ce - x_{\text{orig}} + \text{zoomx} * c1) / \text{zoomx};$

$cc = -(\text{double})cce + y_{\text{orig}} + \text{zoomy} * cc1 / \text{zoomy};$

$x = 0.2; \text{cumulog} = 0;$  /\* conditions initiales \*/

for ( $i = 0; i < n; i++$ ) /\* trajectoire de  $x$  de longueur  $n$ , par exemple  $n = 500$  \*/

{ if( $i \% 2 == 0$ )  $C = cc$ ; else  $C = c$ ; /\* alternance 12 \*/

$x = C * x * (1 - x); \text{der} = C - 2 * C * x; \text{absder} = \text{fabs}(\text{der});$

$\text{cumulog} += \log(\text{absder}) / \log(2.);$

}

```

mu=cumulog/(double)n; /* exposant de Liapounov */
if (mu<0.)
{ icolor=128.+((int)(-mu*500.))%128; /* gamme de dégradés de couleurs pour mu < 0 */
  putpixel(ce,cce,coul[icolor]); /* utiliser un tableau de couleurs de 128 à 255 */
}
else
{ icolor=((int)(200.*mu))%128; /* gamme de dégradés de couleurs pour mu > 0 */
  putpixel(ce,cce, coul[icolor]); /* utiliser un tableau de couleurs de 0 à 128 */
}
}

```

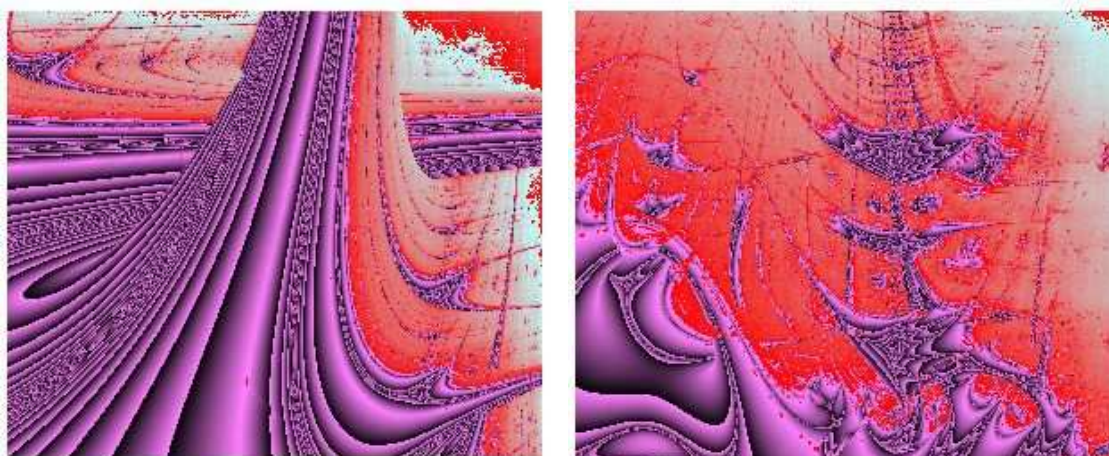


Figure 20 : Espaces de Liapounov pour l'alternance 12 à gauche, et 111111222222 à droite.

### Exercice 11 : Migrations réciproques de population

On considère une population répartie dans deux zones géographiques limitrophes A et B, avec  $x$  individus du côté A et  $y$  du côté B. Si ces deux groupes restaient autonomes, leur loi d'évolution serait celle de la suite logistique, de la forme  $x_{n+1} = c x_n (1 - x_n)$  pour l'une comme pour l'autre. Mais on ajoute un effet de diffusion, à travers la frontière. Plus le groupe des  $x$  individus est grand par rapport au groupe des  $y$  individus, plus il y a passage de la zone A à la zone B, et inversement quand  $y$  est plus grand que  $x$ . Autrement dit, on assiste à une tentative d'équilibrage évitant une surpopulation dans une zone par rapport à l'autre. Le système correspondant est :

$$\begin{cases} x_{n+1} = c x_n (1 - x_n) + d (y_n - x_n) \\ y_{n+1} = c y_n (1 - y_n) + d (x_n - y_n) \end{cases} \text{ où } d \text{ est un certain coefficient de diffusion.}$$

Dans ce qui suit, on choisit  $c = 2d + 1$ . Ainsi, lorsque  $d$  varie entre 0 et 1,  $c$  varie entre 1 et 3, ce qui évite à la suite logistique d'entrer dans la zone de chaos, ni même dans la zone des bifurcations.

1) Remplacer les suites  $x_n$  et  $y_n$  par les suites  $u_n$  et  $v_n$  telles que  $x_n = (d/c) u_n$  et  $y_n = (d/c) v_n$ , et vérifier que les nouvelles suites  $u_n$  et  $v_n$  obéissent à des relations de récurrence de la forme :

$$\begin{cases} u_{n+1} = u_n + f(x_n, y_n) \\ v_{n+1} = v_n + g(x_n, y_n) \end{cases} \text{ où } f \text{ et } g \text{ sont des fonctions à déterminer.}$$

Précisons que ce changement de suites n'a pas d'autre but que de simplifier les calculs par la suite. Les suites  $x_n$  et  $y_n$  étant les mêmes que les suites  $u_n$  et  $v_n$  à un facteur près, soit  $d/c = d/(2d + 1)$ , leur comportement sera le même.

Les relations de récurrence sur  $x_n$  et  $y_n$  deviennent :

$$(d/c)u_{n+1} = d u_n (1 - (d/c)u_n) + d (d/c)(v_n - u_n) \text{ et de même pour l'autre, ou encore :}$$

$$\begin{aligned}
u_{n+1} &= c u_n (1 - (d/c) u_n) + d (v_n - u_n) \\
&= c u_n - d u_n^2 + d (v_n - u_n) \\
&= (c - d) u_n - d u_n^2 + d v_n \\
&= (d + 1) u_n - d u_n^2 + d v_n \\
&= u_n + d (u_n - u_n^2 + v_n)
\end{aligned}$$

Finalement :

$$\begin{cases} u_{n+1} = u_n + d(u_n - u_n^2 + v_n) \\ v_{n+1} = v_n + d(u_n + v_n - v_n^2) \end{cases}$$

2) Chercher les points fixes de  $(u_n, v_n)$ , c'est-à-dire les points tels que  $u_{n+1} = u_n$  et  $v_{n+1} = v_n$  pour tout  $n$ . On trouvera deux points fixes  $(0, 0)$  et  $(2, 2)$ , et les points correspondants pour  $(x_n, y_n)$  seront  $(0, 0)$  et  $(2d/(2d+1), 2d/(2d+1))$ .

$u_{n+1} = u_n$  et  $v_{n+1} = v_n$  s'écrivent :

$$\begin{cases} u_n - u_n^2 + v_n = 0 \\ u_n + v_n - v_n^2 = 0 \end{cases}$$

ce qui impose par soustraction  $u_n^2 = v_n^2$ , soit  $u_n = \pm v_n$ . Si l'on prend  $u_n = -v_n$ , ce qui est impossible pour des populations non nulles, la première équation donne d'ailleurs  $u_n = 0$ . Et si  $u_n = v_n$ , la première équation (ou la deuxième) donne  $2 u_n - u_n^2 = 0$ ,  $u_n (2 - u_n) = 0$ . On trouve  $u_n = v_n = 0$ , ou  $u_n = v_n = 2$ . Finalement si les deux suites commencent à 0 ou à 2, elles restent constantes.

3) Montrer que pour  $(u_n, v_n)$  le point  $(0,0)$  est instable. Pour cela, on se place au voisinage de ce point, et on linéarise les deux relations en éliminant les termes négligeables, ce qui donne une matrice dont on cherchera les valeurs propres et les vecteurs propres, ce qui permettra de conclure.

Lorsque  $u_n$  et  $v_n$  sont proches de 0, les termes  $u_n^2$  et  $v_n^2$  sont négligeables. On aboutit au système linéaire :

$$\begin{cases} u_{n+1} = (1+d)u_n + d v_n \\ v_{n+1} = d u_n + (1+d)v_n \end{cases} \text{ de matrice } M = \begin{pmatrix} 1+d & d \\ d & 1+d \end{pmatrix}$$

Les vecteurs propres  $\mathbf{V}$  sont les vecteurs non nuls qui sont transformés en vecteurs  $\mathbf{V}' = M \mathbf{V}$  parallèles à  $\mathbf{V}$  sous l'effet de  $M$ , soit  $M \mathbf{V} = \lambda \mathbf{V}$ , le nombre  $\lambda$  étant le nombre appelé valeur propre associé à  $\mathbf{V}$ . Cette équation matricielle devient  $(M - \lambda I) \mathbf{V} = 0$ , et l'on ne peut trouver de vecteurs différents de 0 que si le déterminant de  $M - \lambda I$  est nul, les deux équations se réduisant alors à une seule.

$\begin{vmatrix} 1+d-\lambda & d \\ d & 1+d-\lambda \end{vmatrix} = 0$ , ou  $(1+d-\lambda)^2 - d^2 = 0$ . La résolution de cette équation donne  $\lambda = 1$  ou  $\lambda = 1 + 2d$ .

Pour  $\lambda = 1$ , avec  $\mathbf{V} (x, y)$ , le système  $M \mathbf{V} = \mathbf{V}$  conduit à l'équation  $y = -x$ . Un vecteur propre est  $\mathbf{V}_1(1, -1)$ . Pour  $\lambda = 1 + 2d$ , le système devient  $M \mathbf{V} = (1 + 2d) \mathbf{V}$  et se réduit à l'équation  $y = x$ . Un vecteur propre est  $\mathbf{V}_2(1, 1)$ . Prenons un point  $M$  proche de  $O$ . Le vecteur  $OM$  est une combinaison de  $\mathbf{V}_1$  et  $\mathbf{V}_2$  :  $OM = k \mathbf{V}_1 + k' \mathbf{V}_2$ . Le transformé  $M'$  de  $M$  est tel que  $OM' = k \mathbf{V}_1 + k' (1 + 2d) \mathbf{V}_2$ . La valeur propre  $1 + 2d > 1$  provoque une expansion au voisinage de  $O$  et le point  $M'$  s'éloigne de  $O$  par rapport à  $M$ , suivant la direction de la première bissectrice. Le point  $O$  est instable, en son voisinage.

4) Montrer que l'autre point fixe  $A(2, 2)$  est stable ou instable en son voisinage, selon les valeurs de  $d$  à préciser. Il conviendra de prendre les suites  $u'_n = u_n - 2$ , et  $v'_n = v_n - 2$ , proches de 0 au voisinage de  $O$ , et de procéder à une linéarisation comme au 3°.

Avec  $u_n = u'_n + 2$ , et  $v_n = v'_n + 2$ , les relations de récurrence du 1° deviennent :

$$\begin{cases} u'_{n+1} = u'_n + d(u'_n + 2 + v'_n + 2 - (u'_n + 2)^2) \\ v'_{n+1} = v'_n + d(u'_n + 2 + v'_n + 2 - (v'_n + 2)^2) \end{cases} \quad \text{ou} \quad \begin{cases} u'_{n+1} = (1-3d)u'_n + d v'_n \\ v'_{n+1} = d u'_n + (1-3d)v'_n \end{cases}$$
 après avoir supprimé les termes  $u_n^2$  et  $v_n^2$  négligeables par rapport aux termes du premier degré au voisinage de  $(2, 2)$ . La matrice associée est :

$$M = \begin{pmatrix} 1-3d & d \\ d & 1-3d \end{pmatrix}$$
. Par le même calcul que précédemment, on trouve qu'elle admet deux valeurs propres réelles,  $\lambda_1 = 1 - 2d$  et  $\lambda_2 = 1 - 4d$ .

Lorsque  $d$  va de 0 à 0,5,  $\lambda_1$  va de 1 à 0 et  $\lambda_2$  de 1 à  $-1$ . En valeur absolue, les deux valeurs propres sont inférieures à 1, ce qui montre que les vecteurs propres associés sont soumis à une contraction. Pour un point  $M(u_n, v_n)$  cela signifie qu'il se rapproche de  $A$ , qui est donc un point fixe stable. Lorsque  $d$  dépasse 0,5,  $|\lambda_2| > 1$ , et le point fixe  $A$  devient instable.

5) Programmer, afin d'avoir les trajectoires des points  $(x_n, y_n)$  pour diverses valeurs de  $d$ . Vérifier que les trajectoires issues de points assez proches du point fixe  $(2d/(2d+1), 2d/(2d+1))$  convergent vers ce point pour  $d < 0,5$ . Que se passe-t-il au-delà ? Et qu'est-ce que tout cela signifie pour les deux groupes de personnes ?

```
for(d=0.483; d<0.64; d+=0.01)
{
  c=2.*d+1.; zoom=400.;
  u0=0.2; v0=0.4; /* les points (x_n, y_n) sont notés ici (u_n, v_n) */
  u=u0;v=v0;filldisc(xorig+zoom*u0,yorig-zoom*v0,2,red); /* conditions initiales */
  for(i=0;i<100000;i++) /* trajectoire de la suite */
  {
    ue=xorig+zoom*u;ve=yorig-zoom*v;
    newu=c*u*(1.-u)+d*(v-u);
    newv=c*v*(1.-v)+d*(u-v);
    newue=xorig+zoom*newu;newve=yorig-zoom*newv;
    if (ue>10 && ue<699 && ve>20 && ve<579 )
    {
      if (i>98000) filldisc(ue,ve,2,black);else filldisc(ue,ve,1,red);
      if (i>99990) line(ue,ve,newue,newve,blue); /* jonctions entre deux points successifs */
    }
    u=newu;v=newv;
  }
  SDL_Flip(screen); pause();SDL_FillRect(screen,0,white);
}
```

Comme prévu, on constate que la suite converge vers le point  $(2d/(2d+1), 2d/(2d+1))$  lorsque  $d < 0,5$ . Juste au-delà de 0,5, une bifurcation se produit et le point fixe est remplacé par un cycle fixe de deux points, qui s'éloignent l'un de l'autre lorsque  $d$  augmente. Puis lorsque  $d$  dépasse une valeur proche de 0,6, les points sont remplacés par un attracteur plus complexe, formé de deux formes ovales qui grossissent quand  $d$  augmente (figure 21).

Pour les deux groupes de personnes, cela signifie que pour les faibles valeurs du coefficient de diffusion  $d$  ( $d < 0,5$ ), les deux groupes finissent par s'équilibrer à égalité, avec la disparition des migrations réciproques. Puis, pour les valeurs du coefficient  $d$  entre 0,5 et 0,6, les deux groupes se mettent à osciller à chaque étape de façon périodique. Au-delà de 0,6, les deux groupes subissent des oscillations de plus en plus fortes et non régulières. Vers  $d = 0.65$ , les oscillations sont telles que par moment chacun des groupes a un nombre d'individus presque nul. Au-delà, le phénomène n'est plus



représentatif d'une population dont le nombre d'individus doit être positif, ou bien on peut dire qu'au bout d'un nombre d'étapes fini, un des groupes va disparaître.

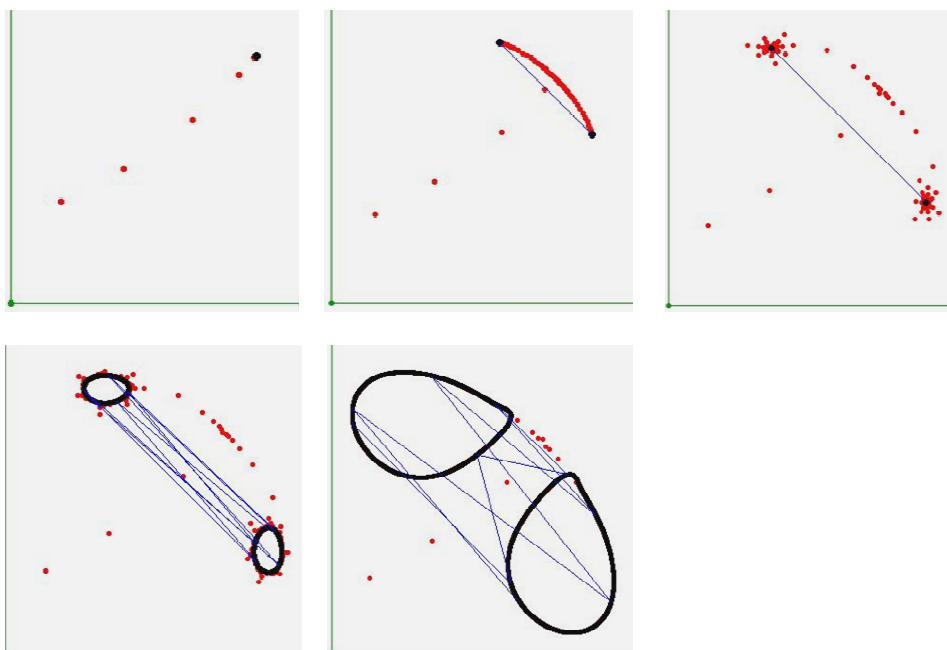


Figure 21 : Evolution de la suite  $(x_n, y_n)$  pour des valeurs croissantes de  $d$  de 0,48 à 0,64. On voit la suite converger d'abord, puis osciller sur deux points qui s'éloignent de plus en plus, puis pour  $d$  de l'ordre de 0,6 l'apparition de formes ovales qui grossissent. Les points de la suite sont *en rouge*, l'attracteur final *en noir*, et des lignes de jonctions entre quelques points successifs sont indiquées *en bleu*, montrant comment les points de la suite passent alternativement d'un ovale à l'autre de façon irrégulière.

6) Sans se préoccuper du fait que les points  $(x_n, y_n)$  aillent dans la zone négative, observer la forme étrange de l'attracteur pour  $d = 0,685$ .

Le programme précédent donne le résultat de la figure 22.

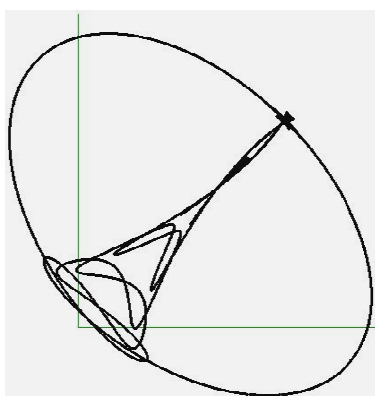


Figure 22 : Attracteur étrange pour  $d = 0,685$ .

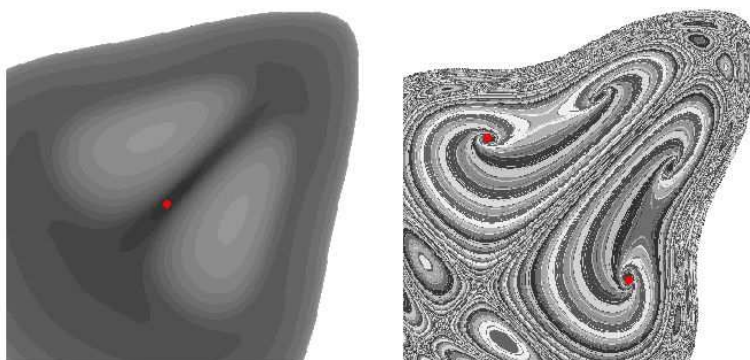
7) Dans le plan, on distingue deux types de points initiaux  $(x_0, y_0)$ , ceux dont la trajectoire s'en va à l'infini, et ceux dont la trajectoire converge vers l'attracteur. Ces derniers forment ce que l'on appelle le bassin d'attraction de l'attracteur. Programmer pour visualiser le bassin d'attraction, dans le cas où l'attracteur est un point, et dans celui où l'attracteur est un cycle de deux points.

Nous donnons ci-dessous le programme du bassin d'attraction lorsqu'il existe un point fixe, dont on connaît les coordonnées. Dans le cas d'un cycle de deux points, on doit précalculer les coordonnées



de ces points, par exemple, pour  $d = 0,6$ , on trouve  $(0,25, 0,66)$  et  $(0,66, 0,25)$ . Les résultats sont donnés sur la *figure 23*.

```
d=0.45;c=2.*d+1.;zoom=300.;
xf=2.*d/(2.*d+1); yf = xf;
for(u0=0.01;u0<1.9;u0+=0.003)for(v0=0.01;v0<1.9;v0+=0.003) /* parcours d'une zone carrée */
{ u=u0;v=v0;flag=0;
  for(i=0;i<1000;i++) /* trajectoire à partir de (u0, v0) */
  { newu=c*u*(1.-u)+d*(v-u);
    newv=c*v*(1.-v)+d*(u-v);
    u=newu;v=newv;
    if (u*u+v*v>16.) {flag=1;break;} /* cas où la trajectoire s'en va à l'infini */
    if (fabs(u-xf)<0.01 && fabs(v-yf)<0.01) {ii=i; break;} /* cas où la trajectoire converge vers (xf, yf)*/
  }
  if (flag==0)
  { ue=xorig+zoom*u0;ve=yorig-zoom*v0; putpixel(ue,ve,coul[ii%20]);
  }
}
filldisc(xorig+zoom*xf,yorig-zoom*yf,3, red); /* dessin du point fixe */
```



*Figure 23* : Bassins d'attraction pour un point fixe à gauche, et pour un cycle de deux points à droite. On a utilisé un dégradé cyclique de gris pour indiquer la vitesse de convergence.