

Introduction à Maple :

Le modèle de Day. De Solow au chaos.

Thomas Delclite et David Desmarchelier

t.delclite@gmail.com

david.desmarchelier@etudiant.univ-lille1.fr

Bernard.Dupont@univ-lille1.fr

▼ Préalable

Après avoir présenté dans les deux chapitres précédents quelques possibilités de Maple en macroéconomie et microéconomie statiques, il est logique d'aborder la dynamique économique. Le plus spectaculaire aurait été de traiter un modèle en temps continu dans la mesure où les possibilités du logiciel dans le domaine des équations différentielles sont époustouflantes. On a préféré cependant retenir dans ce chapitre le travail de deux étudiants inscrits en master 1 d'économétrie appliquée à la faculté de sciences économiques et sociales de l'université de Lille 1. Il porte sur le modèle de Day qui est un modèle de croissance en temps discret dans la veine du modèle néo-classique fondateur de Solow avec introduction d'une externalité négative (pollution). Ce choix présente deux avantages.

Le premier est heuristique. Thomas Delclite et David Demarchelier ont suivi, comme les autres étudiants de leur promotion, un cours de 30 heures de modélisation économique sous Maple au terme duquel ils devaient remettre à l'enseignant un worksheet complet sur un thème de leur choix. Ce chapitre reprend quasi intégralement leur dossier. Ses qualités en font une démonstration magistrale de ce qu'il est possible de faire après 10 semaines d'initiation et d'approfondissement mâtinées d'une bonne dose d'enthousiasme et de motivation personnelle (impliquant pas mal de nuits blanches). Certes, l'ensemble est très inspiré par l'ouvrage de Gilbert Abraham-Frois et Edmond Berrebi "Instabilité, cycles, chaos", mais il reste que la modélisation Maple leur revient totalement.

Le second avantage est technique. Les auteurs laissent entrevoir les performances de Maple dans le domaine difficile des dynamiques non linéaires en temps discret. Par touches progressives, ils abordent frontalement la théorie des bifurcation et celle du chaos, essentiellement au moyen d'une approche graphique. Cette option les a contraints à mettre au point des **procédures** qui sont des programmes écrits dans le langage Maple. Mettre au point une procédure parfaitement opérationnelle est un travail d'informaticien-programmeur nécessitant beaucoup de connaissances techniques, d'attention et de persévérance. Les cinq procédures détaillées dans ce worksheet ont été testées et retestées. Verdict : elles fonctionnent à merveille et peuvent être reprises telles quelles par d'autres économistes passionnés de bifurcation et de chaos.

▼ Introduction

"Il peut arriver que de petites différences dans les conditions initiales en engendrent de très grandes dans les phénomènes finaux ; une petite erreur sur les premières produirait une erreur

énorme sur les dernières. La prédiction devient impossible... " (Henri Poincaré, *Science et méthode*, 1909).

Cette citation du grand mathématicien français indique fort justement l'intuition que sous tend la dynamique chaotique. La sensibilité aux conditions initiales est la condition primordiale pour l'apparition de chaos. Cet aspect fondamental a été observé en 1963 par le météorologue américain Edward Norton Lorenz, à qui l'on doit la découverte du chaos et des attracteurs étranges. En 1972, il présentera ce qui deviendra "l'effet papillon" devant l'Association Américaine pour le progrès des Sciences et ce en introduisant ces travaux par une étonnante question : "Le battement d'aile d'un papillon au Brésil peut-il déclencher une tornade au Texas?"

La présence de chaos ne peut survenir que dans les systèmes dynamiques non linéaires. En temps continu, il faut que ces derniers comportent au minimum 3 équations alors qu'une seule suffit en temps discret. En économie, les systèmes dynamiques non linéaires sont extrêmement fréquents, il n'est donc pas surprenant que les économistes s'intéressent également au chaos et à la dynamique complexe en général. Deux grandes familles de modèles coexistent dans l'école néo-classique : les modèles à générations imbriquées, dans lesquels les agents vivent un laps de temps limité, et les modèles à la Ramsey, dont l'espérance de vie des individus est infinie. Ces 2 types de modèles sont concernés par la présence de chaos. Pour ce qui est des modèles à générations imbriquées, le lecteur pourra se référer à l'article de Jean-Michel Grandmont (1985) "On Endogenous Competitive Business Cycles" et pour les modèles à la Ramsey, se référer au survey de Kazuo Nishimura et Gerhard Sorger (1996) "Optimal Cycles ans Chaos: A Survey".

Dans ce worksheet, nous nous concentrerons uniquement sur le modèle de Day publié en 1982 sous le titre "Irregular growth cycles" dans l'*American Economic Review*. Il s'agit d'une variante du modèle de Solow, lui même cas particulier de modèle à la Ramsey. L'objectif affiché étant d'exhiber une dynamique chaotique, l'auteur s'est naturellement tourné vers le temps discret. La démarche est simple : reprendre le modèle de Solow en temps discret et y ajouter un effet de pollution.

On commence par exposer la structure du modèle de Day. Après avoir établi les procédures informatiques utiles, on montre par étapes progressives comment est générée une dynamique d'accumulation chaotique.

▼ Présentation du modèle de Day

Partons de l'équation caractéristique de l'accumulation de capital par tête dans le modèle en

temps discret de Solow : $k_{t+1} = \frac{s k_t^\alpha}{1+n}$. Rappelons que k_t est le capital par tête à la date t , s est le taux d'épargne (supposé constant, compris entre 0 et 1), n est le taux de croissance de la population active (supposé constant), α est l'exposant de la fonction de production Cobb-Douglas par tête (supposé constant, compris strictement entre 0 et 1).

Si on suppose que le processus de production génère de la pollution, alors on peut considérer que la productivité diminue quand l'intensité capitalistique augmente. Ainsi, nous multiplions la fonction de production par $\beta(m-k_t)^\delta$.

Day nous propose de considérer les valeurs de paramètres suivants : $m = 1$, $\delta = 1$, $\alpha = 1$. De plus,

il pose $a = \frac{\beta s}{1+n}$.

Sous ces conditions, l'équation caractéristique du modèle de Solow se transforme en équation caractéristique du modèle de Day. Cette dernière s'apparente à une fonction logistique puisqu'on a maintenant :

$$> k[t+1] := a * k[t] * (1 - k[t]);$$

(3.1)

$$k_{t+1} := a k_t (1 - k_t) \quad (3.1)$$

Dans la fonction logistique, le paramètre a est essentiel pour caractériser la dynamique du système. C'est en faisant varier ce paramètre que nous expliciterons l'apparition des cycles et du chaos. De manière similaire à Day, nous posons une restriction sur la valeur du paramètre a : il doit être compris entre 0 et 4.

```
> assume(a>0 and a<4);
```

Encart : Apparition d'un coeur de croissance

En posant $\alpha = 1$, Day suppose des rendements constants sur le seul facteur accumulable de l'économie. S'ensuit donc la formation d'un coeur de croissance (Rebello, 1990).

Le modèle devient alors un modèle de croissance endogène.

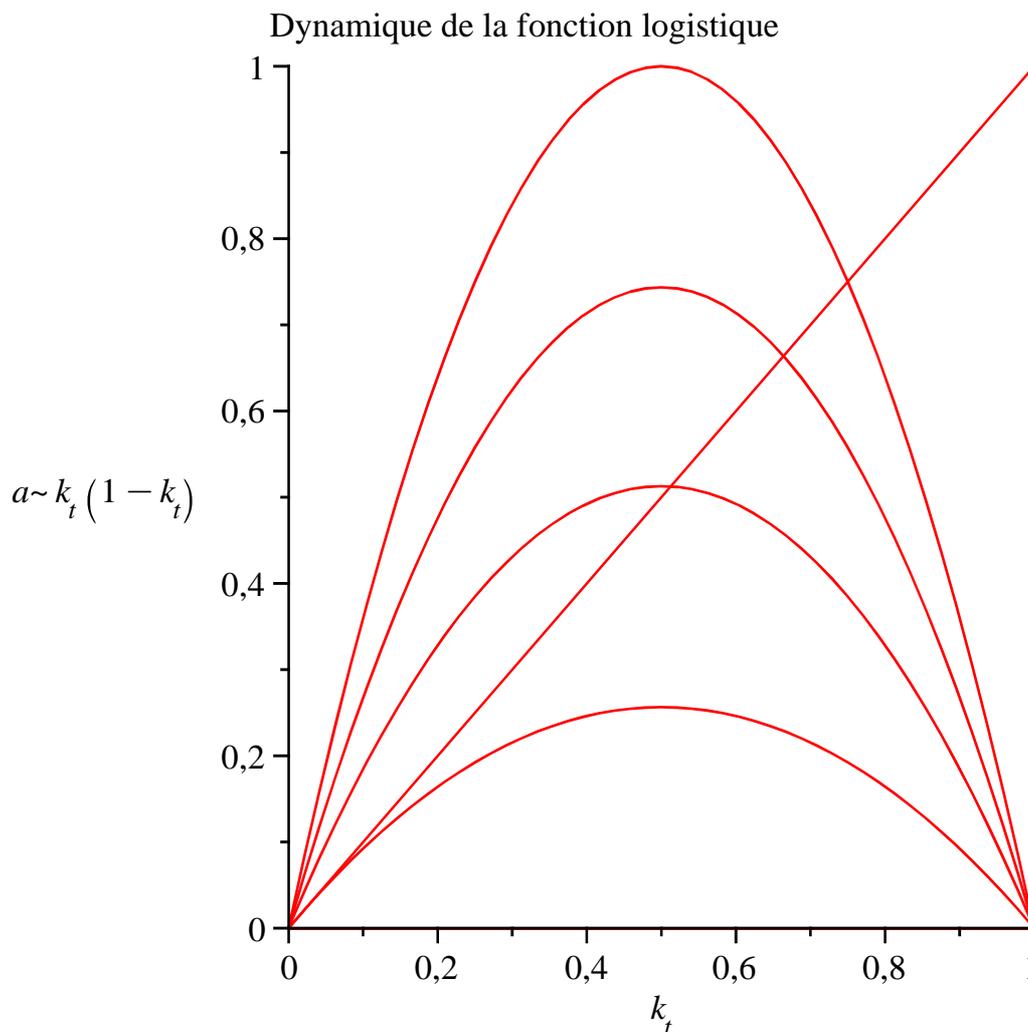
On cherche si des points fixes existent pour cette équation :

```
> solve(k[t+1]=k[t],k[t]);
```

$$0, \frac{a-1}{a} \quad (3.2)$$

Lorsque $a \leq 1$, alors seul 0 est point fixe. Pour $a > 1$, les points fixes sont au nombre de 2. L'un est trivial et l'autre est non trivial, avec la particularité que ce dernier dépend de a . Le graphique animé ci dessous présente l'évolution de la fonction logistique en fonction des valeurs de a .

```
> with(plots):
kk:=animate(plot,[k[t+1],k[t]=0..1],a=0..4,frames=40,trace=
4):
bi:=plot(k[t],k[t]=0..1):
display({kk,bi},labels=[k[t],k[t+1]],title="Dynamique de la
fonction logistique");
```



La présence d'un ou de deux points fixes pose la question de leur stabilité. Les sections qui suivent explorent ce problème.

▼ Procédures utiles

Avant d'analyser la stabilité du ou des points fixes, il nous faut d'abord présenter les procédures qui permettront d'illustrer graphiquement nos propos.

▼ Diagramme de phase

Cette procédure est inspirée de celle du worksheet de Gregory Moore, intitulé "recurrence relations and recursion", disponible sur le site de Maplesoft à l'adresse <http://www.maplesoft.com/applications/Author.aspx?mid=151&page=1>.

```
> phase := proc(k0,a,n)
  option remember;
  local f,i,segment,K,Y,graph,couleur;
```

```

  description "La procédure phase a trois arguments
obligatoires : k0, le niveau de capital initial; a,
paramètre crucial de la fonction logistique; n, le nombre
d'itérés souhaités. Elle renvoie un graphique en 2D avec
3 éléments : la fonction logistique, la bissectrice
```

principale et les itérés de la fonction sous forme de segments. Si un 4ème argument est présent, le graphique est animé selon les itérés successifs ":

```
if k0 < 0 or k0 > 1 then do
    print("Valeur du capital intial incompatible")
:# Vérification du capital initial entré
    return:
end do:end if:

if a < 0 or a > 4 then do
    print("Valeur du paramètre a incompatible"):#
Vérification du paramètre a entré
    return:
end do:end if:

if n < 0 then do
    print("Valeur du nombre d'itérés incompatible")
:# Vérification du nombre d'itérées entré
    return:
end do:end if:

f:=k->a*k*(1-k): # Rappel de la fonction de production

if nargs =4 then couleur:= COLOR(RGB,1,i/n,0): #
Détermination de la couleur
else couleur:= red:
end if:

segment[1] := plot( [[k0, 0], [k0, (f@@1)(k0)]] , k = 0.
.1,thickness = 2,color =red):
segment[2] := plot( [[k0, (f@@1)(k0)],[(f@@1)(k0),(f@@1)
(k0)]] , k = 0..1 ,thickness = 2,color = red): # Mise en
place des deux premiers segments représentant les itérés

for i from 3 to 2*n by 2 do # Définition de la boucle
    K:=(f@@((i-1)/2))(k0): # itéré (i-1)/2
    Y:=(f@@((i-1)/2+1))(k0): # itéré (i-1)/2 +1
    segment[i]:=plot([[K, K], [K, Y]], k = 0..1,
thickness = 2, color = couleur):# Segment vertical
    segment[(i+1)]:=plot([[K, Y],[Y,Y]], k = 0..1 ,
thickness = 2,color =couleur):# Segment horizontal
end do:

graph:=T->display( plot( f(k), k = 0..1, thickness = 3,
```

```

color = blue),
      plot(k, k = 0..1, thickness = 1, color =
black),
      seq(segment[i], i = 1..2*T),title=cat("a =
",convert(a,string)),labels=[k[t],k[t+1]]):#
Représentation graphique des 3 éléments

```

```

if nargs = 4 then animate(graph,[t],t=0..n,frames=(2*n
+1);
else graph(n); # Animation ou représentation statique
end if:

```

```

end proc;

```

```

phase := proc(k0, a, n)

```

(4.1.1)

```

  option remember;

```

```

  local f, i, segment, K, Y, graph, couleur;

```

```

  description

```

"La procédure phase a trois arguments obligatoires : k_0 , le niveau de capital initial; a , paramètre crucial de la fonction logistique; n , le nombre d'itérés souhaités. Elle renvoie un graphique en 2D avec 3 éléments : la fonction logistique, la bissectrice principale et les itérés de la fonction sous forme de segments. Si un 4ème argument est présent, le graphique est animé selon les itérés successifs";

```

  if  $k_0 < 0$  or  $1 < k_0$  then

```

```

    do

```

```

      print("Valeur du capital initial incompatible"); return

```

```

    end do

```

```

  end if;

```

```

  if  $a < 0$  or  $4 < a$  then

```

```

    do

```

```

      print("Valeur du paramètre a incompatible"); return

```

```

    end do

```

```

  end if;

```

```

  if  $n < 0$  then

```

```

    do print("Valeur du nombre d'itérés incompatible"); return end do

```

```

  end if;

```

```

  f :=  $k \rightarrow a * k * (1 - k)$ ;

```

```

  if nargs = 4 then couleur := COLOR(RGB, 1, i/n, 0) else couleur := red end if;

```

```

  segment[1] := plot([[k0, 0], [k0, @@(f, 1)(k0)]], k=0..1, thickness=2,
color = red);

```

```

  segment[2] := plot([[k0, @@(f, 1)(k0)], [@@(f, 1)(k0), @@(f,
1)(k0)]], k=0..1, thickness=2, color = red);

```

```

  for i from 3 by 2 to 2 * n do

```

```

    K := @@(f, 1/2 * i - 1/2)(k0);

```

```

    Y := @@(f, 1/2 * i + 1/2)(k0);

```

```

    segment[i] := plot( [[K, K], [K, Y]], k=0..1, thickness=2, color=couleur);
    segment[i+1] := plot( [[K, Y], [Y, Y]], k=0..1, thickness=2, color
    =couleur)
end do;
graph := T→plots:-display(plot(f(k), k=0..1, thickness=3, color=blue),
plot(k, k=0..1, thickness=1, color=black), seq(segment[i], i=1..2*T), title
=cat("a = ", convert(a, string)), labels=[k[t], k[t+1]]);
if nargs=4 then
    plots:-animate(graph, [t], t=0..n, frames=2*n+1)
else
    graph(n)
end if
end proc

```

Processus temporel

Cette procédure, appelée **tempo**, nous permettra de visualiser aisement l'évolution du niveau du capital durant la dynamique.

```

> tempo := proc(k0,a,n)
    option remember;
    local f,i,point,graph:

    description "La procédure tempo a trois arguments
obligatoires : k0, le niveau de capital initial; a,
paramètre crucial de la fonction logistique; n, le nombre
d'itérés souhaitées. Elle renvoie un graphique en 2D avec
l'ensemble des itérés représentés à la suite. Si un 4ème
argument est présent, le graphique est animé selon les
itérés successifs.":

    if k0 < 0 or k0 > 1 then do
        print("Valeur du capital intial incompatible")
    :# Vérification du capital initial entré
        return:
    end do:

    end if:

    if a < 0 or a > 4 then do
        print("Valeur du paramètre a incompatible"):#
    Vérification du paramètre a entré
        return:
    end do:

    end if:

    if n < 0 then do
        print("Valeur du nombre d'itérés incompatible")

```

```

:# Vérification du nombre d'itérées entré
      return:
      end do:
end if:
f:=k->a*k*(1-k): # Rappel de la fonction de production

for i from 0 to n do #Déclaration de la boucle pour créer
les points
point[i] := pointplot( [i, (f@@i)(k0)],thickness = 2,
color =red):
end do:

graph:=T->display(seq(point[i], i = 0..T),title=cat("a =
",convert(a,string)," : k0 = ",convert(k0,string)),
labels=[t,k[t]]): # Représentation graphique des itérés

if nargs = 4 then animate(graph,[t],t=0..n,frames=n+1);

else graph(n); # Animation ou représentation statique
end if:

```

```
end proc;
```

```
tempo := proc(k0, a, n)
```

(4.2.1)

```
option remember;
```

```
local f, i, point, graph;
```

```
description
```

"La procédure tempo a trois arguments obligatoires : k0, le niveau de capital initial; a, paramètre crucial de la fonction logistique; n, le nombre d'itérés souhaités. Elle renvoie un graphique en 2D avec l'ensemble des itérés représentés à la suite. Si un 4ème argument est présent, le graphique est animé selon les itérés successifs.";

```
if k0 < 0 or 1 < k0 then
```

```
do
```

```
print("Valeur du capital initial incompatible"); return
```

```
end do
```

```
end if;
```

```
if a < 0 or 4 < a then
```

```
do
```

```
print("Valeur du paramètre a incompatible"); return
```

```
end do
```

```
end if;
```

```
if n < 0 then
```

```
do print("Valeur du nombre d'itérés incompatible"); return end do
```

```
end if;
```

```
f:=k->a*k*(1-k);
```

```

for  $i$  from 0 to  $n$  do
     $point[i] := plots:-pointplot([i, @@(f, i)(k0)], thickness = 2, color = red)$ 
end do;
 $graph := T \rightarrow plots:-display(seq(point[i], i = 0 .. T), title = cat("a = ", convert(a, string)), " : k0 = ", convert(k0, string)), labels = [t, k[t]]);$ 
if  $nargs = 4$  then
     $plots:-animate(graph, [t], t = 0 .. n, frames = n + 1)$ 
else
     $graph(n)$ 
end if
end proc

```

▼ Diagramme de bifurcation

Le programme présenté ici nous a été proposé par M. Jerzy Ombach (Jerzy.Ombach@im.uj.edu.pl). Celui-ci fonctionne en deux temps :

La première procédure nommée **iter** permet de créer les valeurs des différentes itérations de notre fonction à partir de la valeur initiale k_0 , pour une valeur donnée du paramètre a . La seconde procédure **bif** permet de réaliser la première procédure **iter** pour l'ensemble des valeurs du paramètre a compris entre a_0 et a_1 et de condenser les graphiques obtenus sur un seul et même graphique.

Remarque : cette procédure ne permet pas de représenter le diagramme de bifurcation avec les valeurs exactes des états stationnaires stables ou des cycles, mais de s'en approcher le plus fidèlement possible (d'autant plus fidèlement que le nombre d'itérés demandé augmente). En outre, il ne serait pas possible de réaliser ce type de graphique pour toutes les valeurs possibles du paramètre a , dans la mesure où il existe justement des zones de chaos.

```

> iter := proc(a, k0, n, m, sym)
  option remember;
  local f, k, points, Y;

  description "Cette procédure requiert 4 paramètres obligatoires : a, le paramètre pour la fonction logistique ; k0, le niveau de capital initial ; n, le nombre d'itérés souhaité ; m, l'itéré à partir duquel on souhaite afficher les valeurs des itérés. Cette procédure renvoie un graphique en 2D, avec en abscisse la valeur du paramètre a et en ordonnée les valeurs possibles des itérés. Sur le graphique sont affichés pour la même abscisse a les (n-m) itérés souhaités.";

  f := k -> a*k*(1-k): # Rappel de la fonction logisitique

  k := k0: # Initialisation de la variable k au niveau du capital initial

```

```
points := [a,k0]: # Initialisation de la variable point,
liste des points souhaités
```

```
from 1 to n do # Définition de la boucle
    Y := f(k): # Calcul de l'itéré
    points := points,[a,Y]: # Ajout du point [a,
itéré]
    k := evalf(Y,10) # Nouvelle valeur de k
    (précision : 10 décimales)
end do:
```

```
plots[pointplot]([points][m..n],color =RED,symbol=sym): #
Représentation graphique (la variable sym est utilisée à
l'aide de la procédure bif)
```

```
end proc;
```

```
iter := proc(a, k0, n, m, sym)
```

(4.3.1)

```
option remember;
```

```
local f, k, points, Y;
```

```
description
```

"Cette procédure requiert 4 paramètres obligatoires : a, le paramètre pour la fonction logistique ; k0, le niveau de capital initial ; n, le nombre d'itérés souhaité ; m, l'itéré à partir duquel on souhaite afficher les valeurs des itérés. Cette procédure renvoie un graphique en 2D, avec en abscisse la valeur du paramètre a et en ordonnée les valeurs possibles des itérés. Sur le graphique sont affichés pour la même abscisse a les (n-m) itérés souhaités."

```
f := k → a * k * (1 - k);
```

```
k := k0;
```

```
points := [a, k0];
```

```
to n do Y := f(k); points := points, [a, Y]; k := evalf(Y, 10) end do;
```

```
plots[plots:-pointplot]([points][m..n], color = RED, symbol = sym)
```

```
end proc
```

```
> bif := proc(amin,amax,k0,n,m)
```

```
option remember:
```

```
global a:
```

```
local pas, disp,sym:
```

```
description "Cette procédure requiert 5 paramètres
obligatoires : amin et amax, la valeur minimale et
maximale du paramètre a souhaité ; k0, la valeur du
capital initial ; n, le nombre d'itérés total souhaités
et m, l'itéré à partir duquel les calculs des itérés
seront affichés. Elle renvoie un graphique en 2D avec la
concaténation des graphiques de la procédure iter, pour
un paramètre a variant de amin à amax. Si un 6ème
```

paramètre est présent, le graphique est animé selon les différentes valeurs du paramètre a " :

```
if amin < 0 or amax > 4 then do
    print("Valeur des bornes du paramètre a
incompatible"):# Vérification des bornes a entrées
    return:
end do:end if:

if amax < amin then do
    print("Valeur des bornes du paramètre a
incompatible"):# Vérification du paramètre a entré
    return:
end do:end if:

if k0 < 0 or k0 > 1 then do
    print("Valeur du capital initial incompatible")
:# Vérification du capital initial entré
    return:
end do:end if:

if m < 0 then do # Vérification du numéro d'itérée
initial entré
    print("Valeur de l'itéré initial incompatible")
:# Vérification du numéro d'itéré initial entré
    return:
end do:end if:

if n < m then do # Vérification du nombre d'itérés total
entré
    print("Itéré total inférieur a l'itéré
initial"):# Vérification du nombre d'itérés total entré
    return:
end do:end if:

pas := (amax-amin)/100: # Initialisation du pas souhaité
pour le graphique
disp := NULL: # Initialisation de la variable disp,
sequence des procédures iter
a := amin: # Initialisation de la variable a

if nargs = 6 then sym := DIAMOND # Choix du symbole pour
le graphique
    else sym := POINT:
    end if:
```

```

while a <= amax do # Déclaration de la boucle
    disp := disp,iter(a,k0,n,m,sym):
    a := a + pas:
end do:

a:='a': # Désassignation de la valeur du paramètre a

if nargs = 6 then
    plots[display]([disp],view = [amin..amax,0..1],
insequence = true,labels=['a',itéré])
    else plots[display]([disp],view = [amin..amax,
0..1],labels=['a',itéré])# Graphique statique ou animé
end if;

end proc;

```

bif := **proc**(*amin*, *amax*, *k0*, *n*, *m*)

(4.3.2)

option *remember*;

local *pas*, *disp*, *sym*;

global *a*;

description

"Cette procédure requiert 5 paramètres obligatoires : *amin* et *amax*, la valeur minimale et maximale du paramètre *a* souhaité ; *k0*, la valeur du capital initial ; *n*, le nombre d'itérés total souhaités et *m*, l'itéré à partir duquel les calculs des itérés seront affichés. Elle renvoie un graphique en 2D avec la concaténation des graphiques de la procédure *iter*, pour un paramètre *a* variant de *amin* à *amax*. Si un 6ème paramètre est présent, le graphique est animé selon les différentes valeurs du paramètre *a* ";

if *amin* < 0 **or** 4 < *amax* **then**

do

print("Valeur des bornes du paramètre *a* incompatible"); **return**

end do

end if;

if *amax* < *amin* **then**

do

print("Valeur des bornes du paramètre *a* incompatible"); **return**

end do

end if;

if *k0* < 0 **or** 1 < *k0* **then**

do

print("Valeur du capital initial incompatible"); **return**

end do

end if;

if *m* < 0 **then**

do *print*("Valeur de l'itéré initial incompatible"); **return** **end do**

end if;

```

if  $n < m$  then
    do print("Itéré total `inférieur` a l'itéré initial"); return end do
end if;
pas := 1/100 * amax - 1/100 * amin;
disp := NULL;
a := amin;
if nargs = 6 then sym := DIAMOND else sym := POINT end if;
while  $a \leq amax$  do disp := disp, iter(a, k0, n, m, sym);  $a := a + pas$  end do;
a := 'a';
if nargs = 6 then
    plots[plots:-display]( [disp], view = [amin..amax, 0..1], insequence = true,
    labels = ['a', itéré])
else
    plots[plots:-display]( [disp], view = [amin..amax, 0..1], labels = ['a', itéré])
end if
end proc

```

▼ Comparaison de processus

La procédure **compa** permettra de comparer visuellement deux dynamiques pour une même valeur du paramètre a mais avec un niveau de capital initial différent. L'objectif est de montrer la convergence ou la divergence des dynamiques.

```

> compa := proc(k0A,k0B,a,n)
    option remember;
    local f,A,B,i,graph,Afix,Aani,Bfix,Bani;

    description "La procédure compa a trois arguments
    obligatoires : k0A et k0B, le niveau de capital initial
    des deux fonctions à comparer ; a, paramètre de la
    fonction logistique; n, le nombre d'itérés souhaité. Elle
    renvoie un graphique animé en 2D avec les dynamiques
    temporelles des deux fonctions.";

    if  $k0A < 0$  or  $k0A > 1$  then do
        print("Valeur du capital initial incompatible")
    ;# Vérification du capital entré
        return;
    end do: end if;

    if  $k0B < 0$  or  $k0B > 1$  then do
        print("Valeur du capital initial incompatible")
    ;# Vérification du capital entré
        return;
    end do: end if;

    if  $a < 0$  or  $a > 4$  then do

```

```

        print("Valeur du paramètre a incompatible");#
Vérification du paramètre a entré
        return;
    end do:end if:

if n < 0 then do
    print("Valeur du nombre d'itérés incompatible")
;# Vérification du nombre d'itérés entré
    return;
    end do:end if:

f:=k->a*k*(1-k); # Rappel de la fonction de production

A[0]:=plot( [[0,0],[0,0]],thickness = 2,color =red);
B[0]:=plot( [[0,0],[0,0]],thickness = 2,color =green);

for i from 1 to n do #Déclaration de la boucle pour créer
les points
A[i] := plot( [[i-1, (f@@(i-1))(k0A)],[i, (f@@i)(k0A)]],
thickness = 2,color =red):
B[i] := plot( [[i-1, (f@@(i-1))(k0B)],[i, (f@@i)(k0B)]],
thickness = 2,color =green):
end do:

Afix:=t->display(seq(A[i],i=0..t));
Bfix:=t->display(seq(B[i],i=0..t));

Aani:=animate(Afix,[t],t=0..n,frames=n+1);
Bani:=animate(Bfix,[t],t=0..n,frames=n+1);

display(Aani,Bani,title="Comparaison des deux
dynamiques",labels=[t,k[t]]);
end proc;
compa := proc(k0A, k0B, a, n)
    option remember;
    local f, A, B, i, graph, Afix, Aani, Bfix, Bani;
    description
    "La procédure compa a trois arguments obligatoires : k0A et k0B, le niveau de
capital initial des deux fonctions à comparer ; a, paramètre de la fonction
logistique; n, le nombre d'itérés souhaité. Elle renvoie un graphique animé en 2D
avec les dynamiques temporelles des deux fonctions.";
    if k0A < 0 or 1 < k0A then
        do
            print("Valeur du capital intial incompatible"); return
        end do

```

(4.4.1)

```

end if;
if  $k_{OB} < 0$  or  $1 < k_{OB}$  then
  do
    print("Valeur du capital initial incompatible"); return
  end do
end if;
if  $a < 0$  or  $4 < a$  then
  do
    print("Valeur du paramètre a incompatible"); return
  end do
end if;
if  $n < 0$  then
  do print("Valeur du nombre d'itérés incompatible"); return end do
end if;
 $f := k \rightarrow a * k * (1 - k)$ ;
 $A[0] := plot([ [0, 0], [0, 0] ], thickness=2, color=red)$ ;
 $B[0] := plot([ [0, 0], [0, 0] ], thickness=2, color=green)$ ;
for  $i$  to  $n$  do
   $A[i] := plot([ [i - 1, @@(f, i - 1)(k_{OA}) ], [i, @@(f, i)(k_{OA}) ] ],$ 
   $thickness=2, color=red)$ ;
   $B[i] := plot([ [i - 1, @@(f, i - 1)(k_{OB}) ], [i, @@(f, i)(k_{OB}) ] ],$ 
   $thickness=2, color=green)$ 
end do;
 $Afix := t \rightarrow plots:-display(seq(A[i], i=0..t))$ ;
 $Bfix := t \rightarrow plots:-display(seq(B[i], i=0..t))$ ;
 $Aani := plots:-animate(Afix, [t], t=0..n, frames=n+1)$ ;
 $Bani := plots:-animate(Bfix, [t], t=0..n, frames=n+1)$ ;
plots:-display(Aani, Bani, title="Comparaison des deux dynamiques", labels
= [t, k[t]])
end proc

```

▼ Stabilité dans le modèle de Day

La stabilité du ou des points fixes du modèle de Day peut être étudiée d'un point de vue analytique ou graphique.

▼ Approche analytique

Nous avons démontré dans la première section que la fonction logistique admettait 1 ou 2 points fixes que nous rappelons :

$$> 0, (a-1)/a;$$

$$0, \frac{a-1}{a} \tag{5.1.1}$$

Pour qu'un point fixe du plan soit stable, il faut et il suffit que la dérivée de la fonction logistique, estimée au point considéré, soit inférieure à l'unité en valeur absolue.

On décide d'abord d'observer si le point 0 vérifie ou non cette proposition :

```
> f:=k->a*k*(1-k);
```

$$f := k \rightarrow a k (1 - k) \quad (5.1.2)$$

```
> Diff(f,k) =D(f);
eval(Diff(f,k),k=0)=eval(diff(f(k),k),k=0);
```

$$\frac{\partial}{\partial k} f = (k \rightarrow a (1 - k) - a k)$$

$$\left. \frac{\partial}{\partial k} f \right|_{k=0} = a \quad (5.1.3)$$

La valeur de cette expression dépend de la valeur du paramètre a . On a vu que a prenait ses valeurs dans l'intervalle ouvert $]0; 4[$. Aussi, il est trivial de constater que la valeur absolue de la dérivée de f estimée en 0 est inférieure à l'unité si et seulement si $a < 1$. Dans une pareille configuration, 0 est un point fixe pour la fonction logistique et ce dernier est localement stable.

Lorsque $a > 1$, il est clair que 0 est localement instable. Au même moment, lorsque 0 perd sa stabilité, un nouveau point fixe émerge, soit $\frac{a-1}{a}$. Il s'agit donc désormais d'interroger la stabilité de ce "nouveau" point fixe.

```
> eval(Diff(f,k),k=(a-1)/a)=eval(diff(f(k),k),k=(a-1)/a);
lhs(%)=simplify(rhs(%));
```

$$\left. \frac{\partial}{\partial k} f \right|_{k = \frac{a-1}{a}} = a \left(1 - \frac{a-1}{a} \right) - a + 1$$

$$\left. \frac{\partial}{\partial k} f \right|_{k = \frac{a-1}{a}} = 2 - a \quad (5.1.4)$$

Recherchons les conditions à porter sur a pour que cette expression soit comprise dans le cercle unitaire.

```
> assume(rhs(<1 and rhs(>-1);
about(a);
```

Originally a , renamed $a\sim$:

Involved in the following expressions with properties

$3-a$ assumed real

is assumed to be: `RealRange(Open(1),Open(3))`

also used in the following assumed objects

`[3-a]` assumed real

On sait donc que l'état stationnaire est localement stable pour $1 < a < 3$.

A ce stade, on peut donc dire que si $a < 1$, seul 0 est un point fixe localement stable pour la fonction logistique. Cependant, lorsque $a > 1$, la fonction logistique n'exhibe non plus 1, mais 2 points fixes. On a vu plus haut que sous la condition que $a > 1$, l'état 0 perd sa

stabilité et $\frac{a-1}{a}$ est localement stable. Cette situation est vraie pour tout a compris dans

l'intervalle ouvert $]1; 3[$.

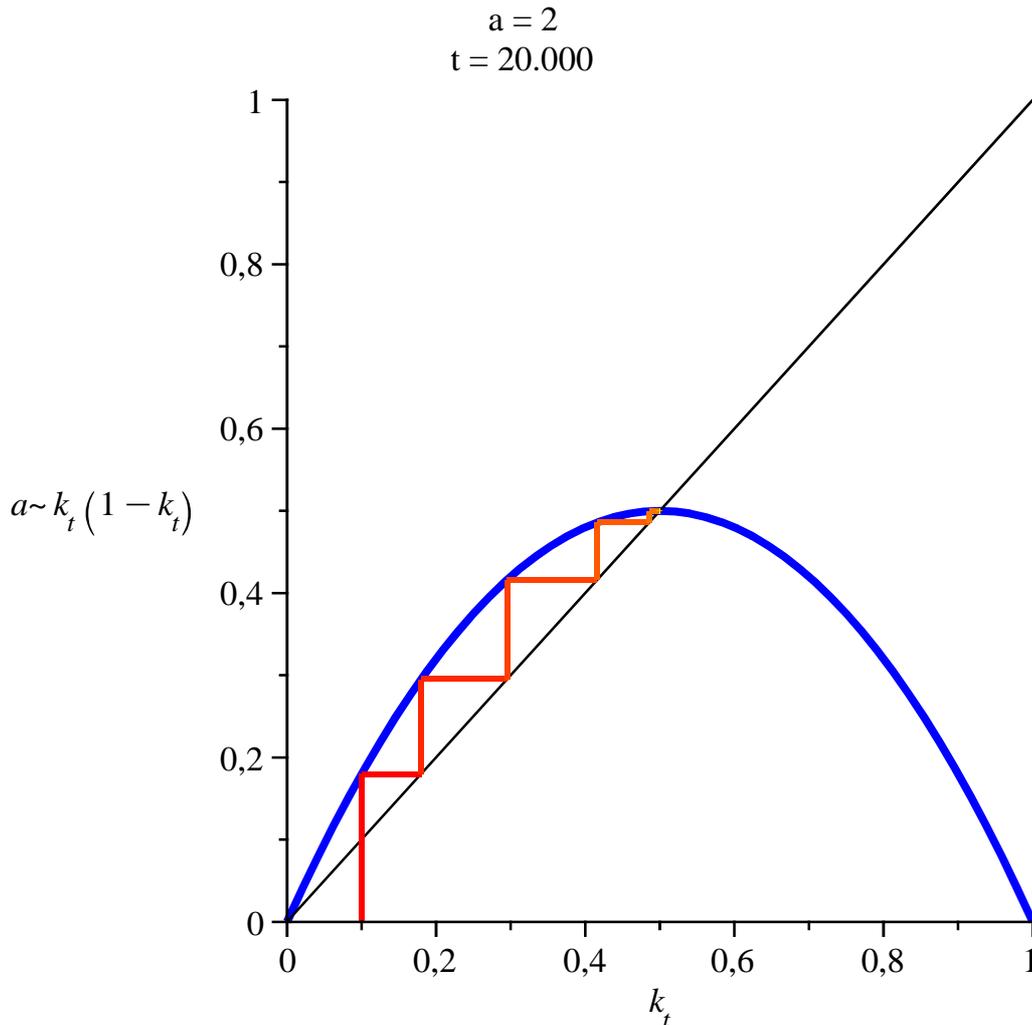
Approche graphique

Après l'étude mathématique, nous souhaitons représenter graphiquement la propriété de stabilité de l'état stationnaire $\frac{a-1}{a}$ quand a est dans l'intervalle $]1; 3[$. On utilise les

procédures phase,

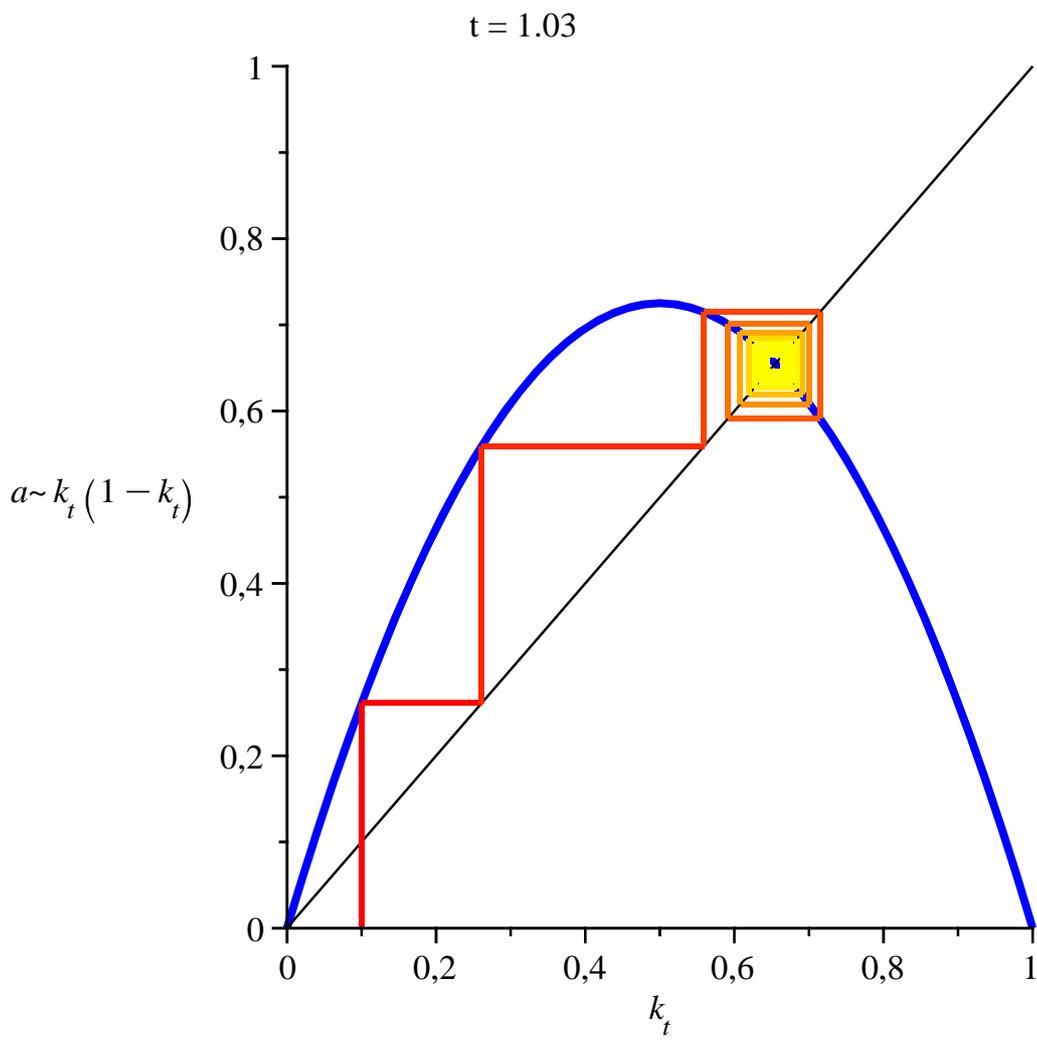
Le graphique animé suivant montre le diagramme de phase pour la fonction logistique avec un paramètre a égal à 2 et un niveau de capital initial de 0.1. Nous représentons 20 itérés successifs de cette fonction. Il montre effectivement la convergence du niveau de capital vers la valeur 0.5.

```
> phase(0.1,2,20,ani);
```



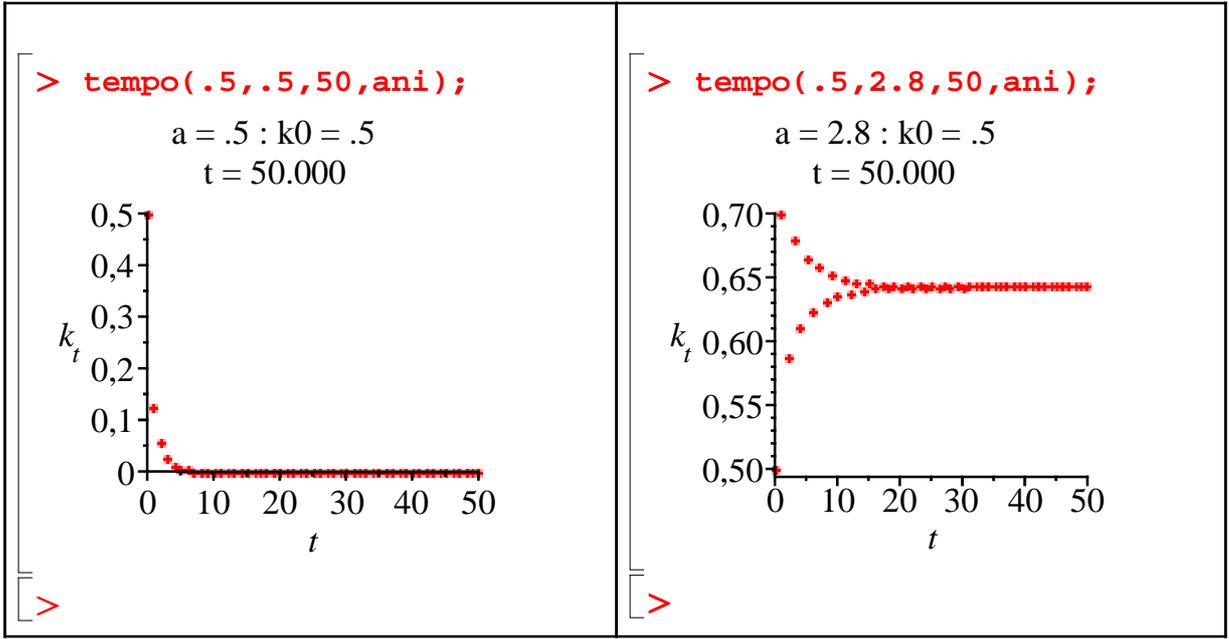
Plus général, le graphique animé suivant confirme la convergence de la fonction logistique pour toute valeur du paramètre a supérieure à 1 et inférieure à 3.

```
> animate(phase,[0.1,t,20,ani],t=1.03..2.9,frames=5);
```

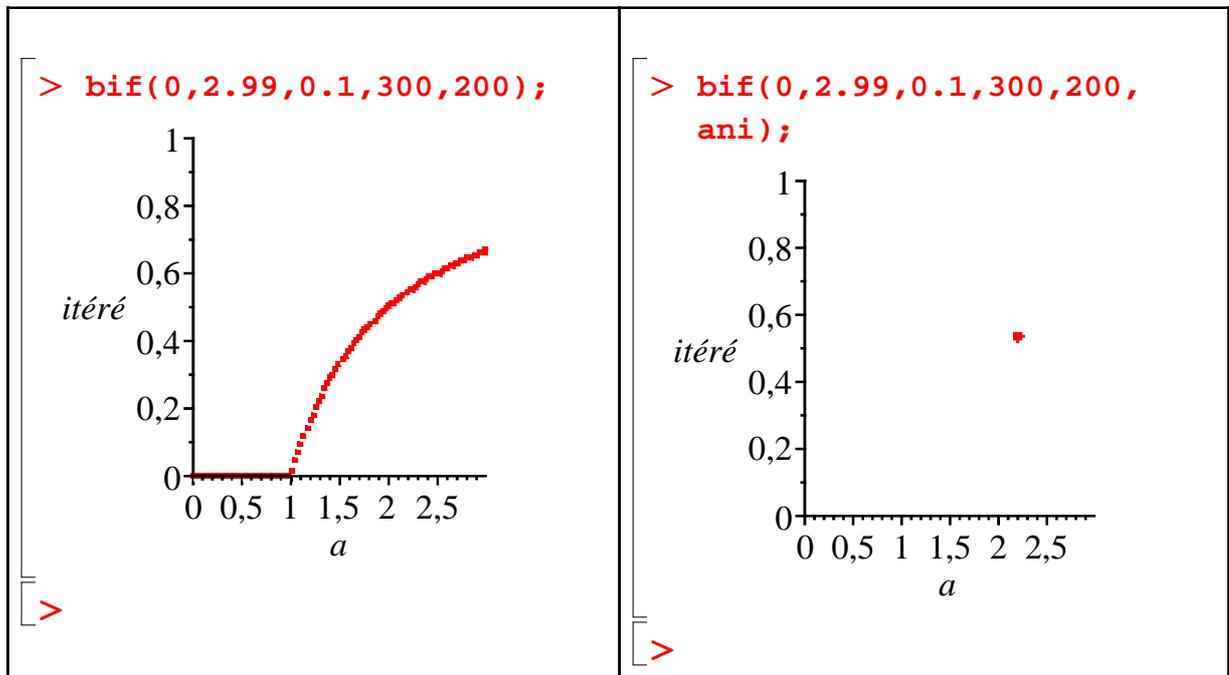


Les deux graphiques précédents ne permettent pas clairement de visualiser les valeurs du capital au fil de la dynamique. C'est dans ce but que nous réalisons les graphiques suivants à l'aide des procédures **tempo** et **bif**.

Avec **tempo**, regardons la trajectoire du capital par tête pour deux cas : $a = 0.5$ et $a = 2.8$ quand le capital initial est égal à 0.5. Dans le premier cas, il y a convergence vers l'état 0. Dans le deuxième cas, le capital par tête tend vers l'état non trivial.

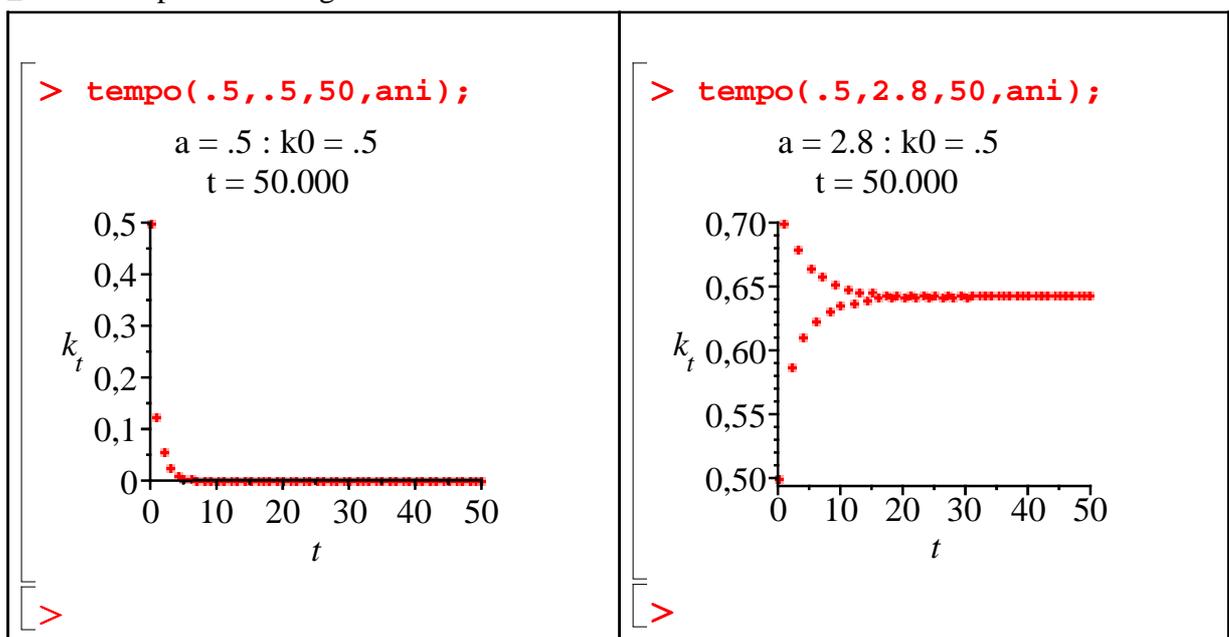


En fait, il est tout à fait possible de représenter la valeur de l'état stationnaire selon la valeur du paramètre a . C'est le diagramme de bifurcation qui nous donne ce résultat. Nous avons remarqué mathématiquement qu'une seule valeur d'état stationnaire existe pour des valeurs du paramètre a comprises entre 0 et 3 exclu. Avec la procédure **bif**, représentons le diagramme de bifurcation pour le paramètre a allant de 0 à 2,99. Le graphique de gauche est statique alors que celui de droite est animé.



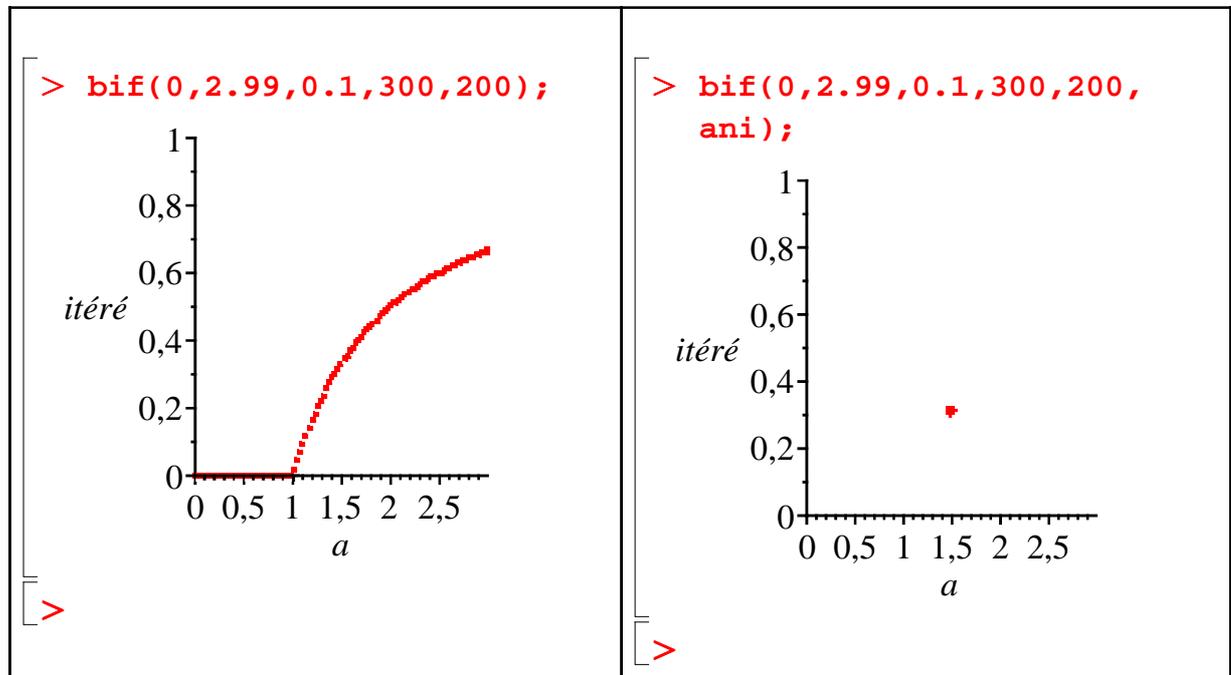
On voit bien que la dynamique converge vers 0 pour un a compris entre 0 et 1. Ensuite, lorsque a se situe dans l'intervalle ouvert $]1; 3[$, on s'aperçoit que la dynamique converge vers une valeur différente de 0. En fait, comme il a été démontré plus haut, elle converge vers $\frac{a-1}{a}$.

Les graphiques précédents ne permettent pas clairement de visualiser les valeurs du capital au fil de la dynamique. C'est dans ce but que nous réalisons les graphiques suivants. Regardons ici la dynamique de notre fonction logistique pour deux cas : $a = 0.5$ et $a = 2.8$, avec un capital initial égal à 0.5.



Il est possible de représenter la valeur de l'état stationnaire selon la valeur du paramètre a . C'est le diagramme de bifurcation qui nous donne ce résultat. Nous avons remarqué mathématiquement qu'une seule valeur d'état stationnaire existe pour des valeurs du paramètre a comprises entre 0 et 3 exclu.

Représentons le diagramme de bifurcation pour le paramètre a allant de 0 à 2,99.



On voit bien que la dynamique converge vers 0 pour un a compris entre 0 et 1. Ensuite, lorsque a se situe dans l'intervalle ouvert $]1,3[$, on s'aperçoit que la dynamique converge vers une valeur différente de 0. En fait, comme il a été démontré plus haut, elle converge vers $\frac{a-1}{a}$.

▼ Cycles

La section précédente a permis d'explorer le comportement dynamique de la fonction logistique pour a compris entre 0 et 3 (exclu). Ce qui suit a pour ambition d'explicitier le comportement dynamique de l'équation de Day lorsque a dépasse 3. Cette section est consacrée au phénomène d'apparition de cycles économiques.

▼ Cycles d'ordre 2

▼ Approche analytique

Calculons tout d'abord le premier itéré de f :

```
> f1:=f(f(k));
```

$$f1 := a^2 k (1 - k) (1 - a k (1 - k)) \quad (6.1.1.1)$$

On cherche les points fixes de l'itéré et on obtient :

```
> solve(f1=k,k);
```

$$0, \frac{a-1}{a}, \frac{\frac{1}{2} a + \frac{1}{2} + \frac{1}{2} \sqrt{a^2 - 2a - 3}}{a}, \quad (6.1.1.2)$$

$$\frac{\frac{1}{2} a \sim + \frac{1}{2} - \frac{1}{2} \sqrt{a \sim^2 - 2 a \sim - 3}}{a \sim}$$

> **simplify(%);**

$$\left[0, \frac{a \sim - 1}{a \sim}, \frac{1}{2} \frac{a \sim + 1 + \sqrt{a \sim^2 - 2 a \sim - 3}}{a \sim}, \right. \\ \left. - \frac{1}{2} \frac{-a \sim - 1 + \sqrt{a \sim^2 - 2 a \sim - 3}}{a \sim} \right] \quad (6.1.1.3)$$

Nous constatons que cette équation exhibe 4 points fixes et c'est sans surprise que nous reconnaissons les 2 premiers, déjà points fixes de f . On a déjà démontré que si a était supérieur ou égal à 3, alors aucun des 2 premiers point fixe n'était stable. On va donc concentrer nos efforts sur les 2 derniers :

> **%[3], %[4];**

$$\frac{1}{2} \frac{a \sim + 1 + \sqrt{a \sim^2 - 2 a \sim - 3}}{a \sim}, - \frac{1}{2} \frac{-a \sim - 1 + \sqrt{a \sim^2 - 2 a \sim - 3}}{a \sim} \quad (6.1.1.4)$$

Nommons ces deux valeurs :

> **C1:= %[1];**

C2:= %%[2];

$$C1 := \frac{1}{2} \frac{a \sim + 1 + \sqrt{a \sim^2 - 2 a \sim - 3}}{a \sim} \\ C2 := - \frac{1}{2} \frac{-a \sim - 1 + \sqrt{a \sim^2 - 2 a \sim - 3}}{a \sim} \quad (6.1.1.5)$$

C'est une propriété remarquable qui s'offre à nous. En effet, il apparaît que $f(C1) = C2$ et que $f(C2) = C1$!

> **f(C1)-C2;**

'f(C1)-C2'=simplify(%);

$$\frac{1}{2} (a \sim + 1 + \sqrt{a \sim^2 - 2 a \sim - 3}) \left(1 - \frac{1}{2} \frac{a \sim + 1 + \sqrt{a \sim^2 - 2 a \sim - 3}}{a \sim} \right) \\ + \frac{1}{2} \frac{-a \sim - 1 + \sqrt{a \sim^2 - 2 a \sim - 3}}{a \sim} \\ f(C1) - C2 = 0 \quad (6.1.1.6)$$

> **f(C2)-C1;**

'f(C2)-C1'=simplify(%);

$$- \frac{1}{2} (-a \sim - 1 + \sqrt{a \sim^2 - 2 a \sim - 3}) \left(1 + \frac{1}{2} \frac{-a \sim - 1 + \sqrt{a \sim^2 - 2 a \sim - 3}}{a \sim} \right) \\ - \frac{1}{2} \frac{a \sim + 1 + \sqrt{a \sim^2 - 2 a \sim - 3}}{a \sim} \\ f(C2) - C1 = 0 \quad (6.1.1.7)$$

On peut donc conclure à l'émergence d'un cycle d'ordre 2 et c'est inévitablement que se pose la question de la stabilité de ce dernier.

Dans le but de discriminer les valeurs de a pour lesquelles le cycle $(C1; C2)$ est stable ou instable, nous cherchons à déterminer pour quelles valeurs de a les états $C1$ et $C2$ sont stables simultanément pour $f^2 = f \circ f$.

Comme $f(C1) = C2$, alors $f^2(C1) = f'[C2]f'(C1)$.

Le calcul du membre de droite de cette expression donne :

```
> m:=eval(diff(f(k),k),k=C2)*eval(diff(f(k),k),k=C1);
```

$$m := \left(\left(1 + \frac{1}{2} \frac{-a\sim - 1 + \sqrt{a\sim^2 - 2 a\sim - 3}}{a\sim} \right) a\sim - \frac{1}{2} a\sim - \frac{1}{2} + \frac{1}{2} \sqrt{a\sim^2 - 2 a\sim - 3} \right) \left(\left(1 - \frac{1}{2} \frac{a\sim + 1 + \sqrt{a\sim^2 - 2 a\sim - 3}}{a\sim} \right) a\sim - \frac{1}{2} a\sim - \frac{1}{2} - \frac{1}{2} \sqrt{a\sim^2 - 2 a\sim - 3} \right) \quad (6.1.1.8)$$

Ou encore, après simplification :

```
> 'm'=simplify(%);
```

$$m = 4 - a\sim^2 + 2 a\sim \quad (6.1.1.9)$$

En résolvant cette équation en a , on trouve que :

```
> solve(m>-1,a);
```

```
solve(m<1,a);
```

$$\text{RealRange}(\text{Open}(1 - \sqrt{6}), -1), \text{RealRange}(3, \text{Open}(1 + \sqrt{6})) \quad (6.1.1.10)$$

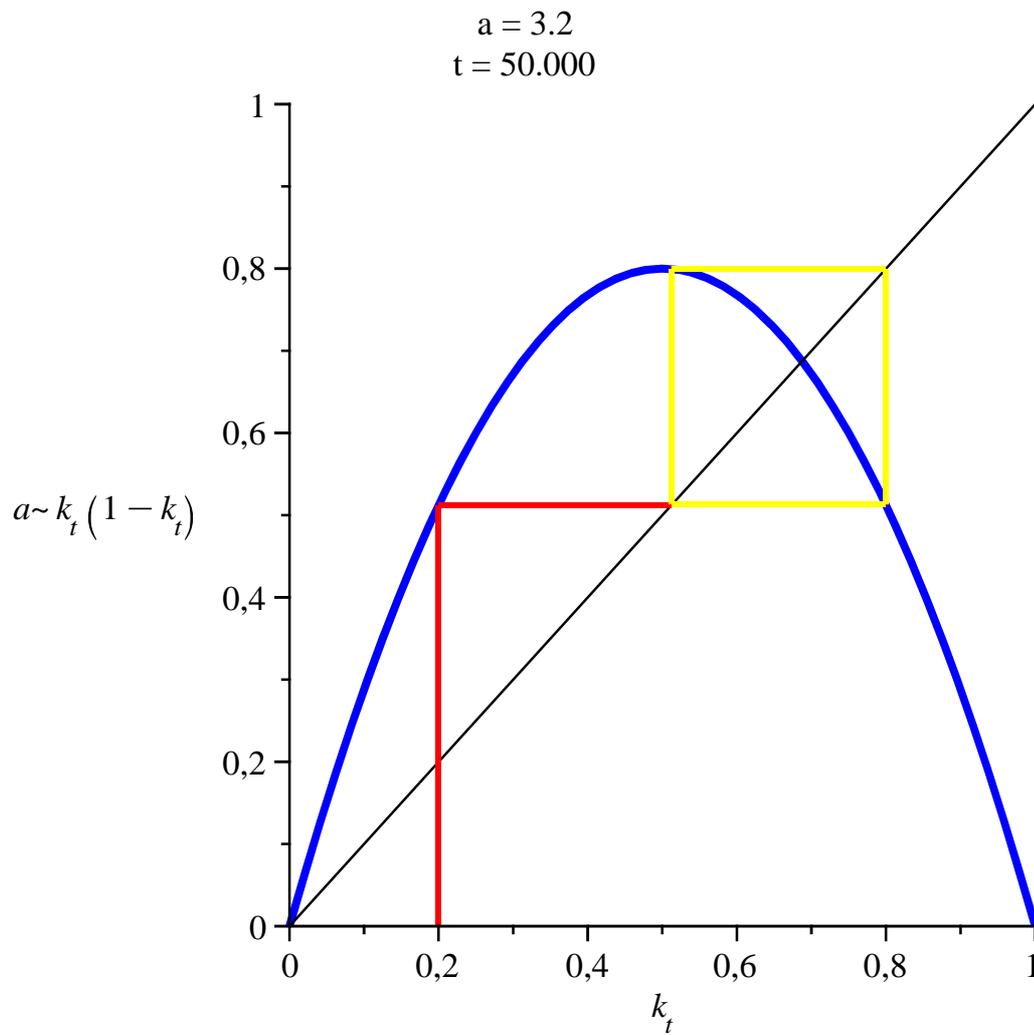
Puisque $a > 0$, on trouve que $|f^2(C1)| < 1$ si et seulement si a est compris entre 3 et $1 + \sqrt{6}$. Donc, si l'on dépasse $1 + \sqrt{6}$, le cycle d'ordre 2 n'est plus stable.

▼ Approche graphique

Nous allons chercher à représenter graphiquement la dynamique cyclique de la fonction logistique.

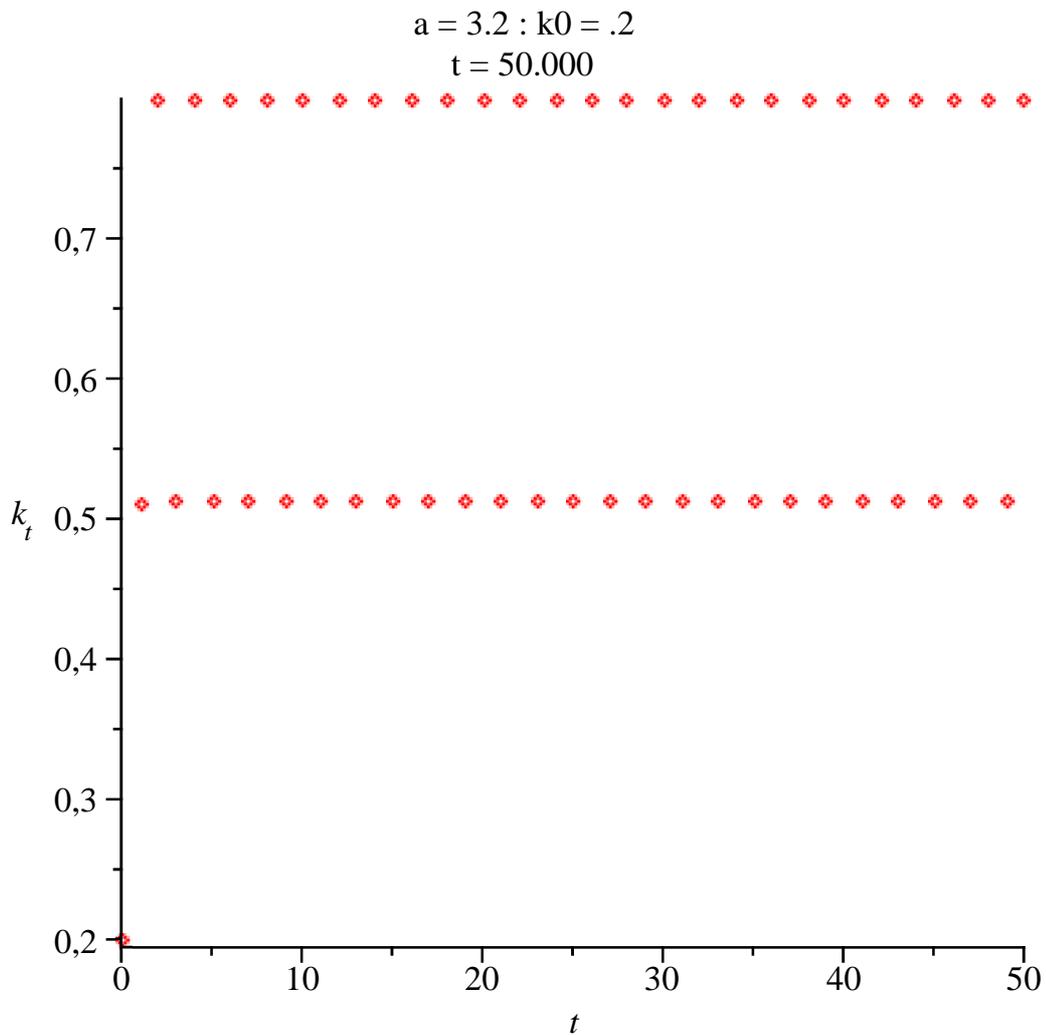
Ci dessous, on représente à l'aide de la procédure **phase** la dynamique de la fonction logistique pour $a = 3.2$. C'est sans surprise qu'un cycle stable d'ordre 2 émerge. Malgré 50 itérés, on peut remarquer que les valeurs cycliques du niveau de capital ne changent plus après le deuxième itéré.

```
> phase(0.2,3.2,50,ani);
```

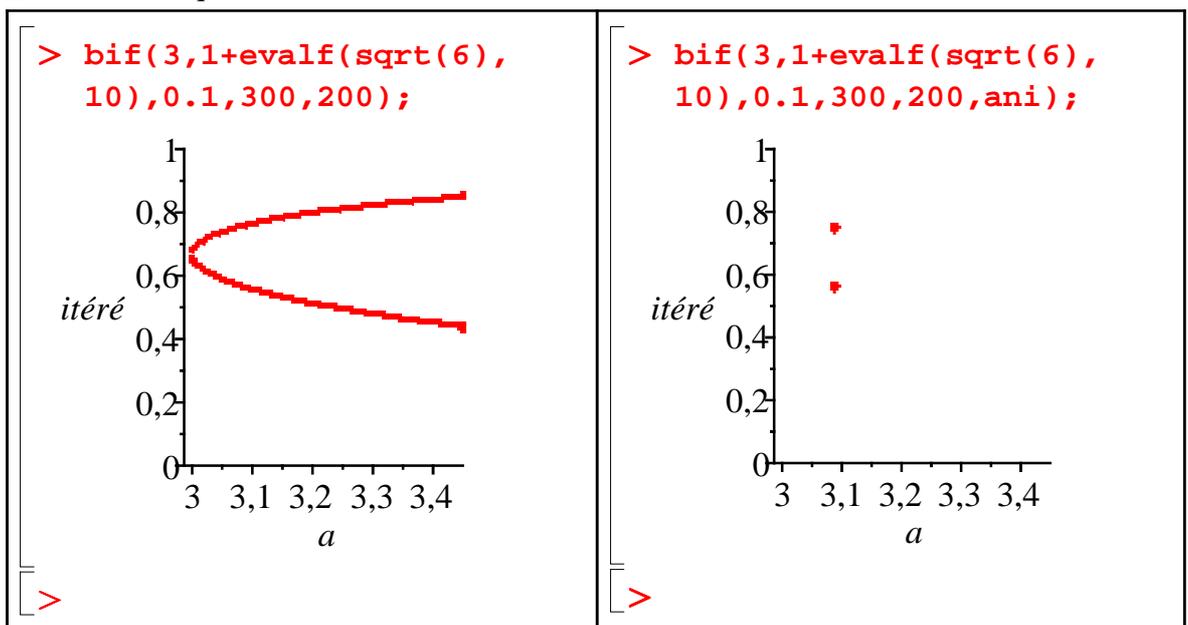


Il est possible d'observer ce phénomène à l'aide de la procédure **tempo** qui représente le processus temporel, pour une fonction logistique similaire à celle du graphique précédent.

> tempo(0.2,3.2,50,ani);



Pour des valeurs du paramètre a comprises entre 3 et $1+\sqrt{6}$ (exclu), nous observons un cycle d'ordre 2. Regardons les valeurs des points fixes pour les différentes valeurs du paramètre a grâce à la procédure **bif** qui fournit le diagramme de bifurcation dans ses versions statique et animée.



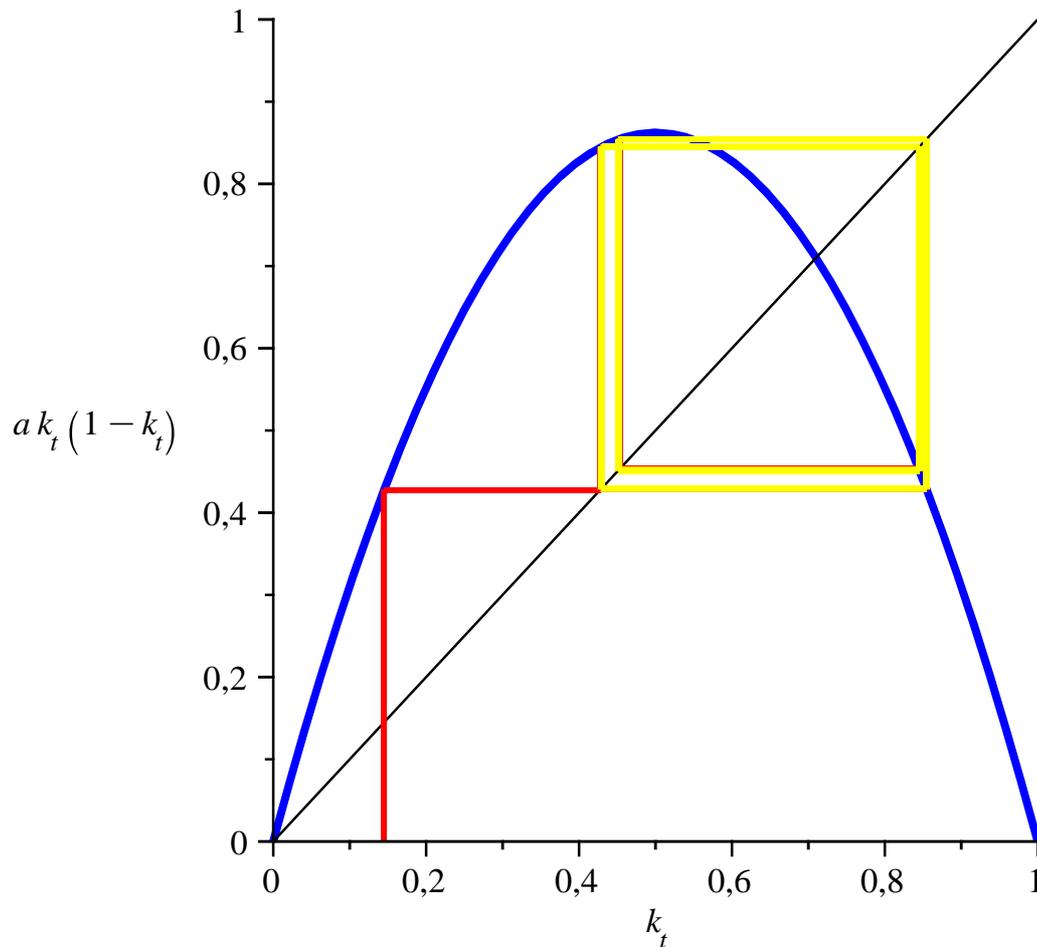
Cycles d'ordre 4

Pour le cycle d'ordre 4, nous n'aborderons pas ici l'approche mathématique. Nous resterons donc à une visualisation graphique du phénomène. Pour plus de détails, on consultera avec profit Abraham-Frois et Berrebi (1995).

Pour une valeur du paramètre a égale à 3.4495, on peut remarquer l'existence d'un cycle d'ordre 4. Observons le diagramme de phase pour ce paramètre.

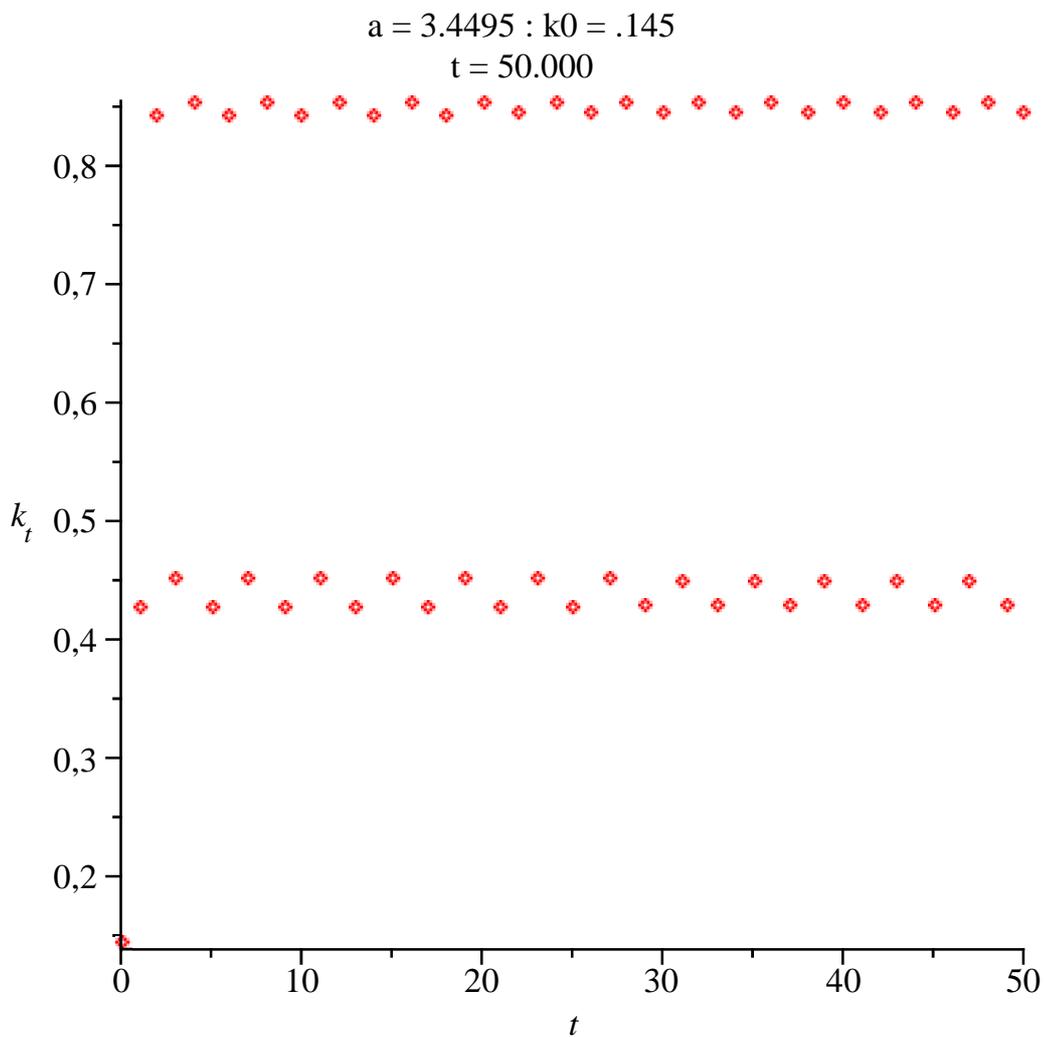
```
> phase(0.145, 3.4495, 50, ani);
```

$a = 3.4495$
 $t = 50.000$

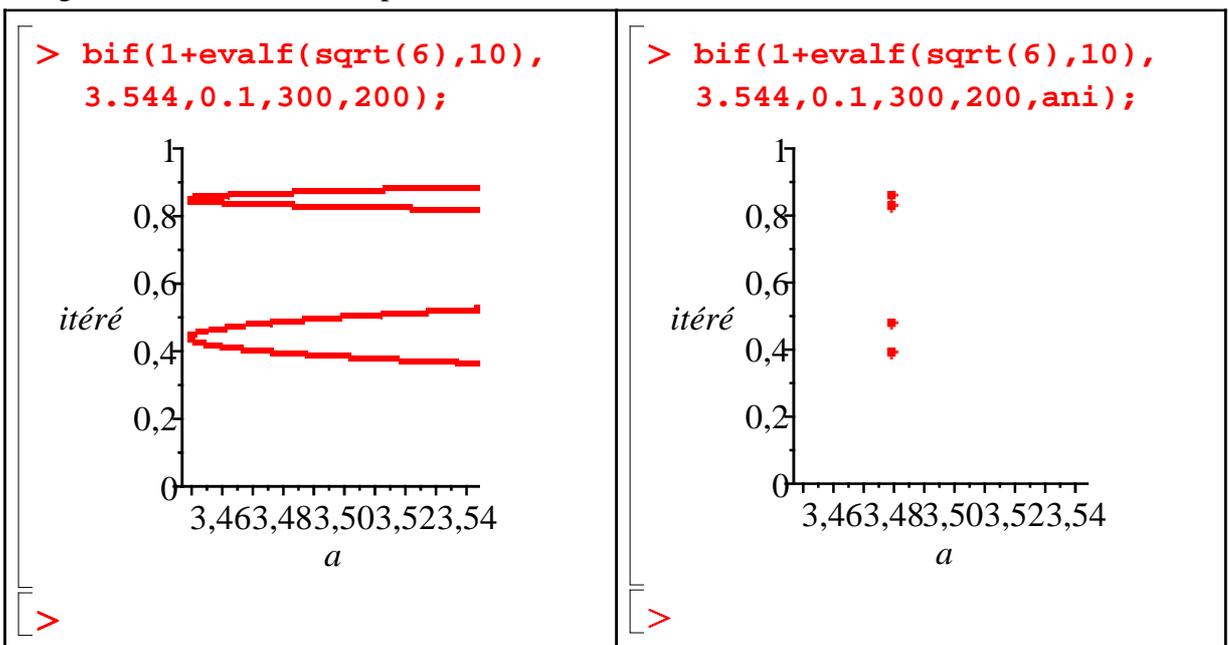


Les valeurs du capital se répètent donc toutes les 4 itérés. Nous pouvons tracer la trajectoire correspondante à l'aide de `tempo`.

```
> tempo(0.145, 3.4495, 50, ani);
```



Dans la sous-section précédente, nous nous étions arrêtés sur une valeur du paramètre a égale à $1 + \sqrt{6}$, valeur pour laquelle la stabilité du cycle d'ordre 2 s'arrête. Continuons notre diagramme de bifurcation à partir de cette valeur.

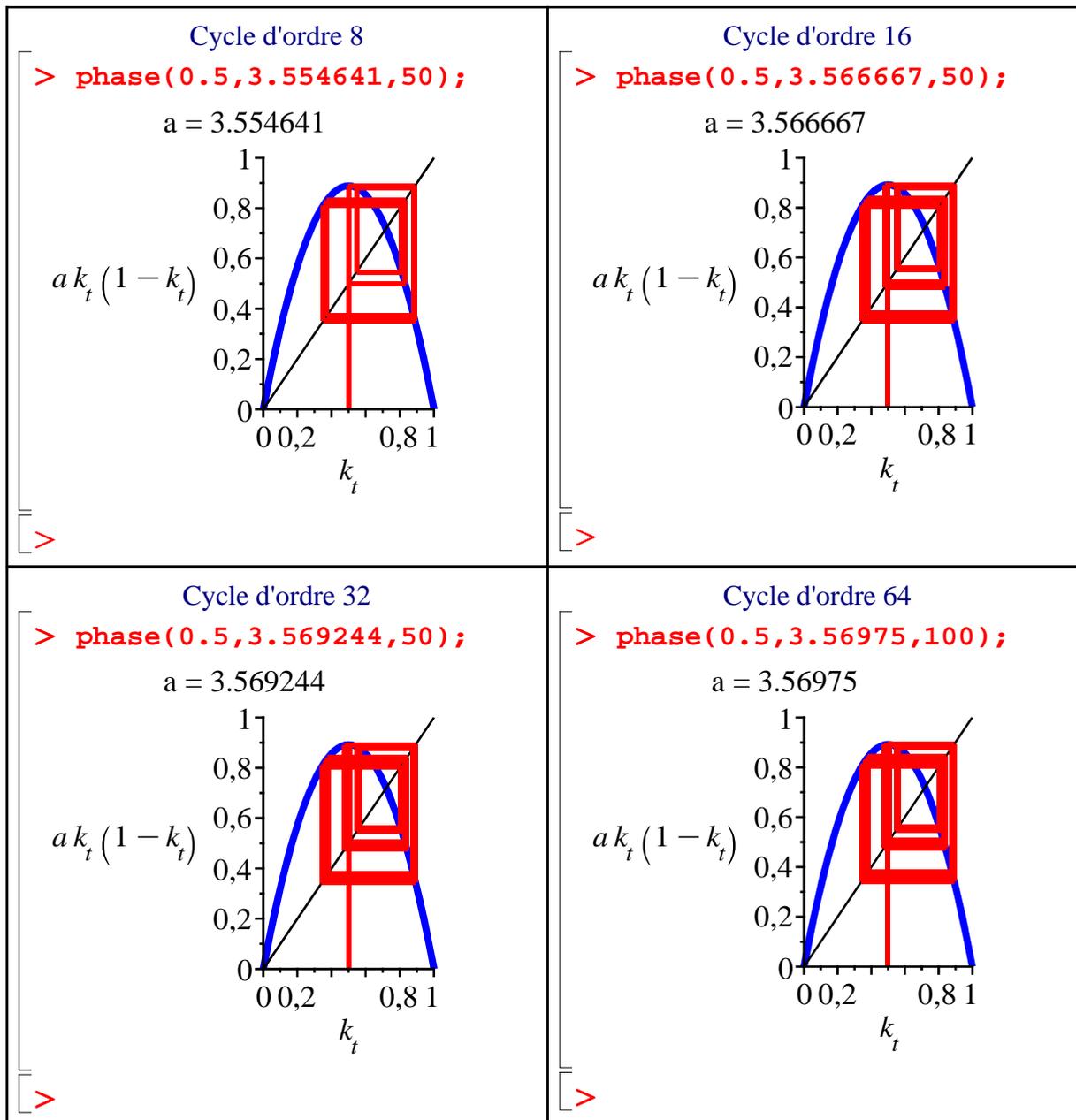


On remarque nettement l'existence de quatre valeurs stables pour la fonction logistique pour des valeurs du paramètre a comprises entre $1 + \sqrt{6}$ et 3.544.

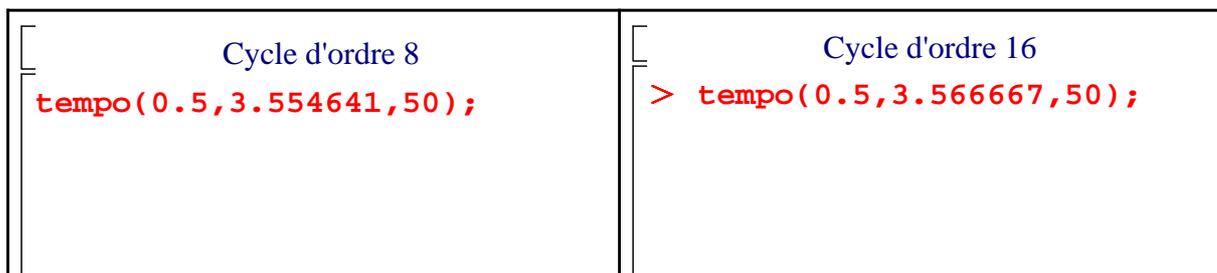
Au delà de cette valeur, le cycle se dédouble à nouveau. Passons à l'examen graphique des dynamiques cycliques d'ordre supérieur à 4.

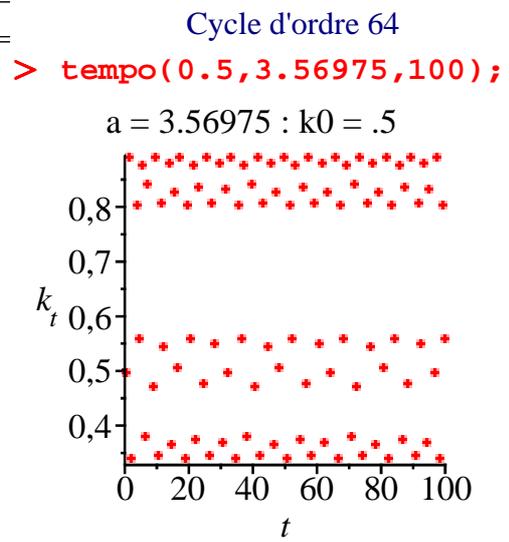
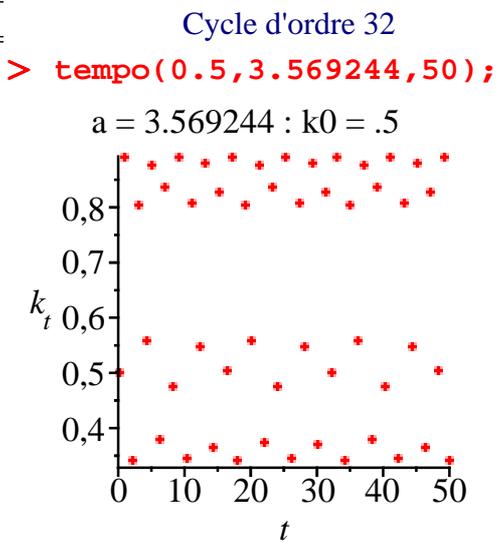
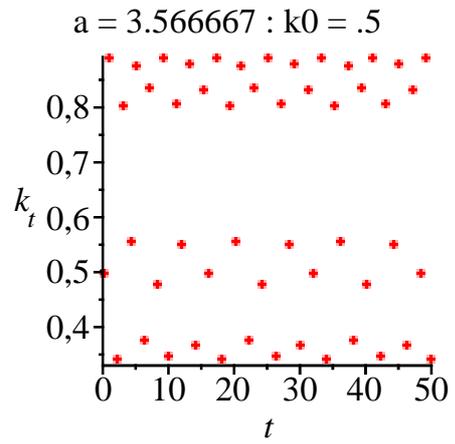
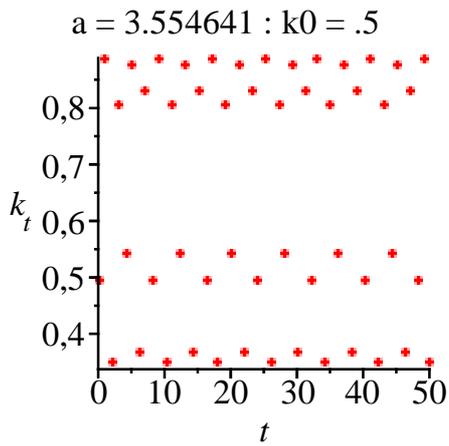
Cycles d'ordre pair supérieur

Regardons tout d'abord l'allure des diagrammes de phase pour des cycles d'ordre 8, 16, 32 ou 64.

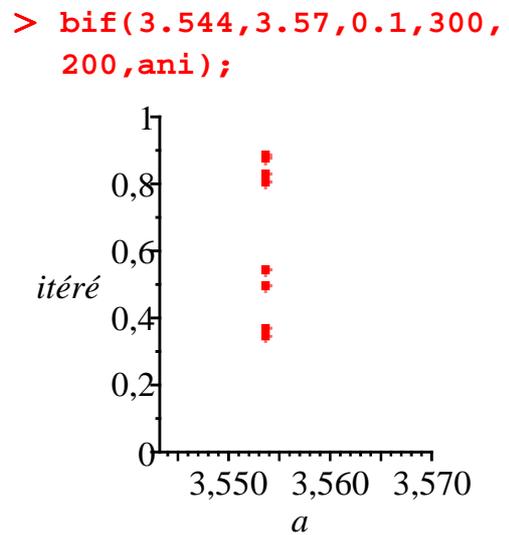
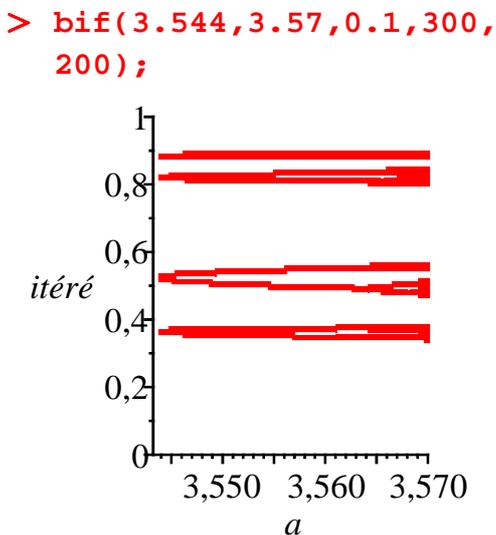


Malheureusement, la visualisation des cycles n'est pas très claire, notamment car les valeurs du capital restent très proches. Essayons alors de voir si les graphiques de processus temporels améliorent la visibilité des cycles.





Le moins qu'on puisse dire est que le problème de visibilité n'est pas vraiment amélioré. Nous admettons alors l'approche mathématique qui prouve la démultiplication des cycles stables au fil de la progression du paramètre a . Toutefois, grâce au diagramme de bifurcation, il est possible de se convaincre que les états stationnaires se multiplient mais restent stables.



Au vu de ses résultats, la question suivante se pose : existe t-il une dynamique chaotique pour une logistique "à la Day"? En effet, s'il est clair que les cycles se dédoublent continuellement au fil de l'accroissement du paramètre a , y-a-t-il une valeur de ce paramètre à partir de laquelle régularités cycliques s'évanouissent pour donner naissance au chaos?

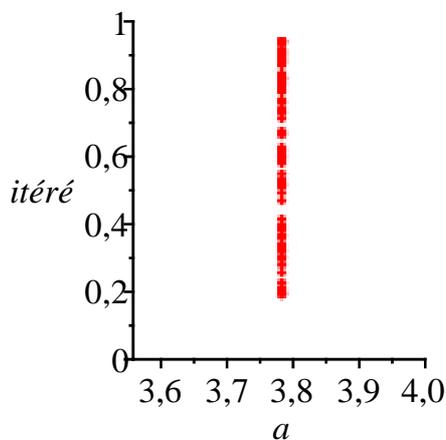
Chaos

Cette section aborde la question des dynamiques chaotiques générées par le modèle de Day. Dans un premier temps, nous montrons que le théorème de Sarkovskii est vérifié pour la dynamique d'accumulation du capital avec effets polluants pervers. Dans un second temps, on illustre le phénomène d'existence de trajectoires chaotiques.

Cycles d'ordre 3

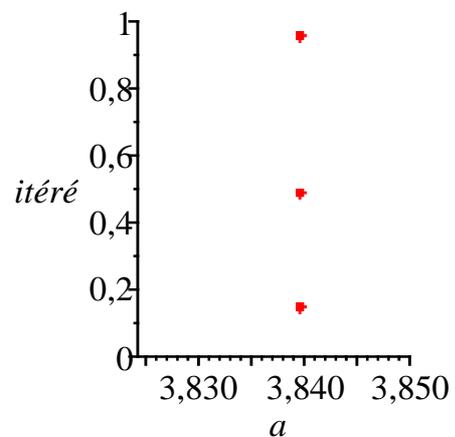
Le théorème de Sarkovskii énonce que, si le nombre de cycles que rencontre un processus dynamique est fini, alors l'ordre de tous ces cycles est pair. A l'inverse, si la dynamique présente des cycles d'ordres impair, alors le nombre de cycles possible est infini et on en conclut à l'existence de chaos. En d'autres termes, si nous montrons qu'un cycle d'ordre 3 est présent dans la dynamique de Day, alors nous pourrions prouver l'existence de chaos. Dans ce qui suit, nous nous contenterons d'une approche exclusivement graphique.

```
> bif(3.57,4,0.1,300,200,
  anime);
```



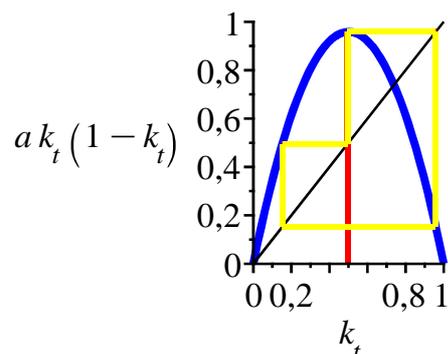
```
>
```

```
> bif(3.825,3.85,0.1,300,
  200,ani);
```



```
> phase(.5,3.835,50,ani);
```

```
a = 3.835
t = 50.000
```



```
>
```

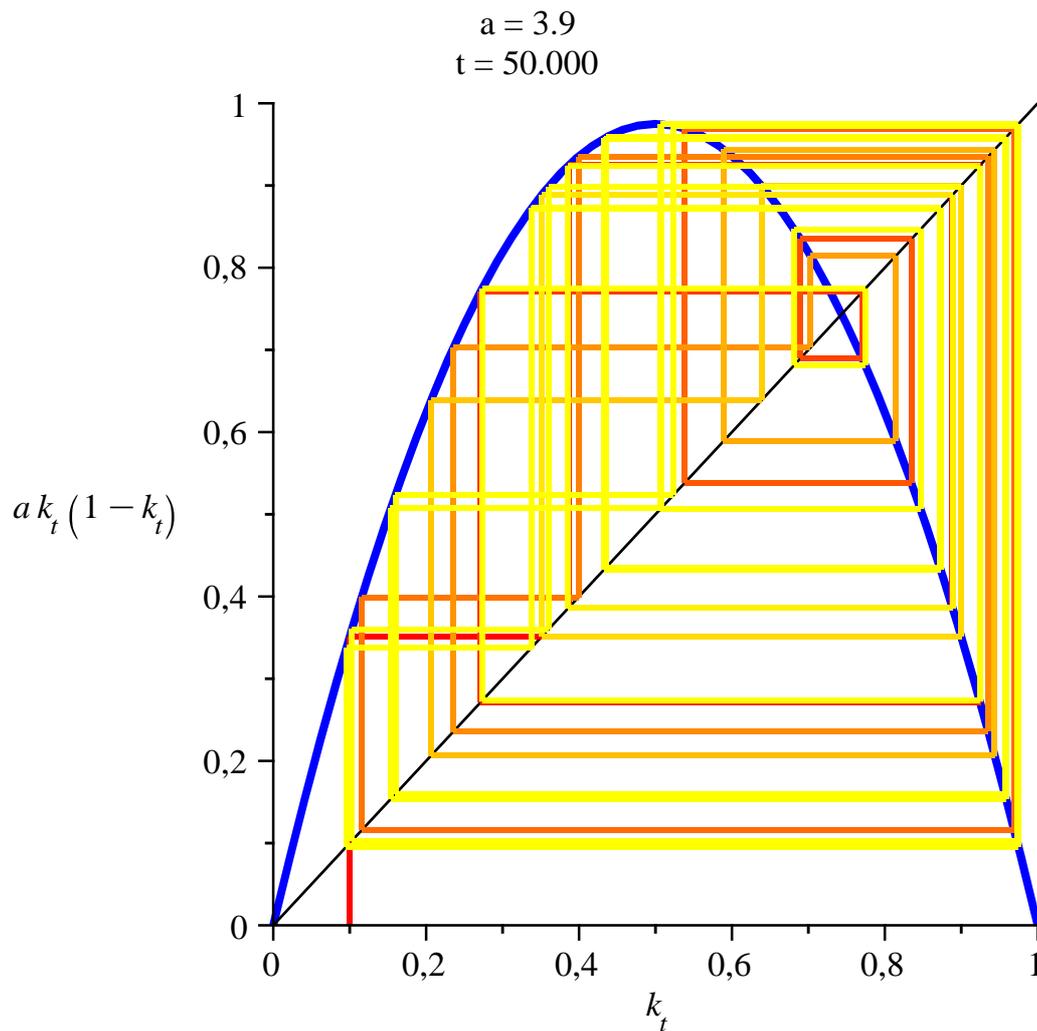
En tâtonnant (graphiques 1 et 2), on constate qu'il existe au moins une valeur de a (soit $a = 3.835$) pour laquelle la loi d'évolution exhibe un cycle d'ordre 3 (graphique 3). D'après le théorème de Sarkovskii, on peut en conclure l'existence de chaos.

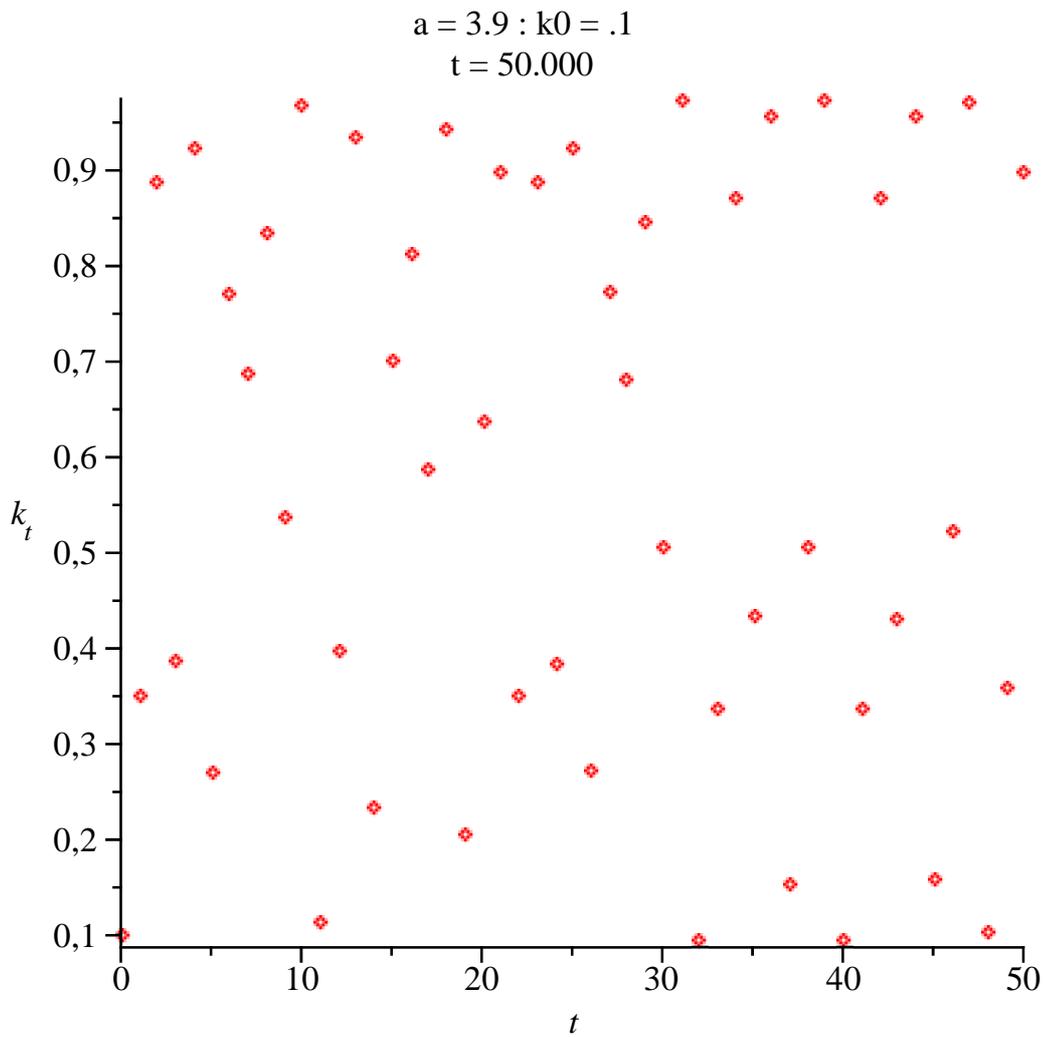
Représentations graphiques du chaos

Les graphiques qui suivent illustrent la présence de chaos dans le modèle de Day, que ce soit pour une valeur particulière de a (en l'occurrence, 3.9 dans les deux premiers graphiques) ou une plage de valeurs (entre 3.57 et 4 pour les deux derniers graphiques).

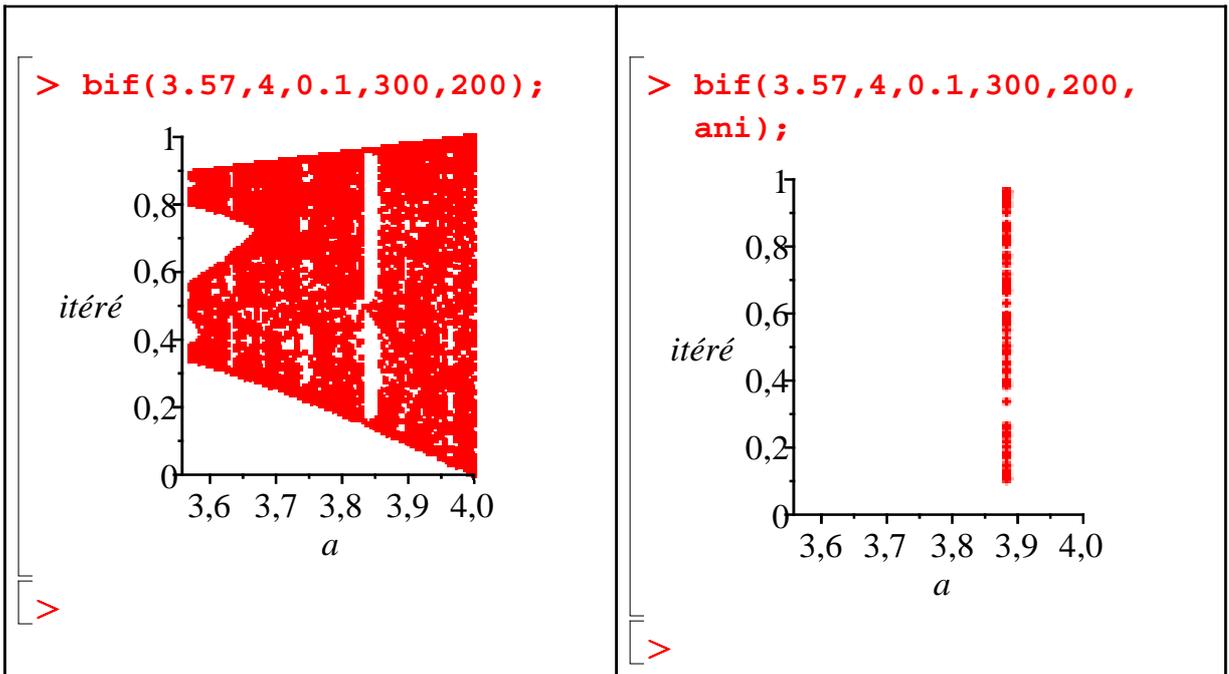
Quand $a = 3.9$, le diagramme de phase et la représentation d'une trajectoire montrent que les évolutions du stock de capital sont clairement chaotiques.

```
> phase(0.1, 3.9, 50, ani);  
tempo(0.1, 3.9, 50, ani);
```





En balayant les valeurs entre 3.57 et 4 pour a , le diagramme de bifurcation fait nettement apparaître l'existence de chaos même s'il existe des fenêtres de stabilité (on retrouve d'ailleurs le cycle d'ordre 3).



Synthèse

Pour synthétiser, nous choisissons de montrer quatre diagrammes de phase, pour des valeurs du paramètre "a" permettant de visualiser les dynamiques que nous avons présentées.

Pour synthétiser les résultats auxquels nous sommes parvenus, nous choisissons de montrer quatre diagrammes de phase animés, chacun correspondant à une dynamique remarquable dépendant de la valeur prise par le paramètre a : 0 est le seul état stationnaire stable ($a = 0.5$);

$\frac{a-1}{a}$ est le seul état stationnaire stable ($a = 2.5$); présence d'un cycle d'ordre 4 ($a = 3.4495$);

chaos ($a = 3.8$).

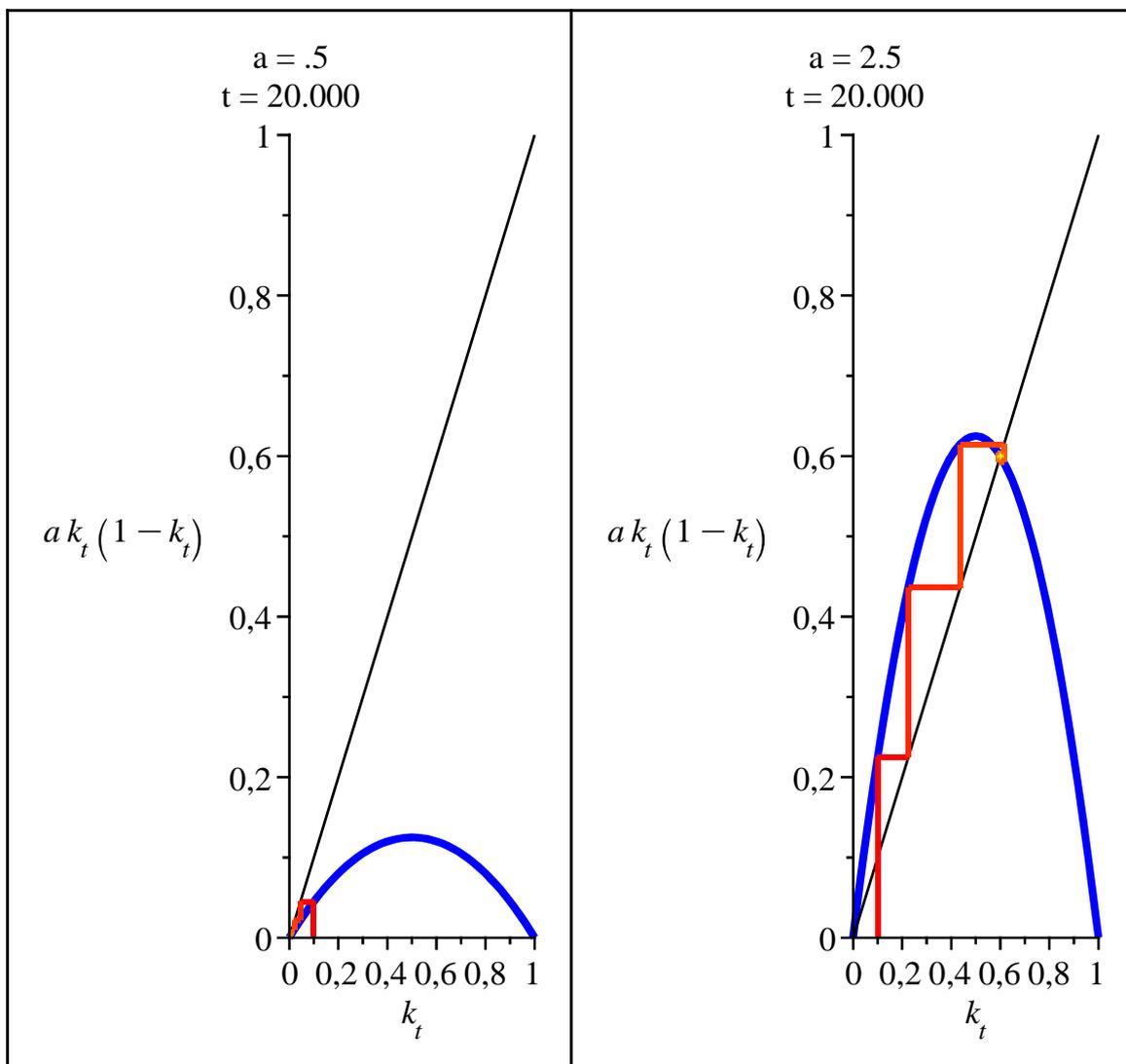
```
> A:=array(1..2,1..2):
```

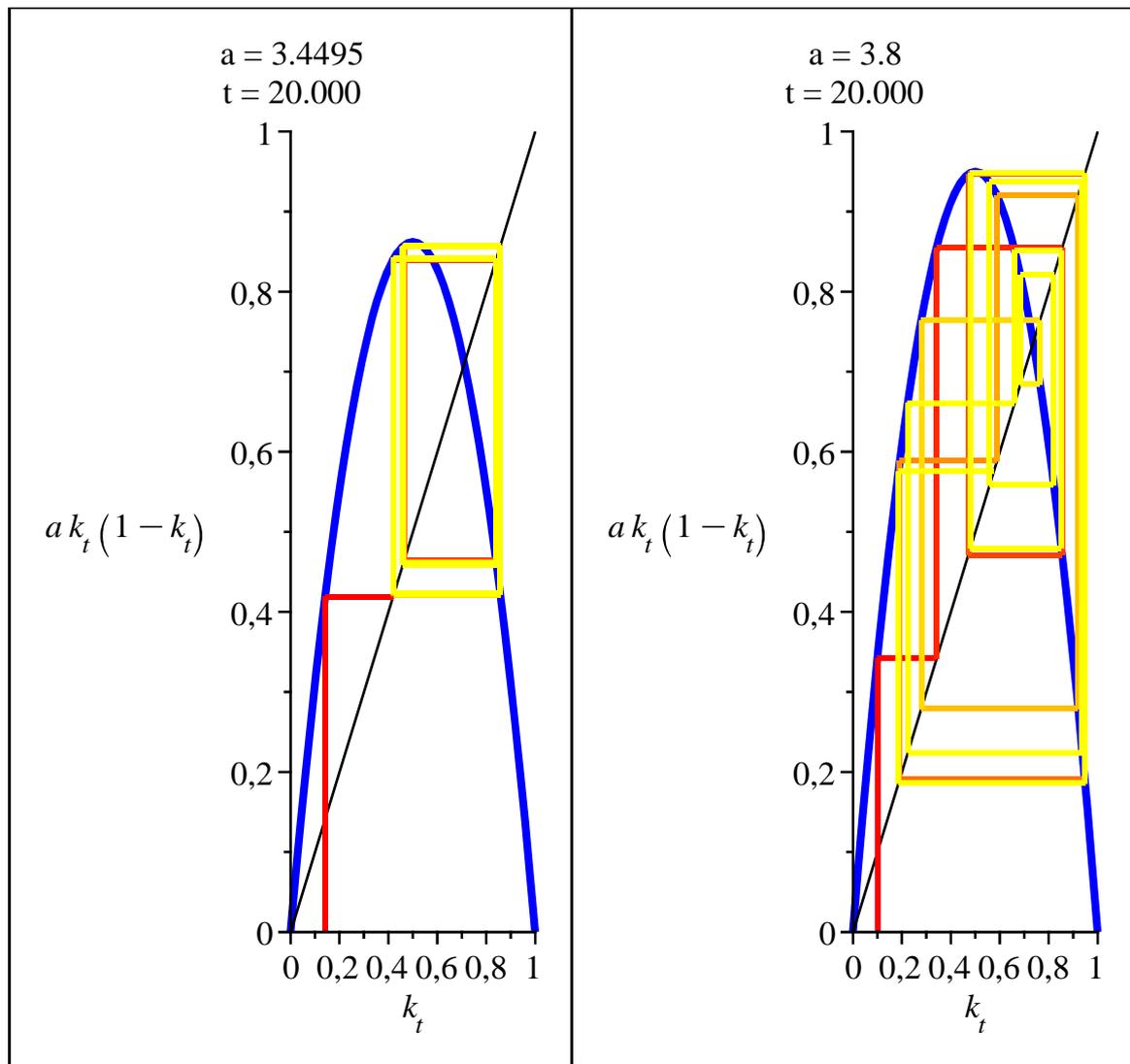
```
A[1,1]:=phase(0.1,0.5,20,ani):A[1,2]:=phase(0.1,2.5,20,ani)
```

```
:A[2,1]:=phase(0.1414848399,3.4495,20,ani):A[2,2]:=phase
```

```
(0.1,3.8,20,ani):
```

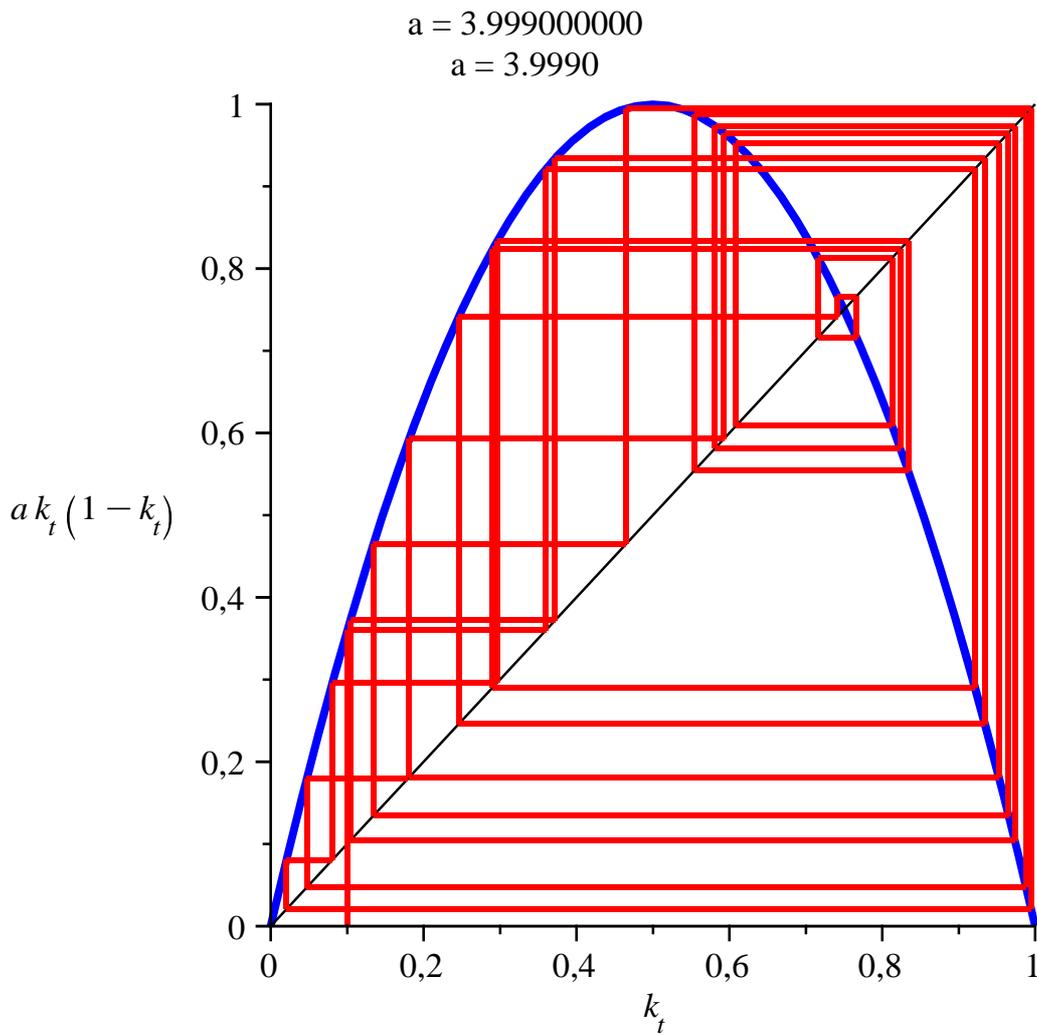
```
display(A);
```



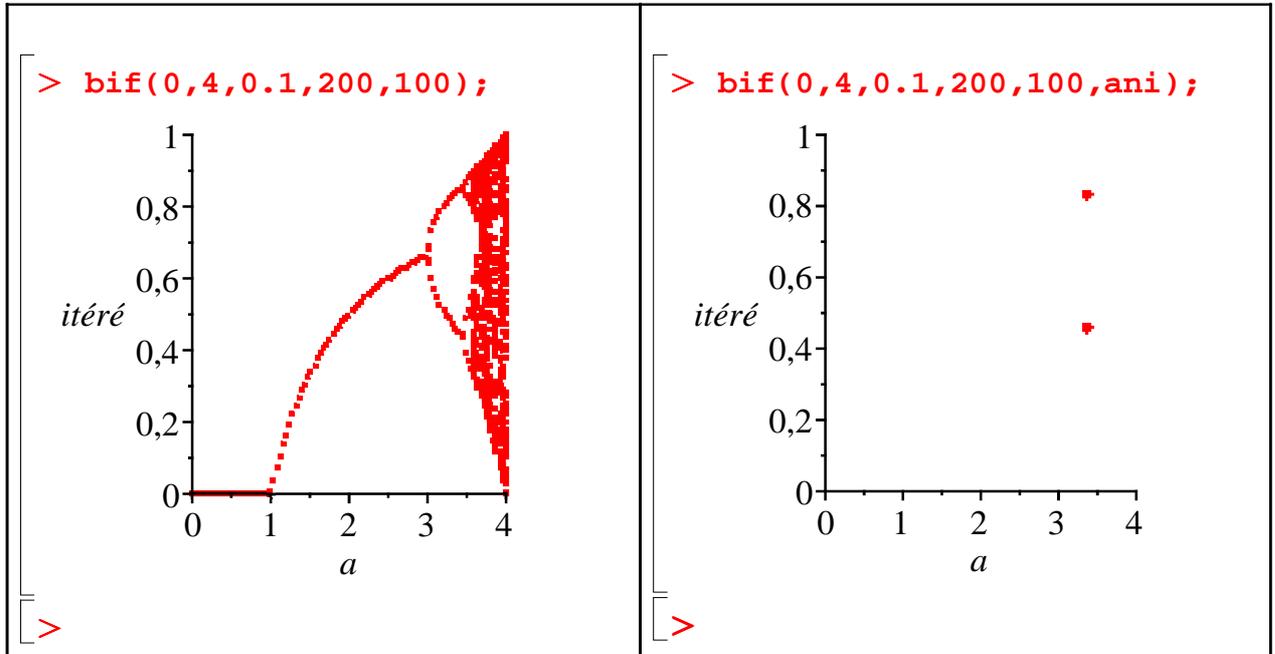


Une autre manière de visualiser nos résultats consiste à ne plus animer le diagramme de phase selon les itérés mais selon la valeur du paramètre a . On doit alors fixer le niveau de capital initial (ici, à 0.1).

```
> graph:=k0->animate(phase,[k0,a,30],a=0..3.999,frames=100):
graph(0.1);
```



Enfin, voici le diagramme de bifurcation dans son ensemble, c'est à dire pour a variant de 0 à 4 :

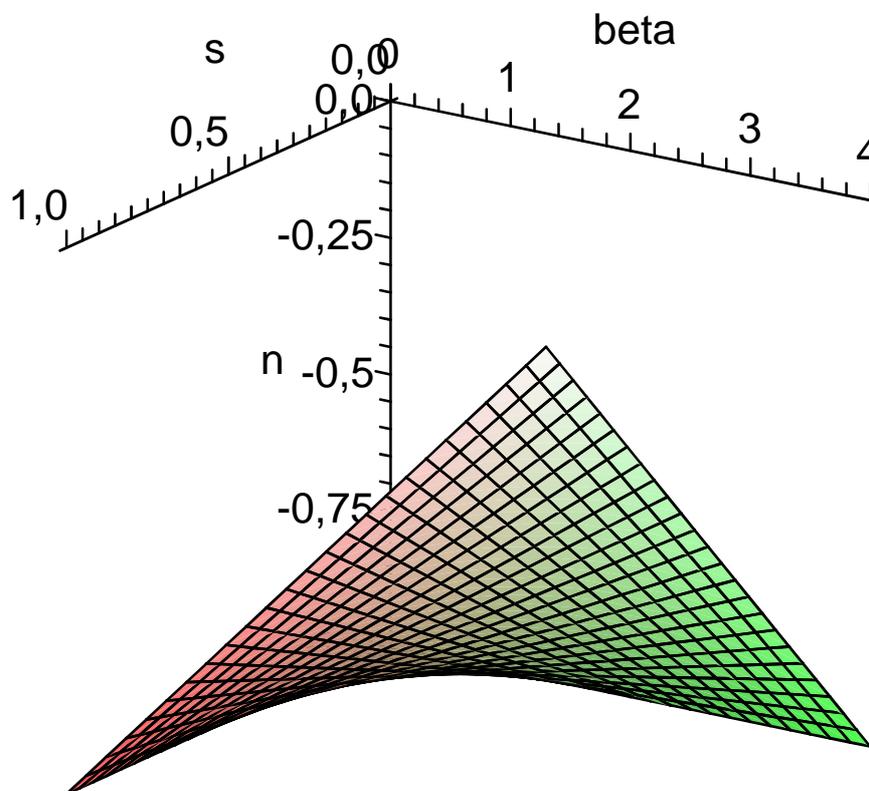


▼ Conclusion

Les sections précédentes ont permis de montrer que la théorie du chaos peut intéresser

l'économiste. En effet, le modèle de Day montre qu'avec une simple dynamique à la Solow en temps discret, l'ajout d'une externalité polluante peut amener l'économie dans une dynamique chaotique. Cette possibilité est apparue en faisant varier le paramètre a . Pour autant, rappelons que $a = \frac{\beta s}{1+n}$. Afin de s'assurer de la plausibilité de l'occurrence de chaos, représentons n en fonction de β et de s , en ayant posé au préalable $a = 4$ (chaos total). On obtient la surface suivante :

```
> f:=(s,beta)->(s*beta-4)/4;
plot3d(f,0..1,0..4,axes=normal,labels=[s,beta,n]);
```

$$f := (s, \beta) \rightarrow \frac{1}{4} s \beta - 1$$


Supposons que $\beta = 4$. On évoque 2 scénarios : l'un probable, l'autre un peu plus "excentrique" :

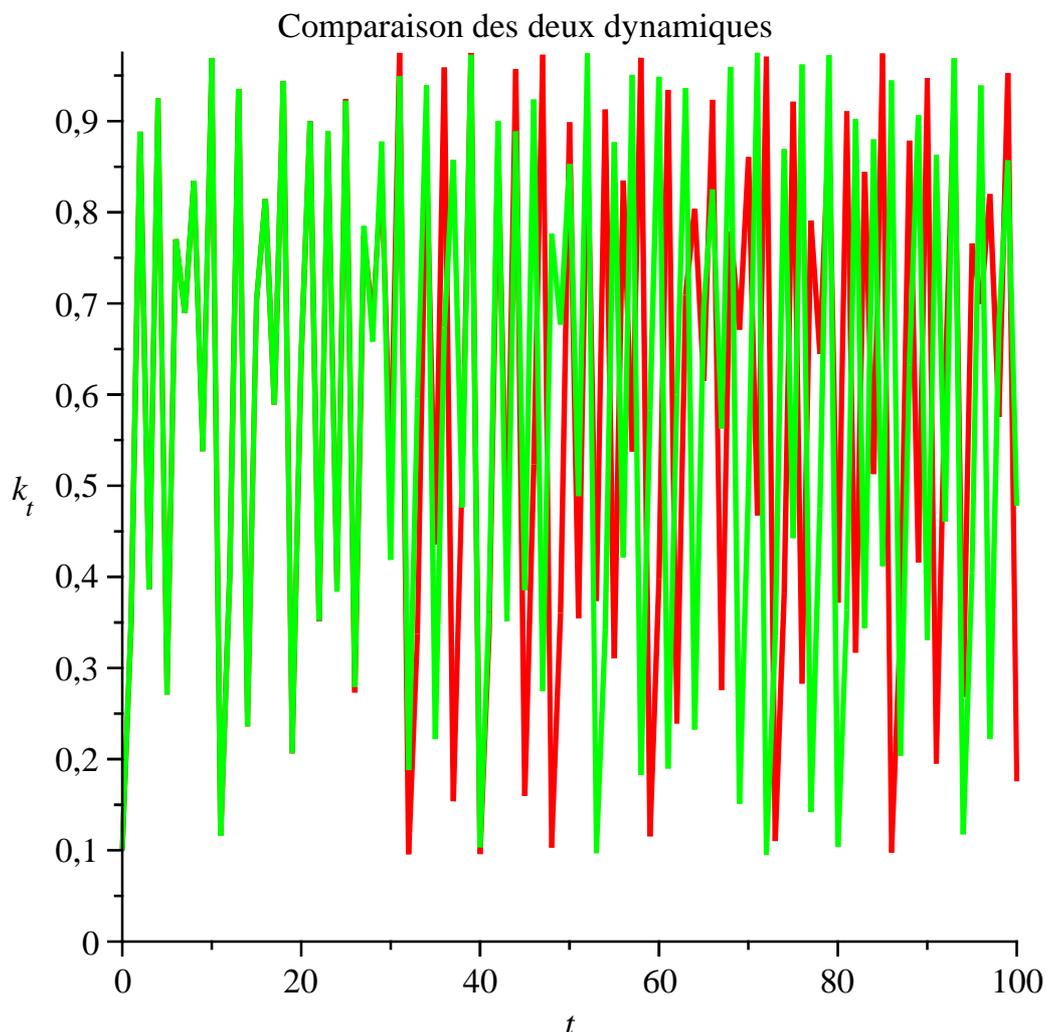
- Scénario probable : On pose la propension marginale à épargner égale à 0.5 ($s = 0.5$). La relation précédente demande que $n = -0.5$ Autrement dit pour que $a = 4$, il faudrait que la population soit divisée par 2 en une période. Ce scénario, plus ou moins en phase avec les données réelles, montre que le chaos n'est vraiment pas une situation très inquiétante puisque sa "probabilité" d'occurrence est quasiment nulle ($n = -0.5$).
- Scénario "excentrique" : Supposons maintenant que, dans un futur plus ou moins proche, les entreprises cessent de créer de nouveaux biens ou cessent d'améliorer les produits déjà existants. Ce scénario improbable n'est pas inintéressant sur le plan théorique. En effet, dans une pareille situation, on peut penser que l'épargne augmente de façon très importante ($s = 0.9$) car les consommateurs anticipent que des biens de consommation aux concepts

novateurs arriveront dans le futur.

Or nous raisonnons dans un contexte d'équilibre intertemporel, par conséquent l'équilibre sur le marché du capital est vérifié à toute date, donc l'investissement suivra la hausse de l'épargne. A première vue, on pourrait se dire qu'une hausse de l'investissement est formidable pour la croissance économique. Et pourtant, il suffit que le taux de croissance de la population n soit égal à -0.1 (valeur plutôt réaliste, surtout si on considère que 30 ans séparent les dates t et $(t + 1)$ pour que a soit égal à 4. Autrement dit, une situation certes presque idéale (investissement élevé), s'avérerait être en réalité une véritable catastrophe pour l'économie.

On peut considérer cette situation comme abominable pour deux raisons : la première est la très forte instabilité du taux de croissance, la seconde est l'impossibilité de prédire ce qui se passera à cause de "l'effet papillon" (sensibilité aux conditions initiales) induit par le chaos. En effet, une infime erreur dans l'estimation du capital par tête initial change du tout au tout la valeur du capital par tête aujourd'hui. Le lecteur pourra s'en rendre compte dans l'annexe et dans l'immédiat avec le graphique suivant (la différence initiale est ici de $1 \cdot 10^{-8}$).

```
> compa(0.1,0.10000001,3.9,100);
```



L'élément dérangeant de cette étude est que des taux de croissance démographiques négatifs ne se rencontrent généralement que dans les économies occidentales, et en particulier européennes ..

▼ Références bibliographiques

Abraham-Frois G. et E. Berrebi (1995) : *Instabilité, cycles, chaos*, Economica.

Day R.H. (1982) : "Irregular Growth Cycles", *American Economic Review*, vol.72, Pages. 406

-414.

Gandolfo G. (1997) : *Economic Dynamics* (Study Edition), Springer-Verlag.

Grandmont J-M (1985) : "On Endogenous competitive Business Cycles", *Econometrica*, volume 53, Pages 995-1046.

Grandmont J-M (2008) : "Nonlinear Difference Equations, Bifurcations and Chaos: An Introduction", *Research in Economics*, volume 62, Issue 3, Pages 122-177.

Nishimura K. et G. Sorger (1996) : "Optimal Cycles and Chaos : A Survey", *Studies In Nonlinear Dynamics and Econometrics*, Volume 1, Number 1, MIT Press.