

SUITES NUMÉRIQUES RÉELLES et ÉQUATIONS DE RÉCURRENCE

Bernard Dupont

Bernard.Dupont@univ-lille1.fr

En mathématiques, une suite est une application, génériquement notée u , de \mathbb{N} dans \mathbb{R} . Si on connaît la correspondance u , on dit que la suite est définie explicitement et il est facile de calculer les images $u(n) = u_n$. Dans la pratique, les économistes travaillent sur des équations de récurrence, qui donnent une liaison entre au moins deux images d'une suite u . Le plus souvent, il s'agit d'étudier un phénomène dans le temps dont on suppose connue la loi d'évolution. Pour déterminer le comportement asymptotique de la variable étudiée, il est important de résoudre l'équation de récurrence, c'est à dire trouver une suite explicite solution. Quand c'est impossible, il existe des méthodes graphiques/qualitatives permettant de conclure sur la nature de la suite-solution.

Ce chapitre commence par l'étude des suites définies explicitement au moyen de la commande **seq**. La seconde section est consacrée aux équations de récurrence d'ordre 1 : méthode de calcul des itérations, techniques de résolution explicite par la commande **rsolve** appliquée à des équations linéaires ou non linéaires, méthode graphique dite en escalier ou en escargot. La troisième section traite des équations de récurrence linéaire d'ordre 2. La quatrième section aborde les systèmes récurrents linéaires.

Suites définies explicitement (**seq**)

Soit la suite numérique réelle $u_n = u(n)$. On obtient les $(T + 1)$ premiers termes de la suite par la commande **seq(u(n), n=0..T)** où **u(n)** est une expression de **n**. Le résultat est une expression du type séquence.

On peut ainsi demander la liste des carrés des 11 premiers entiers naturels :

```
> restart;
  seq(n^2, n=0..10); whattype(%);
                                0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100
                                exprseq
```

ou encore suivre l'évolution d'une variable économique $k(t)$ évoluant au taux de variation constant g :

```
> seq(k0*(1+g)^t, t=0..10); whattype(%);
k0, k0(1+g), k0(1+g)^2, k0(1+g)^3, k0(1+g)^4, k0(1+g)^5, k0(1+g)^6, k0(1
+g)^7, k0(1+g)^8, k0(1+g)^9, k0(1+g)^10
                                exprseq
```

Pour connaître le comportement asymptotique de telles suites, il suffit d'utiliser la commande **limit(u(n), n=infinity)** où **u(n)** est une expression de **n**. Les trois exemples suivants montrent bien que Maple ne fait pas de différence entre un argument entier naturel et un argument réel.

```
> u:=n->cos(exp(-n)); #définition de l'application-suite
  limit(u(n), n=infinity); #requête sur le comportement
```

asymptotique

$$u := n \rightarrow \cos(e^{-n})$$

```
> f:=n->exp(-n^2);  
limit(f(n),n=infinity);
```

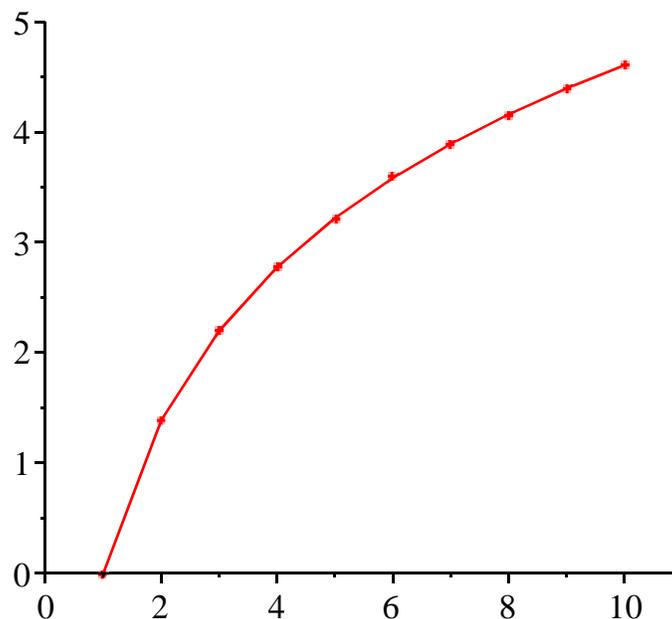
$$f := n \rightarrow e^{-n^2}$$

```
> h:=n->exp(n)/log(n);  
limit(h(n),n=infinity);
```

$$h := n \rightarrow \frac{e^n}{\log(n)}$$

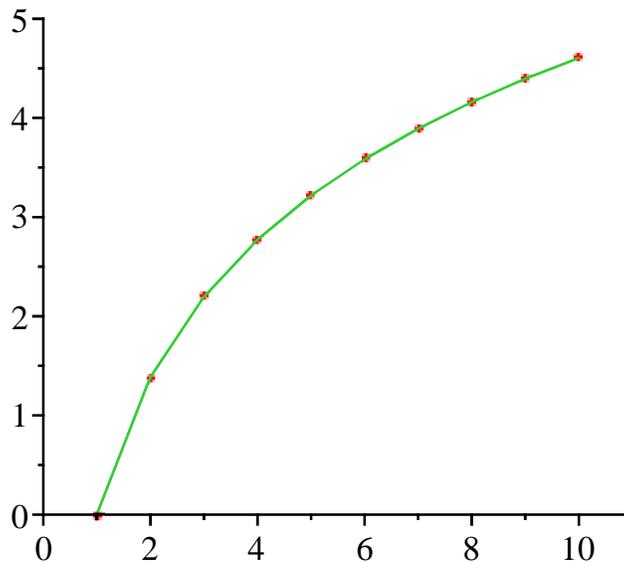
On obtient une représentation graphique de la suite avec `plot([seq([n,u(n)],n=0..N]), style=point,options éventuelles)`. Il est d'usage de joindre les points par des segments de droite et c'est pourquoi on superpose dans l'exemple suivant les graphiques `g1` et `g2` à l'aide de `display`.

```
> g1:=plot([seq([n,log(n^2)],n=1..10)],style=point)  
:#représentation de la suite de points  
g2:=plot([seq([n,log(n^2)],n=1..10)]):#représentation des  
segments de droite reliant les points  
with(plots):display({g1,g2},view=[0..11,0..5]);
```



En fait une seule ligne de commande peut suffire.

```
> plot([[seq([n,log(n^2)],n=1..10)],[seq([n,log(n^2)],n=1..10)]  
],style=[point,line],view=[0..11,0..5]);
```



Suites définies par une équation de récurrence d'ordre 1

Une suite peut être solution d'une équation de récurrence. Il n'existe pas de méthode générale de résolution des équations de récurrence d'ordre 1, à l'exception des équations linéaires dont la théorie est bien connue. Pour avoir une idée du phénomène, on peut demander à Maple de calculer et tracer les premiers termes d'une suite à partir de l'équation de récurrence et le premier terme.

Calcul et représentation graphique des termes d'une suite récurrente

Soit l'équation de récurrence d'ordre un : $u_{n+1} = f(u_n)$ où f est une fonction numérique d'une variable réelle. On veut calculer les termes de la suite à partir du premier terme u_0 .

Le réflexe immédiat est de procéder comme on le ferait "à la main" en commençant par définir la manière d'obtenir les termes de la suite par la fonction f :

```
> restart; u := n -> f(u(n-1));
      u := n -> f(u(n-1))
```

Ensuite, on donne le premier terme de la suite :

```
> u(0) := u0;
      u(0) := u0
```

On peut alors demander le calcul des termes de la suite à partir du premier :

```
> seq(u(i), i=0..10);
u0, f(u0), f(f(u0)), f(f(f(u0))), f(f(f(f(u0)))), f(f(f(f(f(u0))))),
  f(f(f(f(f(f(u0))))), f(f(f(f(f(f(u0))))),
  f(f(f(f(f(f(f(u0))))), f(f(f(f(f(f(f(u0))))),
  f(f(f(f(f(f(f(f(u0))))))
```

On obtient des itérés. En pratique, cette première méthode est très gourmande en calculs et peut prendre beaucoup de temps car le logiciel calcule isolément les termes : il commence par la condition initiale; pour calculer le second terme, il applique la correspondance à la condition initiale; pour le troisième terme, il constate qu'il a besoin du second terme et le calcule au lieu de réinjecter le résultat de l'itération précédente; etc. A titre d'exemple, prenons l'équation

d'accumulation du capital par tête du modèle de Solow en temps discret.

```
> k:=t->0.2*k(t-1)^(1/3)+(1-0.05)*k(t-1);  
k(0):=1;  
seq(k(t),t=0..20);
```

$$k := t \rightarrow 0.2 k(t-1)^{1/3} + (1 - 0.05) k(t-1)$$
$$k(0) := 1$$

1, 1.15, 1.302037911, 1.455328592, 1.609209358, 1.763118256, 1.916577297,
2.069179234, 2.220576959, 2.370474861, 2.518621703, 2.664804674,
2.808844383, 2.950590606, 3.089918645, 3.226726196, 3.360930633,
3.492466650, 3.621284195, 3.747346656, 3.870629271

Ces calculs demandent environ 85 secondes sur un ordinateur cadencé à 1 Ghz. Comparons avec la méthode suivante qui commence par définir la fonction f , puis pose la relation de récurrence et le premier terme, enfin calcule les termes de la suite-solution.

```
> f:=k->0.2*k^(1/3)+(1-0.05)*k;#on définit d'abord la  
fonction f  
u:=t->f(u(t-1));#la fonction u permet d'exprimer le (t+1)-  
ième terme en fonction du t-ième  
u(0):=1;#condition initiale  
seq(u(t),t=0..20);
```

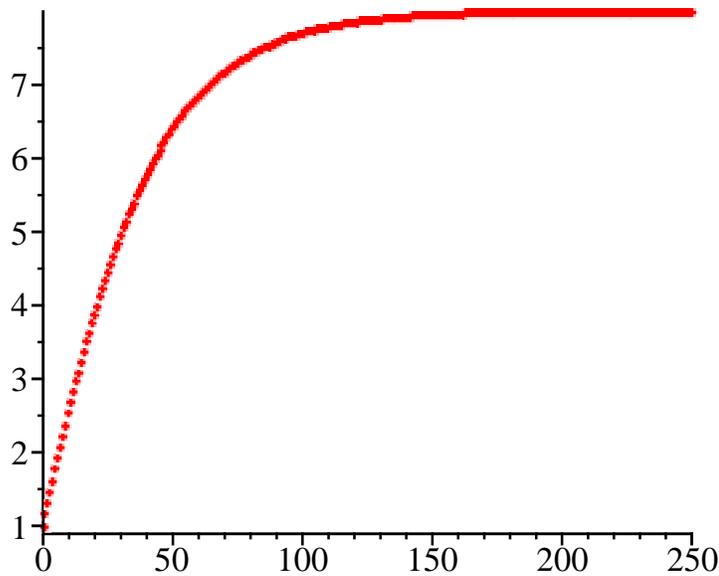
$$f := k \rightarrow 0.2 k^{1/3} + (1 - 0.05) k$$
$$u := t \rightarrow f(u(t-1))$$
$$u(0) := 1$$

1, 1.15, 1.302037911, 1.455328592, 1.609209358, 1.763118256, 1.916577297,
2.069179234, 2.220576959, 2.370474861, 2.518621703, 2.664804674,
2.808844383, 2.950590606, 3.089918645, 3.226726196, 3.360930633,
3.492466650, 3.621284195, 3.747346656, 3.870629271

Avec cette seconde méthode, le résultat est quasi instantané. Morale : **c'est toujours cette méthode qu'il faut utiliser.**

On trace aisément les points de coordonnées $(t, u(t))$ grâce à **plot** avec l'option **style=point**.
Noter la syntaxe : les points sont des couples ordonnés rassemblés dans une liste.

```
> plot([seq([t,u(t)],t=0..250)],style=point);
```



Le graphique fait apparaître clairement que le capital par tête converge vers un état stationnaire dans le modèle de Solow.

▼ Résolution des équations de récurrence du premier ordre

(**rsolve**)

La résolution d'une équation de récurrence du premier ordre se fait par la commande universelle **rsolve**. Maple résout toutes les équations de récurrence linéaires du premier ordre à coefficients constants à condition que le second membre soit standard. Il résout quelques équations non linéaires bien maîtrisées par les mathématiciens.

▼ *Syntaxe de **rsolve***

En l'absence de condition initiale, Maple recherche la solution générale, notée par exemple **u(n)** ou (en dynamique) **u(t)**, de l'équation de récurrence proposée **eqrec** sous la forme **u(n+1)=f(u(n))** (ou **u(t+1)=f(u(t))**). On écrit alors l'instruction **rsolve(eqrec, u(n))** ou **u(t)**. Quand il y a une condition initiale **u(0)=u0**, elle est adjointe à l'équation de récurrence dans un ensemble (set) et on écrit **rsolve({eqrc, u(0)=u0}, u)**.

▼ *Equations de récurrence linéaire du premier ordre à coefficient constant*

Une équation de récurrence linéaire du premier ordre à coefficient constant a pour forme générale $u_{n+1} + a u_n = b_n$. Maple est très efficace dans ce domaine. L'exemple suivant se contente de traiter le cas d'un second membre constant mais l'exercice M1 concerne divers types de second membre b_n .

```
> restart;
eqrec1:=u(n+1)=4*u(n)+2;soll1:=rsolve(eqrec1,u(n));
#équation de récurrence linéaire du premier ordre avec
second membre constant
      eqrec1 := u(n + 1) = 4 u(n) + 2
```

$$sol1 := u(0) 4^n - \frac{2}{3} + \frac{2}{3} 4^n$$

> `sol1:=rsolve({eqrec1,u(0)=1/3},u);#la même équation de récurrence avec une condition initiale spécifiée`

$$sol11 := 4^n - \frac{2}{3}$$

Equations de récurrence linéaires à coefficient variable

Plus difficile. Mais Maple s'en sort honorablement comme le montrent les exemples suivants.

> `restart;`
`rsolve({u(n) = n*u(n-1), u(0)=1}, u);#exemple pour les statisticiens`

$$\Gamma(n+1)$$

> `rsolve({u(n)=(1/n^2)*u(n-1)+1,u(0)=1},u);#exemple difficile`

$$\frac{\sum_{n1=0}^{n-1} \Gamma(n1+2)^2 + 1}{\Gamma(n+1)^2}$$

(2.2.3.1)

> `rsolve({u(n) = 2^n*u(n-1)+n^3, u(0)=1}, u);#exemple beaucoup plus difficile`

$$2^{\frac{1}{2}} (2+n)(n-1) \left(\sum_{n1=0}^{n-1} 2^{-\frac{1}{2} n1} (n1+3) (n1+1)^3 \right) + 2^{\frac{1}{2} n} (n+1)$$

Equations de récurrence non linéaires

Rien n'est vraiment assuré dans ce domaine. Dans les trois exemples suivants, `rsolve` s'avère un puissant auxiliaire pour les deux premiers mais ne peut rien faire dans le dernier cas.

Exemple 1 : Un classique! Soit la suite $\{u_n\}$ définie par la relation de récurrence $u_n = 3 u_{n-1}^2$ et le premier terme $u_0 = a$, avec a réel strictement positif. On veut connaître la nature de cette suite suivant les valeurs du paramètre a .

Demandons d'abord à Maple de résoudre l'équation de récurrence en toute généralité :

> `restart;eqrec:=u(n)=3*u(n-1)^2;assume(a>0);init:=u(0)=a;`
`sol:=rsolve({eqrec,init},u);`

$$eqrec := u(n) = 3 u(n-1)^2$$

$$init := u(0) = a$$

$$sol := \frac{1}{3} a^{2^n} 3^{2^n}$$

Il est immédiat que si $a = \frac{1}{3}$, alors la solution est la suite constante $u_n = \frac{1}{3}$, de limite $\frac{1}{3}$; si

$\frac{1}{3} > a$, alors la suite diverge; et si $a < \frac{1}{3}$, la suite converge et tend vers 0. L'appel à

`limit` le confirme.

```
> sol:=unapply(sol,a);
limit(sol(1/3),n=infinity);#cas a=1/3
limit(sol(1/4),n=infinity);#cas a<1/3
limit(sol(1),n=infinity);#cas a>1/3
```

$$sol := a \rightarrow \frac{1}{3} a^{2^n} 3^{2^n}$$

$$\frac{1}{3}$$

$$0$$

$$\infty$$

(2.2.4.1)

Exemple 2 : Soit la suite homographique $u_n = \frac{u_{n-1} + 3}{2 u_{n-1} + 2}$, $u_0 = 2$. On veut montrer qu'elle est convergente.

Le premier réflexe est de demander à Maple de résoudre directement l'équation proposée :

```
> restart;eqrec:=u(n)=(u(n-1)+3)/(2*u(n-1)+2);init:=u(0)=2;
sol:=rsolve({eqrec,init},u);
```

$$eqrec := u(n) = \frac{u(n-1) + 3}{2 u(n-1) + 2}$$

$$init := u(0) = 2$$

$$sol := rsolve\left(\left\{u(0) = 2, u(n) = \frac{u(n-1) + 3}{2 u(n-1) + 2}\right\}, u\right)$$

Catastrophe! Maple "sèche". En réalité, le programme est construit de telle manière qu'il faut éliminer les dénominateurs.

```
> restart;eqrec:=u(n)*(2*u(n-1)+2)=(u(n-1)+3);init:=u(0)=2;
sol:=rsolve({eqrec,init},u);
```

$$eqrec := u(n) (2 u(n-1) + 2) = u(n-1) + 3$$

$$init := u(0) = 2$$

$$sol := \frac{7 \cdot 2^n + 3 \left(-\frac{1}{2}\right)^n}{7 \cdot 2^n - 2 \left(-\frac{1}{2}\right)^n}$$

Tout est rentré dans l'ordre. Demandons alors la limite de la suite-solution quand n tend vers l'infini :

```
> limit(sol,n=infinity);
```

1

Exemple 3. Plus difficile. Soit l'équation de récurrence $u_n = \frac{u_{n-1}^2 + 3}{2 u_{n-1} + 2}$ avec $u_0 = 2$. Comme dans l'exemple précédent, on veut montrer qu'elle est convergente.

```
> restart;eqrec:=u(n)*(2*u(n-1)+2)=(u(n-1)^2+3);init:=u(0)=2;
sol:=rsolve({eqrec,init},u);
```

$$eqrec := u(n) (2 u(n-1) + 2) = u(n-1)^2 + 3$$

$$init := u(0) = 2$$

```
sol := rsolve( {u(n) (2 u(n - 1) + 2) = u(n - 1)^2 + 3, u(0) = 2}, u)
```

Maple renvoie en écho la question posée, ce qui signifie qu'il ne peut rien faire dans ce cas précis. Il faut se tourner vers d'autres méthodes, dites qualitatives ou graphiques.

Méthode graphique

Soit l'équation de récurrence d'ordre un : $u_{n+1} = f(u_n)$. On représente dans le plan xOy la courbe représentative de la fonction f ainsi que la bissectrice principale.

Les équilibres sont par définition les solutions de l'équation $x = f(x)$. Graphiquement, ils correspondent à l'abscisse des points d'intersection de la courbe et de la bissectrice principale.

Une valeur initiale u_0 étant donnée, on peut visualiser par une courbe dite en escalier ou en escargot le comportement de la suite-solution puisque $u_1 = f(u_0)$, $u_2 = f(u_1)$, etc.. Il est nécessaire de calculer des itérés en utilisant la méthode appropriée décrite dans la sous-section "Calcul et représentation graphique des termes d'une suite récurrente".

A titre d'exemple, étudions l'équation $u_{n+1} = u_n^{\left(\frac{1}{3}\right)}$. On commence par poser la fonction $y = x^{\frac{1}{3}}$.

```
> restart; f:=x->x^(1/3);
```

```
f:=x->x1/3
```

Ensuite, on définit la suite grâce à l'équation de récurrence :

```
> u:=n->f(u(n-1));
```

```
u:=n->f(u(n-1))
```

et on donne la condition initiale (de préférence en nombre float, faute de quoi Maple calcule très lentement) :

```
> u(0):=0.25;
```

```
u(0) := 0.25
```

D'ores et déjà, on peut placer dans un même graphique la courbe représentative de f et la bissectrice principale.

```
> graf1:=plot([f(x),x],x=0..1.5,color=[black,black],  
thickness=2):
```

Pour construire le graphique en escalier/escargot, on a besoin de représenter une fonction en escalier. Un segment horizontal joint les points de coordonnées (u_n, u_{n+1}) et (u_{n+1}, u_{n+1}) et, à partir de ce dernier point, un segment vertical joint les points de coordonnées (u_{n+1}, u_{n+1}) et (u_{n+1}, u_{n+2}) . En fait, avec Maple, il n'est pas nécessaire de programmer les segments horizontaux, car ils seront automatiquement tracés. On a donc besoin de construire une suite de segments verticaux à l'aide d'une fonction **marche** puis de construire une fonction-suite "escalier".

Commençons par **marche** :

```
> marche:=i->([u(i),u(i)], [u(i),u(i+1)]);
```

```
marche:=i->([u(i),u(i)], [u(i),u(i+1)])
```

La fonction escalier associe au nombre k de marches désiré les marches correspondantes. Elles sont visualisées avec le premier segment vertical reliant $(u_0, 0)$ à (u_0, u_1) avec **plot** (noter que la présence de crochets) :

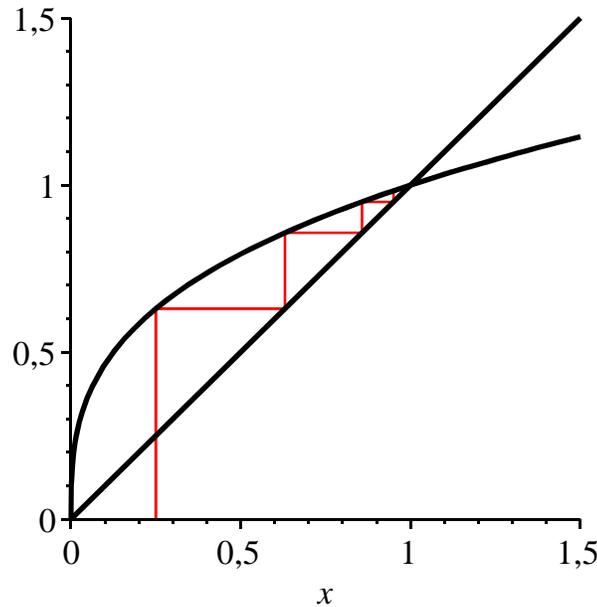
```
> escalier:=k->seq(marche(i),i=1..k);graf2:=plot([u(0),0],[u
```

```
(0),u(1)],escalier(3)]:
```

```
escalier:=k→seq(marche(i),i=1..k)
```

On superpose les graphiques **graf1** et **graf2** à l'aide de **display** :

```
> with(plots):display({graf1,graf2});
```



Pour améliorer la présentation, on n'hésitera pas à exploiter les possibilités de Drawing dans la barre contextuelle.

Suites définies par une équation de récurrence linéaire d'ordre

2

La commande **rsolve** résout toutes les équations de récurrence linéaire du second ordre qui sont du type $u_{n+2} + a u_{n+1} + b u_n = g_n$ avec a et b réels. Dans l'exemple suivant, le second membre est constant.

```
> restart;eqrecl:=2*u(n)-3*u(n-1)+u(n-2)=2;#second membre constant
```

```
init1:=u(0)=3,u(1)=3;
```

```
soll:=rsolve(eqrecl,u(n));#recherche de la solution générale
```

```
soll1:=rsolve({eqrecl,init1},u);#recherche de la solution avec conditions initiales
```

```
eqrecl := 2 u(n) - 3 u(n - 1) + u(n - 2) = 2
```

```
init1 := u(0) = 3, u(1) = 3
```

```
soll := -\frac{1}{2} (-4 u(0) + 4 u(1)) \left(\frac{1}{2}\right)^n - u(0) + 2 u(1) + 4 \left(\frac{1}{2}\right)^n - 4 + 2 n
```

```
soll1 := -1 + 4 \left(\frac{1}{2}\right)^n + 2 n
```

L'exercice M2 porte sur un éventail plus large de seconds membres.

▼ Systèmes d'équations de récurrence linéaires

Les systèmes d'équation de récurrence linéaires sont également résolus au moyen de la commande "à tout faire" **rsolve**, avec la syntaxe minimale **rsolve(equations,suites)** où **suites** est un ensemble renfermant les suites inconnues, $\{u,v,w,\dots\}$, et **equations** est un ensemble d'équations de récurrence $\{eq1,eq2,eq3,\dots\}$ ou d'équations de récurrence avec conditions initiales $\{eq1,eq2,eq3,\dots,u(0)=u0,v(0)=v0,w(0)=w0,\dots\}$. Dans le premier cas, Maple recherche la solution générale du système et dans le second cas la solution particulière.

Les deux exemples suivants traitent successivement de systèmes de deux équations de récurrence linéaires sans second membre et avec second membre. On demande d'abord la solution générale puis la solution particulière. Dans ce dernier cas, on réunit les ensembles (sets) d'équations et de conditions initiales par l'opérateur **union**.

```
> restart;
eq1:={u(n+1)=2*u(n)+3*v(n),v(n+1)=3*u(n)+2*v(n)};#système de
deux équations sans second membre
cint1:={u(0)=2,v(0)=0};#conditions initiales
suites1:={u,v};#désignation des suites inconnues
rsolve(eq1,suites1);#requête de la solution générale
rsolve(eq1 union cint1,suites1);#requête de la solution
particulière (union réunit les deux ensembles eq1 et cint1
```

$$eq1 := \{u(n+1) = 2u(n) + 3v(n), v(n+1) = 3u(n) + 2v(n)\}$$

$$cint1 := \{u(0) = 2, v(0) = 0\}$$

$$suites1 := \{u, v\}$$

$$\left\{ u(n) = \frac{1}{2} 5^n v(0) + \frac{1}{2} 5^n u(0) - \frac{1}{2} (-1)^n v(0) + \frac{1}{2} (-1)^n u(0), v(n) = \frac{1}{2} 5^n v(0) + \frac{1}{2} 5^n u(0) + \frac{1}{2} (-1)^n v(0) - \frac{1}{2} (-1)^n u(0) \right\}$$

$$\{u(n) = (-1)^n + 5^n, v(n) = -(-1)^n + 5^n\}$$

```
> eq2:={u(n+1)=5*u(n)+2*v(n)+1,v(n+1)=-2*u(n)+v(n)-1};#système
de deux équations avec second membre
cint2:={u(0)=5,v(0)=-2};#conditions initiales
suites2:={u,v};#désignation des suites inconnues
rsolve(eq2,suites2);#requête de la solution générale
rsolve(eq2 union cint2,suites2);#requête de la solution
particulière
```

$$eq2 := \{u(n+1) = 5u(n) + 2v(n) + 1, v(n+1) = -2u(n) + v(n) - 1\}$$

$$cint2 := \{u(0) = 5, v(0) = -2\}$$

$$suites2 := \{u, v\}$$

$$\left\{ u(n) = \frac{1}{2} 3^n - \frac{1}{2} + \frac{2}{3} 3^n u(0) + 3^n u(0) + \frac{2}{3} 3^n v(0), v(n) = \frac{1}{2} - \frac{1}{2} 3^n - \frac{2}{3} 3^n v(0) + 3^n v(0) - \frac{2}{3} 3^n u(0) \right\}$$

$$\left\{ u(n) = 2 \cdot 3^n n + \frac{11}{2} \cdot 3^n - \frac{1}{2}, v(n) = -\frac{5}{2} \cdot 3^n - 2 \cdot 3^n n + \frac{1}{2} \right\}$$

Applications économiques

En économie, les équations de récurrence sont surtout utilisées en dynamique quand le temps est considéré comme une variable discrète, c'est à dire une succession de dates ou de périodes de même longueur. Cette section développe deux exemples : un modèle microéconomique d'évolution du prix sur un marché et un modèle macroéconomique de croissance par accumulation du capital par tête.

Application microéconomique : le modèle du Cobweb

Le modèle dit du Cobweb (ou encore de la toile d'araignée) décrit la dynamique du prix d'un bien. Sur le marché d'un produit, on note $O(t)$ l'offre à la période t , $D(t)$ la demande à la période t et $P(t)$ le prix en t . On définit l'équilibre en t par l'égalité $O(t) = D(t)$. Dans une des versions du modèle, l'offre courante dépend du prix du bien de la période précédente alors que la demande tient compte du prix courant.

On commence par chercher l'évolution du prix équilibrant le marché quand les fonctions d'offre et de demande ont une spécification affine : $O(t) = c + d P(t - 1)$ et $D(t) = a - b P(t)$, tous les paramètres étant strictement positifs. Une fois exprimées les fonctions d'offre et de demande puis la condition d'équilibrage du marché, on fait appel directement à la commande **rsolve**.

```
> restart;
of:=p->c+d*p;
dd:=p->a-b*p;
eqrec:=of(p(t-1))=dd(p(t));
P:=rsolve(eqrec,p);
```

$$\begin{aligned} of &:= p \rightarrow c + d p \\ dd &:= p \rightarrow a - b p \\ eqrec &:= c + d p(t-1) = a - b p(t) \end{aligned}$$

$$P := p(0) \left(-\frac{d}{b} \right)^t - \frac{(-c+a) \left(-\frac{d}{b} \right)^t}{d+b} - \frac{c-a}{d+b} \quad (5.1.1)$$

Cette solution permet de distinguer 3 cas : $\frac{d}{b} < 1$, $1 < \frac{d}{b}$ et $\frac{d}{b} = 1$. Le calcul des limites est demandé à Maple. Il faut savoir interpréter les réponses. Quand $\frac{d}{b} < 1$, la réponse est sans ambiguïté. De fait, le prix de marché converge vers un état stationnaire avec des oscillations amorties.

```
> limit(P,t=infinity) assuming a>0,b>0,c>0,d>0,d/b<1;
      -c+a
      d+b
```

(5.1.2)

Dans les deux autres cas, il faut savoir interpréter les réponses. Quand $1 < \frac{d}{b}$, la réponse dépend de l'humeur du système au moment où on lance la requête. Parfois, on obtient :

```
> limit(P,t=infinity) assuming a>0,b>0,c>0,d>0,d/b>1;
```

$$\lim_{t \rightarrow \infty} \left(p(0) \left(-\frac{d}{b} \right)^t - \frac{(-c+a) \left(-\frac{d}{b} \right)^t}{d+b} - \frac{c-a}{d+b} \right) \quad (5.1.3)$$

et parfois, la réponse est :

```
> limit(P,t=infinity) assuming a>0,b>0,c>0,d>0,d/b>1;
      undefined
```

Quoiqu'il en soit, Maple répond de manière satisfaisante : la suite n'a pas de limite finie puisque le prix prend alternativement les valeurs $+\infty$ et $-\infty$ quand t tend vers l'infini. On est en présence d'oscillations explosives.

Enfin, quand $d = b$, l'écho renvoie systématiquement la question, ce qui signale que la suite n'a pas de limite (finie ou infinie). En fait, elle prend successivement les valeurs $P(0)$ et

$P(0) - \frac{2(c-a)}{d+b}$ et on est en présence d'un cycle perpétuel autour de la valeur d'équilibre $\frac{-c+a}{d+b}$.

```
> limit(P,t=infinity) assuming a>0,b>0,c>0,d>0,d=b;
```

$$\lim_{t \rightarrow \infty} \left(p(0) \left(-\frac{d}{b} \right)^t - \frac{(-c+a) \left(-\frac{d}{b} \right)^t}{d+b} - \frac{c-a}{d+b} \right) \quad (5.1.4)$$

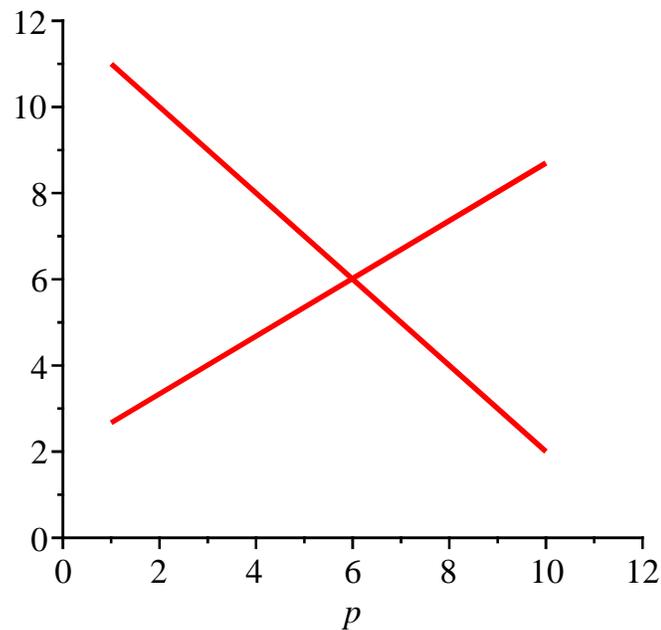
On pourrait bien entendu représenter les trajectoires du prix dans les trois cas mais il est d'usage de visualiser les phénomènes d'oscillations convergentes, explosives et cycliques dans le plan OpQ au moyen d'une toile d'araignée. Cet exercice n'est pas sans intérêt du point de vue programmatique puisque, au delà des difficultés de construction des graphiques, il contraint à réfléchir sur la signification même du modèle.

Dans le cas $d < b$, on commence par injecter un jeu de paramètres adéquats puis on récupère l'expression du prix d'équilibre, de la demande et de l'offre.

```
> c_init:={p(0)=4,a=12,b=1,c=2,d=0.67};
subs(c_init,P);Pel:=unapply(%,t);
subs(c_init,dd(p));Dd1:=unapply(%,p);
subs(c_init,of(p));Of1:=unapply(%,p);
      c_init := {a = 12, b = 1, c = 2, d = 0.67, p(0) = 4}
      -1.988023952 (-0.67)t + 5.988023952
      Pel := t → -1.988023952 (-0.67)t + 5.988023952
      Dd1 := p → 12 - p
      Of1 := p → 2 + 0.67 p
```

Un premier graphique donne dans le plan OpQ les droites de demande et d'offre.

```
> graf1:=plot([Dd1(p),Of1(p)],p=1..10,color=red,thickness=2,
      view=[0..12,0..12]):graf1;
```



L'objectif est de superposer à ce graphique un second chargé de visualiser "en toile d'araignée" les évolutions chronologiques du prix d'équilibre. C'est le plus délicat. Pour tout prix $p(t)$, il faut d'abord déterminer si le marché est en excès d'offre ou de demande ou encore équilibré. Les fonctions **infl** et **supl** calculent respectivement le côté court et le côté long du marché.

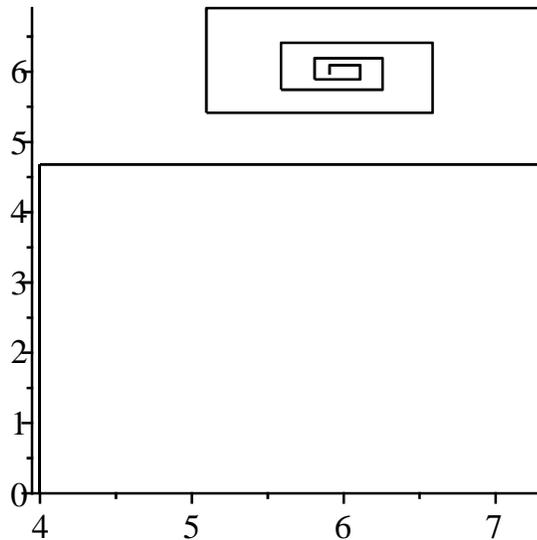
```
> infl:=p->if Of1(p)<= Dd1(p) then Of1(p) else Dd1(p) end if;
supl:=p->if Of1(p)>= Dd1(p) then Of1(p) else Dd1(p) end if;
infl := p → if Of1(p) ≤ Dd1(p) then Of1(p) else Dd1(p) end if
supl := p → if Dd1(p) ≤ Of1(p) then Of1(p) else Dd1(p) end if
```

La fonction **fill** génère **N** suites de 5 points reliant 4 segments de droite : l'abscisse et l'ordonnée du premier point correspondent à un prix de départ et les quantités échangées à ce prix par l'agent non rationné; le second exprime le prix satisfaisant l'agent rationné pour ces quantités échangées, prix qui devient le prix d'équilibre de la période suivante; le troisième point décrit la réaction de l'agent non rationné pour ce nouveau prix, et donc le volume d'échanges recherché; le quatrième point montre la formation du prix à la période suivante après ajustement de l'agent rationné; le cinquième retrace la réaction de l'agent non rationné en termes de volume d'échanges. Etc.. On a besoin de répéter deux fois la procédure d'ajustement pour revenir à une situation comparable au point de départ, autrement dit pour que fil de la toile puisse faire un carré.

```
> fill:=N->seq([[Pe1(2*k),infl(Pe1(2*k))],[Pe1(2*k+1),infl
(Pe1(2*k+1))],[Pe1(2*k+1),supl(Pe1(2*k+1))],[Pe1(2*k+2),
supl(Pe1(2*k+2))],[Pe1(2*k+2),infl(Pe1(2*k+2))]],k=0..N);
fill := N → seq([[Pe1(2k),infl(Pe1(2k))],[Pe1(2k+1),infl(Pe1(2k+1))],
[Pe1(2k+1),supl(Pe1(2k+1))],[Pe1(2k+2),supl(Pe1(2k+2))],
[Pe1(2k+2),infl(Pe1(2k+2))]],k=0..N)
```

Le graphique **cobweb1** trace tous les segments de la toile d'araignée ainsi que le segment de départ montrant comment s'initialise le processus.

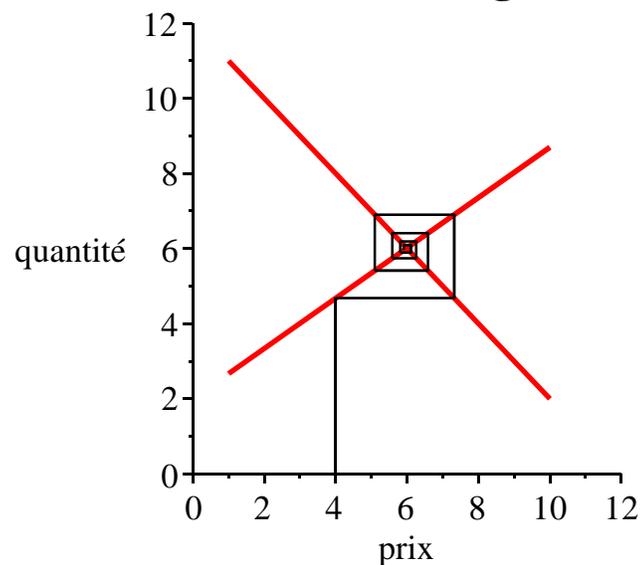
```
> cobweb1:=plot([[Pe1(0),0],[Pe1(0),infl(Pe1(0))]],fill(3)),
style=line,color=black):cobweb1;
```



Il ne reste plus qu'à superposer les deux graphes.

```
> with(plots):
  display({graf1,cobweb1},labels=["prix","quantité"],title=
  "Cobweb : cas convergent",titlefont=[TIMESNEWROMAN,BOLD,14]
  );
```

Cobweb : cas convergent



Pour représenter le cas "explosif", on répète les mêmes commandes (non sans quelques aménagements) pour un jeu différent de paramètres :

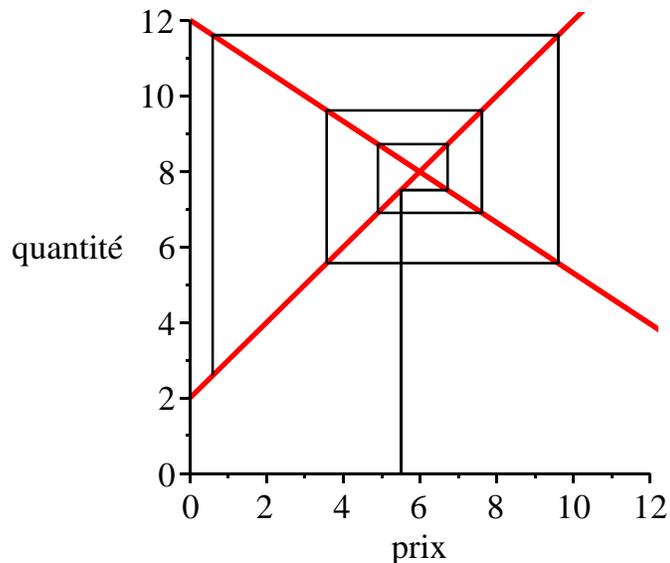
```
> c_init:={p(0)=5.5,a=12,b=0.67,c=2,d=1}:
  subs(c_init,P):Pe2:=unapply(%,t):
  subs(c_init,dd(p)):Dd2:=unapply(%,p):
  subs(c_init,of(p)):Of2:=unapply(%,p):
  graf2:=plot([Dd2(p),Of2(p)],p=0..25,color=red,thickness=2,
```

```

view=[0..12,0..12]):graf2:
inf2:=p->if Of2(p)<= Dd2(p) then Of2(p) else Dd2(p) end if:
sup2:=p->if Of2(p)>= Dd2(p) then Of2(p) else Dd2(p) end if:
fil2:=N->seq([[Pe2(2*k),inf2(Pe2(2*k))],[Pe2(2*k+1),inf2
(Pe2(2*k+1))],[Pe2(2*k+1),sup2(Pe2(2*k+1))],[Pe2(2*k+2),
sup2(Pe2(2*k+2))],[Pe2(2*k+2),inf2(Pe2(2*k+2))]],k=0..N):
cobweb2:=plot([[Pe2(0),0],[Pe2(0),inf2(Pe2(0))]],fil2(2),
style=line,color=black):cobweb2:
display({graf2,cobweb2},labels=["prix","quantité"],title=
"Cobweb : cas divergent",titlefont=[TIMESNEWROMAN,BOLD,14])
;

```

Cobweb : cas divergent



On procède de même pour le cas "oscillatoire" :

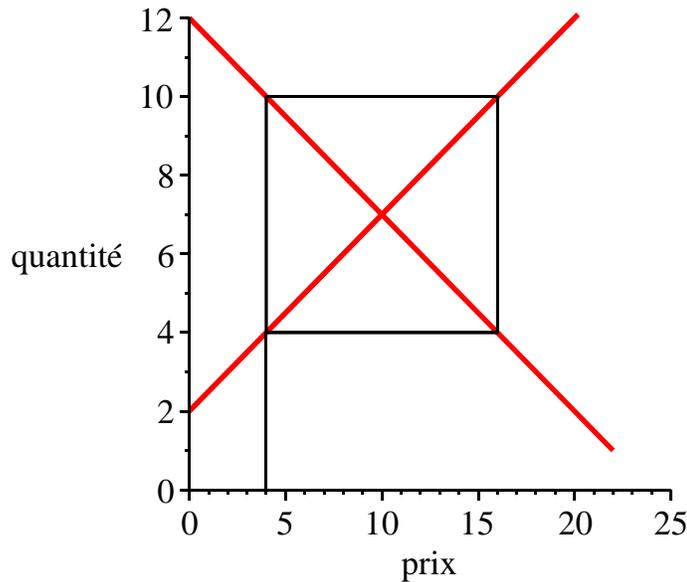
```

> c_init:={p(0)=4,a=12,b=0.5,c=2,d=0.5}:
subs(c_init,P):Pe3:=unapply(%,t):
subs(c_init,dd(p)):Dd3:=unapply(%,p):
subs(c_init,of(p)):Of3:=unapply(%,p):
graf3:=plot([Dd3(p),Of3(p)],p=0..22,color=red,thickness=2,
view=[0..25,0..12]):
inf3:=p->if Of3(p)<= Dd3(p) then Of3(p) else Dd3(p) end if:
sup3:=p->if Of3(p)>= Dd3(p) then Of3(p) else Dd3(p) end if:
fil3:=N->seq([[Pe3(2*k),inf3(Pe3(2*k))],[Pe3(2*k+1),inf3
(Pe3(2*k+1))],[Pe3(2*k+1),sup3(Pe3(2*k+1))],[Pe3(2*k+2),
sup3(Pe3(2*k+2))],[Pe3(2*k+2),inf3(Pe3(2*k+2))]],k=0..N):
cobweb3:=plot([[Pe3(0),0],[Pe3(0),inf3(Pe3(0))]],fil3(2),
style=line,color=black):
display({graf3,cobweb3},labels=["prix","quantité"],title=
"Cobweb : cas oscillatoire",titlefont=[TIMESNEWROMAN,BOLD,

```

14]);

Cobweb : cas oscillatoire



Application en macroéconomie : modèle de croissance de Solow

Le modèle de croissance solovien donne l'occasion de mener une étude qualitative (graphique) du phénomène de l'accumulation du capital par tête dans le cas discret. L'équation étudiée s'écrit

$k_{t+1} = \frac{s k_t^\alpha}{1+n} + \frac{1-\beta}{1+n} k_t$ où k_t est le capital par tête à la date t , s est le taux d'épargne fixe, n le taux de croissance de la population active, α est l'élasticité du capital à la production (la fonction de production sous-jacente est une Cobb-Douglas) et β est le taux de dépréciation du capital. Elle n'est pas linéaire puisque le paramètre α est un réel compris entre 0 et 1. S'il est impossible de dégager la solution générale, du moins est-il relativement aisé de montrer que le capital par tête converge toujours vers un équilibre. Autrement dit, cet équilibre est globalement stable sur \mathbb{R}_+^* .

On commence par donner la forme générale de la fonction qui intervient dans l'équation d'accumulation du capital par tête :

```
> restart; f := (k, s, n, alpha, beta, A) -> (s / (1+n)) * A * k^alpha + ((1-beta) / (1+n)) * k;
```

$$f := (k, s, n, \alpha, \beta, A) \rightarrow \frac{s A k^\alpha}{1+n} + \frac{(1-\beta) k}{1+n}$$

L'appel à `rsolve` est sans espoir. Maple renvoie la requête, ce qui signale qu'il ne peut pas trouver la solution générale.

```
> eqrec := k(t+1) = f(k(t), s, n, alpha, beta, A);  
rsolve(eqrec, k(t)) assuming 0 < s, s < 1, n > 0, alpha > 0, alpha < 1,  
beta > 0, beta < 1, A > 0;
```

$$eqrec := k(t+1) = \frac{s A k(t)^\alpha}{1+n} + \frac{(1-\beta) k(t)}{1+n}$$

(5.2.1)

$$\text{rsolve}\left(k(t+1) = \frac{sA k(t)^\alpha}{1+n} + \frac{(1-\beta) k(t)}{1+n}, k(t)\right) \quad (5.2.1)$$

Afin de préparer l'étude graphique, on donne des valeurs précises aux paramètres. La fonction fs est un cas particulier de la fonction f :

```
> fs:=k->f(k,0.15,0.01,1/3,0.1,1);fs(k);
      fs := k -> f(k, 0.15, 0.01, 1/3, 0.1, 1)
      0.1485148515 k1/3 + 0.8910891089 k
```

On exprime que le stock de capital en t est fonction du capital en $(t-1)$:

```
> k:=t->fs(k(t-1));
      k := t -> fs(k(t-1))
```

On donne la condition initiale :

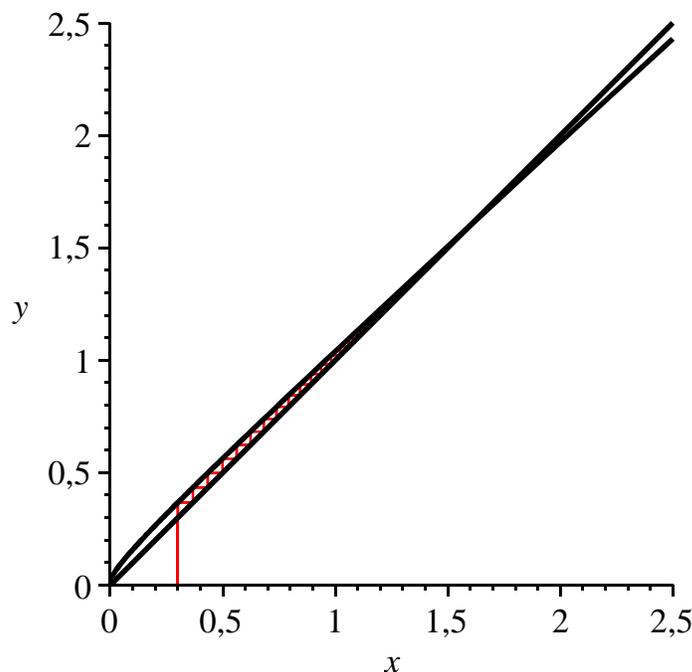
```
> k(0):=0.3;
      k(0) := 0.3
```

On crée une fonction en escalier qui illustrera la trajectoire du capital par tête :

```
> escalier:=t->([evalf(k(t)),evalf(k(t))],[evalf(k(t)),evalf(k(t+1))]);
      escalier := t -> ([evalf(k(t)), evalf(k(t))], [evalf(k(t)), evalf(k(t+1))])
```

Enfin, on superpose deux graphiques : le premier trace l'"escalier" (et son amorce) ; le second réunit la fonction fs et la bissectrice principale.

```
> graf1:=plot([[k(0),0],[k(0),k(0)]],[seq(escalier(t),t=0.40)],color=[red,red]):
graf2:=plot([fs(x),x],x=0..2.5,y=0..2.5,color=[black,black],scaling=constrained,thickness=2,numpoints=250):
with(plots):display({graf1,graf2},scaling=constrained);
```



Naturellement, on peut enjoliver la figure. Reste que la convergence vers un état d'équilibre stable est clairement mise en évidence ici. On peut d'ailleurs inférer la stabilité globale en multipliant les essais avec des conditions initiales différentes.

Exercices

Exercice M1

Résoudre les équations de récurrence suivantes, d'abord en donnant la solution générale puis en tenant compte de la condition initiale :

1. $u(n) + 2u(n-1) = 3n^2 + 1$ et $u(0) = 1$

2. $u(n+1) = \frac{u(n)+1}{2}$ et $u(0) = 3$

3. $3u(n-1) - 3u(n) + n = 1$ et $u(0) = 1$

4. $3u(n) - u(n-1) = 3^n$ et $u(0) = 0$

5. $u(n) + 2u(n-1) + e^{-n} = 0$ et $u(0) = \frac{2e}{2e+1}$

6. $u(n) - \frac{1}{2}u(n-1) = \frac{1}{2^n} - 1$ et $u(0) = 0$

7. $2u(n) - u(n-1) = n^2 + \frac{1}{2^{n-1}}$ et $u(0) = 1$

Exercice M2

Résoudre les équations de récurrence suivantes, d'abord en donnant la solution générale puis en tenant compte des conditions initiales :

1. $u(n) - 5u(n-1) + 6u(n-2) = 2n + 1$ avec $u(0) = 1$ et $u(1) = 4$

2. $u(n+2) - 2u(n+1) + 2u(n) = 2^n$ avec $u(0) = 0$ et $u(1) = 1$

3. $2\sqrt{3}u(n-1) - 4u(n-2) + 2^n \cos\left(\frac{1}{6}n\pi\right) = 0$ avec $u(0) = 0$ et $u(1) = 2 + \sqrt{3}$

Exercice M3

On reprend l'équation de récurrence qui n'a pas pu être résolue avec **rsolve**, soit

$$u_n = \frac{u_{n-1}^2 + 3}{2u_{n-1} + 2} \text{ avec } u_0 = 2. \text{ Montrer graphiquement que la suite } (u) \text{ est convergente.}$$

Exercice M4

Soit la suite $u_n = u_{n-1}^\alpha$ pour α réel strictement positif différent de 1. On sait que si $\alpha < 1$, alors la suite est convergente et de limite 1 et si $\alpha > 1$, alors la suite diverge. Visualisez le comportement de cette suite dans les cas $\alpha = 0.25$ et $\alpha = 1.5$.

Exercice E1 (Multiplicateur-accélérateur de Hicks-Samuelson)

Dans sa version la plus simple, le modèle du multiplicateur-accélérateur de Hicks-Samuelson a pour forme structurelle :

- (1) $Z_t = Y_t$ (équilibre sur le marché des biens et services)
- (2) $Z_t = C_t + I_t$ (la demande globale est la somme de la consommation des ménages et de l'investissement des firmes. Il n'y a pas d'Etat et pas de relations avec le reste du monde)
- (3) $C_t = c Y_{t-1}$ (fonction de consommation keynésienne avec retard)
- (4) $I_t = v (Y_{t-1} - Y_{t-2})$ (fonction d'investissement dans l'optique théorique d'un effet d'accélération)

où Y_t l'offre globale en t , C_t la consommation des ménages, I_t l'investissement des firmes. La propension à consommer c est comprise entre 0 et 1. Le coefficient d'accélération v est strictement positif.

1. Montrer que l'évolution de l'offre est régie par une équation de récurrence linéaire d'ordre 2.
2. Résoudre cette équation de récurrence dans le cas général à l'aide de **rsolve**. Quels commentaires vous inspire l'output?
3. Mener une étude classique de l'équation de récurrence à partir de son équation caractéristique.
4. Représenter graphiquement les différents types de trajectoires-solutions suivant les valeurs prises par c et v .

Exercice E2 (Evolution du prix d'un titre)

Pour toute période t , on note $P(t)$ le prix d'une action et $P^a(t+1)$ son prix anticipé pour la période suivante. Ce titre rapporte un dividende constant d à chaque période.

Un opérateur sur le marché des titres compare le prix de ce titre $P(t)$ et la somme actualisée au taux constant i ($0 < i < 1$) du dividende et de sa valeur prévue de revente, soit

$$\frac{1}{1+i} (d + P^a(t+1)).$$

L'arbitrage se définit par la condition $P(t) = \frac{1}{1+i} (d + P^a(t+1))$ puisque la somme investie à la période t est égale, du point de vue de l'opérateur, à la somme récupérée à la période $t+1$.

1. Le prix du titre est à son niveau d'équilibre P_e si le prix anticipé est égal au prix courant : $P_e = P(t) = P^a(t+1)$. Calculer ce prix d'équilibre. Montrer qu'il est égal à la somme infinie actualisée des dividendes que l'action rapporte. Comment interpréter ce résultat?
2. On suppose que les opérateurs du marché font des anticipations adaptatives sur le prix du titre. Le prix anticipé pour la période suivante est égal au prix qu'il avait anticipé pour la période courante corrigé de l'erreur d'anticipation constatée : $P^a(t+1) = P^a(t) + h(P(t) - P^a(t))$ avec $0 < h < 1$.
 - a) Donner la loi séquentielle d'évolution du prix du titre.
 - b) Résoudre cette équation de récurrence.
 - c) Montrer que toutes les trajectoires du prix convergent vers le prix d'équilibre.
 - d) Donner une représentation graphique du phénomène.
3. On suppose à présent que les opérateurs font des anticipations rationnelles parfaites. Le prix prévu est toujours égal au prix de la période suivante : $P^a(t+1) = P(t)$.
 - a) Donner la loi séquentielle d'évolution du prix du titre.
 - b) Résoudre cette équation de récurrence.
 - c) A quelle condition le prix du titre converge vers sa valeur d'équilibre? Comment se produit une bulle spéculative?
 - d) Donner une représentation graphique du phénomène.