Populations et démographie

Problème avec correction proposé par Stéphan Tassart
IRCAM, 1 place Igor Stravinsky, 75004 PARIS

`tassart@ircam.fr` `http://www.ircam.fr/equipes/analyse-synthese/tassart`

Mars 1997

Résumé

Ce problème consiste à modéliser l'évolution de la population de plusieurs espèces intéragissant les unes avec les autres. Dans la première partie, la modélisation se fait dans un domaine temporel discrétisé, et relève donc du cours sur les suites numériques. Cette partie est très largement inspirée d'un exercice de *La Recherche* n° 296. La deuxième partie concerne la modélisation à temps continue. Elle relève donc dans un premier temps du cours d'analyse numérique sur la résolution des équations différentielles. Cette partie consiste également en une introduction au contrôle des systèmes dynamiques, et à l'automatisme.

1 Problèmes

1.1 La guerre des lapins

Dans un grand domaine, des lapins s'ébattent librement. Les premières années, leur taux d'accroissement annuel est à peu près égal à une constante R, comprise entre 1 et 4. Autrement dit, la population passe d'une année sur l'autre de u_n à $u_n = Ru_{n-1}$. On suppose que la population initiale est de 500 lapins.

- Donner la loi d'évolution de la population de lapin `[rsolve]`.
- Dans quels cas est-elle divergente? Dans quels cas cette suite est-elle convergente. Quelle est sa limite?

Mais plus la population augmente, plus les ressources en nourriture diminuent, ce qui réduit l'accroissement de population. Finalement les démographes lapins constatent que l'évolution de la population respecte la loi: $u_n = Ru_{n-1}(1 - u_{n-1}/10^5)$ à ceci près que le nombre u_n est entier (il prend la valeur entière immédiatement inférieur `[floor]`). Autrement dit, plus u_n se rapproche de l'effectif 10^5, plus le taux de croissance diminue.

- Programmer la procédure **u** qui permet de calculer u_n.
- Représenter graphiquement l'évolution de la population sur 100 ans quand R vaut successivement 1.005, 1.1, 2.95, 3.05, 3.336 et 3.56, 3.57, 3.58, puis enfin 4. Dans le cas où la population n'est pas constante, évaluer la périodicité de cette suite (si tant est qu'elle est périodique) pour chacune de ces valeurs.

Nous allons tenter de prévoir pour quelques cas le comportement à long terme de l'évolution de la population.

- Dans l'hypothèse où la suite u_n est convergente, on note u^0 sa limite. Calculer u^0 en fonction de R (on admettra que la suite est convergente quand $R \in [1, 3]$).
- On suppose à présent que la suite oscille entre deux valeurs (autrement dit, que les suites extraites de rang respectivement pair et impair convergent vers deux limites différentes notés u^1 et u^2). Calculer u^1 et u^2 en fonction de R.

1.2 Des lapins et des lynx

En fait des lynx vivent aussi dans ce domaine. Les 200 lynx, seuls, mourraient rapidement de faim, mais heureusement il y a 38 195 lapins! L'évolution de la population de lapin doit, cette fois, dépendre du nombre v de lynx: $u_n = Ru_{n-1}(1 - u_{n-1}/10^5 - v_{n-1}/5000)$ (on prend toujours la partie entière). Mais parallèlement le nombre de lynx varie également, son taux de variation $K = 1/4$ (inférieur à 1 en l'absence des lapins) subissant lui aussi une correction due au nombre de lapins. D'une année sur l'autre, le nombre de lynx vérifie la relation $v_n = v_{n-1}(1 - v_{n-1}/25000 + a)/4$, où a est le plus petit des nombres 4 et $u_n/1000$.

- Programmer la procédure `u` en tenant compte de la correction due à la présence des lynx. Programmer également la procédure `v`.
- Représenter graphiquement l'évolution des populations de lapins et de lynx sur 100 ans quand R vaut successivement 1.1, 1.6, 2.1, et 2.5.

1.3 Le retour des lapins

Le modèle précédant possède l'avantage de la simplicité, mais l'inconvéniant de ne considérer la réactualisation de la population qu'une fois par an, alors que les lapins ne manquent pas de s'ébattre et de se reproduire à tous les moments de l'année. Autrement dit, un modèle plus réaliste consiste à considérer la fonction du temps $f(t)$, qui mesure en continue la population lapine. Les démographes lapins constatent que le taux d'accroissement instantanné de la population est donné par: $f'(t) = rf(t)$. On suppose que la population initiale, $f(0)$, est de 500 lapins, et que le temps est mesuré en années.

- Donner la loi d'évolution instantanée $f(t)$ de la population des lapins `[dsolve]`.
- De combien cette population augmente-t-elle d'une année sur l'autre. Par comparaison avec le modèle discrétisé précédant, déduire une relation entre r et R.

Comme d'habitude, les lapins consomment trop rapidement leur ressource en nourriture, ce qui réduit leur taux d'accroissement instantané. Les démographes lapins constatent donc que l'évolution de leur population suit le schéma suivant: $f'(t) = rf(t)(1 - f(t)/10^5)$. Autrement dit, plus le nombre de lapin s'approche de 10^5, plus le taux de croissance diminue.

- Donner la nouvelle loi d'évolution instantanée $f(t)$ de la population des lapins `[dsolve]`.
- Quelle est la limite de $f(t)$?
- Représenter graphiquement sur 10 années l'évolution de la population pour r valant successivement 1,2 et 4.

1.4 Les lynx contre-attaquent

Forcément, des lynx vivent également dans ce domaine, de pauvres lynx qui mourraient de faim sans la présence des lapins. Les démographes lapins notent $g(t)$ la mesure instantanée de la population de lynx. Ils constatent en outre que la présence des lynx perturbent l'évolution de leur propre population de la façon suivante: $f'(t) = rf(t)(1 - f(t)/10^5 - g(t)/5000)$. Parallèlement, l'évolution de la population lynx suit la loi $g'(t) = g(t)(-1 - g(t)/25000 + f(t)/1000)$.

- Tracer l'évolution sur 20 années des deux population pour $r = 1.6$.
- En considérant que quand t est grand, les variations de population $f'(t)$ et $g'(t)$ sont négligeables, évaluer les limites possibles que peuvent atteindre les populations des deux espèces.

2 Rappels

2.1 Programmation de suites itératives

Nous rappelons ici la syntaxe qui permet de programmer explicitement une suite itérative, comme par exemple celle de Fibonacci:

$$s_0 = s_1 = 1, \ \forall n > 1, \ s_n = s_{n-1} + s_{n-2} \quad (1)$$

```
> s := proc(n:integer)
  option remember;
    if n>1 then s(n-1)+s(n-2)
         else 1
    fi
  end;
```

On obtient alors s_8 en évaluant `s(8)`.

2.2 Représentation graphique d'une suite

Nous rappelons ici une méthode qui permet d'obtenir la représentation graphique d'un suite numérique. On suppose que `s` est une procédure qui permet de calculer explicitement les valeurs de la suite numérique s_n. Un représentation graphique s'obtient alors par la suite de commande suivante:

```
> readlib(forget);
> forget(s); plot([seq([i,s(i)],i=1..100)],style=LINE,color=red);
```

L'option `style` comprend les mots clés `LINE` et `POINT` suivant que l'on désire relier tous les points par des traits ou pas. L'option `color` comprend les mots clés habituels `blue`, `green`, `yellow`, `white`... afin de colorier le graphique selon sa convenance.

2.3 Superposition de plusieurs graphes

La superposition de plusieurs graphes en même temps s'obtient avec la commande `display` de la bibliothèque de fonctions `plots`. On suppose par exemple que `s1` et `s2` sont deux procédures permettant de calculer explicitement les valeurs de deux suites numériques différentes que l'on veut comparer. La superposition des deux graphes se fait par exemple de la façon suivante:

```
> readlib(forget);
> forget(s1); premier := plot([seq([i,s1(i)],i=1..100)],style=LINE,color=red):
> forget(s2); deuxieme := plot([seq([i,s2(i)],i=1..100)],style=LINE,color=blue):
> with(plots):
> display([premier,deuxieme]);
```

2.4 Tracé de solutions d'équations différentielles

Quand un système différentiel n'a pas de solution analytique, il est nécéssaire d'utiliser des algorithmes complexes pour obtenir des représentations graphiques des solutions. Sous Maple, la résolution numérique des systèmes différentiels se fait en utilisant le mot clé `numeric` dans la fonction `dsolve`:

```
> sys :=      {diff(y(x),x)=z(x), diff(z(x),x)=y(x)}:
> fcns :=     {y(x), z(x)}:
> cond_init:= {y(0)=0,z(0)=1}:
> F := dsolve(sys union cond_init, fcns, numeric);
```

Pour obtenir les solutions approchées du système en $x = 1.0$, il suffit alors de faire:

```
> F(1.0);
        {x = 1., y(x) = 1.175201186, z(x) = 1.543080624}
```

Pour tracer les courbe $y(x)$ et $z(x)$ en fonction de x, il suffit de faire:

```
> plot({'subs(F(xx),y(x))','subs(F(xx),g(x))'},xx=0..100);
```

3 Solution

3.1 La guerre des lapins

Si on ne se souvient pas directement de son cours, pour obtenir la forme explicite correspondant à la suite $u_n = Ru_{n-1}$ s'obtient sous Maple par:

```
> rsolve(u(n)=R*u(n-1),u(n));
```

$$u(0)R^n \tag{2}$$

Cette suite est convergente dès lors que $|R|$ est inférieur à 1, auquel cas la suite tend vers 0, ce qui correspond à une extinction de population. Donc du point de vue des lapins, il faut que $R > 1$.

La procédure suivante permet de simuler l'évolution de la population lapine, tout en tenant compte que l'accroissement de population diminue quand la population se rapproche de 100 000.

```
> u := proc(n:integer)
  option remember;
    if n>0 then floor(R*u(n-1)*(1-u(n-1)/100000))
         else 500
    fi
  end;
```

On obtient par exemple le tracé de l'évolution de la population lapine pour $R = 1.1$ par les instructions suivantes:

```
> readlib(forget);
> R:=1.1; forget(u); plot([seq([i,u(i)],i=1..100)],style=LINE,color=red);
```

On constate que tant que $R \leq 3$, l'évolution de la population est convergeante. Pour $R = 3.05$, on constate que l'évolution de la population devient cyclique, de période 2. Pour $R = 3.336$ et $R = 3.56$, l'évolution est toujours cyclique mais de période 4 (c'est à dire qu'on a assisté à un *doublement de période*). Pour $R = 3.57$, l'évolution est encore cyclique, mais la période est 32 (c'est à dire que le phénomène de doublement de période s'est répété 3 fois). Pour $R = 3.58$, l'évolution n'est plus du tout cyclique: elle est devenue *chaotique*. L'évolution reste chaotique tant que $R \leq 4$; au delà, le système diverge!

Si on suppose que u_n est une suite convergente, sa limite u vérifie alors l'équation $u = Ru(1 - u/10^5)$:

```
> R:='R'; g := u -> R * u * ( 1 - u/100000);
> solve(u = g(u), u);
```

$$0, 100000\frac{R-1}{R} \tag{3}$$

Pour étudier la stabilité de ces limites éventuelles, il faut vérifier que la dérivée de $Ru(1 - u/10^5)$ par rapport à u soit inférieur à 1 en valeur absolue:

```
> op(simplify(map(x->subs(u=x, diff(g(u),u)),["])));
```

$$R, 2 - R \tag{4}$$

Donc la première solution, 0, est stable si $-1 < R < 1$. La seconde solution est stable si $R \in [1, 3]$. Ceci constitue un argument qui plaide en la faveur de la convergence de u_n quand $-1 < R < 3$, vers 0 quand $-1 < R < 1$ et vers une valeur entière proche de $10^5\frac{R-1}{R}$ quand $1 < R < 3$.

3.2 Des lapins et des lynx

Les procédures se programment de la façon suivante:

```
> u := proc(n:integer)
  option remember;
    if n>0 then floor(R*u(n-1)*(1-u(n-1)/100000-v(n-1)/5000))
         else 38195
    fi
  end;
> v := proc(n:integer)
```

```
option remember;
  if n>0 then floor(v(n-1)*(1-v(n-1)/25000 +min(4,u(n-1)/1000))/4)
       else 200
  fi
end;
```

Le tracé s'obtient alors de la façon suivante:

```
> with(plots):
> R:=1.6; forget(u);forget(v);
> du:=plot([seq([i,u(i)],i=1..100)],style=LINE,color=red):
> dv:=plot([seq([i,10*v(i)],i=1..100)],style=LINE,color=yellow):
> display([du,dv]);
```

Le tracé fait clairement apparaître une évolution cyclique de l'ordre de 20 ans, qui permet aux lynx de proliférer quand les lapins sont en population importante. Les lynx épuisent alors quasi-complètement en 5 ans leur réserve de nourriture. À ce moment, les lynx disparaissent quasiment à leur tour. La disparition des lynx permet alors aux lapins de repeupler le territoire.

Il est clair qu'un déséquilibre léger des rythmes de reproduction des espèces peut avoir des conséquences catastrophiques sur ces mêmes espèces, et pas forcément dans le sens que notre intuition nous dicterait. Par exemple, une surreproduction des lapins, $R = 2.5$, entraine la disparition complète de l'espèce lynx en 61 ans!

Nous ne saurions donc qu'être particulièrement vigilant sur l'action que l'homme peut avoir sur son environnement, tant l'écologie d'un écosystème est fragile et complexe!

3.3 Le retour des lapins

La solution de la première équation fait partie du cours, mais dans le cas où un troui de mémoire surviendrait:

```
> dsolve({diff(g(t),t) = r*g(t),g(0)=g0},g(t));
```

$$g(t) = e^{rt} g(0) \tag{5}$$

D'une année sur l'autre, la population se trouve multiplié par e^r, donc, la correspondance entre r et R est la suivante: $R = e^r$.

La nouvelle équation différentielle ne fait pas parti du cours. Il faut donc faire confiance à l'outils de calcul symbolique:

```
> dsolve({diff(g(t),t) = r*g(t)*(1-g(t)/100000),g(0)=g0},g(t));
```

$$g(t) = 10^5 \left(1 + \frac{10^5 - g(0)}{g(0)} e^{-rt}\right)^{-1} \tag{6}$$

Les représentations graphiques des solution s'obtiennent par exemple en faisant:

```
> dsolve({diff(g(t),t) = r*g(t)*(1-g(t)/100000),g(0)=g0},g(t)):
> g_sol:=subs(",g(t));
> plot({subs(A=1,g0=500,g_sol),subs(A=2,g0=500,g_sol),
  subs(A=4,g0=500,g_sol)},t=0..10);
```

3.4 Les lynx contre-attaquent

Le nouveau système dequation est couplé, mais cela ne pose pas de problèmes particuliers. On définit chacun des systèmes par:

```
> lapin := {diff(f(t),t) = r*f(t)*(1-f(t)/f1-g(t)/g2),f(0)=f0};
> lynx := {diff(g(t),t) = g(t)*(-1-g(t)/g1+f(t)/f2),g(0)=g0};
> sys := lapin union lynx;
> F:=dsolve(subs(r=1.6,g2=5000,g1=25000,f2=1000,f1=100000,f0=38195,g0=200,sys),
    {f(t),g(t)},numeric);
> plot({'subs(F(x),f(t))','subs(F(x),g(t))'},x=0..20);
```

L'équilibre est caractérisé par l'anullation des quantités $f'(t)$ et $g'(t)$, ce qui nous donne:

```
> solve(eval(subs(diff=0,g(t)=g,f(t)=f,f(0)=f0,g(0)=g0,sys)),{f,g,f0,g0});
```

Ce qui donne:

$$\left\{\begin{array}{lcl} f & = & 0 \\ g & = & 0 \end{array}\right., \left\{\begin{array}{lcl} f & = & 0 \\ g & = & -g1 \end{array}\right., \left\{\begin{array}{lcl} f & = & f1 \\ g & = & 0 \end{array}\right., \left\{\begin{array}{lcl} f & = & f_1 f_2 \dfrac{g_1+g_2}{f_1 g_1 + f_2 g_2} \\ & & \\ g & = & g_1 g_2 \dfrac{f_1 - f_2}{f_1 g_1 + f_2 g_2} \end{array}\right. \tag{7}$$

En l'occurence, une application numérique donne:

```
> op(evalf(subs(g2=5000,g1=25000,f2=1000,f1=100000,{"})));
```

$$\left\{\begin{array}{lcl} f & = & 0 \\ g & = & 0 \end{array}\right., \left\{\begin{array}{lcl} f & = & 0 \\ g & = & -25000 \end{array}\right., \left\{\begin{array}{lcl} f & = & 100000 \\ g & = & 0 \end{array}\right., \left\{\begin{array}{lcl} f & = & 1197 \\ g & = & 6940 \end{array}\right. \tag{8}$$

On remarque toutefois que le système est pseudo-oscillant, de période à peu près 6 ans.