



Département  
Licence

K1BE6W14      Biomodélisation  
Mathématiques    TP machine V  
Ph. Thieullen

## Travaux pratiques V

### Modèles retardés pour une seule espèce

### Modèles discrets à plusieurs espèces

**TP 1.** On se propose de comprendre comment fonctionne un solveur d'équations différentielles retardées (dde), c'est-à-dire, comment résoudre numériquement l'équation

$$\frac{dx}{dt} = f(t, x(t - \tau_1), x(t - \tau_2), \dots, x(t - \tau_d)), \quad t \geq t_0$$

où  $0 \leq \tau_i \leq T$  sont des temps de retard intermédiaires entre 0 et le temps de retard maximum  $T$ . Comme pour toute équation différentielle, il faut commencer par l'initialiser. Si  $t_0$  est l'instant initial, l'utilisateur doit donc nécessairement choisir une condition initiale sous la forme : une fonction  $x = x_0(t)$  retournant des valeurs prescrites  $x(t)$  pour tout instant  $T - t_0 \leq t \leq t_0$ .

Comme Scilab ne possède pas de tel solveur, nous allons commencer par en récupérer un, à l'adresse suivante :

<http://www.math.u-bordeaux1.fr/~thieulle/Enseignement/K1BE6W14/dde.sci>

- Récupérez le fichier `dde.sci` à l'adresse précédente, puis enregistrez le dans le répertoire courant sous le nom de fichier, `dde.sci`
- Commencez votre script `TP05.sci` par la commande `exec("dde.sci")` qui charge en mémoire les fonctions internes du fichier.
- Tous les solveurs EDO fonctionnent sur le même principe : le code de base prend le format

```
function xdot = f(t,x)
    x_backward = ddeGetBackwardPositions(t,x,t_backward);
    xdot = ...(se servir de x_backward)...
endfunction
```

où  $\mathbf{x}$  est un vecteur colonne de taille  $L$  représentant  $x(t)$  pour  $L$  conditions initiales, et  $\mathbf{xdot}$  est un vecteur colonne de même taille  $L$  représentant  $\frac{dx}{dt}(t)$ . Pour récupérer les valeurs de  $x$  aux instant antérieurs ( $t - \tau_i$ ), on utilise la fonction `ddeGetBackwardPositions` qui retourne une matrice `x_backward` de taille  $L \times d$  si `t_backward` est un vecteur ligne de dimension  $d$  représentant les instants antérieurs ( $t - \tau_1, \dots, t - \tau_d$ ). La fonction d'initialisation  $x_0(t)$  doit nécessairement prendre la forme

```
function x = x_0(t)
    x = ...(utiliser length(t) pour calculer la taille de t)...
endfunction
```

où  $\mathbf{t}$  est un vecteur ligne de taille quelconque  $\mathbf{r}$  représentant des instants quelconques entre l'instant initial et son retard maximum  $t_0 - T \leq t \leq t_0$ , et  $\mathbf{x}$  est une matrice de taille  $\mathbf{L} \times \mathbf{r}$  représentant les  $\mathbf{L}$  fonctions initiales. Il est important de ne faire aucune hypothèse sur la taille de  $\mathbf{t}$  dans la fonction `x_0`. Après avoir défini les deux fonctions  $f(t, x)$  et  $x_0(t)$ , il suffit alors d'appeler la fonction `dde`, comme on aurait appelé `ode`, pour obtenir les valeurs suivantes de  $x(t)$

```
x = dde(T,x_0,intervalle,f)
```

où `intervalle` est un vecteur ligne de taille  $\mathbf{N}$  représentant un échantillon de temps strictement croissant  $(t_0, t_1, \dots, t_{N-1})$  démarrant à  $t_0$ , et  $\mathbf{x}$  est une matrice de taille  $\mathbf{L} \times \mathbf{N}$  représentant  $x(t_j)$  pour chacune des conditions initiales.

– Testez maintenant le solveur `dde` sur l'équation différentielle

$$\frac{dN}{dt} = -\frac{\pi}{2T}N(t-T), \quad T = 1 \quad (1)$$

On admettra que  $N(t) = A \cos(2\pi \frac{t-\phi}{4T})$  est une solution explicite de période  $4T$  quelque soient l'amplitude  $A$  et la phase  $\phi$ . Dans le cas du dessin de gauche de la figure 1, la condition initiale

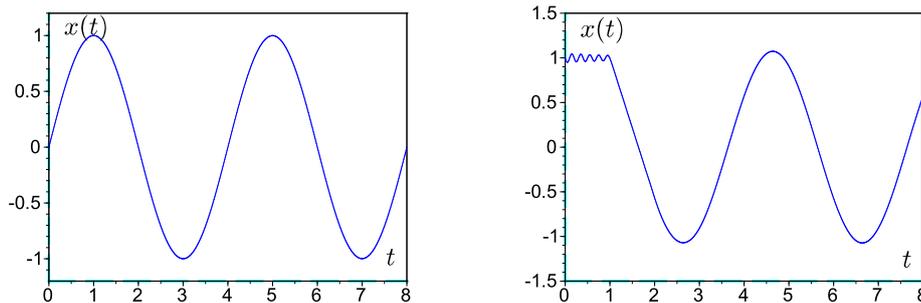


FIGURE 1 – Conditions initiales :  $\cos(2\pi \frac{t-T/2}{4T})$  à gauche et  $\cos(2\pi t \frac{t}{T/5})$  à droite.

est déjà solution de l'équation (1); la réponse du système est simplement le prolongement de cette condition initiale. Dans le cas du dessin de droite, la condition initiale est une fonction périodique de période  $\frac{T}{5}$ ; le système met un certain temps pour retourner à la solution de période  $4T$ .

**TP 2.** On considère le modèle logistique retardé écrit sans dimension

$$\frac{dx}{dt} = x(t)(1 - x(t-T)), \quad T \simeq \pi/2.$$

On constate que  $x_* = 1$  est un état d'équilibre. L'objectif de cette simulation est de montrer que  $x_*$  est stable pour  $T < \frac{\pi}{2}$  et instable pour  $T > \frac{\pi}{2}$ . On prendra comme condition initiale, une fonction constante  $x_0 = 0.1$ . On utilisera le code suivant

```
x = x_0*ones(1,length(t));
```

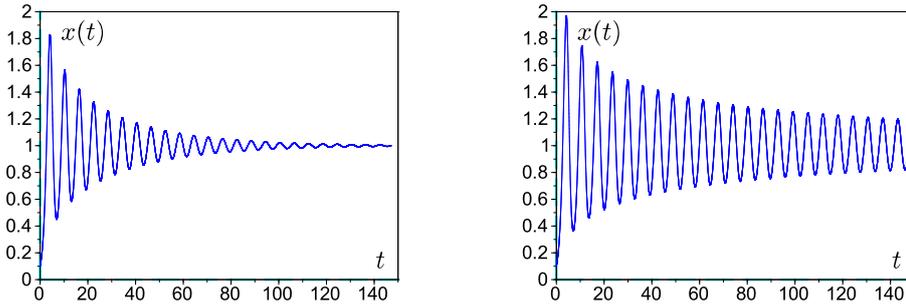


FIGURE 2 – Cas stable :  $T = \pi/2 - 0.1$  à gauche,  $T = \pi/2 - 0.01$  à droite.

dans la fonction d'initialisation  $x_0$ .

– En prenant différentes valeurs de  $T < \frac{\pi}{2}$ , montrez que le système se stabilise toujours vers  $x_* = 1$ . On trouvera la figure 2

– En prenant des valeurs de  $T > \frac{\pi}{2}$ , mettez en évidence l'apparition de cycles périodiques. On trouvera la figure 3.

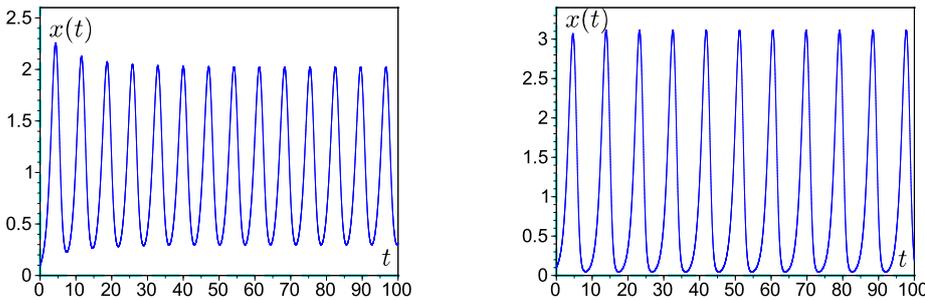


FIGURE 3 – Cas instable :  $T = \pi/2 + 0.15$  à gauche,  $T = \pi/2 + 0.5$  à droite.

**TP 3.** On se propose de décrire le diagramme de bifurcation d'un modèle de Nicholson-Bailey d'équations sans dimension

$$\begin{aligned} x_{t+1} &= x_t \exp(r(1 - x_t) - y_t) \\ y_{t+1} &= \gamma x_t (1 - \exp(-y_t)) \end{aligned}$$

où  $x_t$  On admettra qu'il existe un point d'équilibre non trivial  $(x_*, y_*)$ , à coordonnées strictement positives, pour tout  $r > 0$  et  $\gamma > 1$ , donné implicitement par le système

$$y_* = r(1 - x_*) \quad \text{et} \quad \gamma x_* = \frac{y_*}{1 - \exp(-y_*)}.$$

Il sera par la suite plus judicieux de prendre  $r > 0$  et  $x_* \in ]0, 1[$  comme variables indépendantes et d'utiliser les équations précédentes pour calculer  $y_*$  et  $\gamma$ . On admettra que la condition de stabilité est donnée par le système d'équations

$$1 - r + \gamma x_* > r(\gamma + r)x_*^2 - (\gamma + r^2)x_* - 1 \quad \text{et} \quad -r(\gamma + r)x_*^2 + (\gamma + r^2)x_* < 1.$$

– Définissez d’abord une fonction `N_plus = NicholsonBailey(t,N)` qu’on utilisera comme argument de `Nf=ode("discrete",N0,...)` ulérieurement; `N0` désigne une matrice  $L \times 2$ ,  $L$  est le nombre de conditions initiales; `N(:,1)` et `N(:,2)` désignent respectivement  $x_0$  et  $y_0$  pour chaque choix de conditions initiales; `Nf(:,1:2:($-1))` désigne  $x_t$  et `Nf(:,2:2:$)` désigne  $y_t$ . Il sera aussi commode d’introduire une fonction

```
[y_etoile,GAMMA] = paramNicholsonBailey(R,x_etoile)
```

retournant  $y_*$  et  $\gamma$  lorsqu’on s’est fixé  $r$  et  $x_*$ .

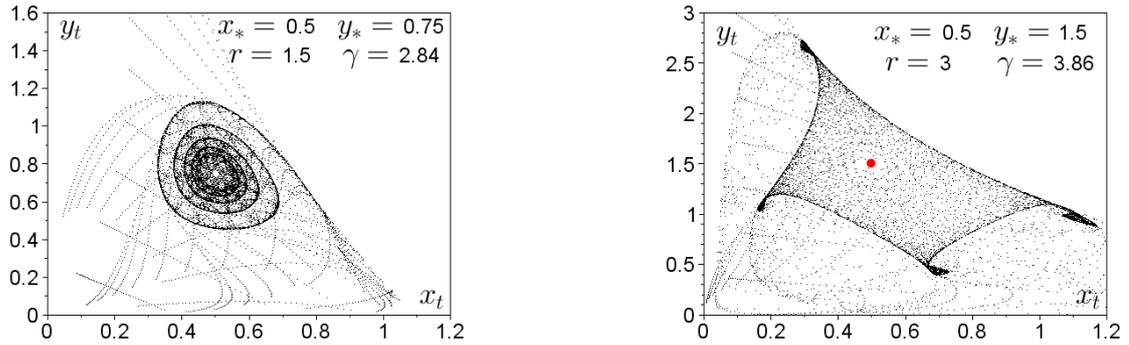
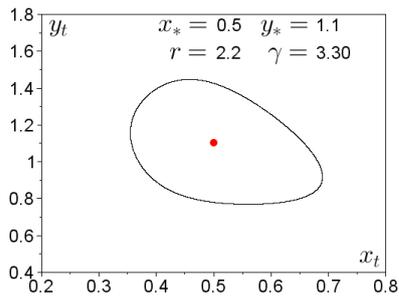


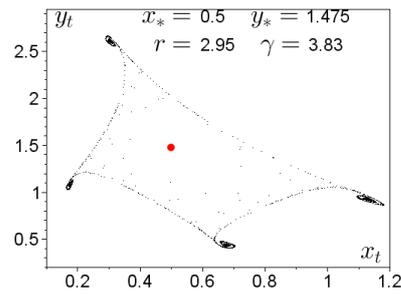
FIGURE 4 – Modèle de Nicholson-Bailey :  $(x_*, y_*)$  est stable à gauche et instable à droite

– On choisit  $x_* = \frac{1}{2}$  et  $r = \frac{3}{2}$ . Tracez alors les 30 premiers itérés de  $(x_t, y_t)$  à partir d’un quadrillage de conditions initiales  $(x_0, y_0)$  de pas 0.1 du domaine  $[0, 1?2] \times [0, r]$ . On constate que  $(x_*, y_*) = (\frac{1}{2}, \frac{3}{4})$  est un point stable. On obtiendra le dessin de gauche dans la figure 5. Modifiez les paramètres en prenant  $x_* = \frac{1}{2}$  et  $r = 3$  et obtenez le dessin de droite. On constate maintenant que  $(x_*, y_*) = (\frac{1}{2}, \frac{3}{2})$  n’est plus stable.

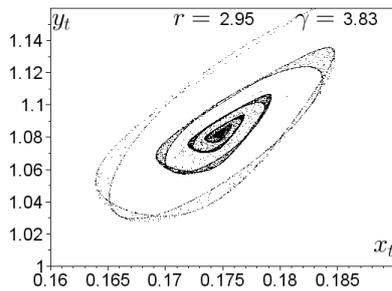
– Expérimenter d’autres valeurs de  $r = 2.2, 2.95, 3.3$ . Lorsqu’on cherche une orbite périodique stable, on considère seulement des temps d’itération court. Lorsqu’on cherche à obtenir des cycles limites ou des comportements plus complexes, les temps d’itération sont long et on ne trace que les derniers instants.



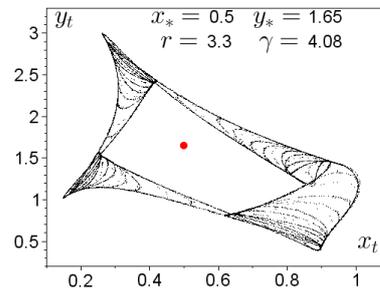
Apparition d'un cycle limite autour de l'état d'équilibre instable



Destruction du cycle et création d'une orbite périodique de période 4



Agrandissement autour du point de période 4 en bas à gauche



Destruction de l'orbite périodique vers un comportement plus complexe

FIGURE 5 – Différents comportements du modèle de Nicholson-Bailey