

# SYSTÈMES D'ÉQUATIONS DIFFÉRENTIELLES

Bernard Dupont

[Bernard.Dupont@univ-lille1.fr](mailto:Bernard.Dupont@univ-lille1.fr)

Ce chapitre adopte la même démarche que celui consacré aux équations différentielles ordinaires. La première section passe en revue les méthodes permettant une résolution explicite d'un système d'équations différentielles. La seconde expose les méthodes de résolution numérique. La troisième montre que Maple peut être très performant dans l'étude qualitative graphique des systèmes de deux EDO du premier ordre. On y retrouvera les deux commandes majeures **dsolve** et **DEplot**. La quatrième section passe en revue le "bonus" interactif que Maple met à disposition de l'utilisateur sous forme d'un "assistant" nommé ODE Analyzer.

## Résolution explicite

Pour résoudre explicitement les systèmes d'équations différentielles avec ou sans conditions sur les valeurs prises par les fonctions inconnues ou leur dérivée première en certains points de leur domaine de définition, on dispose de la même commande "à tout faire" que pour les EDO, à savoir **dsolve**.

## Écriture et résolution d'un système sans condition

En présence d'un système sans condition, on utilise **dsolve** avec les deux arguments suivants : d'abord, les équations différentielles qui sont rassemblées dans un ensemble (entre deux parenthèses) ou dans une liste (entre deux crochets); ensuite les fonctions inconnues qui sont rassemblées dans un ensemble ou une liste. La syntaxe est donc **dsolve({equa\_diff\_1, equa\_diff\_2, ..., equa\_diff\_n}, {x1(t), x2(t), ..., xn(t)})** ou **dsolve([equa\_diff\_1, equa\_diff\_2, ..., equa\_diff\_n], [x1(t), x2(t), ..., xn(t)])** pour  $n$  équations différentielles et  $n$  fonctions inconnues. Il est d'usage de commencer par assigner le système et les inconnues avant d'invoquer **dsolve**.

```
> restart;
eqd:={diff(x(t),t)=-y(t),diff(y(t),t)=x(t)};#assignation du
système différentiel
fonc:={x(t),y(t)};#assignation des fonctions inconnues
sol:=dsolve(eqd,fonc);#requête de résolution
```

$$eqd := \left\{ \frac{d}{dt} x(t) = -y(t), \frac{d}{dt} y(t) = x(t) \right\}$$

$$fonc := \{x(t), y(t)\}$$

$$sol := \{x(t) = \_C1 \sin(t) + \_C2 \cos(t), y(t) = -\_C1 \cos(t) + \_C2 \sin(t)\}$$

Maple renvoie ici un ensemble de deux **équations** et chacune donne la fonction solution, qui dépend de deux constantes notées par les identificateurs internes **\_C1** et **\_C2**. Chaque solution est récupérable en tant qu'**expression** par **assign** puis appel de la fonction.

```
> assign(sol);#transformation des égalités en assignations.
x(t);#récupération de l'expression de x(t)
```

$$\begin{aligned}
 & y(t); \# \text{récupération de l'expression de } y(t) \\
 & \quad \_C1 \sin(t) + \_C2 \cos(t) \\
 & \quad - \_C1 \cos(t) + \_C2 \sin(t)
 \end{aligned} \tag{1.1.1}$$

Après quoi on peut retirer par **unapply** les **fonctions-procédures** solutions.

$$\begin{aligned}
 & > \mathbf{x\_sol := unapply(x(t), t);} \\
 & \quad \mathbf{y\_sol := unapply(y(t), t);} \\
 & \quad \quad x\_sol := t \rightarrow \_C1 \sin(t) + \_C2 \cos(t) \\
 & \quad \quad y\_sol := t \rightarrow - \_C1 \cos(t) + \_C2 \sin(t)
 \end{aligned} \tag{1.1.2}$$

La représentation graphique des trajectoires devient possible en attribuant des valeurs numériques aux paramètres  $\_C1$  et  $\_C2$ .

## ▼ Ecriture et résolution d'un système avec conditions

Pour résoudre un problème de Cauchy, on utilise **dsolve** avec les deux arguments suivants : les équations différentielles et les conditions, qui sont le plus souvent des conditions initiales, sont écrites entre accolades ou crochets et constituent donc un ensemble ou une liste d'équations différentielles et d'égalités; les fonctions inconnues sont elles aussi écrites entre accolades ou crochets pour former un ensemble ou une liste. La syntaxe est donc **dsolve**(**{système\_différentiel, conditions}**, **{fonctions\_inconnues}**) ou **dsolve**(**[système\_différentiel, conditions]**, **[fonctions\_inconnues]**).

$$\begin{aligned}
 & > \mathbf{restart;} \\
 & \quad \mathbf{eqd := \{diff(x(t), t) = -y(t), diff(y(t), t) = x(t), x(0) = 1, y(0) = 1\};} \\
 & \quad \# \text{écriture du système différentiel et des conditions} \\
 & \quad \text{initiales} \\
 & \quad \mathbf{fonc := \{x(t), y(t)\};} \# \text{écriture des fonctions inconnues} \\
 & \quad \mathbf{sol := dsolve(eqd, fonc);} \# \text{requête de résolution} \\
 & \quad \quad \mathit{eqd} := \left\{ x(0) = 1, y(0) = 1, \frac{d}{dt} x(t) = -y(t), \frac{d}{dt} y(t) = x(t) \right\} \\
 & \quad \quad \mathit{fonc} := \{x(t), y(t)\} \\
 & \quad \quad \mathit{sol} := \{x(t) = \cos(t) - \sin(t), y(t) = \sin(t) + \cos(t)\}
 \end{aligned}$$

Certains utilisateurs préfèrent séparer l'écriture du système différentiel de celle des conditions initiales et de les réunir par **union** dans la requête de résolution.

$$\begin{aligned}
 & > \mathbf{restart;} \\
 & \quad \mathbf{eqd := \{diff(x(t), t) = -y(t), diff(y(t), t) = x(t)\};} \# \text{écriture du} \\
 & \quad \text{système différentiel} \\
 & \quad \mathbf{condinit := \{x(0) = 1, y(0) = 1\};} \# \text{écriture des conditions} \\
 & \quad \text{initiales} \\
 & \quad \mathbf{fonc := \{x(t), y(t)\};} \# \text{écriture des fonctions inconnues} \\
 & \quad \mathbf{sol := dsolve(eqd union condinit, fonc);} \# \text{requête de résolution} \\
 & \quad \text{avec réunion de eqd et condinit} \\
 & \quad \quad \mathit{eqd} := \left\{ \frac{d}{dt} x(t) = -y(t), \frac{d}{dt} y(t) = x(t) \right\} \\
 & \quad \quad \mathit{condinit} := \{x(0) = 1, y(0) = 1\} \\
 & \quad \quad \mathit{fonc} := \{x(t), y(t)\}
 \end{aligned}$$

$$sol := \{x(t) = \cos(t) - \sin(t), y(t) = \sin(t) + \cos(t)\} \quad (1.2.1)$$

Maple renvoie un ensemble d'équations à partir duquel on récupère les expressions des solutions par **assign**.

```
> assign(sol);
x(t);
y(t);
```

$$\begin{aligned} & \cos(t) - \sin(t) \\ & \sin(t) + \cos(t) \end{aligned} \quad (1.2.2)$$

Si nécessaire, on transforme ces expressions en procédures avec **unapply**.

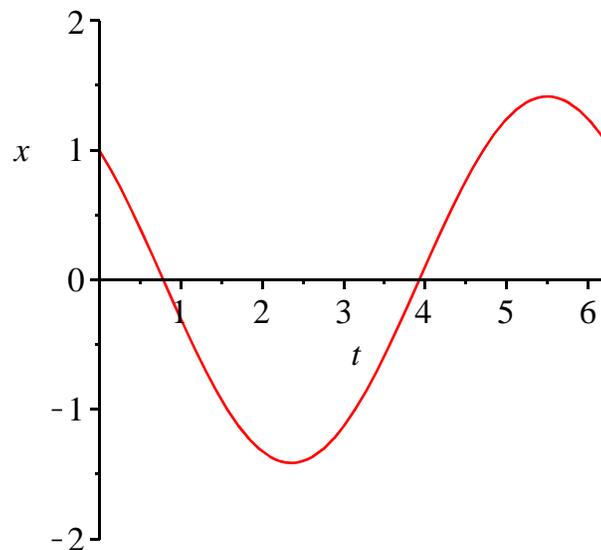
```
> x_sol:=unapply(x(t),t);
y_sol:=unapply(y(t),t);
```

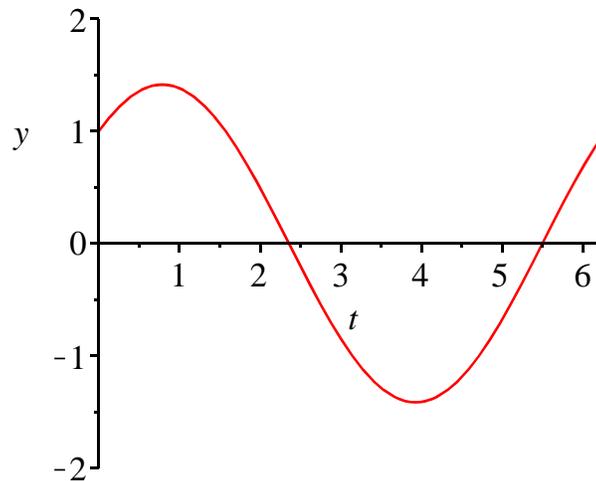
$$\begin{aligned} x\_sol &:= t \rightarrow \cos(t) - \sin(t) \\ y\_sol &:= t \rightarrow \sin(t) + \cos(t) \end{aligned} \quad (1.2.3)$$

## Représentation graphique

On obtient aisément les représentations graphiques des solutions d'un système de deux équations différentielles avec conditions à l'aide de **plot**. Si le nom de la variable doit figurer en ordonnée, la syntaxe de **plot** exige que le premier argument soit une expression et que l'intervalle d'arrivée soit explicité. En poursuivant l'exemple précédent, on aura :

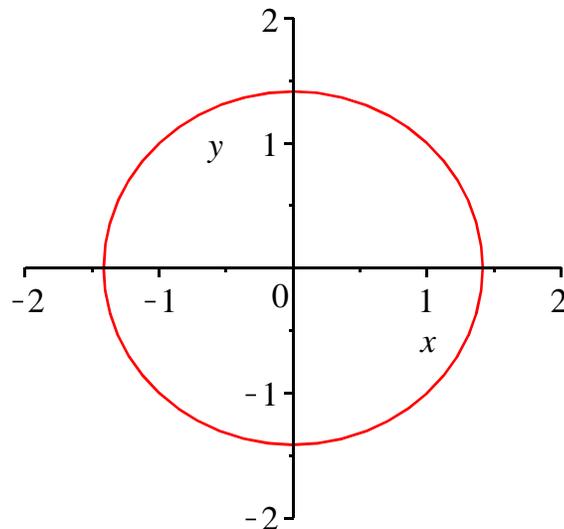
```
> plot(x(t),t=0..2*Pi,x=-2..2);#trajectoire de x
plot(y_sol(t),t=0..2*Pi,y=-2..2);#trajectoire de y
```





Les économistes préfèrent généralement tracer l'orbite de la solution d'un système avec conditions initiales, c'est à dire la courbe paramétrée  $(x(t), y(t))$  où  $x(t)$  et  $y(t)$  sont les fonctions solutions du système. Alors, `plot` a pour premier argument une liste à trois composantes du type `[x(t),y(t),t=a..b]`.

```
> plot([x(t),y(t),t=0..2*Pi],x=-2..2,y=-2..2);
```



## ▼ Commandes avancées

Des commandes spécifiques portent sur la vérification des résultats et la recherche d'informations sur le système différentiel étudié.

### ▼ Vérification des résultats

La commande `odetest` vérifie si des fonctions satisfont un système différentiel. Son premier argument est un ensemble ou une liste d'égalités de la forme `xi(t)=fi(t)` où les `xi` sont les inconnues du système différentiel et les `fi` des expressions de la variable `t`. Son second argument est l'ensemble ou la liste des équations différentielles constituant le système étudié augmentée des éventuelles conditions initiales. Cette syntaxe particulière conduit à utiliser ce test sitôt que `dsolve` a renvoyé une réponse. Si les solutions conviennent, l'écho est `{0}`. Dans le cas contraire, Maple renvoie les différences entre les dérivées des fonctions

"fausses" et les membres de droite des équations différentielles.

```
> restart;
eqd:={diff(x(t),t)=-y(t),diff(y(t),t)=x(t)};
condinit:={x(0)=1,y(0)=1};
fonc:={x(t),y(t)};
sol:=dsolve(eqd union condinit,fonc);
odetest(sol,eqd);#test sur les solutions affichées par
dsolve
odetest({x(t)=sin(t)+cos(t),y(t)=sin(t)+cos(t)},eqd);
#test sur des fonctions non satisfaisantes
```

$$\begin{aligned} \text{eqd} &:= \left\{ \frac{d}{dt} x(t) = -y(t), \frac{d}{dt} y(t) = x(t) \right\} \\ \text{condinit} &:= \{x(0) = 1, y(0) = 1\} \\ \text{fonc} &:= \{x(t), y(t)\} \\ \text{sol} &:= \{x(t) = -\sin(t) + \cos(t), y(t) = \cos(t) + \sin(t)\} \\ &\quad \{0\} \\ &\quad \{2 \cos(t), -2 \sin(t)\} \end{aligned}$$

(1.4.1.1)

### Recherche d'informations sur le système différentiel

Si le système n'est pas linéaire, suivre pas à pas les étapes de sa résolution par un logiciel de calcul formel est souvent très formateur et rend modeste tant il devient vite clair que Maple couvre et maîtrise un champ très vaste du savoir. On peut faire afficher ces informations en paramétrant l'instruction `infolevel[dsolve]` à 2 ou, de préférence, 3. Pour revenir à l'état par défaut, qui correspond à l'absence d'information, on assigne 0.

```
> infolevel[dsolve]:=3;
dsolve({diff(x(t),t)=x(t)*y(t),diff(y(t),t)=y(t)+z(t),
diff(z(t),t)=z(t)+x(t)},{x(t),y(t),z(t)});
infoleveldsolve:=3
```

```
-> Solving each unknown as a function of the next ones
using the order: [z(t), y(t), x(t)]
-> Calling odsolve with the ODE diff(diff(y(x) x) x) =
2*(diff(y(x) x))-y(x) y(x) ODE_singsol = none
Methods for second order ODEs:
--- Trying classification methods ---
trying a quadrature
checking if the LODE has constant coefficients
<- constant coefficients successful
-> Calling odsolve with the ODE diff(diff(diff(y(x) x)
x) x) = (-2*(diff(y(x) x))^3+y(x)^4-2*y(x)*(diff(y(x) x)
)^2+3*y(x)*(diff(y(x) x))*(diff(diff(y(x) x) x))+2*y(x)
^2*(diff(diff(y(x) x) x))-y(x)^2*(diff(y(x) x)))/y(x)^2
y(x) ODE_singsol = none
Methods for third order ODEs:
```

```

--- Trying classification methods ---
trying differential order: 3; missing variables
-> Computing canonical coordinates for the symmetry [1,
0]
-> Computing the inverse transformation to construct and
ODE in canonical coordinates
-> Rewriting ODE using canonical coordinates
Try computing 2 more symmetries for ODE written in
canonical coordinates
  -> Computing symmetries using: way = 3
  -> Computing symmetries using: way = exp_sym
-> Calling odsolve with the ODE (diff(diff(_b(_a) _a)
_a))*_b(_a)^2+((diff(_b(_a) _a))^2*_b(_a)*_a^2+2*_b(_a)
^3-_a^4+2*_a*_b(_a)^2-3*_a*_b(_a)^2*(diff(_b(_a) _a))-2*
_a^2*(diff(_b(_a) _a))*_b(_a)+_a^2*_b(_a))/_a^2 = 0 _b
(_a)
  *** Sublevel 2 ***
  Methods for second order ODEs:
  --- Trying classification methods ---
  trying 2nd order Liouville
  trying 2nd order WeierstrassP
  trying 2nd order JacobiSN
  differential order: 2; trying a linearization to 3rd
order
  trying 2nd order, 2 integrating factors of the form
mu(x,y)
  trying differential order: 2; missing variables
  trying symmetries linear in x and y(x)
  trying differential order: 2; exact nonlinear
  trying 2nd order, integrating factor of the form mu
(y')
  trying 2nd order, integrating factor of the form mu
(x,y')
  trying 2nd order, integrating factor of the form mu
(y,y')
  trying differential order: 2; mu polynomial in y'
  trying 2nd order, integrating factor of the form mu
(x,y)
  differential order: 2; looking for linear symmetries
  --- Trying Lie symmetry methods, 2nd order ---
  -> Computing symmetries using: way = 3
  -> Computing symmetries using: way = 5

```

```

-> Computing symmetries using: way = formal
trying differential order: 3; exact nonlinear
Trying the formal computation of integrating factors
depending on any 2 of [x, y, y', y'']
--- Trying Lie symmetry methods, high order ---

```

$$\left[ \left\{ x(t) = 0 \right\}, \left\{ y(t) = -C1 e^t + -C2 e^t \right\}, \left\{ z(t) = \frac{d}{dt} y(t) - y(t) \right\} \right], \left[ \left\{ x(t) \right. \right. \quad (1.4.2.1)$$

$$\begin{aligned}
&= -a \&where \left[ \left\{ \left( \frac{d^2}{d_a^2} -b(-a) \right) -b(-a)^2 \right. \right. \\
&+ \frac{1}{-a^2} \left( \left( \frac{d}{d_a} -b(-a) \right)^2 -b(-a) -a^2 + 2 -b(-a)^3 - -a^4 + 2 -a -b(-a)^2 \right. \\
&- 3 -a -b(-a)^2 \left( \frac{d}{d_a} -b(-a) \right) - 2 -a^2 \left( \frac{d}{d_a} -b(-a) \right) -b(-a) \\
&+ \left. \left. -a^2 -b(-a) \right) = 0 \right], \left\{ -a = x(t), -b(-a) = \frac{d}{dt} x(t) \right\}, \left\{ t = \int \frac{1}{-b(-a)} d_a \right. \\
&+ \left. \left. -C1, x(t) = -a \right\} \right], \left\{ y(t) = \frac{\frac{d}{dt} x(t)}{x(t)} \right\}, \left\{ z(t) \right. \\
&= \left. \left. \frac{x(t) \left( \frac{d^2}{dt^2} x(t) \right) - \left( \frac{d}{dt} x(t) \right)^2 - x(t) \left( \frac{d}{dt} x(t) \right)}{x(t)^2} \right\} \right]
\end{aligned}$$

## ▼ Résolution numérique d'un système d'équations différentielles

Si Maple échoue dans la résolution explicite d'un système avec conditions initiales ou encore renvoie un résultat peu lisible, on peut demander une résolution numérique en l'indiquant en option de

**dsolve**. La syntaxe est alors **dsolve({système\_différentiel+ conditions\_initiales}, {fonctions\_inconnues}, numeric, autres\_options\_éventuelles)**.

Dans cet exemple de système différentiel particulièrement difficile, **dsolve** ne donne rien.

```

> restart;
eqd:={diff(f(x),x)=f(x)*(f(x)-g(x)),diff(g(x),x)=g(x)*(f(x)+g
(x)),f(0)=1,g(0)=1};
var:={f(x),g(x)};
infolevel[dsolve]:=3;#demande de renseignements sur la

```

## méthode de résolution

**dsolve(eqd,var);**

$$\text{eqd} := \left\{ f(0) = 1, g(0) = 1, \frac{d}{dx} f(x) = f(x) (f(x) - g(x)), \frac{d}{dx} g(x) = g(x) (f(x) + g(x)) \right\}$$

$\text{var} := \{f(x), g(x)\}$

$\text{infolevel}_{\text{dsolve}} := 3$

-> Solving each unknown as a function of the next ones using the order: [f(x), g(x)]

-> Calling odsolve with the ODE  $\text{diff}(y(x) \ x) = y(x)^2 y(x)$   
explicit ODE\_singsol = none

Methods for first order ODEs:

--- Trying classification methods ---

trying a quadrature

trying 1st order linear

trying Bernoulli

<- Bernoulli successful

-> Calling odsolve with the ODE  $\text{diff}(\text{diff}(y(x) \ x) \ x) = -(2*y(x)^2*(\text{diff}(y(x) \ x))-2*(\text{diff}(y(x) \ x))^2-2*y(x)^4)/y(x) \ y(x)$

explicit ODE\_singsol = none

Methods for second order ODEs:

--- Trying classification methods ---

trying 2nd order Liouville

trying 2nd order WeierstrassP

trying 2nd order JacobiSN

differential order: 2; trying a linearization to 3rd order

trying 2nd order, 2 integrating factors of the form  $\mu(x,y)$

trying differential order: 2; missing variables

-> Computing canonical coordinates for the symmetry [1, 0]

-> Rewriting ODE in canonical coordinates by means of differential invariants

Try computing 1 more symmetries for ODE written in canonical coordinates

-> Computing symmetries using: way = 3

Found another symmetry:

$[-x, y]$

Computing a convenient ordering to use the 2 symmetries available

-> Calling odsolve with the ODE  $(\text{diff}(\_b(\_a) \ \_a))*\_b(\_a)-2*(-\_a^2*\_b(\_a)+\_b(\_a)^2+\_a^4)/\_a = 0 \ \_b(\_a)$  HINT = [[\_a 2\*\_b]]

\*\*\* Sublevel 2 \*\*\*

```

symmetry methods on request
1st order, trying reduction of order with given symmetries:
      [_a, 2*_b]
1st order, trying the canonical coordinates of the
invariance group
  -> Computing canonical coordinates for the symmetry [_a, 2*_b]
      -> Calling odsolve with the ODE diff(y(x) x) = 2*y(x)/x
y(x)
      *** Sublevel 3 ***
      Methods for first order ODEs:
      --- Trying classification methods ---
      trying a quadrature
      trying 1st order linear
      <- 1st order linear successful
  -> The canonical coordinates may not have unique inverse.
      Trying gauging the symmetry to the form [0, eta(x,y)]
  -> Computing canonical coordinates for the symmetry [0, 2*_a^2-2*_a^4/_b]
      <- 1st order, canonical coordinates successful
<- differential order: 2; canonical coordinates successful
<- differential order 2; missing variables successful

```

On tente donc une résolution numérique.

```

> infolevel[dsolve]:=0:#"neutralisation" des commentaires sur
la résolution
soleqd:=dsolve(eqd,var,numeric);
      soleqd := proc(x_rkf45) ... end proc

```

Maple renvoie une procédure, qui est en fait une boîte noire puisqu'on ne sait rien d'autre que la méthode d'approximation utilisée (par défaut, *rkf45* pour Runge-Kutta-Fehlberg au 4° et 5° ordre). Ceci dit, on connaît les valeurs prises par les solutions pour diverses valeurs de  $x$ .

```

> soleqd(0);soleqd(0.5);soleqd(10);
      [x=0., f(x) = 1., g(x) = 1.]
      [x=0.5, f(x) = 0.559654068094077916, g(x) = 4.25898581985721680]

```

Error, (in soleqd) cannot evaluate the solution further right of .72477842, probably a singularity

L'appel de la procédure pour différentes valeurs de la variable indépendante produit deux types d'écho. La détection d'un point singulier produit un message d'erreur qui nous donne une indication sur le domaine de différentiabilité des fonctions solutions. Pour les points ordinaires, le résultat est une liste d'égalités qu'on lit comme suit : quand " $x$  est égal à  $a$ ", " $f(x)$  est égal au nombre  $f(a)$ " et " $g(x)$  est égal au nombre  $g(a)$ ".

On génère les fonctions-solutions à partir de ces égalités soit au moyen de la commande **subs** soit au moyen de la commande **rhs**. Il est impératif de procéder comme suit :

```

> f_sol:=t->subs(soleqd(t),f(x));#utilisation de subs

```

```
g_sol:=t->rhs(soleqd(t)[3]);#utilisation de rhs
f_sol(0.6);
g_sol(0.25);
```

```
f_sol := t → subs(soleqd(t), f(x))
```

```
g_sol := t → rhs(soleqd(t)3)
```

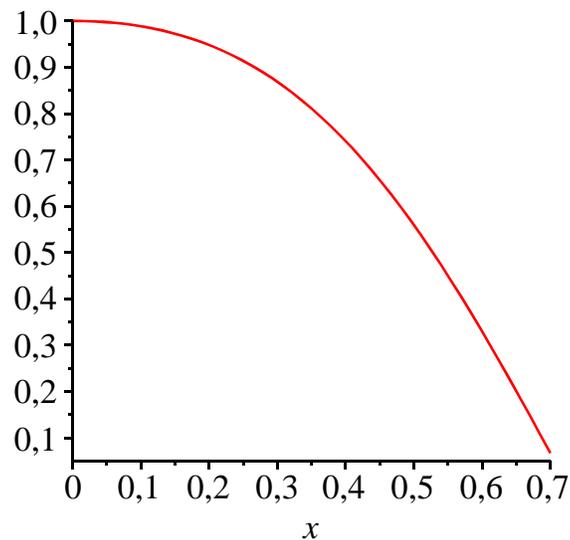
```
0.329869132065849002
```

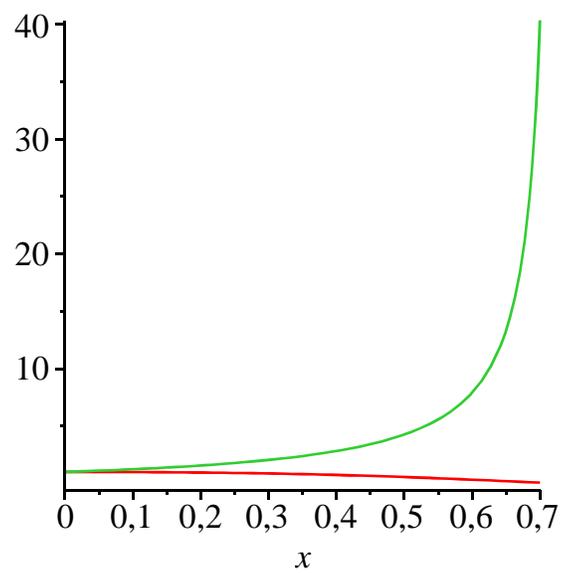
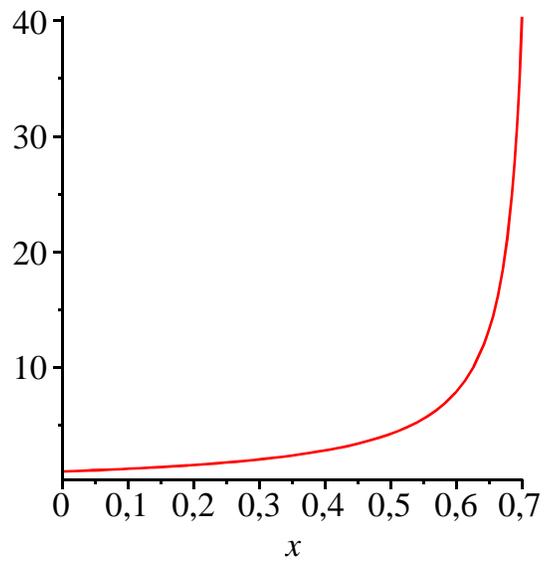
```
1.78029049822201868
```

(2.1)

Evidemment, il est plus parlant de construire une représentation graphique des solutions, soit en traçant leur trajectoire propre, soit la courbe paramétrée. A cet égard, il existe deux possibilités. La première consiste à utiliser **plot** où on place en premier argument les fonctions qu'on vient de créer entre deux "apostrophes" afin d'éviter leur évaluation immédiate. Pour les trajectoires, on aura :

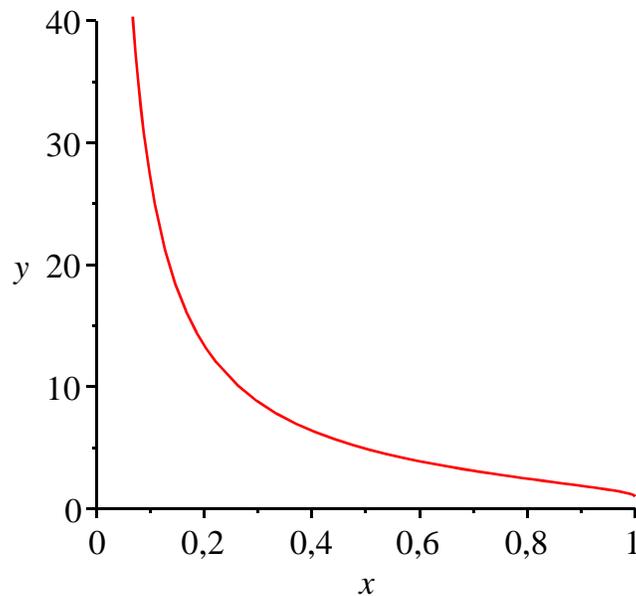
```
> plot('f_sol(x)',x=0..0.7);#trajectoire de f_sol. Notez la
présence des apostrophes.
plot('g_sol(x)',x=0..0.7);#trajectoire de g_sol.
plot({'f_sol(x)','g_sol(x)'},x=0..0.7);#trajectoires de f_sol
et g_sol dans le même graphique.
```





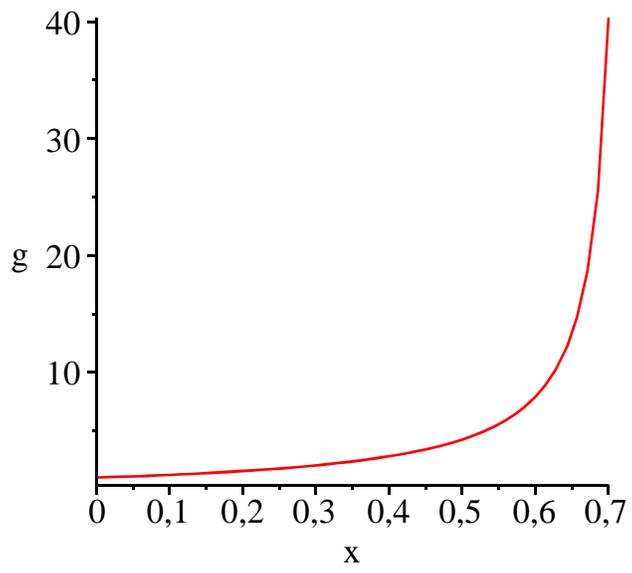
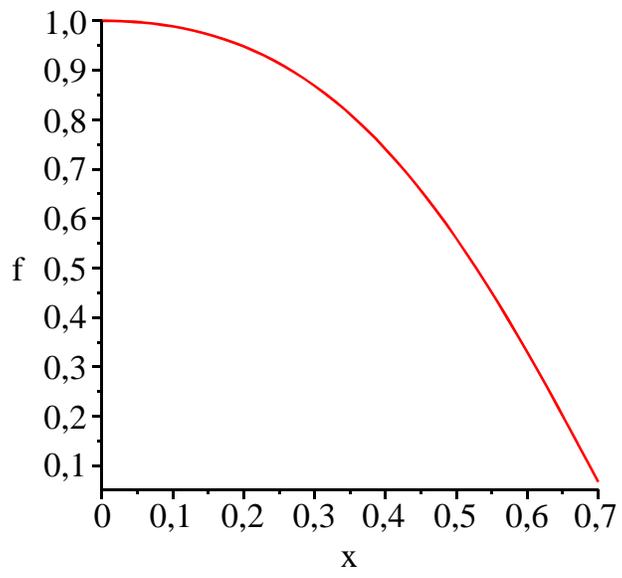
Et pour la courbe paramétrée, on procède comme suit :

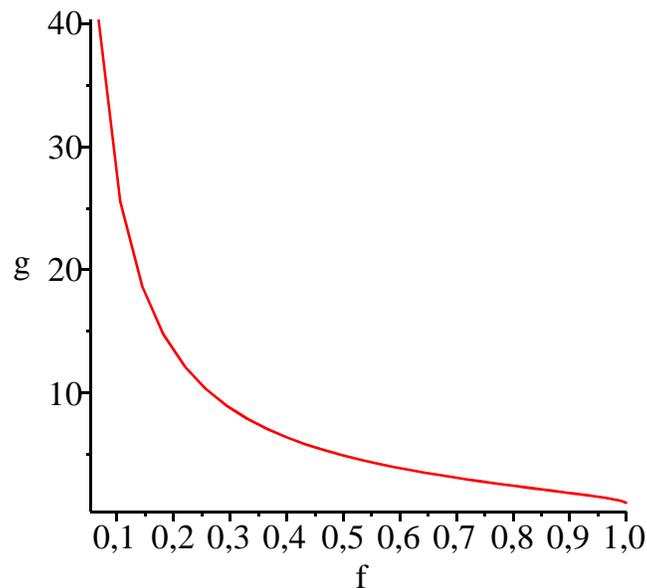
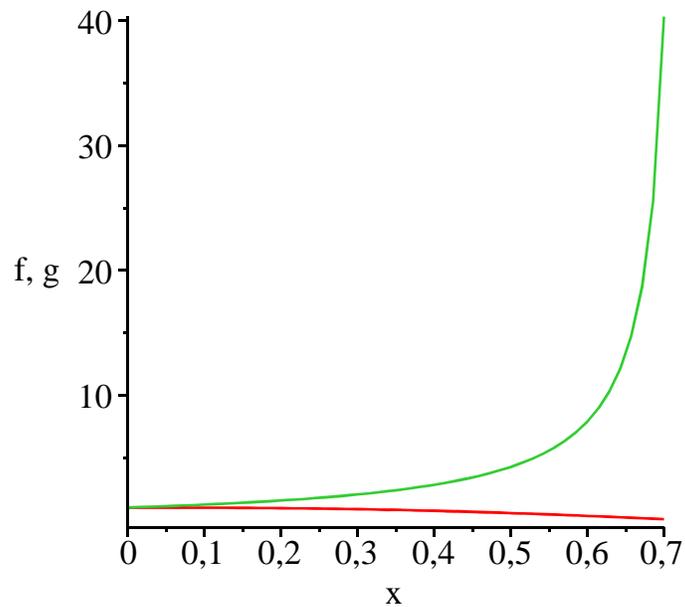
```
> plot(['f_sol(t)', 'g_sol(t)', t=0..0.7], x=0..1, y=0..40); #Notez  
la présence des apostrophes.
```



La deuxième possibilité de production de graphique à partir d'une procédure calculée par **dsolve** avec l'option **numeric** est d'employer la commande dédiée **odeplot** du paquetage **plots**. En effet, elle récupère les fonctions-solutions à l'intérieur de la liste affichée par **dsolve**. Sa syntaxe est **odeplot(soleqd, [abscisse, ordonnée], intervalle, options\_éventuelles)** où **soleqd** est le nom d'assignation de la solution du système différentiel, **abscisse** est la variable figurant en abscisse, **ordonnée** est la variable figurant en ordonnée, **intervalle** est l'intervalle de la variable indépendante, **options\_éventuelles** sont les options graphiques également proposées par **plot**.

```
> with(plots):
odeplot(soleqd, [x, f(x)], 0..0.7); #trajectoire de f
odeplot(soleqd, [x, g(x)], 0..0.7); #trajectoire de g
odeplot(soleqd, [[x, f(x)], [x, g(x)]], 0..0.7); #trajectoires de f
et g dans le même graphique
odeplot(soleqd, [f(x), g(x)], 0..0.7); #courbe paramétrée
```





Dans la pratique, l'appel naïf à **odeplot** s'avère moins fiable que la première méthode. On améliore très sensiblement ses performances en lui ajoutant l'option **refine=K** avec **K** nombre réel qui s'interprète comme un coefficient multiplicateur pour l'évaluation des points de la courbe (**K=2** double le nombre d'évaluations). Cette option précieuse doit être préparée dans la commande **dsolve** en ajoutant l'option **range=a..b**. A titre d'exemple, examinons le célèbre système prédateurs-proies de Lotka-Volterra dans lequel on a fixé (assez arbitrairement) les paramètres. Cette première approche est maladroite, décevante ... et fautive :

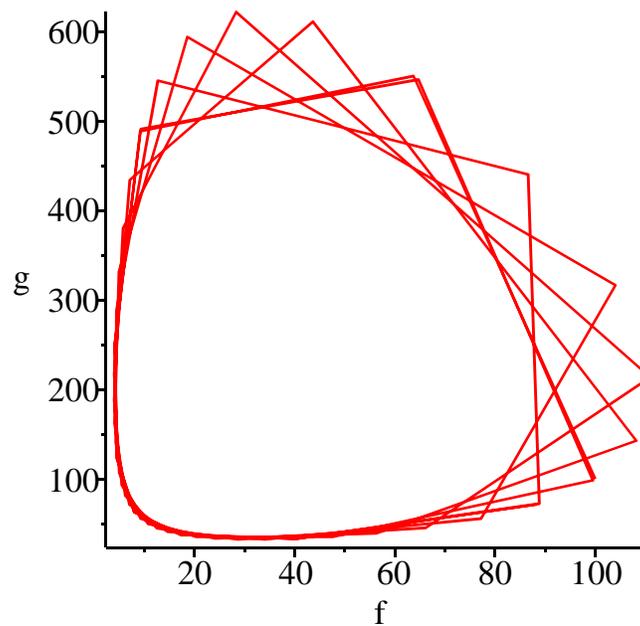
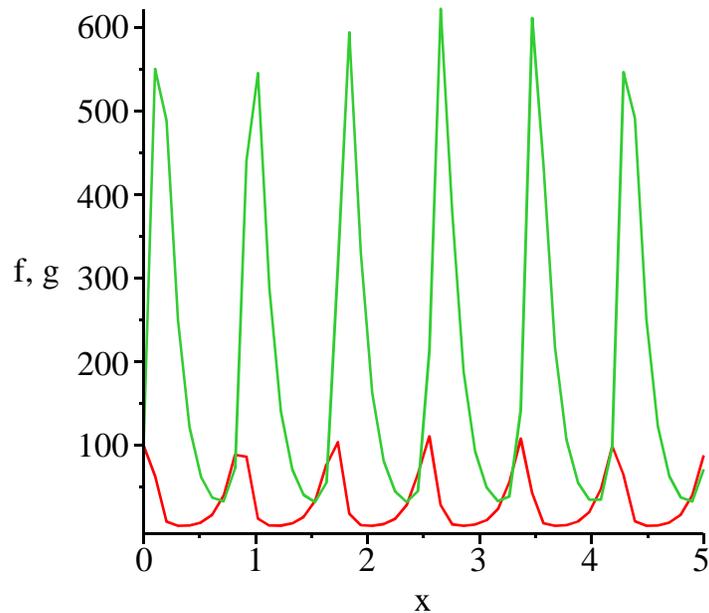
```
> restart;
with(plots):
eqd:={diff(f(x),x)=f(x)*(10-0.05*g(x)),diff(g(x),x)=g(x)*
(0.25*f(x)-8),f(0)=100,g(0)=100};
var:={f(x),g(x)};
soleqd:=dsolve(eqd,var,numeric);
```

```
odeplot(soleqd,[[x,f(x)],[x,g(x)]],x=0..5);
odeplot(soleqd,[f(x),g(x)],x=0..5);
```

$$eqd := \left\{ f(0) = 100, g(0) = 100, \frac{d}{dx} f(x) = f(x) (10 - 0.05 g(x)), \frac{d}{dx} g(x) = g(x) (0.25 f(x) - 8) \right\}$$

```
var := {f(x), g(x)}
```

```
soleqd := proc(x_rkf45) ... end proc
```

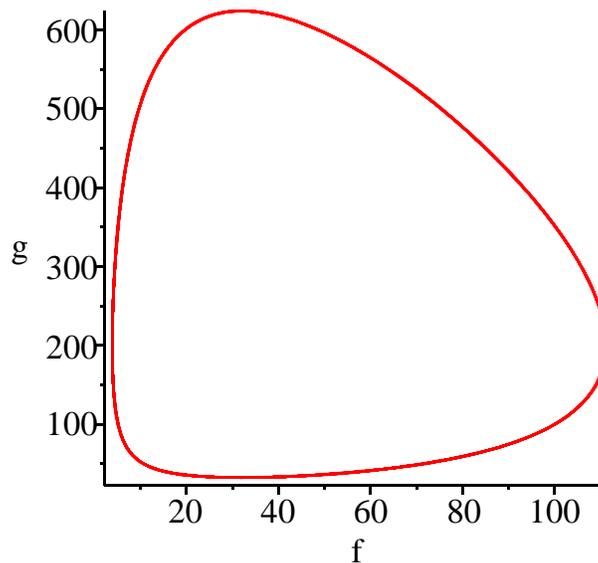
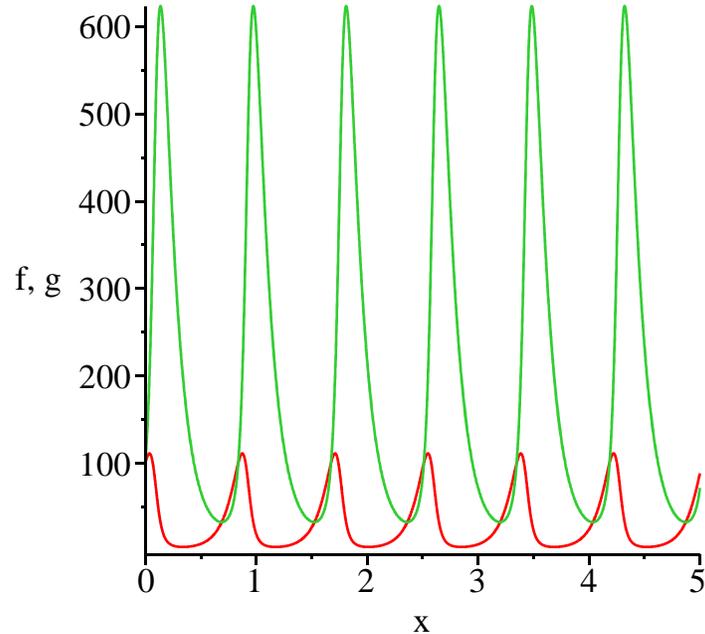


L'adjonction des options **range** dans **dplot** et **refine** dans **odeplot** améliore spectaculairement les sorties graphiques.

```
> soleqd:=dsolve(eqd,var,numeric,range=0..1);
odeplot(soleqd,[[x,f(x)],[x,g(x)]],x=0..5,refine=2);
```

```
odeplot(soleqd,[f(x),g(x)],x=0..5,refine=2);
```

```
soleqd := proc(x_rkf45) ... end proc
```



## ▼ Résolution graphique/qualitative

La résolution qualitative graphique d'un système différentiel en dimension deux est basée sur la commande **DEplot** du paquetage **DEtools**, qu'il faut donc d'abord charger. On obtient un diagramme de phase et des orbites particulièrement précis à condition d'être méthodique.

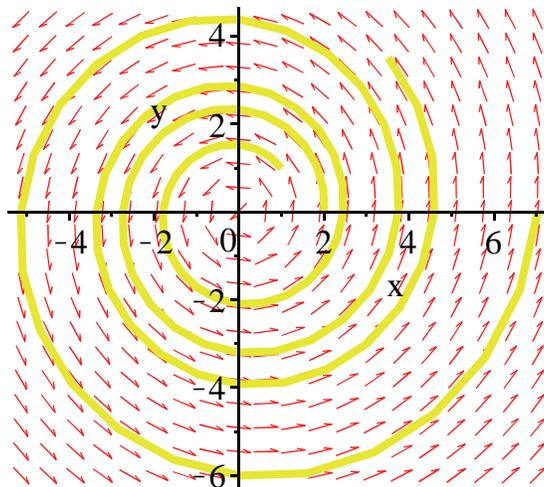
La commande **DEplot** exige au minimum quatre paramètres qui sont dans l'ordre :

- Le système d'équations écrit sous forme de liste, donc entre crochets.
- Une seconde liste contenant les fonctions inconnues, par exemple **[x(t),y(t)]**. L'ordre de ces fonctions doit être celui dans lequel les dérivées interviennent dans la liste définissant le système.

- L'intervalle de variation de la variable indépendante, par exemple :  $t=a..b$ .
- Une liste ou un ensemble de listes dont les composantes expriment des conditions initiales, par exemple  $\{[x(0)=1,y(0)=1],[x(0)=1,y(0)=0]\}$  ou  $[[x(0)=1,y(0)=1],[x(0)=1,y(0)=0],x(0)=2,y(0)=4]]$ .

L'exemple suivant va servir de fil conducteur pour montrer comment améliorer progressivement le premier graphique obtenu.

```
> restart;
with(DEtools):#chargement du paquetage
eqd:=[diff(x(t),t)=x(t)/10-y(t),diff(y(t),t)=x(t)+y(t)/10]
:#écriture du système différentiel
DEplot(eqd,[x(t),y(t)],t=0..4*Pi,{[x(0)=1,y(0)=1],[x(0)=2,y(0)=0]});#requête
```



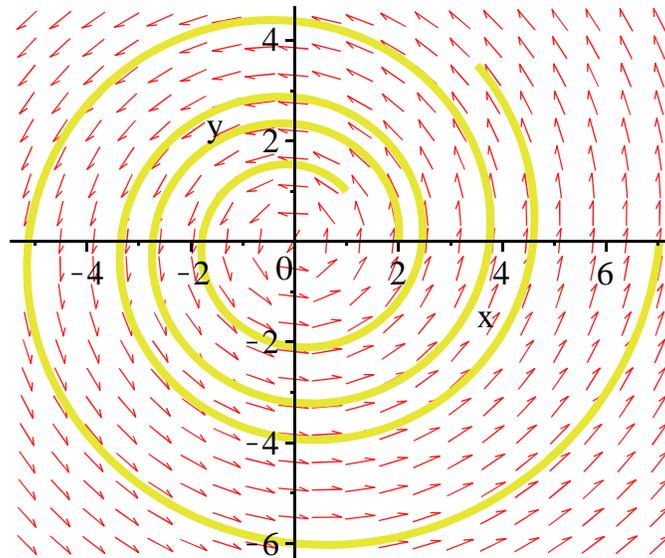
On obtient une orbite assez grossière, les courbes paramétrées étant formées d'une succession de segments. Plusieurs options permettent d'améliorer le graphique.

La première améliore la précision des tracés à l'aide de  $\text{stepsize}=R$ , où  $R$  est un réel. Par défaut,

Maple retient  $R = \frac{b-a}{20}$ ,  $a$  et  $b$  étant les bornes du domaine parcouru par la variable indépendante  $t$ .

On a intérêt à fixer  $R$  à un niveau plus faible tout en évitant des représentations farfelues car il arrive qu'un excès de précision mène droit à une représentation erronée.

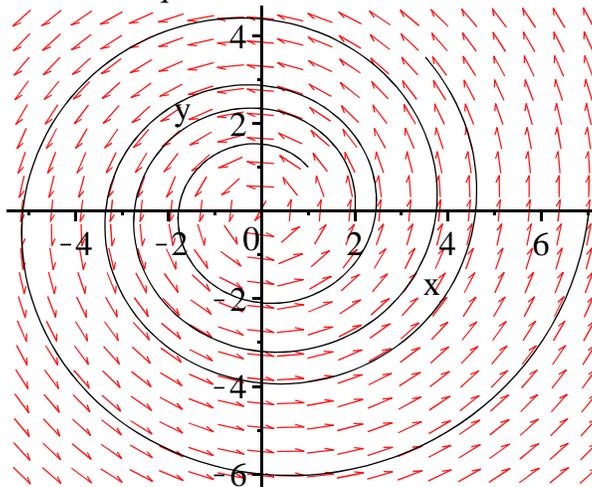
```
> DEplot(eqd,[x(t),y(t)],t=0..4*Pi,{[x(0)=1,y(0)=1],[x(0)=2,y(0)=0]},stepsize=0.05);
```



La seconde est d'utiliser la palette des options de **plot**. Par exemple, l'épaisseur de la courbe se modifie par **thickness=N**, où **N** est un entier naturel. La couleur de la courbe, quant à elle, est fixée par **linecolor=nom de couleur** ou toute autre façon de régler un colori sous Maple. On peut aussi ajouter un titre, une légende, etc..

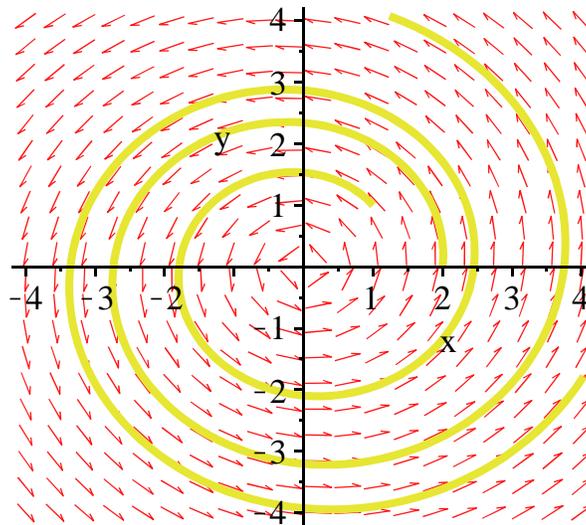
```
> DEplot(eqd,[x(t),y(t)],t=0..4*Pi,{[x(0)=1,y(0)=1],[x(0)=2,y(0)=0]},stepsize=0.05,thickness=0,linecolor=black,title='Fig.1. Un exemple de résolution qualitative');
```

*Fig.1. Un exemple de résolution qualitative*



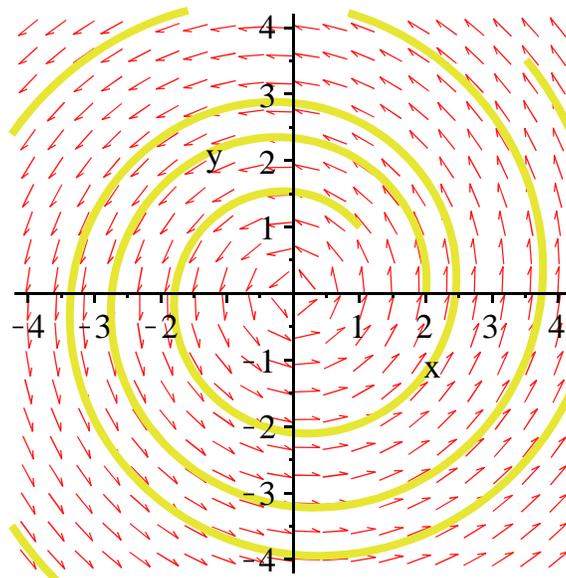
On peut régler les intervalles des variables dépendantes avec les options **x(t)=m..n** et **y(t)=p..q**.

```
> DEplot(eqd,[x(t),y(t)],t=0..4*Pi,{[x(0)=1,y(0)=1],[x(0)=2,y(0)=0]},stepsize=0.05,x(t)=-4..4,y(t)=-4..4);
```



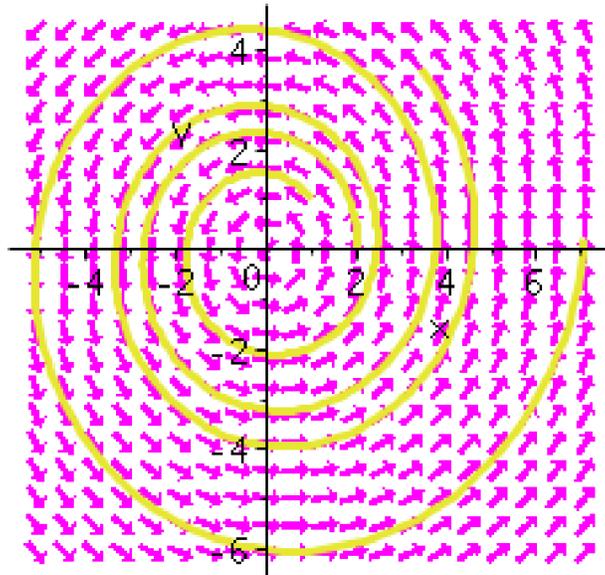
Par défaut, Maple cesse de calculer les trajectoires dès qu'une variable sort du domaine précisé en option. L'option **obsrange=false** l'oblige à calculer toutes les trajectoires et à afficher le résultat dans le rectangle retenu.

```
> DEplot(eqd,[x(t),y(t)],t=0..4*Pi,{[x(0)=1,y(0)=1],[x(0)=2,y(0)=0]},stepsize=0.05,x(t)=-4..4,y(t)=-4..4,obsrange=false);
```



Quand on étudie un système de deux équations différentielles autonomes, Maple affiche non seulement la courbe paramétrée mais aussi le champ de vecteurs (appelé aussi champ de directions ou champ directionnel) associés au système, c'est à dire un ensemble de flèches tangentes aux courbes solutions du système sans condition initiale. Dans tous les autres cas (systèmes de plus de deux équations différentielles, systèmes non autonomes), le champ de vecteurs n'apparaît pas. Les flèches sont réglables par **arrows=small** (qui est l'option par défaut), **medium**, **large**, **line** ou **none** et par **color** (ou **colour**)=**nom de couleur** ou toute autre façon de régler un colori sous Maple.

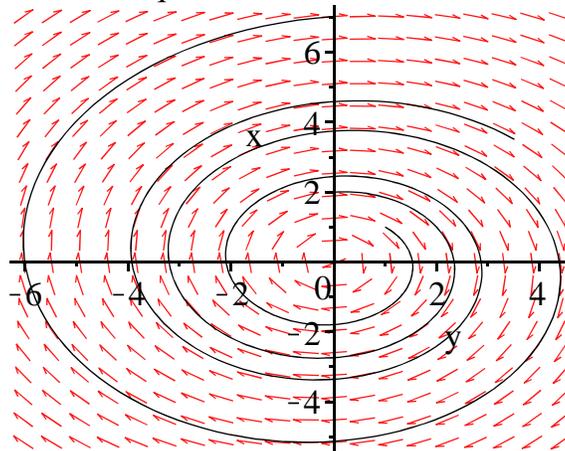
```
> DEplot(eqd,[x(t),y(t)],t=0..4*Pi,{[x(0)=1,y(0)=1],[x(0)=2,y(0)=0]},stepsize=0.05,arrows=large,color=magenta);
```



Enfin, l'option `scene=[nom1,nom2]` indique quelle doivent être la variable en abscisse et la variable en ordonnée.

```
> DEplot(eqd,[x(t),y(t)],t=0..4*Pi,{[x(0)=1,y(0)=1],[x(0)=2,y(0)=0]},stepsize=0.05,scene=[y,x],thickness=0,linecolor=black,title=`Fig.1. Un exemple de résolution qualitative`);
```

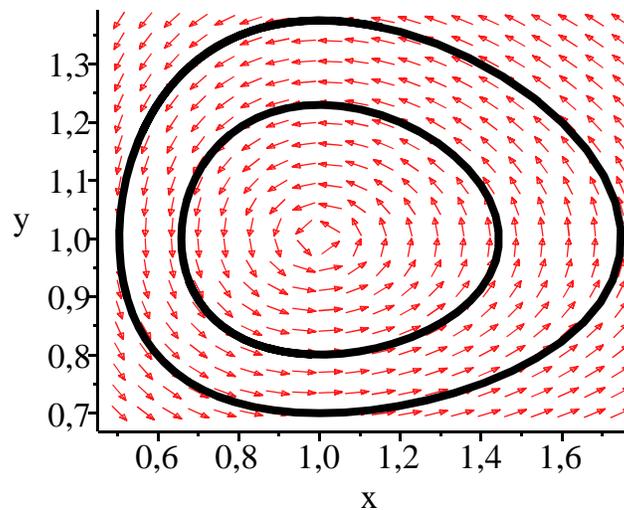
*Fig.1. Un exemple de résolution qualitative*



Terminons cette section en revenant sur le modèle prédateurs-proies dont l'intérêt méthodologique est immense.

```
> DEplot([diff(x(t),t)=x(t)*(1-y(t)),diff(y(t),t)=.3*y(t)*(x(t)-1)],
[x(t),y(t)],t=-7..7,[x(0)=1.2,y(0)=1.2],[x(0)=1,y(0)=.7]],
stepsize=.2,
title=`Fig. 2. Modèle de Lotka-Volterra`,linecolor=black,
arrows=MEDIUM,method=rkf45);
```

Fig. 2. Modèle de Lotka-Volterra



On remarque que l'option **method** a été réglée sur **rkf45**, signifiant qu'on force Maple à adopter la méthode numérique "approche de Fehlberg pour Runge-Kutta au 4° et 5° ordres". Par défaut, la méthode d'intégration est **classical[rk4]**. D'autres méthodes sont possibles : **classical**, **classical[adambash]**, **classical[abmoulton]**, **classical[foreuler]**, **classical[heunform]**, **classical[impoly]**, **classical[rk2]**, **classical[rk3]**, **dverk78**, **gear**, ..., **isode**, ..., **mgear**, ..., **rosenbrock**, **taylorseries**. On se référera à l'aide en ligne pour en connaître les particularités.

```
> DEtools[DEplot][interactive]();
```

## Bonus

Il faut bien comprendre que les logiciels de calcul formel sont devenus extrêmement performants depuis quelques années dans le domaine des équations différentielles, que ce soit dans la recherche de solutions explicites que dans celui de la résolution numérique. Ces progrès correspondent à un besoin croissant des scientifiques et des techniciens, grands consommateurs de systèmes dynamiques. Désormais, nombre de problèmes sont classés, typés, et il est devenu possible d'automatiser leur résolution. Maple offre logiquement un "outil" pédagogique et un "outil" interactif qui facilite l'écriture et l'analyse des systèmes différentiels et permet d'accéder rapidement à leur solution.

### Bonus pédagogique

Pour qui est fasciné par les systèmes différentiels en dimension deux, un tutorat introduisant aux problèmes célèbres est proposé dans Tools→Tutors→Differential Equations→DE Plots. Une fenêtre intitulée **DEplot[interactive]** invite à explorer les propriétés de 9 modèles dynamiques non linéaires physiques ou médicaux non sans rappeler leur forme structurelle. On peut en particulier y visualiser les conséquences d'une modification des paramètres ou de l'intervalle de valeurs de la variable indépendante.

```
> dsolve[interactive]();
```

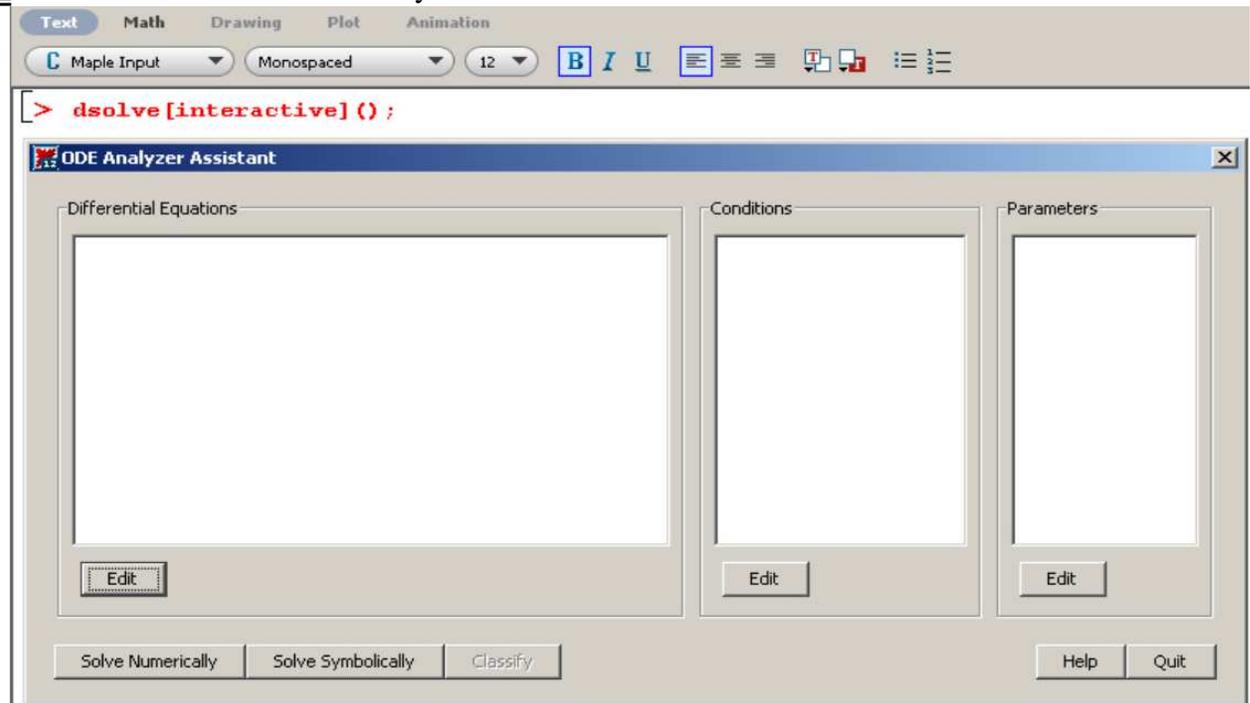
### Bonus interactif

En plus d'un tuteur, Maple offre les services d'un assistant spécialisé dans la résolution des

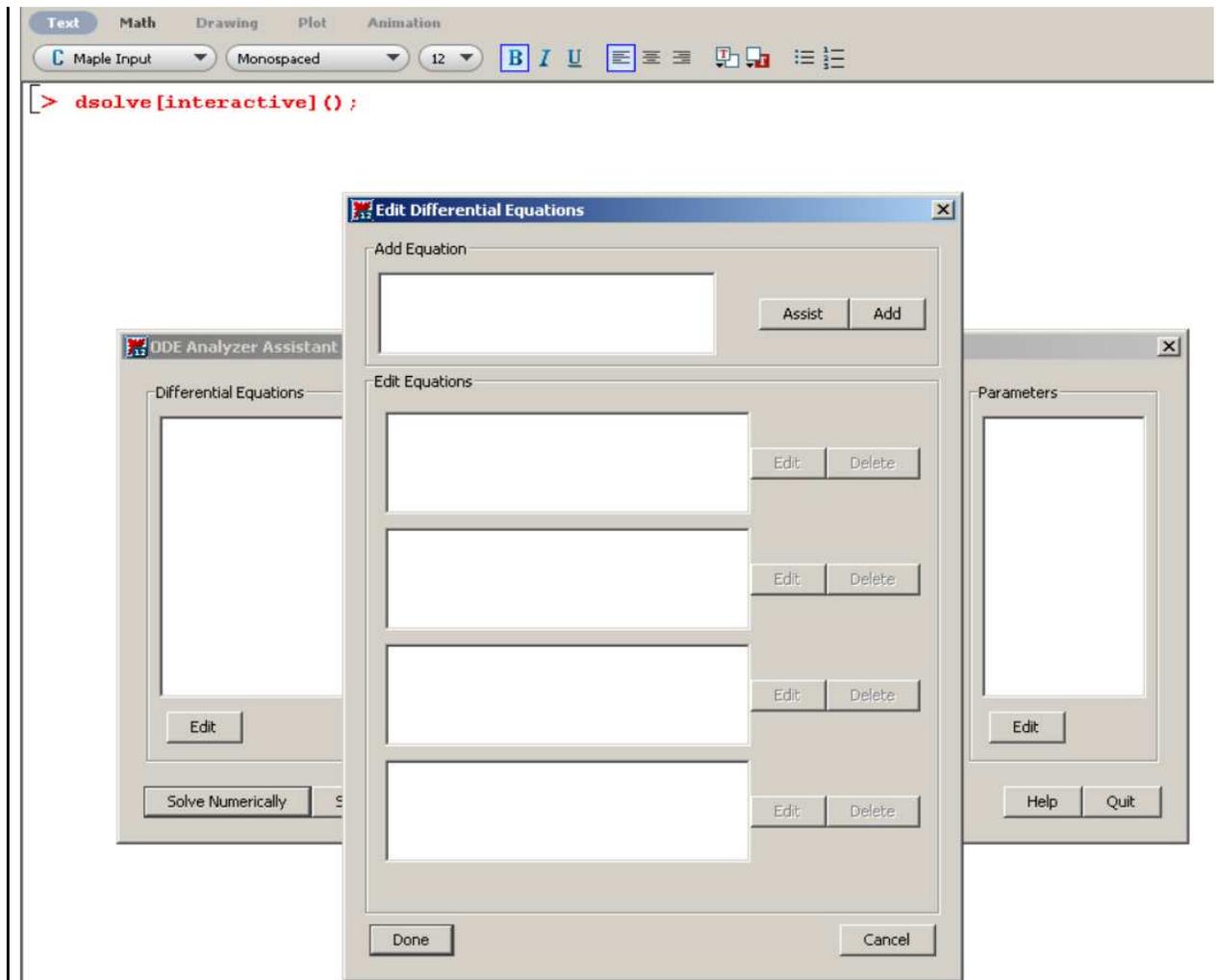
systèmes différentiels appelé ODE Analyzer Assistant. On y accède par la combinaison Tools→Assistants→ODE Analyzer ou par validation de la commande suivante :

```
> dsolve[interactive]();
```

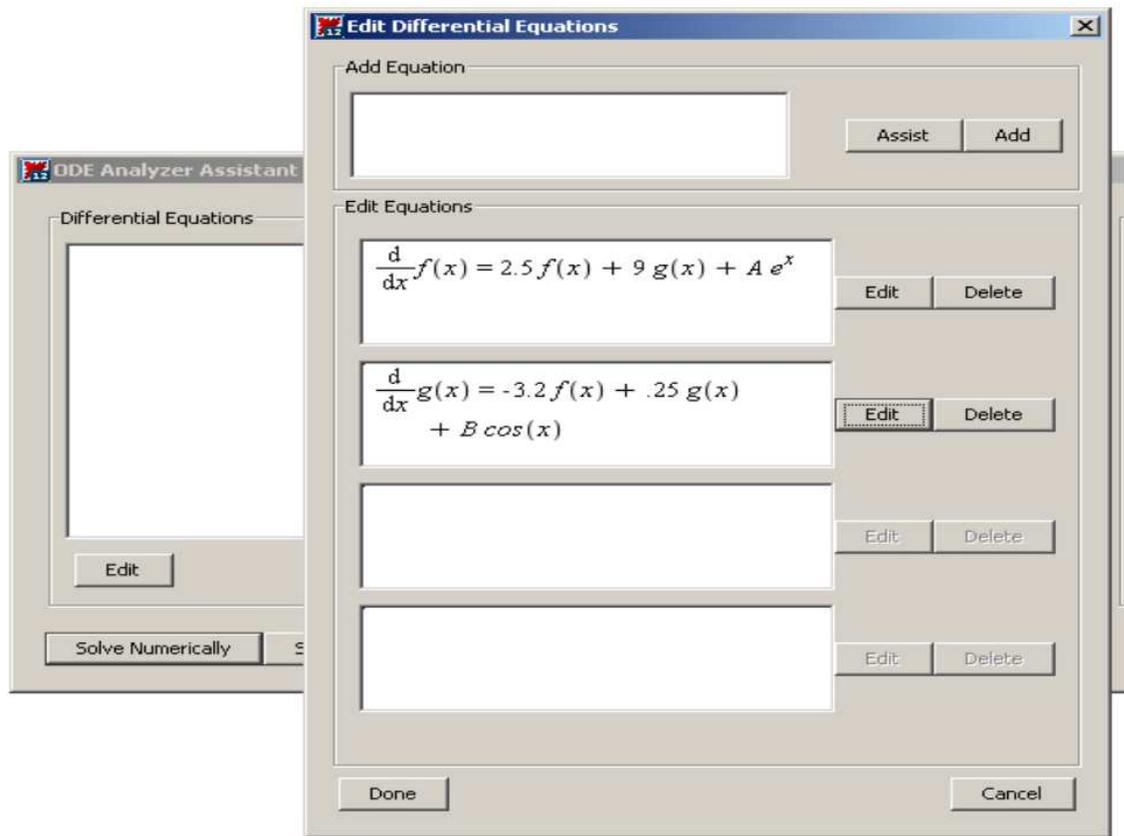
Une fenêtre intitulée ODE Analyzer Assistant s'affiche :



En cliquant sur le bouton Edit sous l'intitulé Differential Equations, on voit s'afficher la sous-fenêtre Edit Differential Equations :



Le cadre Add Equation permet d'écrire les équations différentielles en respectant la syntaxe des inputs Maple. Le cas échéant, on se fera aider par le bouton Assist pour insérer des dérivées. Une fois la frappe de l'équation terminée, on clique sur Add pour l'édition et l'équation se place dans un des quatre cadres situé sous le label Edit Equations.

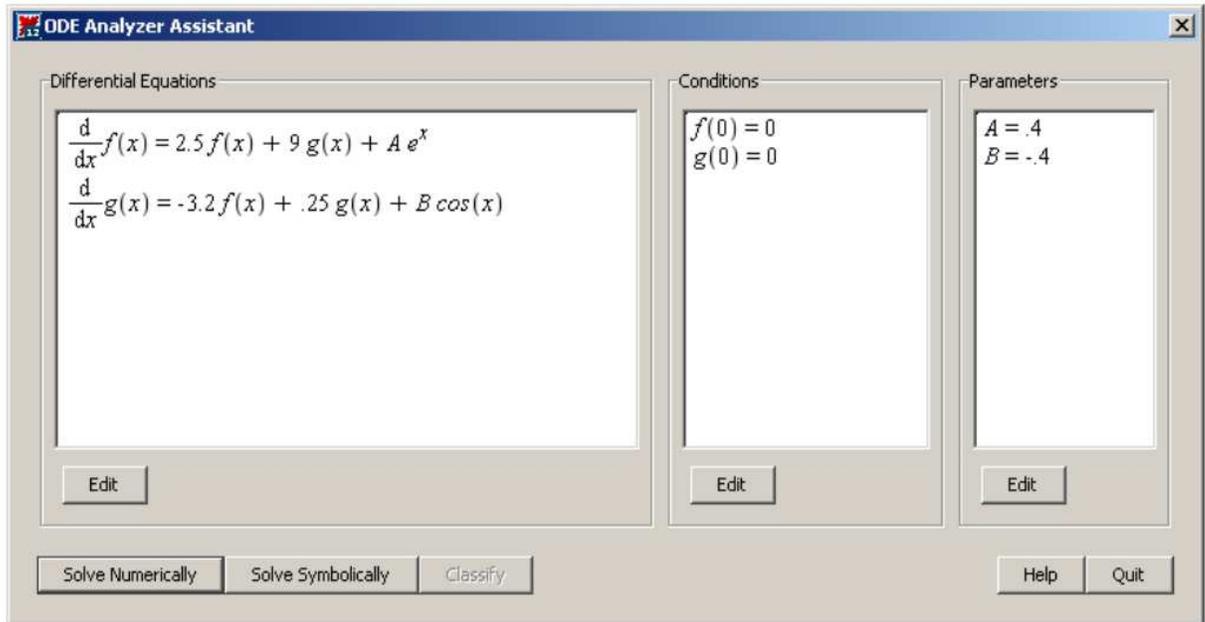


Il est toujours possible de rectifier une équation en cliquant sur le bouton Edit correspondant. Sa suppression pure et simple se fait par Delete.

Une fois le système différentiel écrit, on revient à la page ODE Analyzer Assistant avec Done. Le système à traiter s'affiche sous le cadre Differential Equations. Eventuellement, Maple détecte des paramètres et les affiche dans le cadre Parameters.

L'édition des conditions et des paramètres se fait de la même manière via Edit. Une fois remplie, la fiche ressemble à celle-ci :

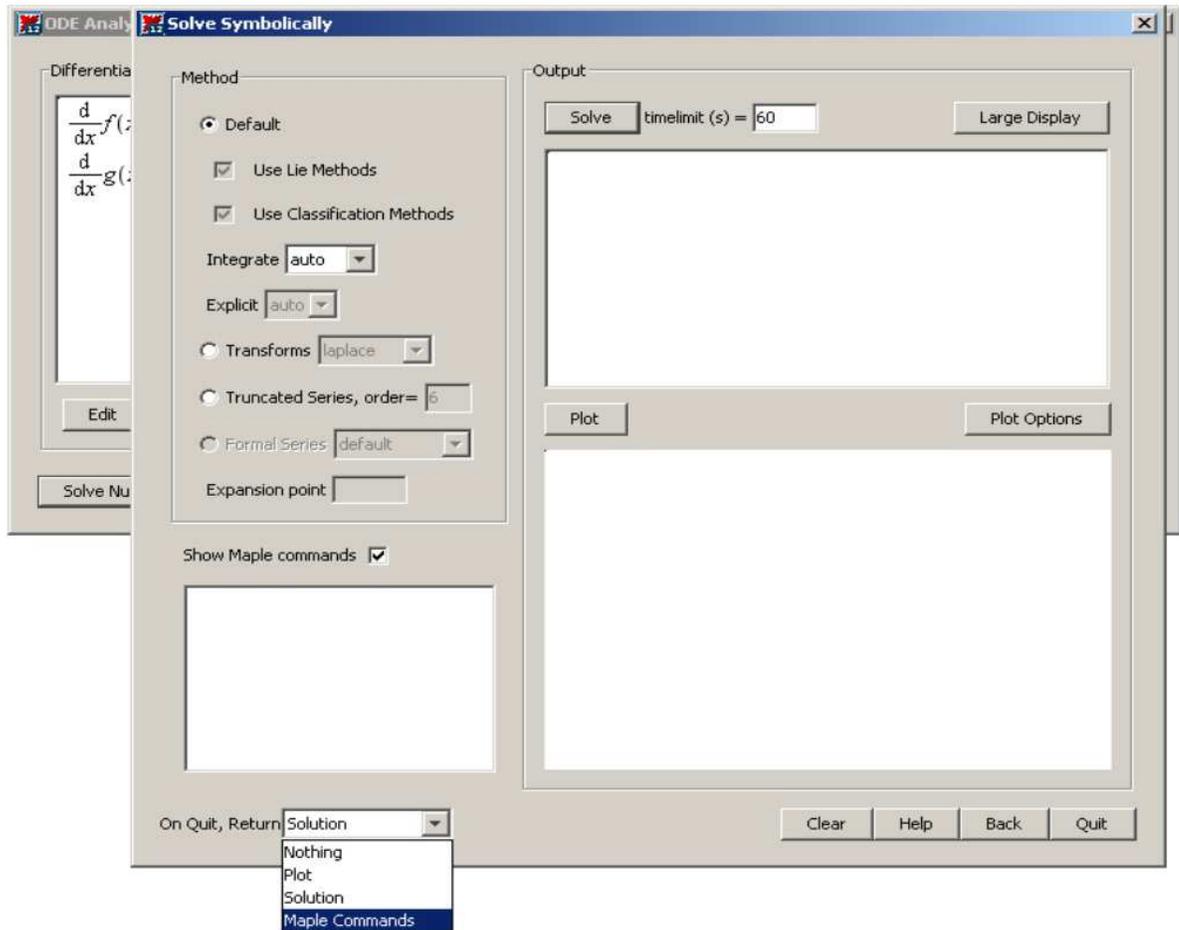
> dsolve [interactive] ();



L'assistant propose deux modes de résolution : numérique et symbolique (explicite).

Commençons par la résolution explicite puisque les économistes ont le plus souvent affaire à des systèmes linéaires dont il est toujours possible - sauf cas pathologiques - de connaître les trajectoires-solutions. Il suffit de remplir correctement la fiche Solve Symbolically.

`> dsolve[interactive]();`



Le cadre Method fixe la méthode de résolution. En économie, ce sera généralement Default et parfois Transforms laplace. Si on est curieux, il est bon de cocher la case Show Maple commands afin de visionner les inputs correspondants à la requête. Le résultat est affiché en cliquant sur le bouton Solve (dans les problèmes ardues, ne pas hésiter à fixer timelimit(s) à une valeur élevée). L'édition est en pleine page avec le bouton Large Display. Si les paramètres sont tous numériques et si les conditions sont complètes, on obtient une représentation graphique, paramétrable par la fiche Plot Options, des solutions. On sort de l'assistant par Quit. En paramétrant le questionnaire On Quit, Return ... le résultat recherché s'intercale automatiquement dans la feuille de travail. Ainsi la fiche suivante :

Method

- Default
- Use Lie Methods
- Use Classification Methods

Integrate

Explicit

Transforms

Truncated Series, order=

Formal Series

Expansion point

Show Maple commands

```
soll :=
dsolve([diff(f(x), x) =
2.5*f(x)+9*g(x)+.4*exp
(x), diff(g(x), x) =
-3.2*f(x)+.25*g(x)-.4*
cos(x), f(0) = 0, g(0)
= 0], {f(x), g(x)});
```

Output

Solve | timelimit (s) =  | Large Display

$$\left\{ \begin{aligned} f(x) &= \frac{1408526852}{2356926155595} \%1 \%3 \sqrt{4895} \\ &+ \frac{55196156}{481496661} \%1 \%2 \\ &+ \frac{15840}{1304869} \sin(x) + \frac{4}{369} e^x \\ &- \frac{163728}{1304869} \cos(x), g(x) = \end{aligned} \right.$$

Plot | Plot Options

On Quit, Return

Clear | Help | Back | Quit

donne la sortie suivante dans la feuille de travail :

```
> dsolve[interactive] ();
```

$$\left\{ \begin{aligned} f(x) &= \frac{1408526852}{2356926155595} e^{\frac{11}{8}x} \sin\left(\frac{3}{40} \sqrt{4895} x\right) \sqrt{4895} \\ &+ \frac{55196156}{481496661} e^{\frac{11}{8}x} \cos\left(\frac{3}{40} \sqrt{4895} x\right) + \frac{15840}{1304869} \sin(x) + \frac{4}{369} e^x \\ &- \frac{163728}{1304869} \cos(x), g(x) = -\frac{1456565432}{1414155693357} e^{\frac{11}{8}x} \sin\left(\frac{3}{40} \sqrt{4895} x\right) \sqrt{4895} \\ &+ \frac{72573064}{7222449915} e^{\frac{11}{8}x} \cos\left(\frac{3}{40} \sqrt{4895} x\right) + \frac{47240}{1304869} \cos(x) - \frac{256}{5535} e^x \\ &+ \frac{13792}{1304869} \sin(x) \end{aligned} \right. \quad (1)$$

```
>
```

La résolution numérique doit être réservée aux systèmes non linéaires. La fiche Solve Numerically se remplit intuitivement : choix d'une méthode d'approximation dans le cadre Parameters; possibilité de connaître la valeur prise par la fonction-procédure en un point en remplissant le cadre Show function values at x=; représentation graphique en cliquant sur Plot, sachant que Plot Options permet d'améliorer le graphe construit par défaut; mise à disposition du code Maple à la demande (ces commandes sont "copiables" puis "collables" dans la feuille de travail); enfin, récupération du graphe, de la solution ou des commandes par la rubrique On Quit,

Return.

**Solve Numerically**

Parameters

- Runge-Kutta-Fehlberg 4-5th order
- Dverk 7-8th order interpolant
- Gear single step extrapolation rational
- Rosenbrock stiff 3-4th order
- Livermore stiff adams iterative
- Boundary Value Problem solver
- trapezoidal richardson extrapolation
- Range of x: 0 to 10
- Taylor series lazy series
- Modified Extended BDF Implicit
- Fixed step methods
- .5e-2 forward Euler
- Absolute: 1.000000e-07 default
- Relative: 1.000000e-06 default

On Quit, Return Plot

Output

Show function values at x = 1 Solve

f =  
.607494158413483464e-1  
g = .168187957719889620 Plot

Plot Options

Show Maple commands

```
sol1 := dsolve([diff(f(x),x) =  
2.5*f(x)+9*g(x)+.4*exp(x), diff(g(x),x)  
= -3.2*f(x)+.25*g(x)-.4*cos(x), f(0) =  
0, g(0) = 0], numeric);  
sol1(1);  
sol2 := dsolve([diff(f(x),x) =
```

Clear Help Back Quit

## Exercices

### Exercice M1

Résoudre explicitement les systèmes de deux équations différentielles suivants :

1.

$$\begin{cases} x'(t) = x(t) + y(t) \\ y'(t) = 2x(t) \end{cases}$$

$$2. \begin{cases} x'(t) = 2x(t) - 2y(t) \\ y'(t) = x(t) - y(t) \end{cases}$$

### Exercice M2

On considère le système de deux équations différentielles suivant :

$$\begin{cases} x'(t) = x(t) + y(t) \\ y'(t) = 2x(t) - y(t) \end{cases}$$

Résoudre explicitement ce système en tenant compte des conditions suivantes :

1.  $x(0) = y(0) = 1$

2.  $x(1) = \frac{1}{\sqrt{3}}$  et  $y(5) = \frac{1}{5\sqrt{3}}$

3.  $x(0) = 1$  et  $x'(0) = 1$

4.  $x'(0) = -1$  et  $y'(0) = -1$

Dans chaque cas, représenter graphiquement les trajectoires des fonctions solutions.

### Exercice M3

1. Résoudre le système de deux équations différentielles suivant :

$$\begin{cases} x''(t) + 3y'(t) - 4x(t) + 6y(t) = 0 \\ x'(t) + y''(t) - 2x(t) + 4y(t) = 0 \end{cases}$$

2. Montrer que le profil de la trajectoire de  $x$  et l'orbite  $(x(t), y(t))$  dépendent des valeurs prises par les paramètres  $_C1$ ,  $_C2$ ,  $_C3$  et  $_C4$ .

### Exercice M4

Résoudre le système différentiel suivant :

$$\begin{cases} x' = x - y + 4z \\ y' = 3x + 2y - z \\ z' = 2x + y - z \end{cases}$$

On utilisera la notation en **D** pour écrire les équations différentielles dans **dplot**.

### Exercice M5

Caractériser, résoudre à la main puis résoudre avec Maple les systèmes suivants :

1. 
$$\begin{cases} x'(t) + y'(t) - y(t) = \sin(t) \\ x'(t) - y'(t) - x(t) = t \end{cases}$$

2. 
$$\begin{cases} 2x'(t) + y'(t) - 3x(t) - y(t) = t \\ x'(t) + y'(t) - 4x(t) - y(t) = e^t \end{cases}$$

$$3. \begin{cases} x'(t) = \frac{1}{t}x(t) - \frac{1}{t^2}y(t) \\ y'(t) = 2x(t) - \frac{1}{t}y(t) \end{cases}$$

### Exercice M6

On considère le système différentiel suivant :

$$\begin{cases} x' = -x + y \\ y' = -y + z \\ z' = -z + e^{-t} \end{cases}$$

Ecrire ce système sous forme matricielle. Quelle est sa particularité?

Intégrer ce système avec **dsolve** puis représenter, à l'aide de deux méthodes différentes, l'orbite démarrant en  $x(0) = y(0) = z(0) = 0$ .

### Exercice M7

Soit le système différentiel ( $S$ ) suivant :

$$\begin{cases} x' = x + y - z + e^t \\ y' = -x + y + z \\ z' = x - y - z \end{cases}$$

1. Ecrire ( $S$ ) sous la forme matricielle  $X' = AX + B$  puis vérifier que deux valeurs propres de  $A$  sont complexes.

2. Déterminer les solutions réelles de ( $S$ ).

3. Représenter trois orbites solutions de ( $S$ ).

### Exercice M8

On considère le système différentiel suivant :

$$\begin{cases} x' = -y + \sin(\alpha t) \\ y' = x - \cos(\alpha t) \end{cases}$$

où  $\alpha$  est un réel.

1. Résoudre le système quand  $\alpha = 1$ . Représenter les trajectoires de  $x(t)$  et  $y(t)$  pour les conditions initiales  $(x(0), y(0)) = (0, 0)$ ,  $(x(0), y(0)) = (0, 2)$ ,  $(x(0), y(0)) = (0, 4)$  et  $(x(0), y(0)) = (0, 6)$ . A l'aide de deux méthodes différentes, représenter les quatre orbites correspondantes dans le plan des phases.

2. Même question que précédemment en prenant  $\alpha \neq 1$  et  $\alpha \neq -1$ .

### Problème E1 (modèle monétariste inflation-chômage)

Le modèle monétariste canonique d'inflation chômage repose sur 4 équations structurelles, dont deux sont statiques et deux sont dynamiques.

La première équation statique décrit la formation des salaires nominaux à la manière de M. Friedman réinterprétant la fameuse relation de Phillips : le taux de croissance des salaires

nominaux, noté  $w$ , dépend négativement du niveau du taux de chômage, noté  $U$ , et positivement du taux d'inflation anticipé, noté  $\pi$ . Plus le taux de chômage est élevé, moins les salariés sont revendicatifs et "gourmands". Plus les anticipations d'inflation sont élevées, plus ils demandent des augmentations salariales afin de conserver leur pouvoir d'achat. Sous sa forme linéaire, l'équation s'écrit :

$$(1) \quad w(t) = -\beta U(t) + \alpha + h \pi(t) \text{ avec } \alpha \text{ et } \beta \text{ réels strictement positifs et } 0 < h \leq 1$$

La deuxième équation statique dérive le taux d'inflation effectif, noté  $p$ , du taux de croissance des salaires nominaux. Comme les firmes fixent leur prix suivant le principe du mark-up sur les coûts et que les coûts sont essentiellement les coûts salariaux, le taux de croissance des prix est la différence entre le taux de croissance des salaires nominaux et le taux de croissance de la productivité du travail supposé constant, noté  $T$ . On a donc :

$$(2) \quad p(t) = w(t) - T$$

La première équation dynamique explique la formation des anticipations de prix par les agents économiques. L'hypothèse d'anticipations adaptatives stipule que les variations d'anticipations de prix corrigent l'écart constaté entre l'inflation effective et l'inflation prévue :

$$(3) \quad \pi'(t) = j (p(t) - \pi(t)) \text{ avec } 0 < j \leq 1$$

La deuxième équation dynamique décrit l'effet de l'inflation sur les variations du taux de chômage. Pour la comprendre, il faut se placer dans une optique monétariste de la macroéconomie. Les agents dépensent selon l'état de leurs encaisses réelles et donc les variations de la demande globale dépendent des variations de leurs encaisses réelles, soit  $m - p$ . La banque centrale contrôle  $m$ , le taux de croissance de la masse monétaire, qu'on suppose constant. Ainsi, toute augmentation des encaisses réelles a un effet stimulant sur l'activité et fait diminuer le taux de chômage et, inversement, toute baisse des encaisses réelles déprime la demande et fait monter le taux de chômage. Cette relation de causalité des encaisses vers les variations du chômage s'écrit :

$$(4) \quad U'(t) = -k (m - p(t)) \text{ avec } k > 0$$

L'objectif de cet exercice est d'étudier les évolutions conjointes du taux de chômage et des anticipations de prix.

1. A l'aide des équation (1) et (2), donner la relation entre le taux d'inflation et le taux de chômage.

2. En déduire que l'évolution conjointe des anticipations de prix et du taux de chômage est décrite par un système différentiel linéaire.

3. Ecrire ce système différentiel sous la forme matricielle  $X' = A X + B$ . Montrer que la matrice  $A$  est différentiellement stable.

4. Déterminer l'équilibre dynamique du système différentiel. En tenant compte de la question précédente, conclure sur sa nature.

5. On veut représenter plusieurs trajectoires pour les solutions et les orbites correspondantes dans le plan  $\pi$   $U$ . Pour cela, on attribue les valeurs suivantes aux paramètres :

$$\alpha = \frac{1}{2}, T = \frac{1}{6}, m = 10, h = 1, j = \frac{3}{4} \text{ et } k = \frac{1}{2} \text{ et on fait varier le paramètre } \beta \text{ pour repérer trois}$$

types de convergence des solutions vers l'équilibre asymptotique. Quels sont ces trois types de convergence. Illustrer chacun des cas par un jeu de trajectoires et des orbites.

## ▼ Problème E2 (modèle de Tobin)

Le modèle de James Tobin abordé ici est fondamentalement une extension du modèle de croissance néo-classique de Solow au sens où la dynamique monétaire interfère sur la dynamique d'accumulation du capital. Le résultat essentiel, présenté en 1965 dans la revue *Econometrica* sous le titre *Money and Economic Growth*, est que le couplage des marchés des produits et de la monnaie rend la croissance instable sous des hypothèses raisonnables.

La toile de fond est une économie fermée où la population active  $L$  croît au taux "naturel"

$$\text{constant } n : \frac{L'(t)}{L(t)} = n.$$

Sur le marché des produits, l'équilibre se définit par l'égalité de l'offre et de la demande. En notant  $Q$  la valeur ajoutée,  $C$  la consommation et  $I$  l'investissement brut, on a l'égalité usuelle en macroéconomie :  $Q(t) = C(t) + I(t)$ . L'investissement brut est la somme de l'investissement de remplacement  $\delta K(t)$  qui est le produit du stock de capital  $K$  et du taux de dépréciation constant  $\delta$  ( $\delta \geq 0$ ) et de l'investissement net, égal à la variation instantanée du stock de capital productif, soit  $\frac{d}{dt} K(t)$ . L'offre  $Q$  est donnée par la fonction de production à deux facteurs  $F$  supposée

"well behaved" - en particulier, homogène de degré 1 :  $Q(t) = F(K(t), L(t))$ . La consommation des ménages est une fraction constante  $(1 - s)$  de leur budget  $R$ , somme de leurs revenus tirés de l'activité de production qui sont égaux, en économie fermée, à la valeur ajoutée, et de la variation de leurs encaisses réelles  $\frac{d}{dt} \left( \frac{M(t)}{P(t)} \right)$  en notant  $M$  les encaisses nominales et  $P$  le niveau général des prix. Dans ce cadre,

$$R(t) = Q(t) + \frac{d}{dt} \left( \frac{M(t)}{P(t)} \right) = F(K(t), L(t)) + \frac{d}{dt} \left( \frac{M(t)}{P(t)} \right) \text{ et la consommation vaut } C(t) \\ = (1 - s) \left[ F(K(t), L(t)) + \frac{d}{dt} \left( \frac{M(t)}{P(t)} \right) \right], \text{ où le paramètre } s \text{ s'interprète évidemment}$$

comme la propension constante à épargner et vérifie par conséquent  $0 \leq s < 1$ . En rassemblant tous ces éléments, l'équilibre sur le marché des produits s'écrit :

$$F(K(t), L(t)) = (1 - s) \left[ F(K(t), L(t)) + \frac{d}{dt} \left( \frac{M(t)}{P(t)} \right) \right] + \frac{d}{dt} K(t) + \delta K(t) \quad (1)$$

Passons au marché de la monnaie, dont l'équilibre se fait à l'égalité de l'offre de monnaie  $M_s$  et de la demande de monnaie par les agents  $M_d$ . On suppose que les autorités monétaires créent de la monnaie au taux positif constant  $\theta$  :  $\frac{M_s'(t)}{M_s(t)} = \theta$ . La demande de monnaie est conçue comme une demande d'encaisses réelles par travailleur qui s'explique par un motif de transaction et un motif de spéculation. On pose  $\frac{M_d(t)}{P(t) L(t)} = a \frac{K(t)}{L(t)} - b \frac{P'(t)}{P(t)}$  ce qui signifie que toute élévation du revenu d'activité, indirectement mesuré par l'élévation du stock de capital par tête, booste la demande d'encaisses réelles (la sensibilité de la demande d'encaisses au capital par tête  $a$  est donc strictement positive) alors que l'inflation décourage la constitution d'encaisses (le paramètre  $b$  est donc strictement positif). Il y a équilibre économique et équilibre dynamique sur le marché de la monnaie si on a à la fois  $M_s(t) = M_d(t) = M(t)$  et  $\frac{M'(t)}{M(t)} = \theta$ , soit encore :

$$\frac{M(t)}{P(t) L(t)} = a \frac{K(t)}{L(t)} - b \frac{P'(t)}{P(t)} \quad (2)$$

$$\frac{M'(t)}{M(t)} = \theta \quad (3)$$

### Première partie

L'objectif de cette première partie est d'établir un système différentiel à deux variables :  $k(t)$  et  $m(t)$ . La variable  $k$  désigne le capital par tête :  $k(t) = \frac{K(t)}{L(t)}$ . La variable  $m$  désigne les encaisses

réelles par tête :  $m(t) = \frac{M(t)}{P(t) L(t)}$ .

1. On pose  $q = \frac{Q}{L}$ . Donner l'expression de la fonction de production par tête qu'on notera  $f$  à partir de la fonction de production  $F$ .
2. On pose  $p(t) = \frac{P'(t)}{P(t)}$ . Donner une expression simple de  $\frac{m'(t)}{m(t)}$  en fonction  $\theta$ ,  $n$  et  $p(t)$ .
3. Montrer à partir de la relation (2) que  $p$  est une fonction croissante de  $k$  et décroissante de  $m$ .
4. Dédire des questions précédentes les lois d'évolution du capital par tête et des encaisses réelles par tête. On notera les équations différentielles **eqd1** et **eqd2**.

### Deuxième partie

Soit **sysdif** le système différentiel non linéaire formé par **eqd1** et **eqd2**.

1. Montrer que **sysdif** admet trois équilibre dynamique dont un seul a ses composantes strictement positives quand  $n < \theta$ .
2. Montrer que l'équilibre à composantes positives est localement instable et qu'il s'agit d'un équilibre de point-selle.
3. Illustrer les propriétés dynamiques de **sysdif** en construisant le portrait des phases pour les valeurs suivantes des paramètres :

$$s = 0.1, \theta = 0.025, n = 0.005, \delta = 0.1, a = 0.1, b = 0.05$$

et en supposant que la fonction de production est la Cobb-Douglas  $f(k) = k^{\frac{1}{3}}$ .

### Troisième partie

On a critiqué la formulation que donne Tobin sur les comportements monétaires. Reprendre l'étude du modèle en supposant à présent que la demande d'encaisses réelles est une fonction croissante du capital par tête et du taux d'inflation (en période d'inflation, les agents sont enclins à reconstituer leurs encaisses) :  $\frac{Md(t)}{P(t)L(t)} = a \frac{K(t)}{L(t)} + b \frac{P'(t)}{P(t)}$  avec  $a$  et  $b$  strictement positifs.

## ▼ Problème E3 (modèle de la lutte des classes)

Ce problème s'appuie sur le modèle de croissance cyclique que R.M. Goodwin a présenté en 1965 au 1<sup>o</sup> Congrès mondial de la Société d'économétrie puis publié sous le titre "A Growth Cycle" dans l'ouvrage édité par C.H. Feinstein "Socialism, Capitalism and Economic Growth", London, Cambridge University Press, pp. 54-58. L'idée générale est de fusionner dans une même structure formelle le phénomène de croissance capitaliste et celui de cycles économiques, successions de phases d'expansion et de récession dont le théoricien cherche les causes endogènes. L'inspiration est à la fois keynésienne (pas de prise en compte des prix de biens, production à facteurs complémentaires, mode de formation des salaires, chômage durable, ...) et marxiste en un sens restreint. La société est divisée en deux classes antagonistes qui luttent chacune pour obtenir la plus grosse part possible du revenu national dans un environnement conjoncturel mouvant. Il ne s'agit donc pas d'un modèle ouvertement révolutionnaire mais d'une analyse plutôt convaincante des conséquences économiques de la bipartition tranchée du corps social.

On considère un espace économique capitaliste fermé (au sens macroéconomique). Les capitalistes, propriétaires des moyens de production, perçoivent des profits  $P$ . Les salariés ont pour revenu des salaires  $W$ . Le revenu national  $Y$ , égal au produit intérieur brut en économie fermée, est la somme des profits et des salaires :  $Y = P + W$ . Il en découle que la part des salaires dans le revenu national est  $s = \frac{W}{Y}$  et celle des profits  $p = \frac{P}{Y}$ .

Les salaires sont intégralement dépensés en consommation  $C$  :  $W = C$  et les profits en

investissement  $I: P = I$ . L'investissement correspond à la variation instantanée du stock de capital

$$K: \frac{d}{dt} K(t) = I(t).$$

La fonction de production est à facteurs complémentaires avec un trend de croissance constant de la productivité par travailleur salarié. En notant  $L$  la population active employée et  $y = \frac{Y}{L}$  la

productivité moyenne du travail, on a :  $y(t) = y_0 e^{gt}$ , où  $g$  est un taux de croissance positif constant et  $y_0$  un paramètre strictement positif. Conformément aux faits stylisés de la croissance,

la productivité moyenne du capital reste constante :  $\frac{Y(t)}{K(t)} = c$ , avec  $c$  constante réelle positive.

La formation des salaires obéit à une dynamique spécifiant les conditions dans lesquelles se font les négociations salariales entre les partenaires sociaux. La toile de fond est la situation

démographique. On admet que la population active  $L^*$  croît au taux constant  $n (> 0)$  :  $L^*(t) =$

$L_0^* e^{nt}$ . Par définition, le nombre de chômeurs  $U$  est la différence entre la population active et la

population active occupée :  $U = L^* - L$ , de sorte que le taux de chômage vaut  $u = (L^* - L)/L^*$  et le

taux d'emploi  $v = L/L^*$ . Plus le taux de chômage est élevé, moins les salariés sont en situation de

force dans les négociations salariales; inversement, plus le taux d'emploi s'élève, plus les salariés

sont en mesure d'arracher des augmentations de salaire. Cette relation positive entre le taux de

croissance du salaire moyen  $w = \frac{W}{L}$  et le taux d'emploi s'exprime par la relation de Phillips

$$\text{basique : } \frac{\frac{d}{dt} w(t)}{w(t)} = a v(t) - b, \text{ avec } a \text{ et } b \text{ strictement positifs.}$$

1. Dégager du texte de l'énoncé les 16 équations qui constituent la forme structurelle du modèle de Goodwin. Préciser les variables de niveau, les ratios et les paramètres. Expliquer pourquoi les variables et les ratios doivent être datés.

2. Déduire de la forme structurelle un système différentiel non linéaire en  $s$  et  $v$ . On pourra commencer par exprimer les taux de croissance des ratios puis chercher une expression du taux de croissance de  $s$  en fonction de  $v$  et une expression du taux de croissance de  $v$  en fonction de  $s$ .

3. Déterminer les deux équilibres de ce système puis étudier leur nature locale en tenant compte des relations d'inégalité  $g + b < a$  et  $g + n < c$ .

4. Représenter graphiquement la dynamique de ce modèle dans le plan des phases  $vOs$ . On prendra les valeurs suivantes pour les paramètres :  $a = 5, b = 1, g = 1, c = 3, n = 0$ .