TP MAPLE n°3
Equations différentielles

Benjamin Favetto
Vincent Mercat

9 octobre 2008

1 Introduction

Le but de cette série d'exercices est de comprendre ce qu'il est possible (et impossible) de faire avec Maple pour résoudre (tant d'un point de vue formel que numérique) une équation différentielle. On n'hésitera pas à se reporter au cours de mathématiques en cas de besoin.

1.1 Un peu de théorie

On appelera dans la suite problème de Cauchy $\mathcal{C}$ la donnée d'une équation différentielle sous forme résolue et d'une condition initiale en 0 :

$$\left\{ \begin{array}{rcl} y' & = & F(t,y) \\ y(0) & = & y_0 \end{array} \right.$$

1.2 Procédures Maple

On rappelle que le préalable à toute conception de procédure en Maple est une rédaction propre de l'algorithme à mettre en oeuvre sur une feuille. On fera attention aux variables à utiliser à l'intérieur de la procédure (dites variables locales) c'est à dire à leur initialisation et leur mise à jour, au type de boucle le plus approprié, aux valeurs que la procédure doit restituer à l'utilisateur, et bien sûr aux point-virgules ! Un squelette-type de procédure Maple est par exemple :

```
ma_procedure := proc(argument1, ... , argumentk)
local var_locale1 , ... , var_localep;
.
.
(ici le corps de la procédure)
.
.
return resultat;
end proc;
```

Dans la suite, on chargera la bibliothèque de fonctions *DEtools* avec la commande

```
> with(DEtools);
```

2 Résolution formelle

2.1 Un premier exemple

On souhaite résoudre le problème de Cauchy suivant :

$$\left\{ \begin{array}{rcl} y' & = & -y+t \\ y(0) & = & 1 \end{array} \right.$$

Pour cela, on définit l'équation différentielle

```
> eq:=diff(y(t),t) + y(t) = t;
```

et on la fait résoudre par Maple via la commande *dsolve*

```
> dsolve(eq,y(t));
```

Afin de récupérer le résultat dans une expression, il suffit de demander le membre de droite de l'expression puis de le transformer en fonction

```
> h:=rhs(%);
> f:=unapply(h,'t','_C1');
```

Pour prendre en compte la condition initiale, il suffit de la préciser à *dsolve* :

```
> dsolve({eq,y(0)=1}, y(t));
```

Pour observer l'influence des valeurs affectées aux constantes, on pourra utiliser l'instruction *seq* :

```
> plot({seq(f(t,_C1), _C1=-5..5},t=0..1);
```

2.2 Equations différentielles du second ordre

On peut aussi faire résoudre à Maple des équations différentielles du second ordre. Par exemple, on peut s'intéresser à

$$\left\{\begin{array}{rcl} y'' + 5y' + 6y & = & te^{2t} \\ y(0) & = & 0 \\ y'(0) & = & 1 \end{array}\right.$$

Ce qui donne

```
> eq2:=diff(y(t),t$2) + 5*diff(y(t),t) + 6*y(t) = t*exp(2*t);
> dsolve({eq2,y(0)=0,D(y)(0)=1},y(t));
```

Exercice : Tracer les courbes représentatives des solutions des deux précédents problèmes de Cauchy sur $[0, 1]$.

2.3 Recollement et unicité de la solution

A l'aide de la commande *dsolve*, résoudre les équations différentielles suivantes

$$xy' = 1 - (1-x)y \quad y' = y^2 \quad y' = 3|y|^{\frac{2}{3}}$$

en essayant plusieurs conditions initiales et en précisant les intervalles d'étude. Discuter l'unicité de la solution au problème de Cauchy et tracer les solutions obtenues.

Exercice : On considère l'équation différentielle suivante

$$2x(1-x)y' + (1-x)y = 1.$$

1. Résoudre avec Maple cette équation différentielle. Sur quel intervalle se place-t-il par défaut ?
2. Tracer cette première solution. On affectera une valeur arbitraire à la constante.
3. Utiliser l'instruction `assume` pour spécifier l'intervalle d'étude. Résoudre complètement cette équation différentielle.
4. Tracer les solutions sur les différents intervalles de résolution.

3 Résolution numérique

Pour un grand nombre d'équations différentielles issues de la modélisation de phénomènes physiques ou biologiques, la théorie nous apprend qu'une solution existe au problème de Cauchy correspondant, mais la pratique nous enseigne qu'il n'est parfois pas possible d'expliciter cette solution à l'aide de fonctions *élémentaires* (cette notion étant toutefois relativement floue, on retiendra qu'une fonction élémentaire appartient à la famille des fonctions que l'on sait manipuler en classe prépa : exponentielle, fonctions trigonométriques et réciproques, logarithme, puissances, et composées de celles-ci ...).
On peut toutefois s'intéresser aux solutions numériques approchées d'un tel problème, et au tracé des courbes intégrales, qui sont la plupart du temps suffisantes pour donner une idée du comportement de la solution réelle. On prêtera attention à l'erreur commise entre la solution exacte et la solution approchée.

3.1 Avec les outils de Maple

La bibliothèque *DEtools* comprend une fonction *DEplot* dont la syntaxe est la suivante

```
> eq3:=diff(y(t),t$2) + diff(y(t),t) + y(t)^2 = 0;
> DEplot(eq3, y(t), t=0..4*Pi, [[y(0)=0,D(y)(0)= 1], [y(0)=0,D(y)(0)=0.5]], stepsize=0.01);
```

Observer ce que donne l'utilisation de *dsolve* sur `eq3` . Observer aussi l'influence du paramètre `stepsize` dans l'instruction précédente.

Un peu de physique : On souhaite étudier l'équation différentielle qui régit l'évolution temporelle du pendule simple

$$y'' + \sin(y) = 0$$

Commencer par essayer de résoudre formellement cette équation différentielle. Commenter.
On veut ensuite comparer, s'étant donné une condition initiale y_0, la solution de ce problème de Cauchy (qui existe, d'après la théorie), avec la solution du problème linéarisé

$$y'' + y = 0$$

(et la même condition initiale). Pour cela, observer ce que donne

```
> eq:=diff(y(t),t$2) + sin(y(t))= 0;
> eqlin:=diff(y(t),t$2) + y(t)= 0;
> sol:=DEplot(eq,y(t),t=0..2*Pi,[[y(0)=0.1,D(y)(0)=0]],stepsize=0.1);
> sollin:=DEplot(eqlin,y(t),t=0..2*Pi,[[y(0)=0.1,D(y)(0)=0]],stepsize=0.1);
> display({sol,sollin});
```

et reprendre ce qui précède avec $y(0) = \pi/4$ et $y'(0) = 1$.

Remarque : On peut aussi utiliser les fonctions *dsolve* avec l'option `numeric` . Observons tout cela sur l'exemple du modèle proie-prédateur de Volterra : soit S le nombre de sardines et R le nombre de requins dans une zone de l'océan, commenter les equations différentielles suivantes

$$\begin{aligned} \frac{dS}{dt} &= fS(t) - \delta R(t)S(t) \\ \frac{dR}{dt} &= -dR(t) + \kappa R(t)S(t) \end{aligned}$$

Observer la solution numérique donnée par Maple :

```
> eq1:=diff(S(t), t)=f*S(t)-delta*R(t)*S(t);
> eq2:=diff(R(t), t)=-d*R(t)+kappa*R(t)*S(t);
>
> f:=0.2: delta:=0.01: d:=10: kappa:=0.11:
> sol:=dsolve({eq1, eq2, S(0)=100, R(0)=20}, {S(t), R(t)}, type=numeric,
 output=listprocedure):
> s:=subs(sol, S(t)): r:=subs(sol, R(t)):
```

```
>
> f:=0.3:
> sol2:=dsolve({eq1, eq2, S(10)=s(10), R(10)=r(10)}, {S(t), R(t)}, type=numeric,
output=listprocedure):
> s2:=subs(sol2, S(t)): r2:=subs(sol2, R(t)):
>
> plot([r, s], 0..30);
> plot(['r'(t)/'s'(t)], t=0..30);
```

3.2 Ce qui se cache derrière : la méthode d'Euler

Derrière la fonction *DEplot* se cache un algorithme bien connu en analyse numérique : la méthode d'Euler[1].
Considérons le problème de Cauchy $\mathcal{C}$: on se propose de fournir une solution approchée sous la forme d'une fonction affine par morceaux (i.e. dont la courbe représentative sera une ligne polygonale) sur $[0, T]$.
Pour cela, on se donne un entier N et on subdivise l'intervalle en sous-intvervalles de longueur $h = T/N$.
Aux points d'abscisse $\frac{kT}{N}$, la valeur de la solution approchée y_k est donnée par

$$y_{k+1} = y_k + hF(kh, y_k)$$

Commencer par justifier cette approximation sur un dessin (penser à l'approximation d'une intégrale par un rectangle).
Ecrire une procédure $eqdiffeuler(F, N, T, y0)$ prenant en paramètres une fonction F, un entier N, un temps T et une condition initiale y_0 et qui fournit une telle solution approchée. On commencera par

```
> eqdiffeuler:=proc(F,N,T,y0)
local y,k,h,liste;
h:=T/N;
y=evalf(y0);
liste=[[0,y]];
for k from 1 to N do
.
.
end;
RETURN ...;
end proc;
> s:=eqdiffeuler((t,x)->x,100,1,0.5);
```

Il suffit ensuite de demander à Maple de dessiner la solution :

```
> plot(s);
```

On pourra chercher à comparer graphiquement les solutions exactes et approchées pour des équations différentielles que l'on sait résoudre en superposant les deux courbes.

[1]En réalité, Maple utilise une méthode de Runge-Kutta, mais ceci serait trop long à expliquer en une note de bas de page ...