

COLLE MAPLE N°3 Équations Différentielles

Nous allons découvrir aujourd'hui comment Maple résout les équations différentielles et comment se débrouiller quand même Maple ne parvient pas à déterminer une solution explicite : voilà qui devrait éveiller l'attention des brillant(e)s physicien(ne)s que vous êtes.

fonction - expression

Nous aurons besoin de faire la différence entre une fonction et une expression, car Maple la fait...

Pour définir une fonction, il faut la rentrer sous la forme

```
fonction :=variable->expression(variable);
```

Ainsi,

```
> a:=x^2+2;
```

est une expression, alors que

```
> f:=x->x^2+2;
```

est une fonction.

Première différence : `a(32)` ; `f(32)` ;

On peut transformer une expression en fonction à l'aide de

```
unapply(expression, variable)
```

Par exemple, observez

```
> g:=unapply(a,x); g(32);
```

On peut évidemment définir des fonctions de plusieurs variables :

```
> k:=(r,t)->(r*cos(t),r*sin(t)); k(32,Pi/3);
```

diff - D

Nous n'emploierons pas la même syntaxe selon que l'objet à dériver est une fonction ou une expression. Dans le cas d'une expression, on emploie `diff(expression, variable1, variable2, ...)`. Par exemple,

```
> E:=sin(4*x^3+5*x); diff(E,x);
```

On peut dériver plusieurs fois

```
> diff(E,x,x,x); diff(E,x$3);
```

pour dériver une fonction, on utilise l'opérateur `D(fonction)`. Notez bien que `d(fonction)` est une fonction. Par exemple,

```
e:=unapply(E,x); D(e); D(e)(x); D(e)(32); (D@@3)(e); (D@@3)(e)(x);
```

dsolve

Maple sait résoudre pas mal d'équations différentielles grâce à

```
dsolve(equa diff, conditions initiales, fonction(variable)).
```

Par exemple,

```
> dsolve(diff(y(t), t, t) + y(t) = 0, D(y)(0) = 1, y(0) = 1, y(t));
```

est sans surprise. On peut même omettre les conditions initiales

```
> dsolve(diff(y(t), t, t) + y(t) = 0, y(t));
```

Au passage, notons que Maple permet de résoudre des équations différentielles dépendant de paramètres

```
> dsolve(diff(R * diff(q(t), t) + (q(t)) / C = E, q(0) = 0, q(t));
```

Testez par exemple Maple sur les équations suivantes

a) $(x+1)y' - xy = 0$

c) $(1+x^2)y' + xy = 1$

e) $y'\sqrt{1-x^2} - y^2 - 1 = 0$

b) $(1+x^2)y' + xy = 3x^3 + 3x$

d) $xy' + 2y = x/(1+x^2)$

f) $y'' - 3y' + 2y = xe^x + \sin(x)$

plot - odeplot

L'instruction de base pour représenter graphiquement une fonction est

```
plot(expression(variable), variable=mini..maxi, options graphiques);
```

Par exemple

```
> plot(ln(sinh(x)) - x, x = 0..10);
```

C'est parfois un peu farfelu

```
> plot(cos(x) * exp(tan(x)), x = -4..4);
```

Il faut recadrer

```
> plot(cos(x) * exp(tan(x)), x = -4..4, -4..4);
```

C'est déjà mieux. Mais il manque la touche finale

```
> plot(cos(x) * exp(tan(x)), x = -4..4, -4..4, title = 'C'EST BEAU');
```

Ce qui nous intéressera plus aujourd'hui, c'est de tracer des lignes brisées

```
> plot([[0, 0], [1, 3], [2, -4], [3, 2]]);
```

Notez bien l'utilisation des crochets : on a construit une liste de points différente de

```
> plot([[0, 0], [2, -4], [1, 3], [3, 2]]);
```

On peut comparer deux tracés sur un même graphique en utilisant la fonction `display` qu'il faut aller chercher dans la bibliothèque `plots`

```
> with(plots):
> d1 := plot([0, 0], [1, 3], [2, -4], [3, 2], color = blue):
> d2 := plot(sin(2*x), x = 0..3, style = point):
> display(d1, d2);
```

Pour terminer ce petit tour d'horizon, parlons de la fonction `odeplot` qui, elle aussi, fait partie de la bibliothèque `plots`. Il faut d'abord définir une solution d'une équation différentielle, par exemple

```
> s:=dsolve((x+1)*diff(y(x),x)-x*y(x)=0,y(0)=1,y(x)=numeric):
```

L'argument `numeric` ajouté à la fin impose à Maple de résoudre l'équation numériquement. Pour comprendre ce qui se passe, demandez

```
> s(32);
```

Maple renvoie les coordonnées d'un point de la courbe solution. Pour tracer la courbe correspondante, il suffit d'entrer

```
> odeplot(s,[x,y(x)],-0.5..2);
```

Le tracé semble manquer de précision. On peut rajouter comme option `numpoints=1000` sachant que la valeur par défaut est 50.

Vous voici armé(e) pour entamer les hostilités

Résolution d'équations différentielles par discrétisation

Exemple guidé

Soit f une fonction dérivable sur \mathbb{R} vérifiant $f(0) = 1$ et, pour tout x , $f'(x) = f(x)$ (oui, oui, c'est bon, moi je sais aussi qui c'est, on n'a pas besoin de Maple pour ça, mais mieux vaut découvrir sur un exemple simple...).

1. Soit h un réel voisin de 0. Montrez que, pour tout réel a , l'approximation affine donnée par le calcul des dérivées s'écrit

$$f(a+h) \simeq (1+h) \times f(a)$$

2. On prend $h = 1/1000$. On note (a_n) la suite définie par $a_0 = 0$ et $a_{n+1} = a_n + h$. donnez une approximation de $f(a_{n+1})$ en fonction de $f(a_n)$. Déduisez-en que la suite des approximations de $f(a_n)$ est une suite géométrique que vous caractériserez.
3. Faites de même avec $h = -1/1000$.

Voici un petit programme Maple qui construit la courbe approchée par la méthode d'Euler que nous venons d'utiliser.

```
> with(plots);
> Euler:=proc(n,xmin,xmax)      #n, c'est 1/h
> local x,ax,y,ay,k,P,s:
> S:=NULL;      # On construit une suite de points vide au édpert
> y:=1;      # y=f(0)=1 au édpert
> for k from xmin*n to xmax*n do
> ax:=k/n, ay:=y;      # écoordonnes d'un point
> x:=k/n+1/n; y:=(1+1/n)^k;      # écoordonnes du point suivant
> P(k):=[x,y]:      # on édfini le point P(k)
> od:
> plot([S]);
> end:

> Euler(100,-2,2);
```

Sans filet

Le but de notre nouveau jeu va être de discrétiser une équation différentielle du second ordre, de tracer le graphe correspondant et de comparer le résultat obtenu avec les solutions proposées par Maple grâce à `dsolve` et `odeplot`. Nous utiliserons le fait que pour une fonction y deux fois dérivable sur un intervalle I et h proche de 0, on a

$$y(x+h) \simeq y(x) + y'(x) \times h$$

et donc

$$y'(x+h) \approx y'(x) + y''(x) \times h$$

Il s'agit maintenant de mettre au point une procédure `EquaDiff :=proc(f,xmin,xmax,y0,yp0,n)` où $y_0 = y(0)$, $yp_0 = y'(0)$ et f est la fonction de 3 variables définie par

$$y'' = f(y', y, x)$$

Le résultat doit faire apparaître sur le même graphique la courbe obtenue par la méthode d'Euler et celle obtenue à l'aide de `odeplot`.

Il ne restera plus qu'à tester votre procédure sur les nombreuses équations différentielles du second ordre que vous avez résolues en cours avec Madame Gornet.