

Equations différentielles

Méthodes générales

La résolution explicite des équations différentielles

La fonction Maple **dsolve** est l'outil essentiel de résolution des équations différentielles. Elle prend comme premier argument soit une équation différentielle (dans laquelle la dérivée $\frac{dy}{dt}$ de la fonction $y(t)$ par rapport à la variable t est notée $\text{diff}(y(t),t)$, la dérivée seconde étant notée $\text{diff}(y(t),t^2)$, et ainsi de suite), soit un ensemble constitué d'une ou plusieurs équations différentielles et éventuellement de conditions initiales ou aux limites; dans ces conditions initiales ou aux limites, on dénote par $y(a)$ la valeur de y au point a , $D(y)(a)$ celle de $y'(a)$ et plus généralement $(D@@n)(y)(a)$ la dérivée n -ième de y au point a . Le deuxième argument doit être soit la fonction inconnue, soit l'ensemble des fonctions inconnues. Dans la résolution d'équations différentielles, Maple introduit éventuellement des constantes arbitraires qu'il dénote par $_C1, _C2, _C3 \dots$

Un exemple de résolution générale:

```
> dsolve(diff(y(t),t$3)-y(t)=t,y(t));
```

$$y(t) = -t + _C1 e^t + _C2 e^{(-1/2)t} \sin \frac{1}{2} \sqrt{3} t + _C3 e^{(-1/2)t} \cos \frac{1}{2} \sqrt{3} t$$

Avec des conditions initiales:

```
> dsolve({diff(y(t),t$3)-y(t)=t,y(0)=0,D(y)(0)=1,(D@@2)(y)(0)=0},y(t));
```

$$y(t) = -t + \frac{2}{3} e^t - \frac{2}{3} e^{(-1/2)t} \cos \frac{1}{2} \sqrt{3} t + \frac{2}{3} \sqrt{3} e^{(-1/2)t} \sin \frac{1}{2} \sqrt{3} t$$

Avec des conditions aux limites:

```
> dsolve({diff(y(t),t$2)-y(t)=cos(t),y(0)=0,y(Pi)=1},y(t));
```

$$y(t) = -\frac{1}{2} \frac{e^t \cos(t) - \frac{(-1+e)(e^t)^2}{e^{(2)}-1} - \frac{-e+e^{(2)}}{e^{(2)}-1}}{e^t}$$

Pour un système d'équations différentielles, la méthode est similaire:

```
> dsolve({diff(x(t),t)=x(t)+y(t),diff(y(t),t$2)=x(t)-diff(x(t),t)-y(t)},{x(t),y(t)});
```

$$\{y(t) = \cos(\sqrt{2} t) _C2 + \frac{1}{2} \sqrt{2} \sin(\sqrt{2} t) _C3, x(t) = e^t _C1 + \frac{1}{3} \sqrt{2} \sin(\sqrt{2} t) _C2 - \frac{1}{3} \cos(\sqrt{2} t) _C2 + \frac{1}{3} _C2 e^t - \frac{1}{3} \cos(\sqrt{2} t) _C3 - \frac{1}{6} \sqrt{2} \sin(\sqrt{2} t) _C3 + \frac{1}{3} _C3 e^t\}$$

et avec des conditions initiales:

```
> dsolve({diff(x(t),t)=x(t)+y(t),diff(y(t),t$2)=x(t)-diff(x(t),t)-y(t),x(0)=0,y(0)=1,D(y)(0)=0},{x(t),y(t)});
```

$$\{x(t) = \frac{1}{3} \sqrt{2} \sin(\sqrt{2} t) - \frac{1}{3} \cos(\sqrt{2} t) + \frac{1}{3} e^t, y(t) = \cos(\sqrt{2} t)\}$$

Récupération des résultats de la fonction dsolve

Nous venons de voir dans les exemples précédents que les résultats de la fonction **dsolve** étaient retournés (dans les bons cas) sous forme soit d'une égalité entre la fonction inconnue et son expression, soit d'un ensemble d'égalités entre les fonctions inconnues et leurs expressions (parfois aussi sous forme implicite comme nous le verrons sur certains exemples, auquel cas il n'y a malheureusement pas grand chose d'autre à faire que d'essayer la fonction **solve** avec un très maigre espoir de succès). En général, nous serons amenés à essayer de récupérer, en vue de manipulations extérieures, l'expression trouvée. Cela peut se faire à la main, à l'aide la fonction **op** qui retourne les différents opérandes de

l'expression:

```
> solution:=dsolve({diff(x(t),t)=x(t)+y(t),diff(y(t),t)=x(t)-diff(x(t),t)-y(t),x(0)=0,y(0)=1,D(y)(0)=0},{x(t),y(t)});
```

$$\text{solution} := \{x(t) = \frac{1}{3}\sqrt{2} \sin(\sqrt{2} t) - \frac{1}{3} \cos(\sqrt{2} t) + \frac{1}{3} e^t, y(t) = \cos(\sqrt{2} t)\}$$

```
> op([1,2],solution);
```

$$\frac{1}{3}\sqrt{2} \sin(\sqrt{2} t) - \frac{1}{3} \cos(\sqrt{2} t) + \frac{1}{3} e^t$$

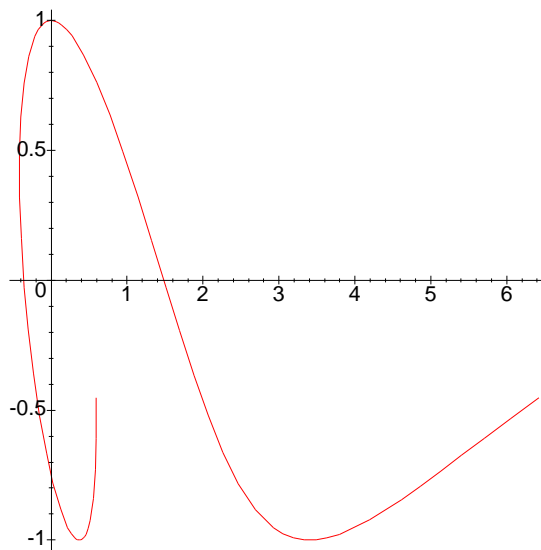
Outre le fait que cette méthode est peu commode, elle présente le gros inconvénient lorsqu'un ensemble de solutions est retourné, que l'ordre des solutions dans l'ensemble soit modifié par Maple en cours de calcul (ce qui est toujours possible, l'ordre dans un ensemble n'étant pas spécifié) ce qui peut conduire à l'intervention des solutions trouvées, ce qui est catastrophique. La bonne méthode est d'utiliser la solution renvoyée par Maple comme règle de substitution dans la fonction **subs**. Cette fonction utilise en fait une égalité ou un ensemble d'égalités pour réaliser des substitutions dans une expression. Or, il est bien évident que remplacer $x(t)$ par t^2+2 dans l'expression $x(t)$ va renvoyer t^2+2 . Autrement dit, le bon moyen de récupérer les expressions des solutions dans l'exemple précédent est le suivant

```
> x:=subs(solution,x(t)); y:=subs(solution,y(t));
```

$$x := \frac{1}{3}\sqrt{2} \sin(\sqrt{2} t) - \frac{1}{3} \cos(\sqrt{2} t) + \frac{1}{3} e^t$$
$$y := \cos(\sqrt{2} t)$$

ce qui permet ensuite un tracé de la solution

```
> plot([x,y,t=-3..3]);
```



■ Résolution numérique des équations différentielles

Très souvent, même dans les problèmes avec condition initiale, Maple ne sera pas capable de trouver une solution explicite, la plupart du temps parce qu'il n'en existe pas. On peut lui demander, à condition qu'il ne subsiste aucun paramètre formel dans l'équation et que les conditions initiales soient entièrement spécifiées sous forme numérique, de construire des fonctions numériques qui approximent les solutions de l'équation, par une méthode à la Runge-Kutta. Il suffit pour cela d'ajouter le paramètre **numeric** dans la fonction **dsolve**.

```
> f:=dsolve({diff(y(t),t)=y(t)^2+t,y(0)=1},y(t),numeric);
```

```
f:=proc(rkf45_x) ... end
```

```
> f(0.5);
```

[$t = .5$, $y(t) = 2.234532993944992$]

La fonction numérique construite par Maple, et dont le code résumé est ce poétique `proc (rkf45_x) ... end` est une fonction qui prend en argument la valeur du paramètre réel (ici appelé `rkf45_x`, le `rk` étant là pour Runge-Kutta) et qui retourne une liste d'égalités dont la première est `nom_du_paramètre=valeur`, les suivantes étant du type `fonction_inconnue=valeur`. Le résultat est un peu difficile à exploiter, sauf pour des fonctions spécialisées de Maple. On préférera utiliser une option supplémentaire de la fonction `dsolve` de manière à retourner non pas une fonction à valeur liste, mais une liste de fonctions à valeurs réelles.

```
> solution:=dsolve({diff(y(t),t)=y(t)^2+t,y(0)=1},y(t),numeric,output=listprocedure);
```

```
      solution := [ t =(proc(t) ... end), y(t) =(proc(t) ... end) ]
```

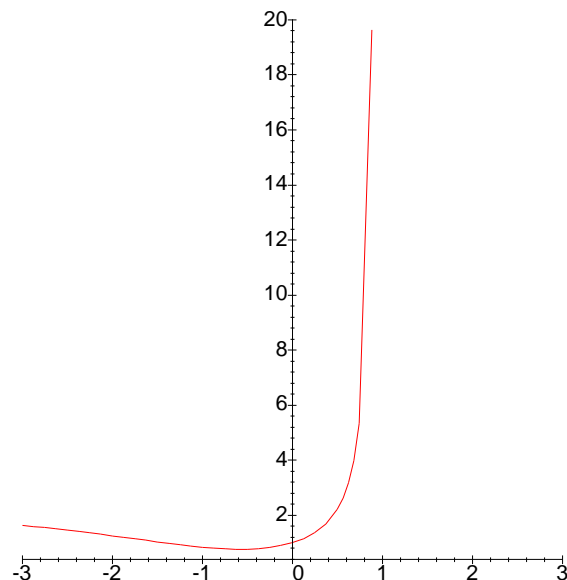
Ici Maple a construit deux fonctions, la fonction $t \rightarrow t$ (c'est évidemment l'identité) et la fonction $t \rightarrow y(t)$ qui nous intéresse. On récupérera sa valeur facilement à l'aide de la fonction `subs`, comme on l'a vu dans la section précédente:

```
> f:=subs(solution,y(t));
```

 Attention: $f := \dots$ et non $f(t) := \dots$

```
      f := proc(t) ... end
```

```
> plot(f,-3..3);
```



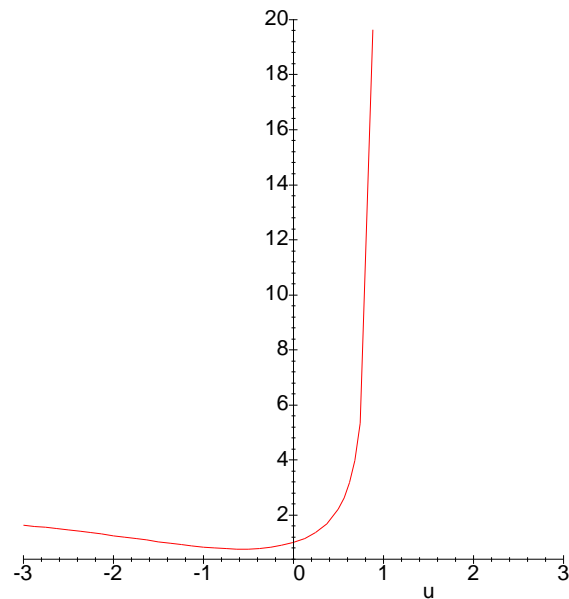
Remarque: `plot(f(u),u=-3..3)` ne marche pas. En effet, Maple cherche d'abord dans ce cas à calculer $f(u)$ pour une valeur formelle du paramètre u . Or la fonction f n'accepte que les paramètres numériques, car elle a besoin d'effectuer des comparaisons:

```
> plot(f(u),u=-3..3);
```

 Error, (in f) cannot evaluate boolean

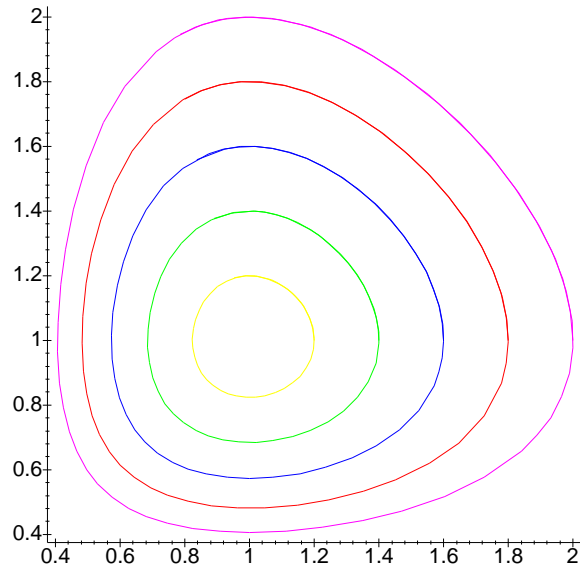
On peut cependant utiliser cette syntaxe en bloquant l'évaluation de $f(x)$ par des apostrophes

```
> plot('f(u)',u=-3..3);
```



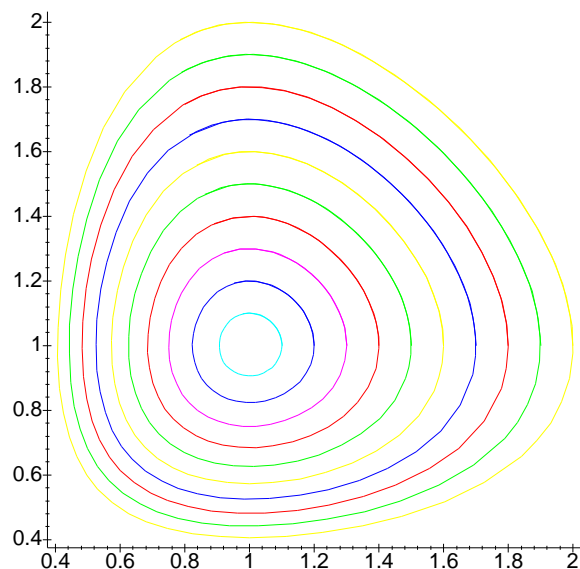
Voici un exemple plus complexe avec un système dynamique classique et une famille de courbes intégrales

```
> x:='x': y:='y':
equations:=diff(x(t),t)=x(t)*(1-y(t)),
diff(y(t),t)=y(t)*(x(t)-1),y(0)=1:
fonctions:={x(t),y(t)}:
sol1:=subs(dsolve({equations,x(0)=1.2},fonctions,numeric,output=listprocedure),
[x(t),y(t),0..8]):
sol2:=subs(dsolve({equations,x(0)=1.4},fonctions,numeric,output=listprocedure),
[x(t),y(t),0..8]):
sol3:=subs(dsolve({equations,x(0)=1.6},fonctions,numeric,output=listprocedure),
[x(t),y(t),0..8]):
sol4:=subs(dsolve({equations,x(0)=1.8},fonctions,numeric,output=listprocedure),
[x(t),y(t),0..8]):
sol5:=subs(dsolve({equations,x(0)=2},fonctions,numeric,output=listprocedure),
[x(t),y(t),0..8]):
plot({sol1,sol2,sol3,sol4,sol5});
```



Bien entendu le procédé peut être automatisé

```
> plot({seq(subs(dsolve({equations,x(0)=1.+i*0.1},fonctions,
numeric,output=listprocedure),[x(t),y(t),0..8]),i=1..10)})
;
```



La bibliothèque DEtools

La bibliothèque DEtools contient quelques fonctions utilitaires sur les équations différentielles.

```
> with(DEtools);
[DENormal, DEplot, DEplot3d, Dchangevar, PDEchangecoords, PDEplot,
autonomous, convertAlg, convertsys, dfieldplot, indicialeq, phaseportrait,
reduceOrder, regularsp, translate, untranslate, varparam]
```

Le rôle de la fonction **Dchangevar** est d'effectuer des changements de variables ou de fonctions inconnues dans les équations différentielles. Voici un exemple de changement de variable $x = \exp(t)$ dans une équation d'Euler, la fonction $y(x)$ étant notée $z(t)$.

```
> Dchangevar(x=exp(t),y(x)=z(t),x^2*diff(y(x),x$2)+y(x)=x,x,
```

`t);`

$$\frac{1}{t^2} z(t)^2 + z(t) = e^t$$

et un exemple plus général de changement de variable $x = \phi(t)$ dans une équation linéaire d'ordre 2

> `Dchangevar(x=phi(t),y(x)=z(t),diff(y(x),x$2)+a(x)*diff(y(x),x)+b(x)*y(x)=0,x,t);`

$$\frac{1}{\phi'(t)^2} z(t)^2 + \frac{a(\phi(t))}{\phi'(t)} z(t) + b(\phi(t)) z(t) = 0$$

La fonction **convertsys** se charge de réduire une ou plusieurs équations différentielles en un système d'ordre 1.

> `convertsys(diff(y(x),x$2)+cos(y(x))=sin(omega*x),{y(0)=Pi/2,D(y)(0)=0},y(x),x,z,zp);`

$$[z_{p1} = z_2, z_{p2} = -\cos(z_1) + \sin(\omega x)], \quad z_1 = y(x), z_2 = \frac{1}{\omega} y'(x), 0, \frac{1}{2}, 0$$

Nous laissons au lecteur le soin de découvrir les autres fonctions de cette bibliothèque (réduction de l'ordre par **reduceOrder** et méthode de variation des constantes par **varparam**)

> `reduceOrder((x^2-1)*diff(y(x),x$3)+diff(y(x),x)-x^2*y(x)=0,y(x),exp(x));`

$$(x^2 - 1) \frac{1}{x^2} y(x)^2 + (3x^2 - 3) \frac{1}{x} y(x) + (-2 + 3x^2) y(x)$$

> `varparam([sin(x),cos(x)],f(x),x);`

$$-C_1 \sin(x) + -C_2 \cos(x) + \cos(x) f(x) dx \sin(x) + -\sin(x) f(x) dx \cos(x)$$

Equations différentielles d'ordre 1

Equations linéaires d'ordre 1

La fonction **dsolve** permet de résoudre les équations différentielles linéaires d'ordre 1 par la méthode classique (résolution de l'équation homogène par une dérivée logarithmique puis méthode de variation des constantes).

> `soll:=subs(dsolve({equations,x(0)=1.3},fonctions,numeric,output=listprocedure),`

`[x(t),y(t),0..4]);dsolve((t^2+1)*diff(y(t),t)+t*y(t)=1,y(t));`

$$y(t) = \frac{\operatorname{arcsinh}(t) + C1}{\sqrt{t^2 + 1}}$$

On peut comme d'habitude introduire des conditions initiales qui font alors partie des équations à résoudre.

> `dsolve((t^2+1)*diff(y(t),t)+t*y(t)=1,y(1)=1},y(t));`

$$y(t) = \frac{\operatorname{arcsinh}(t) - \frac{1}{2}(\sqrt{2} \ln(\sqrt{2} + 1) - 2)\sqrt{2}}{\sqrt{t^2 + 1}}$$

Attention au fait que la solution trouvée par Maple peut avoir un domaine de validité restreint (ici les calculs ne sont valables qu'en dehors de l'intervalle [-1,1])

> `dsolve((t^2-1)*diff(y(t),t)+t*y(t)=1,y(t));`

$$y(t) = \frac{\ln(t + \sqrt{(-1+t)(1+t)}) + _CI}{\sqrt{-1+t} \sqrt{1+t}}$$

et aboutir même à des résultats délicats à interpréter, ou même franchement contestables:

> **dsolve**({(t^2-1)*diff(y(t),t)+t*y(t)=1,y(0)=1},y(t));

$$y(t) = \frac{\ln(t + \sqrt{(-1+t)(1+t)}) - \frac{1}{2} I(-2)}{\sqrt{-1+t} \sqrt{1+t}}$$

et rien n'y fait

> **assume**(-1<t,t<1): **simplify**(");

$$y(t) = -\frac{1}{2} \frac{-2 \ln(t + \sqrt{(1-t)(1+t)}) + I(-2)}{\sqrt{-1+t} \sqrt{1+t}}$$

Maple peut également résoudre les équations de manière formelle

> **dsolve**(diff(y(t),t)+a*y(t)=f(t),y(t));

$$y(t) = e^{(-at)} \int e^{(at)} f(t) dt + _CI$$

et même plus généralement

> **dsolve**(diff(y(t),t)+a(t)*y(t)=b(t),y(t));

$$y(t) = e^{(-\int a(t) dt)} \int e^{(\int a(t) dt)} b(t) dt + _CI$$

Equations classiques d'ordre 1

Maple sait résoudre (au moins de manière implicite) les équations classiques d'ordre 1, qu'elles soient des équations aux différentielles totales

> **dsolve**((2*t+y(t))+(t+2*y(t))*diff(y(t),t)=0,y(t));

$$t^2 + t y(t) + y(t)^2 = _CI$$

à variables séparables

> **dsolve**(cos(t)=y(t)*diff(y(t),t),y(t));

$$y(t)^2 = 2 \sin(t) + _CI$$

incomplètes

> **dsolve**(diff(y(t),t)=sin(y(t)),y(t));

$$-\ln(\csc(y(t)) - \cot(y(t))) + t = _CI$$

> **convert**("",sincos);

$$-\ln \frac{1}{\sin(y(t))} - \frac{\cos(y(t))}{\sin(y(t))} + t = _CI$$

Remarquez que les méthodes mécaniques de Maple ont perdu au passage un certain nombre de solutions constantes évidentes $y(t) = k$

Les équations homogènes font aussi partie des équations résolues par Maple sous forme implicite

> **dsolve**(diff(y(t),t)=sin(y(t)/t),y(t));

$$t = _CI e^{\int \frac{y(t)}{t} dt - \int \frac{1}{u \ln(u)} du}$$

Maple ayant perdu au passage les solutions évidentes $y(t) = k t$. Voici un autre exemple d'équation classique: une équation de Clairaut, qui admet comme solution une famille de droites et l'enveloppe de cette famille

> **dsolve**(y(t)=diff(y(t),t)*t-(diff(y(t),t))^2,y(t));

$$y(t) = -_CI^2 + _CI t, y(t) = \frac{1}{4} t^2$$

>

Equations différentielles d'ordre 2

Résolution formelle

Maple résoud formellement (mais parfois de manière implicite) un certain nombre d'équations différentielles d'ordre 2 comme les équations différentielles linéaires à coefficients constants

```
> dsolve({diff(y(x),x$2)+y(x)=cos(omega*x),y(0)=0,D(y)(0)=0},y(x));
```

$$y(x) = \frac{1}{2} \sin(x) \sin((-1 +)x) + \sin(x) \sin((-1 +)x) - \sin(x) \sin((1 +)x) \\ + \sin(x) \sin((1 +)x) - \cos(x) \cos((1 +)x) + \cos(x) \cos((1 +)x) \\ - \cos(x) \cos((-1 +)x) - \cos(x) \cos((-1 +)x) - 2 \frac{\cos(x)}{-1 + }^2 + 2 \frac{\cos(x)}{-1 + }^2 \\ / (-1 +)^2$$

```
> expand(");
```

$$y(x) = -\frac{\sin(x)^2 \cos(x)}{-1 + }^2 - \frac{\cos(x)^2 \cos(x)}{-1 + }^2 - \frac{\cos(x)}{(-1 +)^2} + \frac{\cos(x)}{(-1 +)^2}$$

ou les équations d'Euler

```
> dsolve(t^2*diff(y(t),t$2)+2*t*diff(y(t),t)+y(t)=0,y(t));
```

$$y(t) = \frac{-C1 \cos \frac{1}{2} \sqrt{3} \ln(t)}{\sqrt{t}} + \frac{-C2 \sin \frac{1}{2} \sqrt{3} \ln(t)}{\sqrt{t}}$$

Il résoud de manière implicite les équations autonomes du type de celles que l'on peut rencontrer en mécanique sans frottement (équations conservatives), à condition qu'elles ne contiennent pas de conditions initiales:

```
> dsolve(diff(y(x),x$2)+sin(y(x))=0,y(x));
```

$$x = \int_0^{\frac{1}{\sqrt{2 \cos(y1) + C1}}} dy1 - C2, x = \int_0^{\frac{1}{\sqrt{2 \cos(y2) + C1}}} dy2 - C2$$

```
> dsolve({diff(y(x),x$2)+sin(y(x))=0,y(0)=Pi/2},y(x));
```

Il résoud même des équations autonomes plus complexes:

```
> dsolve(diff(y(x),x$2)*y(x)-diff(y(x),x)^2=1,y(x));
```

$$x = \frac{\ln(\sqrt{-C1} y(x) + \sqrt{-1 + y(x)^2 - C1})}{\sqrt{-C1}} - C2,$$

$$x = -\frac{\ln(\sqrt{-C1} y(x) + \sqrt{-1 + y(x)^2 - C1})}{\sqrt{-C1}} - C2$$

```
> dsolve({diff(y(x),x$2)*y(x)-diff(y(x),x)^2=1,D(y)(0)=0},y(x));
```

$$y(x) = \frac{1}{2} \frac{1 + (e^{(-x\sqrt{-C1})})^2}{e^{(-x\sqrt{-C1})} \sqrt{-C1}}, y(x) = \frac{1}{2} \frac{1 + (e^{(x\sqrt{-C1})})^2}{e^{(x\sqrt{-C1})} \sqrt{-C1}}$$

Résolution par séries entières

L'option **series** de la fonction dsolve permet d'obtenir, dans le cas où une solution explicite ne pourrait pas être trouvée (ou dans tout autre cas où cela pourrait être intéressant) les premiers termes du développement en séries entières d'une solution au voisinage de 0.

```
> dsolve({(1-t^2)*diff(y(t),t$2)+t*diff(y(t),t)+(t^2-m^2)*y(t)=0,y(0)=a,D(y)(0)=b},y(t),series);
```


$$y(t) = a + b t + \frac{1}{2} a m^2 t^2 + \frac{1}{6} b m^2 - \frac{1}{6} b t^3 + \frac{1}{24} m^4 a - \frac{1}{12} a t^4 + \frac{1}{120} m^4 b + \frac{1}{60} b m^2 - \frac{3}{40} b t^5 + O(t^6)$$

On peut jouer sur l'ordre du développement en affectant la variable Order (qui par défaut vaut 6)

```
> Order:=12;
dsolve({(1-t^2)*diff(y(t),t$2)+t*diff(y(t),t)+(t^2-m^2)*y(
t)=0,y(0)=0,D(y)(0)=b},y(t),series);
Order:=12
```

$$y(t) = b t + \frac{1}{6} b m^2 - \frac{1}{6} b t^3 + \frac{1}{120} m^4 b + \frac{1}{60} b m^2 - \frac{3}{40} b t^5 + \frac{1}{5040} m^6 b + \frac{17}{5040} m^4 b + \frac{1}{5040} b m^2 - \frac{23}{1008} b t^7 + \frac{13}{90720} m^6 b + \frac{277}{181440} m^4 b - \frac{41}{90720} b m^2 - \frac{521}{51840} b + \frac{1}{362880} m^8 b t^9 + \frac{23}{7983360} m^8 b + \frac{1879}{19958400} m^6 b + \frac{16757}{19958400} m^4 b - \frac{14051}{39916800} b m^2 + \frac{1}{39916800} m^{10} b - \frac{8203}{1478400} b t^{11} + O(t^{12})$$

>

Résolution numérique

Dans le cas de la résolution numérique d'une équation différentielle d'ordre 2, Maple commence par la transformer en un système d'ordre 1 qu'il résoud ensuite par une méthode à la Runge-Kutta, si bien qu'il considère comme fonctions inconnues aussi bien la fonction qu'on lui a fourni, que sa dérivée. Les fonctions numériques retournées seront donc $t \rightarrow t$, $t \rightarrow y(t)$ et $t \rightarrow y'(t)$ si l'on choisit l'option `output=listprocedure` (voir plus haut)

```
> solution:=dsolve({diff(y(t),t$2)+sin(y(t))=0,y(0)=Pi/2,D(y)
(0)=0},y(t),numeric, output=listprocedure);
solution :=
```

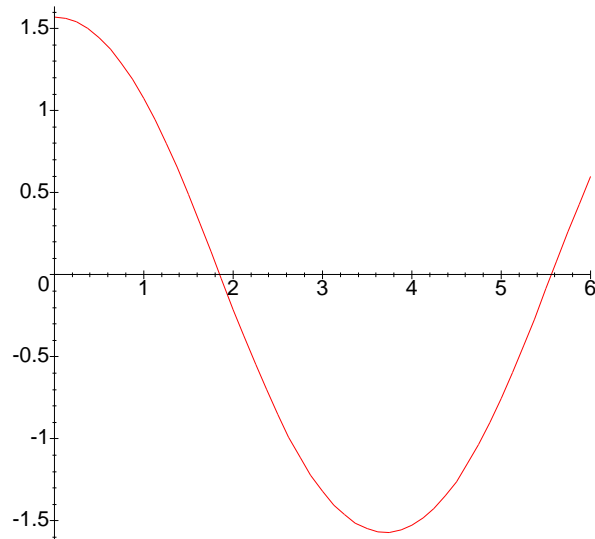
$$t = (\text{proc}(t) \dots \text{end}), y(t) = (\text{proc}(t) \dots \text{end}), \frac{dy}{dt} y(t) = (\text{proc}(t) \dots \text{end})$$

On peut récupérer les fonctions $f : t \rightarrow y(t)$ et $df : t \rightarrow y'(t)$ de la manière usuelle

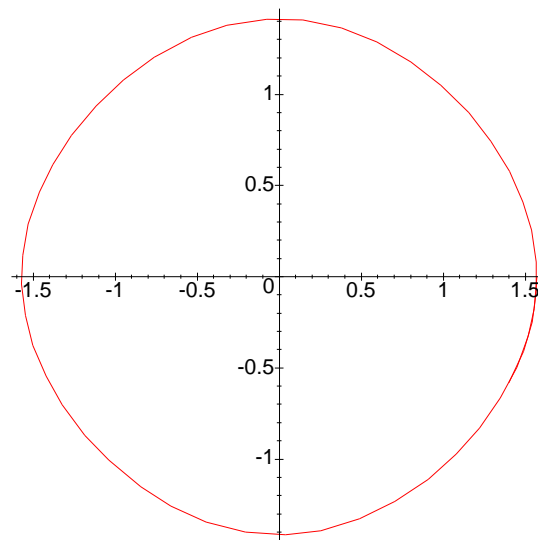
```
> f:=subs(solution,y(t)); df:=subs(solution,diff(y(t),t));
f:=proc(t) ... end
df:=proc(t) ... end
```

ce qui permet ensuite divers tracés

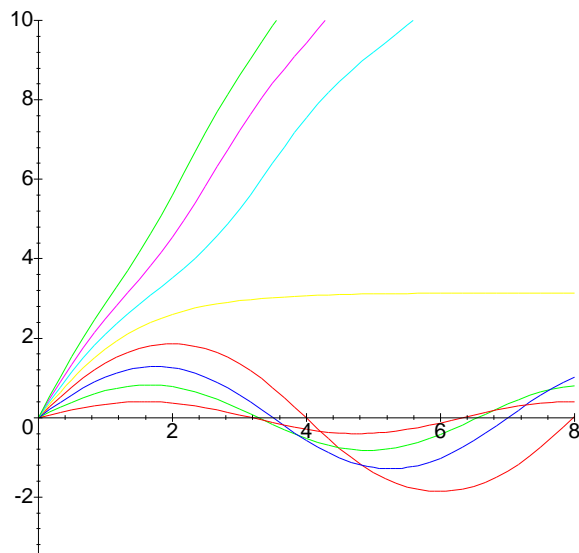
```
> plot({f,fd},0..6);
```



[ou dans l'espace des phases (on trace l'arc paramétré $t \rightarrow (y(t), y'(t))$)
 > `plot([f,df,0..8]);`

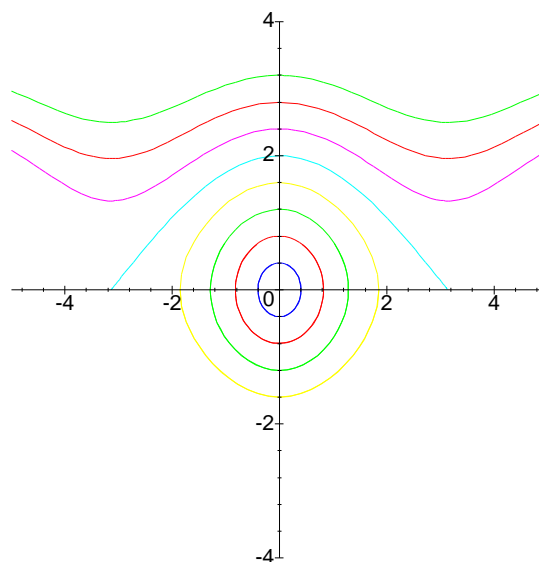


[Comme friandise, voici quelques tracés de solutions pour l'équation du pendule $y'' + \sin(y) = 0$, avec la condition initiale $y(0)=0$ et correspondant à diverses vitesses initiales
 > `plot({seq(subs(dsolve({diff(y(t),t$2)+sin(y(t))=0,y(0)=0,D(y)(0)=0.4*i},y(t),numeric,output=listprocedure),y(t)),i=1..8)},0..8,view=[0..8,-3.5..10]);`



et un tracé similaire dans l'espace des phases

```
> plot(
    {seq(
        subs(
            dsolve({diff(y(t),t$2)+sin(y(t))=0,
                    y(0)=0,D(y)(0)=0.4*i},
                    y(t),numeric, output=listprocedure),
            [y(t),diff(y(t),t)],-6..6]
        ),
        i=1..8
    )},
    view=[-5..5,-4..4]
);
```



>

Systèmes différentiels

└ Toutes les techniques précédentes s'appliquent de manière similaire aux systèmes différentiels,

qu'ils soient d'ordre 1

```
> dsolve({diff(x(t),t)=x(t)-y(t)+exp(t),diff(y(t),t)=2*x(t)+y(t)-t},
          {x(t),y(t)});
```

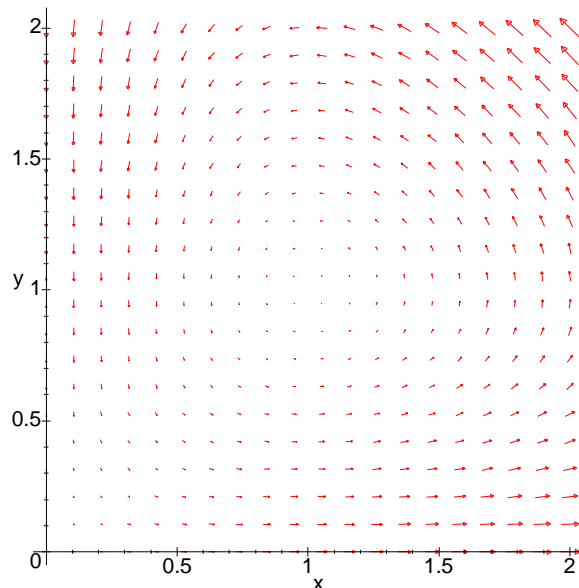
ou d'ordre supérieur (ici 2)

```
> dsolve({diff(x(t),t$2)=2*x(t)-3*y(t),diff(y(t),t$2)=x(t)-2*y(t)},
          {x(t),y(t)});
```

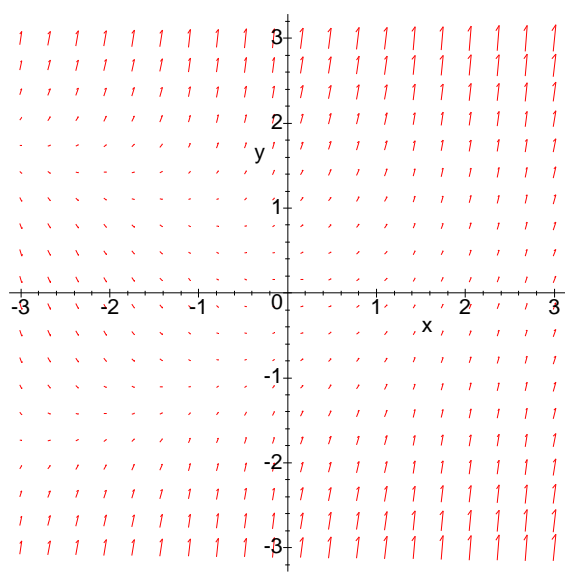
$$\begin{aligned} \{x(t) = & -\frac{1}{2}C1 \cos(t) + \frac{3}{4}C1 e^{(-t)} + \frac{3}{4}C1 e^t - \frac{3}{4}C2 e^{(-t)} + \frac{3}{4}C2 e^t - \frac{1}{2}C2 \sin(t) \\ & + \frac{3}{2}C3 \cos(t) - \frac{3}{4}C3 e^{(-t)} - \frac{3}{4}C3 e^t + \frac{3}{2}C4 \sin(t) - \frac{3}{4}C4 e^t + \frac{3}{4}C4 e^{(-t)}, y(t) = \\ & -\frac{1}{2}C1 \cos(t) + \frac{1}{4}C1 e^{(-t)} + \frac{1}{4}C1 e^t - \frac{1}{2}C2 \sin(t) + \frac{1}{4}C2 e^t - \frac{1}{4}C2 e^{(-t)} \\ & + \frac{3}{2}C3 \cos(t) - \frac{1}{4}C3 e^{(-t)} - \frac{1}{4}C3 e^t + \frac{1}{4}C4 e^{(-t)} - \frac{1}{4}C4 e^t + \frac{3}{2}C4 \sin(t)\} \end{aligned}$$

En complément, pour les systèmes autonomes et les champs de vecteurs, la fonction **fieldplot** de la bibliothèque **plots** permettra par un tracé du champ de vecteurs, d'avoir une bonne idée des solutions:

```
> plots[fieldplot]([x*(1-y),y*(x-1)],x=0..2,y=0..2,arrows=SLIM,
                    color=red);
```



```
> plots[fieldplot]([1,x+y^2],x=-3..3,y=-3..3,color=red);
```



Cours de mathématiques

par Denis Monasse

Ed. Vuibert

Table des matières

- Plan général
- Algèbre générale
- Algèbre linéaire
- Réduction des endomorphismes
- Topologie des espaces métriques
- Espaces vectoriels normés
- Comparaison des fonctions
- Suites et séries numériques
- Fonctions d'une variable réelle
- Intégration
- Suites et séries de fonctions

- Séries entières
- Formes quadratiques
- Formes hermitiennes
- Séries de Fourier
- Calcul différentiel
- Equations différentielles
- Espaces affines
- Courbes
- Surfaces
- Intégrales multiples
- Index