Chapitre S2 : Equations différentielles avec Scilab

1 Les commandes de calcul d'intégrales (boîte noire ou grise)

Au T.P. 6. nous avons vu deux algorithmes de calcul numérique d'intégrales : méthode des rectangles et méthode des trapèzes. Nous reviendrons sur des algorithmes plus performants. Ici, il s'agit seulement de donner les fonctions SciLab faisant des calculs numériques d'intégrales.

1.1 Intégration d'une expression (boîte noire pour nous)

Disons qu'on veut une valeur approchée de $2\int_{-1}^{1}\sqrt{1-t^2}dt$.

```
integrate('2*sqrt(1-t^2)','t',-1,1)
```

N.B. Notez que l'expression est donnée par une chaîne de caractères (entre *quotes*) et la variable d'intégration *aussi.*

1.2 Intégration d'un tableau de valeurs

Parfois on ne connaît pas de formule pour une fonction (qui correspond p. ex. à un relevé expérimental) on a juste ses valeurs pour différentes valeurs de la variable. On peut alors considérer l'interpolation affine par morceaux de cette fonction et ensuite sommer des trapèzes. Pour tricher voyons ce que cela donne pour (suffisamment de) valeurs de la fonction précédente !

```
x=-1:0.01:1
y=2*sqrt(1-x^2)
inttrap(x,y)
```

C'est moins bon, même si on remplace par le pas par 0.001, donc la technique d'intégration numérique de SciLab est plus efficace.... il y a en effet des algo. plus efficace que la méthodes des trapèzes !

2 Résolutions d'E.D. d'ordre 1

2.1 Point de terminologie et de forme

Terminologie : Une équation différentielle comme nous les connaissons est une *Equation Différentielle Ordinaire*, en anglais *Ordinary Differential Equation* donc `ode`.

Par opposition, pour les fonctions à plusieurs variables les équations avec des dérivées partielles différentes seront appelées *Equations aux Dérivées partielles* (E.D.P. et en anglais `PDE`).

Forme : Une équation différentielle du premier ordre normalisée peut toujours s'écrire sous la forme :

$$y' = f(x,y) \text{ c'est-à-dire aussi } y'(x) = f(x,y(x))$$

Par exemple $y'(x) = 2x\sin(y(x)) + 1$ s'écrira $y' = f(x,y)$ avec $f(u,v) = 2u\sin(v) + 1$.

Parfois la fonction f ne dépend que de y, par exemple $y'(x) = 2.y(x)^2$. On dit que l'équation est *autonome* : ces équations ont des propriétés particulières. Mais :

Contrainte en SciLab :

> Pour définir une E.D. en SciLab la fonction f doit toujours être déclarée avec les *deux* variables `x` et `y` (*dans cet ordre : variable puis fonction*) même si elle ne dépend que de `y`.

Par exemple, si on veut résoudre $y'(x) = 2x\sin(y(x)) + 1$, on déclarera :

```
 function yprime=f(x,y)
    yprime=2*x*sin(y)+1 // le nom yprime  n'est qu'un choix de nom de variable !
endfunction
```

Et si on veut résoudre $y'(x) = 2.y(x)^2$:

```
function yprime=g(x,y)
    yprime=2*y^2 // ici x n'apparaît pas mais il doit être mis comme argument de g
endfunction
```

2.2 L'appel de la commande `ode` de SciLab

Motivation : la résolution numérique d'un problème de Cauchy

Autrement dit on se donne $(x_0, y_0) \in \mathbb{R}^2$ (pour l'instant) et l'E.D. $y'(x) = f(x,y)$ et on cherche « la » solution du problème de Cauchy associé par un algorithme d'approximation numérique.

Nous avons vu la méthode d'Euler qui est un algorithme très élémentaire. La fonction `ode` de SciLab utilise des méthodes plus sophistiquées, dont nous reparlerons...

Pour cela on doit se donner aussi l'intervalle des `x` sur lequel on veut calculer des valeurs de la solution approchée. D'où la nécessité de préciser :

Ce qu'on doit mettre en argument de `ode`

L'appel de la fonction `ode` nécessite l'entrée dans l'ordre de quatre arguments :

a) La valeur initiale `y0`,
b) L'abscisse initiale `x0`,
c) Les valeurs de `x` pour lesquels on veut calculer la solution approchée,
d) La fonction `f` définie suivant les recommandations du § 2.1.

Avec tout cela :

```
y=ode(y0,x0,x,f)
```

affecte à `y` la liste des valeurs de la fonction solution approchée de l'équation différentielle pour les valeurs de x données.

Exemple très simple

```
y0=1;
x0=0;
x=0:0.1:10
deff('yprime=f(x,y)','yprime=-y')
y=ode(y0,x0,x,f)
clf()
plot2d(x,y)
```

Si on trace la solution théorique via `z=exp(-x)` la superposition sera parfaite...

3 Comment transformer une E.D. d'ordre 2 en E.D. vectorielle d'ordre 1

3.1 La méthode illustrée sur une E.D. Linéaire

Motivation :

On se donne une EDL du second ordre $y''(x) = a(x)y'(x) + b(x)y(x) + c(x)$. On va la ramener à une E.D. du premier ordre mais à inconnue une *fonction vectorielle*.

Intérêt ici : On pourra lui appliquer alors la commande `ode` qui, suivant la philosophie de SciLab s'applique aussi à des fonctions *vectorielles.*

Intérêt mathématique plus général : *On peut ramener toute la théorie des EDL d'ordre quelconque à la théorie des E.D.L. d'ordre 1 dans le cadre vectoriel (cf. cours de maths de 2ème année).*

Définition de la dérivée d'une fonction de $\mathbb{R}$ dans $\mathbb{R}^2$: coordonnées par coordonnées Si $f : x \mapsto (f_1(x), f_2(x))$, on définit $f'(x) = (f_1'(x), f_2'(x))$.

Retour à notre E.D. du second ordre :

On considère toujours l'E.D. $y''(x) = a(x)y'(x) + b(x)y(x) + c(x)$.
Idée : on pose $Y(x) = (y(x), y'(x))$. Avec la déf. précédente de la dérivée $Y'(x) = (y'(x), y''(x))$.
L'E.D. initiale est équivalente au système $\begin{cases} y'(x) = y'(x) \text{ (oui c'est trivial)} \\ y''(x) = a(x)y'(x) + b(x)y(x) + c(x) \end{cases}$.

L'E.D. initiale peut donc s'écrire comme une équation du premier ordre pour la fonction $x \mapsto Y(x)$ sous la forme :

$$Y'(x) = F(x, Y(x)),$$

où en notant $x \in \mathbb{R}$ et $Y = (Y_1, Y_2) \in \mathbb{R}^2$, $F : (x, Y) \mapsto (Y_2, a(x)Y_2 + b(x)Y_1 + c(x))$.

3.2 Traduction en SciLab : exemple pour l'O.H.

N.B. *On choisit, pour la commodité des applications à la physique, d'appeler plutôt t la variable des fonctions.*

On considère l'E.D. $y''(t) + \omega_0^2 y(t) = 0$. On peut la réécrire en posant $Y(t) = (y(t), y'(t))$ sous la forme $Y'(t) = F(t, Y(t))$ où pour $t \in \mathbb{R}$ et $Y = (Y_1, Y_2) \in \mathbb{R}^2$:

$$F(t, Y) = (Y_2, -\omega_0^2.Y_1).$$

Application en SciLab : on veut tracer la solution pour $\omega_0 = 2$, au problème de Cauchy posé par cette E.D. $y''(t) + 4y(t) = 0$ avec $y(0) = 1$ et $y'(0) = 0$. On utilise le script suivant :

```
clf()
function Yprim=F(t,Y);
    Yprim(1)=Y(2)
    Yprim(2)=-4*Y(1)
endfunction
t0=0
Y0=[1;0]// C.I. rentrées comme un vecteur vertical
t=0:0.01:10
Y=ode(Y0,t0,t,F) // Y est une matrice à 2 Lignes et  length(x) colonnes
plot2d(x,Y(1,:))
```

3.3 Une leçon importante de l'exemple précédent

Attentions aux vecteurs verticaux ou horizontaux

Fabrication d'un vecteur entrées par entrées

Par défaut si un vecteur `u` n'existe pas et qu'on rentre :

```
-->u(1)=12;
-->u(2)=3;
-->size(u)
 ans=
    2.    1.
```

On a un vecteur à deux lignes et une colonne.

Moralité : un vecteur fabriqué par cette méthode est vertical

Application à la fonction définie F définie au § 3.2

Cette fonction prend en entrée un vecteur `Y` qui peut indifféremment être vertical ou horizontal, car les commandes `Y(1)` et `Y(2)` s'appliquent dans les deux cas. En revanche elle renvoie un vecteur qui est vertical.

La variable de retour de `ode` est un tableau à deux lignes : avec en première ligne les valeur de y et en seconde ligne les valeurs de y'

Retenir aussi :

La commande d'extraction d'une ligne (ou d'une colonne) dans un tableau :

```
-->A=[1,2,3;4,5,6]
 A  =

    1.    2.    3.
    4.    5.    6.

-->A(1,:)
 ans  =

    1.    2.    3.

-->A(2,:)
 ans  =

    4.    5.    6.
-->A(:,1)
 ans  =

    1.
    4.
```

3.4 Application au portrait de phase de l'O.H.

Rappel sur l'obtention de l'intégrale première de l'énergie :

A partir de l'E.D. donnée par la R.F.D. pour un oscillateur harmonique :

$$my''(t) = -ky(t)$$

par multiplication des deux membres par $y'(t)$, on obtient

$$my'(t)y''(t) = -ky'(t)y(t)$$

qui équivaut à l'égalité :

$$\frac{d}{dt}(\frac{1}{2}my'(t)^2 + \frac{1}{2}ky(t)^2) = 0$$

et finalement à la conservation de l'énergie : il existe une constante E telle que pour tout t

$$\frac{1}{2}my'(t)^2 + \frac{1}{2}ky(t)^2 = E \quad (\dagger)$$

Une courbe dans l'espace des phases

Par déf. la courbe dans l'espace des phases correspondant à une solution de l'équation de l'O.H. est *la courbe paramétrée* $t \mapsto (y(t), y'(t))$. Avec le résultat du § 3.2, son tracé est immédiat :

```
scf(1);
plot2d(Y(1,:),Y(2,:))
```

On obtient bien des ellipses comme prévus par (†). Mieux (cf. cours de physique) si on prend comme variables $(y(t), \dfrac{y'(t)}{\omega_0})$ on obtient un cercle puisque :

$$\frac{1}{2}ky^2 + \frac{1}{2}m(y')^2 = E \Leftrightarrow y^2 + (\frac{y'}{\omega_0})^2 = \frac{2E}{k} = C$$

ce qui se voit bien à condition de prendre des coordonnées en base orthonormées :

```
scf(2)
clf()
a=gca()//donnée axe
a.isoview="on"
plot2d(Y(1,:),Y(2,:)/(2))
```

Tout un portrait

Pour avoir tout un portrait de phase, on doit *faire varier les C.I.* Cf. T.P.