Université Pierre et Marie Curie
LM 336
Année universitaire 2007-2008

Méthodes numériques pour les équations différentielles

D. Yakoubi

Première séance SCILAB
Représentation et approximation de fonctions d'une variable

Cette première séance *Scilab a plusieurs objectifs: tout d'abord, reprendre contact avec le logiciel, que vous avez déjà utilisé pendant le module d'algèbre linéaire. Pour cela, la plupart des commandes Scilab nécessaires ont été rappelées, en tout cas pour répondre aux premières questions, mais cela ne sera pas le cas à toutes les séances et vous devez aussi vous servir de l'aide en ligne et du manuel.*
Par ailleurs, le thème des premières séances correspond au premier chapitre du cours c'est-à-dire l'approximation des fonctions.
Tout d'abord, quelques étapes à faire à chaque séance:

- *Créer un répertoire portant votre nom de famille, et un sous-répertoire spécifique à la séance d'aujourd'hui (dans le bureau, cliquez sur le bouton droit de la souris et pointer sur* `nouveau` *puis sur* `dossier` *et cliquez avec le bouton gauche).*

  Attention: il n'y a pas de sauvegarde automatique sur ces machines et vous n'êtes pas sûrs de retrouver vos fichiers d'une semaine à l'autre. Vous devez donc faire une copie de votre travail (sur clé usb ou sur votre compte email).

- *Lancer une session* `Scilab`*, et se positionner dans votre répertoire de travail avec le menu* `file`*.*

- *Ouvrir une fenêtre d'édition de fichier (on utilisera exclusivement l'éditeur* `BlocNote`*), dans laquelle vous allez sauvegarder (recopier avec la souris) les commandes que vous taperez dans la fenêtre Scilab .*

1. **Représentation graphique d'une fonction d'une variable réelle.**
   Dans cette première partie, on va rappeler les différents points de syntaxe Scilab permettant de définir et de représenter graphiquement des fonctions d'une variable.

   (a) *Créer un tableau x (sous forme de vecteur colonne) contenant les abscisses auxquelles on va représenter la fonction. Ce sont les points $x_i = a * i + b$. Retrouver différentes syntaxes pour initialiser le tableau x avec les valeurs $0, 0.02, 0.04, ..., 1$. Par exemple*

   ```
   x=[0:0.02:1]';
   ```

   (b) *Définir localement, à l'aide de $deff$ la fonction $g(x) = \sin(\pi x) + \cos(\pi x)$*

   ```
   deff("[y]=g(x)","y=sin(%pi*x)+cos(%pi*x)");
   ```

   (c) *Représenter graphiquement la fonction g en utilisant* `plot2d`*.*

   ```
   plot2d(x,g(x));
   ```

   (d) *Afficher la courbe en rouge, avec des axes gradués (de -2 à 2 pour l'axe vertical) et comme légende $''g(x)''$*

```
xbasc();   // effacage de la fenetre graphique
plot2d(x,g(x),style=5,leg="g(x)",nax=[2,10,2,10],rect=[0,-2,2*%pi,2]);
```

`style` *donne la couleur de la courbe (ici style=5 est le code du rouge),* `leg` *donne la légende de la courbe,* `nax` *donne l'échelle de graduation des axes :* `nax=[nx,Nx,ny,Ny]` *produira Nx grands tirets et nx petits tirets sur l'axe des x, enfin* `rect=[xmin,ymin,xmax,ymax]` *donne les dimensions du rectangle dans lequel représenter la fonction.*

(e) *Définir la fonction* $f(x) = x^3 - x^2 + x - 1$ *dans un fichier interpol.sci. (Utiliser la bonne syntaxe pour function).*

```
function [y]=f(x)
y=x^3-x^2+x-1;
```

(f) *Définir un deuxième tableau d'abscisses* $xf_i = -1, -0.9, -0.8, ..., 2$.

```
xf=[-1:0.1:2]';
```

(g) *Définir un tableau* yf *contenant les valeurs de la fonction* f *aux abscisses* xf. *Représenter sur le même graphique et dans le même repère la fonction* g *sur l'intervalle* $[0, 1]$ *en rouge, la fonction* f *sur l'intervalle* $[-1, 2]$ *en vert.*

```
getf('interpol.sci')
// chargement des fonctions definies dans interpol.sci
yf=f(xf);
box=[-1,-4,2,5];
xbasc();
plot2d(x,g(x),style=[5,1],leg="g(x)",rect=box);
// ici style prend deux valeurs : 5 est le code couleur et 1 est
la position dans la liste des legendes.
plot2d([-1,2],[0,0],style=[2,2],leg="ox",rect=box);
//on trace l'axe des x pour x compris entre -1 et 2
plot2d(xf,yf,style=[3,3],leg="f(x)",rect=box);
```

2. **Interpolation par des splines**

Soient $[a, b]$ *un intervalle de* $\mathbb{R}$ *et* $a = x_0 < x_1 < \ldots < x_n = b$ *des points distincts de* $[a, b]$. *On dit qu'une fonction* $s \in \mathcal{C}^2_{[ab]}$ *est une fonction spline cubique sur* $[a, b]$ *si pour* $0 \leq i \leq n-1$, *on a* $s \in P_3$ *sur* $[x_i, x_{i+1}]$, *i.e.* s *est un polynôme de degré au plus trois sur cet intervalle. Si* $f : [a, b] \to \mathbb{R}$ *est une fonction, on posera* $f_i = f(x_i)$ *pour* $0 \leq i \leq n$. *On définit conventionnellement la spline associée à* f *pour les abscisses* x_i *par l'unique fonction spline cubique* s *telle que*

$$s(x_i) = f_i \quad \forall 0 \leq i \leq n \tag{1}$$

Le calcul de cette fonction s avec Scilab se fait avec la fonction splin qui calcule la valeur des dérivées de s aux points $(x_i, f(x_i))$.
La séquence d'appel est:
$d = splin(x, f[, "periodic"])$
Les paramètres signifiant:

```
x          : vecteur des abscisses \\
f          : vecteur des ordonnees de meme taille que x
"periodic"  : (si on impose que s soit periodique)
```

Description de la méthode:
Etant données les valeurs f_i de la fonction f aux point x_i, splin calcule une spline cubique s. Les abscisses doivent être en ordre croissant. Si on recherche une spline périodique on doit avoir $f_1 = f(x_1) = f(x_n) = f_n$. En sortie, on récupère le tableau d contenant les dérivées de la spline s aux abscisses x_i.

On va vouloir ensuite calculer les valeurs de la spline s aux abscisses xp, différentes de x. (Aux points x la spline s et la fonction f coincident). Pour cela, on utilise la fonction interp qui a la syntaxe suivante $[f0[, f1[, f2[, f3]]]] = interp(xp, x, f, d)$
En entrée:
xd: tableau d'abscisses où on veut calculer la spline
x,f,d: tableaux de définition de la spline: $x_i, f_i = s(x_i), d_i = s'(x_i)$ (d a été calculé au préalable par splin)
En sortie:
fn, pour $n = 0, ...3$: tableaux contenant les valeurs de $s(n)$ aux abscisses xp

(a) *Définir un tableau xp d'abscisses plus espacées pour la fonction g. Par exemple $0, 0.2, ..., 1$. Calculer la spline périodique s définie par les valeurs de g aux abscisses xp.*

(b) *Représenter simultanément les deux courbes:*
g aux abscisses x et
la spline s aux abscisses x,

3. **Interpolation de Lagrange**
Soit $F : [a, b] \longrightarrow \mathbb{R}$ une fonction continue, et $x_0, x_1, ..., x_n$ $(n+1)$ points dans $[a, b]$ deux à deux distincts, non nécessairement rangés par ordre croissant. Alors il existe un unique polynôme $P_n \in P_n[x]$ tel que:

$$P_n(x_i) = f(x_i), \ \forall i = 0, 1, ..., n$$

La qualité de l'interpolation varie avec le nombre de points, avec beaucoup de points on arrive à une bonne approximation à condition que la stabilité soit assurée.

(a) **Construction de Lagrange:** *Le polynôme d'interpolation de Lagrange s'écrit:*

$$P(x) = \sum_{j=0}^{n} f(x_j) l_j(x),$$

où: $l_j(x) = \prod_{k \neq j} \frac{(x - x_k)}{(x_j - x_k)}$

- *les $l_j(x)$ sont appelés polynômes de base de Lagrange.*
- *la famille $\{l_j(x)\}_{j=0,n}$ forme une base de P_n.*

(b) *Inconvénients de l'écriture*

i. *Les $l_i(x_j)$ sont lourds à calculer quand n est grand.*
ii. *Le procédé n'est pas récursif. Il faut reconstruire tous les l_j si on ajoute un point d'interpolation.*

On va donc décrire ici la méthode d'interpolation de Newton utilisant les différences divisées qui sont simples et bien adaptées au calcul scientifique, grâce au caractère récursif de la constuction.

(c) *Forme de Newton*
soit $f[x_0, ..., x_j]$ le coefficient directeur de p_j (coefficient devant le terme de plus haut degré, x^j), alors:

$$P_n(x) = f[x_0] + \sum_{j=1}^{n} f[x_0, ..., x_j](x - x_0)...(x - x_{j-1}).$$

Cette représentation du polynôme p_n est appelée forme de Newton.
$f[x_0] = f(x_0)$, *et pour* $k \geq 1$,

$$f[x_0, .., x_k] = \frac{f[x_1, .., x_k] - f[x_0, .., x_{k-1}]}{x_k - x_0}$$

La quantité $f[x_0, ..., x_k]$ *est appelée différence divisée d'ordre* k *de* f *aux points* $x_0, ..., x_k$.

(d) **Algorithme de construction et convention d'écriture:**

$$\begin{array}{cccccc} x_0 & f(x_0) & & & & \\ x_1 & f(x_1) & f[x_0, x_1] & & & \\ x_2 & f(x_2) & f[x_1, x_2] & f[x_0, x_1, x_2] & & \\ \vdots & \vdots & & & f[x_0, ..., x_{n-1}] & \\ x_{n-1} & f(x_{n-1}) & f[x_{n-1}, x_n] & & f[x_1, ..., x_n] & f[x_0, ..., x_n] \\ x_n & f(x_n) & & & & \end{array}$$

Avantage: Construction récursive:

$$p_{n+1}(x) = p_n(x) + (x - x_0)(x - x_1)....(x - x_n) f[x_0, .., x_{n+1}]$$

Il suffit d'ajouter une branche dans l'arbre (calcul de n *différences divisées).*

4. **Implémentation de l'interpolation de Lagrange**

(a) Ecrire sur le papier l'algorithme de Newton pour calculer les différences divisées. Traduire en langage Scilab et écrire la procédure
$function \quad [c] = DiffDiv(x, h)$ *dans le fichier interpol.sci*
h *est la fonction à interpoler, accessible depuis interpol.sci*
x *est un tableau contenant les* $n + 1$ *abscisses d'interpolation.*
c *contient en sortie les* $n + 1$ *coefficients* $c_i = f[x_0, ...x_i]$ *pour* $i = 0, ...n$
Attention *au décalage des indices, en Scilab les tableaux démarrent à l'indice* 1, *pas zéro!.*

(b) Ecrire sur le papier l'algorithme de calcul du polynôme d'interpolation en un point quelconque t, *connaissant les coefficients* c_i. *Pour cela adapter l'algorithme de Horner pour calculer la valeur* v *au point* x *d'un polynôme connu par ses coefficients dans la base canonique*

$$p(x) = \sum_{i=0}^{n} a_i x^i = a_0 + x\,(a_1 + x\,(a_2 + ...))$$

$v = a_n$
Pour i *allant de* $n - 1$ à 0 *faire:* $v = x * v + a_i$

(c) Ecrire l'algorithme adapté dans la fonction Scilab
$function \quad [y] = HHorner(c, x, t)$
x *contient les* $n + 1$ *abscisses d'interpolation,*
c *les* $n + 1$ *différences divisées calculées au préalable en utilisant la fonction* $DiffDiv$,
t *le point ou on calcule la valeur du polynôme,*
y, *en sortie, contient cette valeur.*

5. Utiliser ces fonctions pour tracer l'interpolation de Lagrange de la fonction g *aux points* xp. *La représentation graphique sera faite, elle, aux points* x.

6. **Phénomène de Runge**
Comparer toutes ces méthodes d'approximation pour la fonction

$$g(x) = \frac{1}{x^2 + \lambda^2} \quad \forall x \in [-1, 1]$$

(a) *Pour des points d'interpolation xp régulièrement espacés, jouer sur la valeur de λ et sur le nombre de points pour montrer que l'interpolation de Lagrange peut être instable.*

(b) *Sélectionner les points d'interpolation avec la souris. Montrer que pour les deux méthodes, splines et Lagrange, leur placement influe sur la précision de l'interpolation.*