MT26 TP
séances numéro 1 et 2

UTBM-année 2000/2001
Printemps 2001

SÉANCE D'INITIATION

PARTIE FACULTATIVE

Les étudiants qui connaissent déjà bien matlab ou qui ont fini la séance d'initiation pourront regarder cette partie facultative.

1. Les polynômes

Le polynôme

$$P(X) = a_n X^n + a_{n-1} X^{n-1} + ... + a_1 X + a_0,$$

est représenté en matlab par le vecteur ligne $[a_n\, a_{n-1} ... a_1\, a_0]$ (de longueur $n+1$).

Outre les opérations sur les matrices qui s'appliquent donc aux polynômes, il existent un certain nombre de fonctions spécifiques aux polynômes.

>> *conv*(A,B) multiplie les deux polynômes A et B.

>> [Q,R]=*deconv*(A,B) renvoie les deux polynômes Q et R tels que $A = BQ + R$ avec degré de R strictement inférieur à celui de B.

>> *roots*(P) renvoie les racines du polynôme P (dans $\mathbb{C}$). Réciproquement, *poly* calcule le polynôme à partir de ces racines.

>> P=[1 -6 11 -6] ;
>> S=roots(P)
>> poly(S)
>> poly(S')

>> *polyval*(f,X) renvoie les valeurs de la fonction f sur la matrice X.

>> *polyder*(f) dérive le polynôme f.

>> [q,d]=*polyder*(a,b) calcule la dérivée de a/b (où a et b sont deux polynômes) sous la forme q/d.

>> *polyval*(f,A) renvoie $f(A) = p_1 A^n + p_2 A^{n-1} + ... + p_n A + p_{n+1} I$, où A est une matrice et f un polynôme.

Exercice 1.1. Faire une fonction qui enlève les zéros superflus des polynômes (par exemple, elle renverra [0 1 2] sur [1 2]).

Exercice 1.2. Faire une fonction qui permette de faire la somme de deux polynômes de degrés différents (ce qui n'est pas possible avec matlab, car on ne peut additionner deux vecteurs de tailles différentes).

Exercice 1.3. Calculer le reste et le quotient de la division de A par B, où

$$A = 3X^6 - 8X^5 + X^4 + 16X^3 - 19X^2 + 9X + 6,$$
$$B = 3X^2 + 4X - 1.$$

Que remarquez-vous ? Comment pourrez on y remédier ? Faites en une fonction et testez la sur les calculs de la division euclidienne de A par B où

$$A = X^5 - X^4 + 6X^3 + 2X^2 - 6X - 9,$$
$$B = X^3 - X^2 + 7X + 1,$$

et

$$A = X^5 - X^4 + 6X^3 + 2X^2 - 7X - 1,$$
$$B = X^3 - X^2 + 7X + 1.$$

Matlab permet aussi de déterminer la décomposition en éléments simples dans $\mathbb{C}$. Si A et B sont deux polynômes, et si A/B n'admet que des pôles simples, on peut calculer, avec matlab, les résidus r_k, les pôles p_k et le polynôme K tels que

$$\frac{A}{B} = K + \frac{r_1}{X - p_1} + \frac{r_2}{X - p_2} + ... + \frac{r_n}{X - p_n} + K.$$

Pour cela on tape, >> [r,p,k]= *residue*(A,B).

Les résidus r_k sont stockés dans le vecteur r, les pôles p_k sont stockés dans le vecteur p et K dans le vecteur k.

Si la fraction A/B admet pour le pôle p_j une multiplicité m, matlab remplacera la fraction $r_j/(X_{-}p_j)$ par

$$\frac{r_j}{X - p_j} + \frac{r_{j+1}}{(X - p_j)^2} + ... + \frac{r_{j+m-1}}{(X - p_j)^m}.$$

Si on cherche la décomposition en éléments simples dans $\mathbb{R}$, on utilisera la décomposition dans $\mathbb{C}$ et on regroupera les termes conjugués deux à deux.

Exercice 1.4. Calculer la décomposition dans $\mathbb{C}$ de

$$\frac{17X^2 + 36X + 9}{2X^3 + 4X^2 + 2X + 4}.$$

et en déduire sa décomposition dans $\mathbb{R}$.

Exercice 1.5. Calculer la décomposition de

$$\frac{2X^6 - 32X^5 + 188X^4 - 516X^3 + 709X^2 - 479X^2 - 479X + 134}{X^5 - 8X^4 + 24X^3 - 34X^2 + 23X - 6}.$$

2. Les chaînes et autres types de données

2.1. Les chaînes

une chaîne est un vecteur ligne dont le nombre de composantes est égal à la longueur de la chaîne.

Exemple :

```
>> ch='une chaîne'
>> size(ch)
>> length(ch)
```

Pour concaténer deux chaînes, on écrit celle-ci comme un vecteur ligne :

```
>> ch=[ch,' c''est pratique ','et agréable'] ;
>> ch
>> disp(ch)
>> disp('ch')
```

Pour rentrer une chaîne, on tape

```
>> ch= input('entrez la chaîne : ', 's') ;
```

Les fonctions *num2str* et *int2str* transforment respectivement un réel et un entier en une chaîne (cf. exercice suivant).

Exercice 2.1. Écrire une fonction qui transforme un polynôme (stocké sous forme de tableau) en une chaîne de caractère ; par exemple, le polynôme [3 8 -7 0 1] aura pour image la chaîne '3X^4 +8X^3-7X^2+X' .

Il existe un certain nombre de fonctions sur les chaînes dont voici quelques exemples :

>> *abs*
>> *setstr*
>> *isstr*
>> *'*
>> *str2num*
>> *sprintf*
>> *upper*
>> *lower*
>> *eval*
>> *blanks*
>> *deblank*

2.2. Les entiers

Il existe un certain nombre de fonctions spécifiques aux entiers :

>> *mod*(a,b) : renvoie le quotient de la division entière de a par b (cf. *rem*).
>> *factor*(a) : renvoie les facteurs premiers de a.
>> *primes*(a) : renvoie un vecteur ligne avec les nombres premiers plus petit que a.
>> *isprime*(a) : retourne 1 si a est un nombre premier.

>> *nextpow2*(a) : renvoie n tel que $2^{n-1} \leq a < 2^n$.
>> *perms(c)* : renvoie toutes les permutations possibles du vecteur c.
>> *nchoosek*(v,k) : v est un vecteur de longueur supérieure ou égale à k ; cette fonction renvoie toute les permutations possibles de k éléments parmi les composantes de v.

2.3. Les ensembles

Les vecteurs lignes de matlab peuvent être aussi considérés comme des ensembles, dont on donne quelques fonctions :
intersect
ismenber
setdiff
sextor
unique
union

3. Les sons

Si la carte son est correctement installée, vous pouvez jouer de la muisque avec matlab. Vous pouvez aussi créer des sons de façon analytique et les écouter ensuite. Regardez la démonstration qui contient quelques sons en lançant *xpsound*.

4. Analyse de fonctions

Représenter la fonction g de $\mathbb{R}$ dans $\mathbb{R}$ telle que

$$g(x) = \frac{5x - 6,4}{(x-1,3)^2 + 0,002} + \frac{9x}{x^3 + 0,03} - \frac{x - 0.4}{(x-0,92)^2 + 0,005}.$$

En trouver tous les zéros en utilisant la fonction x=*fzer*('exozero',alpha) où alpha désigne la valeur initiale de recherche. Pour la première racine, calculer x=*fzer*('exozero',alpha,tol) où tol est l'erreur admise. On prendra tol dans l'ensemble $\{10^{-6}, 10^{-10}, 10^{-15}, \text{eps}, 10^{-20}\}$. Que remarquez vous ?

Trouver le minimum de g grâce à la fonction *fmin*(g,a,b).

5. L'utilisation du debugger

Avec matlab, il est possible de debugger agréablement ses script, en suivant l'évolution de l'exécution d'un script, pas à pas, ou en s'arrêtant à des endroits précis, tout en contrôlant ou modifiant les différentes variables mises en jeu.

Pour cela, une fois que l'on a ouvert les différents scripts avec l'éditeur de matlab, il faut placer, en déplaçant le curseur, des “points d'arrêt” là ou l'on désirer voir le programme s'arrêter provisoirement.

On lance ensuite le programme de la fenêtre matlab (ou on appelle la fonction) ; il apparaît une invite “K>>”. On contrôle ensuite l'exécution du programme, à partir de l'éditeur de plusieurs façons :
- on fait debugg/continu pour aller d'un point d'arrêt à l'autre.
- on fait F10 pour avancer pas à pas.

- on fait F11 pour avancer pas à pas en entrant dans chacune des sous-procédures appelées . **Attention**, dans ce cas, on pourra éventuellement entrer dans des sources de fonctions déjà programmées de matlab, qu'il ne faut pas modifier.

Dans tous les cas, apparaît une petite flèche dans l'éditeur qui indique à quel endroit du programme on se trouve. Parallèlement, dans la fenêtre matlab, on peut contrôler les variables et même les modifier. Ces variables sont spécifiques à un espace de travail, indiqués dans le menu stack

Quand le programme a tourné ou est en train de tourner avec le debugger, on peut accéder aux valeurs des variables en déplaçant la souris sur chacune des variables dans l'éditeur matlab : sa valeur apparaît alors.

Exercice 5.1. Ecrire un script faisant appel à plusieurs sous fonctions et tester le debugger.

6. Calcul d'intégrales et résolution d'équations différentielles

Outre les méthodes d'intégration numériques déjà vues (rectangles et trapèzes), on peut aussi citer la règle de Simpson ; matlab l'utilise avec l'instruction *quad*(f,a,b).

Calculer l'intégrale double

$$\int_0^1 \int_0^1 e^{-x^2-y^2} dxdy,$$

en utilisant *dblquad*(f,a,b,c,d). **attention**, dans la définition de f, il faudra prendre garde à taper f(x,y)=exp(-x.^2-y.^2).

Matlab permet aussi d'intégrer des équations différentielles ordinaires du type :

$$\begin{cases} X'(t) = F(t, X(t)), \\ X(t_0) = X_0. \end{cases}$$

en utilisant *ode45*.

Par exemple, on intégrera numériquement le problème sur [01]

$$\begin{cases} x'(t) = -x^2(t), \\ x(0) = 1, \end{cases}$$

en utilisant la fonction xprim définie par le fichier suivant :

```
function res=xprim(t,x);
% intégration de l'edo
% y'(t)=xprim(t,y(t)) et y(t0)=y0

res=-x.^2
```

et en tapant

```
>> [T,X]=ode45('xprim',[0,1],1);
>> plot(T,X);
```

On pourra aussi regarder les fonctions *ode23*, *ode113*, *ode23t*... qui correspondent à d'autres méthodes d'intégration.