Méthodes itératives

Il s'agit d'illustrer des méthodes de point fixe, en particulier la méthode de Newton. Ces méthodes permettent par exemple de résoudre une équation en cherchant la limite d'une suite récurrente. On supposera que la suite u_n est donnée par la relation de récurrence $u_0 = a, \quad u_{n+1} = f(u_n)$.

Ces méthodes numériques peuvent aussi être programmées sur des calculatrices qui ne font pas de calcul formel.

1 L'algorithme

Il faut tout d'abord écrire le programme ci-dessous qui affiche la valeur de u_i si $|u_i - u_{i-1}| < eps$ ou "non trouve" si $i > n$, puis définir la fonction à itérer.
ATTENTION : trouver u_i tel que $|u_i - u_{i-1}| < eps$ ne prouve pas la convergence de la suite u_n (ex $u_n = \Sigma_{i=1}^n \frac{1}{i}$).

```
fonction iter(f,a,n,eps)
local b,i
i:=1
repeter
  b:=a
  a:=f(a)
  i:=i+1
jusqua (i>n) ou (abs(b-a)<eps)
si (i>n) alors
  afficher ‘‘non trouve’’
sinon
  afficher a, i
fsi
ffonction
```

Les paramètres de la fonction ci-dessus sont :
`a` qui repésente u_0
`n` qui repésente le nombre maximum d'itérations
`eps` qui est l'écart toléré entre deux u_i consécutifs.
En cours de programme, `i` est l'indice, `b` représente u_{i-1} et `a` représente u_i.

La section 3 donne la traduction pour l'émulateur de HP49. La traduction pour `sa` calculatrice est laissée en exercice au lecteur.

2 Application Sequence des calculatrices

ATTENTION sur les calculatrices on ne peut pas utiliser des paramètres qui soient des fonctions. On suppose donc que la fonction à itérer s'appelle f.

2.1 Casio

Touche MENU, choisir RECUR. Dans le bandeau, on peut sélectionner avec TYPE si on étudie une suite récurrente à un ou deux termes. Indiquez ensuite les valeurs initiales avec le menu RANG du bandeau. Affichez la table des valeurs numériques de la suite, puis tracez le graphe. Si la suite récurrente est à un terme, le bandeau de TABL comporte l'option WEB.

2.2 TI

Touche MODE, puis choisir SEQUENCE ou Seq en face de GRAPH. Valider ce choix par ENTER. Ensuite ouvrir le menu Y= et entrer la suite récurrente. Dans le menu Axes (F7 sur la TI92) choisir WEB. Pour tracer l'escargot, taper sur ◇ GRAPH.

2.3 HP40

On peut se servir de l'aplet Sequence (touche APLET puis on met Sequence en surbrillance, puis on appuie sur ENTER).
Il suffit alors de définir la suite u_n en donnant la valeur de u_1 et de u_2, puis la relation de récurrence $u_n = f(u_{n-1})$ ou $u_n = f(u_{n-1}, u_{n-2})$.
La touche NUM donne le tableau des différentes valeurs de u_n.
La touche PLOT donne "l'escargot" si on a réglé le SETUP du PLOT en choisissant cobweb.

3 Programmation sur emu49

Pour exécuter plusieurs fois de suite une seule itération, l'interface la plus efficace est le mode RPN. Tapez sur la touche MODE puis sur la touche +/- puis sur ENTER.
On peut définir une fonction de la façon suivante :
On tape 'F(X)=COS(X)/2'
puis on appuie sur le shift-bleu et la touche DEF pour définir F.
Appuyez ensuite sur la touche VAR : F se trouve dans la bandeau (la dernière ligne de l'écran) et correspond à la touche F1.
Placez la valeur initiale sur la pile puis tapez la touche F1 correspondant à l'intitulé F du menu du bandeau. Ceci exécute une itération. Il suffit d'appuyer plusieurs fois de suite pour itérer plusieurs fois.

On peut également écrire un programme, pour simplifier on va supposer que la fonction à itérer est stockée dans la variable F et la tolérance dans la variable EPS. Revenez en mode algébrique en tapant sur les touches MODE, +/- et ENTER. puis saisissez la valeur de EPS, par exemple:
0.0001 STO> EPS (STO> s'obtient en appuyant sur la touche STO>).
et tapez le texte suivant (pour saisir le séparateur d'instruction ; tapez simultanément sur le shift-rouge et la touche SPC)

```
<< -> X N                 @ 2 arguments: u_0, nb max d'iterations
```

```
  << 0. -> I              @ variable locale compteur d'iteration
     <<
       DO
         STO+(1.,'I') ;   @ incremente le compteur d'iteration
         F(X) STO> X  ;   @ u_{i+1}=f(u_i)
       UNTIL
         ABS(F(X)-X)<EPS OR N<I  @ test d'arret
       END ;
       IF N<I THEN "Non trouve" ELSE ->TAG(X,I) END
     >>
   >>
>> -> ITER
```

Pour exécuter le programme, tapez par exemple: `ITER(0.,30.)`
où 0. est la valeur initiale de la suite et 30. est le nombre maximal d'iterations sur la pile. Le programme renvoie u_i ou affiche un message en cas de non succès (plus de 30 itérations).
Voici aussi la version `RPN` de ce programme qui est plus rapide et s'exécute aussi sur les HP48 (saisir le texte suivant puis entrez les deux arguments u_0 et n maximal puis tapez `ITER`)

```
<<                      @ X N
   0. -> N I
   <<
     DO                 @ X
       1. 'I' STO+      @ X (incremente le compteur I)
       DUP F            @ X F(X)
       SWAP OVER -      @ F(X) X-F(X)
     UNTIL
       ABS EPS <        @ F(X) ?
       I N > OR
     END
     IF
       I N >
     THEN
       "Non trouve" MSGBOX
     END
   >>                  @ F(X)
>> 'ITER' STO
```

4 Avec MuPAD

Pour définir la même fonction, on taperait :
`f:= ( x -> (cos(x)/2) )`
Attention aux parenthèses, la priorité de l'opérateur `->` n'est pas celle qu'on pense.

Pour calculer plusieurs itérations, initialiser `a` avec une valeur numérique, par exemple `a=0.0`, puis taper :
`a:=f(a);`
puis tapez sur la flèche vers le haut et `ENTER`. On peut aussi programmer avec un test d'arrêt. Ouvrez un fichier (menu `Files` puis `Open File`, nommez-le `iter.mu`) et tapeză:

```
iteration:=proc(f,x,maxiter,epsilon)
// f est la fonction, x la valeur initiale,
//  maxiter le nombre maximal d'iterations, epsilon la tolerance
local i,oldx;
begin
 i:=0;
 x:=float(x);
 epsilon:=float(epsilon);
 repeat
  oldx:=x;
  x:=f(x);
  i:=i+1;
 until ( (abs(x-oldx)<epsilon) or (i>maxiter) ) end_repeat;
 if (i>maxiter) then x,maxiter else x end_if ;
end_proc;
```

ATTENTION il ne faut rien taper après `end_proc;` (ni espace, ni enter).
Revenez dans le buffer `*MuPAD*` et taper `read("iter.mu");`
Ne pas oublier d'utiliser des valeurs numériques pour x, par exemple :
`iteration(cos,0.2,50,1e-3)`). On peut définir le nombre de chiffres significatifs (menu `MuPAD`, environnement puis `DIGITS`).

5 Complément de cours

5.1 Méthode de Horner

Soit le polynôme $P(X) = \Sigma_{i=0}^{n} a_i X^i$.
Etant donné c, on cherche le polynôme $Q(X) = \Sigma_{i=1}^{n} b_i X^{i-1}$ vérifiant :
$P(X) = (X - c)Q(X) + b_0$
On a donc :
$b_n = a_n$
$b_i = a_i + c.b_{i+1}$ $(i = n - 1..0)$
$b_0 = P(c) = a_0 + c.b_1$
On remarquera que le calcul de $b_0 = P(c)$ nécessite seulement le calcul de $b_{n-1}...b_0$ soit n additions et n multiplications.
Voici l'algorithme du calcul des b_i :
on suppose qu'en entrée on donne P sous la forme de la liste $\{a_0, ..., a_n\}$ et qu'en sortie on représente le polynôme $Q(X) = \Sigma_{i=1}^{n} b_i X^{i-1}$ par la liste $\{b_1, ..., b_n\}$
Attention ! si $m =$ la longueur de la liste représentant P on a :
$P[1] = a_n,\ P[2] = a_{n-1},\ ...,\ P[m] = a_1$

```
fonction horner(P,c)
local m,b,Q
m:=dim(P)
b:=P[1]
Q:={b}
pour i de  2 a m-1 faire
  b:=P[i]+c*b
  Q:=append(Q,b)
fpour
b:=P[m]+c*b
retourne Q,b
ffonction
```

Remarques : En MuPAD les listes s'écrivent avec des crochets (exemple Q:=[b];)
`HORNER` est implanté sur la HP49G les paramètres sont par exemple P[X] et c.
Exemple : $\mathtt{HORNER}(\mathtt{X}^4 + 2.\mathtt{X}^3 + \mathtt{X}, 1) = \{\mathtt{X}^3 + 3.\mathtt{X}^2 + 3.\mathtt{X} + 4, 1, 4\}$
la liste résultat est : {Q[X],c,P[c]}.

5.2 Théorème du point fixe

Soit f est une application continue de I dans I (I désigne un intervalle fermé de $\mathbb{R}$). Si f est contractante ($\exists k < 1,\ \forall x,\ y \in I \times I,\ |f(x) - f(y)| \leq k|x - y|$) alors f admet un point fixe unique a et la suite des itérées ($u_0 \in I\ u_{n+1} = f(u_n)$ converge vers a).
Exemples : $f(x) = \cos(x)$ est contractante sur [0,1],
$f(x) = \tan(x)$ n'est pas contractante sur $[\frac{\pi}{2},\ \frac{3\pi}{2}]$

5.3 Méthode de Newton

Pour résoudre l'équation $f(x) = 0$, on itère la fonction : $g(x) = x - \dfrac{f(x)}{f'(x)}$.
Puisque $g'(x) = \dfrac{f(x).f''(x)}{f'(x)^2}$, si a est un zéro de f, on a :
$g(a) = a$, $g'(a) = 0$ et la suite des itérées de g ($u_n = g(u_{n-1})$ converge de façon quadratique ($|u_n - a| \leq k|u_{n-1} - a|^2$)

5.4 Algorithme de Héron

Cet algorithme, basé sur la méthode de Newton, permet de déterminer des valeurs approchées de racines d'entiers.
Exemple Calcul de $\sqrt{b}$ (b désigne un réel positif).
On pose :
$u_1 = a$ (a désigne un réel positif proche de $\sqrt{b}$)
$u_n = \dfrac{1}{2}(u_{n-1} + \dfrac{b}{u_{n-1}})$.

5.5 Accélération de convergence: procédé d'Aitken

Soit S_n une suite convergente vers S, on note :
$\Delta S_n = S_{n+1} - S_n$ et $\Delta^2 S_n = \Delta S_{n+1} - \Delta S_n = S_{n+2} - 2S_{n+1} + S_n$.
On pose alors :

$$S'_n = S_n - \frac{(\Delta S_n)^2}{\Delta^2 S_n}$$

La suite S'_n converge en général plus vite que S_n, on peut montrer que c'est le cas si $(S_{n+1} - S)/(S_n - S)$ converge vers une limite $\rho \in]-1, 1[$.

6 Exercices

- Résolution de $\cos(x) = x$ sur $[0, 1]$, de $\tan(x) = x$ sur $]\pi/2, 3\pi/2[$ en utilisant le théorème du point fixe, cf. la section 5.2. Si la fonction de votre équation n'est pas contractante, cherchez une équation équivalente dont la fonction est contractante (indication: si $|f'| > 1$ que peut-on dire de $|f^{-1}{}'|$ où f^{-1} désigne la fonction réciproque de f?).

- Recherche de valeur approchée de $\sqrt{2}$
  Vous cherchez une équation dont cette valeur est racine et appliquez une méthode de point fixe (par exemple $f(x) = \frac{2x+2}{x+2}$ admet $\sqrt{2}$ comme point fixe).
  Donner une valeur de $\sqrt{2}$ à 10^{-5} près en utilisant l'algorithme de Héron.
  Comparez avec l'itération de $f(x) = \frac{2x+2}{x+2}$.

- Recherche d'une valeur approchée de $\sqrt[3]{3}$ en utilisant la méthode de Newton.

- Recherche de racines d'un polynôme par la méthode de Newton.
  Exemple, racines de $x^4 + 1$.
  il faut partir d'une valeur initiale complexe (prendre une valeur non imaginaire pure).

- Recherche de toutes les racines d'un polynôme par élimination d'une racine (méthode de Horner).
  Exemple, factorisation de $x^5 + x + 1$.
  Pour éliminer les erreurs d'arrondi on applique une itération de Newton sur une estimation de racine avec le polynôme originel :
  On se rapproche des racines en utilisant un préfacteur ($\lambda = 0.9$ par exemple) dans la méthode de Newton :

  $$x_{n+1} = x_n - \lambda \frac{f(x_n)}{f'(x_n)}$$

  puis on prend comme valeur de u_0 l'estimation précédente et on utilise la méthode de Newton.

- Reprendre en utilisant le procédé d'accélération de convergence d'Aitken la résolution de $\cos(x) = x$ sur $[0, 1]$ et de $\tan(x) = x$ sur $]\pi/2, 3\pi/2[$.
  Résoudre par ce procédé l'équation :
  $10\cos(x) = x$ sur $[0,\ 10]$