UNIVERSITE DE GENEVE — Section de mathématiques

Faculté des sciences — Mercredi 25 octobre 2006

Optimisation – Série 1
Introduction à Matlab

Exercice 1 a) Implémenter la fonction Hello :

```
function Hello(n)
% HELLO print a welcome message
%   Hello(n) print the name of the author n times.
for i=1:n
  display('NomDeFamille');
end
```

b) En se rappelant que

$$\sin(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!},$$

Implémenter la fonction

```
function y= MySin(x,n)
% MYSIN compute the sine function
%   y=MySin (x,n) compute the sine of x using the first n
%   terms in the power series around 0.

y=0;
s=x;
for i=0:n
  y=y+s;
  s=-s*(x/(2*i+2))*(x/(2*i+3));
end
```

Exercice 2 Implémenter la fonction

```
function [y,a,b]=Bisection(f,a,b)
% BISECTION finds a zero of f in the interval [a,b]
%  y=Bisection(f,a,b) finds a zero of f in the interval [a,b] using the
%   method of bisection up to machine accuracy
```

```
if a>b,
  h=a;a=b;b=h;
end;
fa=feval(f,a);
fb=feval(f,b);
if fa*fb > 0,
  disp('Error: There may be no root in interval chosen');
else
  y=(a+b)/2;
  while y>a & y<b,
    fy=feval(f,y);
    if fa*fy>0,
      a=y; fa=fy;
    else
      b=y; fb=fy;
    end;
    y=(a+b)/2;
  end;
end;
```

Implémenter la fonction

a)
```
function [x1,x2]=Minimize(f,a,b)
% MINIMIZE finds a minimum of f in the interval [a,b]
%   x=Minimize(f,a,b) finds a minimum of f in the interval
%   [a,b]
alpha=(3-sqrt(5))/2;

if a>b,
  h=a;a=b;b=h;
end;
x1=a+alpha*(b-a);
x2=b-alpha*(b-a);
fx1=feval(f,x1);
fx2=feval(f,x2);

while a<x1 & x1<x2 & x2<b
  if fx1<fx2
    b=x2;
    x2=x1;
    fx2=fx1;
```

```
    x1=a+alpha*(b-a);
    fx1=feval(f,x1);
  else
    a=x1;
    x1=x2;
    fx1=fx2;
    x2=b-alpha*(b-a);
    fx2=feval(f,x2);
  end
end
```

b) C'est le seul choix de lambda qui permet d'économiser une évaluation de fonction par itération. En effet, $x_1 = (1-\lambda)a + \lambda x_2$ et $x_2 = (1-\lambda)a + \lambda x_2$. Ainsi, dans l'algorithme, soit x_1 devient x_2 si x_2 devient b soit x_2 devient x_1 si x_1 devient a.

Exercice 3 Utiliser les deux algorithmes précédents sur les exemples suivants :

a) La fonction cosinus entre 2.5 et 4.

```
> Bisection(@(x) sin(x), 2.5, 4)
       ans = 3.14159265358979
> Minimize (@(x) cos(x), 2.5, 4)
       ans = 3.14159266412651
```

La valeur de π est de 3.14159265358979. Bisection marche mieux car Minimize est mal conditionné. En effet, proche du minimum, la fonction à minimiser est très plate alors que la dérivée coupe nettement l'axe des abcisses.

b) Le modèle des battements du cœur vu en cours :

```
> Bisection (@(x) 70-30*exp(-100*x)-3000*(1-x)*exp(-100*x),0,1)
        ans = 0.03730190794993
> Minimize (@(x) 120*x+(50+30*exp(-100*x))*(1-x),0,1)
        ans = 0.03730190838817
```

c) La fonction de profit

```
> Bisection (@(p) exp(-p)/(1+exp(-p))^2*((p-1)-exp(-p)),0,100)
        ans = 1.27846454276107
> Minimize (@(p) -p*exp(-p)/(1+exp(-p)),0,100)
        ans = 1.27846456348519
```

d) Un modèle simple de gestion des ressources halieutiques : D'abord

```
function [p]=Fish(p0,delta,a,b,t)
% FISH Computes the number of fishes present in the ocean
%  p=Fish(p0,delta,a,b,T) computes the solution p(t) of
```

```
%  the ODE p'=a(1-delta)p-bp^2, p(0)=p0
p=a*delta*(1-delta)*(b*p0+(a*(1-delta)-b*p0).* ...
     exp(-a*(1-delta).*t)).^(-1);
```

```
function [p]=Fishder(p0,delta,a,b,t)
% Fishder Computes the derivative of Halieutique
%  p=Fishder(p0,delta,a,b,T) computes the derivative of the
%  solution p(t) of the ODE p'=a(1-delta)p-bp^2, p(0)=p0
p=a*(1-2*delta)*(b*p0+(a*(1-delta)-b*p0)...
 *exp(-a*(1-delta)*t)).^(-1)...
  +a^2*delta*(1-delta)*(1-t*(a*(1-delta)-b*p0))...
  .*exp(-a*(1-delta)*t)...
  .*(b*p0+(a*(1-delta)-b*p0)*exp(-a*(1-delta)*t)).^(-2);
```

Puis,

```
> Bisection(@(delta) quad(@(t) Fishder(1,delta,0.34,0.01,t),0,20),0,1)
        ans = 0.36258676667614
> Minimize(@(delta) -quad(@(t) Fish(1,delta,0.34,0.01,t),0,20),0,1)
        ans = 0.36258676689113
```

Evaluation du cours d'optimisation

- Les exercices : Les séries d'exercices rendues en retard seront comptées comme non rendues (*i.e.*, note 1 sur 6) dans le calcul de la note finale.
- Un examen oral durant la session d'examens sur le cours.

La note finale est de : $\frac{1}{5}$(exercices.) + $\frac{4}{5}$(note examen oral).

Assistant :	Kévin Santugini
Adresse électronique :	`Kevin.Santugini@math.unige.ch`
Page web :	`http://www.unige.ch/~santugin/index.php?page=enseignement`