

Tarea 5

Fecha de entrega: 19 de octubre de 2006

Entregas individuales e impresas

1. Escribir la función del método del acorchamiento (bracketing) y aplicarlo a la función que teníamos de funNR.m para que converja a la solución óptima

$$\theta^* = \frac{3}{11} = 0.2727\dots$$

Primero modificamos el signo de nuestra función objetivo, porque recordemos que estamos el programa que escribimos minimiza la función. Entonces nos queda:

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

Luego escribimos nuestro programa para el acorchamiento:

```
function br=bracketingm(fun,xmin,xmax,tol,maxit)
iter=1;
a=xmin;
c=xmax;
b=(c-a)/2;
fa=feval(fun,a);
fc=feval(fun,c);
fb=feval(fun,b);
if fb<=fa & fb<=fc
    disp('se puede inicializar el programa')
else
    disp('escoger nuevos xmin y xmax')
end
while iter<=maxit
    if b-a<c-b
        d=(b+c)/2;
    else
        d=(a+b)/2;%.25
    end
    fd=feval(fun,d);
```

```

fb=feval(fun,b);
if d<=b
    if fd>=fb
        a=d;
    else
        c=b;
        b=d;
    end
else
    if fd<=fb
        a=b;
        b=d;
    else
        c=d;
    end
end
if (c-a)<tol
    disp('hemos convergido')
    break;
end
disp([iter a b c d])
iter=iter+1;
end
b

```

Por lo tanto el resultado nos queda en Matlab de la siguiente manera:

```
>> bracketingm('funNRbr',0,1,.0001,50)
```

se puede inicializar el programa

```

1.0000 0 0.2500 0.5000 0.2500
2.0000 0.1250 0.2500 0.5000 0.1250
3.0000 0.1250 0.2500 0.3750 0.3750
4.0000 0.1875 0.2500 0.3750 0.1875
5.0000 0.1875 0.2500 0.3125 0.3125
6.0000 0.2188 0.2500 0.3125 0.2188
7.0000 0.2500 0.2813 0.3125 0.2813

```

```

8.0000 0.2500 0.2656 0.2813 0.2656
9.0000 0.2578 0.2656 0.2813 0.2578
10.0000 0.2656 0.2734 0.2813 0.2734
11.0000 0.2695 0.2734 0.2813 0.2695
12.0000 0.2695 0.2734 0.2773 0.2773
13.0000 0.2715 0.2734 0.2773 0.2715
14.0000 0.2715 0.2734 0.2754 0.2754
15.0000 0.2715 0.2725 0.2734 0.2725
16.0000 0.2720 0.2725 0.2734 0.2720
17.0000 0.2725 0.2729 0.2734 0.2729
18.0000 0.2725 0.2727 0.2729 0.2727
19.0000 0.2726 0.2727 0.2729 0.2726
20.0000 0.2726 0.2727 0.2728 0.2728
21.0000 0.2726 0.2727 0.2728 0.2726
22.0000 0.2726 0.2727 0.2728 0.2728

```

hemos convergido

```

b =
0.2727

```

2. Resolver el sistema no lineal a través de encontrar el mínimo de las funciones $f^2 + g^2 + h^2$, en donde

$$f(x, y, z) = 10x - x^3 + y^2 - z - 5 = 0$$

$$g(x, y, z) = 10y + 0.5x^2 - y^2 - z - 3 = 0$$

$$h(x, y, z) = 10z - x - y^2 - z^3 - 6 = 0$$

(Notar que este sistema de funciones ya lo tienen de la tarea 3)

```
function F=ht5(x)
```

```

F=(10*x(1)-x(1)^3+x(2)^2-x(3)-5)^2+(10*x(2)+.5*x(1)^2-x(2)^2-x(3)-3)^2
+(10*x(3)-x(1)-x(2)^2-x(3)^3-6)^2;

```

Notar que en el archivo .m tienen que escribir toda la función en un sólo renglón, aquí yo lo separe para que se viera toda la función en la hoja de pdf. Análogo para el caso del gradiente.

```
function F=ht5g(x)
```

```

F=[-(2*(10*x(1)-x(1)^3+x(2)^2-x(3)-5)*(10-3*x(1)^2)+2*(10*x(2)+.5*x(1)^2

```

```

-x(2)^2-x(3)-3)*x(1)-20*x(3)+2*x(1)+2*x(2)^2+2*x(3)^3+12),...
-(4*(10*x(1)-x(1)^3+x(2)^2-x(3)-5)*x(2)+2*(10*x(2)+1/2*x(1)^2-x(2)^2-x(3)
-3)*(10-2*x(2))-4*(10*x(3)-x(1)-x(2)^2-x(3)^3-6)*x(2)),...
-(-20*x(1)+2*x(1)^3+4*x(3)+16-20*x(2)-x(1)^2+2*(10*x(3)-x(1)-x(2)^2-x(3)^3
-6)*(10-3*x(3)^2))]' ;

```

```
>> fmin('ht5','ht5g',[1 1 1],.001,100)
```

```

0 1 1 1
1.0000 0.4922 0.2188 1.0391 -35.2289
2.0000 0.7207 0.5112 0.7044 -8.0298
3.0000 0.5171 0.2872 0.7863 -2.1909
4.0000 0.6327 0.4273 0.6634 -0.8314
5.0000 0.5376 0.3234 0.7466 -0.4016
6.0000 0.6081 0.4009 0.6796 -0.2467
7.0000 0.5534 0.3423 0.7290 -0.1328
8.0000 0.5944 0.3867 0.6904 -0.0803
9.0000 0.5629 0.3529 0.7192 -0.0447
10.0000 0.5867 0.3786 0.6970 -0.0266
11.0000 0.5685 0.3591 0.7138 -0.0151
12.0000 0.5823 0.3740 0.7009 -0.0089
13.0000 0.5717 0.3626 0.7106 -0.0051
14.0000 0.5798 0.3713 0.7032 -0.0030
15.0000 0.5736 0.3647 0.7088 -0.0017

```

las iteraciones han convergido

ans =

```
0.5783 0.3697 0.7045
```

Notemos que es el mismo resultado que teníamos en el último ejercicio de la tarea 3.

3. **Encontrar los valores de x que minimizan la función $f(x) = -x_1x_2x_3$ empezando en el punto $x^0 = [10; 10; 10]$ y sujeto a las restricciones $0 \leq x_1 + 2x_2 + 2x_3 \leq 72$**

Primero creamos un archivo .m con nuestra función objetivo:

```

function f=myfun(x)
f=-x(1)*x(2)*x(3);

```

Luego escribimos las restricciones como menores o iguales a una constante:

$$-x_1 - 2x_2 - 2x_3 \leq 0$$

$$x_1 + 2x_2 + 2x_3 \leq 72$$

Dado que las dos restricciones son lineales, formularlas como la desigualdad matricial $Ax \leq b$ donde

```
>> A=[-1 -2 -2;1 2 2]
```

```
A =
```

```
-1 -2 -2
```

```
1 2 2
```

```
>> b=[0;72]
```

```
b =
```

```
0
```

```
72
```

Luego seleccionamos un *guess* inicial:

```
>> x0 = [10; 10; 10];
```

Entonces nuestro resultado será:

```
>> [x,fval] = fmincon('myfun',x0,A,b)
```

```
Warning: Large-scale (trust region) method does not currently solve this  
type of problem,
```

```
switching to medium-scale (line search).
```

```
> In fmincon at 260
```

```
Optimization terminated: Magnitude of directional derivative in search  
direction less than 2*options.TolFun and maximum constraint violation  
is less than options.TolCon.
```

```
Active inequalities (to within options.TolCon = 1e-006):
```

```
lower upper ineqlin ineqnonlin
```

```
2
```

```
x =
```

```
24.0000
```

```
12.0000
```

```
12.0000
```

```
fval =
```

```
-3.4560e+003
```