Ingeniero Industrial
Asignatura: *Optimización y Simulación*
Examen de Prácticas (A). Convocatoria Junio 2009

NOMBRE Y APELLIDOS:

1. **(1 Pto)** Resuelve, usando el Toolbox de Optimización de MatLab, el siguiente problema de programación matemática:

$$\text{Maximizar} \qquad f(x_1, x_2, x_3) = 2x_1 + 2x_2 + 4x_3$$

sujeto a

$$\begin{cases} 2x_1 + x_2 + x_3 \leq 20 \\ x_1 + x_2 + 5x_3 \leq 32 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

Se han de entregar tanto los resultados como los programas `.m` utilizados para resolver el problema.

`Solución:`

```
f = [-2 -2 -4];
A = [2 1 1; 1 1 5];
b = [20; 32];
lb = [0; 0; 0];
[x,fval,exitflag] = linprog(f,A,b,[],[],lb)
```

$x = [0.0000,\ 17.0000,\ 3.0000]$
`fval` $= -46.0000$
`exitflag` $= 1$

2. **(1 Pto)** Resuelve, usando el Toolbox de Optimización de MatLab, el siguiente problema de programación matemática:

$$\text{Maximizar} \qquad f(x_1, x_2, x_3, x_4) = x_1^2 + x_2^2 + x_3^2$$

sujeto a

$$\begin{cases} x_2 x_4 & \leq 1000 \\ x_2 - \frac{x_2 x_4}{10} & \leq 100 \\ \frac{3x_2 x_4}{40} & \leq 200 \\ \frac{x_1 x_3}{10} + \frac{2x_2 x_4}{25} & \leq 300 \\ 1 \leq x_1 \\ 1 \leq x_2 \\ 0 \leq x_3 \leq 10 \\ 0 \leq x_4 \leq 10 \end{cases}$$

Se han de entregar tanto los resultados como los programas `.m` utilizados para resolver el problema. ¿Te parece razonable el resultado obtenido? ¿Por qué?.

```
function f = coste_eje2(x)

f = -x(1)^2 - x(2)^2 - x(3)^2;
```

```
function [c, ceq] = reseje2(x)
c =[x(2)*x(4)-1000;
x(2) - x(2)*x(4)/10 - 100;
3*x(2)*x(4)/40 - 200;
x(1)*x(3)/10 + 2*x(2)*x(4)/25 - 300];
ceq=[];
```

```
A = [0 0 1 0; 0 0 0 1];
b = [-10; -10];
lb = [0;0;0;0];
x0 = [2;2;0;0];
[x,fval,exitflag,output,lambda,grad,hessian] = ...
fmincon(@coste_eje2,x0,A,b,[],[],lb,[],@restri_eje2)
```

$x = [1.0e+008 * 2.8936\ 0\ 0\ 0]$
`fval` $= -8.3732e+016$
`exitflag` $= 0$

Dado que el coste es una función convexa y el problema es de maximización, obviamente el problema no tiene solución: a medida que x_1 crece, manteniendo $x_2 = x_3 = x_4 = 0$, el coste también lo hace. El algoritmo empleado por el Toolbox de Optimización de MatLab excede el máximo número de iteraciones (`exitflag` $= 0$) y por tanto se detiene dando un valor muy grande para x_1.

3. **(1 Pto)** Elabora un código en MatLab, usando elementos finitos de Lagrange de grado 1, para resolver el siguiente problema :

$$\begin{cases} -u''(x) = 6x - 2 + 2\delta_{0.5}, & 0 < x < 1 \\ u(0) = u(1) = 0, \end{cases}$$

donde $\delta_{0.5}$ es la delta de Dirac centrada en 0.5. La solución exacta de este problema es

$$u(x) = 0.5 - |x - 0.5| + x^2(1 - x).$$

Utiliza dicho código para resolver el problema anterior para 10 elementos finitos, y calcula el error comparando la solución numérica con la exacta. Dibuja las gráficas de la solución exacta y de la numérica.

```
function [vector_x,vector_u]=elfin_modificado(b,c,f,a1,b1,alfa,beta,iopc,num_pas)
%*************************************
% Construcción de la malla 1-dimensional.
%*************************************
nver=num_pas+1; % numero de vertices
nel=num_pas; % numero de elementos
h=(b1-a1)/num_pas; % tamaño de la discretizacion.
i=[1:num_pas+1]; vector_x=a1+(i-1)*h;
%*********************************************
% Inicializacion de la matriz Ah
%*********************************************
Ah=spalloc(nver,nver,3*nver-2);
%*********************************************
% Inicializacion de bh
%*********************************************
bh=zeros(nver,1);
%*********************************************
% Calculo de las longitudes caracterisiticas de cada elemento
%*********************************************
long=diff(vector_x);
pmed=(vector_x(1:nver-1)+vector_x(2:nver))/2;
v_b=feval(b,pmed);v_c=feval(c,vector_x);
v_f=feval(f,pmed);
%*********************************************
% Bucle en elementos
%*********************************************
for k=1:nel
%*********************************************
% Calculo matriz elemental
%*********************************************
Ahk=(1/long(k))*[1, -1;-1, 1]+(v_b(k)/2)*[-1, 1; -1, 1] ...
+(long(k)/2)*[v_c(k), 0; 0, v_c(k+1)];
%*********************************************
% Ensamblado de la matriz
%*********************************************
Ah(k:k+1,k:k+1)=Ah(k:k+1,k:k+1)+Ahk;
%*********************************************
% Calculo 2º miembro elemental
%*********************************************
bhk=v_f(k)*long(k)/2*[1;1];
%*********************************************
% Ensamblado del segundo miembro
%*********************************************
bh(k:k+1)=bh(k:k+1)+bhk;
end
if(iopc==0)
```

```
%************************************************
% Bloqueo de la matriz
%************************************************
Ah(1,1)=1.e+30;Ah(nver,nver)=1.e+30;
%************************************************
% Bloqueo del segundo miembro
%************************************************
bh(1)=alfa*1.e+30;bh(nver)=beta*1.e+30;
else
%************************************************
% Modificacion del segundo miembro
%************************************************
bh(1)=bh(1)-alfa;bh(nver)=bh(nver)+beta;
end
bh(6) = bh(6) + 2;
%************************************************
% Resolucion del sistema
%************************************************
vector_u=(Ah\bh)';
```

```
function f = bexajunio(x)
f = zeros(size(x));
```

```
function f = fexajunio(x)
f = 6*x-2;
```

```
[x,u] = elfin_modificado('bexajunio','bexajunio','fexajunio',0,1,0,0,0,10);
exacta = '0.5 - abs(x-0.5) + x.^2-x.^3';
y = eval(exacta);
error1 = norm(abs(u-y),Inf)
plot(x,u,'--',x,y,'o')
```

error1 = $8.3267e-017$

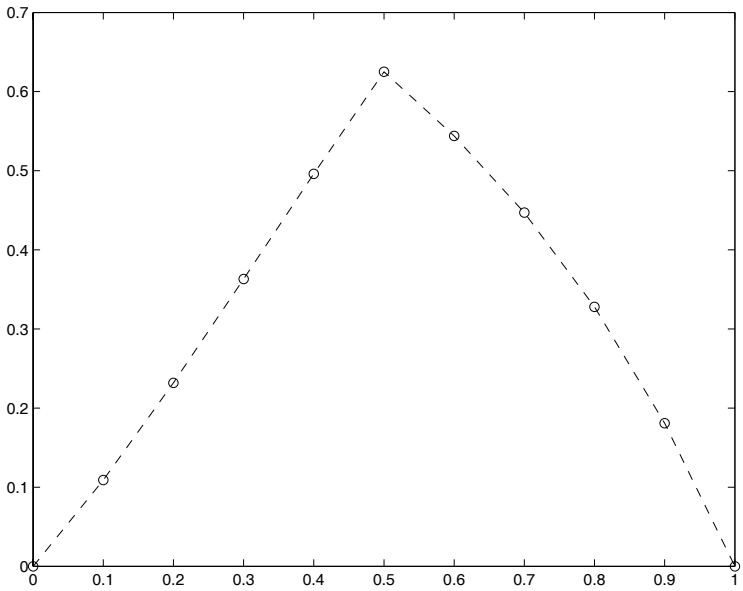

Figure 1: Gráficas de la solución numérica (- - -) y la exacta (o o o).