Ingeniero Industrial
Asignatura: *Optimización y Simulación*
Examen de Prácticas (A). Convocatoria Junio 2007

(i) Resuelve, usando el Toolbox de Optimización de MatLab, los siguientes problemas de programación matemática:

(a) (0.75 ptos)

$$\text{Minimizar} \quad f(x_1, x_2) = (x_1 - 5)^2 + (x_2 - 5)^2$$

sujeto a

$$\begin{aligned} & x_1^2 - x_2 \leq 6 \\ & x_1 + 3x_2 \leq 12 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{aligned}$$

Definimos la función coste:

```
function f = fcoste(x)
f = (x(1) - 5)^2 + (x(2) - 5)^2;
```

Definimos una función para la restricción no lineal:

```
function [c,ceq] = frestri(x)
c = [x(1)^2-x(2)-6];
ceq = [];
```

Finalmente, escribimos el programa principal:

```
A = [1 3];
b = [12];
x0 = [0;0]; %punto inicial de iteracion
lb = [0;0];
[x,fval,exitflag,outplut,lambda,grad,hessian] = ...
 fmincon(@fcoste,x0,A,b,[],[],lb,[],@frestri)
eig(hessian)
```

Se obtienen los siguientes resultados:

```
x=[3 3]
fval = 8
```

(b) (0.25 ptos)

$$\text{Minimizar} \quad -3x_1 - 5x_2$$

sujeto a

$$\begin{cases} x_1 + 2x_2 \leq 36 \\ 2x_1 + x_2 \leq 36 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases}$$

Se trata de un problema de programación lineal. Ejecutamos el código:

```
f = [-3 -5];
A = [1 2;2 1];
b = [36;36];
lb = [0;0];
[x,fval,exitflag,output,lambda] = ...
 linprog(f,A,b,[],[],lb)
```

y se obtienen los resultados

```
x=[12 12]
fval = -96
```

(ii) **(1 pto)** Consideremos el problema de Cálculo de Variaciones

$$\min_{u\in\mathcal{A}} \quad I(u) = \int_0^1 \left[\frac{1}{2}\left(u'(x)\right)^2 + \frac{1}{4}u(x)^2 + u(x)\,\delta_{0.2}(x)\right] dx$$

donde $\delta_{0.2}(x)$ es la delta de Dirac localizada en el punto 0.2 y

$$\mathcal{A} = \{u : [0,1] \to \mathbb{R} : u \text{ es derivable y } u(0) = u(1) = 0 \}\,.$$

La forma débil de la ecuación de Euler-Lagrange asociada a este problema viene dada por: encontrar $u \in \mathcal{A}$ tal que

$$\int_0^1 u'(x)\, v'(x)\, dx + \frac{1}{2}\int_0^1 u(x)\, v(x)\, dx = -v(0.2) \quad \text{para toda } v \in \mathcal{A}.$$

Elabora un código de elementos finitos en Matlab para resolver este problema. Dibuja la gráfica de la solución para el caso de 10 elementos. Se ha de entregar el código elaborado (o indicar claramente los cambios realizados en `delta10.m` ó `elfin.m`) y dibujar a mano la gráfica de la solución que obtengas con dicho código.

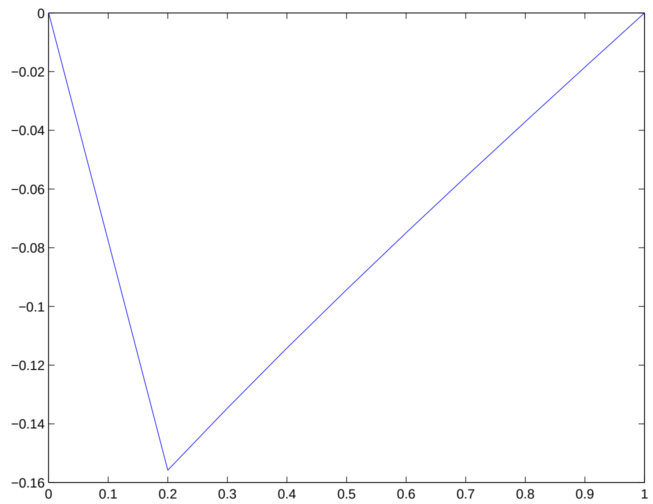
`Indicación:` se recomienda modificar el programa `delta10.m` y/o `elfin.m` de forma conveniente para adaptarlos a esta situación.

`Solución:` Elaboramos el código siguiente:

```
function [vector_x,vector_uh]=e2_os_junio_07
%****************************************************************
% Construcción de la malla 1-dimensional.
%****************************************************************
 nver=11; % numero de vertices
 nel=10; % numero de elementos
 h=0.1; % tamaño de la discretizacion.
 i=[1:11]; vector_x=(i-1)*h;
%***************************************************************
% Inicializacion de la matriz Ah
%***************************************************************
 Ah=spalloc(nver,nver,3*nver-2);
%***************************************************************
% Bucle en elementos
%***************************************************************
 for k=1:nel
 %*************************************************
 % Calculo matriz elemental
 %*************************************************
```

```
 Ahk=(1/h)*[1, -1;-1, 1]++(h/2)*[0.5, 0; 0, 0.5];
 %************************************************
 % Ensamblado de la matriz
 %************************************************
 Ah(k:k+1,k:k+1)=Ah(k:k+1,k:k+1)+Ahk;
 %************************************************
 end
%*************************************************
% Bloqueo de la matriz
%*************************************************
 Ah(1,1)=1.e+30;Ah(nver,nver)=1.e+30;
%**************************************************************
% Resolucion del sistema
%**************************************************************
bh=[0;0;-1;0;0;0;0;0;0;0;0];
vector_uh=(Ah_bh)';
```

Al ejecutar el código se obtiene la siguiente gráfica



(iii) **(1 pto)** Consideremos el problema de control óptimo

$$\text{Minimizar} \quad \int_0^T dt$$

sujeto a

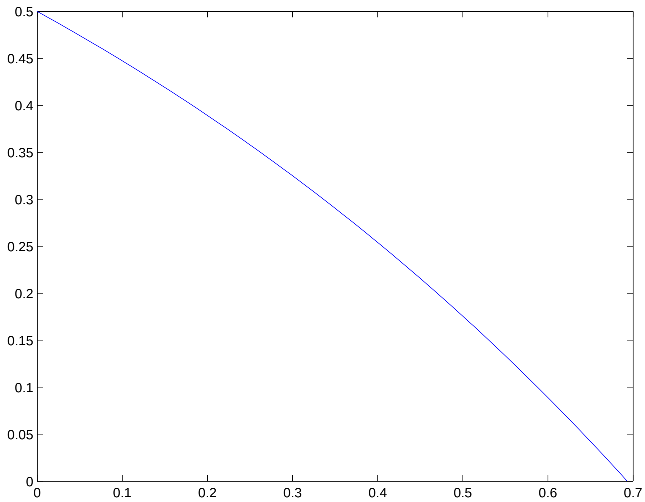
$$\begin{cases} x'(t) = x(t) - u(t) \\ x(0) = x^0 \\ x(T) = 0 \\ |u(t)| \leq 1 \quad \text{para todo } t > 0, \end{cases}$$

donde $x : [0,T] \to \mathbb{R}$ es la variable de estado y $u : [0,T] \to \mathbb{R}$ es el control. Para el caso $0 < x^0 < 1$, hemos visto en el examen de teoría que el control óptimo es $u(t) = +1$ para todo $0 < t < T$, con $T = \log\left(-\frac{u}{x^0-u}\right)$ el tiempo óptimo de control (log denota el logaritmo neperiano), y el estado óptimo es $x(t) = \left(x^0 - u\right)e^t + u$. Elabora un código en Matlab donde el usuario introduzca el estado inicial x^0 y el programa calcule el tiempo óptimo de control T y el estado óptimo. También ha de devolver una gráfica donde se dibuje el estado óptimo $x(t)$, $0 < t < T$. Se ha de considerar únicamente el caso $0 < x^0 < 1$ y se ha de responder a esta pregunta escribiendo el código y dibujando la gráfica del estado para el caso $x^0 = 0.5$.

Indicación: la función de Matlab para $\log(a)$ es `log(a)` y para e^a es `exp(a)`.

`Solución :` Elaboramos el código siguiente:

```
clear all;
x0 = input('Introduzca dato inicial:  ');
u = 1;
T = log(-u/(x0-u))
h = T/100;
t = 0:h:T;
for j=1:length(t)
 x(j)=(x0-u)*exp(t(j))+u;
end
plot(t,x)
```



y al ejecutarlo para `x0 = 0.5` se obtiene esta gráfica para el estado óptimo.