

Práctica 7: Programación con Matlab

1. Se pide:

a) Definir la función $f(x) = \begin{cases} x^2 - 1 & \text{si } -2 \leq x \leq 2 \\ \frac{1}{|x|} & \text{si } x < -2 \text{ ó } x > 2 \end{cases}$

b) Diseñar un programa que obtenga $f(x)$ si x es un escalar, pero si es un intervalo (vector de dos componentes) dibuje la función en ese intervalo; en caso contrario debe aparecer un aviso en pantalla.

c) Realizar la llamada al programa para:

1: Evaluar la función en los valores 1 y 8.

2: Obtener la gráfica de la función en el intervalo [-15,15].

d) Arreglar el apartado a) para que f se pueda aplicar sobre vectores y nos de las salidas de la aplicación de la función sobre cada componente.

% Apartado a.

Se crea el fichero .m:

```
function p=f(x);
```

```
if x>=-2 & x<=2
```

```
    p=x.^2-1;
```

```
else
```

```
    p=1./abs(x);
```

```
end
```

```
>> % Nota: el problema de esta definición es que es válida para evaluar en valores numéricos de x y en vectores donde todas las componentes se encuentran en el mismo intervalo de definición. En otro caso no evalúa bien:
```

```
>> f(0)
```

```
ans =
```

```
    -1
```

```
>> f(9)
```

```
ans =
```

```
    0.1111
```

```
>> f([0,1])
```

```
ans =
```

```
    -1     0
```

```
>> f([3,9])
```

```
ans =
```

```
    0.3333    0.1111
```

```
>> f([1,7])
```

```
ans =
```

```
1.0000 0.1429
```

```
>> % Ha evaluado los dos elementos como pertenecientes a la segunda definición de f. Se tendrá que arreglar esta situación (apartado d).
```

```
>> % Apartado b. Utilizamos la función f ya definida
```

```
function p=g(x);
```

```
if length(x)==1
```

```
p=f(x);
```

```
elseif length(x)==2
```

```
fplot('f',[min(x),max(x)])
```

```
else
```

```
error('La función no esta preparada para esto')
```

```
end
```

```
>>% Apartado c
```

```
>> g(1)
```

```
ans =
```

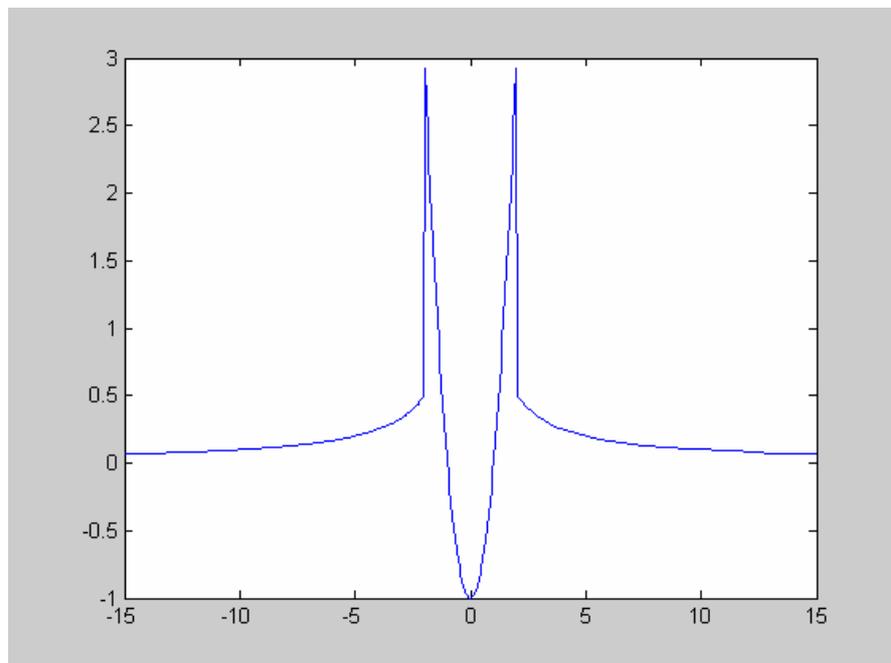
```
0
```

```
>> g(8)
```

```
ans =
```

```
0.1250
```

```
>> g([-15,15])
```



```

>> % Apartado d. Utilizamos la función f ya definida
function p=h(x);
for i=1:length(x);
    p(i)=f(x(i));
end
>> % Comprobamos con algún ejemplo:
>> h(0)
ans =
    -1
>> h(8)
ans =
    0.1250
>> h([0,8])
ans =
   -1.0000    0.1250
>>

```

- 2.** Construir un programa que calcule los cubos de los números naturales cuyo cuadrado sea menor que un número m (que se pide al usuario) y los introduzca en un vector v.

```

>> % Una posibilidad es crear el siguiente fichero .m:
m=input('introduce un numero entero ')
i=1;
while i^2<m
    i^3
    i=i+1;
end
% Este programa calcula los cubos de los números naturales cuyos cuadrados
% son menores que un numero m elegido por el usuario

```

- 3.** Construir una función r(a,b,c) que calcule las raíces de un ecuación de 2º grado ax^2+bx+c y que indique:
- “La ecuación no es de 2º grado” si $a=0$.
 - Que existen 2 raíces reales distintas.
 - Que existe una raíz real doble.
 - Que existen dos raíces complejas.

```

function p=raiz(a,b,c);
d=b^2-4*a*c;
disp(' La ecuación a resolver es...')
disp(['(',num2str(a),')x^2+',(num2str(b),')x+',(num2str(c),')'])

```

```

if a==0
disp('la ecuación no es de segundo grado')
else
if d==0
disp('la ecuacion tiene una raiz doble:')
p=-b/2*a;
elseif d>0
disp('la ecuacion tiene dos raices reales:')
p(1)=(-b+sqrt(d))/2*a;
p(2)=(-b-sqrt(d))/2*a;
else
disp('la ecuacion tiene dos raices complejas:')
p(1)=(-b+sqrt(d))/2*a;
p(2)=(-b-sqrt(d))/2*a;
end
end
end

```

4. Construir un programa que nos de los n primeros números de la sucesión $a(n)=n^2-4n+3$. Añadir una sentencia al programa que permita representar los términos en el plano.

% Existen múltiples posibilidades para realizar el problema. Proponemos crear la siguiente función:

```

function p=suce(n);
p=n.^2-4*n+3;
% Esta función evaluada sobre vectores de la forma 1:n da los términos de
% la sucesión indicada

```

% Para que pinte los términos como puntos (n, a_n) , utilizaremos otra forma de crear los términos:

```

n=input('Introduce el numero de términos a calcular, n= ')
for i=1:n
a(i)=i^2-4*i+3;
plot(i,a(i), '*')
hold on
end
a

```

5. Crear un programa que calcula la matriz A de elementos $a_{ij}=i^2+j$.

```

n=input('Introduce el numero de filas ');
m=input('Introduce el numero de columnas ');
a=[];
% crea una matriz a vacía para que no exista problemas con matrices ya
% definidas en la ventana de trabajo

```

```

for i=1:n
    for j=1:m
        a(i,j)=i^2+j;
    end
end
a

```

6. Dada una recta $ax+by+c=0$, crear un programa que nos diga si un punto (x,y) pertenece a dicha recta, está por encima o por debajo de ella.

```

recta=input('Introduce los coeficientes de la recta ax+by+c como [a,b,c] ');
x=input('coordenada x del punto a verificar ');
y=input('coordenada y del punto a verificar ');
p=recta(1)*x+recta(2)*y+recta(3);
if p==0
    disp('el punto pertenece a la recta');
elseif p<0
    disp('el punto esta por debajo de la recta ');
else
    disp('el punto esta por encima de la recta ');
end

```

7. Crear un programa que detecte si en un determinado intervalo $[a,b]$ existe una raíz de una función continua $f(x)$.

```

f=input('Introduce la función continua a estudio entre comillas y en la variable x ');
x=input(' Introduce el extremo inferior del intervalo a estudio ');
fa=eval(f,x);
x=input(' Introduce el extremo superior del intervalo a estudio ');
fb=eval(f,x);
if fa*fb<0
    disp('Existe raíz de la función en el intervalo')
elseif fa*fb>0
    disp('No se puede asegurar que exista raíz de la función continua en el intervalo');
else
    disp('algún extremo del intervalo es raíz de la función')
end

```

8. En un comercio se venden cajas de tornillos. Su precio depende del número de cajas comprados: hasta 100 cajas el precio de la caja es de 2 euros, desde 101 a 200 cajas el precio es 1,5 euros y, a partir de 201 cajas el precio es de 1

euro. Elaborar un programa que pregunte el número de cajas demandadas y que indique el precio de la unidad y el coste total del pedido.

```
p=input('Numero de cajas de tornillos demandadas ');
if p<= 100
    preciounidad=2;
    Total=preciounidad*p;
elseif 100<p & p<=200
    preciounidad=1.5;
    Total=preciounidad*p;
else
    preciounidad=1;
    Total=preciounidad*p;
end
preciounidad
Total
```

9. Realiza una tabla que incorpore el valor de los números enteros menores que 20, sus inversos, sus cuadrados y sus raíces cuadradas. Hacerlo primeramente con el comando while y luego con el comando for.

>> % Con el comando while:

```
i=0
while i<20
    i=i+1;
    num(i)=i
    inv(i)=1/i;
    cua(i)=i^2;
    raiz(i)=sqrt(i);
end
[num',inv',cua',raiz']
```

% Con el comando for:

```
num=[];inv=[];cua=[];raiz=[];
for l=1:20
    num(i)=i;
    inv(i)=1/i;
    cua(i)=i^2;
    raiz(i)=sqrt(i);
end
[num;inv;cua;raiz']
```