Ejemplos de Matlab (v. 211017)

Pablo González Nalda

Depto. de *Lenguajes y Sistemas Informáticos* lsi.vc.ehu.eus/pablogn

Universidad del País Vasco Euskal Herriko Unibertsitatea

GASTEIZKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE VITORIA-GASTEIZ

Enlaces

Ficheros en Matlab Online: https://drive.matlab.com/

Matlab Online: https://matlab.mathworks.com/

Ejemplos

Programas ejemplo de Matlab/Octave.

Precisión

En Matlab/Octave no se pueden representar infinitos reales entre dos reales. Por eso `eps` nos da el salto entre un número real y el siguiente distinto que se puede representar.

```
eps(e100)
eps(0)
```

Entrada/salida

Ejemplo de `fprintf`

```
clc
clear
res=2*pi;
fprintf('\ndatos\n-----\nPi vale %f y 2 pi vale %f\n',pi,res)
fprintf('\nUn entero con signo y alineado con espacios %+4d y con ceros a la izquierda %+04d\n',6,6)
```

Mostrar el valor de constantes matemáticas y cambiarlas:

```
clc     % limpiar salida
clear   % borrar variables anteriores
disp(pi);
disp(i);
i=8;
pi=17;
disp(pi);
disp(i);
clear
disp(pi);  % vuelven a los valores originales, los matemáticos
disp(i);
```

Ejemplo de un programa con entrada y salida de datos por teclado y pantalla usando `input` y `fprintf`

```
clc     % limpiar salida
clear   % borrar variables anteriores
x=input('Dame un número: ');
x2= x*2;
fprintf('El número que me has dado es %f y el doble es %f\n',x, x2);
```

Forma de salida alternativa de cadenas:

```
clc;
clear all;
close all;

M=1000;
TextoM=['Masa = ',num2str(M),' Kg'];
disp(TextoM);

fprintf ('Masa %f Kg \n' , M);
```

Se puede introducir un vector con `input` y se puede elevar al cuadrado:

```
clc
clear
n=input('Dame un número: ');
m= n.^2;
fprintf('El cuadrado de :');
display(n)
fprintf('es')
display(m);
```

`sprintf` crea cadenas.

```
clc
clear
x=fprintf('Hello %d\n',pi);
disp(x);
c=sprintf('Hola %d\n',pi);
disp(c);
```

Mostrar números complejos:

```
clc
clear
fprintf('La raíz es: ');
disp(sqrt(-1))
disp(i); % constante imaginaria

x = sqrt(-2) +3;
fprintf('%f   %+f  j\n', real(x), imag(x))
```

Cálculo de e^3 por Taylor:

```
clc
clear
n=3; % queremos e elevado a 3
numerador = n.^[1:100]; % n
denominador=factorial([1:100]);
res=1+sum(numerador./denominador);
fprintf('e^3 es %f y por Taylor es %f', exp(3), res);
```

Calcula π calculando cuántos puntos aleatorios caen en un círculo frente a los totales del cuadrado.

```
R=1;
Npuntos=100000000; % columnas
Datos=random('Uniform',-R, R, 2, Npuntos);

% Distancia al centro de todos los puntos (cada punto es una columna)
distancias=sqrt(Datos(1,:).^2+Datos(2,:).^2);

% Filtramos y nos quedamos todos los de dentro del círculo
DatosCirculo=Datos(:,distancias>R);
distancias(distancias>R)=[];
Ncirculo=length(distancias);

PiEstimado=4*Ncirculo/Npuntos % calculamos ?
```

Gráficas

Crear una gráfica:

```
clc
clear
close all
x=[0:0.5:2*pi];
y=sin(x);
plot(x,y,'r*')
```

Programa que separa los valores en los incrementos y decrementos.

```
m=readmatrix('track.csv');
h=m(:,3);
dh=diff(h);
a=dh;
a(a<0)=0;
d=dh;
d(dh>0)=0;
plot(cumsum(a));
hold on
plot(-cumsum(d));
```

Generar dos figuras distintas:

```
clc
clear
close all
x=randi(10,[1 25]);
bar(x);
figure
plot(x);
```

Ejemplo de creación de figura con subgráficas.

```
clc
clear
close all  % cierra ventanas de gráficas
nombreFichero=input('Dame el nombre del fichero: ');
n= readmatrix(nombreFichero);
urteak=n(:,1);  % años
nactot=n(:,2);
c3=n(:,3);
c4=n(:,4);

subplot(2,3,[1 3]);
plot(urteak,nactot,'rs-.');
hold on
plot(urteak,c3);
plot(urteak,c4);
title('Nacimientos')
xlabel('Años')

subplot(2,3,6);
plot(urteak,nactot,'go');
```

Cómo modificar sólo los valores de un vector si los valores que están en la misma posición de otro vector cumple una condición (valen 3 o 4).

```
clc
clear
m=readmatrix('studentMarks.csv');
clasper=m(:,5);
f= (clasper==3 | clasper==4);
nota=m(:,3);
nota(f)=nota(f)+44;
disp(nota)
```

Giro de puntos.

```
clc
clear
m = [1, 1, 2, 2; 1, 2, 1, 2];
plot(m(1,:) , m(2,:), 'b*')
axis([-3 3 -3 3])
a=pi/3;
r=[cos(a) -sin(a); sin(a) cos(a)];
p= r*m;
hold on
f=plot(p(1,:) , p(2,:), 'r*');
pause
for a=[0:pi/16:20*pi]
  r=[cos(a) -sin(a); sin(a) cos(a)];
  p= r*m;
  hold on
  f=plot(p(1,:) , p(2,:), 'r*');
  pause(0.2)
  delete(f)
end
```

Funciones

Función que devuelve dos valores:

```
function [rp,rn]=rcs(a)
% devuelve las dos raíces de la raíz cuadrada del número
  rp=+sqrt(a);
  rn=-sqrt(a);
end
```

Se puede pedir en un `input` los valores de un vector, o pedir escalares para pasarlos a una función.

```
clc
clear
coef=input('Dame los coeficientes: ');
a=coef(1); % alternativa a esto: a=input('Dame el coef. a: ');
b=coef(2);
c=coef(3);
[h,j]=funcion(a,b,c);
```

Programa que usa una función que convierte cm a pies y pulgadas.

```
clc
clear
h=input('Dame tu altura en cm');
[f,in]=conviertecmafin(h);
fprintf('Mides %f pies y %f pulgadas\n', f,in);
```

Función `conviertecmafin.m`

```
function [ pies,pulgadas ]=conviertecmafin(hcm)
  pies=floor(hcm/30.48); % podemos usar fix
  pulgadas= (hcm - pies*30.48) / 2.54;
end
```

Función que eleva al cuadrado, que funciona también con vectores por poner el punto antes del signo de elevar.

```
function r=cuadr(x)
% eleva al cuadrado el parámetro
% vector friendly f
  r=x.^2;
end
```

Esta función anterior se puede usar varias veces:

```
clc
clear
n1=input('Dame un número entero: ');
n2=input('Dame otro número, éste real: ');
c1=cuadr(n1);
c2=cuadr(n2);
fprintf('El cuadrado del primero ');
fprintf('es %d y el de segundo %.2f', c1,c2);
```

Programa que usa la función `isnumeric` de Matlab/Octave:

```
clc
clear
n=input('Dame un número: ');
if isnumeric(n)
  fprintf('El número es %f\n',n)
else
  fprintf('No me has escrito un número\n');
end
```

Programa que lee las notas del fichero y las faltas, calcula la nota media de dos columnas, y les suma 2 si una persona no tiene faltas.

Posteriormente elige las columnas 2, 3 y 4, y calcula el mínimo por filas. Si el mínimo por persona es menor a 3, asigna a esa persona esa nota mínima y al resto se les mantiene la media.

```
n=readmatrix('studentMarks.csv');
faltas=n(:,5);
m= (faltas==0);
notas= (n(:,2)+n(:,3)) / 2;
notas(m)=notas(m)+2;

t=n(:,[2:4]);
vm=min(t,[],2);
f=(vm <3);
notas(f)=vm(f);
```

A veces el objetivo de la función es imprimir en pantalla elaboradamente ciertos datos:

```
function imprime(cadena, v1)
  fprintf('%s %f\n', cadena,v1);
end
```

En un programa se usaría simplemente con `if esnegativo(5)` esta función `esnegativo.m` (o cualquier otra verificación):

```
function res=esnegativo(n)
  if n<0
    res = true;
  else
    res = false;
  end
end
```

Programa que muestra condicionales con rangos, usando `if elseif`:

```
clc
clear
precio=input("¿Cuánto te piden por ese piso? Dime: ");
mensaje=escaroelalquiler(precio);
fprintf('Mi consejo es que es %s \n',mensaje);
```

Función `escaroelalquiler.m`

```
function clasif=escaroelalquiler(alquiler)
  if alquiler < 400
    clasif="Es un zulo o estafa";
  elseif alquiler<600
    clasif="Baratísimo";
  elseif alquiler<800
    clasif="La clavada habitual";
  elseif alquiler < 900
    clasif="clavada grande";
  elseif alquiler < 1000
    clasif="Carísimo";
  else
    clasif="¿Te has equivocado?";
  end % end del if
end % end de la función
```

Programa que muestra un menú:

```
clc
clear
disp('1. opción 1')
disp('2. opción 2')
disp('3. opción Salir')
opc=input('Opción: ');
while opc~=3
  switch opc
    case 1
      disp('Has elegido 1');
    case 2
      disp('Has elegido 2');
    otherwise
      disp('error')
    end
  disp('1. opción 1')
  disp('2. opción 2')
  disp('3. opción Salir')
  opc=input('Opción: ');
end
```

Si no existe un fichero se queja, y si existe pide un vector de columnas y muestra una gráfica con las dos primeras.

```
clc
clear
close
f=input('Dame el nombre del fichero');
if exist(f,'file')~=2
  disp('Ese fichero no existe');
else
  c=input("Dime columnas");
  m=readmatrix(f,'Range',c);
  x=m(:,1);
  y=m(:,2);
  plot(x,y,'r*');
end
```

Programa que muestra que si nos dan un vector, el `if` puede tener problemas y debemos usar `any` para filtrar y saber si nos han dado algún negativo.

```
n=input('Dame un número: ');
v=input('Dame otro número: ');
if any(n<0) || any(v<0)
  fprintf('ALGUNO es negativo\n');
else
  fprintf('Es positivo todo\n');
end
```

Función que verifica si los vectores tienen la misma longitud antes de calcular e informar si es determinante es negativo.

```
function [iguales,im]=correctos(a,b,c)
  if length(a) == length(b) && length(b)==length(c)
    iguales = true; % son iguales
  else
    iguales = false; % son distintos
  end
  if iguales && any((b.^2-4*a.*c) <0)
    im = true;    % alguna raíz es imaginaria
  else
    im = false;   % raíces reales
  end
end
% Es lo mismo poner estas dos expresiones:
% if iguales
% if iguales == true
```

Programa que filtra valores negativos y aplica la función.

```
clc
clear
close
vr = [5.499 -5.498 5.5 5.5 5.52 5.51 5.5 5.48];
vh = [11.1 11.12 11.09 11.11 11.11 11.1 -11.08 11.11];
[v,m,e1,e2]=volcil(vr,vh);
media=mean(v);
dt=std(v);
plot(v,'r*');
disp('lista de radios incorrectos por posición: ');
disp(e1);
disp('lista de h incorrectas por posición: ');
disp(e2);
```

Función `volcil.m`

```
function [v,mensaje,ver,veh]=volcil(r,h)
  if any(r<0) || any(h<0)
    mensaje='Datos negativos';
    ver=find(r<0); % ind radios neg
    veh=find(h<0); % ind h neg
    r=r(r>=0);
    h=h(r>=0);
    r=r(h>=0);
    h=h(h>=0);
    v=pi*r.^2.*h;
  else
    v=pi*r.^2.*h;
    mensaje='todo bien';
    ver=[];
    veh=[];
  end
end
```

Ciclos, bucles, iterativas

Crear un vector vacío para ir luego llenándolo con valores generados:

```
clc
clear
v=zeros(20,1);
t=0;
c=1;
while t<=10
  v(c)=t;
  fprintf('he colocado t  %d en la posición c  %d de v\n',t,c)
   % actualizo para sig vuelta
  t=t+0.5;
  c=c+1;
end
disp(v);
```

Mostrar cadenas hasta que no se responda "s.º "S".

```
clc
clear
r='s';
while r=='s' || r=='S'
  nom=input('Dame el nombre del fichero: ');
  disp(nom); % t=readmatrix(nom);
  r=input('¿Quieres seguir preguntando sobre ficheros? ');
end
disp('Pasa un buen día.')
```

Formas de hacer ciclos con vectores de cadenas (sólo usar comillas dobles):

```
clc
clear

for saludo=["Hola","Kaixo","hello"]
  disp(saludo)
end

for ciudad=["Vitoria-Gasteiz","DSS","BI"] % sólo con " "
  f=sprintf("datos-%s.csv",ciudad);
  disp(f);
end

ciudades= ["Vitoria-Gasteiz","DSS","BI"];

for nf=ciudades
  f=sprintf("datos-%s.csv",ciudad);
  disp(f);
end

for n=1:length(ciudades)
  disp(ciudades(n));
end
```

Aplicación al uso de ficheros para crear gráficas:

```
clc
clear
close all
ciudades= ["Vitoria-Gasteiz","DSS","BI"];

for n=1:length(ciudades)
  f=sprintf("datos-%s.csv",ciudad(n));
  t=readmatrix(f);
  plot(t(:,1),t(:,2));
  ng=sprintf('graf-%d.png',n);
  saveas(gcf,ng);
  close
end
```

Asignar un valor a cada posición de la matriz bidimensional:

```
clc
clear
maxfilas=6;
maxcolumnas=4;
a=zeros(maxfilas,maxcolumnas);
for f=1:maxfilas
  for c=1:maxcolumnas
    a(f,c)=f+c;
  end
end
disp(a)
```

Ciclos anidados, con suma por columna:

```
clc
clear
for x=1:5
  suma=0;
  for y=1:3
    fprintf('x %d y %d \n',x,y)
    suma=suma+y;
  end
  fprintf('suma=%d----------\n',suma);
end
```

Hacemos 100 tiradas de dado y contabilizamos cuántas veces sale un 5 en la posición 5 del vector c:

```
c=zeros(1,6); % cuenta el número de tiradas de cada valor del dado
for n=1:100
t=randi([1 6],1,1); % tirada de dado del 1 al 6
c(t)=c(t)+1; % sumo 1 en la celda del valor de la tirada
end
tiradas=sum(c);
disp(c)
```

Pedir números mientras se escriba un negativo. Además cuenta el número de veces que se introducen negativos.

```
clc
clear
n=0;
num=input('Dame un número: ');
while num<0
  disp('Es negativo')
  n=n+1;
  num=input('Dame otro: ');
end
disp('Por fin me has dado un positivo');
fprintf('Me has dado el %f después de %d intentos fallidos.\n', num,n);
```

Imprimimos los números del 1 al 5 porque en el `while` se incrementa `n` antes de imprimir. Con las instrucciones en el orden contrario se imprimiría del 0 al 4.

```
clc
clear
n=0;
while (n<5)
  n=n+1;
  fprintf('%d ', n)
end
disp('Fin');
```

`for` hace tomar a la variable los valores del vector de forma consecutiva.

```
clc
clear
for x=[4 2 7 6 5]
  fprintf('otro valor es %d\n',x)
end
```

El uso de la variable `ii` en Matlab/Octave es habitual para no cambiar el valor de la constante imaginaria *i*.

```
clc
clear
n=input('Dame n ');
for ii=1:n %% si pongo i borra el valor de sqrt(-1) const imaginar
  fprintf('%d ',ii);
end
disp(i);
```

Búsqueda en un vector.

```
v=[3 4 5 8];
n=4; % quiero buscar este número en el vector
p=1;
while p<=length(v) && v(p)~=n
  p=p+1;
end
```

Dos formas de almacenar valores en un vector según se van introduciendo:

```
clc
clear
tam=5; % máx cantidad de elementos
v=zeros(1,tam); % vector de 5 ceros creado con anterioridad
c=1;
n=input('Dame un positivo: ');
while c<=tam && n>=0
  v(c)=n; % el número que me dan lo pongo en el vector
  c=c+1;
  n=input('Dame un positivo: ');
end

% Forma alternativa concatenando elementos a w
w=[];  %vector vacío
n=input('Dame un positivo: ');
while n>=0
  w=[w n]; % el número que me dan lo concateno al final del vector
  n=input('Dame un positivo: ');
end
```

Indicar las posiciones de inicio y de final de la secuencia descendente de un vector.

```
clc
clear
p=[4 5 7 10 9 8 7 30 22 11 9 8 7 6 3 5 7 9 11 22 33 44]; % entrada
d=diff(p);
disp(d); % los negativos indican que desciende
pos=1; % posición de cada elemento del for en la secuencia
pinicio=1; % posición del inicio de la secuencia actual
c=0; % longitud de la secuencia
m=0; % máxima longitud de secuencia descendente
for e=d
  if e < 0
    c=c+1;
    pfinal=pos;
  else
    if c>m
      m=c;                %salida
      posInicio=pinicio; %salida
      posFinal =pfinal;  %salida
    end
    c=0;
    pinicio=pos+1;
    pfinal=pos+1;
  end
  fprintf('pos %d \te %+03d  \tc %d  \t m %d pi %d pf %d\n', ...
      pos,e,c,m, pinicio,pfinal);
  pos=pos+1;
```

```
end
```

Procesamiento en lotes de ficheros

Pide los números de los ficheros del tipo `datos-4.csv` en una carpeta o directorio, y si existe grafica las dos primeras columnas y guarda la gráfica en un fichero.

```
clc
clear
v=input('Dame los números de fichero de hoy: ');
for n=v
%%%%%%for n=[8 4 3]  % alternativa al input
  nombreFichero=sprintf('carpetadatos/datos-%d.csv',n);  % creo el nombre
  fprintf('Procesando el fichero %s\n',nombreFichero);
  if exist(nombreFichero,'file')==2
    m=readmatrix(nombreFichero);
    x=m(:,1);
    y=m(:,2);

    plot(x,y);
    nombrefichgraf=sprintf('datos-%d.png',n);
    saveas(gcf,nombrefichgraf);  % get current figure
    close

  end % if
end % for
```

Se puede asignar la figura a una variable por si tenemos varias gráficas activas.

```
fig=figure;  % nombre para gráfica nueva
plot(x,y);
nombrefichgraf=sprintf('grafs/datos-%d.png',n);
saveas(fig,nombrefichgraf)
close fig
```

Nombres que varían en texto.

```
clc
clear
for nom=["monte", "valle"]  % parte del nombre de los ficheros
  nf=sprintf('carpetadatos/datos-%s.csv',nom);  % creo el nombre
  fprintf('Procesando el fichero %s\n',nf);
  m=readmatrix(nombreFichero);
   %%%% resto del programa
end  % for
```

Generación de nombres de fichero combinando números y cadenas.

```
for ps=["bajo","alto"]
  for n=1:4
    nf=sprintf('sensor-%s-num%02d.csv',ps,n);
    disp(nf)
  end
end
```

Integración

Programa que va integrando el movimiento en el tiempo de un objeto móvil cada medio segundo, almacena los valores de tiempos y velocidad y lo grafica.

```
clc
clear all
hold on

d=0;   % distancia
v=3;   % velocidad
s=0;     % segundo
maxseg=30;  % tiempo max
dt=0.5;    % actualizamos cada dt segundos
vd=[d];   % vector con distancias
vt=[0];   % vector con tiempos
while s<=maxseg
  plot([s],[d],'r*');
  vt=[vt s];  % Adjuntar un valor de s al vector vt por la derecha
  vd=[vd d];
  d=d+v*dt;
  s=s+dt;
  v=v*0.9;  % rozamiento
end
plot(vt, vd,'r')  % tiempo contra distancia
plot([0 30],[0 0],'k');  % eje cero en negro
```