APELLIDOS, NOMBRE: SERRANO ARRESE, JULIA

Adjuntar código utilizado, valores y gráficas pedidos

A. Crear vectores y matrices

4. Un vector x con elementos desde 0 a 2π (2*pi) a saltos de 0.01.

x = [0:0.01:(2*pi)];

5. Dar un comando para obtener el valor del último elemento de x, sin conocer el número de elementos de x.

las_el = x(length(x));

B. Crear vectores y matrices utilizando funciones básicas

2. Una matriz con 3 filas y seis columnas cuyos elementos valen 7.

x = zeros(3,6) + 7

C. Acceder a un dato o conjunto de datos de una matriz o vector

2. La primera y la tercera filas de A.

B = A([1,3],:)

3. La última columna de A.

B = A(:,end)

6. Escribir una sentencia para extraer de la matriz A la matriz C=[10 11;6 7].

C = [A(end,[2 3]);A(2,[2 3])]

7. Escribir el comando para obtener la matriz C=[12 11;8 7;4 3] a partir de A.

C = [A(end,[4 3]);A(2,[4 3]);A(1,[4 3])]

10. Sea el vector notas = 6+1.2*randn(1,70) que contienen las notas de 70 alumnos de la asignatura de Algorítmica Numérica:

a) Calcular la nota media (comando mean) y guardar en med el valor y la nota máxima (comando max) de los alumnos y guardar en mx.

```
%nota media
med = mean(notas);

%nota max
mx = max(notas);
```

b) Calcular la nota media de los alumnos aprobados (>=5)

```
media_aprobados = mean(notas(notas >=5))
```

c) Poner un 5 a aquellos alumnos con nota entre 4.5 y 5. En MATLAB se pueden combinar condiciones usando el operador lógico & (AND). Ver ejercicio anterior.

```
notas(notas>= 5.5 & notas<5) = 5;
```

D. Operaciones básicas con vectores y matrices.

1.c. Poner a cero los elementos de x con índice impar.

```
x(1:2:end) = 0
```

3.c. Formar una matriz 4x2 cuyas columnas sean los vectores x e y

```
A = zeros(4,2);
for i = 1:4
   A(i,1) = x(1,i);
   A(i,2) = y(1,i);
end
A;
```

E. Operaciones de álgebra lineal con vectores y matrices

4. Comprobad que la multiplicación matricial no es conmutativa (A*B =/ B*A).

```
if(isequal(A*B,B*A))
        fprintf('Multiplicacion es conmutativa\n')

else
        fprintf('Multiplicacion no es conmutativa\n')
end
```

F. Operaciones punto a punto con vectores

1. Crear los siguientes vectores x=[1 2 3 4]; y=[5 6 7 8];

d) Calcular un vector z conteniendo el recíproco (1/x) de cada elemento de x.

```
z = 1 ./ x
```

f) A partir del vector x, dar un comando para obtener el vector [1 4 27 256]

```
x .^ x
```

G. Gráficas en MATLAB

Vamos a calcular algunos valores aproximados del nº e considerando los primeros mil valores de la sucesión $\lim_{n\to\infty}(1+\frac{1}{n})^n = e$. Para ello:

1. Creamos el vector n de 1 a 1000.

n = [1:0.01:1000]

2. Creamos el vector vap de valores aproximados con los términos de la sucesión dada para n de 1 a 1000 $((1+\frac{1}{1})^1, (1+\frac{1}{2})^2, \ldots\ldots, (1+\frac{1}{1000})^{1000})$ ¿Qué valor toma el último término de la sucesión?

vap = (1 + (1 ./ n)) .^ n;

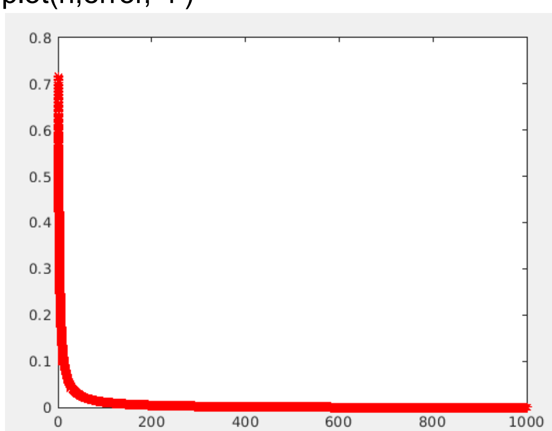
ultimo = vap(end) %ultimo término de la sucesión

El último término es: 2.7169

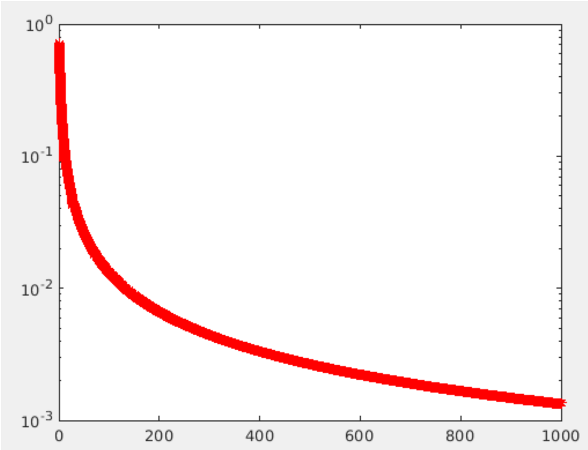
3. Creamos el vector, que llamaremos error, que contiene los errores (en valor absoluto) que se cometen al considerar los valores aproximados anteriores respecto del nº e (exp(1) en Matlab) restando el valor aproximado y el valor exacto: error=abs(vap-exp(1)). Pintamos la gráfica del error respecto de n con asteriscos rojos (plot(n,error,'*r') ¿El error es cada vez menor?
Prueba ahora a pintar la gráfica anterior en escala semilogarítmica en el eje y.

```
%error
error=abs(vap-exp(1));

%grafica
plot(n,error,'*r')
```



Los valores en la función se aproximan rápidamente a 0 y no se aprecia bien su comportamiento en la gráfica, para solucionar este problema utilizamos una escala logarítmica en el eje y (semilogy):



Con esta escala en el eje y, podemos apreciar mejor el error y como es cada vez menor según la n va creciendo.