Laboratorio 1: Introducción a Octave

A lo largo de estas sesiones de prácticas vamos a aprender a usar el programa Octave. Este es un programa que va más allá de lo que puede hacer una calculadora, y nos permitirá realizar no sólo la mayor parte de los cálculos que estáis viendo en clase, sino que os dará mucha libertad para diseñar vuestros propios métodos y algoritmos para resolver los cálculos matemáticos que os encontraréis a lo largo de la carrera. Usaremos octave para:

- realizar operaciones aritméticas,
- programar en un lenguaje interpretado,
- realizar gran variedad de gráficos,
- ...

Las operaciones se realizan normalmente sobre la ventana de comandos, que es la pantalla que aparece nada más iniciar el programa, a continuación de los símbolos >>. Cada vez que se ejecute una operación, hay que darle a la tecla de 'intro'. Si se tiene una idea clara de qué cálculos se van a realizar, es conveniente realizar un programa en la pestaña de 'Editor' (la tercera pestaña en la pantalla de inicio), grabar el trabajo en un archivo y luego ejecutarlo en la ventana de comandos. Así, si se ha cometido un fallo o se quiere hacer una mínima modificación, no hace falta empezar todo desde el principio.

Cuando se realizan las distintas operaciones, además, se podrán almacenar los resultados apelando a variables: así, si queremos guardar el valor resultante de multiplicar dos números, en lugar de introducir `x*y` sin más, lo podemos guardar bajo `q=x*y` y, siempre que no modifiquemos este valor de `q` podremos hacer uso de él mediante una simple llamada, como si fuese una constante más.

Ejemplo. Introducir en Octave las siguientes instrucciones:

- `q=8*5`
- `q^2`

Primeras operaciones

Las operaciones en Octave siguen las reglas usuales de la aritmética, y lo único que hay que tener en cuenta es el código específico para ejecutarlas. A continuación daremos cuenta de los operadores más usuales:

x+y	Suma de los números x e y.
x-y	Resta de los números x e y.
x*y	Producto de los números (o matrices, si las dimensiones lo permiten) x e y.
x.*y	Producto coordenada a coordenada de los vectores x e y.
x/y	Cociente de los números x e y.
x./y	Cociente coordenada a coordenada de los vectores x e y.
x ∧ y	Elevar x a y.
x. ∧ y	Elevar las coordenadas del vector x a las del vector y.
x'	traspuesto de la matriz x.

Introduzcamos las siguientes operaciones en la ventana de comandos de Octave:

- `5 - 3^2*1/4+8`
- `(5 - 3)^((2*1)/(4+8))`

Los formatos numéricos con los que se pueden trabajar en Octave (y a los que se puede cambiar usando el comando `format ...`) son los siguientes:

- `format short` es el formato por defecto; punto fijo y cuatro dígitos decimales.
- `format short e` usa notación científica con cuatro dígitos decimales (observemos que la coma decimal es el punto para Octave).
- `format long` punto fijo y catorce dígitos decimales.
- `format long e` notación científica con quince dígitos decimales.
- `format rat` aproximaciones al número racional más cercano.

También tenemos las constantes usuales: π (pi), e (exp(1)),...

Vectores y Matrices

Un vector es un conjunto de datos dispuestos uno detrás de otro a los que podemos acceder mediante un subíndice. Una matriz es un conjunto de datos dispuestos en un tablero a los que podemos acceder mediante dos subíndices, de tal forma que el primer subíndice indica la fila donde se encuentra el elemento y el segundo indica la columna.

En Octave definimos estos elementos de la siguiente forma: los datos se dispondrán entre corchetes, para indicar los valores de una fila éstos se separarán por espacio o por comas, y para pasar a una columna se usará el ; Así, tenemos los siguientes ejemplos:

$$\texttt{[1 2 -3pi ; 0 sqrt(2) -0.01 4]} = \begin{pmatrix} 1 & 2 & -3 & \pi \\ 0 & \sqrt{2} & -0'01 & 4 \end{pmatrix}$$

$$\texttt{[1 2 -3 pi 0 sqrt(2) -0.01 4]} = \begin{pmatrix} 1 & 2 & -3 & \pi & 0 & \sqrt{2} & -0'01 & 4 \end{pmatrix}$$

Ejercicios.

1. Definir en Octave las matrices y vectores

$$A = \begin{pmatrix} -1 & \sqrt{5} & 2 & \pi \\ 0 & 3 & -1 & 4 \\ \pi & \sqrt[3]{5} & 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 3 & \frac{2}{5} & 0 \\ 0 & -1 & 4 \\ \pi & 0 & 1 \end{pmatrix}, v = \begin{pmatrix} 4 \\ -3 \\ 2 \end{pmatrix}, w = \begin{pmatrix} \sqrt{7} \\ 0 \\ 2 \\ \pi \end{pmatrix},$$

2. Realizar todos los productos matriz-matriz y matriz-vector que las dimensiones admitan.

Sucesiones

Una sucesión es una colección ordenada, finita o no, de números. Desde cierto punto de vista, no son más que casos particulares de vectores. A continuación veremos algunas formas rápidas de definir una sucesión de números, si sabemos qué patrón siguen:

- `n:m` creará un vector cuyo primer elemento es n y donde cada coordenada se calcula sumando una unidad a la coordenada anterior, hasta acabar en m (es decir, creará la sucesión n n+1 n+2 . . . m). Obviamente, n tiene que ser menor que m. Este método viene bien para poder apelar a las distintas coordenadas de un vector o una matriz.

- `n:q:m` creará un vector cuyo primer elemento es n, su último elemento es m y que va a paso q (es decir, creará la sucesión n n+q n+2q . . . m). Si m no se alcanza a base de sumar q unidades, se parará hasta superar m por primera vez. Este método viene bien para considerar una división de un intervalo o un dominio deseado con la finura que se quiera.

- `linspace n m r` creará un vector cuyo primer elemento es n, su último elemento es m y que consta de r elementos.

Ejercicio. Consideramos la sucesión 2, 5, 8, 11, 14, 17, 20, 23, 26. Definir esta sucesión de tres formas distintas y guardarlas bajo sendas variables `v1` `v2` y `v3`.

Funciones Matemáticas

Octave incluye una serie de funciones matemáticas y trigonométricas que nos ayudan a simplificar algunos cálculos, a continuación mostramos algunas de ellas.

Función	**Descripción**
sqrt(x)	Raíz cuadrada de x
abs(x)	Valor absoluto de x
log(x)	Logaritmo neperiano de x
log2(x)	Logaritmo en base a 2 de x
log10(x)	Logaritmo en base a 10 de x
exp(x)	constante *e* elevada a x
pow2(x)	Para cada elemento de x, calcula 2x
rem(x,y)	Resto entre la división de x e y
round(x)	Redondeo de x al entero mas cercano
ceil(x)	Redondeo al entero superior de x
floor(x)	Redondeo al entero inferior de x
fix(x)	Redondeo hacia el entero mas cercano a cero
gcd(x,y)	Calcula el máximo común divisor entre x e y
lcm(x,y)	Calcula el mínimo común múltiplo entre x e y
sin(x) cos(x) tan(x) sec(x) csc(x) ctg(x)	Funciones trigonométricas ordinarias para x
asin(x) acos(x) atan(x) asec(x) acsc(x) actg(x)	Funciones trigonométricas inversas para x
sign(x)	Devuelve 1 para los elementos positivos y 0 para los negativos
v(n)	Devuelve el n-ésimo elemento del vector x. (recorrido por columnas)
A(n,m)	Devuelve el elemento que se encuentra en la fila n y columna m.
v(n:m)	Devuelve el conjunto de elementos partiendo desde el n-ésimo elemento hasta el m-ésimo elemento. (recorrido por columnas)
A(n,:)	devuelve todos los elementos de la fila n
A(:,m)	devuelve todos los elementos de la columna m
A(x,y)	devuelve la submatriz cuyas filas son las indicadas por el vector x y sus columnas son las indicadas por el vector y
rows(x)	devuelve el número de filas del vector x
columns(x)	devuelve el número de columnas del vector x
length(x)	devuelve la longitud de la matriz (columnas)
size(x)	devuelve el numero de filas y el número de columnas
eye(n,m)	Crea una matriz n×m con diagonal 1, si se invoca con un solo escalar crea una matriz identidad cuadrada de n elementos

diag(x,k)	Si x es un vector, la función crea una matriz cuya diagonal serán los elementos del vector x la cual empezará la k-ésima columna de la matriz creada. Si x es una matriz creará un vector columna con la diagonal que empieza en la k-ésima columna de la matriz x. Si k es positivo devuelve la super-diagonal, si es negativo devuelve la sub-diagonal. Si se invoca con un solo escalar el valor k por defecto es 0 y devuelve la super-diagonal principal.
zeros(n,m)	Crea una matriz n×m con sus elementos iguales a 0, si se invoca con un solo escalar crea una matriz cuadrada de n elementos
ones(n,m)	Crea una matriz n×m con sus elementos iguales a 1, si se invoca con un solo escalar crea una matriz cuadrada de n elementos
linspace(p,q,n)	Crea una matriz de n elementos espaciados uniformemente desde p hasta q
rank(A)	Devuelve el rango de la matriz A
inv(A)	Calcula la inversa de la matriz cuadrada A
det(A)	Calcula el determinante de la matriz cuadrada A
trace(A)	Suma los elementos de la diagonal principal de la matriz A (traza)
sum(x)	Devuelve la suma de los elementos del vector x o la suma de los elementos de cada columna de la matriz x
prod(x)	Devuelve el producto de los elementos del vector x o el producto de los elementos de cada columna de la matriz x
max(x)	Devuelve el máximo para cada columna de la matriz x o el máximo de los elementos del vector x
min(x)	Devuelve el mínimo elemento del vector x o el mínimo elemento para cada columna de la matriz x
sort(x)	Ordena de menor a mayor los elementos del vector x los elementos de cada columna de la matriz x

Ejercicios.

1. Resolver el sistema, planteándolo de forma matricial.

$$\begin{aligned} 3x + 2y \qquad\qquad &= 0 \\ 2x \qquad - 2z + t &= 1 \\ y + 4z - 3t &= -2 \\ x + 5y - z - 3t &= 4 \end{aligned}$$

2. Dados los vectores

$$x = \begin{pmatrix} 1 \\ 2 \\ -3 \\ 5 \end{pmatrix}, \; y = \begin{pmatrix} 4 \\ -4 \\ 1 \\ 2 \end{pmatrix}, \; z = \begin{pmatrix} 3 \\ 1 \\ 2 \\ -4 \end{pmatrix},$$

   formar una matriz cuyas dos primeras columnas sean todo ceros, sus siguientes tres columnas sean los vectores x, y y z y sus últimas columnas sean todo unos.

3. En la matriz del apartado anterior, acceder a la submatriz formada por las filas 1, 3 y 4 y a las columnas 1, 2 y 6. ¿Es esta matriz invertible?

Funciones definidas por el usuario

En Octave podemos crear nuestras propias funciones, podemos escribirlas directamente desde la línea de comandos o ejecutarlas desde un archivo externo. Es recomendable crearlas de todas formas en la pestaña de 'Editor'.

Las funciones que creamos desde la línea de comandos deben cumplir con el siguiente formato:

```
function variable_salida = nombre_funcion (argumentos_entrada)
cuerpo _funcion
endfunction
```

En caso de devolver varias variables, éstas deben estar encerradas entre corchetes ?[ ]?:

```
function [salida1,salida2] = nombre_funcion (argumentos_entrada)
cuerpo _funcion
endfunction
```

Por ejemplo, crearemos una función que calcula el seno(x) en grados:

```
function s = sind(x)
\%SIND(X) Calcula seno(x) en grados
s = sin(x*pi/180);
endfunction
```

Y ejecutamos,

```
sind(45)
ans = 0.70711
sind(90)
ans = 1
```

Para que Octave ejecute un archivo éste debe tener extensión .m y debe encontrarse en el directorio desde donde estemos ejecutando Octave.

Ejercicios.

1. Crear un programa cuyos parámetros de entrada sean una matriz invertible, A y un vector, b, y que devuelva la solución del sistema $Ax = b$.

2. Crear un programa cuyos parámetros de entrada sean una matriz, un vector con tantas coordenadas como fila de la matriz y un número no mayor del número de columnas de la matriz, y que devuelva la matriz sustituyendo la columna indicada por el número por el vector.

3. Crear un programa cuyos parámetros sean una matriz y un valor y el programa compruebe si el número es un valor propio.

4. Crear un programa cuyos parámetros de entrada sean dos vectores y una matriz cuadrada de la misma dimensión y un número y que devuelva la matriz formada de añadir el primer vector como columna al final de la matriz, el segundo vector como última fila, y el número cerrando la última posición.

$$f(x, y, A, n) = \left(\begin{array}{c|c} A & x^T \\ \hline y & n \end{array} \right)$$