

ANÁLISIS NUMÉRICO I – 2014

Guía básica de Octave

Recomendaciones:

- No usar espacios en los nombres de los archivos o carpetas

Abrir el Octave:

- Abrir una terminal: **Crtl+Alt+T** o Aplicaciones....
- Ir al directorio donde están guardados los archivos: `~$ cd Documentos/Numerico_I`
- Abrir un editor de texto: `~$ gedit &`
- Abrir el Octave: `~$ octave`

Vectores:

- Un vector de ceros de dimensión n `octave:#>v = zeros(n,1).`
- Escribir un vector fila v : `octave:#>v = [1 2.56 -3.43 pi].`
- Escribir un vector columna v : `octave:#>v = [1 ; 2.56 ; -3.43 ; pi].`
- Hacer una partición uniforme $\{x_i\}_{i=1}^n$ del intervalo $[0, 1]$:

Por ejemplo, para $n = 5$

```
octave:2> x = linspace(0,1,5)
x =
```

```
0.00000    0.25000    0.50000    0.75000    1.00000
```

- Hacer una partición uniforme $\{x_i\}_{i=1}^n$ del intervalo $[0, 1]$, tal que $x_i - x_{i-1} = 0,25$:

```
octave:3> x = 0:0.25:1
x =
```

```
0.00000    0.25000    0.50000    0.75000    1.00000
```

- Hacer un vector aleatorio de dimensión n `octave:#>c = rand(n,1).`

Matrices

- La matriz cero $M \in \mathbb{R}^{m \times n}$ `octave:#>M = zeros(m,n).`
- La matriz identidad $I \in \mathbb{R}^{n \times n}$ `octave:#>M = eye(n).`
- La matriz diagonal $D \in \mathbb{R}^{n \times n}$ `octave:#>D = diag(rand(n,1)).`

- Hacer una matriz $M \in \mathbb{R}^{2 \times 2}$ octave:#>M = [1 2 ; 3 4] .
- Hacer una matriz aleatoria $M \in \mathbb{R}^{m \times n}$ octave:#>M = rand(m,n).
- La estructura if ... end La estructura del if simple es la siguiente:

```
if condicion
    Acciones a realizar si es cierta la condicin
else
    Acciones a realizar si es falsa la condicin
end
```

Hagamos una función que se verifica si un número n es divisible m

```
function multiplo(n,m)

    if mod(n,m) == 0
        disp('Es divisible')
    else
        disp('No es divisible')
    end

end
```

Las operaciones lógicas son:

| | | | |
|----|-----------------|----|-----------------|
| <= | menor o igual a | < | menor que |
| > | mayor que | >= | mayor o igual a |
| == | igual a | ~= | distinto |

- El bucle for ... end

En el bucle for ... end, la ejecución de uno o varios comandos se repite un número fijo y predeterminado de veces.

```
for variable_contador = INICIO : PASO : FIN
    Accion_1
    ...
    Accion_M
end
```

Por ejemplo, si se quiere construir una matriz $A \in \mathbb{R}^{m \times n}$, triangular superior, tal que $(a_{i,j}) = 1/(i+j)$ para $i \leq j$, se debe hacer

```
function A = matriz_vander(m,n,alpha)

    A = zeros(m,n) ;

    for i = 1:m
        for j = i:n
            A(i,j) = 1/(i+j) ;
        end
    end

end
```

- El bucle `while ... end`

Este bucle se usa cuando no se conoce el número de veces que debe repetirse la ejecución de cierto comando. Supongamos que queremos sumar números aleatorios hasta superar una tolerancia `Tol` y queremos saber cuantas veces iteramos; se debe hacer:

```
function [s,contador] = suma_aleatorio(Tol)

    s = 0 ;
    contador = 0 ;

    while s <= Tol
        s = s + randn ;
        contador = contador + 1 ;
    end

end
```