1 Bucles

En esta práctica mostraremos la representación numérica en un ordenador y los errores que lleva aparejada. También repasaremos el manejo de las funciones y condicionales en MATLAB e introduciremos los bucles o ciclos.

1.1 Representación numérica

La representación de valores en el ordenador no siempre es exacta. En ocasiones esto se debe a que sólo se pueden almacenar un número finito de dígitos, y el valor es irracional o periódico. En otros casos se debe a las limitaciones de la representación binaria utilizada por el ordenador. Se debe indicar que MATLAB guarda los valores con casi 16 cifras decimales, si bien sólo muestra 4 en sus resultados. Veamos como se pueden mostrar los valores de diferentes maneras

Representar $\sqrt{2}$	`» sqrt(2)`
Idem con “formato largo”	`» format long; ans`
Idem con formato racional	`» format rat; ans`
Idem con 30 cifras decimales	`» vpa(ans,30)`
Representación binaria del valor 1234	`» dec2bin(1234)`
Representación decimal del valor 1234 en base 6	`» base2dec('1234',6)`

Como resultado de los errores de representación, se pierden propiedades matemáticas básicas como la asociatividad de la suma. Utilizaremos el valor `eps`.

Si a=ε/2: 1+a-1 ¿ = ? 1-1+a

```
» a=eps/2,1+a-1, 1-1+a
```

El orden altera la suma

```
» x=rand(1,1000);y=sort(x);z=y(end:-1:1);
» vpa(sum(x),20),vpa(sum(y),20), vpa(sum(z),20)
```

Ejemplo con serie geométrica

```
» format long;
» a=0;b=-10;
» x=logspace(a,b,10000);
» sx=sum(x);
» y=x(end:-1:1); sy=sum(y);
» r=x(2)/x(1);se=(x(end)*r-x(1))/(r-1);
```

1.2 Bucles simples

Permite repetir un número determinado de veces un conjunto de instrucciones. Su sintaxis es la siguiente:

for var = *vector*

Instrucciones que deben ejecutarse

end

El argumento *vector* puede ser efectivamente un vector, en cuyo caso la variable va tomando los valores de las componentes del vector, o una estructura de la forma *inicio : incremento : fin*, en cuyo caso la variable va tomando valores desde *inicio* hasta *fin* con un determinado *incremento*. Si no se indica el valor del incremento, este se toma como unidad. El número de veces que se repite el bucle viene dado por la dimensión del vector o por la expresión $max\left(0, \frac{fin - inicio}{incremento} + 1\right)$.

La ejecución del bucle se interrumpe en cualquier momento mediante la instrucción **break**.

Veamos algunos ejemplos, como pueden ser el cálculo de una serie o del factorial.

$\sum_{i=1}^{10} i^2$

```
» serie=0;for i=1:n; serie=serie+i*i;end
» serie=1:10;serie=serie*serie'  % alternative sin bucles
```

10!

```
» factorial=1; for i=2:10;factorial=factorial*i; end
```

Se pide:

1. Crear el programa horner[1] que evalúa el valor del polinomio $P(x)=p_1x^{n-1} + p_2x^{n-2} + \ldots + p_{n-1}x + p_n$ en el punto x mediante el algoritmo de Horner (Regla de Rufini). Los coeficientes del polinomio se almacenan en un vector $p = [p_1, p_2, \ldots, p_{n-1}, p_n]$.

   Entrada: vector $p = [p_1, p_2, \ldots, p_{n-1}, p_n]$ y punto x en que se evalúa

   Salida: q, valor de P(x)

   Paso 1: Verificar que p es un vector (1 fila o 1 columna)

   Paso 2: Obtener el número de elementos de p

   Paso 3: Asignar $q= p_1$

   Paso 4: Para k desde 2 hasta n repetir paso 5

   Paso 5: $q = (q * x) + p_k$

Con estos datos, la primera línea del archivo debe ser: `function q = horner(p,x).` En el paso 1 y paso 2 se debe emplear la función `size(p)`, además de los condicionales vistos en la práctica anterior. Comparar los resultados con los obtenidos al usar `polyval(p,x)`

2. Crear `function y=armonica(n)` que evalúa $\sum_{k=1}^{n} \frac{1}{k}$.

---

[1] Los pasos para entrar en el editor y escribir la función son: Archivo→New→M-File.

La sintaxis de la función es **function** [variables_salida]= nombre_función (variables_entrada)

1.3 Bucles condicionales

Permite repetir un conjunto de instrucciones, en tanto se satisfaga una condición lógica. Su sintaxis es la siguiente:

while *condición*

Instrucciones que deben ejecutarse mientras la condición sea cierta.

end

El siguiente ejemplo muestra el uso de esta instrucción para sumar términos de la serie armónica mientras su diferencia sea menor que *error*.

```
function y=armonica2(error)
y=0; k=1;
while (1/k)>error;y=y+1/k;k=k+1;end
```

Esta instrucción puede recrearse combinando un bucle *for* y un condicional *if*.

```
function y=armonica3(error)
y=0;
for k=1:10000;
y=y+1/k; if (1/k)<error;break;end
end
```

Se pide:

3. Modificar la función armónica2 anterior para obtener también el número de términos empleados.

4. Crear una función que calcule la norma *p* de un vector, definida mediante $\|v\|_p = \sqrt[p]{\sum_{k=1}^{n} |v_i|}$.

   Comparar los resultados con los obtenidos mediante la función norm.

2 Anexo: Instrucciones necesarias

2.1.1 Variable especiales

En MATLAb existen variables de uso común, cuyo valor viene ya preasignado.

pi	3'1415926535897...	NaN	Indeterminación (Not a Number, por ejemplo 0/0)
i ó j	Unidad imaginaria (raíz cuadrada de-1)	realmin	El menor número real positivo utilizable
inf	Infinito, por ejemplo 1/0	realmax	El mayor número real positivo utilizable
eps	Menor valor positivo que sumado a la unidad tiene representación diferente a 1.	ans	Variable creada automáticamente para representar el último resultado

2.1.2 Gráficos bidimensionales (2-D)

plot(X)	Representa los puntos (k,X_k). Si X es una matriz, hace lo mismo para cada columna de la matriz. Si X es un vector complejo, representa Real(X) frente a IMAG(X)..
plot(X,Y)	Representa el conjunto de puntos (X,Y). Si X o Y son matrices, representa por filas o columnas los datos de X frente a los datos de Y, dependiendo si el otro vector es fila o columna. Para valores complejos de X e Y, se ignoran las partes imaginaria.
grid	Sitúa rejillas en los ejes de un gráfico 2-D o ·3-D. La opción grid on coloca las rejillas y greed off las elimina. La opción grid permuta entre on y off
hold	Permite mantener el gráfico existente con todas sus propiedades, de modo que el siguiente gráfico que se realice se sitúe sobre los mismos ejes y se superponga al existente. La opción hold on activa la opción y hold off la elimina. La opción hold permuta entre on y off. Válido para 2-D y 3-D
subplot(m,n,p)	Divide la ventana gráfica en mxn subventanas y coloca el gráfico corriente en la ventana p-ésima, empezando a contar por la parte superior izquierda y de izquierda a derecha hasta acabar la línea, para pasar a la siguiente
plot(X,Y,S)	Gráfica de plot(X,Y) con la opciones definidas en S. Usualmente, S se compone de dos caracteres entre comillas simples, el primero de los cuales fija el color de la línea del gráfico, y el segundo fija el carácter a usar en el graficado. Los valores posibles de colores y caracteres son, respectivamente, los siguientes:

y	amarillo	g	verde	.	puntos	-	sólido
m	magenta	b	azul	o	círculos	:	a puntos
c	cyan	w	blanco	x	x-marcas	-.	guiones y puntos
r	rojo	k	negro	+	signo más	--	semisólidos
				*	estrellas		

2.1.3 Titulos, etiquetas, mallas y textos

title('texto')	Añade el texto como título del gráfico en la parte superior del mismo	text(x,y,'texto')	Sitúa el texto en el punto (x,y)
xlabel('texto')	Sitúa el texto al lado del eje x	ylabel('texto')	Sitúa el texto al lado del eje y
gtex('texto')	Permite situar el texto en un punto seleccionado con el ratón	ginput(n)	Permite recuperar las coordenadas de n puntos de un grafico mediante ratón