Práctica 1: Introducción a *Mathematica*

Métodos Numéricos I - Doble Grado en Ingeniería Informática y Matemáticas - Universidad de Granada - Curso 2016_2017

Acceso a *Mathematica*

Los pasos a dar con el ordenador para poder trabajar con Mathematica son:

1. Encender el ordenador y entrar en “acceso general”.
2. En el menú Inicio seleccionamos la opción Programas->Mathematica3.0.

De esta forma el ordenador cargará el programa y ya estará preparado para que podamos trabajar con él.

Celdas de entrada y salida

Al cargar el programa *Mathematica* se abre un fichero en blanco (llamado Untitled - 1). En su interior iremos escribiendo todas las operaciones algebraicas que deseemos realizar. Por ejemplo, podemos pedir a *Mathematica* que calcule el resultado de la sencilla operación 1+1:

```
1 + 1
```

que proporciona el valor 2 como resultado.

Para ejecutar una operación en Mathematica pulsaremos la tecla Intro (teclado numérico) **o simultáneamente las teclas Shift y Enter.**

Toda la información de cada archivo se organiza según una serie de celdas. Cada una de ellas aparece con un corchete en su parte derecha. Fundamentalmente, trabajaremos con dos tipos de celda:

1. **Celdas de entrada (In)**, en las que se introducen datos (en el ejemplo anterior “1+1” es la celda de entrada);
2. **Celdas de salida (Out)**, en las que aparecen los resultados de los cálculos efectuados por el programa (en el ejemplo anterior “2” es la celda de salida).

Se puede hacer referencia e incluso operar con la salida que acaba de obtenerse en forma de %, o bien a una concreta mediante %n, donde n es el número de la salida.
Un primer comando de *Mathematica,* que recoge en esencia la filosofía de la sintaxis de todos los de este programa, es **Print**, cuyo funcionamiento ilustran estos ejemplos:

```
Print[2 * 3]
```

```
Print["Buenos días"]
```

En particular, todas las sentencias utilizan corchetes para escribir sus argumentos.

Operaciones algebraicas básicas

Las operaciones básicas (suma, resta, producto y cociente) se expresan con los símbolos usuales: `+ - * /`

```
2 - 3
```

```
5 * 2
```

```
8 / 4
```

La exponenciación se realiza con el símbolo ^:

```
2^4
```

2^4

La prioridad entre operaciones es la usual, y puede modificarse mediante el uso de paréntesis.

La raíz cuadrada se calcula con el comando **Sqrt**. Por ejemplo, para calcular $\sqrt{4}$ escribiremos

```
Sqrt [4]
```

En el menú File -> Palettes aparece la opción Basic Input, con la que podemos obtener un amplio grupo de símbolos matemáticos. Por ejemplo, podemos calcular la raíz cuadrada:

$\sqrt{4}$

Asignación de valores a variables

Para asignar un valor a una variable se utiliza el signo "="

```
x = 3
```

```
y = 2
```

```
x + y
```

Para eliminar el valor asignado a una variable se utiliza la orden **Clear**

```
Clear[x]
```

```
x
```

La sentencia **Clear["Global`*"]** se utiliza para eliminar los valores que tengan asignados todas las variables que hayan sido utilizadas previamente.

Redondeo decimal de un número

El comando **N** se utiliza para obtener el redondeo decimal con seis cifras significativas de un número real expresado en base decimal. Por ejemplo:

```
N[π]
```

El siguiente ejemplo proporciona el redondeo del número real $e^4 + \sqrt{\pi}$:

```
N[E^4 + Sqrt[Pi]]
```

El comando **N** admite un argumento más, que se utiliza para obtener el redondeo con cierto número de cifras significativas. Por ejemplo, para obtener el redondeo de π con 20 cifras significativas escribimos el comando

```
N[π, 20]
```

Observa qué ocurre cuando la parte entera es cero:

```
N[Cos[Pi / 5], 5]
```

Funciones matemáticas

Algunas funciones (argumentos de trigonométricas en radianes)

$\texttt{Sin}\left[\frac{\pi}{3}\right]$

$\texttt{Cos}\left[\frac{\pi}{2}\right]$

$\texttt{Tan}\left[\frac{\pi}{4}\right]$

```
ArcSin[1]
```

```
ArcCos[1 / 2]
```

```
ArcTan[1]
```

```
Exp[2]
```

```
Log[4.]
```

```
Abs[-4]
```

```
Abs[2 + I]
```

Definición de funciones, derivadas, integrales y gráficas

Para definir una función mediante una expresión concreta se escribe la variable (o variables) entre corchetes junto con un guión bajo a la derecha, además de dos puntos antes del signo de igualdad. Por ejemplo,

```
f[x_] := 2 x^3 - E^x + 5.
```

Podemos, por ejemplo, evaluar dicha función en un punto,

```
f[2]
```

o incluso derivarla, mediante el comando ' (tecla de ?),

```
f'[x]
```

o también con el comando **D**:

```
D[f[x], {x, 2}]
```

Igualmente podemos integrarla mediante la sentencia **Integrate**:

```
Integrate[f[x], {x, -1, 1}]
```

Este comando admite su variante numérica **NIntegrate**

```
NIntegrate[f[x], {x, -1, 1}]
```

Obsérvese el efecto sobre **Integrate** en el siguiente cambio de argumentos:

```
Integrate[f[x], x]
```

También podemos dibujar su gráfica gracias al comando **Plot**:

```
Plot[f[x], {x, 0, 1}]
```

En ocasiones, las funciones están definidas a trozos, en cuyo caso es de utilidad la sentencia **Which**. Por ejemplo,

```
g[x_] := Which[0 ≤ x ≤ 1, x^3, 1 < x ≤ 10, Log[x]]
```

Entonces

```
g[0.5]
```

o

```
g[E]
```

Al igual que comentamos con la asignación de valores a variables, el comando **Clear** permite eliminar la definición de las funciones definidas previamente.

Vectores

Los vectores se introducen en *Mathematica* mediante llaves y comas. Por ejemplo, para el vector v=(1,2,3,4) escribimos

```
v = {1, 2, 3, 4}
```

La coordenada i-ésima de un vector v se calcula mediante la expresión **v[[i]]**. Así, la segunda coordenada del vector v anterior se obtiene como

```
v[[2]]
```

El número de coordenadas de un vector se halla a partir de la sentencia **Length**:

```
Length[v]
```

El producto escalar usual (euclídeo) de dos vectores de $\mathbb{R}^N$ se calcula con un simple punto:

```
v.{-1, 0, 2, -1}
```

Una forma cómoda de introducir un vector cuando cada coordenada depende explícitamente de su posición es mediante el comando **Table**. Para entender su funcionamiento, observa los ejemplos siguientes:

```
Table[i^2, {i, 10}]
```

```
Table[i^2, {i, 7}]
```

```
Table[i / 3., {i, 10}]
```

El valor máximo de las coordenadas de un vector *v* puede determinarse a partir de la sentencia **Max**. Para el vector *v* anterior

```
Max[v]
```

Señalemos además que disponemos de dos comandos,**Sum** y **Product**, que permiten sumar o multiplicar las coordenadas de un vector, o incluso algo más general. Observa los ejemplos

```
Sum[v[[i]], {i, 4}]
```

```
Product[v[[i]], {i, 4}]
```

```
Sum[i, {i, 100}]
```

```
Product[i / (i + 1), {i, 5}]
```

Matrices

Para introducir una matriz en el ordenador utilizando Mathematica, escribiremos, entre dos llaves, y por filas, todos los elementos de la matriz. Las filas se representan a su vez entre llaves y tanto las distintas filas como los distintos elementos de una misma fila se separan por comas. Por ejemplo, para introducir la matriz $a=\begin{pmatrix} 2 & 7 & -1 \\ \frac{1}{2} & 2 & 0 \end{pmatrix}$ escribiremos

```
a = {{2, 7, -1}, {1 / 2, 2, 0}}
```

Mathematica dispone de una opción para visualizar en pantalla la expresión matricial de *a*:

```
MatrixForm[a]
```

El coeficiente de la matriz *a* que ocupa el lugar (*i*,*j*) se calcula mediante la expresión **a[[i,j]]**. Para la matriz anterior,

```
a[[2, 3]]
```

La fila *i*-ésima de *a* se obtiene con **a[[i]]**. Por ejemplo, continuando con la matriz previa,

```
a[[1]]
```

El orden de una matriz también se obtiene fácilmente con el comando **Dimensions**:

```
Dimensions[a]
```

Se pueden introducir de forma más cómoda algunos tipos de matrices: **DiagonalMatrix** para matrices diagonales. Así, la matriz $a=\begin{pmatrix} 4 & 0 & 0 \\ 0 & 1.1 & 0 \\ 0 & 0 & 3 \end{pmatrix}$ se introduce

```
DiagonalMatrix[{4, 1.1, 3}]
```

Aún más; en el caso de la matriz identidad de un orden concreto, disponemos de la sentencia **IdentityMatrix**:

```
IdentityMatrix[6]
```

Al igual que ocurre con los vectores, cuando los coeficientes de una matriz dependen explícitamente de su posición, el mismo comando **Table**, con una adecuada modificación, permite generar la matriz:

```
Table[i * j, {i, 2}, {j, 3}]
```

También podemos hallar el valor máximo de los coeficientes de una matriz *a* mediante la sentencia **Max**. Con la matriz *a* anterior,

```
Max[a]
```

Los comandos **Sum** y **Product** admiten también su versión matricial:

```
Sum[a[[i, j]], {i, 2}, {j, 3}]

Product[a[[i, j]], {i, 2}, {j, 3}]

Sum[i * j, {i, 10}, {j, 3}]

Product[0.0123 (i - 20 j) (i + j), {i, 2}, {j, 4}]
```

La siguiente sentencia **RowReduce** da como salida la única matriz escalonada reducida equivalente (por filas) a la dada:

```
RowReduce[a]
```

Operaciones con matrices

Para determinar la suma o la diferencia de dos matrices *a* y *b* de órdenes adecuados, o el producto de un número real λ por la matriz *a*, basta escribir *a*+*b*, *a*-*b* y λa, respectivamente. Por ejemplo, si $a=\begin{pmatrix} 1 & 2 & 1 \\ 2 & 3 & 4 \\ 5 & 6 & 7 \end{pmatrix}$ y $b=\begin{pmatrix} 1 & 3 & 5 \\ 1 & 0 & 5 \\ -2 & 2 & 1 \end{pmatrix}$, entonces:

```
a = {{1, 2, 1}, {2, 3, 4}, {5, 6, 7}}

b = {{1, 3, 5}, {1, 0, 5}, {-2, 2, 1}}

a + b

a - b

4 a

4 * a
```

El producto de dos matrices *a* y *b* de órdenes compatibles se halla mediante un punto, ., como ilustran las matrices *a* y *b* anteriores:

```
a.b
```

Sin embargo, el símbolo * no calcula el producto usual de matrices, sino el producto coeficiente a coeficiente, que no tiene interés en esta asignatura:

```
a * b
```

La potencia (natural) de una matriz se puede hallar mediante el uso retetido del producto, pero ello no suele ser operativo, por lo que se dispone del comando **MatrixPower** para calcularlo:

```
MatrixPower[a, 4]
```

Para una matriz cuadrada, la sentencia **Det** calcula su determinante y, si la matriz es regular, **Inverse** da su inversa:

```
Det[a]
```

```
Inverse[a]
```

También podemos trasponer con facilidad una matriz cualquiera:

```
a = {{1, 2, 3, 4}, {-5, -6, -7, -8}, {1, 1, 1, 1}}
```

```
Transpose[a]
```

Conviene destacar que no se puede operar con el comando **MatrixForm**. Basta probar con

```
MatrixForm[a] + MatrixForm[b]
```

Cálculo de valores y vectores propios

Las sentencias **Eigenvectors** y **Eigenvalues** proporcionan, respectivamente, los valores propios de una matriz cuadrada y, si es diagonalizable, las correspondientes bases de vectores propios:

```
a = {{1 / 2, 2 / 3, 3 / 4}, {0, 1, 2}, {0, 1, 1}}
```

```
Eigenvalues[a]
```

```
Eigenvectors[a]
```

Rudimentos de programación

Nos ocupamos finalmente de analizar algunas de las herramientas de programación que incluye *Mathematica* y que nos serán de uitildad a lo largo de la asignatura. En la definición de una función a trozos ya hemos introducido una de ellas, **Which**. Se trata de de una sentencia condicional. Otra bastante versáti es **If**. Cuenta contres argumentos, aunque el último es prescindible. La sintaxis específica es **If[condición,V,F]**, que ejecuta V si la condición es cierta y F en caso contrario. Si no se incluye el último argumento, simplemente ejecuta V cuando la condición es cierta:

```
a = {{1, 2}, {3, Pi}}
```

```
If[Det[a] == 0, Print["singular"], Print["regular"]]
```

Observa que cuando se usa el símbolo de igualdad en una condición, debe escribirse dos veces.

Otra estructura esencial en programación es el bucle. En *Mathematica* podemos hacer uso, por ejemplo, del comando **For**, cuya sintaxis responde a la estructura **For[comienzo,condición,incremento,cuerpo]**. Los tres primeros argumentos determinan el rango de variación de la variable de control: su comienzo o inicialización, su incremento y su incremento o paso. La última, en cambio, determina la acción a ejecutar en función de la variable de control. Veamos un ejemplo:

```
For[i = 1, i ≤ 7, i++, Print[i^2 - 5]]
```

La expresión i++ puede reemplazarse por i=i+1, e indica que el incremento es 1. Para un incremento distinto, digamos de 3, se usaría i=i+3, aunque no es esencial cambiar el incremento i++ (¿por qué?).

Ejemplos

Teorema de la serie geométrica

Considera la matriz cuadrada *a* de orden 3x3 cuya entrada (*i*,*j*) es la media artimética de *i* y *j* dividida por $\sqrt{1234.5}$:

```
a = Table[(i + j) / Sqrt[1234.5], {i, 3}, {j, 3}]
```

Calculemos su radio espectral:

```
Eigenvalues[a]
```

```
Abs[%]
```

```
Max[%]
```

Es menor que 1, por lo que el teorema de la serie geométrica garantiza que la matriz *i*-*a* es regular, siendo *i* la matriz identidad de orden 3x3, y que su inversa coincide con la suma de la serie geométrica generada por *a*. El siguiente código, que calcula la suma parcial truncada en la potencia 100-ésima de a, aproxima dicha suma y consecuentemente la inversa de (*i*-*a*):

```
s = IdentityMatrix[3]
```

```
For[i = 1, i ≤ 100, i++, s = s + MatrixPower[a, i]]
```

```
s
```

Así pues, la diferencia entre la solución exacta y la aproximación es

```
Inverse[IdentityMatrix[3] - a] - s
```

Nótese que, de forma alternativa, el comando **Sum** permite hallar la suma parcial anterior:

```
IdentityMatrix[3] + Sum[MatrixPower[a, i], {i, 100}]
```

Norma del máximo de una matriz cuadrada

Para determinar la norma del máximo de una matriz cuadrada, introducimos la matriz. Por ejemplo,

```
a = Table[N[(i - j) / (i + j)], {i, 4}, {j, 4}]
```

```
n = Length[a[[1]]]
```

```
Max[Table[Sum[Abs[a[[i, j]]], {j, n}], {i, n}]]
```

También puede hacerse de forma compacta, mediante un módulo:

```
normainf[b_] :=
 Module[{n}, n = Length[b[[1]]]; Max[Table[Sum[Abs[b[[i, j]]], {j, n}], {i, n}]]]
```

Testémoslo con la matriz anterior:

```
normainf[a]
```

Ejercicios

- Define la función f: [0,1] $\longrightarrow\mathbb{R}$ como f(x)=$\begin{cases} 2x - \log(x), & \text{si } 0.3 \le x \le 0.5 \\ \frac{2}{x} + |x - 0.6|, & \text{si } 0.5 \le x \le 0.8 \\ 0, & \text{en otro caso} \end{cases}$. Calcula su integral y dibuja su gráfica.
- Halla el radio espectral de la matriz de orden 4x4 cuyo coeficiente (i,j) es $\sqrt{|2.3\,i - 2.4\,j|}$.
- Calcula la suma de los cubos de los 23 primeros números naturales mediante un adecuado bucle. Comprueba que el resultado obtenido es correcto mediante el comando **Sum**.
- Halla el producto de los inversos de los números naturales comprendidos entre 5 y 21 haciendo uso de un bucle, y compruba la validez de tu respuesta con la sentencia **Product**.
- Determina el término x_{39} de la sucesión de Fibonacci de forma recurrente y a partir de la expresión explícita de dicho término.
- Calcula los errores absolutos y relativos que se cometen cuando en el primer ejemplo se toma como aproximación de $(i - a)^{-1}$ la suma $\sum_{i=1}^{100} a^i$.
- Programa un módulo que calcule la norma 1 de una matriz cuadrada cualquiera y aplícalo a la matriz $\begin{pmatrix} 0.9 & 1.1 & 3.01 \\ 1 & -3.2 & -2.9 \\ -3/2 & 2/5 & 1.1 \end{pmatrix}$.
- Diseña un programa que, a partir de una matriz cuadrada, genere como salida su condicionamiento (norma infinito) si es regular, o el mensaje “la matriz no es regular” en caso contrario. Úsalo con las matrices $\begin{pmatrix} -1 & -4 & -7 \\ 1 & 5/2 & 4 \\ 3 & 6 & 9 \end{pmatrix}$ y $\begin{pmatrix} 1 & 1.1 & 0 & -2 \\ 1/7 & 2/9 & 1 & 1 \\ 0.1 & -0.4 & 0.87 & 3.14 \\ 2.71 & 0.81 & 0.82 & -2.38 \end{pmatrix}$.

- Calcula la norma euclídea de la matriz $\begin{pmatrix} 1.1 & 2.2 & 2.1 & 0.9 \\ e/6 & \pi/7 & 0.01 & -0.82 \end{pmatrix}$.

- Programa el algoritmo propuesto en clase teoría para aproximar la sucesión de iteradores. Aplícalo a la matriz *A*=-*a*, siendo *a* la matriz del ejemplo, y=(1,2,-1/4,1.1,0,0.32) y obtén 10 iteradores. Comprueba que se cumplen las acotaciones que aparecen en la relación de problemas del Tema 1.

- Calcula, aplicando el comando **N**, el valor de $\sqrt{5+10^{-n}} - \sqrt{5}$, para *n=1,...,20.* Repite los cálculos, reescribiendo la expresión $\sqrt{5+10^{-n}} - \sqrt{5}$ como $\frac{10^{-n}}{\sqrt{5+10^{-n}}+\sqrt{5}}$. Interpreta los resultados.