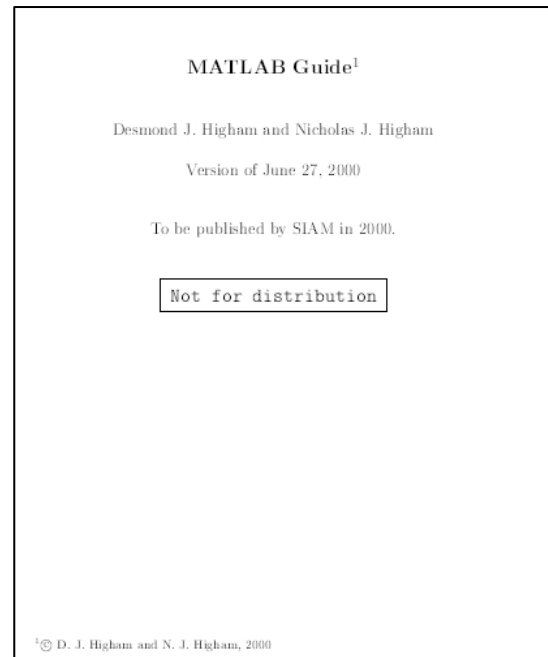
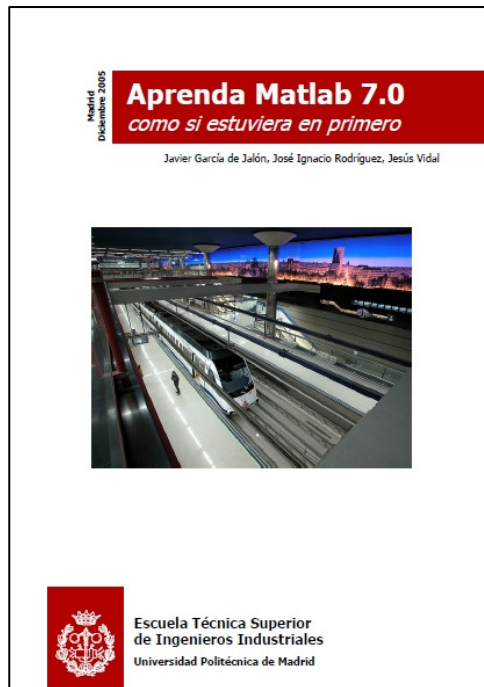




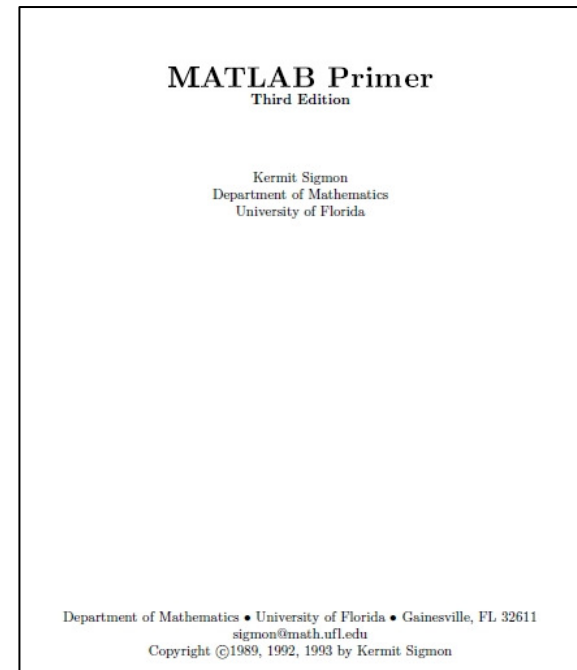
Cómputo Científico I - CO2111 – Sec 3

Bibliografía:

- Curso introductorio de MATLAB. Rene Escalante, Edt. Equinoccio, USB 2006
- Matlab Primer. Kermit Sigmon. Third Edition.
- Matlab Guide. Desmond J. Higham, Nicholas J. Higham. SIAM 2000.
- Aprenda Matlab 7.0 Como si estuviera en primero. Javier García Jalon, José Ignacio Rodríguez, Jesús Vidal, Universidad Politécnica de Madrid 2005



Profs. Buitrago y Jimenez





¿Qué es MATLAB?

MATLAB es un lenguaje de alto rendimiento para computación de carácter técnico. Integra computación, visualización y programación en un entorno de fácil uso donde los problemas y las soluciones están expresados en una notación matemática familiar.

Los usos típicos incluyen:

- Matemáticas y computación
- Desarrollo de algoritmos
- Modelado, simulación y prototipos
- Análisis de datos, exploración y visualización
- Gráficas científicas y de ingeniería
- Desarrollo de aplicaciones, incluyendo la construcción de la interfaz gráfica de usuario



¿Qué es MATLAB?

MATLAB es una abreviación de MATrix LABoratory.

Originalmente fue escrito para proveer fácil acceso al software de matrices desarrollado por los proyectos LINPACK y EISPACK.

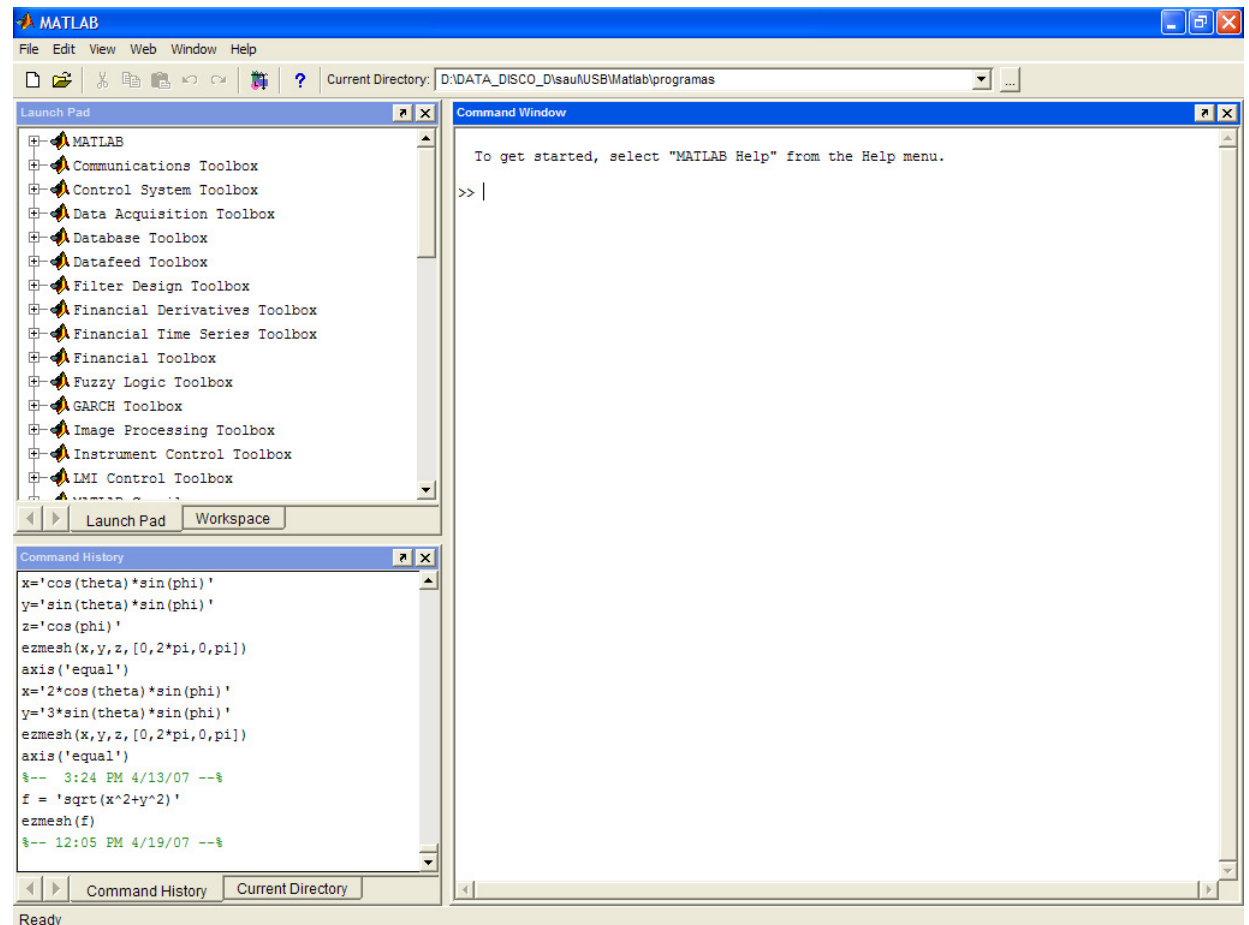
Hoy en día, MATLAB usa software desarrollado por los proyectos LAPACK y ARPACK que juntos representan lo más avanzado en software para cómputo de matrices.

El elemento básico de MATLAB es un arreglo que no requiere ser dimensionado. Esto permite resolver muchos problemas técnicos de cómputo, especialmente aquellos que involucran matrices o vectores, en una fracción del tiempo que tomaría escribir un programa en un lenguaje escalar no interactivo como C o Fortran.

El escritorio de MATLAB

MATLAB en la versión 6 se presenta como un escritorio que contiene múltiples ventanas. Como muestra la figura, principalmente hay tres ventanas:

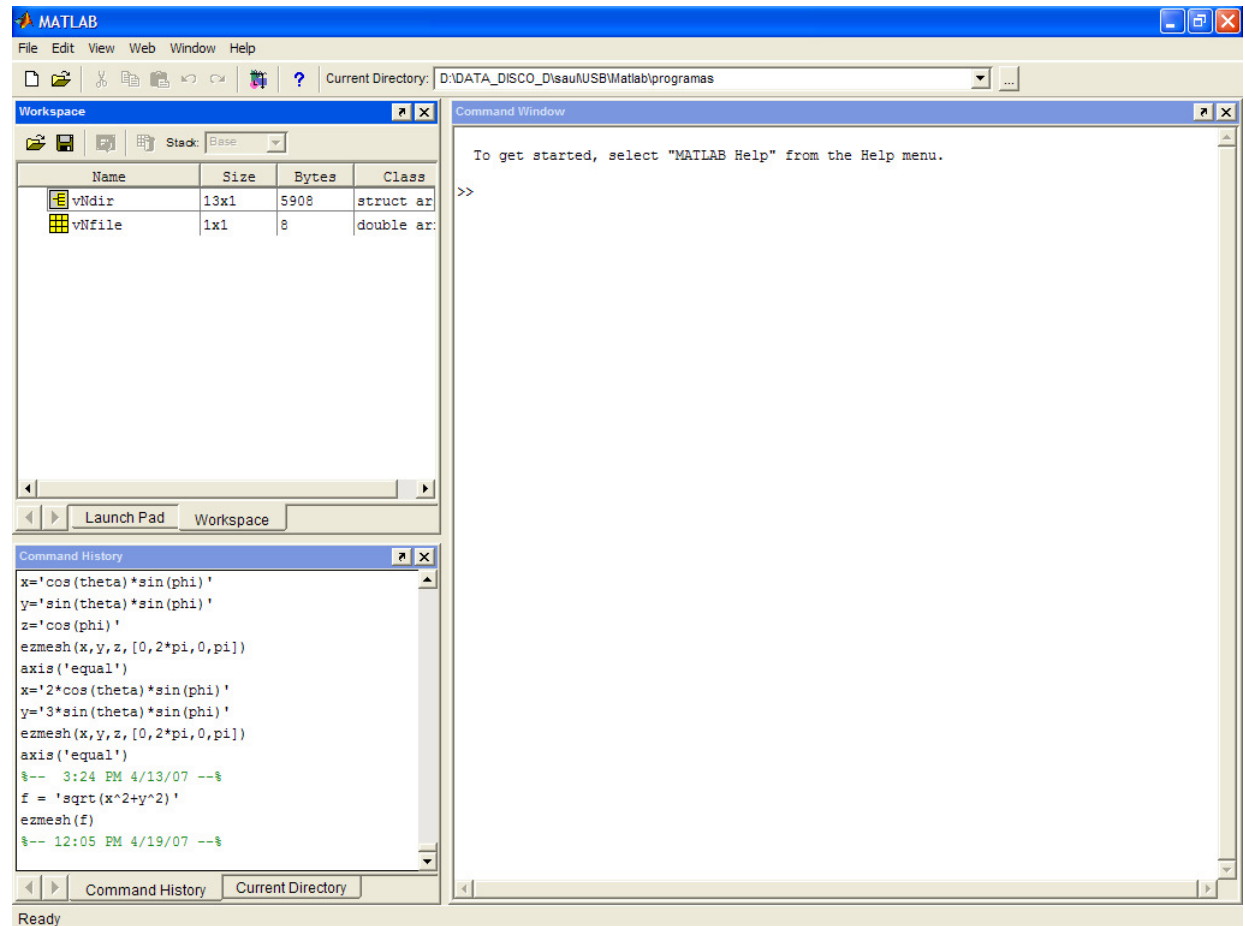
- Launch Pad: provee acceso fácil a herramientas, demostraciones y documentación.
- Command Window: se utiliza para ingresar variables y usar funciones. Es allí donde se escriben los programas.
- Command History: en esta ventana se almacenan todas las líneas escritas en la ventana de comandos (Command Window) a modo de historial. Se pueden seleccionar, copiar y usar funciones previamente ingresadas.



El escritorio de MATLAB

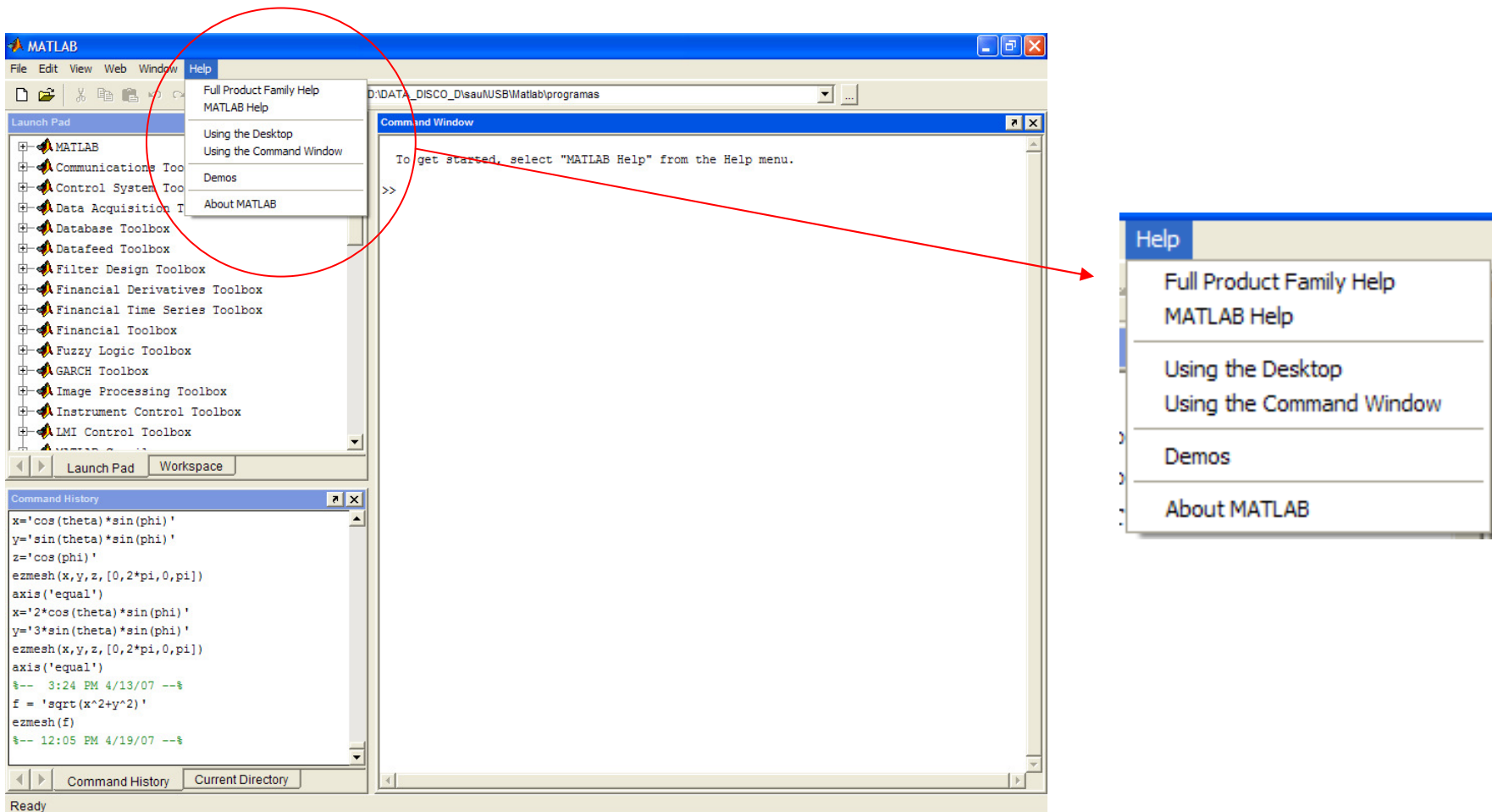
Dos ventanas adicionales que se pueden alternar con las de Command Window y Command History respectivamente son Workspace y Current Directory

- El Workspace contiene información sobre todas las variables que se hayan definido en esta sesión.
- Current Directory muestra los ficheros del directorio activo o actual.



El escritorio de MATLAB

MATLAB dispone de un excelente Help con el que se puede encontrar la información que se desee. La figura muestra las distintas opciones que aparecen en el menú Help.



El escritorio de MATLAB

El concepto de directorio actual es importante en MATLAB. Para cambiar de directorio actual se utiliza el comando `cd` (de change directory), seguido del nombre del directorio, para el cual se puede utilizar un camino (“path”) absoluto

```
cd C:\Matlab\Ejemplos
```

o relativo

```
cd Ejemplos
```

Para subir un nivel en la jerarquía de directorios se utiliza el comando

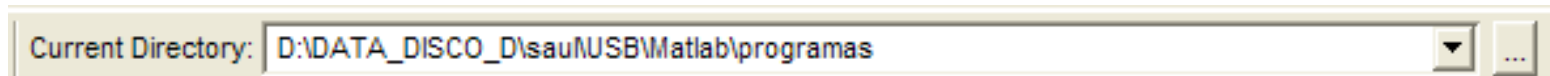
```
cd ..,
```

y

```
cd ../../
```

para subir dos niveles.

Aunque es más fácil cambiar el directorio actual desde la barra situada en la parte superior de la ventana.



Nociones básicas

Ingresar matrices.

Para ingresar una matriz, solo hay que respetar algunas convenciones:

- Separar los elementos de una fila con espacios en blanco o comas.
- Usar el punto y coma, ; , para indicar el fin de cada fila.
- Encerrar la lista de elementos con corchetes, [].

Por ejemplo, escribiendo en la ventana de comandos:

```
>> A = [ 1 2 3 4; 5 6 7 8; 9 10 11 12; 13 14 15 16 ]
```

MATLAB mostrará la matriz recién ingresada:

A =

```
     1     2     3     4
     5     6     7     8
     9    10    11    12
    13    14    15    16
```

Una vez que se ha ingresado la matriz, automáticamente será recordada en el entorno de trabajo de MATLAB. Podemos referirnos a ella simplemente escribiendo A.

Nociones básicas

Suma, transposición y diagonal.

```
>> sum(A)
```

MATLAB responderá con:

```
ans =
```

```
    28    32    36    40
```

que efectivamente corresponde a la sumatoria de cada columna de la matriz A previamente ingresada. Si no se especifica una variable de salida, MATLAB usa la variable ans par guardar el resultado de un cálculo.

Ahora, para efectuar la sumatoria de los elementos de una fila, antes será necesario transponer la matriz, para que las filas se cambien por las columnas. Para eso se usa el apóstrofe después de la matriz. Entonces:

```
A'
```

produce

```
ans =
```

```
    1    5    9   13  
    2    6   10   14  
    3    7   11   15  
    4    8   12   16
```

Nociones básicas

Suma, transposición y diagonal.

Pero si realizamos la operación

```
>> sum(A')
```

produce un vector columna que contiene la suma de las filas de A:

```
ans =
```

```
    10
```

```
    26
```

```
    42
```

```
    58
```

De manera similar, para sumar los elementos de la diagonal, se usa el comando `diag`:

```
>> sum(diag(A))
```

produce:

```
ans =
```

```
    34
```

Nociones básicas

Subíndices.

Para referirse a un elemento en particular de una matriz, se usa la notación

$$A(i, j),$$

donde i representa la fila y j , la columna. Por ejemplo, para la matriz

$A =$

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

el comando siguiente

```
>> A(2,4)
```

produce el elemento de la fila 2 y columna 4 de la matriz A

```
ans =
```

```
8
```

Nociones básicas

El operador dos puntos.

El operador dos puntos, `:`, es uno de los operadores más importantes de MATLAB. Actúa de diferentes maneras. La expresión

```
>> 1:10
```

es un vector fila que contiene los enteros de 1 a 10, es decir produce

```
1 2 3 4 5 6 7 8 9 10
```

Para obtener espacios distintos de la unidad, se especifica un incremento.

Por ejemplo:

```
>> 100:-7:50
```

produce como resultado

```
100 93 86 79 72 65 58 51
```

Si se usa en un subíndice el operador dos puntos se refiere a una porción de la matriz.

Por ejemplo:

```
>> sum(A(1:4,4))
```

```
ans =
```

```
40
```

Nociones básicas

El operador dos puntos.

Ahora, el operador dos puntos sólo, indica todos los elementos de una fila o columna de una matriz.

Por ejemplo:

```
>> sum(A(:,4))
```

```
ans =
```

```
    40
```

```
>> sum(A(1,:))
```

```
ans =
```

```
    10
```

Nociones básicas

Expresiones.

Como la mayoría de los otros lenguajes de programación, MATLAB provee expresiones matemáticas, pero estas expresiones involucran matrices. Las expresiones se componen de:

- Variables
- Números
- Operadores
- Funciones

Nociones básicas

Expresiones.

Las variables no requieren ninguna declaración de tipo o dimensiones. Cuando MATLAB encuentra un nuevo nombre de variable, automáticamente crea la variable y le asigna el espacio de almacenamiento adecuado. Si la variable ya existe cambia su contenido.

Por ejemplo,

```
>> num_estudiantes = 25
```

crea una matriz de 1-por-1 llamada num_estudiantes y guarda el valor 25 en su único elemento.

Un nombre de variable consiste en una letra seguida de cualquier número de letras, números o “underscores” (_), aunque sólo se tomarán en cuenta los primeros 63 caracteres (versión 7 de matlab).

MATLAB es sensible a las mayúsculas, es decir, distingue letras mayúsculas de minúsculas. ‘A’ y ‘a’ no son la misma variable.

Nociones básicas

Expresiones.

Para los números se usa la notación decimal convencional, con la opción del punto decimal y se manejan signos más y menos. La notación científica usa la letra e para especificar el factor de potencia de diez. Los números imaginarios usan indistintamente la i o la j como sufijo.

Algunos ejemplos de números escritos correctamente son:

-99 0.01 1.60210e-20 -3.14159j 3e5i

Los números tienen una precisión de 16 dígitos significativos decimales y un rango entre 10^{-308} y 10^{+308} .

Se usan los operadores aritméticos familiares y las mismas precedencias.

- + adición
- sustracción
- * multiplicación
- / división
- ^ potencia
- ' conjugado
- () orden específico de evaluación

Nociones básicas

Expresiones.

MATLAB provee un gran número de funciones matemáticas elementales. Tomar la raíz cuadrada o el logaritmo de un número negativo no es un error, el respectivo manejo de complejos se efectúa en forma automática. Además se incluyen funciones matemáticas más avanzadas. Para obtener un listado de las funciones matemáticas escribir:

```
>> help elfun
```

```
>> help specfun
```

```
>> help elmat
```

También existen funciones que entregan valores de constantes útiles, como

pi 3.14159265358979...

i unidad imaginaria

realmin el número de máquina más pequeño, 2^{-1022} ($2.2251 \cdot 10^{-308}$)

realmax el número de máquina más grande, 2^{+1023} ($1.7977 \cdot 10^{+308}$)

Inf infinito

Representación de números

Representación en base 10

$$0.101_{10} = 1 \cdot 10^{-1} + 0 \cdot 10^{-2} + 1 \cdot 10^{-3} = \frac{1}{10} + \frac{1}{1000} = 0.101$$

Representación en base 2

$$0.101_2 = 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = \frac{1}{2} + \frac{1}{8} = 0.625_{10}$$

$$\begin{aligned} 11011.01_2 &= 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} \\ &= 16 + 8 + 2 + 1 + \frac{1}{4} = 27.25_{10} \end{aligned}$$

En general

$$\begin{aligned} x &= (a_n a_{n-1} \cdots a_1 a_0 . a_{-1} a_{-2} \cdots a_{-m})_2 = \\ &a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \cdots + a_1 \cdot 2^1 + a_0 \cdot 2^0 + a_{-1} \cdot 2^{-1} + a_{-2} \cdot 2^{-2} + \cdots + a_{-m} \cdot 2^{-m} \end{aligned}$$

Representación de números

Representación en base 2

$$x = (a_n a_{n-1} \cdots a_1 a_0)_2 = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \cdots + a_1 \cdot 2^1 + a_0 \cdot 2^0 = 2(a_n \cdot 2^{n-1} + a_{n-1} \cdot 2^{n-2} + \cdots + a_1) + a_0$$

Luego a_0 es el resto de dividir x entre 2.

$$x = 2 \cdot x_1 + r_0$$

con

$$x_1 = a_n \cdot 2^{n-1} + a_{n-1} \cdot 2^{n-2} + \cdots + a_1 \quad \text{y} \quad r_0 = a_0$$

Para hallar el siguiente dígito a_1 , aplicamos el mismo procedimiento a x_1

$$x_1 = a_n \cdot 2^{n-1} + a_{n-1} \cdot 2^{n-2} + \cdots + a_1 = 2 \cdot x_2 + r_1$$

con

$$x_2 = a_n \cdot 2^{n-2} + a_{n-1} \cdot 2^{n-3} + \cdots + a_2 \quad \text{y} \quad r_1 = a_2$$

Representación de números

Representación en base 2

Ejemplo1:

$$x = 25 \Rightarrow x = 2 \cdot 12 + 1 \Rightarrow a_0 = 1$$

$$x_1 = 12 \Rightarrow x_1 = 2 \cdot 6 + 0 \Rightarrow a_1 = 0$$

$$x_2 = 6 \Rightarrow x_2 = 2 \cdot 3 + 0 \Rightarrow a_2 = 0$$

$$x_3 = 3 \Rightarrow x_3 = 2 \cdot 1 + 1 \Rightarrow a_3 = 1$$

$$x_4 = 1 \Rightarrow x_4 = 2 \cdot 0 + 1 \Rightarrow a_4 = 1$$

$$x_5 = 0 \quad \text{finaliza}$$

$$\text{por lo tanto} \quad (25)_{10} = (11001)_2$$

Representación de números

Representación en base 2

$$x = (.a_{-1}a_{-2} \cdots a_{-m})_2 = a_{-1} \cdot 2^{-1} + a_{-2} \cdot 2^{-2} + \cdots + a_{-m} \cdot 2^{-m}$$

$$\frac{1}{2}(a_{-1} + a_{-2} \cdot 2^{-1} + \cdots + a_{-m} \cdot 2^{-m+1})$$

$$\text{luego } 2x = a_{-1} + a_{-2} \cdot 2^{-1} + \cdots + a_{-m} \cdot 2^{-m+1}$$

así, a_{-1} es la parte entera de $2x$, y repetimos el mismo proceso para calcular el siguiente dígito

Ejemplo2:

$$x = 0.8125 \Rightarrow 2x = 1.625 \Rightarrow a_{-1} = 1$$

$$x_1 = 0.625 \Rightarrow 2x_1 = 1.25 \Rightarrow a_{-2} = 1$$

$$x_2 = 0.25 \Rightarrow 2x_2 = 0.50 \Rightarrow a_{-3} = 0$$

$$x_3 = 0.50 \Rightarrow 2x_3 = 1.0 \Rightarrow a_{-4} = 1$$

$$x_4 = 0.0 \quad \text{finaliza}$$

$$\text{por lo tanto } (0.8125)_{10} = (0.1101)_2$$

Representación de números

Representación en base 8

decimal	0	1	2	3	4	5	6	7
binario	000	001	010	011	100	101	110	111
octal	0	1	2	3	4	5	6	7

así, el número se divide en bloques de 3 dígitos (la parte entera de derecha a izquierda y la parte decimal de izquierda a derecha) desde el punto decimal

$$(101 | 101 | 001.110 | 010 | 100)_2 = (551.624)_8$$

Representación de números

Representación en base 8

Justificación del cálculo

$$(101101001.1100101)_2 = (101 | 101 | 001.110 | 010 | 100)_2 = (551.624)_8$$

$$\begin{aligned}x &= (.b_{-1}b_{-2}b_{-3}b_{-4}b_{-5}b_{-6} \dots)_2 \\&= b_{-1} \cdot 2^{-1} + b_{-2} \cdot 2^{-2} + b_{-3} \cdot 2^{-3} + b_{-4} \cdot 2^{-4} + b_{-5} \cdot 2^{-5} + b_{-6} \cdot 2^{-6} + \dots \\&= (4b_{-1} + 2b_{-2} + b_{-3}) \cdot 2^{-3} + (4b_{-4} + 2b_{-5} + b_{-6}) \cdot 2^{-6} + \dots \\&= (4b_{-1} + 2b_{-2} + b_{-3}) \cdot 8^{-1} + (4b_{-4} + 2b_{-5} + b_{-6}) \cdot 8^{-2} + \dots\end{aligned}$$

notar que los b_{-i} , $i=1,2,3,\dots$, son los números 0 o 1, y la operación entre paréntesis produce dígitos entre 0 y 7

Representación de números en el computador

Sea β la base usada en el computador, $\beta=2,8,16$.

Sea x un número real, con x distinto de cero y normalizado

$$x = \sigma (a_0.a_1a_2 \cdots a_t)_\beta \beta^{(E)_\beta}$$

donde

σ representa el signo de x

$(a_1a_2 \cdots a_t)_\beta$ la mantisa

$(E)_\beta$ el exponente en base β

a_0 es un número entero entre 1 y $\beta - 1$

Ejemplo de normalización de un número en base 2:

Sea el número $x = (49.8125)_{10}$

$$x = (110001.1101)_2 = (1.\underbrace{100011101}_2)_2 2^5$$

5 posiciones

Representación de números en el computador

Una palabra en el computador está formada por 32 bits.

Así, el número x puede ser representado usando (estándar IEEE):

- 1 bit para el signo
- 8 bits para el exponente (la característica)
- 23 bits para la parte fraccionaria (la mantisa)

Observación:

El exponente de 8 dígitos representa un intervalo del 0 al $2^8-1 = 255$

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

$$= 1 \cdot 2^7 + 1 \cdot 2^6 + \dots + 1 \cdot 2^1 + 1 \cdot 2^0 =$$
$$128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$$

A fin de que estos números se puedan representar, se resta 127 del exponente, de manera que el intervalo del exponente es en realidad $[-127, 128]$.

El exponente se almacena sumándole el sesgo correspondiente al número de bits de la representación usada, en este caso el sesgo corresponde a 127. La razón es que el exponente va a ser un número entre -127 y 128.

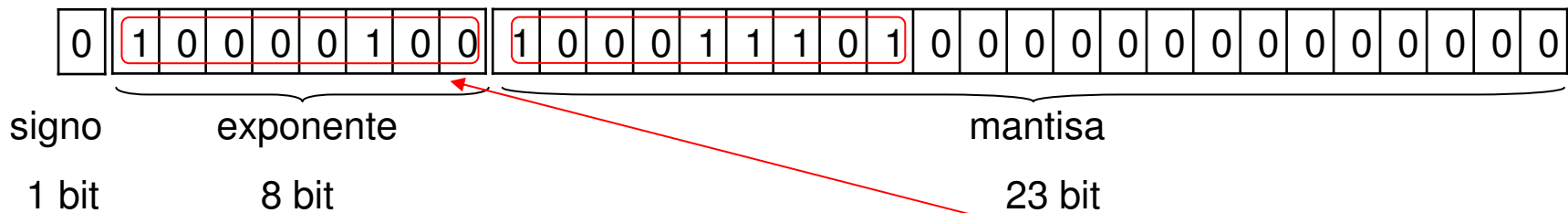
Representación de números en el computador

Retomando el ejemplo $x = 49.8125$

$$x = (110001.1101)_2 = (1.100011101)_2 2^5$$

El número normalizado

$$x = (1.100011101)_2 2^5 \quad \text{se representa como}$$



el exponente $e = (5 + 127 \text{ (sesgo)})_2 : e = 5 + 127 = 132 = (10000100)_2$

el signo s se representa como 0 si es positivo y 1 si es negativo, $s=0$

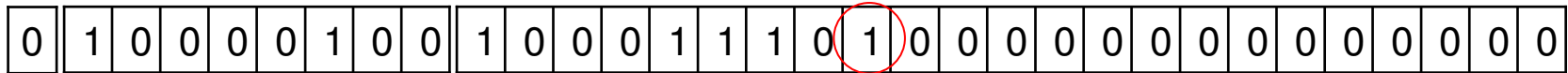
la mantisa es $(100011101)_2$ de donde $m = (0.100011101)_2$

El uso de este sistema proporciona un número en punto flotante de la forma

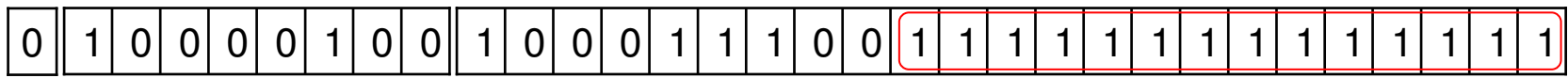
$$(-1)^s 2^{e-127} (1 + m)$$

Representación de números en el computador

El número de máquina menor que le sigue a

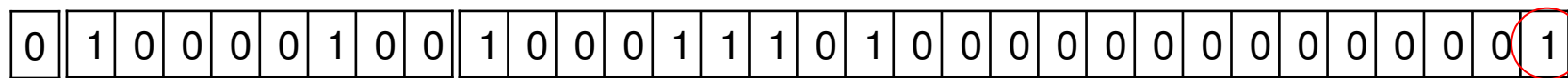


es



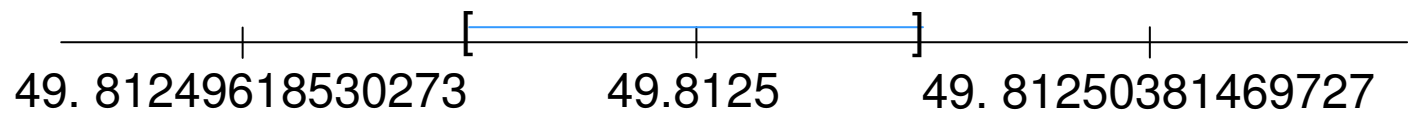
49. 81249618530273

y el siguiente número de máquina mayor es



49. 81250381469727

Esto significa que nuestro número de máquina original representa no sólo a 49.8125 sino también a los números que están en el intervalo entre corchetes (línea azul).



Representación de números en el computador

Ejercicio: Determinar la representación punto flotante del 1 en base decimal

0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Obs. El ϵ de la máquina es el espaciado entre números en punto flotante (ver la ayuda del comando **eps**), se define como la distancia entre 1 al siguiente número más grande representable. En caso de una representación de 32 bits, el ϵ de la máquina es 2^{-23} . Este valor representa el mejor error relativo con precisión simple.

0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Ejercicio: Determinar las representaciones punto flotante del realmax (2^{128}) y realmin (2^{-127}) en base decimal