Grado Ingeniería Informática. Matemáticas 2. Prácticas.

Práctica 1. Introducción a MATLAB

Departamento de Ciencia de la Computación e Inteligencia Artificial,
Universidad de Alicante

5 de febrero de 2017

Introducción

¿Porqué MATLAB?

Lenguaje de alto nivel para la codificación de **objetos matemáticos** tales como funciones o matrices, así como **operar** con ellos en un entorno amigable.
El nombre MATLAB viene de Matrix Laboratory y fue ideado para proporcionar un fácil manejo de matrices. Hoy en día, MATLAB es una potente herramienta para todo tipo de problemas.
Algunas características que se van a ver son:

- *Calculadora:* Puede evaluar (numéricamente) expresiones muy complejas.
- *Calculadora Simbólica:* El paquete **Symbolic Math** puede manipular expresiones simbólicas (por ejemplo para calcular limites, derivadas o integrales) y resolver ecuaciones.

Introducción

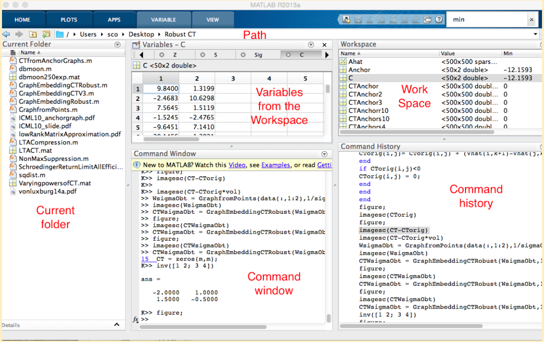
¿Porqué MATLAB?

Más....

- *Interpolación:* MATLAB provee herramientas para calculo de interpolaciones mediante diferentes métodos.
- *Programación:* Es un Lenguaje de Programacion procedimental con un entorno de Depuración.
- *Datos de tipo estructurados:* MATLAB provee objetos predefinidos, tales como polinomios, asi como objetos definidos por el usuario.
- *Recursos Gráficos*: Posee herramientas para el análisis gráfico funciones.

Entorno de Trabajo

Pantalla de inicio



Command window y Workspace son las subventanas principales y en la primera es donde se escriben las órdenes que se desea ejecutar, en concreto a la derecha del indicativo >>.

Entorno de Trabajo

Comandos de entorno

- Para guardar la sesión de trabajo en un archivo
  >> *diary nombre archivo*
- Este archivo aparece en el directorio actual (current folder) y se puede abrir con un editor de textos.
- Para finalizar la sesión de trabajo
  >> *diary off*
- Los comandos utilizados también se guardan en la ventana command history.

Entorno de Trabajo

Comandos de entorno

- Para incluir comentarios se utiliza %.
- Para introducir varias órdenes, se separan por comas.
- Para no mostrar el resultado en pantalla se finaliza con ;
- Más de una línea, se escriben tres puntos seguidos y se continua en la siguiente.
- Las flechas izquierda y derecha sirven para moverse en la misma linea.
- Las flechas arriba y abajo recuperan líneas ya escritas.

$>> B = 15, H = 3$ %B es la base y H la altura
$>> 30/5 + 12/6 + 15/3 + 125/5 + 40/8 + 36/2 + 45/9$ ...
$-25 + 24/12;$

Ayuda

Se puede invocar

- Menú contextual que se abre *help on selection*
- Tecla de función F1
- Tecleando la orden *help*
- Si se desea obtener información específica sobre alguna función definida por MATLAB, *help nombre función*.
- Si se quiere que la ayuda aparezca en una ventana independiente *helpwin plot*
- Si se teclean las primeras letras del nombre de un comando y pulsamos la tecla del tabulador aparece un menú de opciones.

Toolboxes

Paquetes específicos

Los comandos que guardan relación temática, se agrupan en toolboxes que pueden estar instaladas o no. Es decir, MATLAB posee paquetes o librerias con ordenes específicas.
Con el comando *ver* se pueden ver las *Toolbox* instaladas.
$>>$ *ver*

Calculadora

Línea de Comandos

- Si se teclea $>> 2 + 3$
- Se obtiene:
  $ans =$
  5
- ans es la respuesta actual. Si se teclea $>> ans$ se obtiene
  $ans =$
  5
- $>> a = 3 + 2$; el resultado se almacena en la variable a y no se ve la respuesta.

Calculadora

Operadores aritméticos

- $(+)$: suma.
- $(-)$: resta.
- $(*)$: multiplicación.
- $(/)$: división.
- $(\wedge)$: potencia.

El orden de prioridad de las operaciones es el mismo que en las calculadoras. La potencia tiene el orden de prioridad más alto, seguido del producto y división y por último la suma y la resta. Si se desea alterar este orden se deben introducir paréntesis.

Calculadora

Operadores lógicos

- ($>$): mayor que.
- ($<$): menor que.
- ($>=$): mayor o igual que.
- ($<=$): menor o igual que.
- ($==$): igual a.
- ($\sim=$): diferente de.

Se pueden agrupar con los operadores & y $\|$.

Calculadora

Ejemplo

$>> 8 + 4/2 - 1$
ans $=$
9
$>> (8 + 4)/2 - 1$
ans $=$
5
$>> ((8 + 4) * (2 - 6) - 10)/(3 - 1)$
ans $=$
-29

Calculadora

Ejemplo

$>> 3 * 4^{\wedge}2 - 1$

ans =

47

$>> (3 * 4)^{\wedge}2 - 1$

ans =

143

Calculadora

Función	Significado
sin, cos	Seno, coseno
sinh, cosh	Seno hiperbólico, coseno hiperbólico
asin, acos	Inversa del seno, coseno
asinh, acosh	Inversa del seno, coseno hiperbólico
tan, cot	Tangente , cotangente
tanh, coth	Tangent, cotangente hiperbólica
atan, acot	Inversa de la tangente, cotangente
atan2	Inversa de la tangente en el cuadrante cuatro
atanh, acoth	Inversa de la tangente, cotangente hiperbólica
sec, csc	Secante, cosecante
sech, csch	Secante, cosecante hiperbólica
asec, acsc	Inversa de la secante, cosecante
asech, acsch	Inversa secante, cosecante hiperbólica

Calculadora

Funciones de redondeo y residuo

- *fix*: parte entera de un número.
- *floor*: parte fraccionaria.
- *ceil*: parte entere de un número.
- *round*: redondea al siguiente entero.
- *mod*: residuo con signo de una división.
- *rem*: residuo sin signo de una división.
- *sign*: signo.

Calculadora

Formatos de Representación

Los formatos de representación numérica en pantalla pueden ser:

1. *format short*: 3 cifras (como máximo) en la parte entera y 4 en la decimal.
2. *format rat*: número racional.
3. *format long*: 2 cifras en la parte entera y 15 en la decimal.
4. *format short e*: 1 cifra en la parte entera, 4 en la decimal y 3 en el exponente.
5. *format long e*: 1 cifra en la parte entera, 15 en la decimal y 3 en el exponente

Aunque cambie el formato, solo es apariencia.

Calculadora

Representación numérica

MATLAB almacena internamente los números en doble precisión (ocho bytes). Eso significa que solo se representan números reales en un determinado rango: entre 10^{-308} y $2 * 10^{308}$.
A cualquier número menor que el menor número representado (*realmin*) el programa lo considera como cero y a cualquier número de una magnitud mayor que el mayor número representado (*realmax*) el programa lo considera de magnitud infinita (Inf o inf).

Calculadora

Ejemplo.

Una discretización del intervalo $[0, 2\pi]$ cada 0,01 unidades, se puede realizar mediante el comando $>> x = 0 :0.01: 2 * pi;$.

a) Teclea $>> y1 = sin(x);$
b) Usa $size(x, 2)$ y $size(y1, 2)$ para averiguar cuantos números hay en x.
c) Teclea $y2 = cos(x);$.
d) Teclea $>> plot(x, y1)$, aparece el gráfico del la función $y1$
e) Teclea $>> hold\ \ on;$, mantiene el gráfico en la misma ventana.
f) Teclea $>> plot(x, y2, 'color', 'r')$ para dibujar la función $y2$. Ir a la ventana *Window* $\rightarrow$ *Figure* 1 para ver ambas funciones.

Nombres de Variables

Variables

- Una variable es una etiqueta que identifica un trozo de memoria.
- Para asignar el nombre de una variable a un ente (número, vector, matriz o carácter) se debe utilizar el operador de asignación $=$.
- No es necesario declarar los nombres de la variables ni sus tipos.
- El nombre de una variable debe empezar por una letra y puede contener letras, números y guión bajo.
- No se permiten espacios y como máximo puede contener 63 caracteres.
- Distingue mayúsculas y minúsculas.

Nombres de Variables

Variables Especiales

- *eps*: es el epsilon de la máquina $(2{,}2204e - 16)$.
- *pi*: es el número pi $(3{,}14159\ldots)$.
- *i*, *j*: unidad imaginaria.
- *inf*: infinito.
- *NaN*: no es número.
- *exp(1)*: es el número e.
- *date*: representa la fecha.
- *nargin*: número de argumentos de entrada a una función.
- *nargout*: número de argumentos de salida de una función.

Nombres de Variables

Ejemplo. Volumen de una esfera

a) $>> clear;$

b) $>> r = 2$

c) $>> vol = (4/3) * pi * r^\wedge 3$

Nombres de Variables

Cadenas de caracteres

MATLAB puede definir variables que contengan cadenas de caracteres.
Las cadenas de texto van entre apostrofes y se almacenan en un vector, con un carácter por elemento.
Por ejemplo:
$>> fahr = 70, grd = (fahr - 32)/1.8;$
>> title(['Temp. Amb:', num2str(grd), 'grados centigrados'])

Nombres de Variables

Variables Simbólicas

Trabajar con variables simbólicas permite manejar ecuaciones de manera simbólica y resolverlas después reemplazando por un valor cada una de las variables.
Permite resolver, por ejemplo, ecuaciones, integrales definidas o realizar cálculos de primitivas de manera analítica.
Las variables simbólicas son cadenas de caracteres que pueden representar números, funciones, expresiones o variables.

Nombres de Variables

Variables Simbólicas

Por defecto todas las variables son numéricas.
Para definir un objeto simbólico se utiliza el comando *sym*.
Por ejemplo, la orden $>> x = sym('x')$ declara x como una variable simbólica.
Para definir varias variables simbólicas conjuntamente, debemos utilizar *syms*.
Ejemplo, $>>$ *syms a b c* declara a, b y c como variables simbólicas.

Nombres de Variables

Expresiones Simbólicas

Si en una expresión interviene una variable simbólica la expresión es simbólica (aunque también intervengan variables numéricas).
Por ejemplo,
$>>$ *syms a b c x*
$>>$ $poli = a * x^{\wedge}2 + b * x + c$
define una expresión simbólica llamada *poli*.
Si se teclea $>>$ *poli* se obtiene la expresión $ax^2 + bx + c$
Para conocer las variables simbólicas de una expresión: *symvar*.

Nombres de Variables

Expresiones Simbólicas

Si se desea conocer el valor de una expresión simbólica en un punto: *subs*.

Por ejemplo $>> subs(poli, x, 1)$ sustituye en la variable llamada poli el valor de x por 1.

Se pueden hacer substituciones de forma conjunta. Por ejemplo, si $a = 1$ y $b = 2$, la orden que se utiliza es

$>> subs(poli, [a, b], [1, 2])$

o

$>> subs(poli, \{a, b\}, \{1, 2\})$

proporciona como respuesta $x^2 + 2x + c$

Nombres de Variables

Expresiones Simbólicas

Para extraer el numerador o denominador de una expresión se utiliza el comando *numden*.

Por ejemplo:

$>> syms\ x; y = (x^\wedge 2 - 1)/(2 * (x + 4)^\wedge 3)$

$>> numden(y)$

ans $=$

$x^2 - 1$

$>> [num, den] = numden(y)$

$num =$

$x^2 - 1$

$den = 2 * (x + 4)^3$

Nombres de Variables

Expresiones Simbólicas

Dados

$>> syms\ x\ a$

$>> formula1 = 3 * x^\wedge 2 + 2 * a * x + 15 * x^\wedge 3 * a^\wedge 2 + 25 * a$

$>> formula2 = (x + a)^\wedge 3$

$>> formula3 = x^\wedge 3 + x^\wedge 2 + x + 1$

$>> formula4 = (a^\wedge 3 + 3 * a^\wedge 2 + 3 * a + 1)/(a^\wedge 2 + 2 * a + 1)$

Para simplificar una expresión simbólica: *simplify*.

$>> simplify(formula4)$

Para factorizar una expresión: *factor*.

$>> factor(formula3)$

Para desarrollar en sumas: *expand*.

$>> expand(formula2)$

Para simular la escritura matemática habitual: *pretty*.

$>> pretty(formula1)$

Nombres de Variables

Tipos de variables

Para recordar que variables se han definido: $>>$ *who*
Si tecleamos $>>$ *whos* se incluyen también el tipo de variable.
Para saber si ya esta definida una variable $>>$ *exist*('var')
Para borrar una variable *clear* seguido del nombre de la variable (o variables, separados por espacios en blanco).
El comando *clear* borra todas las variables del área de trabajo (workspace).

Gráficos

Consideraciones

MATLAB es una potente herramienta para realizar gráficos.
Los gráficos se abren en una ventana denominada *figure1*.
Se pueden dibujar conjuntamente varias gráficas en los mismos ejes o bien se puede subdividir la ventana en subventanas en las que aparezcan las gráficas.
Se puede influir en la apariencia de los gráficos variando el intervalo de presentación en los ejes cartesianos o incluyendo una cuadricula.

Gráficos

Comando Plot

Dados los puntos $(1,1),(2,-1),(3,7),(-4,4),(5,-2)$.
Para dibujar la poligonal que los une, basta teclear:
$>> x = [1\ 2\ 3\ \ -4\ 5]; y = [1\ \ -1\ 7\ 4\ \ -2]; plot(x,y)$
El comando *plot* dibuja la poligonal que une puntos del plano generados al utilizar las componentes de dos vectores.
Si en lugar de un par de vectores aplicamos el comando *plot* sobre un único vector $v = (v_1, v_2, ..., v_n)$, entonces dibuja la poligonal que une el conjunto puntos (i, v_i), con $i = 1, \ldots, n$.
Por ejemplo, la orden $>> plot([3, 1, 1/2, 7])$
dibuja la poligonal que une los puntos $(1,3),(2,1),(3,1/2)$ y $(4,7)$.

Gráficos

Comando Plot

Se puede añadir un título a la gráfica con *title*.
Se puede incluir etiquetas en los ejes con *xlabel* o *ylabel*.
También se puede introducir texto con *text* en un zona de interés.
En el caso de tener varias curvas, se pueden añadir etiquetas con *legend*.
Para todo ello basta acceder al menú de la ventana del gráfico y en *Insert* escoger la opción deseada.

Gráficos

Comando Plot

Otra forma de añadir etiquetas en un gráfico es escribir las órdenes (cuando está activa una ventana de gráfica).

La sintaxis de cada una de las órdenes es:

\>> *title*('El título')

\>> *xlabel*('Eje de abscisas') (con xlabel off se borra).

\>> *ylabel*('Eje de ordenadas') (con ylabel off se borra).

\>> *text*(a,b,'este es el punto (a,b)')

\>> *legend*('texto identificativo de una linea')

La siguiente orden sirve para introducir texto con el ratón

\>> *gtext*('este texto va aquí')

Gráficos

Comando Plot

MATLAB utiliza ciertas opciones por defecto sobre la apariencia de un gráfico.
Algunas se pueden modificar.
Igual escala en ambos ejes
$>>$ *axis equal*
Representar una curva en el rectángulo $[a, b] \times [c, d]$:
$>>$ *axis*([*a b c d*])
Restaura el rango por defecto $>>$ *axis auto* .
Introducir una cuadrícula $>>$ *grid on* (grid off la desactiva).

Gráficos

Dibujar conjuntamente varias gráficas

Incluir en el comando *plot* tantos pares de vectores como gráficos a representar.

$>> x = 0 : pi/100 : 3 * pi; y = cos(x); z = cos(3 * x);$

$>> plot(x, y, x, z);$

También se puede utilizar:

$>> x = 0 : pi/100 : pi/4; y = tan(x); plot(x, y)$

$>> hold\ \ on$

$>> z = sin(2 * x); plot(x, z)$

Se pueden superponer dos gráficas visualizando las escalas de cada una de ellas, usando el comando *plotyy*.

$>> x = -1 : 0.01 : 12; y = 300 * exp(-0.01 * x); z = 0.2 * exp(-0.1 * x);$

$>> plotyy(x, y, x, z)$

Gráficos

Color

Se debe incluir el color como una opción en el comando plot (entre apóstrofes).

- b: azul (blue)
- c: verde claro (cyan)
- g: verde oscuro (green)
- k: negro (black)
- m: rojo oscuro (magenta)
- r: rojo (red)
- w: blanco (white)
- y: amarillo (yellow)

$>> x = 0 : pi/100 : 3 * pi; y = cos(x);\ \ plot(x, y, 'r - -')$

Gráficos

Trazo

Se debe incluir el trazo como una opción en el comando plot (entre apóstrofes).

- -: trazado lineal continuo
- $--$: trazado lineal discontinuo
- -.: trazado lineal discontinuo intercalando punto y linea
- : : trazado con linea de puntos
- *: marca *
- $+$: marca $+$
- o: círculo
- x: marca con un aspa
- .: un punto

$>> x = 0 : pi/100 : 3 * pi; y = cos(3 * x);\ plot(x, y,\text{'g:'})$

Gráficos

Subplot

Para ver varias gráficas en una misma ventana subdividiendo ésta en subventanas se utiliza $>> subplot(m, n, i)$

La sintaxis indica que la ventana de gráficos activa se divide en m partes horizontales y n verticales. El tercer argumento indica la subdivisión activa (numerada de izquierda a derecha y de arriba abajo).

Veamos esto con un ejemplo:

$>> x = -2 * pi : pi/25 : 2 * pi; y = sin(x); z = abs(x);$

$>> t = log(x + 10); w = exp(x - 1);$

$>> subplot(2, 2, 1); plot(x, y,$'r-')

$>> subplot(2, 2, 2); plot(x, z,$'g-.')

$>> subplot(2, 2, 3); plot(x, t,$'b o')

$>> subplot(2, 2, 4); plot(x, w,$'k*')

Gráficos

Gráficas de funciones simbólicas

MATLAB dispone (en el toolbox de cálculo simbólico) de los comandos *fplot* y *ezplot* que evitan las operaciones sobre cada componente de un vector.

$>> fplot(f, [a\ b])$ dibuja f en el intervalo $[a, b]$,

$>> fplot(f, [a\ b\ c\ d])$ dibuja f en el rectángulo $[a, b] \times [c, d]$.

Por ejemplo

$>> fplot(\text{'}sin(x)\text{'}, [0\ 4 * pi], \text{'g-.'})$

$>> fplot(\text{'}sin(x)\text{'}, [0\ 4 * pi\ 0\ 1.5], \text{'r'})$

Gráficos

Gráficas de funciones simbólicas

La orden *ezplot* es otra alternativa para generar de forma sencilla la gráfica de una función.

$>> ezplot(f)$ o $>> ezplot(f, [a, b])$

La orden

$>> y = sym('x^\wedge 2 - 1'); ezplot(y)$ y

$>> ezplot('x^\wedge 2 - 1')$ son similares.

Para representar una función que viene dada de forma implícita $F(x, y) = 0$, en un rectángulo $[a, b] \times [c, d]$, basta crear una expresión simbólica, F, y, a continuación

$>> ezplot(F, [a, b, c, d])$

Paquete Symbolic

Haciendo derivadas

- *Declaración:* cada variable x debe ser declarada $>> syms\ \ x$. Ahora x es un **objeto especial**.
- Declarar otra variable y: $>> syms\ \ y$
- *Definición de Function:* $>> f(x,y) = x^\wedge 2 + y^\wedge 3$. Ahora $f(x,y)$ es otro **objeto especial**, una función. Los objetos especiales también se almacenan en **workspace**.
- *Derivar:*
  $>> diff(f,x)$
  $ans(x,y) =$
  $2 * x$

Paquete Symbolic

Haciendo derivadas

- *Primera derivada:*
  $>> diff(f, y)$
  $ans(x, y) =$
  $3 * y^2$
- Segunda derivada: $>> diff(diff(f, x))$ o $diff(f, 2)$
  $ans(x, y) =$
  2

Paquete Symbolic

Ejemplo

a) Crea un polinomio de grado n, $p(x) = 2x^n + 5x^{n-1}$
b) Deriva $p(x)$
c) Calcula $diff(p, 2), diff(p, 3), \ldots$ y analiza el nuevo grado n.
d) Sea $>> n = 5$; se produce algún cambio en la derivada de $p(x)$?
e) Haz $>> q = symfun(2 * x^\wedge n + 5 * x^\wedge(n-1), [x])$ y calcula $diff(q, 2)$. Depende el resultado de $n = 5$?

Paquete Symbolic

La derivada como límite

Como se ha visto en teoría, la definición de derivada es

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Usando el Paquete Symbolic, se define
$>> syms\ \ h\ \ n\ \ x.$
Entonces

$$>> limit((cos(x+h) - cos(x))/h, h, 0)$$

produce $-sin(x)$ como respuesta.

$$>> limit((1 + x/n)^{\wedge}n, n, inf)$$

produce la exponencial $exp(x)$ como respuesta.

Paquete Symbolic

La derivada como límite

Algunas veces se necesita el **límite lateral**.

- Se realiza $>>$ *syms x h* si aún no están declaradas como simbólicas.
- Sabemos que $f(x) = |x|$ no es diferenciable en $x = 0$. ¿Porqué?
- Computamos el límite por la izquierda:
  $>> lh = limit((abs(x+h) - abs(x))/h, h, 0,\text{'left'});$

Produce la sorprendente respuesta:
$((real(x)^3 * i + imag(x) * real(x)^2) * i)/(imag(x)^2 + real(x)^2)^{(3/2)}$

Paquete Symbolic

La derivada como límite

(cont)
Y lo mismo ocurre para
$>> lr = limit((abs(x+h) - abs(x))/h, h, 0,\text{'right'});$
Los resultados son -1 y 1 respectivamente.
Realmente, $>> diff(abs(x))$ da $sign(x) = x/|x|$, el signo de la function

$$sign(x) = \begin{cases} +1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \end{cases}$$

Realmente $limit(x/abs(x), x, 0)$ produce NaN por que el limite por la izquierda da -1 Y el limite por la derecha $+1$.

Paquete Symbolic

Solve

Parte de la teoría de la asignatura cubre la **resolución de ecuaciones no lineales**.
Por ejemplo, encontrar las raíces del siguiente polinomio:

$$p(x) = x^3 - x^2 - 14x + 24$$

De forma que, una vez definida x y $p(x)$ llamamos a la función

$$>> solve(p)$$

Y el resultado es $2, 3, -4$ que son las raíces de $p(x)$. Puedes ver esto dibujando la función $p(x)$ en el intervalo $[-5, 5]$.

Paquete Symbolic

Solve

Realmente *solve* devuelve un conjunto de valores $\{-4, 2, 3\}$ en el ejemplo, y un conjunto vacío si no encuentra solución. Algunas veces se obtiene como resultado un conjunto infinito de elementos. Por ejemplo:

$$>> solve(x^\wedge 2 > 2)$$

produce

$Dom :: Interval(2^{(1/2)}, Inf)$
$Dom :: Interval(-Inf, -2^{(1/2)})$

Ajuste de curvas

Interpolación

Uno de los problemas más interesantes que se ven en la parte teórica de la asignatura es el de **interpolación**:

- Dado un conjunto de pares (x_i, y_i), $i = 1, 2, \ldots, N$. Buscamos una función $f(x) = y$.
- $f(x) = y$ debe verificar al menos $f(x_i) = y_i$ pero es deseable que $f(x)$ esté **tan cerca como sea posible** al valor teórico.
- Ademas seria deseable tener una función lo más simple posible.

Ajuste de Curvas

Ejemplo

a) Generar un vector x con enteros de 1 a 10:
$>> x = 1 : 10;$

b) Generar 10 valores aleatorios para $y \in [0, 5]$ asociados con los valores de x:
$>> y = rand(1, 10) * 5;$

c) Dibujar los pares de puntos:
$>> plot(x, y)$
$>> hold\ \ on;$
$>> plot(x, y,$'o','color','r'$)$

d) Llamar a la función básica:
$>> [p2, s2, m2] = polyfit(x, y, 2)$
donde se trata de usar un polinomio simple de segundo grado .

Ajuste de Curvas

Ejemplo (cont)

e) Cálculo del polinomio de segundo grado
$>> px = 1 : 0{,}01 : 10$
$>> [py2, d] = polyval(p2, px, s2, m2);$

f) Dibujar el polinomio de segundo grado
$>> l2 = plot(px, py2, \text{'r'});$
$>> legend(l2, \text{'poly 2º'})$

g) Incrementar el grado del polinomio. Sin embargo produce un resultado similar:
$>> [p3, s3, m3] = polyfit(x, y, 3);$
$>> [py3, d] = polyval(p3, px, s3, m3);$

h) Dibujar el polinomio de tercer grado
$>> l3 = plot(px, py3, \text{'g'});$
$>> legend([l2, l3], \text{'poly 2º'}, \text{'poly 3º'})$

Ajuste de Curvas

Ejemplo (cont)

i) Es necesario incrementar el grado del polinomio más o bien usar un método más preciso como **splines**:
$>> py = spline(x, y, px);$
$>> l = plot(px, py,$'b')
$>> legend([l2, l3, l],$'poly 2º','poly 3º','spline')

g) Salvar la figura como **Interpolacion.fig** así como **Interplacion.png**.

Ajuste de Curvas

Ejemplo (cont)

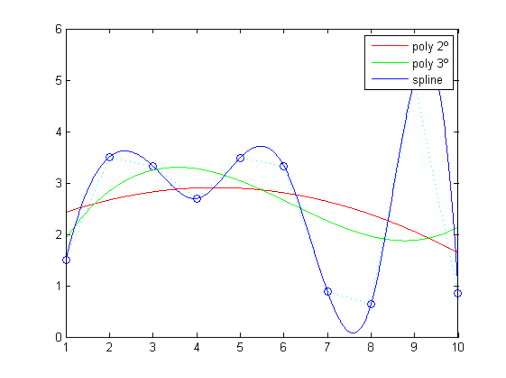


Figura: Diferentes interpolaciones para el mismo conjunto de puntos

Ejercicios

Práctica #1

Calcula:

1. $5^2 + 7 - \frac{26}{2}$
2. $\frac{2^3 - 7 + 11}{3 \cdot 2^2}$

Ejercicios

Práctica #2

Representa en distintos formatos los siguientes números:

1. $\frac{2}{3}$.
2. $\frac{20}{3}$.
3. $\frac{2000}{3}$.
4. $\frac{20000000000}{3}$.
5. $1{,}333$

Ejercicios

Práctica #3

Utiliza la ayuda de MATLAB para obtener información sobre num2str

Ejercicios

Práctica #4

Teniendo en cuenta que una secuencia de comandos se puede almacenar en un fichero con extensión .m mediante el menú New opción script

1. Crea un *script* nuevo.
2. Introduce una secuencia de comandos (por ejemplo la práctica 2).
3. Guarda el *script* asignándole el nombre practica4.
4. Ejecuta el *script* tecleando en línea de comandos el nombre del *script*.
5. Cierra el *script*.

Ejercicios

Práctica #5

Calcula $a^2 - b^2$ **con** $a = 1{,}4 \times 10^{154}$ **y** $b = 1{,}3 \times 10^{154}$

1. ¿Qué ocurre con el resultado?
2. ¿Cómo se puede solucionar el problema?

Ejercicios

Práctica #6

Ejecuta las siguientes órdenes y observa las respuestas:

1. $>> x = 10/7$
2. $>> y = sym('10/7')$
3. $>> double(y)$
4. $>> y + 5$
5. $>> double(y) + 5$

Ejercicios

Práctica #7

Encuentra la hipotenusa de un triangulo rectángulo cuyos catetos miden 12 y 5

Ejercicios

Práctica #8

Dado un triángulo con ángulo $\pi/6$ entre dos lados de longitud 5 y 7, utiliza el teorema del coseno para encontrar el tercer lado c.

Ejercicios

Práctica #9

Obtén las expresiones indicadas:

1. El desarrollo de $(x^3+1)^2$
2. La simplificación de $\frac{x^3+x^2-x-1}{x-1}$
3. La factorización de x^3+x^2-x-1
4. La expresión matemática habitual de $(x^3+1)^2$

Ejercicios

Práctica #10

Se desea medir la altura que alcanza un cultivo, sabiendo que dicha altura es función del tiempo. Las mediciones se hacen una vez al día y los resultados obtenidos se resumen en la siguiente tabla:

Tiempo (días)	1	2	3	4	5
Altura (cm)	5,2	6,6	7,3	8,6	10,7

1. Obtén la gráfica que explique ese crecimiento.
2. Debe incluir un título apropiado.
3. Debe incluir etiquetas en los ejes que indiquen cual es la variable correspondiente.
4. Varia el trazo y el color.

Ejercicios

Práctica #11

Dibuja en el intervalo $[-2,2]$ las curvas $y=x^3$ e $y=x^2+1$ conjuntamente, etiquetándolas con un título que indique cuál es cada una de ellas. Hazlo de tres formas distintas:

1. Superponiendo las gráficas utilizando los mismos ejes para ambas gráficas.
2. En dos subventanas, cada gráfica con sus propios ejes coordenados
3. Generando dos ventanas de gráficos distintas.

Ejercicios

Práctica #12

Dibuja:

1. La función seno en el intervalo $[-3\pi, \pi]$ con el comando *ezplot*.
2. Dibuja la función coseno en ese mismo intervalo, conjuntamente con la función seno.