

# Introduction\_to\_MATLAB

December 19, 2018

## 1 Teoría de elementos finitos y su implementación (10/12/2018)

### 1.1 1. Introducción a MATLAB

MATLAB que derivada de su nombre *laboratorio de matrices*, es un lenguaje de computación desarrollado para el procesamiento de datos en forma de arreglo de números. MATLAB integra computación y visualización y provee diversidad de instrucciones y funciones que pueden ser usados de una manera interactiva para obtener soluciones numéricas para una variedad de problemas de ingeniería.

[Recurso online](#) (es necesario tener conexión a internet e identificarse con su cuenta asociada a MATLAB).

[Introduction to MATLAB – MIT](#)

[MATLAB Primer](#)

[Documentación oficial en PDF](#)

[Math Modeling Computing and Communication \(SIAM\)](#)

### 1.2 Cómo leer los cuadernos

Estos cuadernos están conformados por celdas, algunas de ellas están en formato [Markdown](#) y otras como para ejecutar en el prompt de MATLAB. (Estas últimas las puede identificar porque al lado izquierdo aparece In [x] donde  $x \in \mathbb{N}$ .)

**Nota:** MATLAB distingue entre mayúsculas y minúsculas, lo que significa que  $a + B$  no es lo mismo que  $a + b$ .

#### 1.2.1 1.1. Expresiones numéricas

```
In [1]: 7
        7.1
        7.92
        42.13; % No hay réplica.
```

ans =

7

```
ans =
```

```
7.1000
```

```
ans =
```

```
7.9200
```

**1.1.1. Operaciones aritméticas con escalares** (+, -, \*, /, ^) adición, sustracción, multiplicación, división y exponenciación. Así por ejemplo.

```
In [2]: 2/3 + 4*5 - 3^2
```

```
ans =
```

```
11.6667
```

Nota: `clc` y `clear` son comandos para limpiar la consola y borrar las variables, respectivamente.

### 1.1.2. Asignación de valores numéricos a nombres de variables

```
In [3]: x = 5;  
        y = 2 + 2*x  
        z = (x + y)^2
```

```
y =
```

```
12
```

```
z =
```

```
289
```

### 1.1.3. Números complejos

```
In [4]: z = 2 - 3*i % Otra expresión válida es z = 2+5j
```

z =

2.0000 - 3.0000i

**1.1.4. Funciones matemáticas** Hay una lista grande de funciones matemáticas que son reservadas por MATLAB. Listemos algunas de ellas

Funciones matemáticas	Sintáxis en MATLAB
Valor absoluto de $x =  x $	<code>abs(x)</code>
Raíz cuadrada de $x = \sqrt{x}$	<code>sqrt(x)</code>
Seno de $x = \text{sen}(x)$	<code>sin(x)</code>
Coseno de $x = \cos x$	<code>cos(x)</code>
Coseno hiperbólico de $x = \cosh x$	<code>cosh(x)</code>
Exponencial de $x = \exp x$	<code>exp(x)</code>
Logaritmo natural de $x = \ln x$	<code>log(x)</code>
Logaritmo en base 10 de $x = \log_{10} x$	<code>log10(x)</code>
Arcoseno de $x = \text{asen}x = \text{sen}^{-1}(x)$	<code>asin(x)</code>
Arcocoseno hiperbólico de $x = \text{acosh}x$	<code>acosh(x)</code>

```
In [5]: clear all
        x = 4;
        z = sin(x) - log(cos(x))
```

z =

-0.3316 - 3.1416i

### 1.1.5. Formatos de salida

- format short.
- format long.

```
In [6]: format short
        fprintf('El valor de pi es:'); disp(pi);
        fprintf('El valor de cos(1.57) es:'); disp(cos(1.57));
        format long
        fprintf('El valor de pi es:'); disp(pi);
        fprintf('El valor de cos(1.57) es:'); disp(cos(1.57));
```

El valor de pi es:     3.1416

El valor de `cos(1.57)` es: 7.9633e-04

El valor de `pi` es: 3.141592653589793

El valor de `cos(1.57)` es: 7.963267107332633e-04

### 1.1.6. Vectores Creación de vectores fila.

```
In [7]: u = [2 3 4 5] % Sin separación de comas.
```

u =

2 3 4 5

o equivalentemente

```
In [8]: u = [2, 3, 4, 5]; % Con separación de comas.
x = 2:5;
y = 2:0.1:3
z = 5:-1:1
p = [1, 7 + 3i, 4, 3 - 5i]
```

y =

Columns 1 through 3

2.000000000000000 2.100000000000000 2.200000000000000

Columns 4 through 6

2.300000000000000 2.400000000000000 2.500000000000000

Columns 7 through 9

2.600000000000000 2.700000000000000 2.800000000000000

Columns 10 through 11

2.900000000000000 3.000000000000000

z =

5      4      3      2      1

p =

Column 1

1.0000000000000000 + 0.0000000000000000i

Column 2

7.0000000000000000 + 3.0000000000000000i

Column 3

4.0000000000000000 + 0.0000000000000000i

Column 4

3.0000000000000000 - 5.0000000000000000i

Creación de vectores columna

```
In [9]: w = u'
        m = p'
        q = conj(p')
```

w =

2  
3  
4  
5

m =

1.0000000000000000 + 0.0000000000000000i  
7.0000000000000000 - 3.0000000000000000i  
4.0000000000000000 + 0.0000000000000000i  
3.0000000000000000 + 5.0000000000000000i

q =

```

1.0000000000000000 + 0.0000000000000000i
7.0000000000000000 + 3.0000000000000000i
4.0000000000000000 + 0.0000000000000000i
3.0000000000000000 - 5.0000000000000000i

```

### 1.1.7. Matrices Crear una matriz de $3 \times 4$ .

```

In [10]: A = [2 3 5 6; .2 4 5 1; 0 1 5 2]
          [m, n] = size(A) % Orden de la matriz A.

```

A =

Columns 1 through 3

```

2.0000000000000000    3.0000000000000000    5.0000000000000000
0.2000000000000000    4.0000000000000000    5.0000000000000000
0                    1.0000000000000000    5.0000000000000000

```

Column 4

```

6.0000000000000000
1.0000000000000000
2.0000000000000000

```

m =

3

n =

4

### 1.1.8. Generación de matrices especiales

```

In [11]: X = ones(n, n) % Matriz de unos.
          Z = zeros(m, n) % Matriz de ceros.
          d = eye(n, n) % Matriz identidad.

```

X =

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

Z =

0	0	0	0
0	0	0	0
0	0	0	0

d =

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

### 1.1.9. Operaciones con matrices $A + B$ , $A - B$ , $A * B$

```
In [12]: B = magic(4)
```

B =

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

```
In [13]: help("magic") % Descripción corta del comando magic.
```

MAGIC Magic square.

MAGIC(N) is an N-by-N matrix constructed from the integers 1 through N^2 with equal row, column, and diagonal sums. Produces valid magic squares for all N > 0 except N = 2.

Reference page in Doc Center  
doc magic

## Manipulación con elementos de una matriz

```
In [14]: B(2, :) = 5 * B(2, :)  
         % Nueva matriz B. Multiplicación por 5 a la fila 2 de la matriz B.  
         B_1 = B(:, 3)  
         % Obtención de la tercera columna de B.  
         C = B(:, 2) + B(:, 4)  
         % Suma de las columnas 2 y 4 de la nueva matriz B.
```

B =

16	2	3	13
25	55	50	40
9	7	6	12
4	14	15	1

B\_1 =

3
50
6
15

C =

15
95
19
15

### 1.1.10. P-norma de una matriz Por ejemplo:

```
norm(x, p)  
norm(x, Inf)
```

```
In [15]: help norm
```

```
NORM    Matrix or vector norm.  
        NORM(X,2) returns the 2-norm of X.  
  
        NORM(X) is the same as NORM(X,2).  
  
        NORM(X,1) returns the 1-norm of X.
```



`NORM(X,Inf)` returns the infinity norm of `X`.

`NORM(X,'fro')` returns the Frobenius norm of `X`.

In addition, for vectors...

`NORM(V,P)` returns the  $p$ -norm of `V` defined as  $\text{SUM}(\text{ABS}(V).^P)^{(1/P)}$ .

`NORM(V,Inf)` returns the largest element of `ABS(V)`.

`NORM(V,-Inf)` returns the smallest element of `ABS(V)`.

By convention, NaN is returned if `X` or `V` contains NaNs.

See also `COND`, `RCOND`, `CONDEST`, `NORMEST`, `HYPOT`.

Reference page in Doc Center  
`doc norm`

Other functions named `norm`

<code>codistributed/norm</code>	<code>DynamicSystem/norm</code>	<code>mfilt/norm</code>	<code>tall/norm</code>
<code>dfilt/norm</code>	<code>gpuArray/norm</code>	<code>sym/norm</code>	

### 1.1.11. Determinante e inversa de una matriz

`det(A)`

`inv(A)`

`A\B`

`A/B`

$A \setminus B$  significa  $A^{-1}B$ .

$A/B$  significa  $AB^{-1}$ .

**1.1.12. Operación dot (`.`)** Sean las `X` y `M` dos matrices del mismo orden. Las siguientes instrucciones realizan operaciones elemento a elemento.

`Zm = X .* M`

`Zd = X ./ M`

`Ze = X .^ M`

Además, si `x0` y `m0` son números reales se tiene los siguientes casos particulares:

`Zd = x0 ./ M`

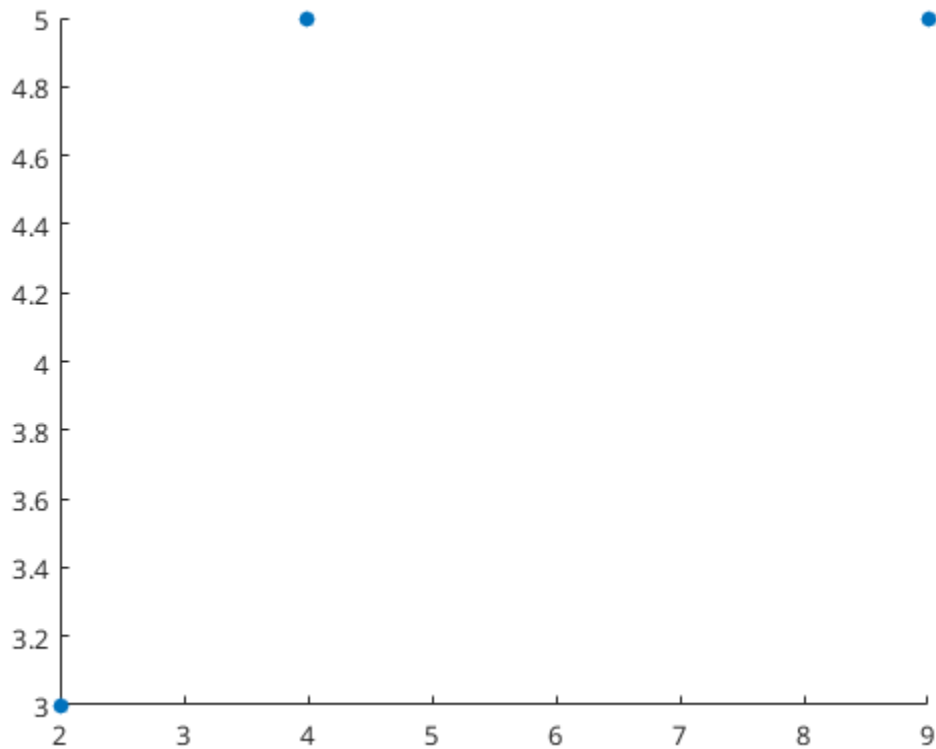
`Ze = x0 .^ M`

`Ze = M .^ m0`

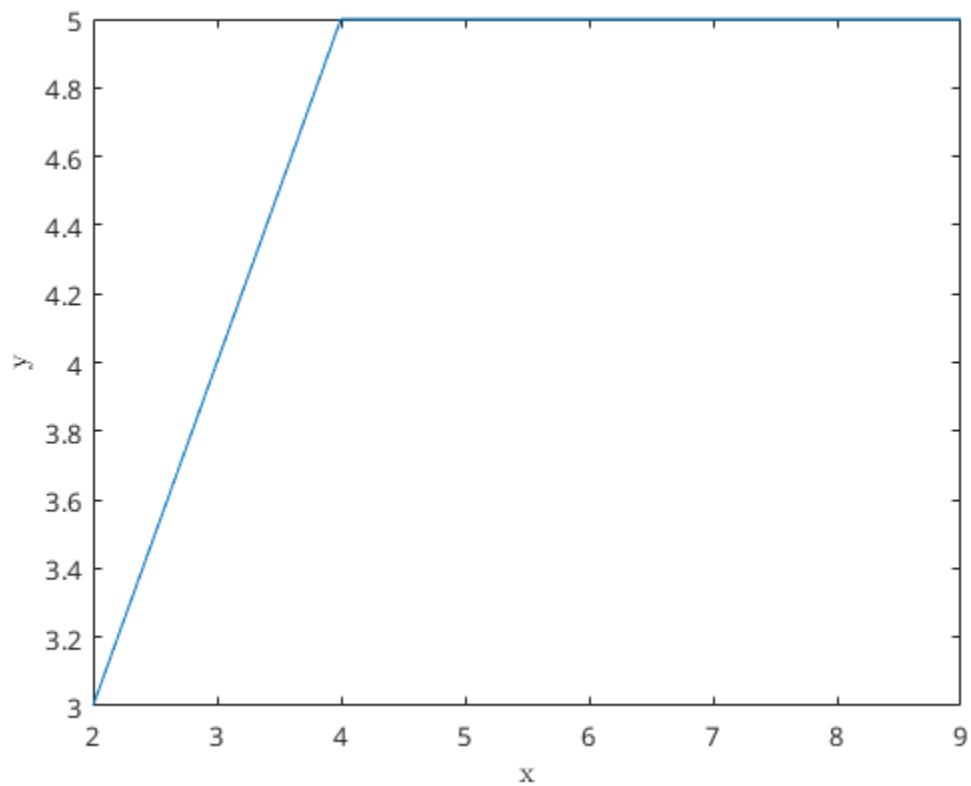
### 1.2.2 1.2 Ploteando en MATLAB con el comando plot

1. Usando el comando `plot(x,y,'o')`, `plot(x,y,'ro-')` plotear
  - a. los puntos:  $(2,3)$ ,  $(4,5)$ ,  $(9,5)$ .
  - b. la poligonal que une los puntos  $(2,3)$ ,  $(4,5)$ ,  $(9,5)$ .
2. Graficar la función  $f(x) = x^3 + 2x - 1$  en el intervalo  $-2 \leq x \leq 2$ .

```
In [16]: % Pregunta 1
         % Parte a
         scatter([2 4 9], [3 5 5], 'filled');
```



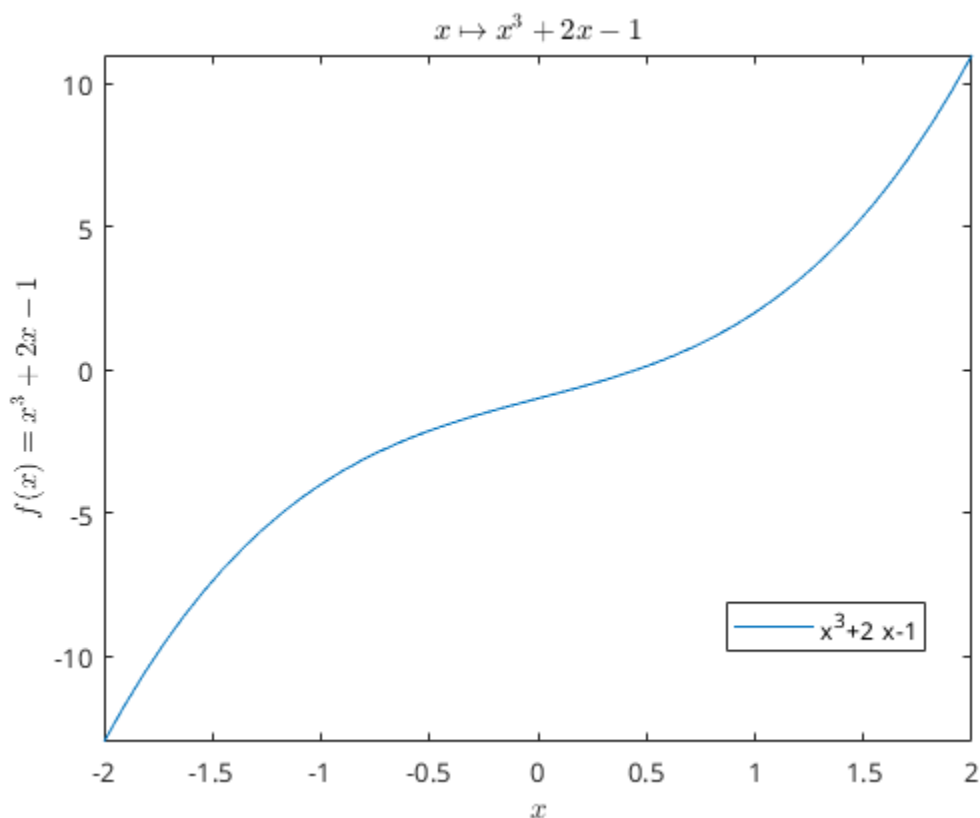
```
In [17]: % Parte b
         syms x y
         plot( [2 4 9], [3 5 5] );
         xlabel('x','Interpreter','latex');
         ylabel('y','Interpreter','latex');
```



In [18]: doc fplot

In [19]: *% Pregunta 2*

```
syms x y
fplot( @(x) x.^3 + 2.*x - 1, [-2 2] )
xlabel('$x$', 'Interpreter', 'latex');
ylabel('$f(x)=x^3+2x-1$', 'Interpreter', 'latex');
legend('show', 'Location', 'best');
title('$x \mapsto x^3+2x-1$', 'Interpreter', 'latex');
```

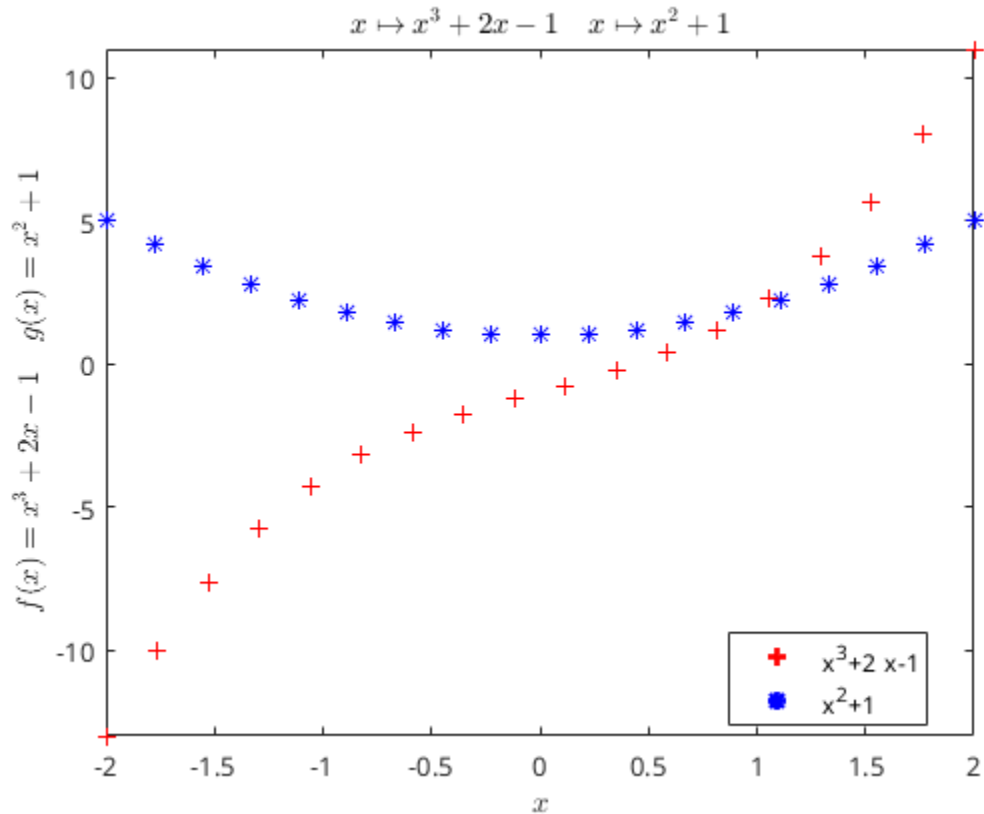


3. Muestre las gráficas de las funciones en un mismo plano las gráficas de las funciones

$$f(x) = x^3 + 2x - 1, \quad g(x) = x^2 + 1$$

en el intervalo  $-2 \leq x \leq 2$ .

```
In [20]: % Pregunta 3
I = [-2 2]; % Intervalo
fplot( @(x) x.^3 + 2.*x - 1, I, 'r+', 'LineWidth', 2);
hold on
fplot( @(x) x.^2 + 1, I, 'b*', 'LineWidth', 2)
xlabel('$x$', 'Interpreter', 'latex');
ylabel('$f(x)=x^3+2x-1$ quad $g(x)=x^2+1$', 'Interpreter', 'latex');
legend('show', 'Location', 'best');
title('$x \mapsto x^3+2x-1$ quad $x \mapsto x^2+1$', 'Interpreter', 'latex');
hold off
```



### 1.2.3 1.3 M-file

#### 1. script M-file

Crear un script M-file para graficar las funciones  $f(x) = kx^2 - 1$  cuando  $k \in \{1, 2, 3\}$  en el intervalo  $-2 \leq x \leq 2$ .

#### 2. function M-file

Crear una función M-file para la función  $f(x) = x^2 - 1$ . Luego, ejecute:

- $f(3)$ .
- $f(x)$ , cuando  $x = 1:5$ .

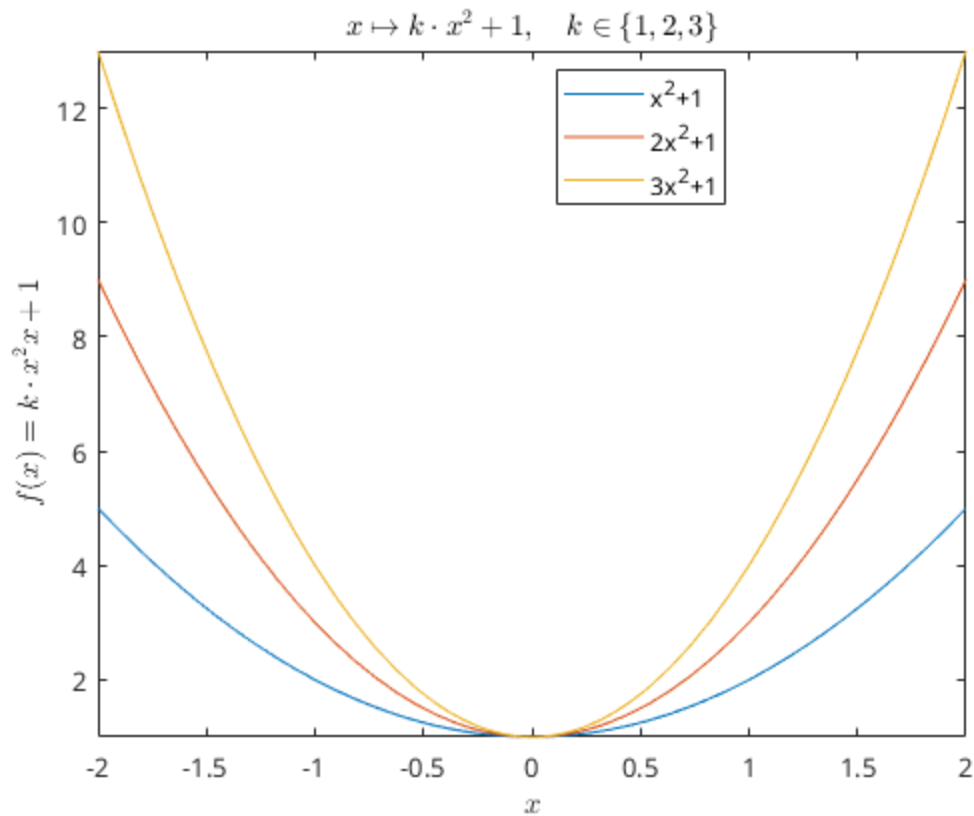
```

In [21]: % Programa script.m
          % Parte a
          for k = 1:3
              fplot( @(x) k * x.^2 + 1, [-2 2] )
              hold on
          end
          xlabel('$x$', 'Interpreter', 'latex');
          ylabel('$f(x)=k \cdot x^2+1$', 'Interpreter', 'latex');
    
```

```

legend('x^2+1','2x^2+1','3x^2+1','Location','best');
title('$x \mapsto k \cdot x^2 + 1, \text{quad } k \in \{1,2,3\}$','Interpreter','latex');
hold off

```



```

In [22]: %%file evalfun.m
function y = evalfun(x)
% evalfun calcula el valor $x \mapsto x^2 - 1$
% https://www.mathworks.com/matlabcentral/answers/25568-defining-functions
if ~isnumeric(x)
    error('Debe ingresar un número real.')
end
y = x^2 - 1;
end

```

Created file '/home/carlosal1015/GitHub/Finite-element-method-FEM/2018/Notebooks/evalfun.m'.

```

In [23]: % Parte a
fprintf('El valor de f(3) es %2.0f.', evalfun(3))

```

El valor de f(3) es 8.

```
In [24]: % Parte b
        for z = 1:5
            fprintf('El valor de f(%i)=%2.0f.\n', z, evalfun(z))
        end
```

El valor de f(1)= 0.

El valor de f(2)= 3.

El valor de f(3)= 8.

El valor de f(4)=15.

El valor de f(5)=24.