



Universidad de Jaén

Facultad de Ciencias Sociales
y Jurídicas

Trabajo Fin de Grado

PRÁCTICAS DE MATEMÁTICAS I EN OCTAVE

Alumno: Banhakeia Martínez, Juana María

Enero, 2019

RESUMEN:

En este trabajo se realizan todas las prácticas de Matemáticas I en el Grado de Administración y Dirección de Empresas en la Universidad de Jaén con el programa “Wolfram Mathematica”, pero usaré el programa Octave, el cual es gratuito; para la realización de dichas prácticas.

La realización de las prácticas de los temas que se estudian en la asignatura, estos temas se han ido desarrollando por pasos y con ejemplos explicativos. Viendo que comandos usar para cada tipo de caso que se nos plantea en los diferentes temas que se estudian en dicha asignatura.

Viendo en qué casos es más complicado el uso de este programa y en cuales situaciones es mejor el uso de Octave.

ABSTRACT:

In this work I perform all practices of Maths I in the degree of Administration and Business management at the University of Jaén with the computer program “Wolfram Mathematica”, but I will use the computer program Octave, which is free; for the realization of such practices.

The realization of the practices of the themes that are studied in the subject, these themes have been developed by steps and with explanatory examples. Seeing what commands to use for each type of case that is posed to us in the different topics that are studied in such subject.

Seeing in what cases is more difficult the use of this computer program and in which situation is better the use of Octave.

ÍNDICE:

1. Introducción.	Pág.5
2. Funciones.	Pág.5
2.1. Funciones reales de variables reales.	Pág.5
2.2. Límites de funciones reales de variables reales.	Pág.6
2.3. Continuidad de funciones de variables reales.	Pág.8
2.4. Representación gráfica de funciones reales de variables reales.	Pág.8
2.5. Ejercicios.	Pág.10
3. Derivación de funciones.	Pág.15
3.1. Derivada de una función real de una variable real.	Pág.15
3.2. Estudio de una función real de una variable real.	Pág.18
3.3. Ejercicios.	Pág.23
4. Integración de funciones.	Pág.30
4.1. Integración indefinida y la integración definida.	Pág.30
4.2. Aplicaciones de la integral definida. Área de una región plana.	Pág.32
4.3. Ejercicios.	Pág.36
5. Matrices.	Pág.41
5.1. Construcción de matrices.	Pág.41
5.2. Operaciones con matrices.	Pág.42
5.2.1. Suma de matrices.	Pág.42
5.2.2. Producto escalar por matriz.	Pág.42
5.2.3. Producto de matrices.	Pág.43
5.2.4. Inversa, transpuesta y determinante de una matriz.	Pág.44
5.2.5. Rango de una matriz.	Pág.45
5.3. Ejercicios.	Pág.45
6. Sistemas lineales. Vectores.	Pág.47
6.1. Sistemas de ecuaciones lineales.	Pág.47
6.2. Ejercicios.	Pág.52
7. Diagonalización de matrices.	Pág.54
7.1. Diagonalización de una matriz.	Pág.54
7.2. Ejercicios.	Pág.55
8. Conclusiones.	Pág.57
9. Bibliografía.	Pág.58

1. INTRODUCCIÓN

En este trabajo de fin de grado sobre “Prácticas de Matemáticas I en Octave”, se intenta analizar el uso del programa gratuito Octave en las prácticas de la asignatura Matemáticas I que se imparte en el Grado de Administración y Dirección de Empresas, por lo que podemos realizar una comparación con el programa no gratuito que utiliza la Universidad de Jaén actualmente que es el “Wolfram Mathematica ®”.

Las prácticas habitualmente se están utilizando en “Wolfram Mathematica ®”, las iremos revisando con Octave, para llegar a la conclusiones.

Como podremos observar durante todo este TFG, la mayoría de los comandos que se usan en el programa Octave la mayoría de ellos necesitan tener instalado el paquete simbólico para su utilización, mientras que en el programa de “Wolfram Mathematica ®” no hace falta instalar ninguna ampliación o paquete del programa para la utilización de sus comandos. Éste sería unos de los inconvenientes que ya le encontraríamos al programa en cuestión, aunque como ventaja es que tanto el programa como el paquete que se utiliza son gratuitos, mientras el programa que “Wolfram Mathematica ®” no es gratuito.

2. FUNCIONES.

Vamos a ver mediante el programa de Octave, como ir calculando todo lo relacionado con el tema de funciones que se da en la asignatura de Matemáticas I, en el grado de Administración y dirección de empresas.

2.1. FUNCIONES REALES DE VARIABLES REALES.

En Octave podemos escribir cualquier función de la siguiente manera, vamos a realizar como

ejemplo la siguiente función, $f(x) = \frac{x^2+1}{x-1}$:

```
>> f = @(x) (x^2+1)/(x-1)
```

```
f =
```

```
@(x) (x ^ 2 + 1) / (x - 1)
```

Una vez que hayamos definido la función en el Octave, podemos calcular cualquier punto, vamos a seguir con el ejemplo anterior y vamos a ver en qué puntos tiene la función $f(x)$ en los puntos 0; 2 y 1:

```
>> f(0)
```

```
ans = -1
```

```
>> f(2)
```

```
ans = 5
>> f(1)
warning: division by zero
warning: called from
    @<anonymous> at line 1 column 17
ans = Inf
```

Pues como observamos la función $f(x)$, en el punto 1; el programa avisa de que hay un posible error, ya que la función en este punto no está definida en su dominio; por lo que el programa nos da como solución infinito.

También podemos calcular funciones a trozos o funciones definidas por intervalos; las podemos definir en el programa de la misma forma que lo hemos realizado anteriormente, veamos la siguiente función que está definida en el intervalo $(-\infty, 2)$:

$$g(x) = \begin{cases} 1 - x; & x \leq 0 \\ \sqrt{x}; & 0 < x < 2 \end{cases}$$

```
>> g = @(x)(1-x).*(x<=0)+(sqrt(x)).*(0<x).*(x<2)
g =
    @(x) (1 - x) .* (x <= 0) + (sqrt (x)) .* (0 < x) .* (x < 2)
```

Como se ha indicado anteriormente, vamos a ver qué pasa cuando a la función le damos los siguientes valores -1, 0, 1:

```
>> g(-1)
ans = 2
>> g(0)
ans = 1
>> g(1)
ans = 1
```

2.2. LÍMITES DE FUNCIONES REALES DE VARIABLES REALES.

Para poder calcular operaciones de límites debemos tener instalado el paquete symbolic de Octave, ya que el programa sin la instalación de este paquete no nos puede calcular por sí solo los límites.

En matemáticas, uno de sus conceptos más importantes son los límites; estos sirven para decirnos el comportamiento que tiene una función y es clave para las derivadas.

Para calcular este tipo de operaciones debemos cargar en el programa anteriormente el paquete symbolic, escribiendo el comando “pkg load symbolic”.

Después de ver realizado la carga del paquete symbolic, tenemos que indicarle al programa cuantas variables simbólicas o incógnitas, tenemos para que haga los cálculos previstos, en este caso debemos escribir “syms x”, ya que nosotros por ahora solo tenemos una variable simbólica. Veamos los siguientes ejemplos:

$$\lim \left(\frac{\tan(x)}{x} \right), x \rightarrow 0$$

```
>> pkg load symbolic
```

```
>> syms x
```

Symbolic pkg v2.7.1: Python communication link active, SymPy v1.3.

```
>> f = tan(x)/x;
```

```
>> limit(f,x,0)
```

```
ans = (sym) 1
```

$$\lim \left(\frac{1}{x} \right), x \rightarrow \infty$$

En la realización de este ejemplo, ya no hemos tenido que cargar el paquete symbolic, ni indicar con cuantas variables simbólicas vamos a trabajar; directamente ponemos el límite para que nos lo calcule. En este caso ponemos la función directamente en el comando, sin que el programa defina la función:

```
>> limit (1/x,x,inf)
```

```
ans = (sym) 0
```

También podemos realizar el cálculo de límites laterales, para indicar la izquierda pondremos en el comando “left” y en el caso de la derecha “right”; veamos como sería con el siguiente ejemplo:

$$\limite \left(\frac{1}{x} \right), X \rightarrow 0$$

```
>> limit (1/x,x,0,'left')
```

```
ans = (sym) -oo
```

```
>> limit (1/x,x,0,'right')
```

```
ans = (sym) oo
```

2.3. CONTINUIDAD DE FUNCIONES DE VARIABLES REALES.

Para decir que una función es continua en un punto, tenemos que realizar el límite de esa función en dicho punto; además para que la función sea continua se debe cumplir lo siguiente:

$$\lim_{x \rightarrow a} f(x) = f(a)$$

Veamos la siguiente función, en el punto 2:

```
>> clear
>> syms x
>> f = @(x)(x^3-1)/(x^2-1);
>> f(2)
ans = 2.3333
>> limit(f,x,2)
ans = (sym) 7/3
```

En este ejemplo, he utilizado el comando “clear”, que sirve para borrar las funciones anteriores, ya que anteriormente se había indicado a otra función con el nombre de f. Pero como se puede ver, ya no habría que cargar el paquete, pero si habría que indicar las variables simbólicas que se utilizarían.

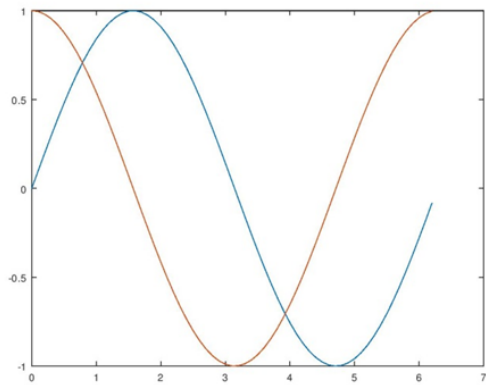
Como podemos observar la función es continua en el punto 2.

2.4. REPRESENTACIÓN GRÁFICA DE FUNCIONES REALES DE VARIABLES REALES.

Octave se puede realizar las gráficas de las funciones, vamos a ver algunos ejemplos:

```
>> pkg load symbolic
>> syms x
Symbolic pkg v2.7.1: Python communication link active, SymPy v1.3.
>> x = 0:0.1:2*pi;
>> f1= sin(x);
>> f2= cos(x);
>> plot(x,f1,x,f2);
```

Obtenemos una ventana aparte de donde escribimos los comandos para que nos realice la gráfica, donde obtenemos:



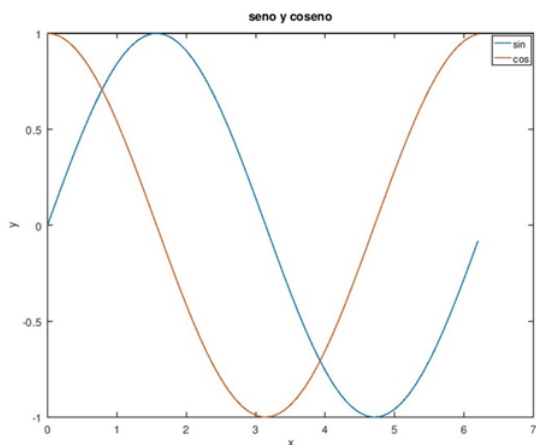
A las gráficas podemos ponerle más opciones, para que sea de forma más completa.

Podemos usar los siguientes comandos:

- Para poner nombre al eje y: “ylabel(‘nombre’)”.
- Para poner nombre al eje x: “xlabel(‘nombre’)”.
- Para ponerle título a la gráfica: “title(‘nombre’)”.
- Para poner leyenda a la gráfica: “legend(‘nombre’)”.

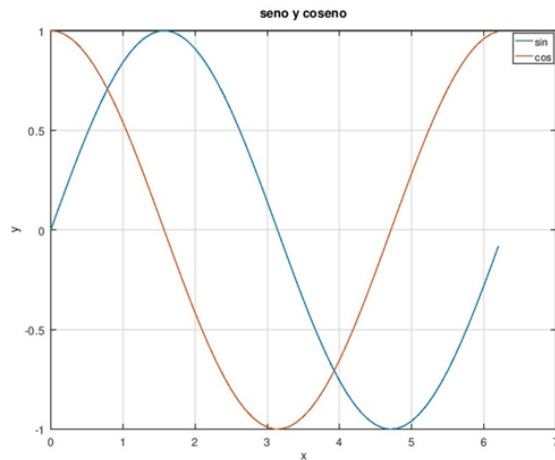
Por ejemplo:

```
>> clear
>> syms x
>> x = 0:0.1:2*pi;
>> f1= sin(x);
>> f2= cos(x);
>> plot(x,f1,x,f2);
>> ylabel('y');
>> xlabel('x');
>> title('seno y coseno')
>> legend('sin','cos')
```



En el caso de querer cuadricular la gráfica anterior, usamos el comando “grid on”:

>> grid on



2.5. EJERCICIOS.

1) Calcular los siguientes límites:

- $\lim_{x \rightarrow 0^+} 5^{1/x}$

>> limit(5^(1/x),x,0,'right')

ans = (sym) oo

- $\lim_{x \rightarrow 0^-} 5^{1/x}$

>> limit(5^(1/x),x,0,'left')

ans = (sym) 0

- $\lim_{x \rightarrow 2} \frac{x^2-4}{x-2}$

>> limit((x^2-4)/(x-2),x,2)

ans = (sym) 4

- $\lim_{x \rightarrow +\infty} \sqrt{x^2-x} - (x+1)$

>> limit(sqrt(x^2-x)-(x+1),x,inf)

ans = (sym) -3/2

- $\lim_{x \rightarrow +\infty} \left(1 + \frac{2^{-x}}{x}\right)^x$

>> limit((1+((2^(-x))/x))^x,x,inf)

ans = (sym) 1

2) Representa gráficamente las siguientes funciones:

- $f(x) = \frac{x^3}{x^2-1}$

```
>> clear
```

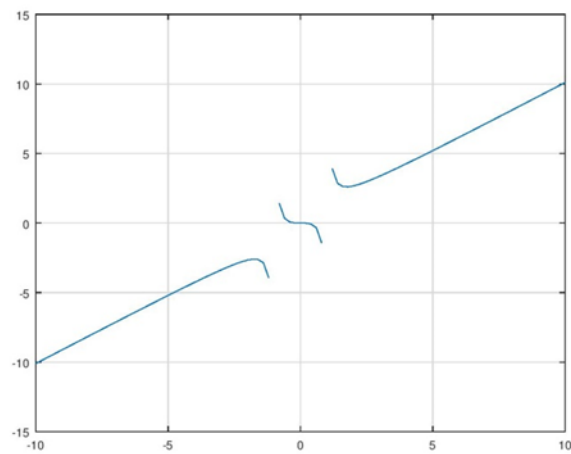
```
>> syms x
```

```
>> x = linspace(-10,10,101);
```

```
>> y = (x.^3)./(x.^2-1);
```

```
>> plot(x,y)
```

```
>> grid on
```



- $g(x) = \frac{x}{\sqrt{x^2+2}}$

```
>> clear
```

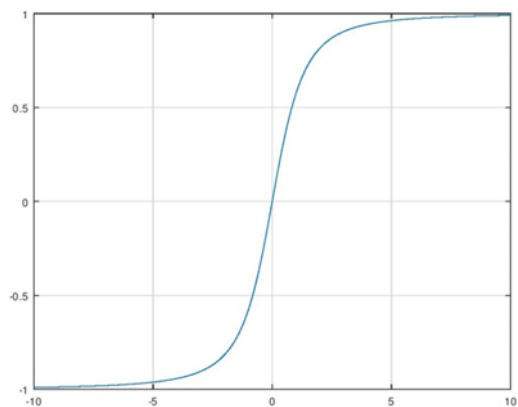
```
>> syms x
```

```
>> x = linspace(-10,10,101);
```

```
>> y = x./(sqrt(x.^2+2));
```

```
>> plot(x,y)
```

```
>> grid on
```



- $h(x) = x^4 - 12x^3 + 48x^2 - 64x$

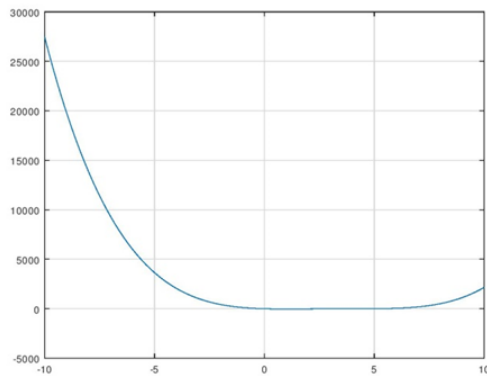
```
>> clear
```

```
>> x = linspace(-10,10,101);
```

```
>> y = x.^4-12*x.^3+48*x.^2-64*x;
```

```
>> plot(x,y)
```

```
>> grid on
```



3) La función f , con dominio en todo $\mathbb{R}$:

$$f(x) = \begin{cases} -x^2 + 2x; & x < 3 \\ -x; & 3 \leq x \leq 5 \\ 1; & 5 < x \end{cases}$$

- Calcula $f(-1)$, $f(5)$ y $f(6)$

```
>> f = @(x)(-x.^2+2*x).*(x<3)+(-x).*(3<=x).*(x<=5)+(1).*(5<x);
```

```
>> f(-1)
```

```
ans = -3
```

```
>> f(5)
```

```
ans = -5
```

```
>> f(6)
```

```
ans = 1
```

- Estudia la continuidad de f .

```
>> f(3)
```

```
ans = -3
```

```
>> f(5)
```

```
ans = -5
```

Si intentamos realizar el límite como lo hemos realizado hasta ahora el programa Octave, en la forma que está definida la función, nos da un mensaje de error. Por lo que para realizar el análisis en funciones a trozos debemos realizarlo trozo por trozo:

```
>> limit (-x.^2+2*x,x,3,'left')
```

```
ans = (sym) -3
```

```
>> limit(-x,x,3,'right')
```

```
ans = (sym) -3
```

```
>> limit(-x,x,5,'left')
```

```
ans = (sym) -5
```

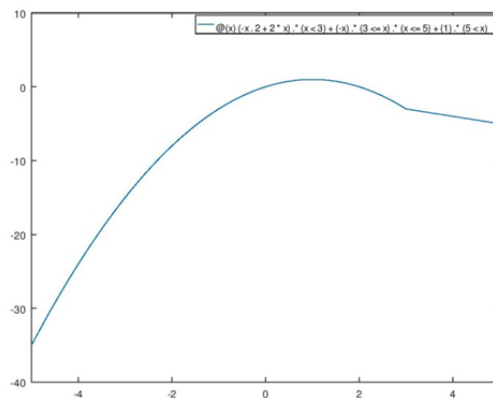
```
>> limit(1,x,5,'right')
```

```
ans = (sym) 1
```

La función es continua en el punto 3, ya que sus límites laterales son iguales e igual a $f(3)$; y no es continua en punto 5, porque sus límites laterales son distintos entre sí y por tanto no existe límite de f en el punto 5. En el resto de puntos, la función f es continua.

- Representa gráficamente la función.

```
>> fplot(@(x)(-x.^2+2*x).*(x<3)+(-x).*(3<=x).*(x<=5)+(1).*(5<x),[-5,5])
```



- 4) Comprueba que las siguientes ecuaciones tienen una raíz y calcúlala:

- $x^3 + 4x^2 = 10$ en el intervalo $[1,2]$.

```
>> syms x
```

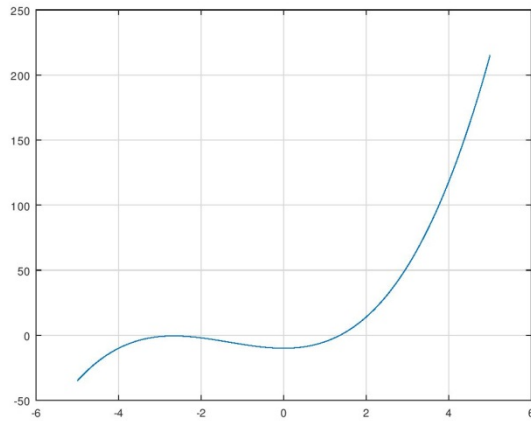
Symbolic pkg v2.7.1: Python communication link active, SymPy v1.3.

```
>> x = linspace(-5,5,101);
```

```
>> y = x.^3+4*x.^2-10;
```

```
>> plot(x,y)
```

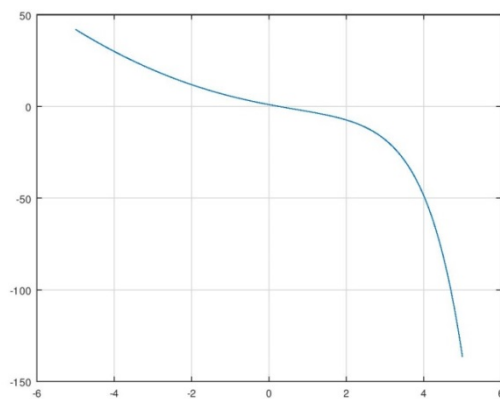
```
>> grid on
```



```
>> syms x
>> f=@(x)x^3+4*x^2-10;
>> fsolve(f,2)
ans = 1.3652
```

- $3x = 2 + x^2 - e^x$ en el intervalo $[-2, 2]$.

```
>> syms x
>> x = linspace(-5,5,101);
>> y = 2+x.^2-exp(x)-3*x;
>> plot(x,y)
>> grid on
```



```
>> clear
>> g=@(x)2+x^2-exp(x)-3*x;
>> fsolve(g,2)
ans = 0.25753
```

3. DERIVACIÓN DE FUNCIONES.

En este tema de matemáticas, es donde se comienza el cálculo diferencial, que es el concepto de derivada, ya que esta nos permite medir la variación de una función en un intervalo o entorno a un punto en concreto.

Vamos a ver mediante el programa de Octave, como ir calculando todo lo relacionado con este tema.

3.1. DERIVADA DE UNA FUNCIÓN REAL DE UNA VARIABLE REAL.

Podemos calcular la derivada de una función mediante su definición, que es la realización del siguiente límite:

$$f'(a) = \lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h}$$

Veamos el siguiente ejemplo:

- ❖ En este caso, definimos la función de forma diferente, ya que de esta manera es más fácil para realizar el cálculo de la definición de la derivada.

```
>> pkg load symbolic
```

```
>> syms x h
```

Symbolic pkg v2.7.1: Python communication link active, SymPy v1.3.

```
>> f(x)=(x^2+1)/(x-1);
```

```
>> limit((f(x+h)-f(x))/h,h,0)
```

```
ans = (sym)
```

$$\frac{x^2 - 2x - 1}{x^2 - 2x + 1}$$

Veamos el siguiente ejemplo, en el punto π :

$$g(x) = \frac{\sin(x)}{x}$$

```
>> g(x)= sin(x)/x;
```

```
>> limit(g(x),x,pi)
```

```
ans = (sym) 0
```

Como hemos visto la función $g(x)$ es continua en el punto π , por lo que ahora comprobamos si es derivable en dicho punto:

```
>> limit((g(pi+h)-g(pi))/h,h,0)
```

```
ans = (sym)
```

$$\frac{-1}{\pi}$$

Por lo que podemos decir, con los resultados obtenidos es que la función es derivable en dicho punto.

Octave dispone de un comando para la realización directa de derivadas, dicho comando es “diff(función)”. Veamos algunos ejemplos:

$$f(x) = \sqrt{x}$$

```
>> f = sqrt(x);
```

```
>> diff(f)
```

```
ans = (sym)
```

$$\frac{1}{2\sqrt{x}}$$

$$f2(x) = 2\arctan * \sqrt{\frac{1 - \cos(x)}{1 + \cos(x)}}$$

```
>> f2=2*atan(sqrt((1-cos(x))/(1+cos(x))));
```

```
>> diff(f2)
```

```
ans = (sym)
```

$$\frac{2 * \frac{\sqrt{-\cos(x) + 1}}{\sqrt{\cos(x) + 1}} * \frac{(-\cos(x) + 1) * \sin(x)}{2 * (\cos(x) + 1)} + \frac{\sin(x)}{2 * (\cos(x) + 1)}}{\frac{\sqrt{-\cos(x) + 1}}{\sqrt{\cos(x) + 1}} + 1 * (-\cos(x) + 1)}$$

Como podemos observar, el resultado de esta derivada es bastante extensa, por lo que la podemos simplificar con el comando “simplify”:

```
>> simplify(diff(f2))
```

```
ans = (sym)
```

$$\frac{-\frac{\sqrt{-(\cos(x) - 1)}}{\sqrt{\cos(x) + 1}} * \sin(x)}{\cos(x) - 1}$$

También podemos calcular la derivada de una función a trozos o por intervalos, veamos el siguiente ejemplo:

$$f(x) = \begin{cases} 1 + x; & x \leq 0 \\ \cos(x); & x > 0 \end{cases}$$

```
>> syms x
>> f = (1+x).*(x<=0)+(cos(x)).*(x>0);
>> diff(f)
ans = (sym)
```

$$(x + 1) \frac{d}{dx} (x \leq 0) - \sin(x) (x > 0) + \cos(x) \frac{d}{dx} (x > 0) + (x \leq 0)$$

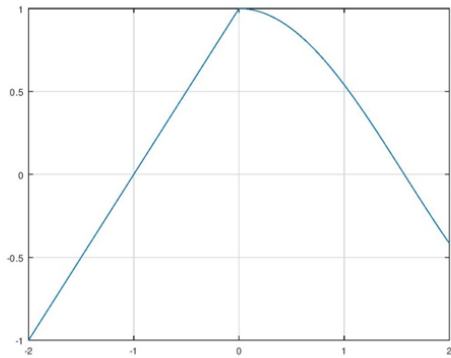
Como podemos observar, el programa no nos calcula bien, por lo que realizaremos la derivada de la función trozo a trozo.

```
>> syms x
>> f1 = 1+x;
>> f2 = cos(x);
>> diff(f1)
ans = (sym) 1
>> diff(f2)
ans = (sym) -sin(x)
```

Para los valores inferiores al valor cero su derivada es 1 y para los valores superiores a cero su derivada es la función $-\sin(x)$

Realizamos la representación gráfica de la función:

```
>> clear
>> syms x
>> x = linspace(-2,2,101);
>> f = (1+x).*(x<=0)+(cos(x)).*(x>0);
>> plot(x,f);
>> grid on
```



Se observa que Octave no nos avisa de que la función no es derivable, en el punto cero. Por lo que debemos realizar los límites laterales de la función en dicho punto:

```
> pkg load symbolic
```

```
>> syms x h
```

```
>> f(x) = (1+x).*(x<=0)+(cos(x)).*(x>0);
```

A la hora de calcular las derivadas de funciones a trozos, nos da directamente un mensaje de error, por lo que tenemos que realizar trozo a trozo:

```
>> syms x h
```

```
>> f1(x)= 1+x;
```

```
>> f2(x)= cos(x);
```

```
>> limit((f1(0+h)-f1(0))/h,h,0,'left')
```

```
ans = (sym) 1
```

```
>> limit((f2(0+h)-f2(0))/h,h,0,'right')
```

```
ans = (sym) 0
```

Como los límites que hemos calculado, son distintos entre ellos, la función no es derivable en dicho punto.

3.2. ESTUDIO DE UNA FUNCIÓN REAL DE UNA VARIABLE REAL.

Para calcular los extremos relativos de una función, es importante la aplicación de la derivada.

Vamos a ver un ejemplo para resolver el problema de calcular los máximos y mínimos:

$$f(x) = \frac{x}{x^2 + 1}$$

```
>> pkg load symbolic
```

```
>> syms x
```

```
>> f = x/(x^2+1)
```

```
f = (sym)
```

$$\frac{x}{x^2 + 1}$$

Debemos tener en cuenta que el dominio de esta función es $\mathbb{R}$, por lo que lo vamos a comprobar, con el comando “solve”:

```
>> solve(x^2+1==0)
```

```
ans = (sym 2x1 matrix)
```

```
[-I]
```

```
[ ]
```

```
[I ]
```

La solución que obtenemos son números imaginarios, por lo que podemos decir que el dominio de toda la función es todo $\mathbb{R}$.

Para calcular los puntos de corte con los ejes, cuando x es igual a cero:

```
>> solve(f==0,x)
```

```
ans = (sym) 0
```

```
>> f(0)
```

```
ans = (sym) 0
```

Podemos ver que el punto de corte con los ejes es el $(0,0)$. Por lo que podemos decir lo siguiente:

- Asíntotas verticales no tiene ya que el denominador no se anula.
- Asíntotas horizontales, vamos a realizar el límite hacia el infinito a ver qué ocurre:

```
>> limit(f(x),x,inf)
```

```
ans = (sym) 0
```

Por lo que podemos decir que su asíntota horizontal es el eje x ($y = 0$).

- Asíntotas oblicuas, realizamos el límite siguiente:

```
>> limit(f(x)/x,x,inf)
```

```
ans = (sym) 0
```

Por lo que podemos decir que no existen asíntotas oblicuas en esta función.

También podemos ver en Octave, los puntos que anulan de derivada primera de la función, para así saber el crecimiento y decrecimiento de la función, al igual que hemos calculado los máximos y los mínimos:

```
>> diff(f(x))
```

```
ans = (sym)
```

$$-\frac{2x^2}{\sqrt{x^2+1}} + \frac{1}{x^2+1}$$

```
>> solve(diff(f)==0)
```

```
ans = (sym 2x1 matrix)
```

```
[-1]
```

```
[ ]
```

```
[1 ]
```

Como vemos nos indica que la derivada se anula en los puntos -1 y 1, por lo que vamos a estudiar los signos de la derivada segunda en los puntos que hemos obtenido:

```
>> f(-1)
```

```
ans = (sym) -1/2
```

```
>> diff(f,2)(-1)
```

```
ans = (sym) 1/2
```

Por lo que podemos decir que la función tiene un mínimo relativo en el punto (-1, -1/2).

```
>> f(1)
```

```
ans = (sym) 1/2
```

```
>> diff(f,2)(1)
```

```
ans = (sym) -1/2
```

Por lo que podemos indicar que la función tiene un máximo relativo en el punto (1, 1/2).

Para realizar un estudio de una función a fondo, debemos ver el crecimiento y decrecimiento, mediante el signo de la derivada primera en los siguientes intervalos de la función $(-\infty, -1)$, $(-1, 1)$ y $(1, +\infty)$:

```
>> diff(f)(-2)
```

```
ans = (sym) -3/25
```

```
>> diff(f)(0)
```

```
ans = (sym) 1
```

```
>> diff(f)(2)
```

```
ans = (sym) -3/25
```

Por lo que podemos decir que la función es creciente en el intervalo $(-1, 1)$ y es decreciente en los intervalos $(-\infty, -1) \cup (1, +\infty)$.

Ahora vamos a buscar la existencia de los puntos de inflexión de la función:

```
>> solve(diff(f,2)==0)
```

ans = (sym 3x1 matrix)

```
[ 0 ]
[   ]
[ -\sqrt{3} ]
[   ]
[ \sqrt{3} ]
```

Los valores que hemos obtenido, los tenemos que sustituir en la derivada tercera de la función:

```
>> diff(f,3)(0)
```

ans = (sym) -6

```
>> diff(f,3)(sqrt(3))
```

warning: passing floating-point values to sym is dangerous, see "help sym"

warning: called from

double_to_sym_heuristic at line 50 column 7

sym at line 379 column 13

subs at line 175 column 13

subsref at line 46 column 11

ans = (sym) 3/16

```
>> diff(f,3)(-sqrt(3))
```

warning: passing floating-point values to sym is dangerous, see "help sym"

warning: called from

double_to_sym_heuristic at line 50 column 7

sym at line 379 column 13

subs at line 175 column 13

subsref at line 46 column 11

ans = (sym)

```

/
|
|      4      2      \
|      10023806115968*pi      8954912*pi      |
|----- - 1 + -----|
|      2      /      2\      |
|      /      |      1119364*pi      |
|      |      |      3682561*|1 + -----|
|      |      |      \      3682561      /
|      \      /
|-----|
|
|      2      2\
|      /      |
|      |      |
|      |1 + -----|
|      \      3682561      /
|-----|
|
|      2\
|      /
|      |
|      |1 + -----|
|      \      3682561      /
|-----|
```

Como vemos el resultado que nos ha dado el programa en esta derivada, vamos a simplificar el resultado que nos ha dado Octave:

```
>> simplify(diff(f,3)(-sqrt(3)))
```

warning: passing floating-point values to sym is dangerous, see "help sym"

warning: called from

double_to_sym_heuristic at line 50 column 7

sym at line 379 column 13

subs at line 175 column 13

subsref at line 46 column 11

```
ans = (sym)
```

$$\frac{-101951547006570264090777696\pi^4 - 11034459074}{183907651244040778781475841 + 223604827322218925687800336\pi^2 + 10195154700654244672688855046 + 2012443445899970331190203024\pi^2}$$

$$\frac{0264090777696\pi^4 + 1569948266414335654134016\pi^8 + 2065969696332932744245273}{}$$

Como podemos ver las derivadas terceras son distintas a cero, por lo que todos ellos son puntos de inflexión en la función.

Para ver la concavidad y convexidad de la función se haría de forma similar a la que hemos empleado hasta ahora, a partir de la derivada segunda de la función, por lo que los intervalos determinados que obtendríamos son los siguientes, $(-\infty, -\sqrt{3})$, $(-\sqrt{3}, 0)$, $(0, \sqrt{3})$ y $(\sqrt{3}, +\infty)$.

Para ver en que intervalos la concavidad es positiva o negativa, realizamos las derivadas segundas en un punto de esos intervalos:

```
>> diff(f,2)(-5)
```

```
ans = (sym)
```

$$\frac{-55}{4394}$$

```
>> diff(f,2)(-1)
```

```
ans = (sym) 1/2
```

```
>> diff(f,2)(1)
```

```
ans = (sym) -1/2
```

```
>> diff(f,2)(5)
```

```
ans = (sym)
```

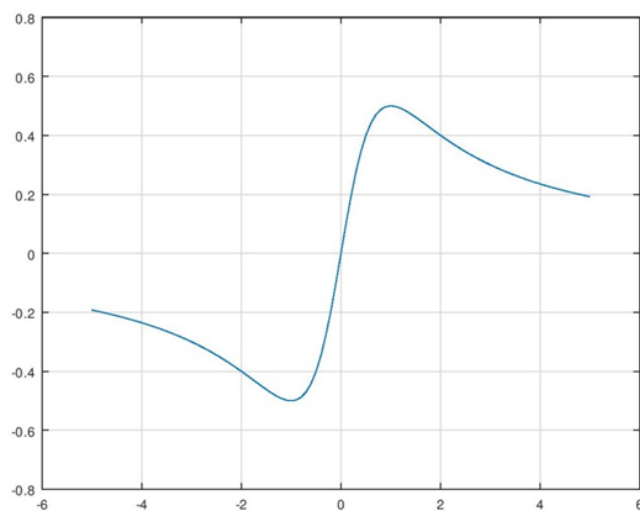
55

4394

Por lo que podemos indicar que la concavidad es positiva en los intervalos $(-\sqrt{3}, 0) \cup (\sqrt{3}, +\infty)$ y es negativa en los intervalos $(-\infty, -\sqrt{3}) \cup (0, \sqrt{3})$.

Realizamos la representación gráfica de la función para comprobar que los resultados obtenidos son los correctos.

```
>> clear  
>> syms x  
>> x = linspace(-5,5,101);  
>> f = x./(x.^2+1);  
>> plot(x,f)  
>> grid on
```



3.3. EJERCICIOS.

1) Calcular a partir de la definición de la derivada, las siguientes funciones:

```
- f(x) = ln(x)  
>> pkg load symbolic  
>> syms x h  
>> f(x)= log(x);  
>> limit((f(x+h)-f(x))/h,h,0)  
ans = (sym)
```

1
-
x

```
- g(x) = cos(x)
>> g(x)= cos(x);
>> limit((g(x+h)-g(x))/h,h,0)
ans = (sym)
-sin(x)
```

2) Estudia la derivabilidad de las funciones:

$$f(x) = \begin{cases} 1 - x^2; & -3 < x \leq 2 \\ -x - 1; & x > 2 \end{cases}$$

Como es una función a trozos debemos realizarlo por trozos:

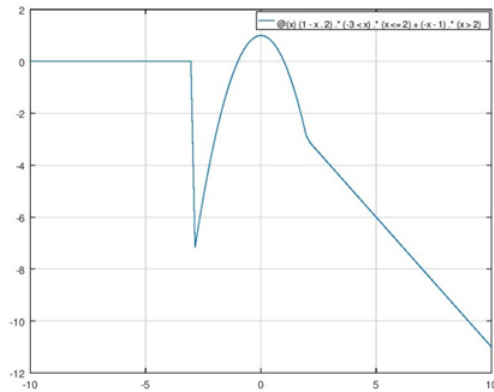
```
>> pkg load symbolic
>> syms x
>> f1(x)= 1-x.^2;
>> f2(x)= -x-1;
>> diff(f1(x))
ans = (sym) -2*x
>> diff(f2(x))
ans = (sym) -1
```

Realizamos la derivada en el punto que engancha a las dos funciones, que en este caso es $x = 2$:

```
>> limit((f1(2+h)-f1(2))/h,h,0,'left')
ans = (sym) -4
>> limit((f2(2+h)-f2(2))/h,h,0,'right')
ans = (sym) -1>> f1(2)
ans = (sym) -3
```

Como los límites laterales en el punto $x = 2$ son distintos entre sí, por lo tanto la función no es derivable en $x = 2$.

```
>> fplot( @(x)(1-x.^2).*(-3<x).*(x<=2)+(-x-1).*(x>2),[-10,10])
>> grid on
```

$$- \begin{cases} x^2 \sin\left(\frac{1}{x}\right); & x \neq 0 \\ 0; & x = 0 \end{cases}$$

```
>> syms x h
>> g(x)=(x.^2*sin(1/x));
>> limit((g(h)-0)/h,h,0,'right')
ans = (sym) 0
>> limit((g(h)-0)/h,h,0,'left')
ans = (sym) 0
```

La función es derivable en todos los valores de x , ya que las derivadas laterales en el punto $x = 0$ son iguales.

3) Calcule y represente la recta tangente:

- En el punto $x = 0.3$ de la función $\frac{x^3}{x^2-1}$ en el intervalo $[-1,1]$

```
>> syms x h
>> f = (x^3)/(x^2-1);
>> subs(f,x,0.3)
```

warning: passing floating-point values to sym is dangerous, see "help sym"

warning: called from

double_to_sym_heuristic at line 50 column 7

sym at line 379 column 13

subs at line 178 column 9

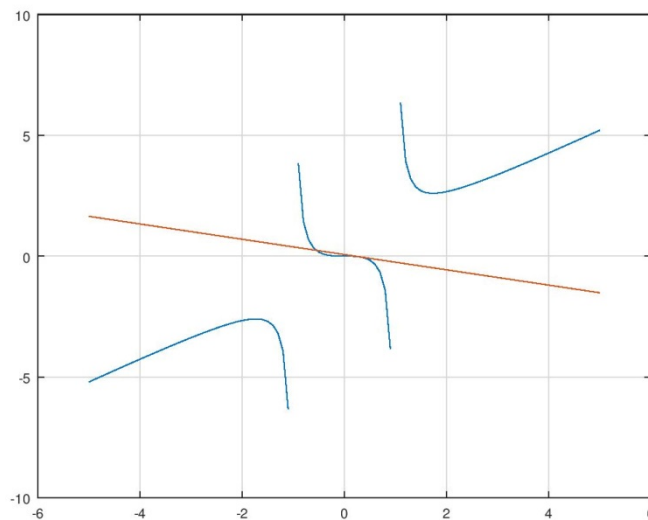
```
ans = (sym)
```

```
-27
----
910
```

```
>> limit((f((0.3)+h)-f(0.3))/h,h,0)
warning: passing floating-point values to sym is dangerous, see "help sym"
warning: called from
    double_to_sym_heuristic at line 50 column 7
        sym at line 379 column 13
        plus at line 63 column 5
warning: passing floating-point values to sym is dangerous, see "help sym"
warning: called from
    double_to_sym_heuristic at line 50 column 7
        sym at line 379 column 13
        minus at line 57 column 5
ans = (sym)
-2619
-----
8281
```

La recta tangente sería la siguiente:

$$y + 27/910 = -2619/8281 (x - 0,3)$$



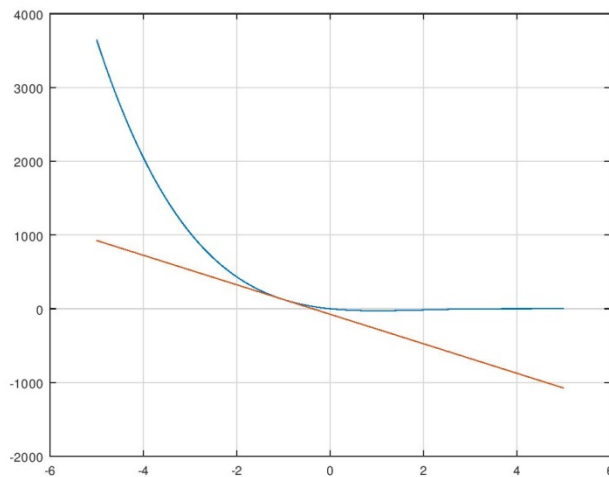
- En el punto $x = -1$ de la función $x^4 - 12x^3 + 48x^2 - 64x$

```
>> syms x h
>> syms x h
>> f=@(x)(x^4)-(12*x^3)+(48*x^2)-(64*x);
```

```

>> subs(f,x,-1)
ans = (sym) 125
>> limit((f(-1+h)-f(-1))/h,h,0)
ans = (sym) -200
La recta tangente seria la siguiente:
y - 125= -200 (x+1)
>> syms x
>> x = linspace(-5,5,101);
>> y1 =(x.^4)-(12*x.^3)+(48*x.^2)-(64*x);
>> y2 = -(200*x)-200+125;
>> plot(x,y1,x,y2)
>> grid on

```



4) Sea la función $f(x) = -x^3 + 3x$, que su dominio es todo $\mathbb{R}$:

- Determina los puntos de corte de la gráfica de f con el eje de coordenadas.

```

>> syms x
>> f(x)= -x^3+3*x
f(x) = (symfun)
      3
    - x  + 3*x
>> solve(f(x)==0,x)
ans = (sym 3x1 matrix)

```

$$\begin{bmatrix} 0 \\ \sqrt{3} \\ -\sqrt{3} \\ \sqrt{3} \end{bmatrix}$$

```
>> f(0)
```

```
ans = (sym) 0
```

```
>> f(-(sqrt(3)))
```

warning: passing floating-point values to sym is dangerous, see "help sym"

warning: called from

double_to_sym_heuristic at line 50 column 7

sym at line 379 column 13

subs at line 175 column 13

subsref at line 46 column 11

```
ans = (sym)
```

$$-\frac{3174\pi}{1919} + \frac{1184287112\pi^3}{7066834559}$$

```
>> f(sqrt(3))
```

warning: passing floating-point values to sym is dangerous, see "help sym"

warning: called from

double_to_sym_heuristic at line 50 column 7

sym at line 379 column 13

subs at line 175 column 13

subsref at line 46 column 11

```
ans = (sym) 0
```

- Calcula los extremos relativos y los puntos de inflexión de f.

```
>> diff(f(x))
```

```
ans = (sym)
```

$$-3x^2 + 3$$

```
>> solve(diff(f(x))==0)
```

```
ans = (sym 2x1 matrix)
```

```
[-1]
```

```
[ ]
```

```
[1 ]
```

```
>> f(-1)
```

```
ans = (sym) -2
```

```
>> diff(f,2)(-1)
```

```
ans = (sym) 6
```

La función tiene un mínimo relativo en este punto, ya que la derivada segunda es positiva.

```
>> f(1)
```

```
ans = (sym) 2
```

```
>> diff(f,2)(1)
```

```
ans = (sym) -6
```

La función tiene un máximo relativo en este punto, ya que la derivada segunda es negativa.

```
>> solve(diff(f(x),2)==0)
```

```
ans = (sym) 0
```

```
>> diff(f,3)(0)
```

```
ans = (sym) -6
```

Como la derivada tercera es distinta de cero, por lo que tenemos un punto de inflexión en cero.

- Representa gráficamente la función.

```
>> clear
```

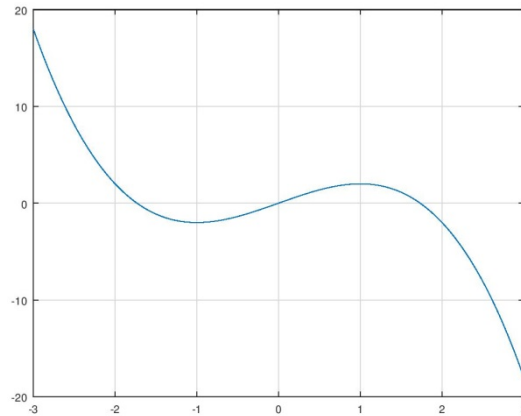
```
>> syms x
```

```
>> x = linspace(-3,3,101);
```

```
>> f = -x.^3+3*x;
```

```
>> plot(x,f)
```

```
>> grid on
```



4. INTEGRACIÓN DE FUNCIONES.

4.1. INTEGRACIÓN INDEFINIDA Y LA INTEGRACIÓN DEFINIDA.

La integración es una herramienta bastante importante, ya que esta resuelve problemas variados, tanto en matemáticas y en el resto de ciencias. Por lo que podemos decir que en Octave, encontramos el siguiente comando para la resolución de una integración.

Tenemos como función: $f(x) = \sin(x)$, para el cálculo de integrales tenemos que cargar en el programa el paquete simbólico.

$$\int f(x) = \int \sin(x) dx$$

```
>> pkg load symbolic
```

```
>> syms x
```

Symbolic pkg v2.7.1: Python communication link active, SymPy v1.3.

```
>> f(x) = sin(x);
```

```
>> int(f)
```

```
ans(x) = (symfun) -cos(x)
```

Vamos a ver el caso en que la integral está acotada entre dos puntos concretos, en este caso en el intervalo $[0, \pi/2]$:

$$\int_0^{\pi/2} \cos(x) dx$$

```
>> f1(x)=cos(x);
```

```
>> int(f1,0,pi/2)
```

warning: passing floating-point values to sym is dangerous, see "help sym"

warning: called from

double_to_sym_heuristic at line 50 column 7

sym at line 379 column 13

int at line 138 column 7

int at line 67 column 5

```
ans = (sym) 1
```

Ahora vamos a ver lo que ocurre cuando en una integral definida, tiene puntos singulares:

$$\int_{-1}^1 \frac{1}{x^2} dx$$

```
>> clear
```

```
>> syms x
```

```
>> f = 1/x^2;
```

```
>> int(f,-1,1)
```

```
ans = (sym) oo
```

Octave nos da que el resultado de la integral es infinito, para poder ver mejor lo que ocurre, vamos a realizar la gráfica de esta función:

```
>> clear
```

```
>> pkg load symbolic
```

```
>> syms x
```

```
>> x = -1:0.1:1;
```

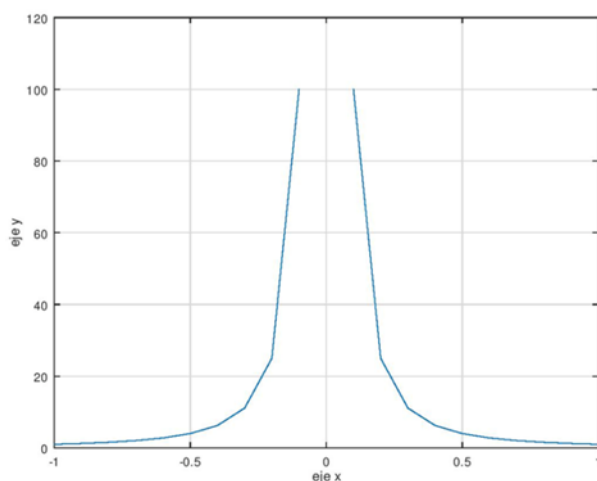
```
>> f = x.^(-2);
```

```
>> plot(x,f)
```

```
>> xlabel('eje x')
```

```
>> ylabel('eje y')
```

```
>> grid on
```



Como podemos ver en esos puntos la función tiene a infinito.

Vamos a realizar algunas integrales más para ver los resultados que nos da Octave en ellas:

$$\int_0^4 \sqrt{16 - x^2} \, dx$$

```
>> clear
>> syms x
>> f = sqrt(16-x^2);
>> int(f,0,4)
ans = (sym) 4*pi
```

$$\int_0^1 e^{-x^2} \, dx$$

```
>> clear
>> syms x
>> g = exp(-x^2);
>> int(g,0,1)
ans = (sym)
```

$$\frac{\sqrt{\pi} \operatorname{erf}(1)}{2}$$

Por lo que podemos decir que esta integral no tiene una solución en Octave.

4.2.APLICACIONES DE LA INTEGRACION DEFINIDA. ÁREA DE UNA REGIÓN PLANA.

En este apartado vamos a ver, como calcular el área que queda determinada entre dos funciones y los ejes de abscisas. Vemos el siguiente ejemplo:

$$\int_{-1}^1 (x^2 + 1) \, dx$$

```
>> pkg load symbolic
>> syms x
Symbolic pkg v2.7.1: Python communication link active, SymPy v1.3.
>> f = x^2+1;
>> int(f,(-1),1)
ans = (sym) 8/3
```

Ahora vamos a ver qué pasa si queremos calcular el área que determina la curva del $\cos(x)$ con el eje de abscisas en el intervalo $[0, \pi]$:

```
>> h = cos (x);
>> int(h,0,pi)
```

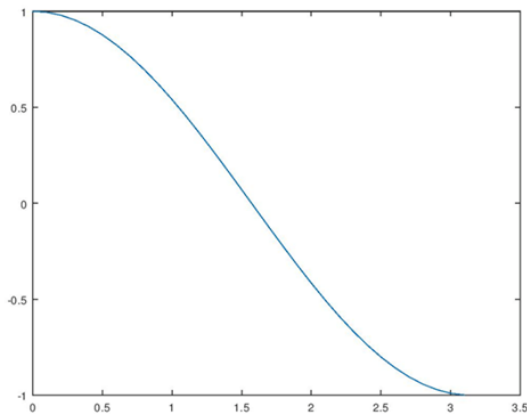


```
ans = (sym) 0
```

Realizamos la representación gráfica de la función, ya que el resultado que hemos obtenido de la integración de la función nos indica que la función en ese intervalo no es positiva, por lo que para calcular el área de la función, tenemos que tener en cuenta los intervalos que sean positivos y negativos:

```
>> x = 0:0.1:pi;
```

```
>> plot(x,cos(x))
```



Para poder determinar el área que determina la función con el eje de abscisas, habrá que considerar las dos integrales definidas:

$$\int_0^{\pi/2} \cos(x) dx$$

```
>> int(h,0,pi/2)
```

warning: passing floating-point values to sym is dangerous, see "help sym"

warning: called from

double_to_sym_heuristic at line 50 column 7

sym at line 379 column 13

int at line 138 column 7

```
ans = (sym) 1
```

$$\int_{\pi/2}^{\pi} \cos(x) dx$$

```
>> int(h,pi/2,pi)
```

warning: passing floating-point values to sym is dangerous, see "help sym"

warning: called from

double_to_sym_heuristic at line 50 column 7

sym at line 379 column 13

int at line 138 column 7

```
ans = (sym) -1
```

El resultado que hemos obtenido en esta última integral, es negativo, ya que en este tramo la función se encuentra por debajo del eje de abscisas. Como ya sabemos, el área se determina por la suma de los dos resultados anteriores, pero debemos tomarlos de forma absoluta:

```
>> area = abs(int(g,0,pi/2))+ abs(int(g,pi/2,pi))
```

```
warning: passing floating-point values to sym is dangerous, see "help sym"
```

```
warning: called from
```

```
double_to_sym_heuristic at line 50 column 7
```

```
sym at line 379 column 13
```

```
int at line 138 column 7
```

```
warning: passing floating-point values to sym is dangerous, see "help sym"
```

```
warning: called from
```

```
double_to_sym_heuristic at line 50 column 7
```

```
sym at line 379 column 13
```

```
int at line 138 column 7
```

```
area = (sym) 2
```

Cuando tengamos una función que no sea positiva, habrá que considerar como en este caso por separado los intervalos en los que la función sea positiva y en aquellos en los que sea negativa.

Planteamos el siguiente problema para calcular el área de los recintos acotados de dos curvas. Por ejemplo las funciones:

$$f = x^2 \qquad g = 2^x$$

Primero representamos gráficamente estas dos funciones:

```
>> clear
```

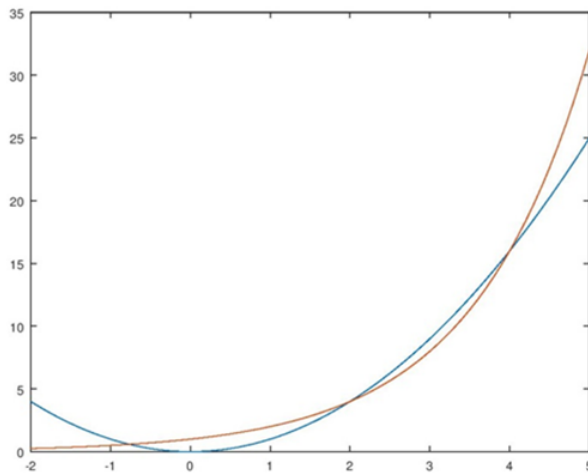
```
>> syms x
```

```
>> x = (-2):0.1:5;
```

```
>> f= x.^2;
```

```
>> g = 2.^x;
```

```
>> plot(x,f,x,g)
```



Por lo que podemos observar, son dos recintos acotados limitados por las gráficas de ambas funciones. Por lo que tenemos que calcular los puntos de corte entre las dos funciones:

```
>> clear
>> syms x
>> solve (x^2==2^x)
ans = (sym 2x1 matrix)
[      2      ]
[      ]
[      /-log(2)  \ ]
[ -2*LambertW|-----, -1| ]
[      \ 2      / ]
[-----]
[      log(2)      ]
```

El comando “solve” no nos da con claridad los puntos de corte entre las dos funciones, por lo que tenemos un problema, por lo que lo vamos a realizar de la siguiente manera, para obtener dichos puntos de corte:

```
>> f2 = @(x)x^2-2^x;
>> fsolve(f2,2)
ans = 2
>> fsolve(f2,4)
ans = 4
>> fsolve(f2,-0.5)
ans = -0.76666
```

Los puntos elegidos para la realización de este comando son los que vemos que se cortan mediante la gráfica realizada con anterioridad.

El área es la suma de los valores absolutos de las áreas de cada uno de los recintos que podemos ver en la gráfica, que es la siguiente:

```
>> abs(int(f-g,(-0.766665),2)) + abs(int(f-g,2,4))
```

warning: passing floating-point values to sym is dangerous, see "help sym"

warning: called from

double_to_sym_heuristic at line 50 column 7

sym at line 379 column 13

int at line 138 column 7

ans = (sym)

$$-\frac{8}{\log(2)} - \frac{2}{2 \cdot \log(2)} + \frac{380223834409013}{23989201619919}$$

4.3. EJERCICIOS.

1) Calcula las siguientes integrales:

- $\int \frac{e^{2x}}{\sqrt{e^x+1}} dx$

```
>> clear
```

```
>> syms x
```

```
>> f = exp(2*x)/sqrt(exp(x)+1);
```

```
>> int(f)
```

ans = (sym)

$$\frac{2 \sqrt[3]{e^x + 1} \cdot e^x}{3} - \frac{4 \sqrt[3]{e^x + 1}}{3}$$

- $\int_0^{\pi/4} \frac{x}{(\cos(x))^2} dx$

```
>> g = x/(cos(x))^2;
```

```
>> int(g,0,pi/4)
```

warning: passing floating-point values to sym is dangerous, see "help sym"

warning: called from

double_to_sym_heuristic at line 50 column 7

sym at line 379 column 13

ans = (sym)

37

$$\frac{\sqrt{1 + 2\sqrt{2}} + 1}{\sqrt{2}}$$

$$\frac{-1 + 2\sqrt{2} + 1}{\sqrt{2}}$$

$$\frac{\log\left(\frac{\sqrt{2} + 1}{\sqrt{2} - 1}\right) + i\pi}{2} + \frac{\log\left(\frac{\sqrt{2} + 1}{\sqrt{2} - 1}\right) + i\pi}{2}$$

$$\frac{-1 + 2\sqrt{2} + 1}{\sqrt{2}}$$

```
>> simplify(int(g,0,pi/4))
```

warning: passing floating-point values to sym is dangerous,
see "help sym"

warning: called from

double_to_sym_heuristic at line 50 column 7

sym at line 379 column 13

int at line 138 column 7

int at line 67 column 5

```
ans = (sym)
```

$$-\frac{\log(2)}{2} + \frac{\pi}{4}$$

2) Calcula el área del recinto limitado por el eje de ordenadas y las curvas de las ecuaciones

$$y = e^{-x}; y = xe^{-x}$$

```
>> syms x
```

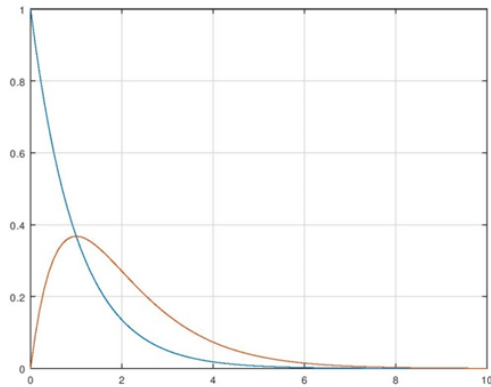
```
>> x = 0:0.1:10;
```

```
>> f1 = exp(-x);
```

```
>> f2 = x.*exp(-x);
```

```
>> plot(x,f1,x,f2)
```

```
>> grid on
```



```
>> abs(int(f1-f2,0,1))
```

```
ans = (sym)
```

```
-1  
e
```

3) Calcula el área del recinto acotado limitado por las curvas $y = x^3 - 2x^2 + x - 1$;

$$y = -x^2 + 3x - 1$$

```
>> pkg load symbolic
```

```
>> syms x
```

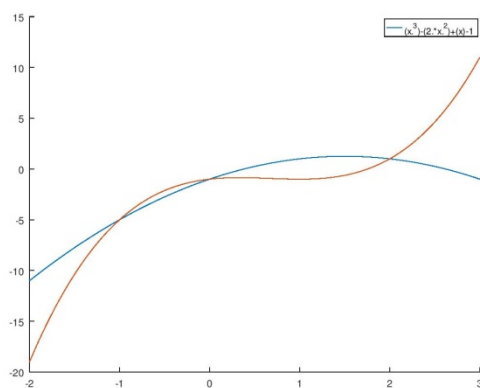
Symbolic pkg v2.7.1: Python communication link active, SymPy v1.3.

```
>> hold on
```

```
>> fplot('(-x.^2)+(3.*x)-1',[-2,3])
```

```
>> fplot('(x.^3)-(2*x.^2)+(x)-1',[-2,3])
```

```
>> hold off
```



Calculamos los puntos de corte entre las dos funciones:

```
>> solve(((x.^2)+(3.*x)-1)==((x.^3)-(2*x.^2)+(x)-1))
```

```
ans = (sym 3x1 matrix)
```

```
[-1]
```

```
[ ]
[0 ]
[ ]
[2 ]
```

Calculamos el área:

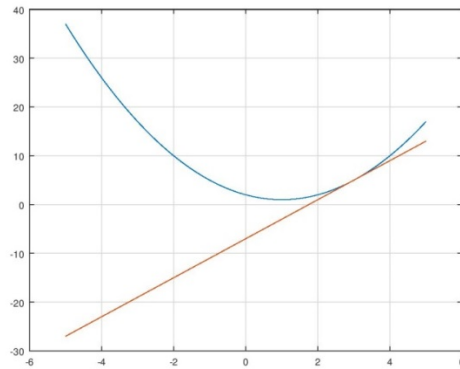
```
abs(int((-x.^2)+(3.^x)-1)-((x.^3)-(2*x.^2)+(x)-1),-1,0))+abs(int((-x.^2)+(3.^x)-1)-
((x.^3)-(2*x.^2)+(x)-1)-((x.^3)-(2*x.^2)+(x)-1),0,2))
ans = (sym)
```

```
11      26
- -- + -----
12      3*log(3)
```

4) La función definida por $f(x) = x^2 - 2x + 2$, con dominio en todo $\mathbb{R}$.

- Halla la ecuación de la recta tangente a la gráfica de f en el punto de abscisa $x=3$.

```
>> syms x h
>> f = @(x)x^2-2*x+2;
>> f(3)
ans = 5
>> limit((f(3+h)-f(3))/h,h,0)
ans = (sym) 4
y - 5 = 4 (x - 3)
y = 4x - 7
>> syms x
>> x = linspace(-5,5,101);
>> f1 = x.^2-2*x+2;
>> f2 = 4*x-7;
>> plot(x,f1,x,f2);
>> grid on
```

- Calcula el área del recinto limitado por la gráfica de f , la recta tangente obtenida y el eje OY.

```
>> syms x
>> f = x^2-2*x+2;
>> g = 4*x-7;
>> int(abs(f-g),0,3)
ans = (sym) 9
```

5. MATRICES.

Para la realización de este capítulo, no es necesario tener que cargar el paquete simbólico en Octave.

5.1. CONSTRUCCIÓN DE MATRICES.

Las matrices es una colección de objetos; que pueden ser números, expresiones u otras listas, que constituyen uno de los elementos más flexibles y poderosos de las Matemáticas.

Para la realización de una matriz o vector en Octave, se realiza de la siguiente forma:

$V = \{1, 2, 3\}$ (Vector con 3 componentes)

```
>> v = [1, 2, 3]
```

v =

```
1 2 3
```

$M = \{\{1, 2, 0\}, \{2, 1, 2\}, \{2, 1, 1\}\}$ (Matriz cuadrada con 3 filas y 3 columnas).

```
>> m = [1, 2, 0; 2, 1, 2; 2, 1, 1]
```

m =

```
1 2 0
```

```
2 1 2
```

```
2 1 1
```

5.2. OPERACIONES CON MATRICES.

Para explicar mejor los siguientes apartados de este punto, vamos a realizarlo con unos ejemplos, que son los siguientes:

$A1 = \{\{1/2, 1/3, 1/4\}, \{1/3, 1/4, 1/5\}, \{1/4, 1/5, 1/6\}\}$

$A2 = \{\{1, 1, 1\}, \{1, 1, 1\}, \{1, 1, 1\}\}$

$A3 = \{\{1, 0\}, \{0, 0\}, \{0, 1\}\}$

$A4 = \{0, 1, 0\}$

```
>> A1 = [1/2,1/3,1/4; 1/3,1/4,1/5; 1/4,1/5,1/6];
```

```
>> A2 = [1,1,1; 1,1,1; 1,1,1];
```

```
>> A3 = [1,0; 0,0; 0,1];
```

```
>> A4 = [0,1,0];
```

Observa que al definir las matrices A1, A2 y A3 hemos puesto “;” al final de la línea, lo que hace que no las escriba en la pantalla al procesar la correspondiente celda. Por otra parte, recordemos que Octave no opera de modo simbólico, a no ser que tenga instalada el paquete con anterioridad y se lo indiquemos para poder usarlo.

5.2.1. SUMA DE MATRICES.

La suma de matrices se expresa de la siguiente manera:

```
>> A1 + A2
```

```
ans =
```

```
1.5000 1.3333 1.2500
```

```
1.3333 1.2500 1.2000
```

```
1.2500 1.2000 1.1667
```

Observa la respuesta del programa cuando tratamos de sumar matrices con dimensiones distintas:

```
>> A2 + A3
```

```
error: operator +: nonconformant arguments (op1 is 3x3, op2 is 3x2)
```

Nos da error, ya que es imposible sumar dos matrices con dimensiones diferentes.

5.2.2. PRODUCTO ESCALAR POR MATRIZ.

La multiplicación de un escalar por un vector o matriz se realiza utilizando el operador “*”. Veamos algunos ejemplos:

```
>> 7 * A3
```

```
ans =
```

```
7 0
```

```
0 0
```

```
0 7
```

En el caso, de no utilizar el operador “*”, nos da el siguiente mensaje:

```
>> 7 A3
```

```
parse error:
```

```
syntax error
```

Ya que Octave no reconoce la operación que le estamos ordenando a realizar, por lo que nos da error.

5.2.3. PRODUCTO DE MATRICES.

Para multiplicar dos matrices utilizando el operador “*” debemos tener cuidado con el número de columnas de la primera matriz que debe coincidir con el número de filas de la segunda matriz, realizamos un ejemplo:

```
>> A4 * A1
```

```
ans =
```

```
0.33333 0.25000 0.20000
```

Podemos utilizar también la multiplicación de elemento a elemento “.*”, que aunque no tiene demasiado sentido como multiplicación de matrices, sí que es muy utilizado en el caso de que la matriz sea un conjunto ordenado de valores:

```
>> A4.*A1
```

```
ans =
```

```
0.00000 0.33333 0.00000
```

```
0.00000 0.25000 0.00000
```

```
0.00000 0.20000 0.00000
```

Realizamos otros ejemplos:

```
>> A1 * A2
```

```
ans =
```

```
1.08333 1.08333 1.08333
```

```
0.78333 0.78333 0.78333
```

```
0.61667 0.61667 0.61667
```

```
>> A1 .*A2
```

```
ans =
    0.50000  0.33333  0.25000
    0.33333  0.25000  0.20000
    0.25000  0.20000  0.16667
```

En el caso del siguiente producto, no es posible multiplicar ya que la matriz A3 y A2 , no tienen el mismo número de columnas de A3 y de filas de A2:

```
>> A3*A2
error: operator *: nonconformant arguments (op1 is 3x2, op2 is 3x3)
```

Por lo que Octave, nos da un mensaje de error en estas situaciones.

El producto de una matriz cuadrada por si misma n veces puede hacerse de la siguiente forma:

```
>> A2^5
ans =
    81    81    81
    81    81    81
    81    81    81
```

5.2.4. INVERSA, TRANSPUESTA Y DETERMINANTE DE UNA MATRIZ.

Veamos con algunos ejemplos, para cada tipo de operación de las matrices:

B1 = {{1,0,0}, {2,2,0}, {1,1,0}}

```
>> B1 = [1,0,0; 2,2,0; 1,0,5];
```

B2 = {{1,0,0}, {2,1,1}, {1,1,0}}

```
>> B2 = [1,0,0; 2,1,1; 1,1,0];
```

Inversa:

Para el cálculo de la inversa usamos el comando “inv”, como vemos a continuación:

```
>> C = inv(B1)
C =
    1.00000  0.00000  0.00000
   -1.00000  0.50000  0.00000
   -0.20000 -0.00000  0.20000
```

Transpuesta:

Para la transpuesta, solo tenemos que indicar a la matriz que queremos transponer añadiéndole una coma, veamos el ejemplo:

```
>> D = B1'
```

```
D =
```

```
1 2 1
0 2 0
0 0 5
```

Determinante:

Para el cálculo de los determinantes, usamos el comando “det”, veamos el ejemplo:

```
>> det(B1)
```

```
ans = 10
```

5.2.5. RANGO DE UNA MATRIZ.

Dada la matriz M, vamos a calcular el rango en Octave con el comando “rank”, al ser la matriz con 4 filas y 3 columnas el rango puede ser como máximo 3:

```
>> rank (m)
```

```
ans = 2
```

Como podemos observar, como el rango de la matriz es menor a orden 3, por lo que nos indica que la matriz es de orden dos, ya que es el valor que obtenemos en este caso.

5.3. EJERCICIOS.

1) Dadas las siguientes matrices, realiza las siguientes operaciones:

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \text{ y } B = \begin{pmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

```
>> A = [1,1,1; 1,0,1; 1,1,1];
```

```
>> B = [0,0,0; 2,0,0; 1,2,1];
```

- 3 A + 2 B:

```
>> 3*A+2*B
```

```
ans =
```

```
3 3 3
7 0 3
5 7 5
```

- $A*B'$:

```
>> A*B'
```

```
ans =
```

```
0    2    4
0    2    2
0    2    4
```

- $\text{Det}(A*B')$:

```
>> det(A*B')
```

```
ans = 0
```

- A^{127} :

```
>> A^127
```

```
ans =
```

```
1.0714e+55    7.8430e+54    1.0714e+55
7.8430e+54    5.7415e+54    7.8430e+54
1.0714e+55    7.8430e+54    1.0714e+55
```

2) Calcula el rango de la siguiente matriz:

$$A = \begin{pmatrix} 1 & -1 & -2 \\ 0 & 1 & 1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \\ 1 & 2 & 1 \end{pmatrix}.$$

```
>> a = [1,-1,-2; 0,1,1; 2,0,-2; 1,0,-1; 1,2,1];
```

```
>> rank(a)
```

```
ans = 2
```

3) Calcula la siguiente ecuación, usando las matrices del ejercicio 1:

$$A * X + B^t * X = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 2 \end{pmatrix}$$

```
>> A = [1,1,1; 1,0,1; 1,1,1];
```

```
>> B = [0,0,0; 2,0,0; 1,2,1];
```

```
>> X= inv(A+B')*[1,1,1; 0,0,1; 1,1,2]
```

```
X =
```

```
3.00000    3.00000    5.50000
0.00000    0.00000   -0.50000
-1.00000   -1.00000   -1.50000
```

- 4) Demuestra, usando las matrices del ejercicio 1, que no es cierta la propiedad conmutativa para el producto de matrices cuadradas:

Propiedad conmutativa: $A*B = B*A$

$A*B$:

```
>> A*B
```

ans =

```
3 2 1
1 2 1
3 2 1
```

$B*A$:

```
>> B*A
```

ans =

```
0 0 0
2 2 2
4 2 4
```

Por lo que no al ser el mismo resultado, podemos decir que no se cumple la propiedad conmutativa.

- 5) Resuelve el sistema $A * X = B$:

$$A = \begin{pmatrix} 1 & 0 & -1 \\ 0 & -1 & 3 \\ 4 & 1 & 1 \end{pmatrix}; B = \begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix} \text{ y } X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

```
>> A = [1,0,-1; 0,-1,3; 4,1,1];
```

```
>> B = [1; -1; 3];
```

```
>> X = inv(A)*B
```

X =

0.75000

0.25000

-0.25000

6. SISTEMAS LINEALES. VECTORES.

6.1.SISTEMAS DE ECUACIONES LINEALES.

Trataremos de encontrar las soluciones de un sistema de ecuaciones lineales, veamos un ejemplo:

$$x + y + z = 0$$

$$x - y + z = 2$$

$$x + 2z = 4$$

Vamos a ver si el comando “solve” nos da la solución del sistema de ecuaciones planteado:

```
>> pkg load symbolic
```

```
>> syms x y z
```

```
>> [X,Y,Z]=solve(x+y+z==0, x-y+z==2, x+0*y+2*z==4)
```

```
X = (sym) -2
```

```
Y = (sym) -1
```

```
Z = (sym) 3
```

También podemos calcular un sistema de ecuaciones con el siguiente procedimiento mediante los rangos de la matriz de coeficientes y la ampliada.

Matriz de coeficientes:

```
>> a = [1,1,1; 1,-1,1; 1,0,2];
```

Matriz de términos independientes:

```
>> b = [0; 2; 4];
```

Matriz ampliada:

```
>> c = [1,1,1,0; 1,-1,1,2; 1,0,2,4];
```

Calculamos los rangos de la matriz de coeficientes y de la matriz ampliada:

```
>> rank(a)
```

```
ans = 3
```

```
>> rank(c)
```

```
ans = 3
```

Como podemos ver los rangos de ambas matrices son 3, que es igual al número de incógnitas, por lo que nos quiere decir que el sistema compatible determinado (tiene una única solución).

Por lo que empleamos el comando “linsolve”, para obtener la única solución que el sistema de ecuaciones tiene:

```
>> linsolve (a,b)
```

```
ans =
```

```
-2
```

```
-1
```

```
3
```

Obtenemos los resultados de la ecuación lineal; podemos decir que en todo este proceso para el cálculo de la solución de un sistema de ecuaciones no hemos tenido que utilizar el paquete simbólico de Octave.

Vamos a realizar otro ejemplo:

$$\begin{aligned}2x + y &= 3 \\ y + 2z &= 1 \\ x + y + z &= 0\end{aligned}$$

Realizamos las matrices de este sistema lineal para poder obtener el rango de dichas matrices, que serían las siguientes:

```
>> clear
>> a = [2,1,0; 0,1,2; 1,1,1];
>> b = [3; 1; 0];
>> c = [2,1,0,3; 0,1,2,1; 1,1,1,0];
>> rank(a)
ans = 2
>> rank(c)
ans = 3
```

Como podemos ver este sistema incompatible, porque no tiene solución, ya que los rangos obtenidos son diferentes entre sí.

Vamos a ver otro tipo de sistema lineal:

$$\begin{aligned}2x + y &= 3 \\ y + 2z &= 1 \\ x + y + z &= 2\end{aligned}$$

```
>> a = [2,1,0; 0,1,2; 1,1,1];
>> b = [3; 1; 2];
>> c = [2,1,0,3; 0,1,2,1; 1,1,1,2];
>> rank(a)
ans = 2
>> rank(c)
ans = 2
```

Como podemos ver este sistema compatible indeterminado, porque tiene infinitas soluciones, ya que los rangos obtenidos son iguales entre sí, y no son iguales al número de incógnitas del sistema lineal.

Para obtener las infinitas soluciones, haríamos lo siguiente:

```
>> [X,Y,Z]=solve(2*x+y+0*z==3, 0*x+y+2*z==1, x+y+z==2, x,y,z)
```

warning: solve: number of outputs did not match solution vars

warning: called from

solve at line 194 column 7

```
X = (sym) z + 1
```

```
Y = (sym) -2*z + 1
```

error: element number 3 undefined in return list
error: element number 3 undefined in return list

El programa Octave, nos deja la opción de elegir que dos incógnitas queremos dejar como principales:

```
>> [Y,Z]=solve(2*x+y+0*z==3, 0*x+y+2*z==1, x+y+z==2,y,z)
```

```
Y = (sym) -2*x + 3
```

```
Z = (sym) x - 1
```

Para resolver sistemas de ecuaciones que dependan de parámetros. Vamos a ver como se resolvería mediante un ejemplo:

$$ax + y + z = a$$

$$x + ay + z = a^2$$

$$x + y + az = a^3$$

Para el cálculo de este sistema de ecuaciones tenemos que utilizar el paquete simbólico.

```
>> clear
```

```
>> syms a
```

```
>> A = [a,1,1; 1,a,1; 1,1,a];
```

```
>> B = [a; a.^2; a.^3];
```

```
>> C = [a,1,1,a; 1,a,1,a.^2; 1,1,a,a.^3];
```

```
>> det(A)
```

```
ans = (sym)
```

$$a^3 - 3*a + 2$$

Para saber que valores toma el parámetro “a”, debemos igualar el resultado del determinante a cero; por lo que realizamos lo siguiente para ver en que valores la ecuación es igual a cero:

```
>> f = a^3-3*a+2;
```

```
>> solve(f)
```

```
ans = (sym 2x1 matrix)
```

```
[-2]
```

```
[ ]
```

```
[1 ]
```

Por lo que vemos que los valores donde el parámetro hace que no tenga una única solución es en -2 y 1. Por lo que vamos a ver qué pasa cuando sustituimos el parámetro por esos valores.

Cuando $a = -2$:

```
>> A1 = [-2,1,1; 1,-2,1; 1,1,-2];
>> B1 = [-2; (-2)^2; (-2)^3];
>> C1 = [-2,1,1,-2; 1,-2,1,(-2)^2; 1,1,-2,(-2)^3];
>> rank(A1)
ans = 2
>> rank(C1)
ans = 3
```

Por lo que podemos decir que el sistema es incompatible, ya que los rangos no son iguales al número de incógnitas.

Cuando $a = 1$:

```
>> A2 = [1,1,1; 1,1,1; 1,1,1];
>> B2 = [1; (1)^2; (1)^3];
>> C2 = [1,1,1,1; 1,1,1,(1)^2; 1,1,1,(1)^3];
>> rank (A2)
ans = 1
>> rank (C2)
ans = 1
```

Por lo que podemos decir que el sistema es compatible indeterminado, ya que los rangos son iguales, pero no son iguales al número de incógnitas. Obtenemos la siguiente solución:

```
>> [x,y,z]=solve(A2*X==B2,x,y,z)
warning: solve: number of outputs did not match solution vars
warning: called from
    solve at line 194 column 7
x = (sym) -y - z + 1
error: element number 2 undefined in return list
```

Cuando $a \neq -2$ y $a \neq 1$:

```
>> syms a
Symbolic pkg v2.7.1: Python communication link active, SymPy v1.3.
>> A = [a,1,1; 1,a,1; 1,1,a];
```

```
>> B = [a; a.^2; a.^3];
>> C = [a,1,1,a; 1,a,1,a.^2; 1,1,a,a.^3];
>> syms x y z a
>> X = [x; y; z];
>> [x,y,z]=solve(A*X==B,x,y,z)
x = (sym)
```

$$\frac{-a*(a + 1)}{a + 2}$$

```
y = (sym)
```

$$\frac{a}{a + 2}$$

```
z = (sym)
```

$$\frac{a*\sqrt{a^2 + 2*a + 1}}{a + 2}$$

6.2. EJERCICIOS.

1) Resuelve la siguiente ecuación lineal, con las incógnitas x, y, z:

$$\begin{cases} x + y + z = 2 \\ 2x - y + 3z = 1 \\ x + z = 1 \end{cases}$$

```
>> syms x y z
>> a=[1,1,1; 2,-1,3; 1,0,1];
>> b=[2;1;1];
>> c=[1,1,1,2;2,-1,3,1;1,0,1,1];
>> rank(a)
ans = 3
>> rank(c)
ans = 3
>> linsolve(a,b)
ans =
1
1
-0
```

2) Resuelve los siguientes sistemas de ecuaciones lineales con incógnitas x, y, z, t:

$$\begin{cases} 3x + 2y - z = 5 \\ x - y + 2z = 1 \\ -x + z - t = 0 \end{cases}$$

$$5x + y - 2z + 3t = 6$$

```
>> pkg load symbolic
```

```
>> syms x y z
```

Symbolic pkg v2.7.1: Python communication link active, SymPy v1.3.

```
>> A = [3,2,-1,0; 1,-1,2,0; -1,1,0,-1; 5,1,-2,3];
```

```
>> B = [5; 1; 0; 6];
```

```
>> C = [3,2,-1,0,5; 1,-1,2,0,1; -1,1,0,-1,0; 5,1,-2,3,6];
```

```
>> rank(A)
```

```
ans = 4
```

```
>> rank(C)
```

```
ans = 4
```

```
>> linsolve(A,B)
```

```
ans =
```

```
1.00000
```

```
1.33333
```

```
0.66667
```

```
0.33333
```

$$\begin{cases} 3x + 6y - 5z + 4w = -1 \\ x + 3z - 2w = 1 \\ 2x + 3y - z + w = 0 \\ 3y - 7z + 5w = -2 \end{cases}$$

```
>> pkg load symbolic
```

```
>> syms x y z w
```

```
>> A = [3,6,-5,4; 1,0,3,-2; 2,3,-1,1; 0,3,-7,5];
```

```
>> B = [-1; 1; 0; -2];
```

```
>> C = [3,6,-5,4,-1; 1,0,3,-2,1; 2,3,-1,1,0; 0,3,-7,5,-2];
```

```
>> rank(A)
```

```
ans = 2
```

```
>> rank(C)
```

ans = 2

Este sistema compatible indeterminado. Calculamos sus infinitas soluciones:

```
>> X = [x; y; z; w];
```

```
>> [x,y,z,w]=solve(A*X==B,x,y,z,w)
```

warning: solve: number of outputs did not match solution vars

warning: called from

solve at line 194 column 7

x = (sym) 2*w - 3*z + 1

y = (sym)

$$-\frac{5w}{3} + \frac{7z}{3} - \frac{2}{3}$$

error: element number 3 undefined in return list

7. DIAGONALIZACIÓN DE MATRICES.

7.1. DIAGONALIZACIÓN DE UNA MATRIZ.

Dada una matriz A cuadrada de orden n, diremos que es una matriz diagonalizable si existe una matriz P (matriz de paso), que cumpla la siguiente ecuación:

$$D = P^{-1} * A * P$$

D es una matriz diagonal.

Tenemos el siguiente comando para obtener el cálculo de autovalores y autovectores de matrices cuadradas, veamos el siguiente ejemplo:

```
>> a = [2,0,-2; -3,-1,2; 2,0,-2]
```

a =

2 0 -2

-3 -1 2

2 0 -2

```
>> [P,D]= eig(a)
```

P =

0.00000 + 0.00000i 0.57735 + 0.00000i 0.57735 - 0.00000i

1.00000 + 0.00000i -0.57735 - 0.00000i -0.57735 + 0.00000i

0.00000 + 0.00000i 0.57735 + 0.00000i 0.57735 - 0.00000i

D =

Diagonal Matrix

-1.00000 + 0.00000i

0

0

```

0 0.00000 + 0.00000i 0
0 0 0.00000 - 0.00000i

```

Realizamos un segundo ejemplo, con otra matriz, usamos el comando “clear”, para borrar los comandos anteriores para que no nos afecte en este ejemplo:

```

>> clear
>> b = [0,0,0; 0,1,-2; 0,-2,1];
>> [P,D]= eig(b)
P =
-0.00000  1.00000  0.00000
-0.70711  0.00000 -0.70711
-0.70711  0.00000  0.70711

```

D =

Diagonal Matrix

```

-1  0  0
0  0  0
0  0  3

```

Otra forma de obtener la matriz D es mediante el producto de la $P^{-1} \cdot B \cdot P$, donde P es la matriz de paso que tiene por columnas a los vectores propios de B.

```

>> x = inv(P)*b*P
x =
-1  0  0
0  0  0
0  0  3

```

7.2. EJERCICIOS.

- 1) Calcular los autovalores y vectores propios de la matriz $\begin{pmatrix} 8 & 1 & 1 \\ 1 & 6 & 2 \\ 1 & 3 & 7 \end{pmatrix}$.

```

>> A = [8,1,1; 1,6,2; 1,3,7];
>> [P,D]= eig(A)
P =
-5.7208e-01 -8.0178e-01  1.3508e-16
-4.7673e-01  2.6726e-01 -7.0711e-01
-6.6742e-01  5.3452e-01  7.0711e-01

```

D =

Diagonal Matrix

```
10.0000    0    0
0    7.0000    0
0    0    4.0000
```

- 2) Dada la matriz $C = \begin{pmatrix} 5 & 0 & 1 \\ 0 & 2 & 0 \\ 5 & -1 & 1 \end{pmatrix}$, calcular C^{20} a partir de la diagonalización de la matriz A.

```
>> C = [5,0,1; 0,2,0; 5,-1,1];
```

```
>> [P,D]=eig(C)
```

P =

```
0.70711 -0.19612 0.11625
0.00000 0.00000 0.92998
0.70711 0.98058 -0.34874
```

D =

Diagonal Matrix

```
6 0 0
0 0 0
0 0 2
```

El cálculo lo podemos realizar directamente en Octave de la siguiente forma;

$F = C^{20}$

```
>> F = P*D.^20*inv(P)
```

F =

```
3.0468e+15 -1.5234e+14 6.0936e+14
0.0000e+00 1.0486e+06 0.0000e+00
3.0468e+15 -1.5234e+14 6.0936e+14
```

```
>> C^20
```

ans =

```
3.0468e+15 -1.5234e+14 6.0936e+14
0.0000e+00 1.0486e+06 0.0000e+00
3.0468e+15 -1.5234e+14 6.0936e+14
```


8. CONCLUSIONES.

Las conclusiones llegadas tras la realización de este trabajo de fin de grado, es que este programa tiene algunos problemas a la hora de darnos algunas soluciones en concreto en los primeros temas (funciones, derivación de funciones e integración de funciones), ya que no nos proporciona en ciertos casos unas soluciones muy claras. En cambio, este programa nos da más facilidades en los temas de matrices que el programa Wolfram Mathematica.

Como indicamos al principio este programa es gratuito y por lo que es más fácil de obtener que el otro programa que se utiliza en la Universidad de Jaén, ya que el programa que se usa no es gratuito y se necesita tener licencia para poder usarlo.

En mi opinión, Octave merece la pena usarlo para la utilización de los últimos temas que se estudian en la asignatura de Matemáticas I en el Grado de Administración y Dirección de empresas; ya que tiene unos comandos más simples en su utilización y que el programa no da tantos problemas en la hora de que nos dé un resultado exacto.

El programa Octave es similar al programa “Matlab”, pero en cambio a otros programas, este programa no hay mucha información de cómo se utiliza, para la hora de realizar dichas cuestiones que se nos pueden plantear.

Para las prácticas que se abordan en esta asignatura, el mejor programa es Wolfram Mathematica, ya que está mucho más preparado para los tipos de ejercicios que se realizan en las prácticas en Matemáticas I que el programa Octave, ya que este es mucho más engorroso a la hora de la realización este tipo de temas. Wolfram Mathematica es mucho más fácil de utilizar sobre todo en los ejercicios de los primeros temas (funciones, derivación de funciones e integración de funciones).

9. BIBLIOGRAFIA.

- Guía para el uso de Octave en Matemáticas, disponible online: http://softlibre.unizar.es/manuales/aplicaciones/octave/manual_octave.pdf
- Paquete simbólico del programa Octave, disponible online: <https://octave.sourceforge.io/packages.php>
- Pérez, M. (2012) “Matlab, algebra en la ingeniería y las ciencias experimentales”, Lexington.
- Prácticas de Matemáticas I del Grado de Administración y Dirección de Empresas de la Universidad de Jaén por el departamento de Matemática aplicada.
- Programa Octave versión 4.4.1, disponible online: <https://www.gnu.org/software/octave/#install>
- Quarteroni, A. y Saleri, F. (2006) “Cálculo Científico con Matlab y Octave”, Milano. Disponible online: <https://0-link.springer.com.avalos.ujaen.es/book/10.1007%2F978-88-470-0504-4> (Biblioteca de la Universidad de Jaén).