

LA CONSTANTE PI SE ESCRIBE COMO : Pi (la p en mayúsculas y la i en minúsculas)

TANGENTE : $\tan(x)$

SENO: $\sin(x)$

Log Nep: $\ln(x)$

Log base10: $\log(x)$

e elevado a x: $\exp(x)$

PRACTICA 1 LIMITES Y CONTINUIDAD DE FUNCIONES

1.- $f:=(x) \rightarrow \text{expr_en_x}$: Define la función escalar de una variable $f(x) = \text{expr_en_x}$; La expresión analítica de la función se obtiene como $f(x)$; y no simplemente como f, y su valor en el punto $x = a$; se obtiene como $f(a)$; De forma análoga, el comando $f:=(x,y) \rightarrow \text{expr_en_x_en_y}$ define la función escalar de dos variables $f(x,y) = \text{expr_en_x_en_y}$;

2.- $\text{evalf}(\text{expr_sim},n)$: Obtiene una aproximación decimal con n dígitos de la expresión simbólica expr_sim . El parámetro n es opcional, y su valor por defecto es 10.

3.- $\text{plot}(\text{ecuacion_en_x}, x=a..b, \text{opciones})$: Representa gráficamente la curva de ecuación $y=\text{ecuacion_en_x}$ para x en el intervalo $[a,b]$. Las opciones son todas ellas de la forma $\text{opcion}=\text{valor}$, y pueden estar o no presentes. También se puede utilizar bajo la forma $\text{plot}(\{\text{ec1_en_x}, \text{ec2_en_x}\}, x=a..b, \text{opciones})$, para representar simultáneamente las dos curvas de ecuaciones $y=\text{ec1_en_x}$ e $y=\text{ec2_en_x}$, cada una en color distinto. Análogamente se pueden representar simultáneamente más de dos curvas.

3.- $\text{plot3d}(\text{ecuacion_en_x_en_y}, x=a..b, y=c..d, \text{opciones})$: Representa gráficamente la superficie de ecuación $z = \text{ecuacion_en_x_en_y}$; para x en el intervalo $[a,b]$ y para y; en el intervalo $[c, d]$; Las opciones son todas ellas de la forma $\text{opcion}=\text{valor}$, y pueden estar o no presentes. También se puede utilizar bajo la forma $\text{plot3d}(\{\text{ec1_en_x_en_y}, \text{ec2_en_x_en_y}\}, x=a..b, y=c..d, \text{opciones})$, para representar simultáneamente las dos superficies de ecuaciones $z = \text{ec1_en_x_en_y}$; e $z = \text{ec2_en_x_en_y}$; cada una en color distinto. Análogamente se pueden representar simultáneamente más de dos superficies.

4.- $\text{limit}(\text{función_en_x}, x=a, \text{dirección})$: Calcula el límite de la función $f(x)$; cuando x; tiende al punto a;. La dirección es opcional, y puede ser left ó right, según se quiera calcular el límite por la izquierda o por la derecha. El comando también funciona para límites múltiples, en la forma $\text{limit}(\text{función_en_x_en_y}, \{x=a, y=b\}, \text{opción})$. El programa admite trabajar con el valor infinito ($\text{infinity} = \text{infinity}$;) como punto de acumulación.

-EXISTEN DOS LIMITES REITERADOS!!!!!!

$\text{LIMIT}(\text{LIMIT}(F, Y=a), X=b)$

Y

$\text{LIMIT}(\text{LIMIT}(F, X=b), Y=a)$

-LOS LIMITES SEGÚN UNA TRAYECTORIA SE HACEN:

LIMIT (SUBS(Y=trayect,F(x,y)),X=a)

–LIM DOBLE:

LIMIT(F,{X=a,Y=b})

–LIMITE EN UNA RECTA (EJEMPLO $X*Y=0$):

CUANDO LOS VALORES DE X E Y ACE Q SE CUMPLA LA CONDICION DE LA RECTA, EN ESTE CASO Q X SEA 0 O Y SEA 0

5.– `expand(expr_en_x)`: Expande la expresión `expr_en_x`; ya sea aplicando la propiedad distributiva, descomponiendo funciones racionales en fracciones simples, etc.

6.– `with(paquete)`: Carga los comandos propios de la librería de nombre paquete.

*COMANDOS DEL PAKETE PLOT:

–Dibujar la gráfica de la función :

`plot3d(k(x,y),x=...,y=..);`

–Dibujar la gráfica de la función con densidad :

`plot3d(k(x,y),x=...,y=..,filled=true);`

–mapa de contorno:

`contourplot(k(x,y),x=...,y=..);`

–mapa de contorno en 3d:

`contourplot3d(k(x,y),x=...,y=..);`

–mapa de contorno en 3d con densidad:

`contourplot3d(k(x,y),x=...,y=..,filled=true);`

–mapa de contorno con densidad:

`contourplot(k(x,y),x=...,y=..,filled=true);`

7.– **`contourplot`**(`expr_en_x_en_y`,`x=a..b`,`y=c..d`,`opciones`): Dibuja el mapa de contorno de la función $f(x) = \text{expr_en_x_en_y}$; cuando la x ; varía en el intervalo $[a, b]$; y la y ; varía en el intervalo $[c, d]$. Una de las opciones disponibles es `filled=true`, que devuelve un mapa de densidad. El comando `contourplot3d(expr_en_x_en_y,x=a..b,y=c..d,opciones)` lleva el mapa a la superficie. En ambos casos, se puede superponer varios mapas de contorno o densidad con tan sólo sustituir la función de entrada por una lista de varias funciones.

8.– `subs(x=a,expr)`: Sustituye la variable x ; por a ; en `expr`;

9.– `piecewise(cond_1,expr_1,...,cond_n,expr_n,expr_por_defecto)`: Define una expresión por partes, que toma

el valor expr_i ; si la condición cond_i es verdadera, y el valor $\text{expr}_{\text{por_defecto}}$; en caso de que ninguna de las condiciones se verifique.

PRACTICA 2 : DIFERENCIACIÓN (ESTUDIO) DE FUNCIONES

LAS COSAS A ESTUDIAR AL HACERLE EL ESTUDIO DE UNA FUNCION MEDIANTE EL MAPLE SON: DOMINIO, EXTREMOS RELATIVOS RECUERDA Q PARA SABER D Q TIPO ERA PUNTO CRITICO EN FUNCIONES DE UNA VARIABLE SE SUSTITUIA EL PUNTO CRITICO EN LA SEGUNDA DERIVADA Y SI DABA <0 ERA UN MAX Y SI DABA >0 ERA UN MINIMO!!!! Y PUNTOS DE INFLEXION, ASINTOTAS (VERTICALES, HORIZONTALES Y OBLICUAS) Y RAMAS PARABOLICAS REPASALO CON TUS APUNTES!!!!

RECORDAMOS Q LAS ASINTOTAS OBLICUAS ERAN:

PRIMERO Hemos de estudiar el límite $\text{Limit}(f(x), x = \text{infinity})$:

SI NOS DA INFINITO SIGNIFICA Q AY UNA ASINTOTA OBLICUA CUANDO X TIENDE A INFINITO. ESTUDIAMOS LA RECTA Q LA DEFINE:

Hemos de obtener el límite $m = \text{Limit}(f(x)/x, x = \text{infinity})$:

Y por último el límite $n = \text{Limit}(f(x) - m \cdot x, x = \text{infinity})$; ESTO ES PORQUE $F(X) = M \cdot X + N$ Y POR TANTO $N = F(X) - M \cdot X$ REPASALO!!!!!!

1.- $\text{unapply}(\text{expr_sim}, \text{variabl})$: Convierte la expresión expr_sim en una función de las variables variabl . En algunos casos equivale a la definición de una función con $(x, y) \rightarrow \text{expr_sim}$, pero con la opción unapply se puede incorporar a la expresión expr_sim no sólo expresiones simbólicas con las variables sino operaciones con otras expresiones. Por ejemplo, el comando $f := (x, y) \rightarrow \sqrt{x^2 + y^2}$ define una función $f(x, y)$ y el comando $g := \text{unapply}(f(x, y) + 2, x, y)$ define la función $g(x, y) = f(x, y) + 2$.

EL UNAPPLY LO USAMOS ASI: $f := \text{unapply}(\text{expresión}, \text{variable})$, es decir, por ejemplo:
 $f := \text{unapply}(\text{diff}(h(x), x), x)$ ó $f := \text{unapply}(h(x) + g(x), x)$ etc... en definitiva tienes q poner $f := \text{unapply}...$ sin poner ni la flecha ni la variable de la función q defines ya que esta la pones al final del unapply , si son varias variables pos las pones al final separas por comas.

EL UNAPPLY SIRVE PA DEFINIR FUNCIONES COMPUESTAS POR OTRAS FUNCIONES (O POR DERIVADAS DE OTRAS FUNCIONES) ACIENDOLO COMO $F := (x) \rightarrow G(X) + H(x)$ NO SE PUEDE HACER TIENES Q PONER $f := \text{unapply}(g(x) + h(x), x)$ ACUERDATE DE Q AL FINAL DEL UNAPPLY TIENES Q PONER UNA COMA Y LA VARIABLE CON LA Q TRABAJA LA FUNCION Q DEFINE, SI SON VARIAS POS PONES OTRA COMA.

CUANDO KIERAS DARLE NOMBRE A LA DAERIVADA PRIMERA Y SEGUNDA Y ESO, UTILIZA EL UNAPPLY, NO PUEDES DEFINIR UNA FUNCION COMO DERIVADA DE OTRA SIN UTILIZAR UNAPPLY!!!

2.- opciones de plot: Utilizaremos algunas opciones del comando plot: Por ejemplo la opción $\text{discont} = \text{true}$ para que una función discontinua no la intente dibujar como si fuera continua, o la opción $\text{color} =$ para dibujar en un determinado color.

3.- $\text{solve}(\text{ecuacion}, \text{variabl})$: Resuelve la ecuación ecuacion en las variables variabl . Si en ecuacion únicamente figura una variable no es necesario indicar el nombre de la variable y si la ecuación es $f(x) = 0$ no es necesario indicar la igualdad, es decir $\text{solve}(f(x))$ resuelve la ecuación $f(x) = 0$ en la variable x .

EL COMANDO SOLVE LO PODEMOS UTILIZAR TO FLAMA PARA ESTUDIARLE EL DOMINIO A LAS FUNCIONES POR EJEMPLO EN UN DENOMINADOR PODEMOS ESTUDIAR SOLVE(DENOM(X)=0) Y NOS DARA LOS VALORES PARA EL CUAL ESTE SE ANULA

TB NOS VALE PARA SISTEMAS DE ECUACIONES PONIENDOLO DE LA FORMA :

`solve({ecuaciones},{variables})`

4.- `diff(expr_sim,x1,x2,...)`: Obtiene la derivada de la expresión simbólica `expr_sim` respecto de las variables `x1` en primer lugar, `x2` en segundo lugar, etc.. Por ejemplo `diff(f(x,y,z),x,z)` calcula la derivada segunda f_{xz} ;

5.- `simplify(expr_sim)`: Simplifica la expresión `expr_sim`

6.- Vectores: Cuando por ejemplo una ecuación produce más de un resultado, estos se almacenan en un vector. Por ejemplo `sols:=solve(x^2-1,x)` almacena en un vector, de nombre `sols`, las dos soluciones de la ecuación $x^2-1=0$. Los elementos de este vector son `sols[1]` y `sols[2]`.

7.- `allvalues(expr)`: Este comando devuelve un vector con todos los valores de `expr`. Este comando se utiliza cuando Maple nos ofrece como resultado de resolver la ecuación $f(x)=0$, con el comando `expr:=solve(f(x))`, la solución empírica `RootOf(f(x))`. Esto ocurre principalmente cuando la ecuación $f(x)=0$ tiene tanto soluciones reales como complejas.

SI MAPLE NOS DEVIUELVE COMO RESULTADE DE UNA ECUACION ROOTOF(UNA JARTA DE PARANOYAS) ES Q NOS A DADO SOLUCIONES REALES Y COMPLEJAS , PARA OBTENER LAS SOLUCIONES NORMALES(AUNQ TB SE INCLUYEN LAS COMPLEJAS) DEBEMOS METER ALLVALUES(ECUACION Q QREMOS RESLVER) Y NOS DEVOLVERA LAS SOLUCIONES NORMALES Y LAS COMPLEJAS RESPECTO DE I ; DE LAS COMPLEJAS NOS OLVIDAMOSSSSSSSSSSS!!!!!!!!!!!!!!! , SI KEREMOS ALMACENAR ESTOS VALORES EN UN VECTOR NO TENEMOS MAS Q PONER NOMBREVECTOR:=ALLVALUES(ECUACION Q QREMOS RESOLVER) . PA DESPUES IR SACANDO LOS VALORES DEL VECTOR NA MAS Q TENEMOS Q PONER NOMBREVECTOR[1] O NOMBREVECTOR[2] ETC..

8.- `display([L],options)`: Dibuja en un mismo gráfico los objetos gráficos contenidos en la lista `[L]`, con las opciones indicadas en `options`. Por ejemplo, las ordenes `g1:=plot(f(x),...)` ; `g2:=plot(g(x),...)` crean dos objetos gráficos `g1` y `g2` con las gráficas de $f(x)$ y $g(x)$, respectivamente, y el comando `display(g1,g2)` los muestra en un sólo gráfico. SIRVE MAS Q NA PA PONE LAS ASINTOTAS (OBLICUAS SOBRE TÓ)

9.- `implicitplot3d(expr_sim,x=x1..x2,y=y1..y2,z=z1..z2,options)`: Dibuja la superficie definida en forma implícita por la ecuación `expr_sim=0` indicándose los intervalos de valores para las tres variables.

10.- `vector(nombre,lista)`: Convierte la lista de números contenida en `lista` en un vector y lo almacena en la variable `nombre`. Este comando está incluido en el paquete `linalg`, por lo tanto para poder utilizarlo hemos de cargar dicho paquete, con `with(linalg)`.

11.- `dotprod(v1,v2)`: Efectúa el producto escalar de los vectores `v1` y `v2`. Este comando está incluido en el paquete `linalg`, por lo tanto para poder utilizarlo hemos de cargar dicho paquete, con `with(linalg)`.

PRACTICA 3: TAYLOR Y ESTUDIO DE PUNTOS CRITICOS (ESTACIONARIOS)

REPASATE LO DEL SIGNO DEL HESSIANO Y DEL F_{xx} PA SABER Q TIPO DE PUNTO ES!!!!

1.- `taylor(f(x),x=a,n)`: Obtiene la fórmula de Taylor para la función $f(x)$ en $x=a$, indicando el polinomio de

Taylor de grado $n-1$ en $x=a$ y expresando el resto en forma de infinitésimo. **TE DA EL POLINOMIO DE GRADO $N-1$!!!!!** Y ADEMÁS COMO TE LO DA EN FORMA DE INFINITESIMO , PARA HACER CUALQUIER COSA CON EL LO TIENES Q PASAR A POLINOMIO, CON EL COMANDO CONVERT

2.- `convert(expresion,forma_o_tipo)`: Permite convertir expresiones de un tipo a otro. La usaremos para convertir el resultado del comando `taylor` en polinomio. Por ejemplo, la orden siguiente devuelve, como polinomio, el polinomio de segundo grado de Taylor para la función $f(x)$ en el punto $x=1$: `convert(taylor(f(x),x=1,3) , polynom )`.

3.- `readlib(comando)`: Carga en memoria un comando adicional, de nombre comando, que no está incluido en el núcleo del programa. Su funcionamiento es parecido al comando `with`, con la diferencia de que con éste último se cargan todos los comandos incluidos en un determinado paquete. Por ejemplo, `readlib(mtaylor)` carga en memoria el comando `mtaylor` que permite obtener el polinomio de Taylor para funciones de varias variables.

4.- `mtaylor(f(x,y,...),[x=a,y=b,...],n)`: Proporciona el polinomio de Taylor de grado $n-1$ para la función $f(x,y,...)$ en el punto $(a,b,...)$.

5.- `hessian(f(x,y,...),[x,y,...])`: Proporciona la matriz hessiana (de las derivadas parciales segundas) de la función $f(x,y,...)$.

6.- `det(matriz)`: Proporciona el determinante de la matriz cuadrada matriz.

Extremos condicionados

Tratamos de encontrar los valores que optimizan (maximizan o minimizan) la función $f(x,y,z)$ estando las variables sometidas a la condición (ligadura) $g(x,y,z)=0$. Tales extremos se llaman extremos condicionados.

El método de los multiplicadores de Lagrange nos permite obtener los extremos condicionados de la función $f(x,y,z)$ mediante el cálculo de los extremos libres de la función $F(x,y,z,t) = f(x,y,z) - t \cdot g(x,y,z)$; , donde t recibe el nombre de multiplicador de Lagrange.

Hemos de resolver el sistema $\{F[x] = 0, F[y] = 0, F[z] = 0, F[t] = 0\}$ y una vez obtenido el (los) punto(s) estacionario(s) P , estudiar la secuencia de menores principales de la matriz hessiana de la función $F(x,y,z,t)$; , en dicho(s) punto(s) P .