MANUAL DE PRÁCTICAS

Matemáticas I

Prácticas con Maxima

María Muñoz Guillermo

Prácticas de Matemáticas I con Maxima

María Muñoz Guillermo

Edición: 2013
ISBN-10: 84-695-7985-1
ISBN-13: 978-84-695-7985-5

Índice general

Práctica 1

Introducción a Maxima

1.1. ¿Qué es Maxima?

Maxima es un sistema para la manipulación de expresiones simbólicas y numéricas, incluyendo diferenciación, integración, expansión en series de Taylor, transformadas de Laplace, ecuaciones diferenciales ordinarias, sistemas de ecuaciones lineales, vectores, matrices y tensores. En resumen, Maxima produce resultados con alta precisión usando fracciones exactas y representaciones con aritmética de coma flotante arbitraria. Maxima nos permite realizar cálculos y gráficas con altas prestaciones.

Maxima es un potente motor de cálculo simbólico aunque, en su origen, no destacaba por tener una interfaz gráfica más amigable para los usuarios que la simple consola de texto. Con el tiempo este hecho ha ido cambiando y han aparecido distintos entornos de ejecución que intentan facilitar la interacción con los usuarios. Utilizaremos *wxMaxima* como interfaz lo hace más sencillo la utilización de Maxima.

1.2. Instalación y localización

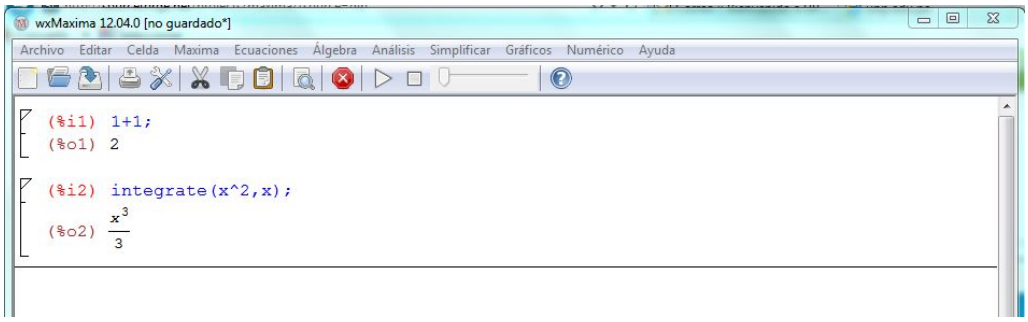
El código fuente de Maxima puede ser compilado sobre varios sistemas incluyendo Windows, Linux y MacOS X. En

http://sourceforge.net/projects/wxmaxima/

encontramos *wxMaxima* así como documentación relativa.

1.3. Descripción del entorno de trabajo

Cada documento de *wxMaxima* está formado por celdas. Las celdas son los elementos básicos. Cada celda tiene un corchete en el borde izquierdo del documento indicando donde empieza y dónde acaba la celda. Las celdas pueden ser de diferente tipo: "title", "text", "section", "input",... Las celdas que utilizaremos para realizar cálculos son las de tipo *input*. El código introducido en estas celdas se envía a *Maxima*, es decir, se realiza el cálculo, pulsando las teclas SHIFT+ENTER. Cada línea de código debe acabar con "$" o ";". En el caso en el que no hayamos añadido alguno de estos símbolos, *wxMaxima* añade un ";" al final de la línea.



Para editar una celda basta hacer click en ella. Para indicar que la celda está siendo editada el corchete de la izquierda aparece en color rojo. Observamos además que de forma automática cada una de las celdas tiene un número asignado cuando se evalúan, para el input y el output.

1.4. Aritmética básica

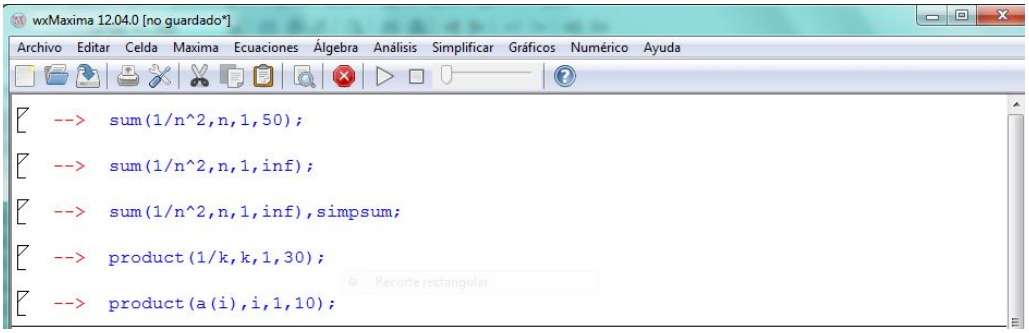
Las operaciones aritméticas básicas utilizan la notación habitual.

- **Sumar**: `x + y`.
- **Restar**: `x - y`.
- **Multiplicar**: `x * y`.
- **Dividir**: `x/y`, equivale a $\frac{x}{y}$.
- **Potencia**: `2 ^6` o `2 **6`. Cuyo equivalente es 2^6.

Otros comandos son:

- `sum(expresion, i, imin, imax)`. Equivale a $\sum_{i=imin}^{i=imax} f(i)$ donde i recorre el conjunto $\{imin, imin+1, imin+2, \ldots, imax\}$.
- `product(expresion, i, imin, imax, salto)`. Idéntico al anterior sólo que con el producto.

En algunos casos es necesario simplicar la expresión que aparece en la suma o en el producto. Para que lo haga de forma explícita hay que especificarlo con las variables globales `simpsum` y `simpproduct`, que se aplican a la suma y al producto respectivamente. Teclee el siguiente ejemplo:



Además hay que considerar algunas reglas adicionales.
Para agrupar operaciones utilizaremos siempre PARÉNTESIS.

1.5. Constantes

Constantes ya incorporadas en *Maxima*:

- La unidad imaginaria i: `%i`
- El número π representado como `%pi`.
- La constante ∞, `inf`.
- La proporción aúrea, `%phi`, cuyo valor es $\frac{1+\sqrt{5}}{2}$.
- El número e, `%e`.

1.6. Funciones incluidas en *Maxima*

Funciones numéricas

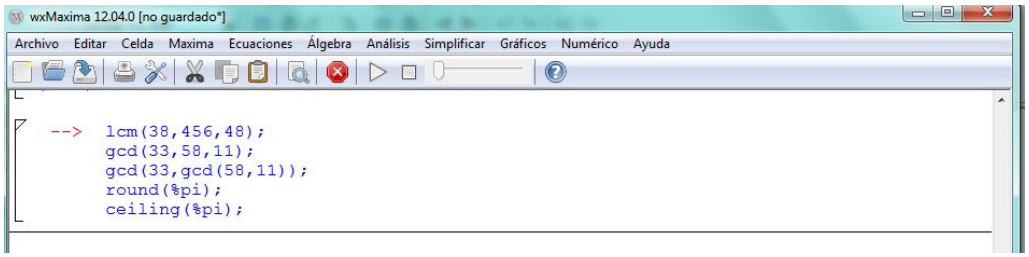
Maxima lleva integradas funciones que nos permiten la manipulación de los datos numéricos.

- `round(x)`. Redondea x a un entero.
- `floor(x)`. Mayor entero menor que x.
- `ceiling(x)`. Menor entero mayor que x.
- `signum(x)`. Signo de x.
- `abs(x)`. Valor absoluto/Módulo.
- `max(x1, ..., xn)`. Máximo del conjunto $\{x_1, \ldots, x_n\}$.
- `min(x1, ..., xn)`. Mínimo del conjunto $\{x_1, \ldots, x_n\}$.

Funciones con enteros

- `divisors(n)`. Lista de los divisores enteros de n.
- `factor(n)`. Lista de los factores primos de n.
- `factorial(n)` o `n!`. Factorial de n.
- `binomial(m, n)`. Número combinatorio m sobre n.
- `remainder(n, m)`. Resto de la división de n entre m.
- `quotient(n, m)`. Da el cociente de la división de n entre m.
- `gcd(n1, ..., ns)`. Máximo común divisor de $\{n_1, \ldots, n_s\}$.
- `lcm(n1, ..., ns)`. Mínimo común múltiplo de $\{n_1, \ldots, n_s\}$.

Teclea los siguientes comandos:



Funciones trigonométricas

- `sin(x)`, `asin(x)`.
- `cos(x)`, `acos(x)`.
- `tan(x)`, `atan(x)`.
- `csc(x)`.
- `sec(x)`.
- `cot(x)`.

Funciones hiperbólicas

- `sinh(x)`, `asinh(x)`.
- `cosh(x)`, `acosh(x)`.
- `tanh(x)`, `atanh(x)`.

Funciones exponencial y logarítmicas

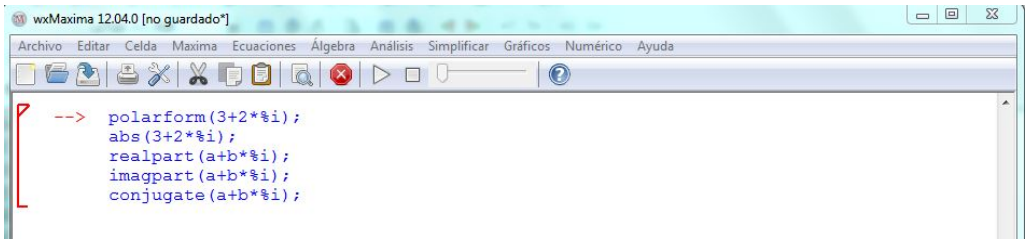
- `exp(x)`. e^x.
- `log(x)`. Función logaritmo en base e (logaritmo neperiano).
- `log(x)/log(b)`. Logaritmo de x en base b.

Funciones con números complejos

Sea z un número complejo.

- `abs(z)`. Módulo de z.
- `arg(z)`. Argumento de z.
- `conjugate(z)`. Conjugado de z.
- `realpart(z)`. Parte real de z.
- `imagpart(z)`. Parte imaginaria de z.
- `polarform(z)`. Forma polar de z.
- `rectform(z)`. Forma binómica de z.

Teclea el siguiente ejemplo:

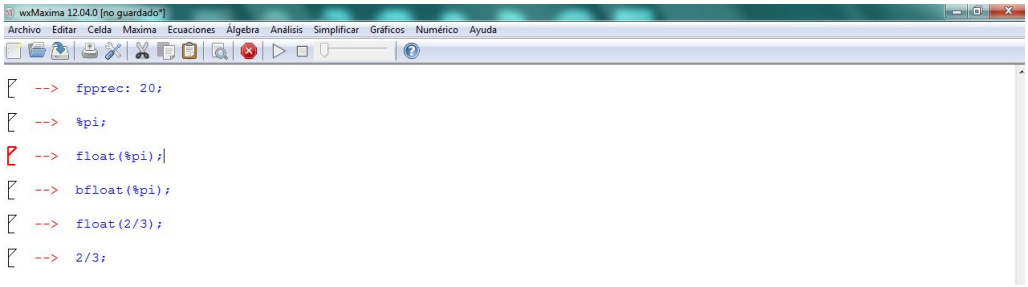


1.7. Aproximaciones numéricas. Precisión

Si observamos el menú de *WxMaxima* podemos advertir que aparece un apartado *Numérico*. Los comandos que utilizaremos son los siguientes:

- `float(x)`. Da la expresión decimal de x (por defecto, con 16 dígitos).
- `x, numer`. Da la expresión decimal de x.
- `bfloat(x)`. Da la expresión decimal "larga" de un número.
- `fpprec` : precisión. Podemos cambiar la precisión, que por defecto es 16.

Teclea los siguientes comandos:

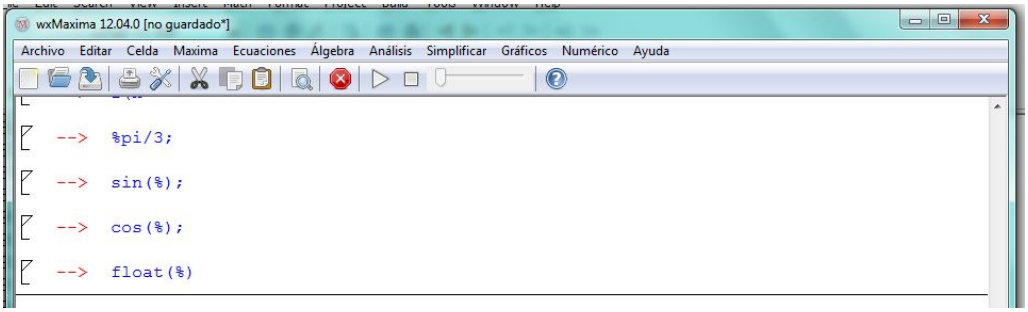


1.8. Utilizando resultados previos

Observamos que cuando ejecutamos algún resultado automáticamente se le asigna un número. Ello nos permite utilizarlos en un determinado momento simplemente haciendo mención de la celda en la que se encuentran.

- % . Se refiere a la celda anterior.
- %onúmero. Celda numerada por "número".

Teclee el siguiente ejemplo:



Ejercicio 1.1 *Realizar los siguientes cálculos:*

- $\cos\left(\frac{\pi}{6}\right)$;
- $\sqrt{50}$;
- $\sqrt[3]{50}$;
- $\text{sen}(30^o)$;
- $\text{sen}(30)$;
- $|3-2i|$;
- $\frac{1}{\infty}$;
- $\sqrt{5}$ *con* 100 *cifras decimales.*

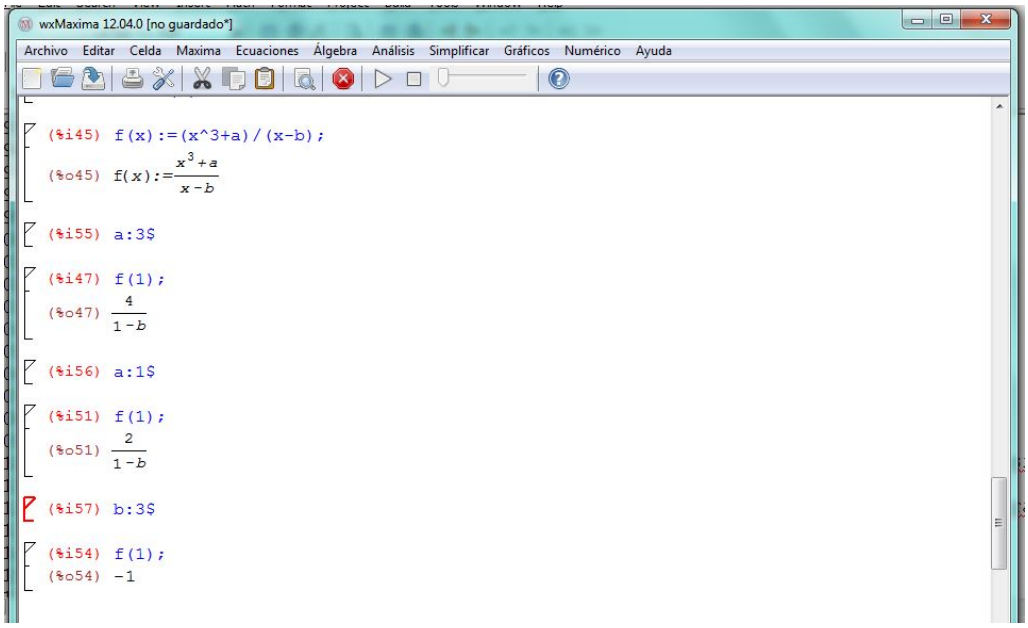
Ejercicio 1.2 *Descomponer en factores primos* 98609.

Ejercicio 1.3 *Calcular la forma binómica del número complejo* $z = 3e^{\frac{\pi}{3}i}$.

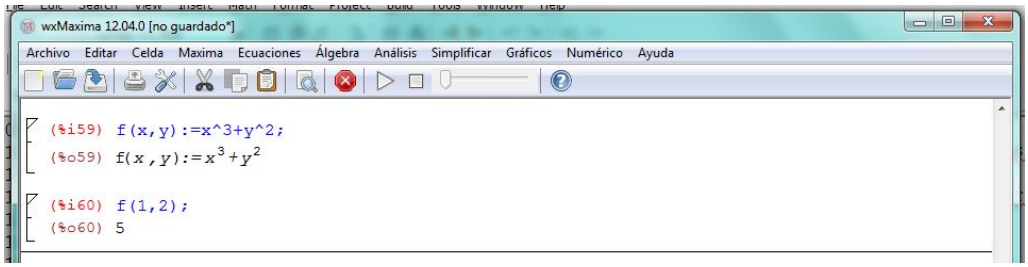
1.9. Asignación de valores y definición de funciones

Para asignar un valor utilizaremos ":", mientras que para definir una función necesitamos ":=".

Definamos la función $f(x) = \frac{x^3+a}{x-b}$ y asignemos distintos valores a los parámetros a y b.



Podemos definir funciones de varias variables simplemente separando con una coma dichas variables. Veamos cómo definir la función $f(x, y) = x^3 + y^2$.



Otros comandos útiles en la definición de variables y funciones son los que siguen:

- `kill(a, b, c, ...)`. Borra todos los valores de los parámetros a, b, c, ...
- `remvalue(var1, var2, ...)`. Borra todos los valores de las variables $var1$, $var2$, ... utilizadas como argumento.
- `values`. Muestra una lista con las variables con valor asignado.

1.10. Utilizando la ayuda de *Maxima*

Disponemos de algunos comandos para obtener ayuda sobre el funcionamiento de *Maxima*. A través de "Ayuda" en el menú tenemos algunas funciones que nos dan información sobre funciones y nos proporcionan ejemplos en el caso en el que así lo requiramos.

- `describe(expr)` o `?expr`. Describe el uso de `expr`.
- `example(expr)`. Muestra un ejemplo de utilización de `expr`.
- `apropos("expr")`. Muestra una serie de comandos relacionados con `expr`.
- `??expr`. Comandos que contienen en su nombre `expr`.

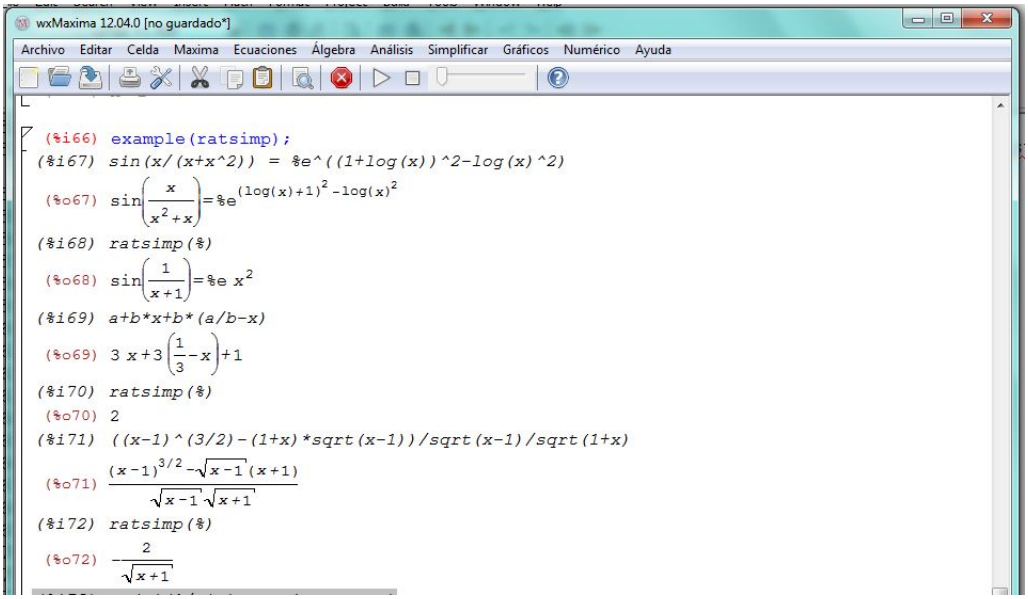
Ejercicio 1.4 *Teclea ??sin y ejecuta este comando. Observa el resultado.*

1.11. Expresiones simbólicas

En el menú de *WxMaxima* disponemos de una pestaña ("Simplificar") con las distintas opciones de simplificación.

- `ratsimp(expresion)`. Simplifica la expresión. A veces hay que aplicarla varias veces. `fullratsimp`
- `expand(expresion)`. Desarrolla los productos y las potencias.
- `factor(expresion)`. Factoriza una expresión.
- `partfrac(frac, var)`. Descompone en fracciones simples respecto de la variable `var`.
- `radcan(expresion)`. Reduce la expresión a común denominador.
- `simp`. Por defecto toma el valor `true`. Si la definimos `false` no se realizarán las simplificaciones.
- `ev(expresion, asig1, asig2)`. Asigna los valores a la expresión.
- `subs(a, b, c)`. Sustituye `a` por `b` en `c`.
- `trigexpand(expresion)`. Expande las expresiones trigonométricas e hiperbólicas.
- `trigsimp(expresion)`. Simplifica las funciones trigonométricas e hiperbólicas utilizando $\text{sen}^2(x) + \cos^2(x) = 1$ y la correspondiente igualdad hiperbólica.
- `trigreduce(expresion, x)`. Combina productos y potencias de senos y cosenos trigonométricos e hiperbólicos de x, transformándolos en otros que son múltiplos de x. También intenta eliminar estas funciones cuando aparecen en los denominadores. Si no se introduce el argumento x, entonces se utilizan todas las variables de expresion.
- `logexpand`. La variable `logexpand` puede tomar los valores `false`, `true` (por defecto) y `super`. La primera opción no realiza transformaciones logarítmicas, la segunda realiza algunas y la tercera las realiza todas.
- `logcontract(expresionlog)`. Permite comprimir expresiones logarítmicas.

Veamos unos ejemplos. El primero es relativo a la función o comando `ratsimp`. Teclea los comandos para comprobar el resultado.



Ejercicio 1.5 *Teclea la expresión:*

$$\texttt{ev(x + y, x : 1, y : 1)}$$

Ejercicio 1.6 *Descompón en fracciones simples la expresión:* $\dfrac{x}{x^3+4x^2+5x+2}$.

Ejercicio 1.7 *Factoriza el siguiente polinomio:* $x^5+11x^4+41x^3+59x^2+40x+48$.

Con el comando `multiplicities` podemos obtener la multiplicidad de cada una de las raíces.

1.12. Relaciones binarias y Símbolos Lógicos

- `x = y`. $x = y$.
- `x < y`. $x < y$.
- `x > y`. $x > y$.

1.13. Comandos útiles en programación

Mostramos sólo algunos comandos, los cuáles serán suficientes para comenzar a trabajar con *Maxima*. Iremos añadiendo aquellos que necesitemos.

- `if condicion then t else f;`. Devuelve t si la condición `condicion` es verdadera y f si la condición es falsa.
- `for variable : valor1 thru valor2 do expresion`.
- `for variable : valor1 while condicion do expresion`.
- `for variable : valor1 unless condicion do expresion`.
- `while condicion do expresion`. Evalúa la condición y mientras ésta es verdadera ejecuta la expresión *expresion*.
- `print(expresion)`. Aparece en la pantalla *expresion*.
- `return(expresion)`. Devuelve el valor *expresion* de una función.

Ejercicio 1.8 *Teclea el siguiente comando*

for i:0 thru 10 do(a:i^ 2,print("El cuadrado de " ,i , " es ", a));

1.14. Gráficos

Por último vamos a comentar brevemente que podemos realizar gráficos de una forma sencilla simplemente a través del menú->Gráficos->, donde hemos de seleccionar si el gráfico es en 2 o 3 dimensiones.

Ejercicio 1.9 *Dibuja la función* $f(x) = \frac{\cos(x^3+3x)}{x^2+1}$ *en el intervalo* $(-10, 10)$.

Ejercicio 1.10 *Dibuja la función* $f(x, y) = x^2 - y^2$ *en el conjunto* $(-4, 4) \times (-4, 4)$.

Veremos más adelante los gráficos con más detalle.

1.15. Ejercicios

Ejercicio 1.11 *Dibuja la función* $f(x) = x^2 \operatorname{sen}(x)$.

Ejercicio 1.12 *Define la función* $f(x) = \frac{x^3-1}{x+1}$ *y dibújala.*

Ejercicio 1.13 *Definir una función que represente una familia de circunferencias de radio a y centro (p, q).*

Ejercicio 1.14 *Define la función*

$$f(x) = \begin{cases} x^3 - 4x & si \ -1 \leq x < 0 \\ \log(x+5) & si \ 0 \leq x \leq 1 \end{cases}$$

y dibújala.

Ejercicio 1.15 *Con el comando* `append` *genera una lista con las funciones* $\{\text{sen}(nx) : n = 1, \ldots, 6\}$. *Dibújalas en unos ejes.*

Práctica 2

Álgebra lineal. Matrices y sistemas de ecuaciones

En esta práctica introduciremos algunos comandos que nos permitan el uso de matrices y vectores.

2.1. Matrices. Operaciones básicas

Lo primero que veremos es cómo introducir matrices. Disponemos de distintas opciones. La primera es escribir:

`matrix`$([a_{1,1}, a_{1,2}, ..., a_{1,n}], [a_{2,1}, a_{2,2}, \ldots, a_{2,n}], \ldots, [a_{m,1}, a_{m,2}, \ldots, a_{m,n}])$

Es decir, introducimos cada fila entre corchetes, separando los elementos con comas. Si observamos el menú de *wxMaxima* podemos comprobar que existe una pestaña con el nombre "Álgebra". Pinchando en dicha opción disponemos de varias opciones:

- Generar matriz. `genmatrix`
- Generar matriz a partir de una expresión. `genmatrix`
- Introducir matriz. `matrix`

La última opción "Introducir matriz" utiliza el comando que hemos descrito al principio con la salvedad que podemos seleccionar si la matriz es general, diagonal, simétrica o antisimétrica, de forma que sólo tendremos que introducir parte de los datos. Además aparece en pantalla la disposición rectangular y sólo hemos de escribir en los huecos.

Ejercicio 2.1 *1. Pulsa en "Introducir matriz" y selecciona en simétrica para introducir la matriz*

$$A = \begin{pmatrix} 1 & -1 & 0 & 1 \\ -1 & 0 & -1 & 2 \\ 0 & -1 & 2 & 5 \\ 1 & 2 & 5 & 3 \end{pmatrix}.$$

2. Genera una matriz B de tamaño 4×3 tal que $a_{i,j} = i^2 j$.

3. Genera una matriz de la forma:

$$C = \begin{pmatrix} a_{1,2} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{pmatrix}.$$

Las operaciones habituales entre matrices las realizaremos de la forma:

- Suma. +
- Resta. -
- Producto de matrices (no conmutativo). ·. Recordamos que la multiplicación de números reales es con el símbolo asterico. Dada una matriz A, para denotar A^2 escribiremos $A\hat{}\,\hat{}\,2$.
- Producto por escalares. *.

Mostramos a continuación algunas funciones y variables que nos serán de utilidad. Algunas de estas funciones requieren para su funcionamiento cargar paquetes adicionales. En concreto el paquete `eigen` es necesario para las funciones: `innerproduct`, `unitvector`, `columnvector`, `gramschmidt`,...Para cargarlo basta con ejecutar:

```
load("eigen");
```

2.1.1. Manipulación de matrices

Sea A una matriz.

- `col(A, i)`. Devuelve la columna i de la matriz A.
- `row(A, i)`. Devuelve la fila i de la matriz A.
- `columnvector([v1, ..., vn])`. Devuelve un vector columna con los datos introducidos en la lista. Hay que cargar previamente el paquete `eigen`.
- `addcol(A, lista1, ..., listan)`. Añade las columnas dadas por las listas a la matriz A.
- `addrow(A, lista1, ..., listan)`. Añade las filas dadas por las listas a la matriz A.
- `adjoint(A)`. Devuelve la matriz de adjuntos de la matriz A.
- `minor(A, i, j)`. Devuelve el menor (i, j) de la matriz A.
- `determinant(A)`. Calcula el determinante de A.
- `transpose(A)`. Traspuesta de `A`.
- `invert(A)`. Matriz inversa de `A`.
- `ident(n)`. Matriz identidad de tamaño $n \times n$.
- `rank(A)`. Calcula el rango de la matriz A.
- `invert(A)`. Calcula la inversa de la matriz A. Otra forma es $A\hat{}\hat{}(-1)$.
- `zeromatriz(n, m)`. Matriz nula de tamaño $m \times n$.

2.1.2. Resolución de sistemas de ecuaciones

Comandos relacionados con la resolución de sistemas de ecuaciones.

Sea A una matriz.

- `linsolve`([`eq1`, ..., `eqn`], [`x`, ..., `xm`]). Resuelve el sistema de ecuaciones lineales.
- `echelon`(`A`). Devuelve la forma escalonada de la matriz A utilizando el método de Gauss.
- `coefmatrix`([`eq1`, ..., `eqm`], [`x1`, ...`xn`]). Devuelve la matriz de coeficientes del sistema de ecuaciones.
- `nullspace`(`A`). Devuelve una base del sistema homogéneo con matriz de coeficientes A, es decir, una base del núcleo de la aplicación lineal con matriz A.

2.1.3. Diagonalización

Comandos relacionados con la diagonalización de matrices.

Sea A una matriz.

- `charpoly`(`A`, `x`). Calcula el polinomio característico respecto de la variable x.
- `eigenvalues`(`A`). Devuelve una lista con 2 sublistas. La primera la forman los valores propios de la matriz A y la segunda sus multiplicidades.
- `eigenvectors`(`A`). Devuelve una lista con 2 sublistas. La primera la forman los valores propios con sus multiplicidades, la segunda está formada por los vectores propios asociados.
- `uniteigenvectors`(`A`). Calcula los vectores propios unitarios de la matriz A.

Veamos un ejemplo.

Ejemplo 2.1 *Consideremos el sistema:*

$$\begin{cases} x + 5y &= a \\ 2x + y &= b \end{cases}.$$

Llamaremos `m` a la matriz de coeficientes y resolveremos el sistema de distintas formas. Véamoslo.

```
(%i18) /*Insertamos las ecuaciones*/
       eq1:x+5*y=a$
       eq2:2*x+y=b$
```

```
(%i13) m:coefmatrix([eq1,eq2],[x,y]);
```

$$(\%o13)\ \begin{pmatrix} 1 & 5 \\ 2 & 1 \end{pmatrix}$$

```
(%i4)  ma:addcol(m,[a,b]);
```

$$(\%o4)\ \begin{pmatrix} 1 & 5 & a \\ 2 & 1 & b \end{pmatrix}$$

```
(%i21) /*Método de Gauss*/
       echelon(ma);
```

$$(\%o21)\ \begin{pmatrix} 1 & 5 & a \\ 0 & 1 & -\frac{b-2\,a}{9} \end{pmatrix}$$

```
(%i6)  rank(m);
```

$$(\%o6)\ 2$$

```
(%i7)  /*Regla de Cramer*/
       vector:[a,b];
```

$$(\%o7)\ [a, b]$$

```
(%i8)  invert(m).vector;
```

$$(\%o8)\ \begin{pmatrix} \frac{5\,b}{9} - \frac{a}{9} \\ \frac{2\,a}{9} - \frac{b}{9} \end{pmatrix}$$

```
(%i10) linsolve([eq1, eq2], [x,y]);
```

$$(\%o10)\ [x = -\frac{a - 5\,b}{9}, y = \frac{2\,a - b}{9}]$$

2.2. Ejercicios

Ejercicio 2.2 *Introduce la matriz* $A = \begin{pmatrix} 3 & -1 & 0 \\ 0 & 3 & -2 \\ 1 & -2 & 1 \end{pmatrix}$. *Calcula el rango de la matriz. ¿Es invertible? En caso afirmativo, calcula la inversa.*

Ejercicio 2.3 *Dada la matriz* $A = \begin{pmatrix} 3 & 1 \\ 5 & 2 \end{pmatrix}$, *se pide:*

1. *Hallar* $3A(A^t) - 2I_2$, *siendo* $I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

2. *Resolver la ecuación matricial* $AX = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$, *donde* $X \in \mathcal{M}_2(\mathbb{R})$.

Ejercicio 2.4 *Calcular el rango de la siguiente matriz en función de los valores de* a *y* b.

$$\begin{pmatrix} a & 0 & 0 & b \\ b & a & 0 & 0 \\ 0 & b & a & 0 \\ 0 & 0 & b & a \end{pmatrix}$$

Ejercicio 2.5 *Determinar si los siguientes sistemas son compatibles o incompatibles. Calcúlese la solución en el caso de que ésta exista.*

$$\begin{cases} 8x + y + 4z &= 9 \\ 5x - 2y + 4z &= 6 \\ x + y = 1 \end{cases},$$

$$\begin{cases} 6x - y + 3z &= 6 \\ -6x + 8y &= -10 \\ 2x - 5y - z &= 4 \end{cases}.$$

Ejercicio 2.6 *Discutir y resolver según el valor del parámetro* α.

$$\begin{cases} \alpha x + y + 2z &= 0 \\ x + 3y + z &= 0 \\ 3x + 10y + 4z &= 0 \end{cases}$$

Ejercicio 2.7 *Discutir y resolver los siguientes sistemas en función de los parámetros.*

$$(a) \begin{cases} 3x - y + 2z = 1 \\ x + 4y + z = \beta \\ 2x - 5y + \alpha z = -2 \end{cases}$$

$$(b) \begin{cases} 2y - z = \alpha \\ 3x - 3z = 11 \\ y + z = 6 \\ 2x + y - 4z = \alpha \end{cases}$$

$$(c) \begin{cases} 2x - y - z = 3a \\ x - az = b \\ x - y + 2z = 7 \end{cases}$$

Práctica 3

Espacios Vectoriales y Aplicaciones lineales

En esta práctica utilizaremos los comandos ya introducidos en la práctica anterior aplicados en esta ocasión a los espacios vectoriales.

3.1. Subespacios vectoriales

Veamos algunos ejemplos:

Ejemplo 3.1 *Halla una base del siguiente subespacio y calcula la ecuación cartesiana.*

$$< (1,2,-1),(3,-1,0),(4,1,-1) > \quad de\ \mathbb{R}^3.$$

```
(%i1)  /*En primer lugar introducimos los vectores*/
       v1:[1,2,-1]$
       v2:[3,-1,0]$
       v3:[4,1,-1]$

(%i4)  /*Calculamos el rango de la matriz formada por dichos vectores*/
       m:matrix(v1,v2,v3);
```

$$(\%o4)\quad \begin{pmatrix} 1 & 2 & -1 \\ 3 & -1 & 0 \\ 4 & 1 & -1 \end{pmatrix}$$

```
(%i5)  rank(m);
```

(%o5) 2

```
(%i7)  /*Vamos a calcular una base equivalente*/
       echelon(m);
```

$$(\%o7)\quad \begin{pmatrix} 1 & -\frac{1}{3} & 0 \\ 0 & 1 & -\frac{3}{7} \\ 0 & 0 & 0 \end{pmatrix}$$

```
(%i8)  w1:row(%o7,1);
```

$$(\%o8)\quad \begin{pmatrix} 1 & -\frac{1}{3} & 0 \end{pmatrix}$$

```
(%i9)  w2:row(%o7,2);
```

$$(\%o9)\quad \begin{pmatrix} 0 & 1 & -\frac{3}{7} \end{pmatrix}$$

```
(%i11) /*Una base vendrá dada por los vectores {w1,w2}*/
       /*Vamos a calcular la ecuación cartesiana correspondiente*/
       newma:matrix([1,-1/3,0],[0,1,-3/7],[x,y,z]);
```

$$(\%o11)\quad \begin{pmatrix} 1 & -\frac{1}{3} & 0 \\ 0 & 1 & -\frac{3}{7} \\ x & y & z \end{pmatrix}$$

```
(%i12) determinant(%o11);
```

$$(\%o12)\quad z + \frac{3\,y}{7} + \frac{x}{7}$$

```
-->    /*La ecuación cartesiana viene dada por x+3y+7z=0*/
```

Ejercicio 3.1 *Siguiendo el ejemplo anterior calcula una base y las ecuaciones cartesianas de los siguientes subespacios:*

1. $< (1,0,2), (-2,1,1), (0,1,5) >$ *de* $\mathbb{R}^3$.
2. $< (1,2,3,4,5), (0,-1,1,2,3), (3,2,1,0,-1), (-4,-3,-2,-1,0) >$ *de* $\mathbb{R}^5$.

Veamos a continuación otro ejemplo.

Ejemplo 3.2 *Hallar una base y las ecuaciones implícitas del siguiente subespacio:*

$$V = \begin{cases} x = \lambda \\ y = \lambda + \mu \\ z = \gamma \\ t = \mu \end{cases}$$

```
(%i13) eq1:x=lambda$
       eq2:y=lambda+mu$
       eq3:z=gamma$
       eq4:t=mu$

(%i23) matrizasociada:coefmatrix([eq1,eq2,eq3,eq4],[lambda,mu,gamma]);
```

$$(\%o23)\quad \begin{pmatrix} -1 & 0 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

```
(%i24) rank(%);
```

$$(\%o24)\quad 3$$

```
(%i25) /*Una base viene dada por las columnas de la matriz asociada*/
       transpose(%o23);
```

$$(\%o25)\quad \begin{pmatrix} -1 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

```
(%i26) /*Las filas de la matriz anterior son una base*/
       /*Vamos a calcular ahora la ecuación cartesiana*/
       nuevamatriz:addcol(matrizasociada,[x,y,z,t]);
```

$$(\%o26)\quad \begin{pmatrix} -1 & 0 & 0 & x \\ -1 & -1 & 0 & y \\ 0 & 0 & -1 & z \\ 0 & -1 & 0 & t \end{pmatrix}$$

```
(%i27) determinant(%);
```

$$(\%o27)\quad y - x - t$$

```
-->    /*La ecuación cartesiana viene dada por: y-x-t=0*/.
```

Realiza ahora el siguiente ejercicio:

Ejercicio 3.2 *Calcula una base y las ecuaciones cartesianas del subespacio dado por:*

$$V = \begin{cases} x = \lambda + \alpha + \beta \\ y = \lambda - \alpha + 3\beta \\ z = \lambda + 2\alpha \\ t = 2\lambda + 3\alpha + \beta \end{cases}.$$

Veamos un nuevo ejemplo:

Ejemplo 3.3 *Hallar una base del subespacio dado por:*

$$V = \{(x, y, z, t) \in \mathbb{R}^4 : x - y + z + t = 0, y - z = 0\}.$$

```
(%i13) kill(eq1,eq1,eq3,eq4,x,y,z,t,lambda,mu);
```

(%o13) *done*

```
(%i14) eq1:x-y+z+t=0$
       eq2:y-z=0$

(%i16) linsolve([eq1,eq2],[x,y,z,t]);
```

(%o16) $[x = -\%r4, y = \%r5, z = \%r5, t = \%r4]$

```
-->    /*De esta forma ya tenemos una base dada por
       (x,y,z,t)=%r4(-1,0,0,1)+%r5(0,1,1,0)*/
```

Realiza los siguientes ejercicios:

Ejercicio 3.3 *Hallar una base del subespacio dado por:*

$$V = \{(x, y, z, t) \in \mathbb{R}^4 : x - 3y + t = 0\}.$$

Ejercicio 3.4 *Calcula las ecuaciones cartesianas del subespacio de* $\mathbb{R}^4$ *dada por:*

$$V =< (1, 2, 0, 1) > .$$

Ejercicio 3.5

1. *Sea* $T = \{(x, y, z, t) \in \mathbb{R}^4 : x + y + t = 0, x + 2y - z + t = 0, x - y + z + t = 0, x + y + 2z - at = 0\}$. *Calcula el valor de* a *para que* T *sea un subespacio de dimensión uno.*

2. *Sea*
$$S =< (1, 2, 1, 3), (2, 5, 3, 1), (1, -1, -2, 2) > .$$
Da una base y la dimensión de los subespacios: S, T, $S + T$ *y* $S \cap T$.

3.2. Cambios de base. Coordenadas

Veamos un ejemplo.

Ejemplo 3.4 *Halla las matrices cambio de base* $M_{BB_c}(id)$ *y* $M_{B_cB}(id)$ *en* $\mathbb{R}^3$, *siendo* $B = \{(0, 1, 1), (1, 0, 1), (1, 1, 0)\}$ *y* B_c *la base canónica.*

```
(%i17) /*Introducimos la matriz del cambio de base de B a Bc*/
       /*En primer lugar introducimos los vectores*/
       kill(v1,v2,v3);
```

(%o17) *done*

```
(%i18) v1:[0,1,1]$
       v2:[1,0,1]$
       v3:[1,1,0]$
```

```
(%i21) MBBc:transpose(matrix(v1,v2,v3));
```

$$(\%o21)\quad \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

```
(%i22) MBcB:invert(%o21);
```

$$(\%o22)\quad \begin{pmatrix} -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \end{pmatrix}$$

Ejercicio 3.6 *En el espacio vectorial* $\mathbb{R}^3$ *se considera el sistema de vectores*

$$B = \{(-1,0,1),(0,1,-1),(4,-2,-1)\}.$$

Comprueba que es base de $\mathbb{R}^3$*. Halla la matriz de cambio de base de la base canónica a la base* B*. Obtén las coordenadas en la base* B *del vector cuyas coordenadas en la base canónica son:* $(-1,1,2)$*.*

Ejercicio 3.7 *Sean las base* $B1 = \{(1,0,1),(1,10),(0,1,1)\}$ *y* $B2 = \{(1,2,3),(1,3,0),(3,2,3)\}$ *de* $\mathbb{R}^3$ *hallar las matrices del cambio de base de* $B1$ *a* $B2$ *y de* $B2$ *a* $B1$*.*

3.3. Aplicaciones lineales

Ejemplo resuelto.

Ejemplo 3.5 *Sea* f *un endomorfismo de* $\mathbb{R}^3$ *dado por* $f(x,y,z) = (x+y+z, x+y-z, z)$*. Hallar* $\ker f$*,* $\mathrm{Im} f$ *y* $f(V)$*, donde* $V = \{(x,y,z) : x+y+z = 0\}$*.*

```
(%i1)  /*Endomorfismo*/
       f(x,y,z):=[x+y+z,x+y-z,z];
```

(%o1) $\mathrm{f}\,(x,y,z) := [x+y+z, x+y-z, z]$

```
(%i2)  /*Matriz asociada al endomorfismo*/
       mf:transpose(matrix(f(1,0,0),f(0,1,0),f(0,0,1)));
```

$$(\%o2)\quad \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}$$

```
(%i8)  m2:%o2 . [x,y,z];
```

$$(\%o8)\quad \begin{pmatrix} z+y+x \\ -z+y+x \\ z \end{pmatrix}$$

```
(%i26) /*ker(f)*/
       eq1:m2[1,1]=0;
       eq2:m2[2,1]=0;
       eq3:m2[3,1]=0;
```

$$(\%o26)\ z+y+x=0$$
$$(\%o27)\ -z+y+x=0$$
$$(\%o28)\ z=0$$

```
(%i29) linsolve([eq1,eq2,eq3],[x,y,z]);
```

$$solve : dependentequationseliminated : (1)$$
$$(\%o29)\ [x=\%r1, y=-\%r1, z=0]$$

```
-->    /*Una base del ker(f) viene dada por (1,-1,0)*/

(%i30) /*La imagen viene dada por las imÃ¡genes
       de los vectores de la base*/
       imagen:matrix(f(1,0,0),f(0,1,0),f(0,0,1));
```

$$(\%o30)\quad \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix}$$

```
(%i31) echelon(imagen);
```

$$(\%o31)\quad \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & -\frac{1}{2} \\ 0 & 0 & 0 \end{pmatrix}$$

```
-->    /*La imagen viene generada por los vectores (1,1,0)
       y (0,1,-1/2)*/
```

```
-->    /*Por último para calcular f(V) tenemos que calcular
       una base de V*/

(%i32) linsolve(x+y+z=0,[x,y,z]);
```

$$(\%o32)\ [x = -\%r3 - \%r2, y = \%r3, z = \%r2]$$

```
(%i33) /*Una base viene dada por (-1,1,0) y (-1,0,1)*/
       fV:matrix(f(-1,1,0),f(-1,0,1));
```

$$(\%o33)\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & -2 & 1 \end{pmatrix}$$

```
-->    /*Luego f(V) viene generado por el vector (0,-2,1).*/
```

Veamos un nuevo ejemplo.

Ejemplo 3.6 *Se considera el homomorfismo* $f : \mathbb{R}^3 \to \mathbb{R}^2$*, que hace corresponder a los vectores* $(1,0,1), (0,1,1)$ *y* $(1,1,0)$ *los vectores* $(0,1), (0,2)$ *y* $(1,1)$ *respectivamente. Hallar la matriz asociada a* f *en las bases canónicas de* $\mathbb{R}^3$ *y* $\mathbb{R}^2$*.*

```
(%i12) kill(f);
```

$$(\%o12)\ done$$

```
(%i14) /*La matriz asociada f considerando la base B2={(1,0,1),
       (0,1,1),(1,1,0)} en el espacio R^3 y la base canónica en el espacio
       R^2*/
       matrizB2Bcf:transpose(matrix([0,1],[0,2],[1,1]));
```

$$(\%o14)\ \begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 1 \end{pmatrix}$$

```
(%i16) /*La matriz del cambio de base de B2 a la base canónica*/
       MB2BI:transpose(matrix([1,0,1],[0,1,1],[1,1,0]));
```

$$(\%o16)\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

```
(%i17) /*Calculamos la matriz asociada a f en las bases canónicas*/
       MBcBcf:matrizB2Bcf . invert(MB2BI);
```

$$(\%o17)\ \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ 0 & 1 & 1 \end{pmatrix}$$

Realiza el siguiente ejercicio.

Ejercicio 3.8 *Sea f un endomorfismo de $\mathbb{R}^3$ dado por $f(x, y, z) = (z + y, -x + y - z, z)$. Dadas las bases de $\mathbb{R}^3$*

$$C = \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$$

y

$$B = \{(1, 1, 1), (1, 1, 0), (1, 0, 0)\}$$

hallar $M_{CC}(f), M_{BC}(f), M_{BC}(f)$ y $M_{CB}(f)$.

Práctica 4

Gráficos, límites, derivadas e integrales

4.1. Gráficos

En la práctica número 1 vimos que podemos dibujar con *wxMaxima*. Veamos un poco más al respecto.

4.1.1. Formatos gráficos

- `gnuplot` Formato por defecto para Windows.
- `gnuplot_pipes` Formato por defecto para plataformas distintas de Windows.
- `mgnuplot` Mgnuplot es una interfaz para Gnuplot basada en Tk.
- `xmaxima` Xmaxima es un interfaz gráfico TCl/Tk de Maxima, que también se puede utilizar para representar gráficos cuando Maxima se ejecuta desde la consola o desde otros interfaces.

4.1.2. Funciones y variables para gráficos

`contour_plot`(`expr`, $\texttt{x}_{\texttt{range}}$, $\texttt{y}_{\texttt{range}}$, `options`, ...) Dibuja las curvas de nivel `expr` en el rectángulo $x_{range} \times y_{range}$. Para aumentar el número de niveles es necesario modificar `cntrparamlevels`. Ver ejemplo

Ejemplo 4.1 *Consideramos la función* $f(x, y) = x^2 - y^2$. *Vamos a dibujar las curvas de nivel de dicha función. Teclea los comandos que siguen a continuación.*

```
(%i1)  /*Definimos f*/
       f(x,y):=x^2-y^2;
```

(%o1) $\mathrm{f}(x,y) := x^2 - y^2$

```
(%i4)  contour_plot(f(x,y),[x,-3,3],[y,-3,3])$
```

```
(%i5)  contour_plot(f(x,y),[x,-3,3],[y,-3,3],
       [gnuplot_preamble,"set cntrparam levels 20"])$
```

- plot2d(plot, $\mathtt{x}_{\mathtt{range}}$, ..., options, ...)
- plot2d([$\mathtt{plot}_1$, ..., $\mathtt{plot}_n$], ..., options, ...)
- plot2d([$\mathtt{plot}_1$, ..., $\mathtt{plot}_n$], $\mathtt{x}_{\mathtt{range}}$, ..., options, ...)

Donde plot, $\mathtt{plot}_1$, ..., $\mathtt{plot}_n$ pueden ser expresiones, nombres de funciones o una lista de cualquiera de las siguientes formas:

- [discrete, [x1, ..., xn], [y1, ..., yn]]
- [discrete, [[x1, y1], ..., [xn, yn]]]
- [parametric, $\mathtt{x}_{\mathtt{expr}}$, $\mathtt{y}_{\mathtt{expr}}$, $\mathtt{z}_{\mathtt{expr}}$]

En el caso de dibujar un conjunto de puntos, éstos aparecen unidos entre sí, para que aparezcan simplemente puntos escribiremos [style,points].

Veamos algunos ejemplos:

```
-->    /*Introducimos una lista de puntos*/
       datos:[[1,3.5],[1.3,4.2],[1.5,5],[1.8,1],[2,4.5]];
```

```
(%i7)  plot2d([discrete,datos])$
```

Añade al comando anterior la opción [style, points].

```
(%i8)  /*Coordenadas paramétricas*/
       plot2d([parametric,cos(t)*sin(t),sin(t)],[t,0,2*%pi])$
```

Ejercicio 4.1 *Dibuja la siguiente curva:*

$$f(t) = \left((e^{\cos(t)} - 2\cos(4t) - \mathrm{sen}^5(t/12))\,\mathrm{sen}(t), (e^{\cos(t)} - 2\cos(4t) - \mathrm{sen}^5(t/12))\cos(t)\right),$$

en el intervalo $(-8\pi, 8\pi)$*, utilizando la opción* [nticks, 2000].

De forma similar tenemos el comando para 3 dimensiones.

- `plot3d(expr, x_range, y_range, ..., options, ....)`
- `plot3d([expr1, ...., exprn], x_range, y_range, ..., options, ...)`

Ejemplo 4.2 *Teclea:*

```
-->   plot3d(x^2-y^2,[x,-4,4],[y,-4,4])$
```

Añade la opción [`palette`, `false`].

Para ver más opciones gráficas véase el manual de ayuda de *Maxima*.

4.2. Límites y continuidad

Cálculo de límites.

- `limit(expr, x, val, dir)`
- `limit(expr, x, val)`
- `limit(expr)`

Calcula el límite de `expr` cuando la variable real x se aproxima al valor `val` desde la dirección `dir` que puede ser `plus`(por la derecha) o `minus` (por la izquierda). Si se omite la dirección se habla de límite en ambos sentidos.

La función `limit` utiliza los símbolos:

- `inf` (más infinito).
- `minf` (menos infinito).
- `und` (indefinido).
- `ind` (indefinido pero acotado).

Veamos algunos ejemplos.

```
(%i7)  f(x):=log(x)/x;
```

$$(\%o7)\quad \mathrm{f}\,(x) := \frac{\log\,(x)}{x}$$

```
(%i8)  limit(f(x),x,inf);
```

$$(\%o8)\quad 0$$

```
(%i9)  limit(f(x),x,0,plus);
```

(%o9) $-\infty$

Ejercicio 4.2 *Calcular los límites de las siguientes funciones cuando x (o n) tiende a ∞.*

- $f(x) = \frac{(x+1)^2}{2x^2}$;
- $f(n) = \frac{(n+2)!+(n+1)!}{(n+3)!}$;
- $f(x) = \sqrt{x^2-2x+1} - \sqrt{x^2-7x+3}$;
- $f(x) = \left(\frac{2}{x+1}\right)^{\frac{2}{2+\log(x)}}$;

4.3. Derivadas

Los comandos básicos para derivar son los que siguen:

- `diff(expr, x)`. Devuelve la derivada respecto de x.
- `diff(expr, x, n)`. Devuelve la derivada n-ésima respecto de x.
- `diff(expr, x1, n1, ...., xm, nm)`. Devuelve la derivada parcial de `expr` con respecto de $x1$ $n1$ veces,...., respecto de xm nm veces.
- `diff(expr)`. Devuelve la diferencial total de `expr`.

Ejercicio 4.3 *Calcula la derivada n-ésima de:*

- $f(x) = x^m$ *para* $n = 7$;
- $f(x) = \tan(x)$ *para* $n = 10$;
- $f(x) = e^{\frac{\cos(x)+\mathrm{sen}(x)}{x^2}}$ *para* $n = 8$;

4.4. Integrales

- `integrate(expr, x)`. Calcula la primitiva.
- `integrate(expr, x, a, b)`. Calcula la integral definida en el intervalo (a, b).

Ejercicio 4.4 *Calcula las siguientes integrales:*

- $\int \frac{1}{(1-x^2)^3} dx$;
- $\int \frac{1}{\cos(x)^3} dx$;
- $\int \sqrt{1+x^2} dx$;
- $\int_{-1}^{1} \frac{1}{1+x^2} dx$;

Veamos un ejemplo de aplicación.

Ejemplo 4.3 *Calcula el área de la región comprendida entre las funciones* $f(x) = x^3 - 3x^2 + 1$ *y* $g(x) = -x + 1$.

```
(%i2)  /*Insertamos las funciones*/
       f(x):=x^3-3*x^2+1$
       g(x):=-x+1$

(%i12) /*Dibujamos las funciones en el intervalo (-1,3)*/
       plot2d([f(x),g(x)],[x,-1,3])$

(%i5)  /*Observamos que tenemos dos puntos de corte en este intervalo*/
       /*Calculamos los puntos de corte*/
       solve(f(x)-g(x),x);
```

$$(\%o5)\quad [x = -\frac{\sqrt{5}-3}{2}, x = \frac{\sqrt{5}+3}{2}, x = 0]$$

```
(%i11) float(%o5);
```

$$(\%o11)\quad [x = 0{,}38196601125011, x = 2{,}618033988749895, x = 0{,}0]$$

```
(%i17) /*Calculamos el área*/
       area:integrate(f(x)-g(x),x,0,-(sqrt(5)-3)/2)+
       integrate(g(x)-f(x),x,-(sqrt(5)-3)/2,(sqrt(5)+3)/2);
```

$$(\%o17)\quad \frac{5^{\frac{3}{2}}+11}{8} + \frac{5^{\frac{3}{2}}-11}{4}$$

```
(%i18) float(area);
```

(%o18) 2,817627457812107

4.5. Ejercicios

Ejercicio 4.5 *Demostrar que la ecuación* $e^x = 1 + x$ *tiene por única solución real* $x = 0$.

Ejercicio 4.6 *Dada la curva* $y = \frac{x-4}{x-2}$ *razonar que las tangentes a la curva en los puntos de corte con los ejes coordenados son paralelas. ¿Cuál es su pendiente?*

Práctica 5

Polinomio de Taylor. Método de Newton

5.1. Desarrollo en serie

Dada una función, $f : \mathbb{R} \to \mathbb{R}$, si admite derivada de todo orden en un entorno de un punto $x = a$, se llama desarrollo en serie de Taylor de $f(x)$ en el punto a, a su expresión en forma de serie de potencias:

$$f(x) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{n)}(a)(x-a)^n.$$

Si $a = 0$, el desarrollo se llama de MacLaurin.

El comando que utilizaremos en *Maxima* para el desarrollo en serie de Taylor hasta el término n-ésimo de la función $f(x)$ alrededor del punto x_0 es:

`taylor(f(x), x, x0, n)`

Sin embargo, no podemos utilizar directamente el resultado por ello utilizaremos la función `subst()`.

```
(%i19) /*Definimos una función*/
       f(x):=sin(x);
```

(%o19) $\mathrm{f}\,(x) := \sin(x)$

```
(%i20) taylor(f(x),x,0,5);
```

(%o20) $x - \frac{x^3}{6} + \frac{x^5}{120} + \ldots$

```
(%i21) p5(x):=subst(t=x,taylor(f(t),t,0,5));
```

(%o21) $\mathrm{p5}\,(x) := \mathrm{subst}\,(t = x, \mathrm{taylor}\,(\mathrm{f}\,(t)\,, t, 0, 5))$

```
(%i22) p5(x);
```

(%o22) $\dfrac{x^5}{120} - \dfrac{x^3}{6} + x$

Vamos a calcular en una tabla distintos polinomios de Taylor de la función $f(x) = \mathrm{sen}(x)$. Después evaluaremos los polinomios en $x = 0{,}01$ para obtener una aproximación del valor de sen(0,01).

```
(%i23) tabla:[]$
```

```
(%i25) for n:5 thru 10 do(tabla:append(tabla,
       [subst(t=x,taylor(f(t),t,0,n))]));
```

(%o25) *done*

```
(%i26) tabla;
```

(%o26) $[\dfrac{x^5}{120} - \dfrac{x^3}{6} + x, \dfrac{x^5}{120} - \dfrac{x^3}{6} + x, -\dfrac{x^7}{5040} + \dfrac{x^5}{120} - \dfrac{x^3}{6} + x, -\dfrac{x^7}{5040} + \dfrac{x^5}{120} - \dfrac{x^3}{6} + x, \dfrac{x^9}{362880} - \dfrac{x^7}{5040} + \dfrac{x^5}{120} - \dfrac{x^3}{6} + x, \dfrac{x^9}{362880} - \dfrac{x^7}{5040} + \dfrac{x^5}{120} - \dfrac{x^3}{6} + x]$

```
-->    append(tabla,[f(x)]);
```

(%o30) $[\dfrac{x^5}{120} - \dfrac{x^3}{6} + x, \dfrac{x^5}{120} - \dfrac{x^3}{6} + x, -\dfrac{x^7}{5040} + \dfrac{x^5}{120} - \dfrac{x^3}{6} + x, -\dfrac{x^7}{5040} + \dfrac{x^5}{120} - \dfrac{x^3}{6} + x, \dfrac{x^9}{362880} - \dfrac{x^7}{5040} + \dfrac{x^5}{120} - \dfrac{x^3}{6} + x, \dfrac{x^9}{362880} - \dfrac{x^7}{5040} + \dfrac{x^5}{120} - \dfrac{x^3}{6} + x, \sin\,(x)]$

```
(%i38) plot2d(tabla,[x,-3,3]);
```

(%o38)

```
(%i39) subst(x=0.01,tabla);
```

(%o39) [0,0099998333341667, 0,0099998333341667, 0,0099998333341667, 0,0099998333341667, 0,0099998333341667, 0,0099998333341667]

Ejercicio 5.1 *Construir una tabla con los polinomios de Taylor de grado 5 hasta grado 8, de la función* $f(x) = \sin(x) - 3cos(x)$ *en el punto* $x = \frac{\pi}{3}$. *Dibujar las gráficas de dichas funciones y observar la aproximación de las mismas.*

5.2. El método de Newton

Este método se basa en la aproximación de una función por su derivada (aproximación de una curva por medio de su tangente, en cada punto).

Dada una curva $f : [a, b] \to \mathbb{R}$, su tangente en un punto $(x_0, f(x_0))$ tiene por ecuación:

$$\texttt{y} - \texttt{f(x0)} = \texttt{f'(x0)(x} - \texttt{x0)}$$

A partir de aquí se construye la sucesión de la forma siguiente:

1. Comenzamos con el punto x_0.
2. El punto x_1 se obtiene como la intersección de la recta tangente $y - f(x_0) = f'(x_0)(x - x_0)$ con el eje $y = 0$. Por tanto,

   $$\texttt{x1} = \texttt{x0} - \texttt{f(x0)/f'(x0)}$$

3. El punto x_2 se obtiene como la intersección de $y - f(x_1) = f'(x_1)(x - x_1)$ con el eje $y = 0$. El punto x_2 vendrá dado por

   $$x2 = x1 - f(x1)/f'(x1)$$

   Reiterando el proceso construimos una sucesión $(x_n)_n$ en la cual cada término viene dado a partir del anterior por la fórmula:

   $$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

El error máximo que se comete en cada paso E_n viene dado por la expresión $E_n \leq E_{n-1}^2 M$, donde

$$M = \text{máx}\{\left|\frac{f''(x)}{2f'(x)}\right| : x \in [a, b]\}.$$

Ejemplo: Obtener una solución aproximada de la ecuación

$$x^5 + 0{,}85x^4 + 0{,}7x^3 - 3{,}45x^2 - 1{,}1x + 1{,}265 = 0$$

En primer lugar borramos las variables anteriores que tenían asignados valores por el ejercicio anterior y definimos la nueva función con la cual trabajaremos. Dibujamos la función para localizar las raíces.

```
(%i1)  /*Introducimos la función*/
       kill(f)$

(%i2)  f(x):=x^5+0.85*x^4+0.7*x^3-3.45 * x^2 -1.1 * x+1.265$

(%i3)  plot2d(f(x),[x,-1.5,1.5]);
```

(%o3)

```
(%i4)  derivadaf(x):=subst(t=x,diff(f(t),t));
```

(%o4) $\text{derivadaf}\,(x) := \text{subst}\,(t = x, \text{diff}\,(\text{f}\,(t)\,, t))$

```
(%i5)  derivadaf(x);
```

(%o5) $5\,x^4 + 3{,}4\,x^3 + 2{,}1\,x^2 - 6{,}9\,x - 1{,}1$

```
(%i6)  x(n):=if n=1 then 0 else x(n-1)-f(x(n-1))/derivadaf(x(n-1))$

(%i7)  tablanewton:[]$

(%i8)  for n:1 thru 5 do(tablanewton:append(tablanewton,[x(n)]));
```

(%o8) *done*

```
(%i9)  tablanewton;
```

(%o9) $[0, 1{,}15, 1{,}15, 1{,}15, 1{,}15]$

Repitiendo el proceso con el punto inicial $x_0 = 0{,}5$ y con el punto $x_0 = 1$ obtenemos distintos resultados. Compruébalo.

Ejercicio 5.2 *Calcular de forma aproximada* $\sqrt{2}$ *como solución de la ecuación* $x^2 - 2 = 0$ *por el método de Newton. Analiza el resultado.*

Bibliografía

[1] Manual de Maxima (en español).
http://maxima.sourceforge.net/docs/manual/es/maxima.pdf

[2] Using wxMaxima tutorial.wxm. Disponible en
http://andrejv.github.io/wxmaxima/help.html