

## 1 Hoja de cálculo

- « ; » evalúa una expresión y devuelve el resultado. Por ejemplo  $1+2/3$  ;
- « \$ » evalúa una expresión sin devolver el resultado. Por ejemplo  $a:2$  \$
- « % » es el valor del último cálculo efectuado
- ? plot2d muestra la ayuda sobre la función plot2d
- example(expand) muestra ejemplos de utilización de la función expand
- kill(all) reinicia el sistema

## 2 Operadores

- + , - , \* , / son las cuatro operaciones usuales
- « ^ » es el operador potencia. Por ejemplo,  $x^3$  es  $x^3$
- « # » es la relación no igual (distinto)
- = , < , <= , > , >= son los operadores de comparación
- « : » es el operador de asignación. Por ejemplo  $a:3$  asigna el valor 3 a la variable  $a$ .
- « := » es el operador para definir una función.
- « = » indica una ecuación en Maxima.
- « ! » es el factorial de un entero natural. Por ejemplo,  $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$ .
- « . » es el producto de dos matrices.

## 3 Constantes

- %pi es el número  $\pi \approx 3,14159$
- %e es el número  $e = \exp(1) \approx 2,7183$
- %i es la unidad imaginaria  $\sqrt{-1}$
- true es el valor lógico “verdadero”
- false es el valor lógico “falso”
- inf es  $+\infty$
- minf es  $-\infty$

## 4 Números reales

### 4.1 Funciones usuales

- abs(x) valor absoluto de  $x$
- floor(x) parte entera de  $x$
- round(x) redondeo entero de  $x$
- sqrt(x) raíz cuadrada de  $x$
- exp(x) es  $e^x$
- log(x) logaritmo neperiano de  $x$
- Funciones trigonométricas:  $\sin(x)$ ,  $\cos(x)$ ,  $\tan(x)$
- Funciones trigonométricas inversas:  $\operatorname{asin}(x)$ ,  $\operatorname{acos}(x)$ ,  $\operatorname{atan}(x)$

## 4.2 Valores aproximados

- `float(x)` es el valor decimal aproximado de  $x$
- `bfloat(x)` es el valor aproximado de  $x$  en notación científica
- `fpprec:20` fija la aproximación dada por `bfloat` a 20 cifras (por defecto es 16)

## 4.3 Trigonometría

- `trigexpand(a)` desarrolla la expresión trigonométrica  $a$  utilizando las fórmulas de la suma de seno y coseno. Por ejemplo, `trigexpand(cos(x+y))` devuelve  $\cos x \cos y - \sin x \sin y$
- `trigreduce(a)` reduce un polinomio trigonométrico  $a$ . Por ejemplo, `trigreduce(sin(x)^3)` devuelve  $\frac{3 \sin x - \sin(3x)}{4}$
- `trigsimp(a)` simplifica la expresión trigonométrica  $a$  usando la relación  $\cos^2 t + \sin^2 t = 1$  y sustituyendo  $\tan t$  por  $\frac{\sin t}{\cos t}$
- `load(ntrig)` carga el paquete, lo que permite obtener los valores exactos de  $\sin x$ ,  $\cos x$  y  $\tan x$  cuando  $x$  es un múltiplo de  $\pi/10$

## 5 Aritmética entera

Sean  $a$  y  $b$  dos enteros. Sean  $n$  y  $p$  dos números naturales.

- `divide(a,b)` es la división euclídea de  $a$  por  $b$ . El resultado es una lista cuyo primer elemento es el cociente y el segundo el resto
- `divisors(a)` es el conjunto de los divisores positivos de  $a$
- `divsum(a)` es la suma de los divisores positivos de  $a$
- `mod(a,b)` es el resto de la división euclídea de  $a$  por  $b$
- `gcd(a,b)` es el máximo común divisor de  $a$  y  $b$
- `load(funcs) $ lcm(a,b)` es el mínimo común múltiplo de  $a$  y  $b$
- `primep(p)` se verifica si  $p$  es primo
- `prev_prime(n)` es el mayor primo menor que  $n$
- `next_prime(n)` es el menor primo mayor que  $n$
- `factor(n)` es la descomposición de  $n$  en producto de factores primos
- `ifactors(n)` es la lista de la descomposición de  $n$  en producto de factores primos
- `binomial(n,p)` es el coeficiente binomial  $\binom{n}{p}$
- `random(n)` es un entero natural, elegido al azar entre 0 y  $n - 1$  cuando  $n \in \mathbb{N}^*$

## 6 Números complejos

Sea  $z$  un número complejo.

- `realpart(z)` es la parte real de  $z$
- `imagpart(z)` es la parte imaginaria de  $z$
- `conjugate(z)` es el conjugado de  $z$
- `abs(z)` es el módulo de  $z$
- `carg(z)` es el argumento de  $z$  (en  $]-\pi, \pi]$ )
- `rectform(z)` es la forma cartesiana de  $z$
- `polarform(z)` es la forma polar de  $z$

## 7 Cálculo algebraico

Sean  $P$  y  $Q$  dos polinomios.

- $R: x^4 - 4x^2 - 5$  define el polinomio  $R = X^4 - 4X^2 - 5$
- `expand(P)` desarrolla  $P$
- `factor(P)` factoriza  $P$
- `gfactor(P)` factoriza  $P$  en  $\mathbb{C}$
- `solve(P, x)` raíces complejas de  $P$
- `subst(4, x, P)` valor de  $P$  cuando  $X = 4$
- `ratcoef(P, x^3)` es el coeficiente del término en  $X^3$  de  $P$
- `divide(P, Q, x)` es la lista formada por el cociente y el resto de la división de  $P$  por  $Q$ .
- `portfrac(P/Q, x)` descompone la función racional  $P/Q$  en elementos simples
- `ratsimp(expr)` simplifica la expresión racional `expr` (reduce a común denominador)
- `subst(1/z, x, expr)` sustituye  $x$  por  $1/z$  en la expresión `expr`

## 8 Funciones numéricas

### 8.1 Definición de funciones

- $f(x) := x^2 + 2x - 3$
- `define(f(x), x^2 + 2*x - 3)`
- `f:lambda([x], x^2 + 2*x - 3)`

### 8.2 Límites, tangentes y asíntotas

- `limit(sin(x)/x, x, 0)` límite en 0
- `limit(1/x, x, 0, plus)` límite en 0 por la derecha
- `limit(1/x, x, 0, minus)` límite en 0 por la izquierda
- `limit(1/x, x, inf)` límite en  $+\infty$
- `limit(x*exp(x), x, minf)` límite en  $-\infty$
- `taylor(f(x), x, a, n)` desarrollo de Taylor de  $f$  respecto de  $x$  en el punto  $a$  con  $n$  términos

### 8.3 Derivación

- `diff(f(x), x)` es la derivada  $f'(x)$
- `diff(f(x), x, 2)` es la derivada segunda  $f''(x)$
- `define(g(x), diff(f(x), x))` define  $g(x)$  como  $f'(x)$
- `taylor(f(x), x, a, 1)` es la ecuación reducida de la tangente a  $f$  en el punto  $(a, f(a))$

### 8.4 Representación de funciones

- `plot2d([f(x), g(x)], [x, x1, x2], [y, y1, y2])` representa las funciones  $f$  y  $g$  en la región  $[x_1, x_2] \times [y_1, y_2]$

## 8.5 Integrales

- `integrate(f(x), x)` es una primitiva de la función  $f$
- `integrate(f(x), x, a, b)` es la integral  $\int_a^b f(x) dx$
- `romberg(f(x), x, a, b)` es una aproximación de la integral  $\int_b^a f(x) dx$

## 9 Ecuaciones

### 9.1 Resolución de ecuaciones

- `solve(x^3-2x^2+x=2, x)` es la resolución exacta en  $\mathbb{C}$
- `find_root(x^5=1+x, x, 1, 2)` solución aproximada en  $[1, 2]$
- `allroots(x^5=1+x)` valores aproximados de todas las soluciones (reales y complejas)

### 9.2 Sistemas lineales

Para resolver el sistema 
$$\begin{cases} 3x + 2y = 1 \\ x - y = 2 \end{cases}$$

- `S1: [3*x+2*y=1, x-y=2]`
- `solve(S1, [x, y])`

## 10 Listas

- `L: makelist(k^2, k, 0, 9)` asigna a  $L$  la lista con los cuadrados de los 10 primeros naturales
- `L[2]: 5` sustituye el segundo elemento de la lista  $L$  por 5
- `length(L)` longitud de la lista  $L$ .
- `first(L)` primer elemento de  $L$
- `second(L)` segundo elemento de  $L$
- `last(L)` último elemento de  $L$
- `member(x, L)` se verifica si  $x$  pertenece a la lista  $L$
- `append([a, 1, 3], [2, 7])` concatena las listas
- `join(l, m)` construye una lista, intercalando los elementos de  $l$  y  $m$ . Se obtiene la lista  $[l[1], m[1], l[2], m[2], l[3], m[3], \dots]$ .
- `sort(L)` ordenación de los elementos de  $L$  en orden creciente.
- `map(f, L)` aplicación de la función  $f$  a todos los elementos de  $L$

## 11 Sumas y productos

### 11.1 Sumas finitas

- `sum(1/k^2, k, 1, 10)` es la suma de los inversos de los cuadrados de los enteros entre 1 y 10.

### 11.2 Productos finitos

- `product(sqrt(k), k, 1, 10)` es el producto de las raíces cuadradas de los enteros entre 1 y 10.

### 11.3 Sumas infinitas

- `load(simplify_sum) $ sum(1/k^2,k,1,inf) $ simplify_sum(%)`  
calcula la suma de la serie  $\sum_{k=1}^n \frac{1}{k^2}$

## 12 Programación

### 12.1 Sintaxis de un programa

- Sintaxis general

```
nombre(parámetros de entrada) := block([variables locales],
  <instrucción 1>,
  ....
  <instrucción n>,
  /* -----Comentario----- */)

```

- Ejemplo simple de programa que suma dos números  
`suma(a,b) := block([c], c:a+b, return(c))`

### 12.2 Estructura condicional

- Sintaxis

```
if (condición)
  then (<instrucción1> , <instrucción2>)
  else (<instrucción3> , <instrucción4>)

```

### 12.3 Estructuras iterativas

- Bucle **For** para escribir la tabla del 7:  
`for k from 1 thru 10 do (print("7 veces",k,"igual a",7*k))`
- Bucle **While** para escribir la tabla del 7:  
`k:1 $ while k<11 do (print("7 veces",k,"igual a",7*k), k:k+1)`

## 13 Matrices

### 13.1 Construcción de matrices

- Por extensión:

```
A:matrix([1,2,3],[4,5,6],[7,8,-9])
define la matriz  $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & -9 \end{pmatrix}$ 

```

- Mediante una función:

```
M:genmatrix(lambda([i,j],i+j),3,2)
define la matriz  $M \in \mathfrak{M}_{3,2}(\mathbb{R})$  cuyos elementos son de la forma  $m_{i,j} = i + j$ .
```

## 13.2 Matrices particulares

- `zeromatrix(5,3)` es la matriz nula de orden  $5 \times 3$
- `ident(5)` es la matriz identidad  $I_5$
- `diag_matrix(a,b,c)` es la matriz diagonal  $\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix}$

## 13.3 Operaciones con matrices

- $A+B$  es la suma de las matrices  $A$  y  $B$
- $3*A$  es el producto de la matriz  $A$  por 3
- $A.B$  es el producto de las matrices  $A$  y  $B$
- $A^{\wedge}3$  es la matriz  $A$  elevada al cubo
- `invert(A)` es la inversa de la matriz  $A$
- `transpose(A)` es la traspuesta de la matriz  $A$
- `determinant(A)` es el determinante de  $A$