



MAXICAJA DE HERRAMIENTAS

Operadores /Funciones/ comandos/Paqueterias	¿Cuál es su sintaxis?	¿Para qué sirve?	Ejemplo
Relacionales	< (menor que) > (mayor que) = (igual a) <= (menor o igual que) >= (mayor o igual que) # (diferente a)	Se usan para comparar dos valores. Si el resultado de la comparación es correcto la expresión considerada es verdadera, en caso contrario es falsa.	(%i1) a:7; b:5; (%i2) is(a>b); (%o2) true (%i3) is(a<b); (%o3) false (%i4) is(a#b); (%o4) true
Operadores Lógicos	and (y) or (o) not (no)	Se utilizan para evaluar expresiones simultáneas. Arrojan verdadero o falso, según sea el caso.	(%i1) a:2; b:4; c:6; (%i2) a<b and a<c; (%o2) true (%i3) a>b or a<c; (%o3) true (%i4) a>b or a>c; (%o4) false (%i5) a>b and a>c; (%o5) false
Funciones para desplegar resultados	print(<exp1>, ..., <expn>)	Permite mostrar varios resultados en una misma línea.	(%i1) v:[1,1,1]\$ (%i2) print("La longitud del vector v es:", sqrt(v.v)); (%o2) La longitud del vector v es: $\sqrt{3}$

Comandos para la solución de ecuaciones algebraicas	linsolve ([expr_1, ..., expr_m], [x_1, ..., x_n])	Resuelve la lista de ecuaciones lineales simultáneas para la lista de variables. Las expresiones deben ser polinomios lineales respecto de las variables o ecuaciones.	<pre>(%i1) e1: x + z = y; (%o1) z + x = y (%i2) e2: 2*a*x - y = 2*a^2; (%o2) 2 a x - y = 2 a (%i3) e3: y - 2*z = 2; (%i7) linsolve ([e1, e2, e3], [x,y, z]); (%t7) z = a - 1 (%t8) y = 2 a (%t9) x = a + 1</pre>
	funcsolve (eqn, g(t))	Devuelve [g(t) = ...] o [], dependiendo de que exista o no una función racional g(t) que satisfaga eqn, la cual debe ser un polinomio de primer orden, lineal para g(t) y g(t+1). Aviso: esta es una implementación rudimentaria, por lo que debe ser utilizada con cautela	<pre>(%i1) eqn:(n + 1)*f(n) - (n + 3)*f(n + 1)/(n + 1) = (n - 1)/(n + 2); (%o1) (n + 3)f(n + 1)/(n + 1) = (n - 1)/(n + 2) (%i2) funcsolve (eqn, f(n)); (%o2) f(n) = (n + 1)/(n + 2)</pre>

Funciones para ecuaciones	<code>lhs(<expr>)</code> (left hand side) <code>rhs(<expr>)</code> (right hand side)	<code>lhs</code> y <code>rhs</code> devuelven el primer y el segundo miembro, respectivamente, de una ecuación o desigualdad.	<pre>(%i1) a:[-2,6,1]; b:[1,1,0]; c:[1/3,13/3,1/2]; expr:r*a+s*b-c; solns:solve(expr,[r,s]); (%i2) val:solns[1]; val[1]; (%o2) r=1/2 (%i3) lhs(val[1]); (%o3) r (%i4) rhs(val[1]); (%o4) 1/2</pre>
Comandos para funciones compuestas	<code>return(<resultado>)</code>	Sirve para interrumpir la ejecución de una función compuesta creada con <code>block</code> y devolver un resultado específico. Recuerda que <code>block</code> sirve para crear funciones que requieren más de una línea de código).	<pre>(%i1) if congruence1 # congruence2 and congruence2 # congruence3 then return ("Los triangulos no son congruentes");</pre>

Paquetería eigen	load (eigen)	<p>Contiene funciones para el cálculo simbólico de valores y vectores propios. Las funciones del paquete eigen son innerproduct, unitvector, columnvector, gramschmidt, eigenvalues, eigenvectors, uniteigenvectors similarity transform</p>	<pre>(%i1) a1:[1,3]; b1:[4,6]; (%o1) [1,3] (%o2) [4,6] (%i2) load (eigen)\$ ua1:unitvector(a1); ua2:unitvector(b1); (%o3) [1/sqrt(10),3/sqrt(10)] (%o4) [2/sqrt(13),3/sqrt(13)]</pre>
<p>Sentencias condicionales: if then, else if-then, else.</p>	<pre>if <condición1 > then <acción1 > else if <condicion2> then <acción2 > else <acción3 ></pre>	<p>Si la <condicion1 > es verdadera, entonces se ejecuta la <accion1 >, si es falsa entonces se evalúa, si es verdadera la <condicion2>, en cuyo caso se ejecuta la <accion2> y así sucesivamente dependiendo el número de condiciones que se le añada</p>	<pre>(%1i) if congruence1 # congruence2 and congruence2 # congruence3 then return ("Los triangulos no son congruentes") else if s1=s2 and s2=s3 then print ("Los triangulos son congruentes, existe una transformacion de semejanza donde el escalar que se encarga de la tranformaion T(P)= {P, aP=Q } es",s1) and graficadora(P1,P2,P3,Q1,Q2,Q3)</pre>

Paquetería draw	Load(draw)\$	Permite cargar los comando draw2d y draw3d con el mismo propósito que los comandos plot2d y plot3d. Sólo funcionan de forma completa con Gnuplot	(%i1)load(draw)\$ draw2d(color=blue, explicit(cos(x),x,0,%pi/2), color=red, explicit(x,x,0,1));
Draw2d	Draw2d(opciones)	draw2d cuenta con variantes que permiten dibujar gráficas de funciones dadas de forma explícita (explicit), paramétrica (parametric), implícita (implicit) y polares (polar). También permite dibujar puntos aislados y funciones lineales a trozos (points, polygom), gráficos de barras (bars), rectángulos (rectangle), vectores (vector), segmentos y arcos de elipse y circunferencias.	(%i1)draw2d(color=blue, parametric(cos(t),sin(t),t,0,2*%pi), color=green, implicit(x^2+y^2=9,x,-3,3,y, 3,3), color=black, polar(2,t,0,2*%pi), color="#ff0000", polar(1+t*0.3,t,0,3*%pi));

Mayor consultas
(link)

<http://webs.um.es/mira/maxima/manualico.php>