

	Beschreibung	Maple, Waterloo Maple (Version V, Release 4, 1996)	Mathematica, Wolfram Research (Version 3, 1996)	Matlab, MathWorks (Version 4, 1994)
Allgemein	Funktion (Ausführen)	phi2:=(x,y)->cos(x)^2+exp(y); [ENTER] (Semikolon)	f[x_,y_]:=Cos[x]^2+E^y [SHIFT-ENTER] (kein Zeichen)	g='cos(x)^2+exp(y)' [ENTER] (Kein Zeichen)
	Ausdruck (Umbruch)	g:=sin(x)/x; [SHIFT-ENTER] (Doppelpunkt: ohne Output)	g:=sin(x)/x; [ENTER] (Semikolon: ohne Output)	;... [ENTER] (Semikolon: oO./3 Punkte: Umbruch)
	Pakete, Neustart & Hilfe	with(linalg), ?packages, ?restart	<<Algebra`SymbolicSum`, ?Clear (auch: ??Clear)	help clear (auch: lookfor clear)
	Approx. & letztes Resultat	Digits:=100: evalf("");	N[%,100], %% vorletztes usw; %15 Output Nr. 15	vpa(A,100) (Logdatei erstellen: diary [File])
	Schreibweisen	Mathematisch klein, Symbolisch gross Zuweisungen :=; Gleichungen =	Alle gross; Zuweisungen := und =; Gleichungen == Verzweigungen für Funktionen mit /; [Condition]	Alle klein; Zuweisungen und Gleichungen = Symbolische Ausdrücke in Hochkomma z.B. 'cos(x)'
Funktionen	Arithmetik	+,-,*,/,^,!, I, Pi, infinity, cos(), sin(), tan(), log(), sqrt()	dito, allerdings mit Grossbuchstaben	/ und \; log() und log10();Präfix "sym", zB. symmul('...')
	Vereinfachen	expand(), factor(), simplify(), assign()	Expand[f(x),Trig->True]; Normal[Series[]] (n-Jet)	simple(), subs()
	Grafik	plot(f,a..b), plot3d(f,a..b,c..d) oder plot(g,x=a..b), plot3d(g,x=a..b,y=c..d)	Plot[Sqrt[1-x^2], {x, 0, 1}, AspectRatio -> Automatic], ParametricPlot[{t Cos[t],t Sin[t]},{t,0,4 Pi}], Animate[]	plot(x,y), plot3(x,y,z), mesh(x) bzw. ezplot('sin(x)')
	Analysis	diff(f(x),x), int(f(x),x=a..b), sum(f(k),k=a..b), series(f(x),x,ord), taylor(f(x),x=0,n);	D[f[x],x], Integrate[f[x],{x,a,b}], SymbolicSum[f[k], {k,0,n,s}], Sum[f[k],{k,a,b}], Series[], Apart[] (PBZ)	diff(g,'x'), int(g,'x')
	Lineare Algebra	det(), inverse(), crossprod()/dotprod() oder &* für R3	Det[], Inverse[], Cross[]/Dot[] für R3	determ(), transpose() oder ', inverse(), symmul()
Glg.	Symbolisch/Numerisch	solve(e, x), solve({e1,...,en},{x1,...,xn}) bzw. fsolve()	Solve[] bzw. NSolve[] (vgl. Algebra`SymbolicSum`)	x=solve('exp(x)=tan(x)') (löst ggf. Numerisch)
	Differentialgleichungen	dsolve({diff(y(t),t\$3)=0,y(0)=0},y(t)); (rekursiv: rsolve)	DSolve[y'''[t]==0, y[t], t]; y[x_]=y[x] /. %[[1]] liefert y(x)	y=dsolve(D2y=cos(2*x)-y', 'Dy(0)00', 'y(0)=1')
Typen	Datentypen	Sequenz S:=a1,...,an; S[i]; S:=seq(f(i),i=a..b)	Liste L={a1,...,an}; L[[i]]; L=Table[f[k],{k,a}], 1<=k<=a	Matrix M=sym('[a,b;c,d]')/sym(3,3,'m','n','m*n') (symbol.)
		Menge M:={a1,...,an}; M union {a1,...,am};	Array[a,7]; Do[a[k]=f[k],{k,a}]	M(i); M(i,j); ; M(i:s) (Spalten i bis j, Schrittweite s)
		Liste L:=[a1,...,an]; L[a..b];	Matrix A={{1,2},{3,4}} (MatrixForm[] für Darstellung)	Array A=(0:0.1:1)*pi (heisst A=i/10*Pi für 0<=i<=1)
		Matrix A:=linalg[matrix]([[1,2],[3,4]]); ebenso: vector	Vektor v={a1,...,an} (dito)	Zusammenfügen von Arrays A=[B C]
Beispiele	Lineare Algebra	evalm(A*A); [Matrix ausrechnen] norm(A,n); [n-Norm] charmat(A,lambda) [Charakteristisches Polynom]	Eigenvalues[A] Eigenvector[A] [Eigenwerte & -vektoren] In Maple: eigenvals() & eigenvects()	singvals(), magic(), rand(), eye() LGS mit x=A\b (verwendet ggf. Meth. der kl. Quadrate)
	Programmierung	# Lösung nichtlinearer Gleichungssysteme (Gauss) leastsqr := proc (f,x) # option remember für rek. Funkt. local phi,g; phi:=sum(f[i]^2,i=1..nops(f)); # nops liefert Grösse g:=grad(phi, convert(g,set), x); end;	/* Kurve in Parameterdarstellung */ ParametricPlot[{t, Cos[t], Sin[t]}, {t, 0, 4 Pi}, AspectRactio -> Automatic] /* Matrix darstellen */ MatrixForm[{{a, b}, {c, d}}]	% Integration einer Differentialgleichung (Euler implizit) function yneu = eulimp (x,y,k) z = (1-k*(x+k))*z; yneu = (y+sqrt(y*y+2*h*z))/z; Aufruf: for i = 1:100 y = eulimp ((i-1)/100,y,0.01); end;