

MATLAB UND MUPAD

Variablen	Vektoren	Matrizen	Indexe	Text	Funktionen
a, b, c	x, y, z	A, B, C	i = Zeile j = Spalte	..	f, g, h

- Filenamen nie mit Umlaute oder Leerzeichen, sondern mit Kleinbuchstaben

	MATLAB				MuPAD			
Interface								
Hilfe, Ausgabe unterdrücken, Kommentar	help ;	% ..	? ..		/* .. /* .. */		
Löschen von Workspace, History, Window, Variable, Fenster	clear all clear ..	clc close all	clear	delete() delete a				
Workspace auflisten	who whos							
.mar Datei laden – Workspace	load Dateiname							
.mat Datei speichern - Workspace	save Dateiname							
Operationen								
Grundoperationen, Potenz, Wurzel	+ - * /	^	sqrt(a) a^(1/2)	+ - * /	^	sqrt(a) a^1/2		
elementweise Multipl / Division / Potenz	.*	./	.^	map(a, operation)				
Betrag, nur Vorzeichen (-1,0-1)	abs(a)		sign(a)	abs(a)		sign(a)		
nat. Log, 10-Log, 2-Log (x zu Basis b)	log(a)	log10(a)	log2(a)	lob(b, x)				
Exponent, Fakultät	3*10^7 3e7	factorial(..)	exp(a)	7.3e7	!	exp(a)		
Funktionen								
exakt rechnen, vereinfachen	format long format short	simplify(..) Simplify(..)		float(..) DIGITS=..	simplify(..) Simplify(..)			
Faktorisieren, Normalform	expand(..)			expand(..)	normal(..)			
sortieren, Minimum, Maximum	sort(A)	min(A)	max(A)	sort(a)	min(A)	max(A)		
Runden, Summe, Durchschnitt	round(a)	sum(A)	mean(A)	round(a)	sum(A)	stats:: mean(A)		
gegen Null/unten/oben runden	fix(a)	floor(a)	ceil(a)	trunc(a)	flood(a)	ceil(a)		
Standardabweichung s	std(x)							
Trigonometrische Funktionen								
sinus, cosinus, tangens, cotangens [°]	sin(a)	cos(a)	tan(a)	cot(a) 1/tan(a)	sin(a)	cos(x)	tan(x)	cot(a)
arcsinus, arccosinus, arctangens [°] Winkel im Koordinatensystem	asin(a)	acos(a)	atan(a) atan2(y, x)		arcsin(a)	arccos(a)	arctan(a)	
Gradmas in Bogenmass	cos(a/180*pi)			cos(a/180*pi)				
Zufallsgenerator								
Zufallsgenerator zurücksetzen	randn('seed', 5);							
Zufallszahl (0..1)	rand(i, j)			random()				
Zufallszahl standardnormalverteilt (-1..+1)	randn(i, j)							
Zufallszahl exponentialverteilt	exprnd(i, j)							

Variablen							
Unendlich, Pi, undefiniert, e	Inf	Pi	NaN	e(1)	infinity	PI	undefined
Definieren, abrufen	a= ..		a		a:= ..		a
Definitionsbereich festlegen	assume(x<0)						
Vektoren							
Zeilenvektor, Spaltenvektor definieren	x=[1 2 3 4 5] x=[1,2,3,4,5]		y=[1;2;3;4;5]		a:=[3,2]		-
Zahl in Vektor setzen, löschen, abrufen	x(6) = 37	x(5) = []	x(5)				
Abrufen von Element							
Auswahl von einem Vektor	a(6:2:10)						
Länge, Betrag eines Vektors	length(x)		norm(a)				
Gleichheit, Addition, Subtraktion	isequal(x,y)	x+y	x-y		matrix::equal(x,y)		x+y x-y
Multiplikation mit Skalar, mit Vektor	2*x		x.*x x'*x		2*x		
Zusammenführen (Zeilen- Spaltenvektor)	z = [x1 x2] z = [x1,x2]		z = [y1;y2]				
Transponieren	x' y'		linalg::transpose(A)				
Skalarprodukt	x*x' sum(x.*x)		dot(x,x)		linalg::scalarProduct(a,b)		
Kreuzprodukt, Spatprodukt/Determinante	cross(a,b)		det([a;b;c])		linalg::crossProduct(a,b)		det(A)
Matrizen - Arrays							
	Operationen werden Spaltenweise ausgeführt						
Definieren	A=[2 5 7;3 6 1]				A:=matrix([[2,5,7],[3,6,1]])		
Abrufen von Element, Zeile, Spalte	A(i,j)	A(2,:)	A(:,3)		A[2,3]	A[2,1..3]	A[1..2,3]
Auswahl von einem Array (:= Alle Elemente)	A(i,j)	A(20:2:40,10:2:20)		A(:,1:3:end) A(10:20,15:20)			
Grösse abfragen	[i,j] = size(A)				linalg::matdim(A)		
Nullmatrix, Einheitsmatrix	zeros(i,j)		eye(3)		matrix(i,j) matrix::identity(5)		
Zahl in Matrix setzen, löschen	A(2,4) = 35		A(2,3) = []				
Gleichheit, Addition, Multiplikation	isequal(A,B)	A+B	2*A		matrix::equal(A,B)		A+B 2*A
Transponieren, Matrizenmultipl.	A'		A*B		linalg::transpose(A)		A*B
Inverse Matrix	inv(A)				A^-1 inverse(A)		
Spatprodukt / Determinante	det(A)				det(A)		
Zusammenführen	C = [A;b;B]						
Betrag	norm(a)						
Auf Stufenform bringen	rref(A)						
LGLs lösen (nur eine Lösung), Rang	A\b		rank(A)				
Maximum jeder Spalte, eines Arrays	max(A)		max(max(A))				
Summe der ganzen Matrix	sum(sum(A))						
n Indexe eines suchenden Elementes	[y,x] = find(A>10) [y,x] = find(A>10,n)						
elementweise divid./multipl./potenzieren	.*		./		.^		
Diagonale einer Matrix	diag(A)						

Folgen / Schleifen		
Werte von -5 bis 5 im Abstand von 0.01 Default = 100 Werte 20 Werte	<pre>x=(-5:0.01:5) x = linspace(-5,5) x = linspace(-5,5,20)</pre>	<pre>a=[i \$ i=-5..5 step 0.01] a:=[i^2 \$ i=1..9] a:=[i^3 \$ i=-9..9 step 3]</pre>
Werte von 10 bis 20 Schritt 1	<pre>x=(10:20)</pre>	
for-Schleife	<pre>for i = 2:n y(i) = y(i-1)+dt*(y(i-1)) end</pre>	
switch	<pre>switch a case 1 ... case 2 ... end</pre>	
Funktionen		
Funktion y(x) definieren	<pre>y=x.^2</pre>	<pre>f:=x-->x*x f := x*x</pre>
Funktionswert abfragen		<pre>f(-2) f(a)</pre>
solve	<pre>solve('...')</pre>	<pre>solve(3*x+5=4) solve(3*x=4,x) solve({3*x+5y=4,5*x=2*y})</pre>
Nullstellen		<pre>solve(f(x)=0,x)</pre>
Verhalten im Unendlichen		<pre>limit(f(x),x=-infinity)</pre>
Maxima, Minima		<pre>solve(f'(x)=0,x)</pre>
ntes-Polynom aus Stützstellen berechnen	<pre>polyfit(x,y,n)</pre>	
Wert eines Polynoms (p=Koeffizienten)	<pre>polyval(p,x)</pre>	
Funktion mit eingesetzten Werten		<pre>subs(f(x,y),x=0) f(x,y) x=0 y=2</pre>
Stückweise definierte Funktion		<pre>piecewise([x>0,f1],[x<0,f2])</pre>
Differenzieren und Integrieren		
Ableitung	<pre>diff(f)/delta_x;</pre>	<pre>f'(x) f''(x)</pre>
Gradient	<pre>[fx,fy]=gradient(f)</pre>	
Partielle Ableitung		<pre>diff(f(x,y), x) diff(f(x,y), x, y)</pre>
Komplexe Zahlen		
Imaginär Einheit	<pre>Ausgabe immer im kartesischen System j</pre>	
Realteil, Imaginärteil, real?	<pre>real(z) imag(z) isreal(z)</pre>	
kartesische, exponentielle Eingabe	<pre>z=3-7j z=sqrt(2)*exp(j*pi/4)</pre>	
trigonometrische Eingabe	<pre>z=sqrt(2)*(cos(pi)+j*sin(pi))</pre>	
Betrag, Winkel im Bodenmas	<pre>abs(z) angle(z)</pre>	
komplex konjugiert	<pre>conj(z)</pre>	
transponieren mit / ohne konjugieren	<pre>z' z.'</pre>	
komplexer Zeiger anzeigen	<pre>compass(..)</pre>	
Fourier		
fast fourier analyse Diskrete Fourier-Transformation (DFT)	<pre>fft(f)</pre>	
inverse fast fourier analyse (IDFT)	<pre>ifft(..)</pre>	

Plot - Arten		
3D-Plot		plotfunc3d(f) plot(f,#3D) plotfunc3d(f1,f2)
2D-Graph	plot(x,y)	plotfunc2d(f)
Histogramm (mit n Klassen)	hist(x,n) bar(x,y)	
Höhenlinien		plot(plot::Implicit2d(f,x=-5..5,y=-5..5))
Plot - Attributen		
Gitternetz ein / aus	grid on grid off	GridVisible=TRUE =FALSE
gleicher Achsenabstand	axis equal	,Scaling=Constrained,
Achsenbeschriftung	xlabel('\...') ylabel('\...')	AxesTitles=["x","y"]
Titel	title('\...')	
in gleichen Plot zeichnen ein/aus	hold on hold off	
logarithmische Achsen (beide,x,y)	loglog() semilogx() semilogy()	
Koordinatensystem		,Axes = Origin,
mit Anzahl Messpunkt		,Mesh = [50, 50],
Symbole: Punkte, Kreise, ...	, 'o', , '*' , , 's',	LineStyle=Dashed LineWidth=1
Linienart: fest, gepunktet, ...	, '-', , ':', , '-.', , '--',	
Bildausschnitt	axis([xmin xmax ymin ymax])	,x=-5..5,y=-5..5, ,Zrange = 0..10,
rot, blau, grün, cyan, schwarz	, 'r', , 'b', , 'g', , 'c', , 'k',	LineColor=RGB::Red
Grösse ändern	, 'MarkerSize',6,	
Farbe ändern	, 'MarkerFaceColor','y',	,FillColorType = Functional,
Text in plot anzeigen	text(x,y,'Text')	
Farbe ändern	, 'Color','r'	
Mehrere Diagramme (n = aktuell)	subplot(i,j,n);	
		fsolve
Parameterdarstellung einer Kurve		

Spaltenbreiten 6.34 cm + 6.34 cm + 5.98 cm = 18.66 cm

Eigene MATLAB Funktionen			
normiertes Histogramm (A=1)	pdfplot(x,n)	x = Daten	n=Klassen
Funktionswert an der Stelle x der Gauss-Normalverteilung	normpdf(x,μ,σ)	x= Stelle	μ=Erwartungswert σ=Standardabweichung
Wahrscheinlichkeit P: von -∞ bis zu Grenze	normcdf(grenze,m,s)	grenze= bis wohin	m=Mittelwert s=Standardabweichung

Programmieren von Methoden		Name der Funktion muss mit dem Namen der Datei übereinstimmen	
Eingabe / Ausgabe	a = input('hoehe=') output('text')		
Text ausgeben	disp(['die breite ist' a])		
Zahl in String umwandeln	num2str(variable)		
Unterbrechung (Taste, Sekunden)	pause pause(t)		
Komentare	%		
Anwendung mit mehreren Parameter	[i,j] = funktion(...)		