



# 1 Systemspezifische Befehle

## Befehle

<code>clc</code>	Commandfenster löschen
<code>clear</code>	Alle Variabelnzuweisungen löschen
<code>help</code>	Hilfe zu einer Funktion
<code>load</code>	Variabelnzuweisung wieder laden
<code>lookfor</code>	Befehle nach einem Suchbegriff durchsuchen
<code>save</code>	Alle Variabelnzuweisungen speichern
<code>what</code>	Dateien im aktuellen Verzeichnis anzeigen
<code>whos</code>	Verwendete Variablen anzeigen
<code>;</code>	Befehl ausführen ohne anzuzeigen
<code>simplify( )</code>	Ausdruck vereinfachen
<code>pretty( )</code>	Ausdruck in lesbarer Form anzeigen
<code>grid on</code>	Gitterlinien in einem Diagramm anzeigen

## Variablen

<code>ans</code>	Letztes Resultat
<code>pi</code>	$\pi$
<code>i, j</code>	Imaginäre Einheit

## Operatoren

<code>*</code>	Matrix Multiplikation
<code>.*</code>	Array Multiplikation
<code>./</code>	Array Division
<code>^</code>	Matrix Exponent
<code>.^</code>	Array Exponente
<code>'</code>	Transponiert (mit konjugiert komplexen Zahlen)
<code>.'</code>	Transponiert (ohne Veränderung der komplexen Zahlen)
<code>\</code>	Lösung eines linearen Gleichungssystems

## Diverse

<code>rand(x)</code>	Zufallszahlen zwischen 0 und 1 gleichverteilt
<code>randn(x)</code>	Zufallszahlen um 0 mit Gaussverteilung

# 2 Funktionen

## 2.1 Allgemeine Mathematische Funktionen

Symbol	Matlab	Beschreibung
$\ln(x)dx$	<code>diff(log(x),x)</code>	
$ x $	<code>abs(x)</code>	Betrag
	<code>angle(x)</code>	Argument, Phase
	<code>ceil(x)</code>	Runden (größer oder gleich) zur nächsten Zahl
	<code>floor(x)</code>	Runden (kleiner oder gleich) zur nächsten Zahl
	<code>round(x)</code>	Rundung zur nächsten ganzen Zahl
	<code>conj(x)</code>	Konjugiert komplexe Zahl
$e^x$	<code>exp(x)</code>	Exponentialfunktion
$\ln(x)$	<code>log(x)</code>	Natürlicher Logarithmus
$\log(x)$	<code>log10(x)</code>	Zehnerlogarithmus
	<code>imag(x)</code>	Imaginärteil einer Zahl
	<code>real(x)</code>	Realteil einer Zahl
$\frac{x}{y}$	<code>rem(x,y)</code>	Ganzzahliger Rest von
$\sqrt{x}$	<code>sqrt(x)</code>	Quadratwurzel
$x!$	<code>factorial(x)</code>	Fakultät
$\binom{n}{k}$	<code>nchoosek(n,k)</code>	Binomialkoeffizient
$X_0 = \mu$	<code>mean(Array)</code>	Linearer Mittelwert
	<code>var(Array)</code>	Varianz
$\sigma$	<code>std(Array)</code>	Standardabweichung
$X^2$	<code>mean(Array.^2)</code>	Quadratischer Mittelwert
	<code>[C, D] = xcorr(A, B)</code>	Auto- bzw. Kreuzkorelation
$\frac{2}{\sqrt{\pi}} \cdot \int_0^x e^{-t^2} dt$	<code>erf(x)</code>	Errorfunktion
$\frac{2}{\sqrt{\pi}} \cdot \int_x^\infty e^{-t^2} dt = 1 - erf(x)$	<code>erfc(x)</code>	Komplementäre Errorfunktion

## 2.2 Funktionen im Zeibereich

Name	Symbol	Matlab	Beschreibung
Einschaltfunktion	$u(t)$	heaviside(t)	
Signumfunktion	$sgn(t)$	sign(t)	
Rampenfunktion	$r(t)$	ramp(t)	$r(t) - t \cdot u(t)$
Rechteckimpuls	$p_a(t)$		$u(t+a) - u(t-a)$
Dreieckimpuls	$\Lambda_a(t)$		
Sincfunktion	$sinc_a(t)$	sinc(t)	$\frac{\sin(\pi \cdot t)}{\pi t}$
Impulsfunktion	$\delta(t)$	dirac(t)	Nur Symbolisch

## 3 Arrays

### 3.1 Erstellung von Arrays

linspace(a,b,c)	Erstellt einen Array von $a$ bis $b$ mit $c$ (Anzahl) Werten. $c$ kann optional auch weggelassen werden, als Standardwert wird dann 100 verwendet.
logspace(a,b,c)	Erstellt einen Array von $10^a$ bis $10^b$ . Für $c$ gilt dasselbe wie bei linspace.
rand(y,x)	Erstellt einen Array mit $y$ Spalten und $x$ Zeilen mit Zufallszahlen zwischen 0 und 1.

### 3.2 Rechnen mit Arrays

## 4 Plotten

hist(Array, [min, step, max])	Erstellt eine Histogramm aus den Werten eines Arrays.
plot(x,y)	Erstellt einen normalen x-y Plot
stem(f)	Erstellt einen Plot bei dem die Werte als horizontale Pfeile dargestellt werden

## 5 Programmier-Syntaxe

### 5.1 Allgemeine Konstrukte

```

if Bedingung1
:
elseif Bedingung2
:
else
:
end

while Bedingung
:
end

for i=Startwert: Schritt: Endwert
:
:
end

```

### 5.2 Zusammenhängende Beispiele

```

function Rxx=Autokor(x)
% Rxx=Autokor(x) berechnet die diskrete zyklische
% Autokorrelationsfunktion des Vektors x.
% Rxx ist eine zweizeilige Matrix:
% Spaltenzahl=Laenge von x
% 1. Zeile: m-Achse (Nummern der Werte)
% 2. Zeile: Autokorrelationsfolge
xx=x.^2; % Wert Nr.0 (keine Verschiebung)
Rx=sum(xx);
N=length(x);
x1=x; % Vektor der verschobenen Werte
for t=1:N-1

```

```

x1=[x1(1,N),x1(1,1:N-1)]; % zykl.Verschiebung
xx1=x.*x1;
R=sum(xx1);
Rx=[Rx,R]; % Konstruieren des Vektors
end
m=0:N-1;
Rxx=[m;Rx];

```

## 6 Beispiele

### 6.1 Lineares Gleichungssystem

Ein Gleichungssystem der Form  $Ax = b$  wird folgendermassen gelöst:

```

x = A \ b
x + y = 62      x = 46
x - 4y = -18    y = 16      x = [1 1; 1 -4] \ [62; -18]    x = [46; 16]

```

### 6.2 Polynomdivision

Ein und Ausgegeben wird jeweils nur ein Array mit den Koeffizienten des Polynoms

$$\frac{x^4 - 3x^3 + 3x^2 - x}{x - 1} = x^3 - 2x^2 + x$$

```

R = deconv([1 -3 3 -1 0],[1 -1])    R = [1 -2 1 0]

```

### 6.3 Laplace

$$f(t) = -1.25 + 3.5te^{-2t} + 1.25e^{-2t} \rightarrow F(s) = \frac{s-5}{s(s+2)^2}$$

```

f = -1.25 + 3.5 * t * exp(-2*t) + 1.25 * exp(-2*t)
F = simplify(laplace(f,t,s)) → F = (s - 5)/(s*(s + 2)^2)

```

### 6.4 Inverse Laplace

$$F(s) = \frac{s-5}{s(s+2)^2} \rightarrow f(t) = -1.25 + 3.5te^{-2t} + 1.25e^{-2t}$$

```

F = (s-5)/(s*(s+2)^2)    simplify(ilaplace(F))

```

### 6.5 Partialbruchzerlegung

```

b = [4 12]      Zählerpolynom
a = [1 3 2 0]   Nennerpolynom
[r,p,k] = residue(b,a)  Partialbruchzerlegung
r = [2; -8; 6]   Residuen
p = [-2; -1; 0]  Pole
k = []           Restterm der Polynomdivision
→  $\frac{r_n}{x-p_n} + \dots + \frac{r_1}{x-p_1} + \frac{r_0}{x-p_0}$ 

```

$$\frac{b}{a} = \frac{4x-12}{x^3+3x^2+2x}$$

$$\frac{2}{x+2} + \frac{-8}{x+1} + \frac{6}{x} + k$$

### 6.6 LTI-Systeme

$$H(s) = \frac{s \frac{L}{R}}{s^2 LC + s \cdot \frac{L}{R} + 1} \Rightarrow \frac{1s+0}{1s^2+2s+1}$$

```

R = 1Ω    L = 1H    C = 1F    →    H = tf([1 0], [12 1])

```

Bode-Diagramm	bode(H)
Impulsantwort	impulse(H)
Sprungantwort	step(H)
Nullstellen- / Sprungstellendiagramm	pzmap(H)
LTI-Eigenschaftsdiagramm	ltiview(H)