

Kurzeinführung in MATLAB

1. Variablen und Rechenoperationen

Sprachelement	Beispiel
Alphanumerische Variablennamen	x, y, z, M, N, x2, y3, alpha, Laenge, delta_t
Gleitkommazahlen (15 Stellen genau)	1, 1., .1, -1.0, 1.23, 1E11, 2.34e-56
Zuweisung eines Variablenwertes	x = 5; y = -10.2E4; z = x + y;
Grundrechenarten	+ - * / ^
Vergleichsoperationen	< > <= >= !=
Logische Operationen (Und, Oder, Nicht)	& ~
Mathematische Funktionen	sin, cos, tan, sqrt, log, log10, exp, ...
π	pi
Kommentare mit % Beispiel: Polarkoordinatentransformation	r = 10 % Radius phi = pi/2 % Winkel x = r*cos(phi) % x-Koordinate y = r*sin(phi) % y-Koordinate
Bildschirmausgabe unterdrücken	R = 10; % usw.

2. Arbeitsumgebung

Befehl	Bedeutung	Beispiel
cd	Verzeichnis wechseln	cd D:\users\maier\mfiles
dir bzw. ls	Alle Dateien zeigen	dir, ls
pwd	Aktuelles Verzeichnis anzeigen	pwd
what	Alle M-Files zeigen	what
Help, doc	Hilfe zu einer Funktion	help sin oder doc sin
who	Alle Variablen zeigen	who
whos	Alle Variablen mit Wert zeigen	whos (oder Workspace Browser benutzen)
clear	Alle Variablen löschen	clear all

3. Matrizen und Vektoren

Bedeutung	Beispiel	Ergebnis
Matrix definieren	M= [1, 2, 3; 4, 5, 6; 7, 8, 9]	1 2 3 4 5 6 7 8 9
Liegender Vektor	v=[1, 2]	1 2
Stehender Vektor	w=[1; 2]	1 2
Matrizeneintrag	M(2, 3)	6
Vektoreintrag	v(2)	2
Transponieren	w'	1 2
Leerer Vektor	[]	[]
Lineare Sequenz: Von:Schrittweite:Bis	0:0.2:1	0.0 0.2 0.4 0.6 0.8 1.0
Lineare Sequenz: Von:Bis:Anzahl	linspace(0, 1, 5)	0.00 0.25 0.50 0.75 1.00
Wert an Zeilenvektor anhängen	v=[v, 4]	1 2 4
Wert an Spaltenvektor anhängen	w=[w; 4]	1 2 4
Matrix aus Nullen (Zeilenzahl x Spaltenzahl)	zeros(2, 3)	0 0 0 0 0 0

Matrix aus Einsen (Zeilenzahl x Spaltenzahl)	ones(2,4)	1 1 1 1 1 1 1 1
Diagonalmatrix aus Vektor	diag(v)	1 0 0 2
Länge eines Vektors	length(w)	3
Zeilenzahl einer Matrix	size(M,1)	3
Spaltenzahl einer Matrix	size(M,2)	3
Eine Zeile einer Matrix	M(2,:)	4 5 6
Eine Spalte einer Matrix	M(:,2)	2 5 8
Ausschnitt eines Vektors	v(2:3)	2 4
Ausschnitt einer Matrix	M(1:2,2:3)	2 3 5 6

4. Matrizenrechnung

Bedeutung	Beispiel	Ergebnis
	v1=[1;2;3]	1 2 3
	v2=[2;3;4]	2 3 4
	w1=[1 2]	1 2
	w2=[2 3]	2 3
	M1=[1 2 3;... 4 5 6]	1 2 3 4 5 6
	M2=[2 3 4;... 5 6 7]	2 3 4 5 6 7
	P=[1 2;... 2 1]	1 2 2 1
Vektoraddition (stehend)	v1+v2	3 5 7
Vektoraddition (liegend)	w1+w2	3 5
Matrizenaddition	M1+M2	3 5 7 9 11 13
Komponentenweise Multiplikation	w1.*w2	2 6
Komponentenweise Potenzieren	P.^2	1 4 4 1
Vektor-Skalar-Multiplikation	2*w1	2 4
Matrix-Skalar-Multiplikation	2*M1	2 4 6 8 10 12
Matrix-Vektor-Produkt von rechts	M1*v1	14 32
Matrix-Vektor-Produkt von links	w1*M1	9 12 15
Matrix-Multiplikation	P*M1	9 12 15 6 9 12
Skalarprodukt	v1'*v2	20
Tensorprodukt	v1*v2'	2 3 4 4 6 8 6 9 12
Matrix-Potenz	P^2	5 4 4 5
Inverse Matrix	inv(P)	-0.3333 0.6667 0.6667 -0.3333
Eigenwerte	eig(P)	-1 3
Funktionsanwendung (Vektor)	sqrt(w1)	1.0000 1.4142
Funktionsanwendung (Matrix)	sin(P)	0.8415 0.9093 0.9093 0.8415
Summe aller Vektor-Einträge	sum(v1)	6
Spaltensummen einer Matrix	sum(M1)	5 7 9

5. Kontrollstrukturen

Bedeutung	Syntax
if-Schleife	<pre>if <i>Bedingung</i> <i>Befehle</i> elseif <i>Bedingung</i> % elseif-Teil kann entfallen <i>Befehle</i> elseif <i>Bedingung</i> % mehrere elseif möglich <i>Befehl</i> ... else % else-Teil kann entfallen <i>Befehle</i> end;</pre>
Zählschleife	<pre>for <i>Zählvariable</i> = <i>Von</i>:<i>Schrittweite</i>:<i>Bis</i> <i>Befehle</i> end;</pre>
Bedingungsgesteuerte Schleife	<pre>while <i>Bedingung</i> <i>Befehle</i> end;</pre>
Fallunterscheidung (entspricht einer mehrfachen Verzweigung)	<pre>switch <i>Auswahlvariable</i> case <i>Wert1</i> <i>Befehle</i> case (<i>Wert2</i>, <i>Wert3</i>, ...) <i>Befehle</i> otherwise % otherwise-Teil kann entfallen <i>Befehle</i> end;</pre>
Fortsetzung langer Zeilen (Zeilenumbruch innerhalb einer Anweisung)	<pre><i>Befehlsanfang</i> ... <i>Befehlsfortsetzung</i> ... <i>Befehlsende</i>;</pre>
Anhalten und Warten auf Tastendruck	<pre>pause;</pre>

6. Funktionen

Bedeutung	Syntax
Funktionsdefinition	<pre>function [<i>Ausgabe1</i>, <i>Ausgabe2</i>, ...] = <i>Name</i> (<i>Eingabe1</i>, <i>Eingabe2</i>, ...)</pre>
Funktionsaufruf	<pre>[<i>Ausgabe1</i>, <i>Ausgabe2</i>, ...] = <i>Name</i> (<i>Eingabe1</i>, <i>Eingabe2</i>, ...)</pre>
Globale Variable	<pre>global <i>Name</i> % bei Einführung und Verwendung angeben</pre>

Beispiel: M-File `stat.m` (Beachte: Dateiname=Funktionsname) zur Berechnung von Mittelwert und Standardabweichung:

```
function [mean, stdev] = stat(x)
% Diese Zeile wird bei "help stat" als Hilfe ausgegeben
n = length(x);
mean = sum(x) / n;
stdev = sqrt(sum((x - mean).^2)/n);
```

7. Graphik

Befehl	Bedeutung
<code>figure(2)</code>	Legt Grafikfenster mit Nummer 2 an, bzw. wechselt zu diesem Fenster
<code>clf</code>	Löscht den Inhalt des aktuell angewählten Fensters
<code>close</code>	Schließt aktuelles Fenster
<code>close all</code>	Schließt alle Fenster
<code>axis([xu, xo, yu, yo])</code>	Zeichnet x-Achse mit Beschriftung von xu bis xo und y-Achse mit Beschriftung von yu bis yo
<code>axis(axis)</code>	Fixiert die Achsenbeschriftung bei nachfolgenden Plots
<code>axis('equal')</code>	Erzwingt gleiche Maßstäbe auf beiden Achsen
<code>hold on</code>	Fixiert die Zeichnung, d.h. kein Löschen bei nachfolgenden plot-Befehlen

<code>plot(x,y, 'b-o')</code>	Zeichnet Werte des Vektors x gegen Werte des Vektors y in blau mit durchgezogener Linie und zusätzlich mit einem Kreis pro Datenpunkt. Farbcodes: y m c r g b w k Liniencodes: - : -. -- Punktcodes: . o x + * s d v ^ < > p h
<code>subplot(Z,S,N)</code>	Ntes Teilbild in einer Plotanordnung mit Z Zeilen und S Spalten in der aktuellen Figur
<code>[x,y,button]=ginput(1)</code>	Mauseingabe eines Punktes (button=1 für links, button=3 für rechts)
<code>plot3(x,y,z)</code>	3D-Kurvenplot
<code>surf(X,Y,XY)</code>	Oberflächenplot der in der Matrix XY gespeicherten Gitterwerte mit Skaleneinteilung X und Y

8. Differentialgleichungen

Beispiel-Differentialgleichung (Pendel):

$$\dot{\varphi} = \omega$$

$$\dot{\omega} = -\frac{d}{ml^2} \cdot \omega - \frac{g}{l} \cdot \sin(\varphi)$$

Codierung der rechten Seite als Matlab-Funktion

pendulum.m mit dem Zustandsvektor $x=(\varphi,\omega)^T$:

```
function dfdx=pendulum(t,x);
```

```
% Sprechende Namen einführen
```

```
phi =x(1);
```

```
omega=x(2);
```

```
% Rechte Seite der Gleichung
```

```
dphi_dt = omega;
```

```
domega_dt= -d/(m*l^2)*omega...  
           -g/l*sin(phi);
```

```
% Rückgabe als Vektor
```

```
dfdx=[dphi_dt; domega_dt];
```

Lösung der Gleichung z.B. mit Zeitspanne $0 \leq t \leq 20$ und Startwerten $(0,1)^T$:

```
[T,X]=ode45(@loopode,[0,20],[1,0]);
```

Lösung ist anschließend gespeichert in **T,X** als Zeitskala bzw. Matrix mit zwei Spalten für φ,ω .

Weitere Solver: **ODE23, ODE113, ODE15S, ODE23S, ODE23T, ODE23TB**

Zusätzliche Steuerparameter für die Solver werden übergeben in der Form:

```
[T,X]=ode45...  
      (@loopode,[0,20],[1,0],opt);
```

Die Steuerparameter in **opt** in werden gesetzt über:

```
odeset(Param1, Wert1, Param2, Wert2, ...);
```

Wichtige Parameter:

RelTol	relative Fehlertoleranz
AbsTol	absolute Fehlertoleranz
OutputSel	Ausgabe-Variablen
MaxStep	maximale Schrittweite

Beispiel:

```
odeset('RelTol',1E-6, 'MaxStep',1);
```