

MATLAB

Eine Einführung

Marina Schneider

18. Mai 2010

- 1 Allgemeines zu MATLAB und erste Schritte
- 2 Vektoren und Matrizen
- 3 Graphiken
- 4 MATLAB als Programmiersprache

Was ist MATLAB?

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

- Softwarepaket für numerische Berechnungen und die Visualisierung von Daten im technisch-wissenschaftlichen Bereich
- MATLAB= MATrix LABoratory
- Einsatz vor allem in Numerik u. Linearer Algebra

MATLAB starten

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmier-
sprache

Linux: Konsole (Shell) öffnen und *matlab* eingeben
Windows: Über Programme auswählen und starten

MATLAB verwenden

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Es gibt zwei Arten MATLAB zu verwenden:

- **Interaktive Verwendung**

Anweisungen direkt über die Tastatur in das Commandfenster eingegeben und sofort ausführen

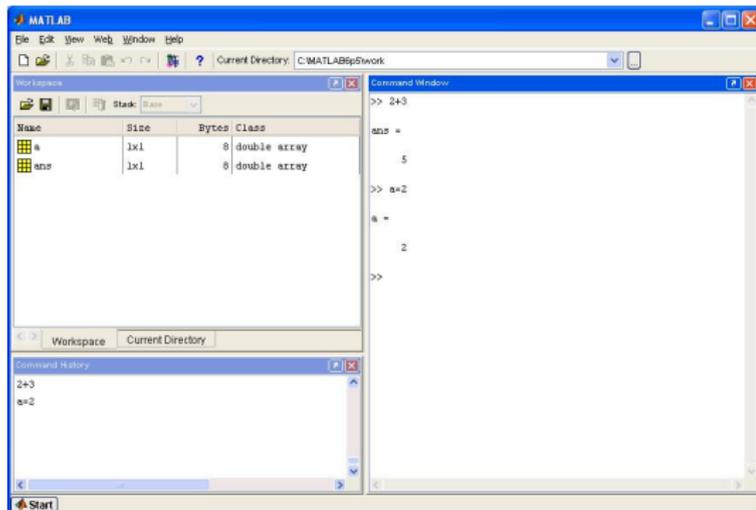
- **Verwendung als Programmiersprache**

→ Empfehlenswert für umfangreiche Probleme !

- 1 Allgemeines zu MATLAB und erste Schritte
 - Interaktiver Modus
 - MATLAB als Programmiersprache - Grundlegendes
- 2 Vektoren und Matrizen
- 3 Graphiken
 - Zweidimensionale Graphiken
 - Dreidimensionale Graphiken
- 4 MATLAB als Programmiersprache
 - Kontrollstrukturen
 - Effizienz von Programmen
 - Interaktive Eingabe

1. Interaktiver Modus

- Anweisungen direkt über die Tastatur in das Commandfenster eingegeben
- MATLAB führt diese sofort aus und gibt das Ergebnis im Commandfenster aus



MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Definition von Variablen

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

- MATLAB speichert Variablen im *Workspace*

$$x = 7$$

- Ein Strichpunkt am Ende der Eingabezeile verhindert, dass MATLAB das Ergebnis ausgibt

$$\text{Variable1} = 5 ;$$

- **Regeln für die Benennung von Variablen**

- Keine Sonderzeichen, außer dem Unterstrich
- Erstes Zeichen muss ein Buchstabe sein
- Nicht mehr als 19 Zeichen

Eingebaute Funktionen

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

- Es gibt in MATLAB einige vordefinierte Funktionen
- *Beispiel:* Exponentialfunktion

```
x = exp(2);
```

Nähere Informationen zur MATLAB-Funktion `exp` erhält man mit

```
help exp
```

- Eine Liste aller eingebauten elementaren Funktionen erhält man mit

```
help elfun
```

Eingebaute Funktionen (2)

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Vorsicht

Die Anweisung $y = \exp(x)$ kann nur dann von MATLAB ausgewertet werden kann, wenn x einen Zahlenwert hat, andernfalls erhält man eine Fehlermeldung!

MATLAB ist vor allem auf die Verarbeitung numerischer Daten ausgerichtet. Symbolische Ausdrücke können nur mit der Symbolic Math Toolbox verarbeitet werden.

- 1 Allgemeines zu MATLAB und erste Schritte
 - Interaktiver Modus
 - MATLAB als Programmiersprache - Grundlegendes
- 2 Vektoren und Matrizen
- 3 Graphiken
 - Zweidimensionale Graphiken
 - Dreidimensionale Graphiken
- 4 MATLAB als Programmiersprache
 - Kontrollstrukturen
 - Effizienz von Programmen
 - Interaktive Eingabe

2. Verwendung als Programmiersprache

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

- Empfehlenswert für umfangreiche Probleme
- Mehrere Anweisungen werden zusammen als sogenannte m-Files abgespeichert
- m-Files werden im Commandfenster aufgerufen und vom MATLAB-Interpreter wie ein Programm ausgeführt

Arten von m-Files

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Man unterscheidet zwei Arten von m-Files:

- 1 Script-Files (→ Programme)
- 2 Function-Files (→ Funktionen)

Beide werden mit der Endung *.m abgespeichert.

Script-Files

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

- Aneinanderreihung von MATLAB-Befehlen, die in einen Texteditor eingegeben werden
- Wird der Name des Files im Commandfenster eingegeben, so führt MATLAB die Befehle im Script-File der Reihe nach aus, so als ob sie einzeln in das Commandfenster eingegeben worden wären

Beispiel: Erstellen eines einfachen Script-Files

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

- **Editor öffnen** : File → New → M-File
- **Gewünschte Befehle eingeben**: z.B.
a=2;
b=3;
c=a+b;
d=a*b;
- **Speichern** : File → Save As → Beispiel.m
- **Ausführen** : Namen des Files (also Beispiel) im
Commandfenster eingeben

Function-Files

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

- Über Function-Files werden neue Funktionen definiert
- Diese Funktion kann dann über das Commandfenster oder in anderen m-Files aufgerufen werden
- Beim Aufruf eines Function-Files wird ihm ein Wert (oder mehrere) als Argument übergeben (z.B. wie bei $\sin(x)$)
- Nach Auswertung der Befehle im Function-File wird ein Wert (oder mehrere) ausgegeben, der Funktionswert

- **Allgemeine Struktur:**

```
function Variablenname =  
Funktionsname(Argumentliste)  
Anweisungen  
Variablenname =...;
```

Beispiel: Erstellen eines einfachen Function-Files

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Beispiel

$$f : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto \frac{3x^2+1}{2}$$

- **Editor öffnen** : File → New → M-File
- **Gewünschte Befehle eingeben**:

```
function y=f(x)  
y=(3*x*x +1)/2;
```
- **Speichern** : File → Save As → f.m
- **Aufrufen** : Eingabe von $f(2)$ (im Commandfenster oder in einem Script-File) berechnet Funktionswert an der Stelle 2

Der Befehl `fprintf`

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

- `fprintf` schreibt Daten in formatierter Form in eine externe Datei
- Ist keine Datei angegeben, werden die Daten auf den Bildschirm geschrieben

Der Befehl fprintf (2)

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Syntax

```
fid =fopen('Beispieldatei.txt', 'w')  
fprintf(fid, format, A, ...)  
fclose(fid)
```

- fopen ist eine Befehl, der die gewünschte Datei Beispieldatei.txt erstellt
- fid (=file identifier) ist die Verknüpfung mit der Datei
- format legt die Form der Ausgabe für die nachfolgende Variable fest (sog. Formatbezeichner)
- A ist eine Matrix, die die auszugebenden Werte enthält
- fclose(fid) schließt die bearbeitete Datei wieder

Einfaches Beispiel - fprintf

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Ziel

Wertetabelle der Exponentialfunktion auf dem Intervall [0,1]

```
x=0:0.1:1;  
fid =fopen('Beispieldatei.txt','w');  
fprintf(fid,'\n');  
fprintf(fid,'Wertetabelle: \n\n');  
fprintf(fid,' x exp(x) \n');  
fprintf(fid,'-----\n');  
fprintf(fid,' % 12.6f % 12.6f \n',[x;exp(x)]);  
fclose(fid);
```

Die wichtigsten Formatbezeichner

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

■ Ganze Zahlen:

`% d` : Ganze Zahl; Länge der Darstellung von MATLAB festgelegt

`% nd` : Darstellung einer ganzen Zahl mit Länge n

■ Reelle Zahlen:

`% f` : Darstellung einer reellen Zahl in Festkommadarstellung. Gesamtanzahl der Stellen legt MATLAB fest.

`% w.nf` : Rechtsbündige Darstellung einer reellen Zahl mit insgesamt w Stellen und n Nachkommastellen.

■ Exponentialdarstellung:

`% e` : Darstellung einer reellen Zahl in Exponentialdarstellung (empfehlenswert für sehr große Zahlen)

- 1 Allgemeines zu MATLAB und erste Schritte
 - Interaktiver Modus
 - MATLAB als Programmiersprache - Grundlegendes
- 2 Vektoren und Matrizen
- 3 Graphiken
 - Zweidimensionale Graphiken
 - Dreidimensionale Graphiken
- 4 MATLAB als Programmiersprache
 - Kontrollstrukturen
 - Effizienz von Programmen
 - Interaktive Eingabe

Eingabe von Vektoren und Matrizen

■ Vektoren

Zeilenvektor: $x = [1 \ 2 \ 3]$ oder $x = [1 \ , \ 2 \ , \ 3]$

Spaltenvektor: $x = [1 \ ; \ 2 \ ; \ 3]$

■ Matrizen

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

$$A = [1 \ 2 \ 3 \ ; \ 4 \ 5 \ 6 \ ; \ 7 \ 8 \ 9]$$

■ Schnelles Erzeugen von Vektoren:

$$x = 0:2:12$$

erzeugt den Vektor

$$x = (0 \ 2 \ 4 \ 6 \ 8 \ 10 \ 12)$$

Allgemein: Anfangswert : Abstand : Endwert

Vektoren und Matrizen

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

- Die Transponierte einer (reellen) Matrix erhalten wir durch Anhängen eines Apostroph

$$C = A'$$

- Die Länge eines Vektors x erhalten wir mit dem Befehl

`length(x)`

- Die Dimension einer Matrix A erhalten wir mit dem Befehl

`size(A)`

Das Ergebnis ist der Vektor [*Zeilenanzahl*, *Spaltenanzahl*]

Vektoren und Matrizen

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Hinweis

I.d.R. können die in Matlab eingebauten Funktionen auch auf Vektoren und Matrizen angewandt werden.

Beispiel:

Für den Vektor

$$x = [4 \ 9 \ 16]$$

liefert der Befehl

$$y = \text{sqrt}(x)$$

das Ergebnis

$$y = [2 \ 3 \ 4] .$$

Matlab berechnet also elementweise

$$y = (\sqrt{x_1} \quad \sqrt{x_2} \quad \sqrt{x_3})$$

Übung

Erzeugen Sie einen Vektor y , der die Funktionswerte des natürlichen Logarithmus an den Stellen $x = 1, 3, 5, 7, 9$ enthält.

Matrixoperationen

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

- $A+B$, $A-B$, $A*B$ bezeichnen die üblichen Matrixoperationen.
Beachte: $A*B$ ist nur definiert, wenn die Anzahl der Spalten von A gleich der Anzahl der Zeilen von B ist.
- $.*$ bewirkt eine elementweise Multiplikation
- $\text{inv}(A)$ liefert die Inverse einer (regulären) Matrix A
- $A \setminus b$ liefert die Lösung des linearen Gleichungssystems $Ax = b$ (wenn eine Lösung existiert).

Matrixoperationen (2)

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

- $\det(A)$ liefert die Determinante einer Matrix A
- $\text{eig}(A)$ liefert einen Vektor mit den Eigenwerten von A
- $\text{poly}(A)$ liefert einen Vektor mit den Koeffizienten des charakteristischen Polynoms von A

Übung

Betrachten Sie die Funktion

$$f : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto x^2$$

Erstellen sie ein MATLAB-Programm (**.m-file*), welches mit Hilfe des Befehls `fprintf` eine Wertetabelle der Funktion f für das Intervall $[-1,1]$ erzeugt. Geben sie diese sowohl auf dem Bildschirm, als auch in eine Datei aus.

- 1 Allgemeines zu MATLAB und erste Schritte
- 2 Vektoren und Matrizen
- 3 Graphiken
- 4 MATLAB als Programmiersprache

1 Allgemeines zu MATLAB und erste Schritte

2 Vektoren und Matrizen

3 Graphiken

4 MATLAB als Programmiersprache

- 1 Allgemeines zu MATLAB und erste Schritte
 - Interaktiver Modus
 - MATLAB als Programmiersprache - Grundlegendes
- 2 Vektoren und Matrizen
- 3 Graphiken
 - Zweidimensionale Graphiken
 - Dreidimensionale Graphiken
- 4 MATLAB als Programmiersprache
 - Kontrollstrukturen
 - Effizienz von Programmen
 - Interaktive Eingabe

Der Befehl plot

- `plot` dient zur Veranschaulichung numerischer Daten
- `plot(x,y)` erzeugt ein Fenster mit der gewünschten Graphik
 x,y sind Vektoren, die die x - bzw. y -Koordinaten der zu zeichnenden Datenpunkte enthalten.

Beispiel

Zeichne $f : [0, 2] \rightarrow \mathbb{R}, x \mapsto e^x$

```
x = 0:0.1:2;  
y = exp(x);  
plot(x,y); grid
```

Der Befehl `plot` (2)

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Bemerkungen:

- Der Befehl `grid` erzeugt das Gitter.
→ Kann auch weggelassen werden !
- Mit dem Befehl `linspace` kann man die Anzahl der gezeichneten Datenpunkte vorgeben.
- Mit dem Befehl `axis` kann man den zu zeichnenden Bereich vorgeben

```
x = linspace( Anfangspunkt, Endpunkt,  
              Zahl d. Punkte );
```

```
axis([xmin, xmax, ymin, ymax])
```

Beschriften von Grafiken

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmier-
sprache

Zum Beschriften von Grafiken dienen die Befehle

- `xlabel`, `ylabel` : Beschriftung der Koordinatenachsen
- `title` Überschrift der Grafik
- `text` Text in die Grafik

Beispiel:

```
plot(x,y);  
xlabel('Wert der Variable x'),  
ylabel('Wert von exp(x) ')  
title('Graph von f(x)= exp(x) ')  
text(-0.8,0.5,'Testtext')
```

Zeichnen mehrerer Graphen

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

1. Mehrere Graphen in einem Koordinatensystem

- Hierzu dient der Befehl `hold`
- Alle folgenden Plots werden in das selbe System gezeichnet
- Wird beendet mit `hold off`

Beispiel:

```
x = -pi:0.1:pi ;  
y1=sin(x); y2=cos(x);  
plot(x,y1); hold;  
plot(x,y2);  
hold off;
```

Zeichnen mehrerer Graphen (2)

2. Mehrere Graphen in verschiedenen Koordinatensystem

- `subplot (m,n,p)` zerlegt das Grafikfenster in mehrere Teilfenster
- `m` Zahl der Teilfenster in vertikaler Richtung
- `n` Zahl der Teilfenster in horizontaler Richtung
- `p` Nummer des Teilfensters, in das der aktuelle Graph gezeichnet werden soll.
(Gezählt wird zeilenweise von links oben nach rechts unten.)

Beispiel:

```
x = -pi:0.1:pi ;  
y1=sin(x); y2=cos(x);  
subplot(2,1,1); plot(x,y1);  
subplot(2,1,2); plot(x,y2);
```

- 1 Allgemeines zu MATLAB und erste Schritte
 - Interaktiver Modus
 - MATLAB als Programmiersprache - Grundlegendes
- 2 Vektoren und Matrizen
- 3 Graphiken
 - Zweidimensionale Graphiken
 - Dreidimensionale Graphiken
- 4 MATLAB als Programmiersprache
 - Kontrollstrukturen
 - Effizienz von Programmen
 - Interaktive Eingabe

Erzeugen eines Gitters

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

- **Ziel:** Wir wollen den Graph einer Funktion $f(x, y)$ als Fläche, über der (x, y) - Ebene zeichnen
- Dazu muss zunächst ein Gitter mit den gewünschten (x, y) -Werten erzeugt werden
- Hierzu dient der Befehl `meshgrid(x, y)`
Zu gegebenen Vektoren x , y erzeugt er zwei Matrizen. Einander entsprechende Elemente dieser beiden Matrizen bilden gerade die beiden Koordinaten unserer Gitterpunkte.

Beispiel: Erzeugen eines Gitters

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Beispiel

Erzeuge ein Gitter für $x \in [-1, 1]$ und $y \in [2, 4]$!

```
x = -1:0.5:1;  
y = 2:0.5:4;  
[ X,Y ] =meshgrid(x,y)
```

liefert

$$X = \begin{pmatrix} -1 & -0.5 & 0 & 0.5 & 1 \\ -1 & -0.5 & 0 & 0.5 & 1 \\ -1 & -0.5 & 0 & 0.5 & 1 \\ -1 & -0.5 & 0 & 0.5 & 1 \\ -1 & -0.5 & 0 & 0.5 & 1 \end{pmatrix}, Y = \begin{pmatrix} 2 & 2 & 2 & 2 & 2 \\ 2.5 & 2.5 & 2.5 & 2.5 & 2.5 \\ 3 & 3 & 3 & 3 & 3 \\ 3.5 & 3.5 & 3.5 & 3.5 & 3.5 \\ 4 & 4 & 4 & 4 & 4 \end{pmatrix}$$

Zeichnen des Graphen

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Beispiel

$$f : [-1, 1] \times [2, 4] \rightarrow \mathbb{R}^2, (x, y) \mapsto x \cdot y$$

- Erzeuge Vektor mit Funktionswerten

$$Z=X.*Y;$$

Wichtig: Nicht `.` für elementweise Operation vergessen !

- Erzeuge Plot mit

`mesh(X, Y, Z)` Jeder dieser Funktionswerte wird über dem entsprechenden Gitterpunkt als Punkt dargestellt und in x - und in y -Richtung durch Strecken mit seinen Nachbarpunkten verbunden werden

`surf(X, Y, Z)` Verwendet kleine Flächenstücke zwischen den Punkten, statt sie durch Strecken zu verbinden

Übung

- Erstellen sie ein MATLAB-Programm (**.m-file*), welches die Funktion

$$f : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto x^2$$

auf dem Intervall $[-1,1]$ graphisch ausgibt.

- Nutzen sie die Befehle `mesh` und `surf`, um die Funktion

$$g : \mathbb{R}^2 \rightarrow \mathbb{R}, (x, y) \mapsto x \cdot \sin(3\pi x) \cdot \cos(2\pi y)$$

auf dem Intervall $[0,1] \times [0,1]$ zu plotten.

- 1 Allgemeines zu MATLAB und erste Schritte
- 2 Vektoren und Matrizen
- 3 Graphiken
- 4 MATLAB als Programmiersprache

Vorbemerkung:

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

- Dieses Kapitel bezieht sich speziell auf das Arbeiten mit m-Files
- Hierbei wird eine Folge von Befehlen in einem eigenen File mit der Endung *.m abgespeichert (vgl. 1.Tutorium)

- 1** Allgemeines zu MATLAB und erste Schritte
 - Interaktiver Modus
 - MATLAB als Programmiersprache - Grundlegendes
- 2** Vektoren und Matrizen
- 3** Graphiken
 - Zweidimensionale Graphiken
 - Dreidimensionale Graphiken
- 4** MATLAB als Programmiersprache
 - Kontrollstrukturen
 - Effizienz von Programmen
 - Interaktive Eingabe

Kontrollstrukturen

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

- Mit Kontrollstrukturen kann der Ablauf eines Programms gesteuert werden
- Wir betrachten hierzu
 - 1 for
 - 2 while
 - 3 if, else, elseif

Syntax

```
for Zählvariable = Anfangswert : Abstand : Endwert  
Anweisungen  
end
```

- Die Anweisungen im `for/end`-Block werden für jeden angegebenen Wert der Zählvariablen wiederholt
- Die Zählvariable wird zunächst auf den Startwert gesetzt und die Anweisungen werden durchgeführt, dann wird sie um die gegebene Schrittweite erhöht und die Anweisungen werden erneut durchgeführt, dies wird wiederholt bis der Endwert erreicht ist
- Wird kein Abstand angegeben wählt MATLAB automatisch den Abstand 1
- Die Schleife kann durch den Befehl `break` beendet werden

Beispiel 1: for-Schleife

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

```
x = [ 0 0 0 0 ]
```

```
for i= 1 : 4  
x(i) = i ;  
end
```

liefert das Ergebnis

```
x = [ 1 2 3 4 ]
```

Beispiel 2: for-Schleife

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

```
summe = 0
```

```
for i= 1 : 100
```

```
summe = summe + i ;
```

```
end
```

Diese Konstruktion addiert alle Zahlen von 1 bis 100 auf.

Syntax

```
while Kriterium für den Eintritt in die Schleife  
Anweisungen  
end
```

- Das Kriterium für den Eintritt in die Schleife ist eine logische Bedingung,
z.B.: $i \leq 100$, auch möglich: $>$, \geq , $==$
- Solange diese Bedingung erfüllt ist, wird die Folge von Anweisungen in der Schleife wiederholt
- Die Schleife kann durch den Befehl `break` beendet werden

Beispiel: while-Schleife

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

```
summe = 0
```

```
while i <= 100
```

```
summe = summe + i ;
```

```
i = i+1 ;
```

```
end
```

Diese Konstruktion addiert alle Zahlen von 1 bis 100 auf.

while-Schleife : Warnung

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Vorsicht

Es ist sehr wichtig darauf zu achten, dass das Abbruchkriterium irgendwann erfüllt wird.

Ist dies nicht der Fall, werden die Anweisungen in der Schleife endlos oft wiederholt, man kommt nicht mehr heraus.

Tipp:

Gerät man in eine Endlosschleife, so kann das laufende Programm mit der Tastenkombination

Strg + C

abgebrochen werden.

if-Bedingungen

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Syntax

`if` *Ausdruck1*

Anweisungen

`elseif` *Ausdruck2*

Anweisungen

`else`

Anweisungen

`end`

Ausdruck1, Ausdruck2 sind wieder logische Bedingung.

Beispiel: if

Wir wollen die folgende Funktion definieren

$$h : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto \begin{cases} 0, & \text{wenn } x < 0 \\ x, & \text{wenn } x \in [0, 1] \\ 1, & \text{wenn } x > 1 \end{cases}$$

Lösung:

```
function y = h(x)
if x < 0
y=0 ;
elseif x >= 0 & x <=1
y = x ;
else
y = 1 ;
end
```

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Vergleichsoperatoren

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Für die Bedingungen in den `for` / `while` / `if`-Konstrukten benötigt man diese Operatoren. Hierzu zählen

- 1 $<$ (kleiner als)
- 2 \leq (kleiner oder gleich)
- 3 $>$ (größer als)
- 4 \geq (größer gleich)
- 5 $==$ (gleich)
- 6 \sim (ungleich)

Beispiel: $x \leq 1$

Matlab prüft den Wahrheitsgehalt dieser Aussage und schafft eine neue Variable, deren Wert 1 ist, wenn die Aussage wahr ist und 0 sonst.

Logische Operatoren

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Diese Operatoren verknüpfen logische Aussagen miteinander.

Logisches Und `Aussage1 & Aussage2`, bzw.

```
and( Aussage1, Aussage2 )
```

Logisches Oder `Aussage1 | Aussage2`, bzw.

```
or( Aussage1, Aussage2 )
```

Logisches EntwederOder `xor(Aussage1, Aussage2)`

Logisches Nicht `~(Aussage)`, bzw. `not(Aussage)`

Beispiel :

```
if x >=0 & x <= 1
```

```
....
```

```
end
```

Übung

- 1** Schreiben Sie ein Funktion namens Betrag, die für jedes $x \in \mathbb{R}$ den Betrag von x berechnet!
(Tipp: if- Konstrukt)
- 2** Schreiben Sie ein Programm welches die kleinste natürliche Zahl n bestimmt, deren Quadrat größer als 80 ist! (Tipp: while-Schleife)
- 3** Schreiben Sie eine Funktion, die für jede natürliche Zahl n die Fakultät berechnet. Zur Erinnerung:

$$n! = 1 \cdot 2 \cdot \dots \cdot (n - 1) \cdot n$$

(Tipp: for-Schleife oder Rekursion)

- 1** Allgemeines zu MATLAB und erste Schritte
 - Interaktiver Modus
 - MATLAB als Programmiersprache - Grundlegendes
- 2** Vektoren und Matrizen
- 3** Graphiken
 - Zweidimensionale Graphiken
 - Dreidimensionale Graphiken
- 4** MATLAB als Programmiersprache
 - Kontrollstrukturen
 - Effizienz von Programmen
 - Interaktive Eingabe

Messung der Programmlaufzeit

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

- Die Effizienz von Programmen kann auf verschiedene Arten verglichen werden.
- Wir betrachten hier zwei Arten, um die Laufzeit von Programmen zu messen.

1.Möglichkeit: `cputime`

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Syntax

```
t = cputime;  
Anweisungen  
time = cputime - t;
```

Nach Ausführung des Programms enthält die Variable `time` die benötigte Zeit.

2.Möglichkeit: tic, toc

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Syntax

```
tic ;  
Anweisungen  
time = toc;
```

Nach Ausführung des Programms enthält die Variable `time` die benötigte Zeit.

Beispiel: tic, toc

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Frage:

Wieviel Zeit benötigt MATLAB zur Lösung des linearen Gleichungssystems

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Lösung:

```
A=[ 1 0 0; 0 1 0; 0 0 4];
```

```
b = [1 ; 1 ;1]
```

```
tic ;
```

```
A\b
```

```
time = toc;
```

- 1** Allgemeines zu MATLAB und erste Schritte
 - Interaktiver Modus
 - MATLAB als Programmiersprache - Grundlegendes
- 2** Vektoren und Matrizen
- 3** Graphiken
 - Zweidimensionale Graphiken
 - Dreidimensionale Graphiken
- 4** MATLAB als Programmiersprache
 - Kontrollstrukturen
 - Effizienz von Programmen
 - Interaktive Eingabe

Programme effizient gestalten

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Vektorisierung:

Oft kann man eine Schleife durch eine Vektoroperation ersetzen.

Beispiel:

```
for i= 1:10  
x(i)= 2*i;  
end
```

kann man ersetzen durch

```
y = 1:10; x=2.*y;
```

Programme effizient gestalten (2)

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Allgemein gilt:

Vektorisierung, d.h. das Ersetzen einer Schleife durch eine Vektoroperation hat verschiedene Vorteile

■ **Geschwindigkeit:**

- MATLAB kann Vektoren und Matrizen schnell verarbeiten, ist aber bei Schleifen nicht effizient
- Viele eingebaute MATLAB-Funktionen werden schneller ausgeführt, wenn man anstelle eines mehrfachen Aufrufs einen Vektor übergibt

■ **Übersichtlichkeit:**

Vektorisierte MATLAB-Programme sind meist übersichtlicher als ihre skalaren Gegenstücke.

Übung

Berechnen sie die Werte der Exponentialfunktion an den Stellen $x = 1, 2, \dots, 5000$ mit einer for-Schleife und mit Hilfe eines Vektors.

Bestimmen sie jeweils die benötigte Zeit.

- 1 Allgemeines zu MATLAB und erste Schritte
 - Interaktiver Modus
 - MATLAB als Programmiersprache - Grundlegendes
- 2 Vektoren und Matrizen
- 3 Graphiken
 - Zweidimensionale Graphiken
 - Dreidimensionale Graphiken
- 4 MATLAB als Programmiersprache
 - Kontrollstrukturen
 - Effizienz von Programmen
 - Interaktive Eingabe

Die Befehle `disp` und `input`

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

`disp` und `input` sind nützlich, wenn man

- Text oder andere Daten im Commandfenster ausgeben möchte
- Den Benutzer eines Programmes auffordern möchte, einer Variablen über das Commandfenster einen Wert zuzuordnen

Der Befehl disp

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

- Will man einen Text im Commandfenster ausgeben, so gibt man (im Script-file oder direkt ins Commandfenster) `disp('text')` ein.

Beispiel:

```
disp('Guten Tag')
```

- Auch der Wert einer Variablen kann auf diese Weise ausgegeben werden.

Beispiel:

```
a = 65; disp(a)
```

Der Befehl input

MATLAB
Eine
Einführung

Marina
Schneider

Inhalt

Allgemeines zu
MATLAB und
erste Schritte

Vektoren und
Matrizen

Graphiken

MATLAB als
Programmiersprache

Mit `input` kann man den Benutzer auffordern, einer Variablen einen Wert zuzuordnen, so verwenden wir

Beispiel:

```
x = input('Geben Sie einen Wert für x ein: ')
```

- Im Commandfenster erscheint der Satz:
“Geben Sie einen Wert für x ein: “
- Eingabe einer beliebigen Zahl ordnet x den entsprechenden Wert zu