

MATLAB

Eingabe und Ausgabe

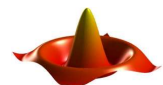


MATLAB Eingabe und Ausgabe

Steuerung der Bildschirmausgabe

Allgemein: `on` aktiviert, `off` deaktiviert Befehls-Funktion,
der Befehl alleine wechselt den Zustand

- Protokolldatei erstellen: `diary`, `diary datei`
- seitenweise Bildschirmausgabe: `more`, `more(n)`
- Befehle anzeigen/verbergen: `echo`
- Bildschirmausgabe anhalten: `pause`, `pause(n)`
- Bildschirm rücksetzen: `clc`



MATLAB Eingabe und Ausgabe

Eingabe über Benutzerdialoge

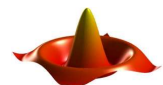
- Strings: – Zeilenvektoren aus Zeichen (char):
`text = ['Das ist', ' ', 'ein Text!']`
– Funktionen für Strings: `help strfun`
- Eingabe von: – Daten: `variable = input(string)`
– Text: `string = input(string, 's')`
- Sonderzeichen: `\n` Zeilenumbruch
`\\` Backslash `\`
`''` Anführungszeichen `'`



MATLAB Eingabe und Ausgabe

Formatierte Ausgabe

- `disp(string)`: Ausgabe des Strings *string*
- Formatierung:
 - `string = sprintf(string,variable)`
 - Syntax entspricht weitgehend C
 - Auch vektorisierte Daten möglich
 - Umwandlung von Zahlen in Strings mittels des Befehls `num2str(variable[,format])`



MATLAB Eingabe und Ausgabe

Import und Export von Daten

- Daten laden: `load datei [-mat| -ascii] [variable...]`
- Daten speichern: `save datei [options] [variable...]`

options:

<code>-mat</code>	Binäre MAT-Datei
<code>-ascii</code>	8-digit ASCII Format
<code>-append</code>	Daten anhängen (MAT)
<code>-ascii -double</code>	16-digit ASCII Format
<code>-ascii -tabs</code>	Mit Tabulator getrennt

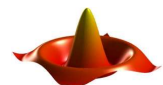
- Formatiertes Schreiben in Textdatei:
 - Datei öffnen: `fid = fopen(datei.endung, zugriff)`
 - Schreiben: `fprintf(fid, string, variable)`
 - Datei schließen: `fclose(fid)`



MATLAB Eingabe und Ausgabe

Betriebssystemaufruf und Dateiverwaltung

- Pfad:
 - aktuell: `pwd`
 - anzeigen: `path`
- Verzeichnis:
 - wechseln: `cd verzeichnis`
 - erstellen: `mkdir verzeichnis`
 - Inhalt anzeigen: `dir [verzeichnis]`
`ls [verzeichnis]`
- Datei:
 - kopieren: `copyfile quelle ziel`
 - löschen: `delete datei`
- Aufruf des Betriebssystems: `! os – befehl`



MATLAB Eingabe und Ausgabe

Graphik

- 2– und 3–dimensionales Plotten von Daten möglich
- Erzeugen einer Graphik (Figure): `figure`
- Aktuelle Figurenummer (*handle*): `gcf`
- Unterplots in einer Figure: `subplot(z,s,n)`
- Figure:
 - rücksetzen: `clf`
 - löschen: `delete.figure(nummer)`
 - schließen: `close(nummer)`



MATLAB Eingabe und Ausgabe

Eigenschaften einer Graphik

- Jedes **Objekt** einer Figure hat einen **Handle** h mit einer Eigenschaft Eig und dessen Wert $EigWert$
Wichtige Objekte: Figure und Achsen
- **Auslesen** einer Eigenschaft: `get(h, 'Eig')`
- **Setzen** einer Eigenschaft: `set(h, 'Eig', EigWert)`
- **Löschen** des Wertes: `delete(h)`
- Alternativ: *Property Editor*



MATLAB Eingabe und Ausgabe

2D–Graphik: Achsen, Skalierung und Beschriftung

- **Aktuelle Achsennummer** (*handle*): `gca`
- **Skalierung:** `axis ([x_min,x_max,y_min,y_max])`
 - Automatisch: `axis('auto')`
 - Gitternetz einblenden: `grid [on | off]`
 - Zoomfunktion aktivieren: `zoom [on | off]`
- **Beschriftung:**
 - Achse: `xlabel(string)`, `ylabel(string)`
 - Überschrift: `title(string)`
 - Text plazieren: `text(x_wert,y_wert,string)`
 - Legende: `legend(str_1,str_2... [, position])`



MATLAB Eingabe und Ausgabe

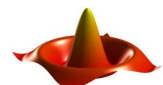
2D-Graphik: Plotbefehle

- Plot-Befehl: `plot(x_werte,y_werte... [,plotstil])`
x_werte, y_werte: gleich lange Vektoren
- Vorhandene Objekte beibehalten: `hold [on | off]`
- Farben und Stil der Linien mit *plotstil* einstellbar:

Farben			
k	schwarz	r	rot
b	blau	m	mangenta
c	cyan	y	gelb
g	grün	w	weiß

Linien und Punkte			
-	durchgezogen	o	Kreise
--	gestrichelt	*	Sterne
:	gepunktet	+	Kreuze
.	Punkte	x	Diagonalkreuze

- Beispiel: `plot(1:0.1:2*pi,sin(1:0.1:2*pi),'r-.')`



MATLAB Eingabe und Ausgabe

2D-Graphik: Spezielle Plotbefehle

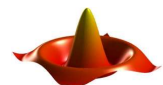
- Treppenförmig: `stairs([x,] y... [,plotstil])`
- Diskret: `stem([x,] y... [,plotstil])`
- Logarithmisch: `loglog(x, y... [,plotstil])`
 - x-Achse: `semilogx(x, y... [,plotstil])`
 - y-Achse: `semilogy(x, y... [,plotstil])`
- Funktionen:
 - explizite Funktion: `fplot(f,bereich)`
 - implizite Funktion: `ezplot(f(x,y),bereich)`
 - Parameterkurve: `ezplot(f_1,f_2,bereich)`



MATLAB Eingabe und Ausgabe

3D-Graphik: Plotbefehle

- Punkt/Linien: `plot3(x,y,z... [, plotstil])`
- Fläche: `surf(x,y,z... [, farbe])`
- Gitter: `mesh(x,y,z... [, farbe])`
- Wasserfall: `waterfall(x,y,z... [...])`
- Höhenlinien: `contour(x,y,z... [...])`
- x, y, z : Matrizen gleicher Dimension
- `[X,Y] = meshgrid(x_vek,y_vek)` erzeugt aus Vektoren x_vek und y_vek Koordinatenmatrizen richtiger Größe



MATLAB Eingabe und Ausgabe

3D-Graphik: Beschriftung, Perspektive und Farben

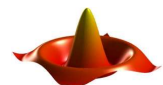
- Skalierung: `axis ([x...,y...,z_min,z_max])`
- Beschriftung z-Achse: `zlabel(string)`
- Box einblenden: `box [on | off]`
- Perspektive ändern: `view(azimuth,elevation)`
- Farben:
 - Farbtabelle: `colormap(name)`
 - Skalierung: `caxis(farbe_min, farbe_max)`
- Online-Hilfe: `graph2d, graph3d, specgraph`



MATLAB Eingabe und Ausgabe

Importieren, exportieren und drucken von Graphiken

- Figure – drucken: `print -fnummer`
– speichern: `print -fnummer -ddevice datei`
`saveas(h,'datei','fig')`
- Ausgabeoptionen (Format, Treiber): `-ddevice`
Formate: PS, EPS, TIFF, HPGL, JPEG, ...
Windows: EMF, BMP, Druckertreiber, Clipboard
- Graphik: – einlesen: `A = imread(datei,format)`
– erzeugen: `imwrite(A,datei,format)`
– plotten: `image(A)`



MATLAB Eingabe und Ausgabe

GUI – Graphical User Interface

- Graphische Benutzerschnittstelle
- Volle Ausnutzung der Objekt–Eigenschaften von MATLAB–Graphiken
- Programmieren mittels GUIDE oder von Hand

