

Programmieren mit MATLAB

MATLAB (Matrix Laboratory) ist ein Programmpaket von The Math Works, Inc., welches die numerische Lösung nicht nur mathematischer Testaufgaben sondern großer Anwendungsprobleme erlaubt. Neuerdings ist auch das symbolische Rechnen möglich.

In MATLAB kann interaktiv gearbeitet werden oder aber alle Befehle werden in einem File, welches auf .m endet, abgespeichert und dann übersetzt.

Beispiele:

<...>% matlab-13	Programmstart für Release 13 (evtl. nur matlab)
>>more on	Text wird nur seitenweise ausgegeben
>>help	Online-Hilfe
>>2*6	gibt das Ergebnis 12
>>a=3;	weist a den Wert 3 zu
>>b=a+2;	weist b den Wert $a + 3$, also 5 zu
>>b	gibt den Wert von b aus
>>A=[1 0;0 1];	weist A die Einheitsmatrix im \mathbb{R}^2 zu
>>y=[2;3];	weist y den entsprechenden Spaltenvektor zu
>>x=A\b	weist x die Lösung von $Ax = y$ zu und gibt diese aus
>>fplot('sin(x)',[0 2*pi])	zeichnet die Sinus-Funktion im Intervall $[0, 2\pi]$
>>u=[1:0.5:10];	weist u den Zeilenvektor $(1, 1.5, \dots, 10)$ zu
>>u(5)	gibt die 5. Komponente von u aus
	<i>Achtung: In MATLAB beginnen Indizes bei 1!</i>
>>v=u.*u;	weist v den Vektor $(1, 2.25, 4, \dots, 100)$ zu
>>plot(u,v)	zeichnet den Polygonzug durch die Punkte (u_i, v_i)
>>hold on	hält das aktuelle Bild an
>>plot(u,v,'or')	zeichnet kleine rote Kreise um die Punkte (u_i, v_i)
>>f=inline('2*z1^2+z2^2','z1','z2')	definiert die Funktion $f(z_1, z_2) = 2z_1^2 + z_2^2$
>>f(1,1)	gibt den Wert $f(1, 1) = 3$ aus
>>ezsurf(f)	zeichnet die Funktion $f = f(z_1, z_2)$
>>help ode45	ruft die Hilfe-Seiten für den Befehl ode45 auf
>>clear all	löscht alle Variablen
>>close all	schließt alle Bilder

Infoblatt # 5

Anfangswertprobleme für gewöhnliche Differentialgleichungen

Sind nicht alle möglichen Lösungen einer Differentialgleichung (DGL) gesucht, sondern nur solche, die für $t = 0$ einer gewissen Bedingung genügen, so spricht man von einem **Anfangswertproblem (AWP)**.

Beispiel: Gesucht sind alle Funktionen $\begin{cases} S = S(t) \\ C = C(t) \end{cases}$, die dem System gewöhnlicher DGL $\begin{cases} \dot{S} = C \\ \dot{C} = -S \end{cases}$ und den Anfangsbedingungen $\begin{cases} S(0) = 1 \\ C(0) = 2 \end{cases}$ genügen. Da mit $\begin{cases} S(t) = a \sin t + b \cos t \\ C(t) = -b \sin t + a \cos t \end{cases}$ ($a, b \in \mathbb{R}$) alle Lösungen des DGL-Systems gegeben sind, folgt wegen der Anfangsbedingungen $\begin{cases} S(0) = b \stackrel{!}{=} 1 \\ C(0) = a \stackrel{!}{=} 2 \end{cases}$, und die Lösung des AWP lautet $\begin{cases} S(t) = 2 \sin t + \cos t \\ C(t) = -\sin t + 2 \cos t \end{cases}$.

In MATLAB sind verschiedene Verfahren zur numerischen Lösung implementiert.

Das DGL-System muß in einem extra **m-File** definiert werden:

Beispiel: File `syst.m`

```
function upunkt = syst(t,u)
%%
a = u(2) ; b = - u(1) ;
upunkt=[a b]';
```

Eine Näherungslösung bekommen wir z. B. mit dem Befehl `ode45`:

Beispiel: File `awp.m`

```
clear all; close all
TT = [0 10] ;                               %% Zeitintervall
u0 = [1 2] ;                                 %% Anfangsbedingungen
[T,U] = ode45('syst',TT,u0) ;               %% numerische Lösung
S = U(:,1) ; C = U(:,2) ;
plot(T,S,'b',T,C,'r')                       %% Ausgabe (blau und rot)
```

