



Universität Konstanz  
FB Mathematik & Statistik  
Prof. Dr. M. Junk  
J. Budday

## Matlab-Routinen für DGL-Verfahren

Die Matlab-Routine eines Verfahrens zur Bestimmung einer numerischen Lösung einer DGL sollte wie folgt konzipiert werden:

```
function L = Name (R,I)

% description: kurze Beschreibung der function
% input: Beschreibung der input-Argumente R und I
% output: Beschreibung des output-Arguments L
% author: Name des Authors
```

### Code

Erklärung:

$R$ ,  $I$  und  $L$  sind Strukturvariablen, d.h. jeweils eine Liste von verschiedenen mathematischen Objekten, die in direktem Zusammenhang zur DGL bzw. zum verwendeten Verfahren stehen. Durch Verwendung solcher Strukturvariablen gelingt ein übersichtlicher und einheitlicher Aufbau Ihrer Matlab-Routinen. Der Vorteil dieser Vorgehensweise liegt u.a. darin, dass die Reihenfolge der Parameter weder bei den Argumenten noch bei den Rückgabewerten eine Rolle spielt, da immer komplette Listen von Objekten übergeben werden. Dadurch bleiben die Routinen offen für weitere Optionen und sind leicht erweiter- und aufrufbar. Machen Sie sich zunächst in Matlab mit Strukturvariablen vertraut, bevor Sie die Anleitung zur genauen Art und Weise der Strukturierung der Matlab-Routinen durchlesen. In einer Strukturvariable  $S$  lassen sich Objekte wie Skalare, Matrizen, Wörter, Mengen, Funktionen etc. gleichzeitig abspeichern. Geben Sie dazu in Matlab zum Beispiel folgendes ein:

```
S.a = 15;
S.b = 'hallo';
S.c = @(t,x) x.^2 - 3 + t;
S.d = {pi, 'r'};
```

Geben Sie nun  $S$  über das Command-Window aus. Rufen Sie die einzelnen Einträge getrennt voneinander auf, testen Sie die Funktionsauswertung beispielsweise durch  $S.c(1,2)$  und testen Sie die Eingabe  $S.d(2)$ . Damit sollte klar sein, wie man mit Strukturvariablen umgeht.

zur Struktur:

In  $R$  sollen alle Informationen zur rechten Seite der DGL gespeichert werden, die das numerische Verfahren für eine schnelle und problemangepasste Berechnung der Approximationen benötigt. Das wird zunächst einmal **verpflichtend** nur die rechte Seite als Funktion sein, die dann unter  $R.F$  angelegt wird. Bei impliziten Verfahren muss man verfahrensintern ja zunächst eine Fixpunktiteration durchführen, um überhaupt die nächste Approximation berechnen zu können. Je nach Art der rechten

Seite bieten sich hier unterschiedlich aufwändige Methoden an. Im Falle einer linearen DGL muss man zum Beispiel keine Fixpunktiteration anwerfen, da man lediglich ein LGS zu lösen hat. Deshalb ist es sinnvoll,  $R$  für weitere **optionale** Objekte oder Eigenschaften offen zu halten, wie zum Beispiel  $R.\text{lin} = \text{'true'}$ ,  $R.A = \text{Matrix}$ ,  $R.b = \text{Inhomogenität}$ ,  $R.dF = \text{Jacobi-Matrix}$  oder  $R.D = \text{Definitions Menge}$ . So kann man bei Bedarf immer einfach ein weiteres Feld in der Strukturvariable anlegen.

In  $I$  werden Informationen gespeichert, die das numerische Verfahren direkt betreffen. Dies werden zunächst einmal **verpflichtend** die Information bezüglich der Dimension  $d$  des Problems, die Information bezüglich des Startwertes und die bezüglich des Zeitgitters sein, die dann unter  $I.d$ ,  $I.xstart \in \mathbb{R}^{d \times 1}$  und  $I.grid \in \mathbb{R}^{1 \times m}$  angelegt werden, wobei  $m - 1$  für die Anzahl der Verfahrensschritte steht.  $I.grid$  enthält aufsteigend sortiert den gewünschten Startzeitpunkt, den gewünschten Endzeitpunkt und das gewünschte Zeitgitter für die Diskretisierung zwischen Start- und Endzeitpunkt. **Optional** können hier auch weitere Informationen zur Schrittweite zum Beispiel unter  $I.h$  angelegt werden.

$L$  ist eine Strukturvariable, in der **verpflichtend** das letztendlich verwendete Zeitgitter und die zugehörigen numerischen Approximationen unter  $L.grid$  und  $L.x$  angelegt werden.

Dabei ist noch folgende Konvention zu beachten: Die Argumente  $(t, x)$  der rechten Seite  $F$  betrachten wir stets als Objekte  $t \in \mathbb{R}$  und  $x \in \mathbb{R}^{d \times 1}$ . Somit ist auch  $F(t, x)$  ein  $d$ -dimensionaler Spaltenvektor, d.h. es gilt  $F(t, x) \in \mathbb{R}^{d \times 1}$ . Da wir in  $I$  den Startwert und das Gitter anlegen, muss die Routine so aufgebaut sein, dass sie nicht nur einen Schritt des Verfahrens berechnet, sondern Approximationen über dem gesamten gewünschten Gitter. Die Ausgabe  $L.x$  wird also eine  $\mathbb{R}^{d \times m}$ -Matrix sein, in deren ersten Spalte der Startwert und in den restlichen  $m - 1$  Spalten die entsprechenden Approximationen stehen.