TU Chemnitz – Professur Numerische Mathematik
Prof. Dr. Oliver Ernst
Dipl.-Math. Björn Sprungk

8. Dezember 2014

Numerik gewöhnlicher Differentialgleichungen

Wintersemester 2014/15

4. Übung: MATLAB-Routinen

Aufgabe 1

In MATLAB gibt es ein ganzes Paket von m-Files zur Lösung von Anfangswertproblemen

$$\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y}), \quad \boldsymbol{y}(t_0) = \boldsymbol{y}_0, \tag{1}$$

für gewöhnliche Differentialgleichungen. Sie werden durch den Befehl

```
[t,y]=ode_solver(ode_fun,t_span,y_0)
```

aufgerufen. Dabei bezeichnen

odesolver den Namen des Lösers (zur Zeit stehen `ode45`, `ode23`, `ode113`, `ode15s`, `ode23s`, `ode23t` und `ode23tb` zur Auswahl; hinter den Namen verbergen sich Algorithmen, die im Verlauf der Vorlesung behandelt werden),

t den Vektor der Zeitpunkte, an denen die Lösung berechnet wurde,

y eine Matrix, deren Zeilen die Komponenten der Lösung zu den Zeitpunkten aus t enthalten,

ode_fun den Namen einer ODE-Funktion; `ode_fun(t,y)` muss den Spaltenvektor $\boldsymbol{f}(t, \boldsymbol{y})$ liefern,

t_span das Lösungsintervall$[t_0, t_{\text{end}}]$,

y_0 den Vektor der Anfangsbedingungen.

Zur Lösung von (1) mit MATLAB muss also lediglich eine ODE-Funktion geschrieben werden, die das Anfangswertproblem beschreibt. (Daneben ist es sinnvoll, sogenannte ODE-Optionen zu setzen, vgl. `odeset`.)

Lösen Sie das restringierte Dreikörperproblem und die Gleichung der autokatalytischen Reaktion aus Kapitel 1.7 der Vorlesung mit sämtlichen ODE-Routinen von MATLAB und stellen Sie fest, welche Routine das jeweilige Problem am schnellsten löst.

Aufgabe 2

Ein Hund ist fatalerweise ans falsche Flussufer gekommen (im Punkt $P = [0, 5]$) und möchte gerne zu seinem Herrchen, der sich im Nullpunkt langweilt, zurück schwimmen. Der Hund schwimme dabei mit einer Geschwindigkeit $v = 2$ und der Fluss habe die Strömungsgeschwindigkeit $c = 1$. Da der Hund etwas dümmlich ist, schwimmt er immer genau auf sein Herrchen zu.

(a) Nutzen Sie die Event-Option der ODE-Optionen um den Zeitpunkt zu bestimmen zu dem der Hund bei seinem Herrchen ankommt (bzw. einen Abstand kleiner als 10^{-14} hat).

(b) Erweitern Sie Ihre Eventfunktion um mittels einer Simulation sowohl die Ankunftszeit als auch den Zeitpunkt (und Ort) zu bestimmen, zu welchem der Hund die Distanz zu seinem Herrchen halbiert hat.

Eine Eventfunktion ist eine Funktion der Bauart

```
[value,isterminal,direction] = events(t,y)
```

wobei

t,y den aktuellen Zeitpunkt und Zustand des dynamischen Systems bezeichnet,

value der Wert der Eventfunktion ist,

isterminal angibt, ob die Integration bzw. Simulation stoppen soll oder nicht, falls $\texttt{value} = 0$ (also $\texttt{isterminal} = 1$ oder 0),

direction die interessante Richtung des Durchganges von `value` durch 0 bezeichnet (also $\texttt{direction} = -1$ für $\texttt{value} \downarrow 0$, $\texttt{direction} = 1$ für $\texttt{value} \uparrow 0$ und $\texttt{direction} = 0$ für beide Richtungen).

Der entsprechende Auruf des ODE-Solver lautet dann

```
[t,y,te,ye,ie]=ode_solver(ode_fun,t_span,y_0, options)
```

und

te gibt die Zeitpunkte des Eintretens der Ereignisse an,

ye ist eine Matrix, welche die Lösung zu den Zeitpunkten `te` beinhaltet,

ie ist ein Vektor, der indiziert, welches Event eingetreten ist.