

# Explizites Eulerverfahren

## Anfangswertaufgabe für gewöhnliche Differentialgleichung

Gegeben  $t_0 < T$ ,  $f : [t_0, T] \times \mathbb{R} \rightarrow \mathbb{R}$ ,  $y_0 \in \mathbb{R}$ .

Gesucht  $y : [t_0, T] \rightarrow \mathbb{R}$ , so dass

$$y'(t) = f(t, y(t)) \quad (t \in [t_0, T])$$

$$y(t_0) = y_0$$

## Näherungsverfahren

Seien  $t_0 < t_1 < \dots < t_N = T$ ,  $h_i = t_{i+1} - t_i$  ( $i=0, \dots, N-1$ )

$$\frac{y_{i+1} - y_i}{h_i} = f(t_i, y_i) \quad (i=0, \dots, N-1) \quad d.h.$$

$$y_{i+1} = y_i + h_i f(t_i, y_i) \quad (i=0, \dots, N-1)$$

*Schreibweise mit Zuwächsen*

$$\left. \begin{aligned} k_i &= f(t_i, y_i) \\ y_{i+1} &= y_i + h_i k_i \end{aligned} \right\} i = 0, \dots, N-1$$

Das explizite Eulerverfahren ist das einfachste Einschrittverfahren zur numerischen Lösung gew. Dgln. Zu dieser Klasse gehören auch die Runge-Kutta-Verfahren.

# Anmerkungen zur Übungsaufgabe

## Allgemeines

- ▶ Das Nyström-Verfahren verwandt mit dem klassischen Runge-Kutta-Verfahren 4. Ordnung für Anfangswertaufgaben von Systemen gewöhnlicher Differentialgleichungen.
- ▶ Schrittweitensteuerung ist in der Regel für Genauigkeit und Effizienz essentiell, soll hier aber nicht programmiert werden.

## Beispiele

- (a) Lösung:  $y_1(t) = \sin(t) + \frac{1}{3} \sin(3t)$ ,  $y_2(t) = \sin(t) - \frac{1}{3} \sin(3t)$ .
- (b) Daten sind im heliozentrischen kartesischen Äquatorialsystem angegeben. (Ursprung: Sonne,  $z$  – Achse senkrecht zur Äquatorebene,  $x$  – Achse liegt in der Äquatorebene und zeigt zum Frühlingspunkt)  
Testmöglichkeiten: Umlaufzeiten der Planeten, Erde zu Frühlings- und Herbstbeginn nahe der  $x$ -Achse

# Anmerkungen zur Übungsaufgabe - Fortsetzung

## Hinweise zur Programmierung

- ▶ Verwendung von Valarrays und Vektoroperationen vereinfacht die Programmierung deutlich.
- ▶ Gute Strukturierung anstreben: Lösungsverfahren mit Ausgabe, Nyströmeinzelschritt und Daten trennen (Ziel: Daten, Einzelschritt und Gesamtmethode sollten möglichst unabhängig voneinander austauschbar sein. Erleichtert Tests und Validierung.)
- ▶ Strukturierung mit Templates komplizierter, allerdings übersichtlicher, z.B.: Rechte Seiten, Nystroemschritt und Ausgabefunktion als Funktionsobjekte, Verfahren als Templatefunktion
- ▶ Unterschiedliche Ansätze sind möglich: Allgemeinheit, Effizienz und einfach benutzbare Schnittstellen sind anscheinend nicht leicht zu erreichen.
- ▶ Weiterführend: Numerical Recipes in C++, boost::odeint ([www.boost.org](http://www.boost.org))