

# Differentialgleichungen

Die Lösung von Differentialgleichungen (DGLs) ist Thema der nun folgenden drei Vorlesungen. Dabei wenden wir uns heute dem einfachsten Fall zu - der Lösung von gewöhnlichen Differentialgleichungen erster Ordnung. DGLs höher Ordnung und partielle DGLs folgen dann in den kommenden Vorlesungen.

## Gewöhnliche Differentialgleichungen (erster Ordnung)

Den ersten Fall, den wir im folgenden betrachten wollen, sind Aufangswertprobleme für gewöhnliche DGLs erster Ordnung der Form

$$\begin{aligned} \frac{dy}{dt} &= f(y, t) \\ y(t_0) &= y_0 \end{aligned}$$

wo also neben dem Anfangswert  $y_0$  zur Zeit  $t_0$  die Zeitableitung  $f(y, t)$  gegeben sei. Gesucht ist nun die zeitliche Entwicklung  $y(t)$ .

Zeichnungen dieser Art modellieren etwa den radioaktiven Zerfall einer Materialmenge

$$\frac{dN}{dt} = -\lambda N$$

$$N(t=0) = N_0$$

oder das Abkühlen eines Kaffees der Temperatur  $T$

$$\frac{dT}{dt} = -\gamma (T - T_{\text{Raum}}), \quad T(t=0) = T_0$$

Wobei also  $T$  die Temperatur des Kaffees,  $T_{\text{Raum}}$  die umgebende Raumtemperatur und  $\gamma$  eine Kühlrate seien.

Für diese einfachen Probleme können analytische Lösungen durch Neuarrangieren der DGL gefunden werden

$$\rightarrow \frac{dT}{T - T_{\text{Raum}}} = -\gamma dt$$

Integration beider Seiten

$$\int_{T(0)}^{T(t)} \frac{dT}{T - T_{\text{Raum}}} = -\gamma \int_0^t dt$$

ergibt somit evaluiert bei den Integralgrenzen

$$\ln(T(t) - T_{\text{Raum}}) - \ln(T(0) - T_{\text{Raum}}) = -\gamma t$$

was aufgelöst nach  $T(t)$  ergibt

$$T(t) = T_{\text{Raum}} + (T(0) - T_{\text{Raum}}) e^{-\gamma t}$$

Während in diesem Beispiel die beiden Hauptschritte, nämlich die Evaluation des Integrals und das Auflösen der sich ergebenden Gleichung, analytisch einfach zu bewerkstelligen sind, ist dies im allgemeinen Fall nicht anzunehmen.

## Die Euler - Methode

-3-

Numerisch können wir den Wert der gesuchten Funktion  $y$  zu einem späteren Zeitpunkt  $t + \Delta t$  annähern wie eine Taylor - Entwicklung erster Ordnung

$$y(t_0 + \Delta t) = y(t_0) + \Delta t \frac{dy}{dt} = y_0 + \Delta t \cdot f(y_0, t_0) + O(\Delta t^2)$$

Wenn wir diese Gleichung iterieren erhalten wir mit der Notation  $t_n = t_0 + n \cdot \Delta t$  und  $y_n(t_n)$  den sogenannten Euler - Algorithmus

$$y_{n+1} = y_n + \Delta t \cdot f(y_n, t_n) + O(\Delta t^2)$$

Die Euler - Methode stellt also ein relativ einfaches Verfahren dar, welches sich auch auf beliebig komplexe DGLs anwenden lässt.

Die Beschränkung der Taylor - Entwicklung nach dem ersten Term in der Euler - Methode führt zu einem Fehler  $O(\Delta t^2)$  bei jedem Zeitschritt. In unseren Simulationen müssen wir diesen Fehler unter Kontrolle halten, um zu garantieren, daß das Endergebnis nicht durch diesen endlichen Zeitschritt beeinflusst wird.

Dazu haben wir im allgemeinen zwei Optionen: i) Wir können einen immer kleineren Zeitschritt  $\Delta t$  wählen oder ii) wir wenden eine Methode höherer Ordnung an.

Allgemein gilt: Eine Methode, welche einen Fehler von  $O(\Delta t^n)$  in jedem einzelnen Zeitschritt produziert, ist lokal von  $n$ -ter Ordnung. Wird diese Methode nun über ein fixes Zeitintervall  $T$  iteriert, addieren sich diese Fehler auf:

Speziell müssen wir  $T/\Delta t$  Zeitschritte absolvieren, bei jedem einzelnen addiert sich dabei ein Fehler von  $O(\Delta t^u)$ . Der gesamte, in der Zeit  $T$  akkumulierte Fehler ist dann

$$\frac{T}{\Delta t} \cdot O(\Delta t^u) = O(\Delta t^{u-1}),$$

d.h. die Methode ist global von  $(u-1)$ -ter Ordnung.

Für die Euler-Methode, welche lokal von 2. Ordnung  $O(\Delta t^2)$  ist, bedeutet dies eine globale 1. Ordnung  $O(\Delta t)$ .

### Runge-Kutta Methoden

Die Runge-Kutta Methoden sind Familien von Methoden, die systematisch höhere Ordnungen gegenüber der Euler-Methode beinhalten und damit systematisch bessere Approximationen liefern.

Die Kernidee ist, die Ableitung  $\frac{dy}{dt}$  nicht nur an den Endpunkten  $t_n$  und  $t_{n+1}$ , sondern auch an Zwischenpunkten eines gegebenen Intervalls  $[t_n, t_{n+1}]$  zu berechnen, etwa:

$$y_{n+1} = y_n + \Delta t \cdot f\left(t_n + \frac{\Delta t}{2}, y\left(t_n + \frac{\Delta t}{2}\right)\right) + O(\Delta t^3)$$

Die zunächst nicht bekannte Lösung  $y\left(t_n + \frac{\Delta t}{2}\right)$  wird hierbei wiederum durch einen einfachen Euler-Schritt gegeben, so daß man ein Runge-Kutta Verfahren zweiter Ordnung erhält:

$$\begin{aligned} k_1 &= \Delta t \cdot f(t_n, y_n) \\ k_2 &= \Delta t \cdot f\left(t_n + \frac{\Delta t}{2}, y_n + \frac{k_1}{2}\right) \\ y_{n+1} &= y_n + k_2 + O(\Delta t^3) \end{aligned}$$

Der allgemeine Ansatz ist

-5.

$$y_{u+1} = y_u + \sum_{i=1}^N \alpha_i k_i$$

wobei die Approximationen  $k_i$  gegeben sind durch

$$k_i = \Delta t \cdot f\left(y_u + \sum_{j=1}^{N-1} v_{ij} k_j, t_u + \sum_{j=1}^{N-1} v_{ij} \Delta t\right)$$

und die Parameter  $\alpha_i$  und  $v_{ij}$  so gewählt werden, daß eine Methode  $N$ -ter Ordnung entsteht. Beachte dabei, daß diese Wahl typischerweise nicht eindeutig ist.

Der am häufigsten benutzte Runge-Kutta Algorithmus ist die Methode 4. Ordnung:

$$k_1 = \Delta t \cdot f(t_u, y_u)$$

$$k_2 = \Delta t \cdot f\left(t_u + \frac{\Delta t}{2}, y_u + \frac{k_1}{2}\right)$$

$$k_3 = \Delta t \cdot f\left(t_u + \frac{\Delta t}{2}, y_u + \frac{k_2}{2}\right)$$

$$k_4 = \Delta t \cdot f(t_u + \Delta t, y_u + k_3)$$

$$y_{u+1} = y_u + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(\Delta t^5),$$

in welchem zwei Abschätzungen am Zwischenpunkt  $t_u + \frac{\Delta t}{2}$  kombiniert werden mit jeweils einer Abschätzung am Startpunkt  $t_u$  und Endpunkt  $t_{u+1}$ .

# Integration der klassischen Bewegungsgleichungen

-6

Wir wollen uns jetzt dem wohl wichtigsten Beispiel für das Auftreten von gewöhnlichen DGLs in der Physik widmen: Newtons Gleichungen für die klassische Bewegung von  $N$  Partikeln

$$m_i \frac{d\vec{v}_i}{dt} = \vec{F}_i(t, \vec{x}_1, \dots, \vec{x}_N, \vec{v}_1, \dots, \vec{v}_N)$$

$$\frac{d\vec{x}_i}{dt} = \vec{v}_i$$

wobei  $m_i$ ,  $\vec{v}_i$  und  $\vec{x}_i$  die Masse, Geschwindigkeit und Ortsraumkoordinate des  $i$ -ten Teilchens seien und  $\vec{F}_i$  die Kraft, welche auf dieses Teilchen wirkt.

Zunächst betrachten wir den Fall eines einzelnen Teilchens in einer Raumdimension (und wenden uns den allgemeineren Fällen später zu).

Notation:  $t_{u+1} = t_u + \Delta t$        $x_u = x(t_u)$        $v_u = v(t_u)$   
 $a_u = a(t_u, x_u, v_u) = F(t_u, x_u, v_u) / m$

Die einfachste Methode, die Bewegungsgleichungen zu integrieren, ist die "forward" Euler-Methode:

$v_{u+1} = v_u + a_u \cdot \Delta t$	forward
$x_{u+1} = x_u + v_u \cdot \Delta t$	Euler

→ instabil für oszillierende Systeme, siehe Übung.

Für geschwindigkeitsunabhängige Kräfte  $\vec{F} = \vec{F}(t_n, x_n)$  reicht ein überraschend einfacher Trick, um die Euler-Methode zu stabilisieren: Statt der "forward" Differenz  $v_n \approx (x_{n+1} - x_n) / \Delta t$ , benutzen wir die "backward" Differenz  $v_{n+1} \approx (x_{n+1} - x_n) / \Delta t$  und erhalten

$$\begin{aligned}
 v_{n+1} &= v_n + a_n \cdot \Delta t \\
 x_{n+1} &= x_n + \underline{v_{n+1}} \cdot \Delta t
 \end{aligned}$$

backward Euler  
(Euler-Cromer Methode)

wo also die neu errechnete Geschwindigkeit  $v_{n+1}$  benutzt wird in der Berechnung von  $x_{n+1}$ .

Eine eng verwandte Methode benutzt den Mittelpunkt:

$$\begin{aligned}
 v_{n+1} &= v_n + a_n \cdot \Delta t \\
 x_{n+1} &= x_n + \frac{1}{2} (v_n + v_{n+1}) \Delta t
 \end{aligned}$$

Ähnlich einfach ist die sogenannte "leap frog" Methode, eine der am häufigsten verwendeten Methoden, welche Orte und Geschwindigkeiten bei unterschiedlichen Zeiten berechnet:

$$\begin{aligned}
 v_{n+\frac{1}{2}} &= v_{n-\frac{1}{2}} + a_n \cdot \Delta t \\
 x_{n+1} &= x_n + v_{n+\frac{1}{2}} \cdot \Delta t \\
 \text{mit } v_{\frac{1}{2}} &= v_0 + \frac{1}{2} a_0 \cdot \Delta t
 \end{aligned}$$

leapfrog Methode

Für geschwindigkeitsabhängige Kräfte wird der Euler-Richardson Algorithmus verwendet:

Euler-  
Richardson  
Methode

$$a_{n+\frac{1}{2}} = a\left(x_n + \frac{1}{2}v_n \cdot \Delta t, v_n + \frac{1}{2}a_n \cdot \Delta t, t_n + \frac{1}{2} \Delta t\right)$$

$$v_{n+1} = v_n + a_{n+\frac{1}{2}} \cdot \Delta t$$

$$x_{n+1} = x_n + v_n \cdot \Delta t + \frac{1}{2} a_{n+\frac{1}{2}} \Delta t^2$$

Der in der Praxis am häufigsten verwendete Algorithmus ist der Verlet - Algorithmus:

$$x_{n+1} = x_n + v_n \cdot \Delta t + \frac{1}{2} a_n (\Delta t)^2$$

$$v_{n+1} = v_n + \frac{1}{2} (a_n + a_{n+1}) \Delta t$$

Verlet -  
Algorithmus