Gewöhnliche Differentialgleichungen

Jonas Pleyer

11. Mai 2022

Table of Contents

Wiederholung ODEs

- Beschreibt Änderung einer/mehrerer Variable/n

Wiederholung ODEs

- Beschreibt Änderung einer/mehrerer Variable/n
- ... welche von sich selbst oder anderen Variablen abhängt/abhängen

Wiederholung ODEs

- Beschreibt Änderung einer/mehrerer Variable/n
- ... welche von sich selbst oder anderen Variablen abhängt/abhängen
- Generelle Form:

$$\dot{A} = f(A, t)$$

Wiederholung ODEs

- Beschreibt Änderung einer/mehrerer Variable/n
- ... welche von sich selbst oder anderen Variablen abhängt/abhängen
- Generelle Form:

$$\dot{A} = f(A, t)$$

- Linke Seite: Änderung von A
  Rechte Seite: Funktion f abhängig von A

Wiederholung ODEs

- Beschreibt Änderung einer/mehrerer Variable/n
- ... welche von sich selbst oder anderen Variablen abhängt/abhängen
- Generelle Form:

$$\dot{A} = f(A, t)$$

- Linke Seite: Änderung von A
  Rechte Seite: Funktion f abhängig von A
-

Beispiele

- Corona-Neu-Infizierungungen (Änderung) sind proportional zu der Anzahl der bereits infizierten Menschen

$$\dot{N} = aN$$

Beispiele

- Corona-Neu-Infizierungungen (Änderung) sind proportional zu der Anzahl der bereits infizierten Menschen

$$\dot{N} = aN$$

$\Rightarrow$ Exponentielles Wachstum $N = N_0 \exp(at)$

Beispiele

- Corona-Neu-Infizierungungen (Änderung) sind proportional zu der Anzahl der bereits infizierten Menschen

$$\dot{N} = aN$$

$\Rightarrow$ Exponentielles Wachstum $N = N_0 \exp(at)$

- Radioaktiver Zerfall: Menge des zerfallenden Materials (Änderung) ist proportional zur Gesamtmenge

$$\dot{N} = -\lambda N$$

Beispiele

- Corona-Neu-Infizierungungen (Änderung) sind proportional zu der Anzahl der bereits infizierten Menschen

$$\dot{N} = aN$$

$\Rightarrow$ Exponentielles Wachstum $N = N_0 \exp(at)$

- Radioaktiver Zerfall: Menge des zerfallenden Materials (Änderung) ist proportional zur Gesamtmenge

$$\dot{N} = -\lambda N$$

$\Rightarrow$ Exponentieller Zerfall: $N = N_0 \exp(-\lambda t)$

Numerische Lösungsverfahren - Euler

Gegeben sei eine Differentialgleichung

$$\dot{A} = f(A, t) \qquad A(0) = A_0$$

Idee: berechne mit Differenzenquotienten neue Werte

$$\lim_{h \to 0} \frac{A(t+h) - A(t)}{h} = f(A(t), t)$$

Teile Zeitinterval auf in n Zeitschritte dt.

$$\frac{A((n+1) \cdot dt) - A(n \cdot dt)}{dt} = f(A(n \cdot dt), t)$$

Umstellen nach $A((n+1) \cdot dt) =: A_{n+1}$ (Kurzschreibweise)

$$A_{n+1} = A_n + dt \cdot f(A_n, n \cdot dt)$$

Numerische Lösungsverfahren - Euler

Zusammenfassung Verfahren: Explizites Euler-Verfahren für Gewöhnliche Differentialgleichungen:

- Solange $n \cdot dt < t_{max}$, berechne:

$$A_{n+1} = A_n + dt \cdot f(A_n, n \cdot dt)$$

- Gesamtergebnis sind einzelne Werte der Funktion A in dem Zeitinterval $I = [t_0, t_{max}]$

$$I = [0, dt, 2dt, \dots, N \cdot dt]$$

$$A = [A_0, A_1, A_2, \dots, A_N]$$

Lösungen für ODEs

Lösungen für ODEs

```
import numpy as np
import matplotlib.pyplot as plt
```

Lösungen für ODEs

```
import numpy as np
import matplotlib.pyplot as plt

def A(y, t):
    return -0.5*y
```

Lösungen für ODEs

```
import numpy as np
import matplotlib.pyplot as plt

def A(y, t):
    return -0.5*y

if __name__ == "__main__":
    t0 = 0.0
    tend = 10.0
    dt = 1.0
    y0 = 4.0
```

Lösungen für ODEs

```
import numpy as np
import matplotlib.pyplot as plt

def A(y, t):
    return -0.5*y

if __name__ == "__main__":
    t0 = 0.0
    tend = 10.0
    dt = 1.0
    y0 = 4.0

    t_vals = np.arange(t0, tend, dt)
    y_vals = np.array([y0]*len(t_vals))
```

Lösungen für ODEs

```python
import numpy as np
import matplotlib.pyplot as plt

def A(y, t):
    return -0.5*y

if __name__ == "__main__":
    t0 = 0.0
    tend = 10.0
    dt = 1.0
    y0 = 4.0

    t_vals = np.arange(t0, tend, dt)
    y_vals = np.array([y0]*len(t_vals))

    # Hier wird das Eulersche Verfahren zum lösen der ODE verwendet
    for i in range(len(y_vals)-1):
        y_vals[i+1] = y_vals[i] + dt * RHS(t_vals[i], y_vals[i])
```

Lösungen für ODEs

```
import numpy as np
import matplotlib.pyplot as plt

def A(y, t):
    return -0.5*y

if __name__ == "__main__":
    t0 = 0.0
    tend = 10.0
    dt = 1.0
    y0 = 4.0

    t_vals = np.arange(t0, tend, dt)
    y_vals = np.array([y0]*len(t_vals))

    # Hier wird das Eulersche Verfahren zum lösen der ODE verwendet
    for i in range(len(y_vals)-1):
        y_vals[i+1] = y_vals[i] + dt * RHS(t_vals[i], y_vals[i])

    plt.plot(t_vals, y_vals, label="Lösung der Differentialgleichung", marker="o")
    plt.legend()
    plt.show()
```

Lösungen für ODEs

- Euler-Verfahren hat schwierigkeiten bei bestimmten ßteifenProblemen
- Deutlich robustere Lösungsmethoden wurden über die letzten Jahrhunderte entwickelt
- Wir brauchen nicht alle Details dazu kennen
- Gibt schon fertig entwickelte libraries, die uns die Arbeit übernehmen zB.

```
from scipy.integrate import odeint
```

Lösungen für ODEs

Lösungen für ODEs

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
```

Lösungen für ODEs

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

def RHS(y, t):
    return -0.5*y
```

Lösungen für ODEs

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

def RHS(y, t):
    return -0.5*y

if __name__ == "__main__":
    t0 = 0.0
    tend = 10.0
    dt = 1.0
    y0 = 4.0
```

Lösungen für ODEs

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

def RHS(y, t):
    return -0.5*y

if __name__ == "__main__":
    t0 = 0.0
    tend = 10.0
    dt = 1.0
    y0 = 4.0

    t_vals = np.arange(t0, tend, dt)
```

Lösungen für ODEs

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

def RHS(y, t):
    return -0.5*y

if __name__ == "__main__":
    t0 = 0.0
    tend = 10.0
    dt = 1.0
    y0 = 4.0

    t_vals = np.arange(t0, tend, dt)
    # Hier wird eine routine von scipy zum lösen der ODE verwendet
    y_vals = odeint(RHS, y0, t_vals)
```

Lösungen für ODEs

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

def RHS(y, t):
    return -0.5*y

if __name__ == "__main__":
    t0 = 0.0
    tend = 10.0
    dt = 1.0
    y0 = 4.0

    t_vals = np.arange(t0, tend, dt)
    # Hier wird eine routine von scipy zum lösen der ODE verwendet
    y_vals = odeint(RHS, y0, t_vals)

    plt.plot(t_vals, y_vals, label="Lösung der Differentialgleichung", marker="o")
    plt.legend()
    plt.show()
```

Lösungen für ODEs

Abbildung: Das Euler-Verfahren löst eine Ordinary Differential Equation (ODE) schrittweise mit konstanter Schrittweite.

Table of Contents

Aufgabe 1

- Löse die folgende ODE mit $A(0) = 2$ per Hand

$$\dot{A} = \alpha$$

Aufgabe 1

- Löse die folgende ODE mit $A(0) = 2$ per Hand

$$\dot{A} = \alpha$$

- Löse die folgende ODE mit $A(0) = N$ per Hand

$$\dot{A} = -\beta A$$

Aufgabe 1

- Löse die folgende ODE mit $A(0) = 2$ per Hand

$$\dot{A} = \alpha$$

- Löse die folgende ODE mit $A(0) = N$ per Hand

$$\dot{A} = -\beta A$$

- Welche Verhaltensweisen zeigen die Lösungen?

Aufgabe 1

- Löse die folgende ODE mit $A(0) = 2$ per Hand

$$\dot{A} = \alpha$$

- Löse die folgende ODE mit $A(0) = N$ per Hand

$$\dot{A} = -\beta A$$

- Welche Verhaltensweisen zeigen die Lösungen?
- Wie verändert sich die Lösung für verschiedene Parameter α, β?

Aufgabe 1

- Löse die folgende ODE mit $A(0) = 2$ per Hand

$$\dot{A} = \alpha$$

- Löse die folgende ODE mit $A(0) = N$ per Hand

$$\dot{A} = -\beta A$$

- Welche Verhaltensweisen zeigen die Lösungen?
- Wie verändert sich die Lösung für verschiedene Parameter α, β?
- Welches Biologische System könnten diese ODEs beschreiben?

Aufgabe 2

Gegeben ist die folgende Differentialgleichung

$$\dot{A} = \alpha - \beta A$$

mit $A(0) = 0$.

1. Löse diese Differentialgleichung numerisch mit dem Euler-Verfahren.

Aufgabe 2

Gegeben ist die folgende Differentialgleichung

$$\dot{A} = \alpha - \beta A$$

mit $A(0) = 0$.

1. Löse diese Differentialgleichung numerisch mit dem Euler-Verfahren.
2. Welche Eigenschaften haben die parameter α, β?

Aufgabe 2

Gegeben ist die folgende Differentialgleichung

$$\dot{A} = \alpha - \beta A$$

mit $A(0) = 0$.

1. Löse diese Differentialgleichung numerisch mit dem Euler-Verfahren.
2. Welche Eigenschaften haben die parameter α, β?
3. Welche Probleme können auftreten für verschiedene parameter α, β?

Aufgabe 2

Gegeben ist die folgende Differentialgleichung

$$\dot{A} = \alpha - \beta A$$

mit $A(0) = 0$.

1. Löse diese Differentialgleichung numerisch mit dem Euler-Verfahren.
2. Welche Eigenschaften haben die parameter α, β?
3. Welche Probleme können auftreten für verschiedene parameter α, β?
4. Welches biologische System könnte dieser Differentialgleichung zugrunde liegen?

Aufgabe 3

Wir betrachten wie eben die Differentialgleichung

$$\dot{A} = \alpha - \beta A$$

diesmal mit zeitabhängigem Parameter α

$$\alpha(t) = \begin{cases} \alpha & 2n < t \leq 2n+1 \\ 0 & \text{sonst} \end{cases}$$

- Wie können wir die Lösung dieser Differentialgleichung berechnen?

Table of Contents

Projektarbeit

- Arbeit in Gruppen (2-3 Gruppen)

Projektarbeit

- Arbeit in Gruppen (2-3 Gruppen)
- Erarbeiten von Model zu biologischem System

Projektarbeit

- Arbeit in Gruppen (2-3 Gruppen)
- Erarbeiten von Model zu biologischem System
- Übersetzen in Computer-Simulation

Projektarbeit

- Arbeit in Gruppen (2-3 Gruppen)
- Erarbeiten von Model zu biologischem System
- Übersetzen in Computer-Simulation
- Diskussion der Ergebnisse in Vortrag

Projektarbeit

- Arbeit in Gruppen (2-3 Gruppen)
- Erarbeiten von Model zu biologischem System
- Übersetzen in Computer-Simulation
- Diskussion der Ergebnisse in Vortrag
- Dauer ca. 3-4 Wochen

Projektarbeit

- Arbeit in Gruppen (2-3 Gruppen)
- Erarbeiten von Model zu biologischem System
- Übersetzen in Computer-Simulation
- Diskussion der Ergebnisse in Vortrag
- Dauer ca. 3-4 Wochen
- Beginn wahrscheinlich gegen Anfang Juni

Projektarbeit

- Arbeit in Gruppen (2-3 Gruppen)
- Erarbeiten von Model zu biologischem System
- Übersetzen in Computer-Simulation
- Diskussion der Ergebnisse in Vortrag
- Dauer ca. 3-4 Wochen
- Beginn wahrscheinlich gegen Anfang Juni
- Themen werden noch bekannt gegeben