

# Reaktionsnetzwerke II

Jonas Pleyer

27. Mai 2022

# Table of Contents

## 1. Wiederholung

# Table of Contents

- 1. Wiederholung
  - 1.1 Lösen von ODEs
  - 1.2 Zugehöriges Reaktionsnetzwerk

# Lösen von ODEs

Die folgende ODE beschreibt Protein-Synthese mit Degradation.

$$\dot{x} = f(x, t) = \alpha - \beta x$$

Hier sind  $\alpha$  die Synthetisierungsrate und  $\beta$  die Degradationsrate. Fragen:

- 1 Welche Einheiten haben  $\alpha, \beta$
- 2 Was erwarten wir für ein Verhalten? Warum?
- 3 Wie gehen die Anfangswerte ein?
- 4 Gibt es einen Gleichgewichtszustand? Ist er positiv?

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

def f(x, t, a, b):
    return a - b*x

if __name__ == "__main__":
    tstart = 0.0
    tend = 10.0
    y0 = 0.0
    a = 0.1
    b = 10.0

    t = np.linspace(tstart, tend)
    results = odeint(f, y0, (a, b))

    plt.plot(t, results, label="Ergebnisse der gelösten ODE")
    plt.legend()
    plt.show()
```

# Zugehöriges Reaktionsnetzwerk

Welche Reaktionen laufen in der eben gelösten ODE ab?

$$\dot{x} = f(x, t) = \alpha - \beta x$$

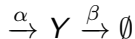
Erstellung von neuem Protein



Degradation von Protein



Insgesamt:



Wie lautet eine ODE zu einem gegebenen Reaktionsnetzwerk?



- Wie viele Komponenten haben wir?  
 $\Rightarrow$  3 Stück:  $A, B, C$
- Betrachte zunächst Gleichung 2. Was passiert hier?  
 $\Rightarrow B$  und  $C$  reagieren und werden vernichtet. Die Konzentration von  $B$  und  $C$  muss also kleiner werden.
- Mit welcher Rate passiert das?

$$\dot{B} = -\phi BC \quad (3)$$

$$\dot{C} = -\phi BC \quad (4)$$

Wie lautet eine ODE zu einem gegebenen Reaktionsnetzwerk?



- Bisher haben wir:

$$\dot{B} = -\phi BC \quad (7)$$

$$\dot{C} = -\phi BC \quad (8)$$

- Betrachte jetzt noch Gleichung 1  
2x Stoff A wird umgewandelt zu B
- Wie lautet die Reaktionsgleichung?

$$\dot{A} = -2\psi A^2 \quad (9)$$

$$\dot{B} = +2\psi A^2 \quad (10)$$

- Wie kombiniere ich nun diese Gleichungen?  
Einzelne Komponenten aufaddieren und dann alles zusammenschreiben



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

def f(y, t, k1, k2):
    return (-2*k1*y[0]**2, 2*k1*y[0]**2 - k2*y[1]*y[2], - k2*y[1]*y[2])

if __name__ == "__main__":
    tstart = 0.0
    tend = 10.0
    y0 = (1.0, 0.0, 0.5)
    k1 = 0.3
    k2 = 0.5
    t = np.linspace(tstart, tend)

    results = odeint(f, y0, t, (k1, k2))

    for i in range(results.shape[1]):
        plt.plot(t, results[:,i], label="Komponente " + str(i))
    plt.legend()
    plt.show()
```