

# Zwischenbilanz

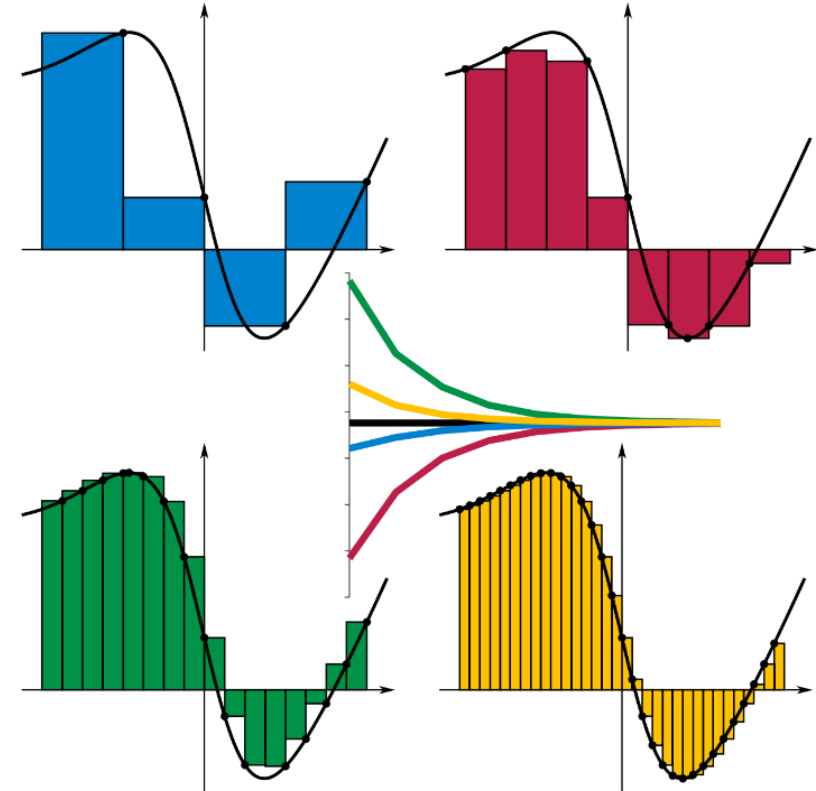
Numerische Grundlagen  
der Data Science

W7



# Inhalt

- Fragen zum Projekt
- Rekapitulation
  - Themenübersicht
  - Typen von Verfahren
- Best Practices
- Performance Profiling



[Riemann sum convergence.png](#), CC BY-SA 3.0, KSmrq

# Projekt: Fragen?

- ...

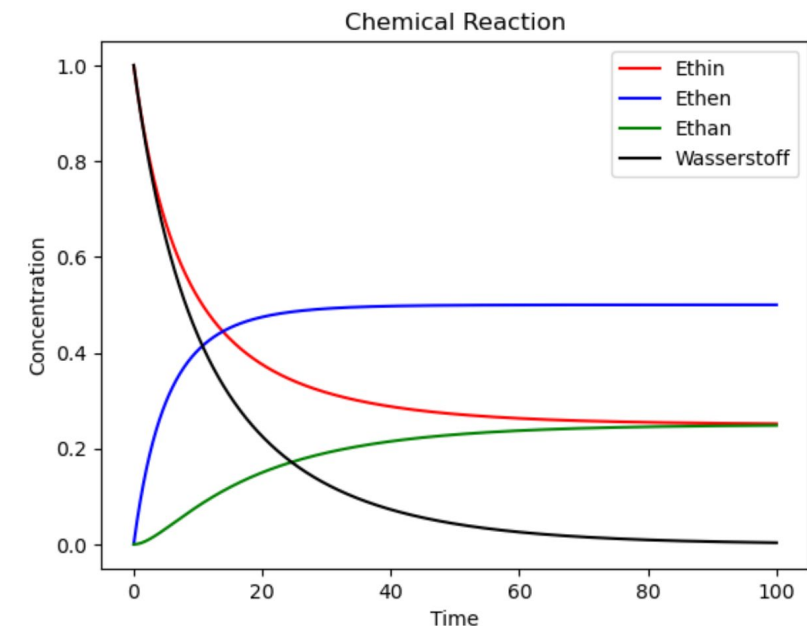
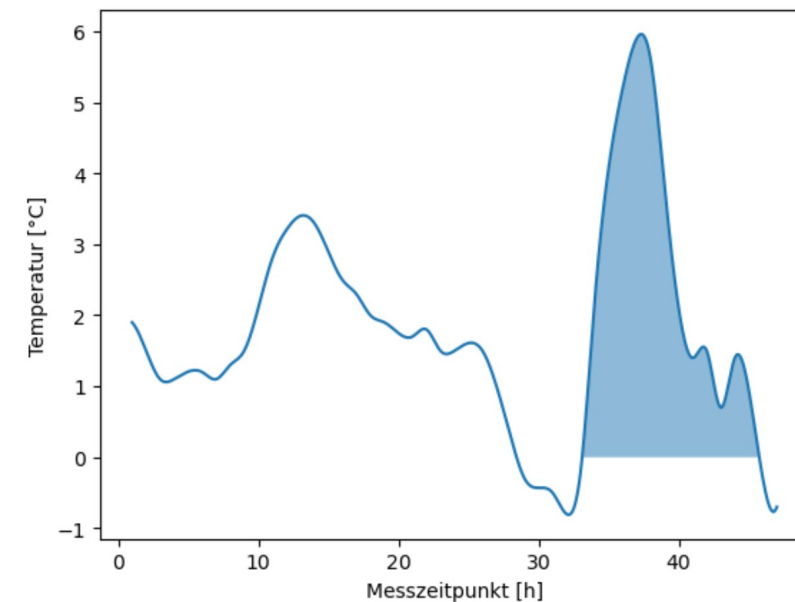
# Themenübersicht

- Grundlagen:
  - Arbeiten mit mehrdimensionalen Arrays
  - Lineare Algebra
  - Funktionen und Daten darstellen
- Zwischenwerte schätzen: Interpolation
- Gleichungen lösen: Nullstellen & Fixpunkte
- Ableitung in einer und mehreren Variablen
- Integration
- Gewöhnliche Differentialgleichungen
- Zufallszahlen simulieren

SW	KW	Tag	Datum	Thema
1	8	Do Fr	23. Feb 24. Feb	Einführung, Zahlendarstellung, Fehlertypen, Numpy
2	9	Do Fr	02. Mär 03. Mär	Darstellung von Daten und Funktionen in mehreren Variablen Isokonturlinien, Höhenlinien
3	10	Do Fr	09. Mär 10. Mär	Interpolation von Daten und numerischen Funktionen
4	11	Do Fr	16. Mär 17. Mär	Nullstellen und Fixpunkte
5	12	Do Fr	23. Mär 24. Mär	Ableitung in einer Variablen
6	13	Do Fr	30. Mär 31. Mär	Ableitungen in mehreren Variablen Darstellung Gradient, Gradient Descent
7	14	Do Fr	06. Apr 07. Apr	Iterationsverfahren, Konvergenz, Performance Karfreitag
8	15	Do Fr	13. Apr 14. Apr	Integration von Funktionen und Daten
9	16	Do Fr	20. Apr 21. Apr	Gewöhnliche Differentialgleichungen Euler-Verfahren
10	17	Do Fr	27. Apr 28. Apr	Zufallszahlen erzeugen
11	18	Do Fr	04. Mai 05. Mai	Projektwoche
12	19	Do Fr	11. Mai 12. Mai	Simulieren von Zufallsvariablen Samplen von Daten
13	20	Do Fr	18. Mai 19. Mai	Auffahrt Brücke
14	21	Do Fr	25. Mai 26. Mai	Projekt Präsentationen
15	22	Do Fr	01. Jun 02. Jun	Projekt Präsentationen Puffer

# Ausblick

- Grundlagen:
  - Arbeiten mit mehrdimensionalen Arrays
  - Lineare Algebra
  - Funktionen und Daten darstellen
- Zwischenwerte schätzen: Interpolation
- Gleichungen lösen: Nullstellen & Fixpunkte
- Ableitung in einer und mehreren Variablen
- **Integration**
- **Gewöhnliche Differentialgleichungen**
- **Zufallszahlen simulieren**



# Typen von Verfahren

- Punktweise Approximation (Interpolation, Ableitung)
  - Näherungsweise Berechnung um Punkte herum
  - Daten vs. Funktion
  - Lohnt sich dann wenn kein Modell (Funktion) bekannt ist, oder dieses aufwändig zu berechnen ist.
  - Anwendung: Grafische Darstellung, Hilfsmittel für iterative Verfahren
- Iterative Verfahren (Newton, Bisektion, Gradient Descent)
  - Schrittweise Annäherung an einen speziellen Punkt
    - Einfachster Fall: Feste Schrittgrösse
    - Erweiterungen: z.B. dynamische Anpassung der Schrittweite oder der Lernrate
  - Abbruchkriterien → Konvergenz, Fehlerabschätzung
  - Anwendungen: Gleichungen lösen, Optimierung (Anpassen, Lernen, etc.)
  - Wenn nur Daten vorhanden, dann müssen Zwischenpunkte geschätzt werden  
→ Interpolation oder Modellierung

# Best Practices: Ziele

- **Genauigkeit:**  
Der Code liefert genaue Ergebnisse und ist robust gegenüber numerischen Instabilitäten und Rundungsfehlern.
- **Effizienz:**  
Der Code ist optimiert, um die bestmögliche Leistung aus der zugrunde liegenden Hardware herauszuholen.
- **Modularität:**  
Der Code ist modular und wiederverwendbar, was das Testen und Debuggen erleichtert.
- **Verständlichkeit:** Der Code ist gut strukturiert, dokumentiert und leicht verständlich.

# Best Practices: Massnahmen (Richtlinien, Prinzipien)

- Genauigkeit
  - Bestehende Lösungen (Packages) verwenden (aber nur vertrauenswürdige Quellen)
- Effizienz
  - IO-Operationen vermeiden (Console, Disk, Network, etc.)
  - Vektorisieren
  - Parallelisieren
  - Gezielt Mutationen zulassen (in-place)
  - C-Code kompilieren (Cython)