Life Sciences and
Facility Management

Institute of
Computational Life Sciences

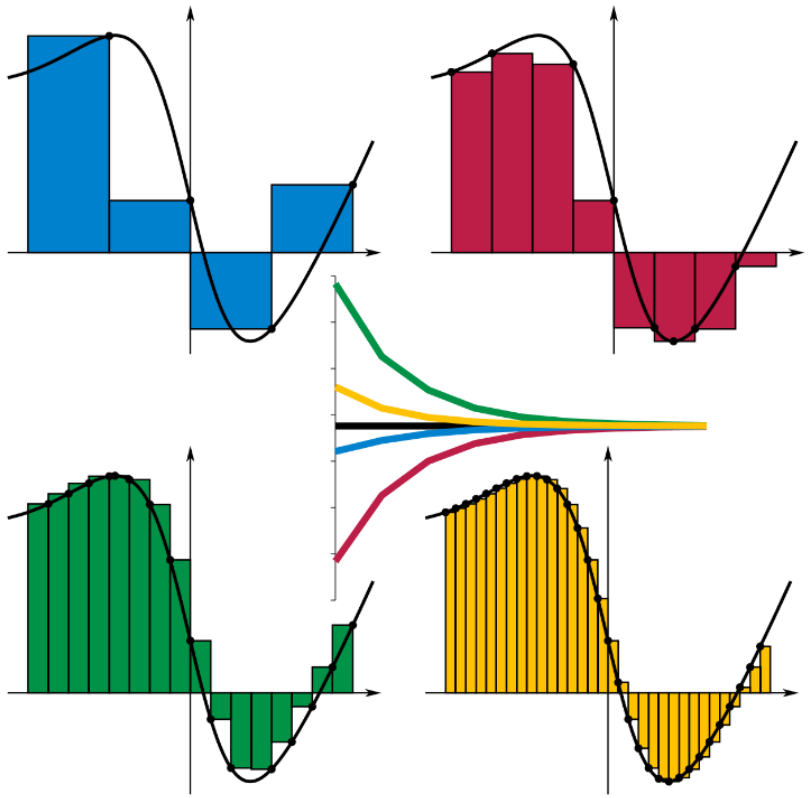
Zwischenbilanz

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?

- …

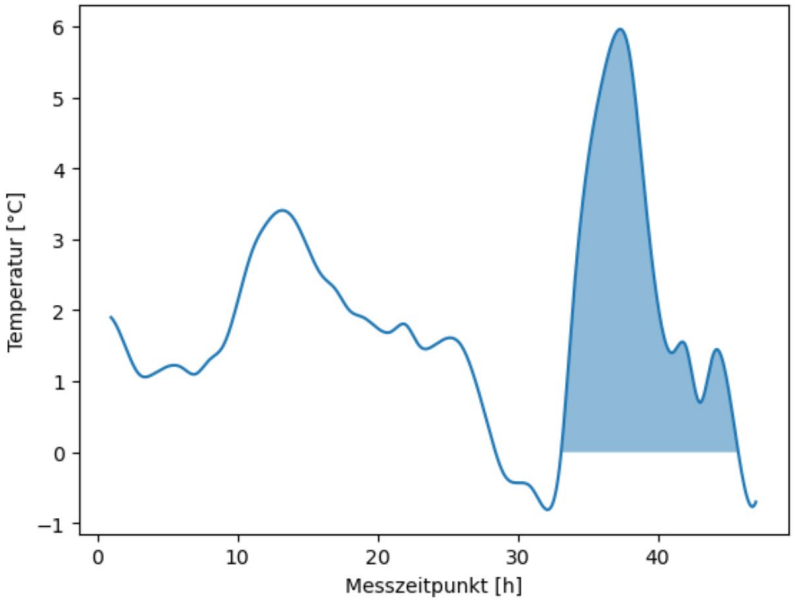
Themenübersicht

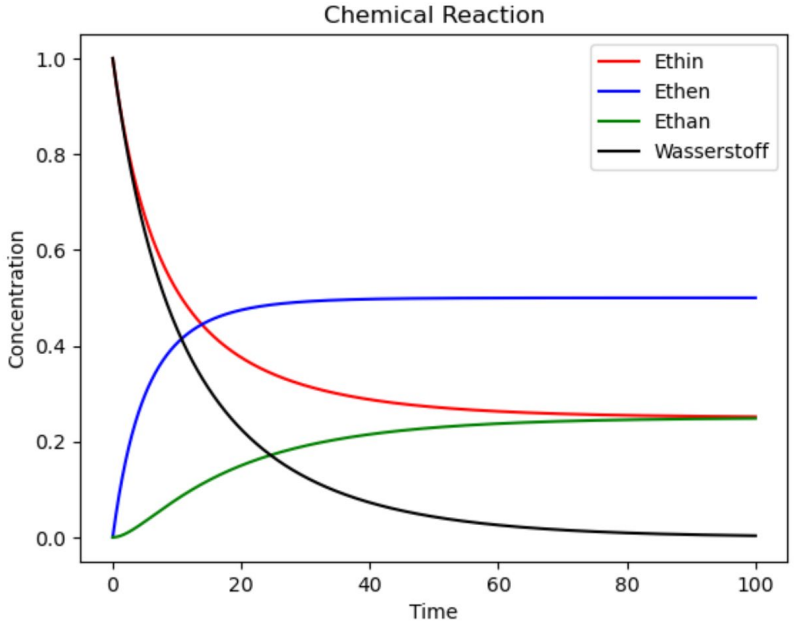
- Grundlagen:
  - Arbeiten mit mehrdimensionalen Arrays
  - Lineare Algebra
  - Funktionen und Daten darstellen
- Zwischenwerte schätzen: Interpolation
- Gleichungen lösen: Nullstellen & Fixpunkte
- Ableitung in einer und mehreren Variablen
- Integration
- Gewöhnliche Differentialgleichungen
- Zufallszahlen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

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

grad: Je kleiner α umso näher beim Ziel

wichtig z.B Newton / Bisect /

Glinear, kubisch

Best Practices: Ziele

- **Genauigkeit:**
  Der Code liefert genaue Ergebnisse und ist robust gegenüber numerischen Instabilitäten und Rundungsfehlern.
  Edgecases/Grenzfälle/Test
- **Effizienz:**
  Der Code ist optimiert, um die bestmögliche Leistung aus der zugrunde liegenden Hardware herauszuholen.
  erst am Schluss
- **Modularität:**
  Der Code ist modular und wiederverwendbar, was das Testen und Debuggen erleichtert.
  Generelle Funktionen / funktionen Aufteilen / Wichtig für Projekt Kombinierbarkeit
- **Verständlichkeit:** Der Code ist gut strukturiert, dokumentiert und leicht verständlich.
  Aussagekräftige Variablen, Nachvollziehbar

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)