Woche 1

Donnerstag, 23. Februar 2023 11:19

Ziel der VL: Tools für das richtige Leben als Physiker kennenlernen

Schwierigkeiten dabei: sehr viel neuer, anspruchsvoller Stoff

- Löst unbedingt "angewandte" Aufgaben (Programmieren mit Templates)

C++ vs. Python: (Python arbeitet vor allem mit packages)

```
int val = 5                          ~>  val = 3
if(i<3){                             ~>  if(i<3):
  std::cout << "Muss net                   print("muss wirge")
               schmegge"
}
```

-> packages fügen Funktionalität ein - wichtige Packages:

numpy	matplotlib.pyplot	scipy
Arrays (& Operationen)	Funktionen darstellen	wiss. Anwendungen
Datenverarbeitung	Plotten	numerische Integrale
		Curvefitting
import numpy as np	import matplot.. as plt	from scipy import ..

- Arrays werden in Python durch numpy bereitgestellt
  ! Arrays $\neq$ Listen

Coding Tutorial ~> s. Templates

Quadratur:

gegeben: Funktion $f: \mathbb{R} \to \mathbb{R}$, wollen $\int_a^b f(x)dx$ berechnen

(Physik $\equiv$ quasi Integration :))

Numerische Motivation:
- komplizierte Integrale berechnen z.B. für $f(x) = \sqrt{\sqrt{x}\, e^{-x^2}}$
- universelles Verfahren

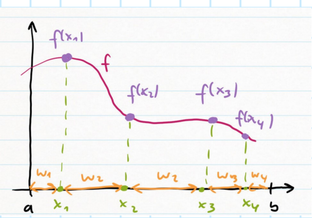
Def.: (Quadratur)

Eine Quadraturformel besteht aus einer Wahl von Stützstellen $\{x_i\}_{i=1}^n \subseteq [a,b]$ und dazugehörigen Gewichten $\{w_i\}_{i=1}^n$

Die Quadratur wird dann wie folgt ausgewertet

$$\int_a^b f(x)dx \approx Q_n(f,a,b) = \sum_{i=1}^n f(x_i)w_i$$

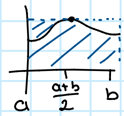
Graphisch:



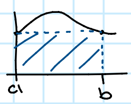
proof by picture:
gute Wahl von $\{x_i\}$, $\{w_i\}$ macht gute Quadratur

Newton-Cotes: (NC)

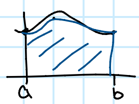
1) Mittelpunktsregel: $x_i = \frac{a+b}{2}$, $w_i = b-a$



2) Trapezregel: $x_1 = a$, $x_2 = b$ $\quad w_1 = w_2 = \frac{b-a}{2}$
(p = 2)



3) Simpsonregel: $x_1 = a$, $x_2 = \frac{a+b}{2}$, $x_3 = b$ $\quad w_1 = \frac{b-a}{6}$, $w_2 = \frac{4}{6}(b-a)$, $w_3 = \frac{b-a}{6}$
(p = 4)



⇝ Wie vergleichen wir die Quadratur?

Def.: (Ordnung)

$Q_n(f;a,b)$ hat Ordnung p, falls ein Polynom vom Grad p-1 exakt integriert werden kann

Def.: (Fehler)

$$E(n) = \Big|\underbrace{\int_a^b f(x)dx}_{\text{exaktes } \int} - \underbrace{Q_n(f;a,b)}_{\text{Quadratur}}\Big|$$

NC: $\forall f \in C^n([a,b])$: $E(n) \leq \underbrace{\frac{(b-a)^{n+1}}{n!}}_{\text{Intervallgröße}} \max_{x\in[a,b]} \underbrace{|f^{(n)}(x)|}_{\text{Glattheit}}$

Allgemeiner: 2 Fehlerklassen

a) algebraische Konvergenz

$\exists c>0\ \exists p>0$ s.d. $\forall n \in \mathbb{N}$: $E(n) \leq \frac{c}{n^p}$

b) exponentielle Konvergenz:

$\exists 0<q<1$, $c>0$ s.d. $\forall n \in \mathbb{N}$ $E(n) \leq c q^n$

algebraisch: $(n, E(n))$ gerade loglog

exponentiell: $(n, E(n))$ gerade linlog

Zusammengesetzte NC

- Linearität Integral: $a = x_0 < x_0 + h < \dots < x_0 + Nh = b$

$$\int_a^b f(x)dx = \sum_{i=0}^{N-1} \underbrace{\int_{x_i}^{x_{i+1}} f(x)dx}_{\to Q_n(f;a,b)} = \sum_{i=0}^{N-1} Q_n(f;x_i,x_{i+1})$$

äquidistant: $x_{i+1} - x_i = \frac{b-a}{N} = h$

Fehler: Sei $Q_n(f;a,b)$ von Ordnung p. Dann gilt für die zQF (äquidistant)

$\forall f \in C^p([a,b])\ \exists c>0$ s.d. $E(n) \leq c\, h^p \max_{x\in[a,b]} |f^{(p)}(x)|$

⇒ algebraische Konvergenz

2D-Quadratur:

geg.: $f(x,y)$ in $[a,b]\times[c,d]$

analytisch: $I = \int_{[\ ]\times[\ ]} f(x,y)\,d(x,y)$

Per Hand: $I = \int_a^b \int_c^d f(x,y)\,dx\,dy = \int_c^d F(y)\,dy$

In Python: 1) Von außen nach Innen arbeiten

$I = \text{Quad}(F(y), c, d)$

2) $F(y) = \int_c^b f(x,y)\,dy$

Python: $F(y) =$ lambda y: Quad $(\ast), a, b)$

$(\ast)$ muss 1-d: $(\ast) =$ lambda x: $f(x,y)$

Lambda Functions ermöglichen schnelle und einfache Definition von Fkt.

Bsp.: f_{y_0} = lambda x: $f(x, y_0)$

Watch it auftauchen:

- Funktionen übergeben

Bsp.:
```
def exp_op(f,a,b):
    return f(a,b)

def mult(a,b):
    return a*b
```

=> exp_op(mult, a, b) Wichtig für Funktionen, die versch. Methoden ausführen sollen func(Trapez ⋮ , more args)

- np.vectorize: ermöglicht Array Eingabe von sonst Skalarfkt.
  - Bei Funktionen mit numpy arrays: iteriert über Eingabearray und versucht nicht ganzes Array einzubauen