

Numerische Methoden

Woche 2

David Nadlinger

nadavid@ethz.ch

27. Februar 2013

Organisatorisches

- Mittwoch, 13:15–15:00, IFW C 35
- Übung entfällt am 1. Mai (Tag der Arbeit)
- Serien: Abgabe in gedruckter Form, Fach im HG G 53
- Testat: 70% sinnvoll bearbeitet
- Python-Fragestunde: Freitag, 12–14 Uhr, HG E 26.1 (Raoul Bourquin)

Was euch erwartet

- Numerik ist in den Naturwissenschaften heute essentiell
- Vorlesung gibt Querschnitt durch viele Themenbereiche
- Nützlich in späteren Vorlesungen (z.B. MMP)
- Anspruchsvoll, aber schaffbar
- Lasst euch bzgl. Python nicht abhängen!

Funktionen und Module

```
1  # foo.py
2  def add_three(x):
3      return x + 3
4
5  def max(x, y):
6      if x > y:
7          return x
8      return y
9
10 if __name__ ==
    '__main__':
11     print(add_three(2))
12     print(max(27, 42))
```

- `import foo`
- `foo.max(3, 4)`
- `from foo import add_three`
- `add_three(2)`
- `from foo import *`
- `max(1, 2)`
- `run foo.py`

Lambda-Funktionen

```
1  add_three = lambda x: x + 3
2  max = lambda x, y: x if x > y else y
```

- Körper besteht aus einem einzigen Ausdruck
- **return** implizit
- Warum verwenden? Lambdas sind selbst Ausdrücke!

NumPy-Cheatsheet

Elementare Funktionen

- `np.exp`
- `np.log`
- `np.sqrt`
- `np.(arc)sin`
- `np.(arc)cos`

- `np.abs`
- `np.real`
- `np.imag`
- `np.conj`

- `np.pi`

Arrays erzeugen

- `np.array`
- `np.empty`
- `np.zeros`
- `np.ones`
- `np.arange`
- `np.linspace`
- `np.logspace`

Arrays manipulieren

- `np.shape`
- `np.reshape`
- `np.hstack`
- `np.vstack`

Array-Algorithmen

- `np.min`
- `np.max`
- `np.where`
- `np.(arg)sort`

NumPy-Cheatsheet

Lineare Algebra

- `np.dot`
- `np.eye`
- `np.diag`
- `np.linalg.norm`
- `np.linalg.solve`
- `np.linalg.eig`

Numerische Algorithmen

- `np.polyfit`
- `scipy.optimize.fsolve`
- `scipy.integrate.quad`

Zufallszahlen

- `np.random.rand`

FFT

- `np.fft.(i)fft`
- `np.fft.fftshift`

- Um selbst Vektor-Funktionen zu schreiben, siehe `np.vectorize`.
- http://www.scipy.org/NumPy_for_Matlab_Users

Nullstellensuche in \mathbb{R}

- Konvergenzordnung iterative Verfahren
- Bisektion
- Fixpunktiteration allgemein
- Newton-Verfahren
- Sekantenmethode

Schluss

- Python-Fragestunde: Freitag, 12–14 Uhr, HG E 26.1 (Raoul Bourquin)