Hydroinformatik II - SoSe 2024

HyBHW-S2-01-V08: Übung: Jupyter Notebook für Diffusionsprozesse

Prof. Dr.-Ing. habil. Olaf Kolditz

[1] Helmholtz Centre for Environmental Research – UFZ, Leipzig

[2] Technische Universität Dresden – TUD, Dresden

[3] Center for Advanced Water Research – CAWR

[4] TUBAF-UFZ Center for Environmental Geosciences – C-EGS, Freiberg / Leipzig

Dresden, 28.06.2024

Zeitplan: Hydroinformatik II - SoSe 2024

Datum	HI	II	Thema	Typ
14.06.2024	14	2-01	Einführung in die Lehrveranstaltung - Teil 2	L
14.06.2024	15	2-02	Werkzeuge | Tools	L
14.06.2024	16	2-03	Grundlagen: Kontinuumsmechanik	L
21.06.2024	17	2-04	Grundlagen: Hydromechanik	L
21.06.2024	18	2-05	Grundlagen: Partielle Partialgleichungen	L
21.06.2024	19	2-06	Übung: Analytische Lösungen	E
28.06.2024*	20	2-07	Grundlagen: Näherungsverfahren	L
28.06.2024*	21	2-08	Übung: Jupyter Diffusionsprozess	E
02.07.2024*	22	2-09	Numerik: Finite-Differenzen-Methode (explizit)	L
02.07.2024*	23	2-10	Numerik: Finite-Differenzen-Methode (implizit)	L
12.07.2024	24	2-11	Übung: Finite-Differenzen-Methoden	E
12.07.2024	25	2-12	Grundlagen: Gerinnehydraulik	L
12.07.2024	26	2-13	Übung: Gerinnehydraulik	E
19.07.2024	27	2-14	Ausblick: Grundwassermodellierung	E
19.07.2024	28	2-15	Klausur/Beleg: Besprechung zur Vorbereitung	L

*online Vorlesung

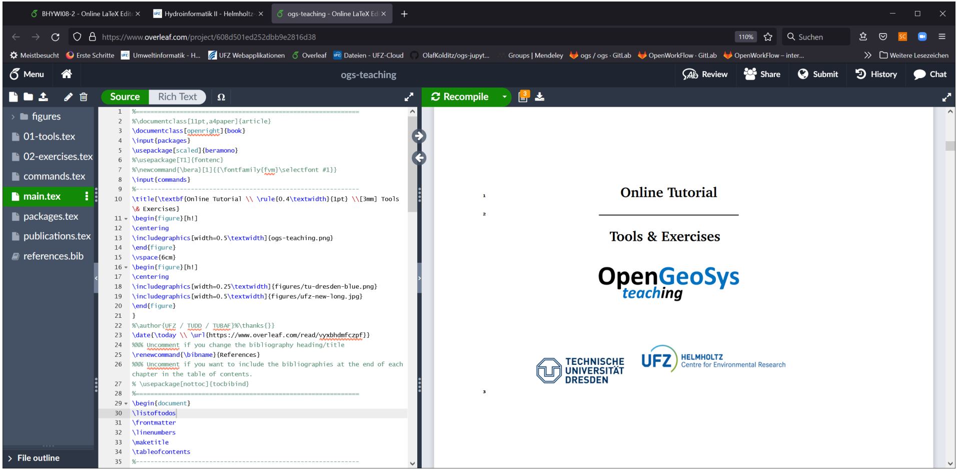
Fahrplan für heute ...

1. HyBHW-S2-01-V08: Übung: Jupyter Notebook für Diffusionsprozesse
   - 28.06.2024
2. Hydroinformatik II - Tutorial
3. Übung: Diffusionsgleichung (recap)
4. Jupyter Notebook (recap)
5. Übung: Diffusionsgleichung - Jupyter Notebook

TECHNISCHE UNIVERSITÄT DRESDEN

Lehre-Webseite ↦ Tutorial

`https://www.overleaf.com/read/vyxbhdmfczpf`



Diffusionsgleichung: Analytische Lösung Tutorial: Abschnitt 2.2

- Repo: BHYWI-08-02-E-Script
- C++ Variante
- C++ / Python Variante
- Python Variante

MinGW + Python: Function plotter (analytical solution) #1

Übung: BHYWI-08-02-E-Script: Skript

```
echo Compilation
g++ main.cpp
echo Execution
a.exe
echo Ploting
data_from_file.py
echo End
```

Listing: Script file for entire workflow

```cpp
#include <cmath>
#include <fstream>
#define PI 3.14159265358979323846
int main(int argc, char *argv[])
{
  //1-Definitionen
  int numPoints = 1000;
  double x,y,alpha=1.,t=0.01;
  std::ofstream out_file;
  out_file.open("out.csv");
  //2-Berechnung
  for (int i = 0; i < numPoints+1; ++i)
  {
    x = double(i)/double(numPoints);
    y=sin(PI*x) * exp(-alpha*PI*PI*t);
    out_file << x << "," << y << std::endl;
  }
  //3-Ausgabe
}
```

Listing: C++ program for analytical solution

MinGW + Python: Function plotter (analytical solution) #3

Übung: BHYWI-08-02-E-Script: File reading and data plotting

```python
import matplotlib.pyplot as plt
import csv

x = []
y = []
with open('out.csv','r') as csvfile:
    plots = csv.reader(csvfile, delimiter=',')
    for row in plots:
        x.append(float(row[0]))
        y.append(float(row[1]))

plt.plot(x,y, label='Data loaded from file: out.cvs')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Hydroinformatics II\nExercise BHYWI-08-02-for-python')
plt.legend()
plt.savefig("diffusion-equation.png.png")
plt.show()
```

Listing: File reading and data plotting

TECHNISCHE UNIVERSITÄT DRESDEN

Übung: EX06-parabolische-gleichung-1D.py

```python
import math
import matplotlib.pylab as plt
PI = 3.14159265358979323846
numPoints = 10
alpha = 1.0
t = [0.1,0.5,1.0,2.0]
x = []
y = []
for n in t:
    for i in range(0,numPoints+1):
        x.append(float(i)/float(numPoints))
        y.append(math.sin(PI*x[i]) * math.exp(-alpha*n*n))
    plt.plot(x,y,color='red',marker="o")
    x = []
    y = []
plt.xlabel('x')
plt.ylabel('u')
plt.savefig("diffusion-equation.png")
plt.show()
```

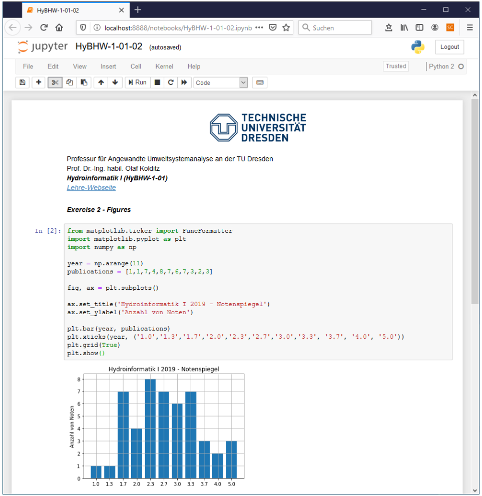
Listing: Analytical solution

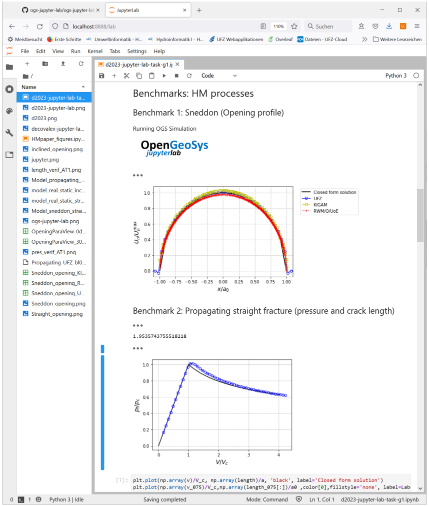
Jupyter

- Jupyter Notebook
- Jupyter Lab
- Browser-basiert

- "The Jupyter Notebook · The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, ..."
- Webseite: `https://jupyter.org/`
- Vorteil: funktioniert auf allen Rechnern
- ... ein Teil unserer (neuen) Übungen machen wir mit Jupyter Notebooks ($>>$ Demo)

Jupyter: Example





Diffusionsgleichung: Analytische Lösung Tutorial: Abschnitt 2.2

- Jupyter Notebook

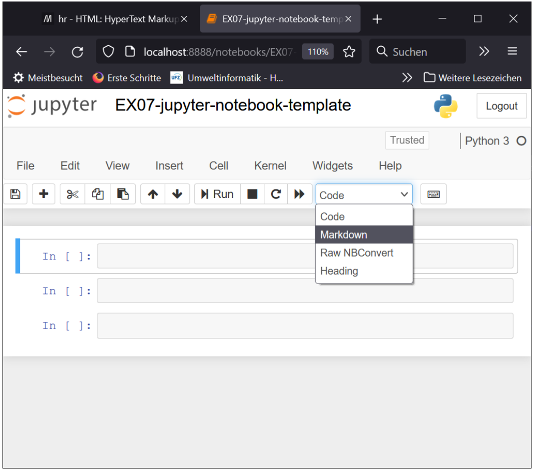
Übung: EX06-parabolische-gleichung-1D.py

```
import math
import matplotlib.pylab as plt
PI = 3.14159265358979323846
numPoints = 10
alpha = 1.0
t = [0.1,0.5,1.0,2.0]
x = []
y = []
for n in t:
    for i in range(0,numPoints+1):
        x.append(float(i)/float(numPoints))
        y.append(math.sin(PI*x[i]) * math.exp(-alpha*n*n))
    plt.plot(x,y,color='red',marker="o")
    x = []
    y = []
plt.xlabel('x')
plt.ylabel('u')
plt.savefig("diffusion-equation.png")
plt.show()
```

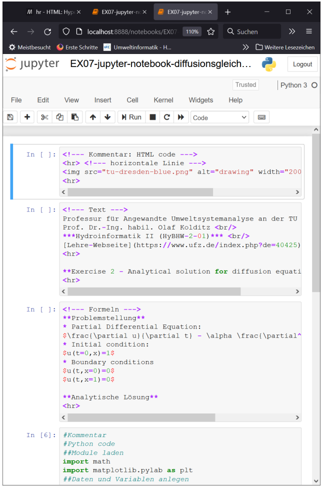
Listing: Analytical solution

Python:

Übung: EX07-jupyter-notebook-template.ipynb



$\Longrightarrow$



TECHNISCHE UNIVERSITÄT DRESDEN

Nächste Veranstaltung am 03.06.2022 ist online

Hausaufgaben

-