

Hydroinformatik II - SoSe 2024

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

Prof. Dr.-Ing. habil. Olaf Kolditz

¹Helmholtz Centre for Environmental Research – UFZ, Leipzig

²Technische Universität Dresden – TUD, Dresden

³Center for Advanced Water Research – CAWR

⁴TUBAF-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

- 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

Lehre-Webseite → Tutorial

<https://www.overleaf.com/read/vyxbhdmfczpf>

The screenshot displays the Overleaf online LaTeX editor interface. The left sidebar shows a file explorer with a project named 'figures' containing files like '01-tools.tex', '02-exercises.tex', 'commands.tex', 'main.tex' (selected), 'packages.tex', 'publications.tex', and 'references.bib'. The main editor area shows the LaTeX source code for 'main.tex', which includes a table of contents and references. The right pane shows the compiled PDF output, which features the title 'Online Tutorial', a subtitle 'Tools & Exercises', the 'OpenGeoSys teaching' logo, and logos for 'TECHNISCHE UNIVERSITÄT DRESDEN' and 'UFZ HELMHOLTZ Centre for Environmental Research'.

```
1 %
2 \documentclass[11pt,a4paper]{article}
3 \documentclass[openright]{book}
4 \input{packages}
5 \usepackage[scaled]{beramono}
6 \usepackage[T1]{fontenc}
7 \newcommand{\libra}[1]{\fontfamily{\libra}\selectfont #1}
8 \input{commands}
9 %
10 \title{\textbf{Online Tutorial} \rule{0.4\textwidth}{1pt} \textbf{Tools
11 \begin{figure}[ht]
12 \centering
13 \includegraphics[width=0.5\textwidth]{ogs-teaching.png}
14 \end{figure}
15 \vspace{6cm}
16 \begin{figure}[ht]
17 \centering
18 \includegraphics[width=0.25\textwidth]{figures/tu-dresden-blue.png}
19 \includegraphics[width=0.5\textwidth]{figures/ufz-new-long.jpg}
20 \end{figure}
21 %
22 \author{UFZ / TUDD / TUDD/100 thanks}
23 \date{\today \url{https://www.overleaf.com/read/vyxbhdmfczpf}}
24 % \comment if you change the bibliography heading/title
25 \renewcommand{\bibname}[References]
26 % \comment if you want to include the bibliography at the end of each
27 % \usepackage{notoc}[tocbibind]
28 %
29 \begin{document}
30 \listoftodos
31 \frontmatter
32 \linenumbers
33 \maketitle
34 \tableofcontents
35 %
```

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

```
1 echo Compilation
2 g++ main.cpp
3 echo Execution
4 a.exe
5 echo Ploting
6 data_from_file.py
7 echo End
```

Listing: Script file for entire workflow

MinGW + Python: Function plotter (analytical solution) #2

Übung: BHYWI-08-02-E-Script: C++

```
1 #include <cmath>
2 #include <fstream>
3 #define PI 3.14159265358979323846
4 int main(int argc, char *argv[])
5 {
6     //1-Definitionen
7     int numPoints = 1000;
8     double x,y,alpha=1.,t=0.01;
9     std::ofstream out_file;
10    out_file.open("out.csv");
11    //2-Berechnung
12    for (int i = 0; i < numPoints+1; ++i)
13    {
14        x = double(i)/double(numPoints);
15        y=sin(PI*x) * exp(-alpha*PI*PI*t);
16        out_file << x << "," << y << std::endl;
17    }
18    //3-Ausgabe
19 }
```

Listing: C++ program for analytical solution

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

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

```
1 import matplotlib.pyplot as plt
2 import csv
3
4 x = []
5 y = []
6 with open('out.csv','r') as csvfile:
7     plots = csv.reader(csvfile, delimiter=',')
8     for row in plots:
9         x.append(float(row[0]))
10        y.append(float(row[1]))
11
12 plt.plot(x,y, label='Data loaded from file: out.csv')
13 plt.xlabel('x')
14 plt.ylabel('y')
15 plt.title('Hydroinformatics II\nExercise BHYWI-08-02-for-python')
16 plt.legend()
17 plt.savefig("diffusion-equation.png.png")
18 plt.show()
```

Listing: File reading and data plotting

```
1 import math
2 import matplotlib.pyplot as plt
3 PI = 3.14159265358979323846
4 numPoints = 10
5 alpha = 1.0
6 t = [0.1,0.5,1.0,2.0]
7 x = []
8 y = []
9 for n in t:
10     for i in range(0,numPoints+1):
11         x.append(float(i)/float(numPoints))
12         y.append(math.sin(PI*x[i]) * math.exp(-alpha*n*n))
13     plt.plot(x,y,color='red',marker="o")
14     x = []
15     y = []
16 plt.xlabel('x')
17 plt.ylabel('u')
18 plt.savefig("diffusion-equation.png")
19 plt.show()
```

Listing: Analytical solution

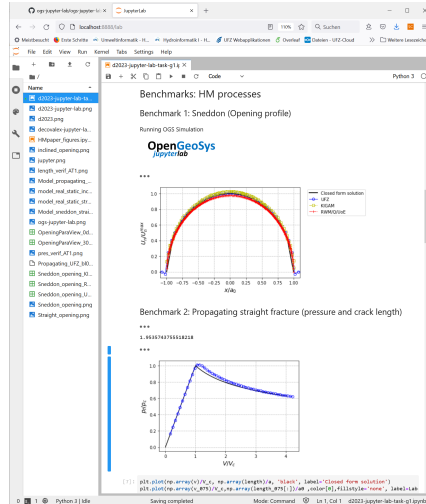
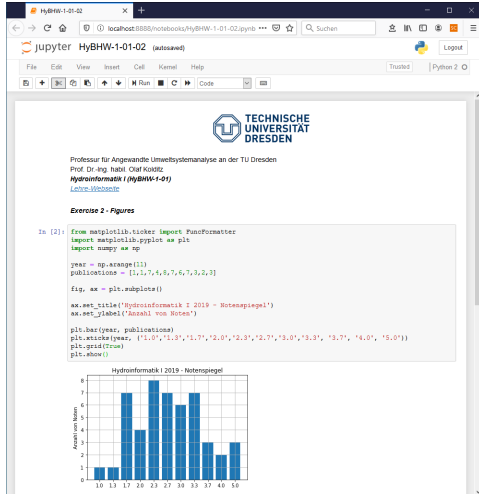
Jupyter

- 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

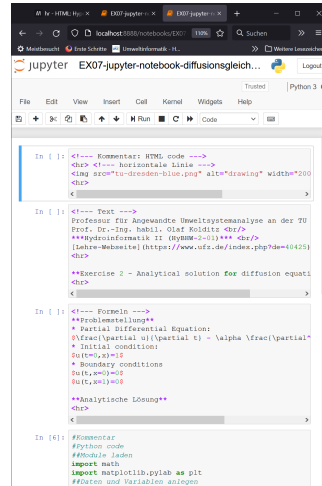
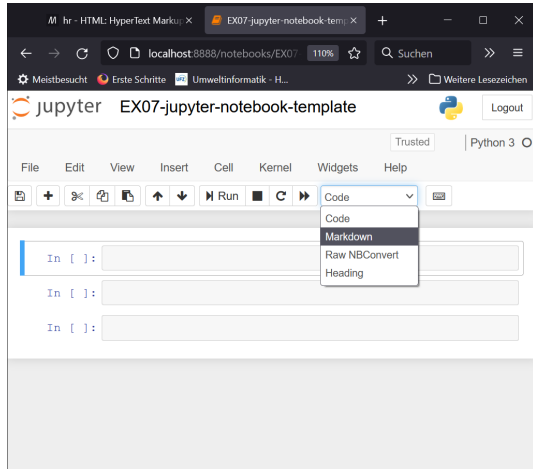
- Jupyter Notebook

```
1 import math
2 import matplotlib.pyplot as plt
3 PI = 3.14159265358979323846
4 numPoints = 10
5 alpha = 1.0
6 t = [0.1,0.5,1.0,2.0]
7 x = []
8 y = []
9 for n in t:
10     for i in range(0,numPoints+1):
11         x.append(float(i)/float(numPoints))
12         y.append(math.sin(PI*x[i]) * math.exp(-alpha*n*n))
13     plt.plot(x,y,color='red',marker="o")
14     x = []
15     y = []
16 plt.xlabel('x')
17 plt.ylabel('u')
18 plt.savefig("diffusion-equation.png")
19 plt.show()
```

Listing: Analytical solution

Python:

Übung: EX07-jupyter-notebook-template.ipynb



Nächste Veranstaltung am 03.06.2022 ist online

Hausaufgaben

