

Hydroinformatik - SoSe 2026

UW-BHW-414-17: Grundwasserhydraulik

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, 10.07.2026

Zeitplan: Hydroinformatik I+II

Sommersemester 2026: Stand: 06.04.2026

Nr.	KW	Datum	ID	Thema
01+02	16	17.04.2026	UW-BHW-414-01/02	Einführung in die Vorlesung, Umweltinformatik
03	16	17.04.2026	UW-BHW-414-03	Werkzeuge, Hello World (in C++)
05	17	24.04.2026	UW-BHW-414-04	Selbststudium: Software-Installationen
07	19	08.05.2026	UW-BHW-414-05	Objekt-Orientierte Programmierung: C++, Klassen
09	20	15.05.2026	UW-BHW-414-06	Programmiersprache Python
11	21	22.05.2026	UW-BHW-414-07/08	Modellierung, Digitalisierung - Wasser 4.0
00	22	29.05.2026		Vorlesungsfreie Woche
13	23	05.06.2026	UW-BHW-414-09/10	KI, Maschinelles Lernen, Neuronale Netzwerke
15	24	12.06.2026	UW-BHW-414-11/12	Kontinuumsmechanik, Hydromechanik
17	25	19.06.2026	UW-BHW-414-13/14	Differentialgleichungen, Näherungsverfahren
19	26	26.06.2026	UW-BHW-414-15	Finite-Differenzen, explizite Verfahren
21	27	03.07.2026	UW-BHW-414-16	Finite-Differenzen, implizite Verfahren
23	28	10.07.2026	UW-BHW-414-17	Gerinnehydraulik, Grundwasserhydraulik
25	29	17.07.2026	UW-BHW-414-M	Grundwasserhydraulik
27	30	24.07.2026	UW-BHW-414-N	Zusammenfassung, Klausurvorbereitung

- 1 UW-BHW-414-17: Grundwasserhydraulik
 - Semesterplan

Fahrplan für heute ...

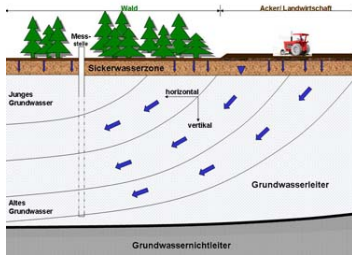
- 3 Gerinne-Hydraulik: Theorie und Übung
 - 2 Energieerhaltung: Bernoulli-Gleichung
 - 1 Impulserhaltung: Saint-Venant-Gleichungen
 - 0 Newton-Verfahren
-

- 1 Zusammenfassung und Abschluss Übungen (Jupyter Notebook)
 - 2 Einführung Grundwassermodellierung (Ausblick 6. Semester MvH)
 - 3 Studierenden-Befragung (QR code)
-

- 4 Klausur am 01.08.2025: 09:20 bis 10:50 Uhr im Gerhart-Potthoff-Bau POT/13/U

Diffusionsprozesse

- Schadstoffe im GW (Quellen)



- Mathematik

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

- Lösung: FDM (ex/im)

Gerinnehydraulik

- Weißeritz (Quellen)



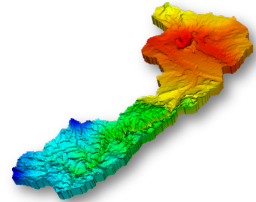
- Mathematik

$$\frac{\partial u}{\partial t} = \alpha(u) \frac{\partial^2 u}{\partial x^2}$$

- Lösung: Newton

Grundwasserströmung

- Selke-Modell



- Mathematik

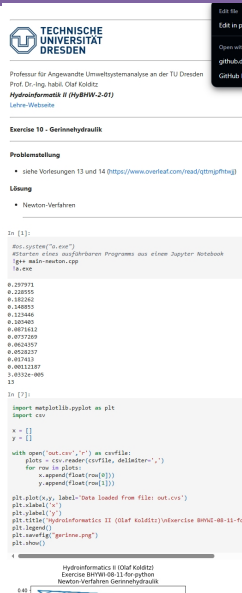
$$\frac{\partial u}{\partial t} = \alpha_x \frac{\partial^2 u}{\partial x^2} + \alpha_y \frac{\partial^2 u}{\partial y^2}$$

- Lösung: Gauss-Seidel, FEM (MvH im 6. Sem)

Übungsabschluss: Jupyter Notebook

- Gerinne-Hydraulik
- HA: implizites Finite-Differenzen Verfahren
- `git merge`

Jupyter Notebook für Gerinne-Hydraulik-Übung



TECHNISCHE UNIVERSITÄT DRESDEN

Professur für Angewandte Umweltsystemanalyse an der TU Dresden
Prof. Dr.-Ing. habil. Olaf Kolditz
Hydroinformatik II (HyBHW-2-01)
Lehre-Webseite

Exercise 10 - Gerinnehydraulik

Problemstellung

- siehe Vorlesungen 13 und 14 (<https://www.overleaf.com/read/qttmjqthwjl>)

Lösung

- Newton-Verfahren

```
In [1]:
!os.system("d.exe")
#Starten eines ausführbaren Programms aus einem Jupyter Notebook
!g++ main-newton.cpp
ls.exe

0.297971
0.228555
0.182582
0.148853
0.123446
0.100485
0.0871612
0.0737269
0.0604357
0.0528257
0.047413
0.0012187
0.0532e-005
13
```

```
In [7]:
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.csv')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Hydroinformatics II (Olaf Kolditz) \n Exercise BHWI-08-11-for-python')
plt.legend()
plt.savefig('gerinne.png')
plt.show()
```

Hydroinformatics II (Olaf Kolditz)
Exercise BHWI-08-11-for-python
Newton-Verfahren Gerinnehydraulik

- ▶ Jupyter notebooks starten von der Konsole/Terminal:
`jupyter notebook your-filename.ipynb`
- ▶ neue Zellen einfügen und code übertragen
- ▶ externe Programme (z.B. Compiler) können mit `!g++` gestartet werden
- ▶ Windows: PATH muss gesetzt sein
- ▶ Mac/Linux/Ubuntu: aktuelles Verzeichnis muss bei jedem Aufruf explizit angegeben sein: `! ./g++`
- ▶ Zellen müssen beim Starten auf code gestellt sein

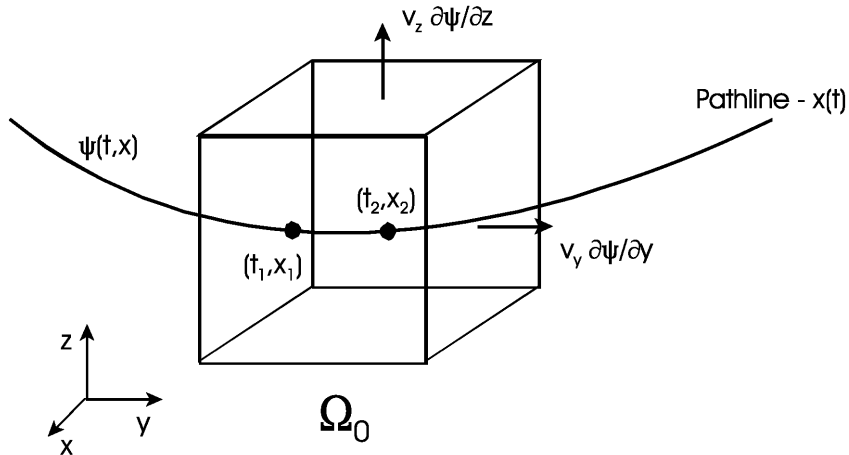
Grundwasserströmung

Einführung

- Theorie
- Prinzipbeispiel

- ▶ Grundlagen (kurze Wiederholung)
- ▶ Grundwassergleichung
- ▶ Prinzip-Beispiel
- ▶ Bilanzierung
- ▶ Berechnungsverfahren
- ▶ ...

Das Euler Prinzip (Wdh)



$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = \mathbf{f}^e + \frac{1}{\rho} \nabla \cdot \sigma \quad (1)$$

In index notation the above vector equation is written as

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} &= \frac{1}{\rho} \left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} \right) \\ \frac{\partial v}{\partial t} + v \frac{\partial u}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} &= \frac{1}{\rho} \left(\frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} \right) \\ \frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} &= g + \frac{1}{\rho} \left(\frac{\partial \sigma_{zx}}{\partial x} + \frac{\partial \sigma_{zy}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} \right) \end{aligned} \quad (2)$$

with $u = v_x, v = v_y, w = v_z$ and $\mathbf{f}^e = \mathbf{g}$.

Flow Equations - Systematic (Wdh)

Stress Tensor

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau} \quad (3)$$

Navier-Stokes Equation

$$\boxed{\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = \mathbf{f}^e - \frac{1}{\rho} \nabla p + \nu \Delta \mathbf{v}} \quad (4)$$

Euler Equation

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = \mathbf{f}^e - \frac{1}{\rho} \nabla p \quad (5)$$

Stokes Equation

$$\frac{\partial \mathbf{v}}{\partial t} = \mathbf{f}^e - \frac{1}{\rho} \nabla p + \nu \Delta \mathbf{v} \quad (6)$$

Darcy Equations

$$0 = \mathbf{f}^e - \frac{1}{\rho} \nabla p + \nu \Delta \mathbf{v}$$

$$\frac{\partial n\rho}{\partial t} + \nabla \cdot (n\rho\mathbf{v}) = Q_\rho \quad (8)$$

Für ein inkompressibles Fluid gilt dann (PF)

$$\rho \frac{\partial n}{\partial t} + \rho \nabla \cdot (n\mathbf{v}) = Q_\rho \quad (9)$$

oder noch besser

$$\frac{\partial n}{\partial t} + \nabla \cdot (n\mathbf{v}) = \frac{Q_\rho}{\rho_0} \quad (10)$$

In der Grundwasserhydraulik gilt

$$\frac{\partial n}{\partial t} = S \frac{\partial h}{\partial t} \quad (11)$$
$$n\mathbf{v} = \mathbf{q} = -\mathbf{K} \nabla h \quad (\text{Darcy Gesetz}) \quad (12)$$

Dabei sind: S der Speicherkoeffizient, h die Piezometer- oder hydraulische Höhe, $\mathbf{q}$ die Darcy- oder Filtergeschwindigkeit und $\mathbf{K}$ der hydraulische Leitfähigkeitstensor.

$$S \frac{\partial h}{\partial t} + \nabla \cdot (n \mathbf{v}) = Q$$

$$S \frac{\partial h}{\partial t} - \nabla \cdot (\mathbf{K} \nabla h) = Q$$

$$S \frac{\partial h}{\partial t} - \frac{\partial}{\partial x} \left(K_x \frac{\partial h}{\partial x} \right) - \frac{\partial}{\partial y} \left(K_y \frac{\partial h}{\partial y} \right) - \frac{\partial}{\partial z} \left(K_z \frac{\partial h}{\partial z} \right) = Q$$

Wir begnügen uns mit einem 2-D horizontalen Modell.

$$S \frac{\partial h}{\partial t} - \frac{\partial}{\partial x} \left(K_x \frac{\partial h}{\partial x} \right) - \frac{\partial}{\partial y} \left(K_y \frac{\partial h}{\partial y} \right) = Q \quad (13)$$

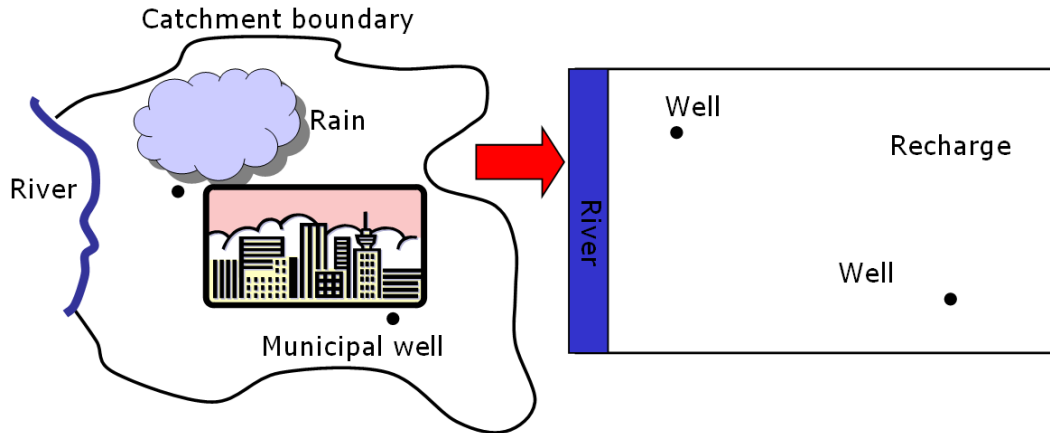
Grundwasserströmung

Einführung

- Prinzipbeispiel
- Grundwassermodellierung mit OpenGeoSys

Grundwasser: Prinzip-Beispiel

Idee



Quelle: Sebastian Bauer (Uni Kiel)

River:
Fixed head
boundary
condition
 $H = 10 \text{ m}$

No flow – e.g. fault

- Pumping well
 $Q_{P2} = 0.001 \text{ m}^3/\text{s}$

Recharge: homogeneous and constant
 $R = 1.0\text{e-}8 \text{ m}^3/(\text{m}^2\text{s})$

- Pumping well
 $Q_{P6} = 0.005 \text{ m}^3/\text{s}$

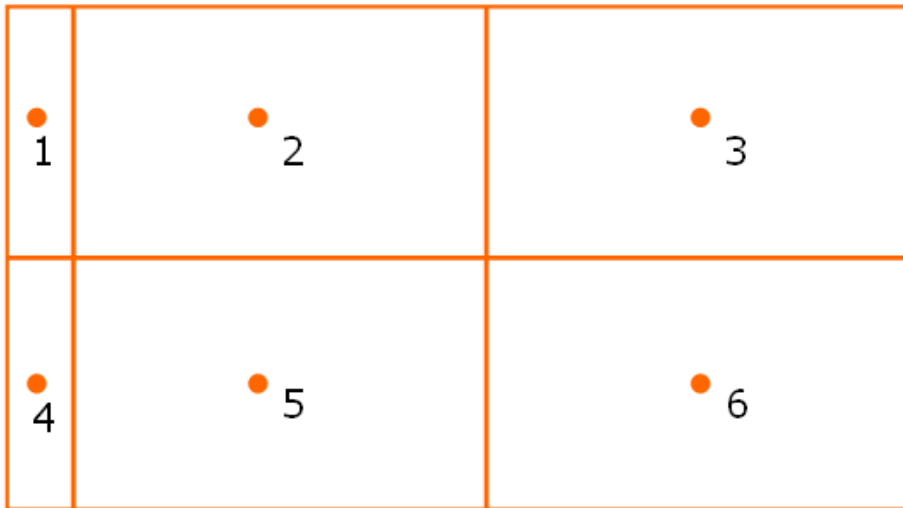
No flow – water divide

No flow - streamline

Fig.: Definition der Randbedingungen

Prinzip-Beispiel

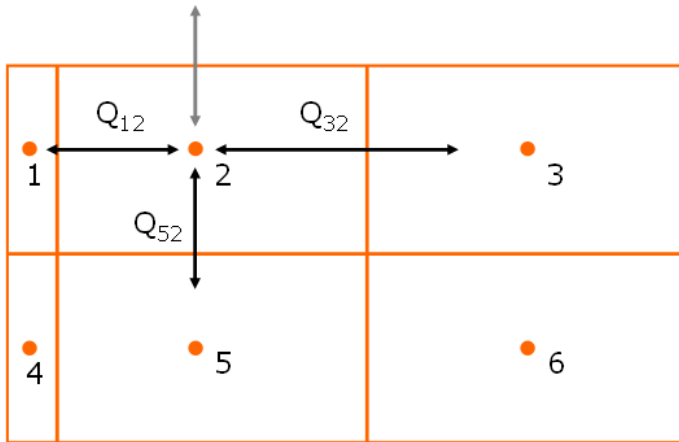
Diskretisierung



$\Delta y_1 = 500 \text{ m}$
 $\Delta y_2 = 500 \text{ m}$

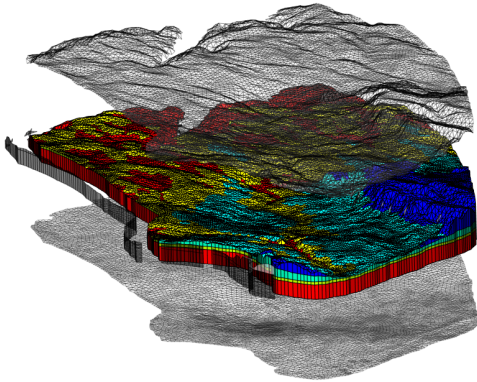
Prinzip-Beispiel

Bilanzierung: Finite-Differenzen >> Hier geht's im 6. Semester weiter ...



$$Q_{12} + Q_{32} + Q_{52} + Q_R + Q_{P2} = 0$$

OpenGeoSys
workflows



- ▶ Finite-Elemente-Methode
- ▶ Benchmarksammlung
- ▶ Brunnenhydraulik nach Theis

Beispiele: <https://www.youtube.com/user/opengeosys>

www.opengeosys.org

GitHub

Arbeiten mit einem eigenen Repo