

Hydroinformatik II - SoSe 2024

HyBHW-S2-01-V14: Zusammenfassung | Ausblick: Grundwassermodellierung

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, 19.07.2024

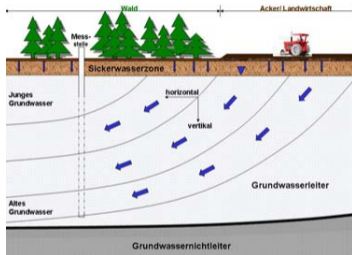
Zeitplan: Hydroinformatik II - SoSe 2024

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

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)



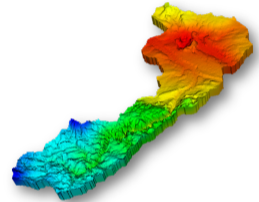
- Mathematik

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

- 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)

Rückblick: Letzte Veranstaltung

Gerinnehydraulik

- Übung
- Jupyter-Notebooks (Ausblick)

Jupyter Notebook für Gerinne-Hydraulik-Übung

Hydroinformatik II - Hydroinformatik II

OlafKolditz / Hydroinformatik-II

Hydroinformatik-II / BHYWI-08-11 / EX11-jupyter-notebook-c++2.ipynb

OlafKolditz update EX11

358 lines (358 sloc) 39.1 KB

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 EX10 - Open channel flow

This figure is summarizing and illustrating the concept of the lecture "Hydroinformatics II"

$$\frac{d\psi}{dt} = \frac{\partial\psi}{\partial t} + v^E \nabla\psi$$

Numerische Methoden

Basics Mechanik

Anwendung

- ▶ Jupyter installation: online Tutorial <https://www.overleaf.com/read/vyxbhdmfczpf>
- ▶ C++ compiler integration: `xeus-cling`
- ▶ Running executable programs from notebooks
- ▶ Markdown descriptions: LaTeX

Exercises:

- ▶ `..\Hydroinformatik-II\BHYWI-08-11`
- ▶ `..\Hydroinformatik-II\EX10-gerinne-python`

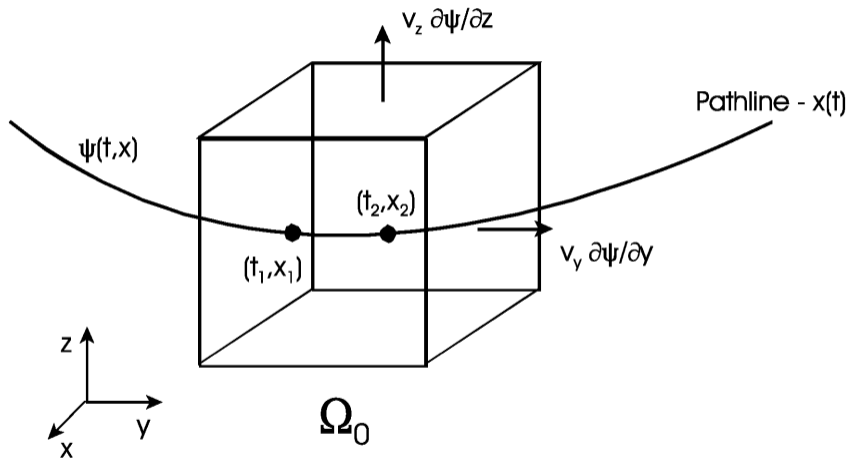
Grundwasserströmung

Einführung

- Theorie
- Prinzipbeispiel

- ▶ Grundlagen (kurze Wiederholung)
- ▶ Grundwassergleichung
- ▶ Prinzip-Beispiel
- ▶ Bilanzierung
- ▶ Berechnungsverfahren
- ▶ Lösung
- ▶ Übung HSA1: Programmierung für Prinzip-Beispiel

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

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

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

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

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.

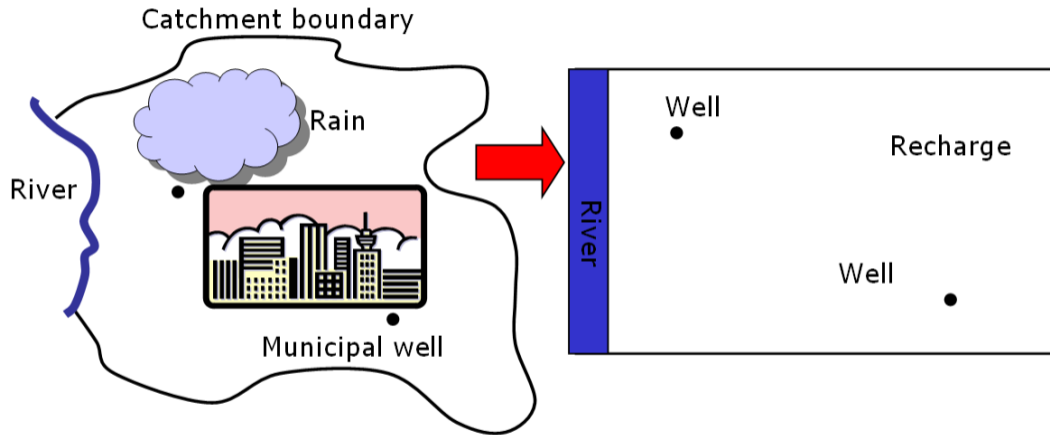
$$S \frac{\partial h}{\partial t} - \frac{\partial}{\partial x} \left(K_x \frac{\partial h}{\partial x} \right) - \frac{\partial}{\partial y} \left(K_y \frac{\partial h}{\partial y} \right) = Q \quad (13)$$

Grundwasserströmung

Einführung

- Prinzipbeispiel

Prinzip-Beispiel



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
 $R = 1.0\text{e-}8 \text{ m}^3/(\text{m}^2\text{s})$

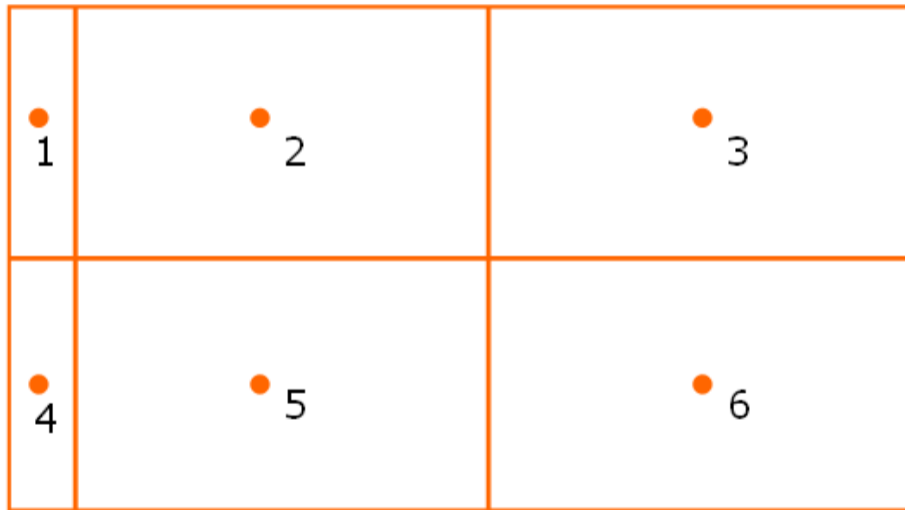
- Pumping well
 $Q_{P6} = 0.005 \text{ m}^3/\text{s}$

No flow – water divide

No flow - streamline

Fig.: Definition der Randbedingungen

Prinzip-Beispiel



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

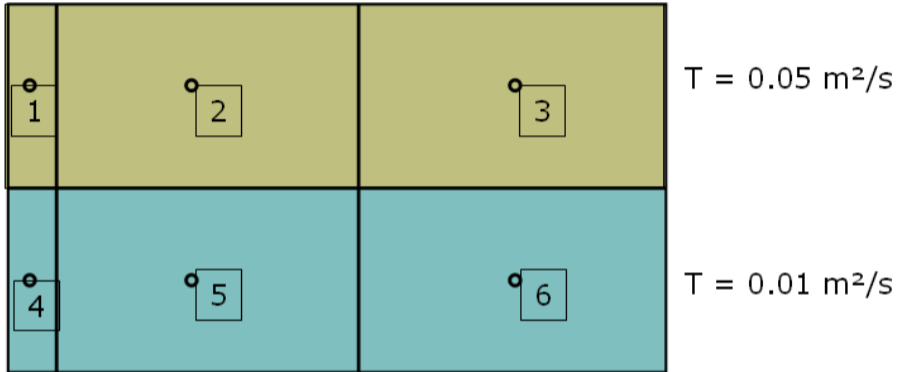


Fig.: Definition der Materialgruppen

$$T = \frac{K}{S}$$

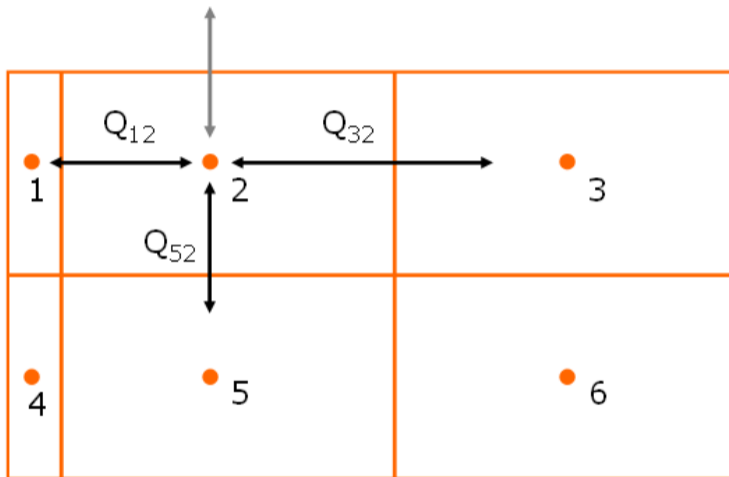


Fig.: Knoten-Bilanz aufstellen

Wir benutzen folgendes Differenzenschema.

$$\frac{\partial h}{\partial x} = \frac{h_i - h_j}{x_i - x_j} \quad (16)$$

$$\frac{\partial h}{\partial y} = \frac{h_i - h_j}{y_i - y_j} \quad (17)$$

Da unser FD-Gitter weder equidistant und unser Aquifer noch heterogen ist, schreiben wir besser.

$$\frac{\partial h}{\partial x} \Big|_{12} = \frac{h_1 - h_2}{\Delta x_1/2 + \Delta x_2/2} \quad (18)$$

- ▶ Berechnung von (hydraulischen) Widerständen - harmonisches Mittel

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2}$$

- ▶ Transmissivität

$$T_{25} = \frac{\Delta y_2 + \Delta y_5}{\Delta y_2/T_2 + \Delta y_5/T_5}$$

$$T_{23} = \frac{\Delta x_2 + \Delta x_3}{\Delta x_2/T_2 + \Delta x_3/T_3}$$

$$Q_x = \Delta y T_x \frac{\partial h}{\partial x}$$

Damit können wir für die Flussterme schreiben.

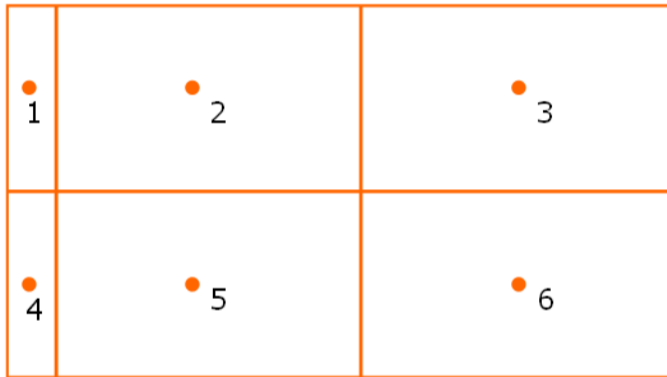
$$Q_{12} = \Delta y_1 \frac{\Delta x_1 + \Delta x_2}{\Delta x_1/T_1 + \Delta x_2/T_2} \times \frac{h_1 - h_2}{\Delta x_1/2 + \Delta x_2/2} \quad (19)$$

$$Q_{52} = \Delta x_2 \frac{\Delta y_5 + \Delta y_2}{\Delta y_5/T_5 + \Delta y_2/T_2} \times \frac{h_5 - h_2}{\Delta y_5/2 + \Delta y_2/2} \quad (20)$$

$$Q_{32} = \Delta x_2 \frac{\Delta x_3 + \Delta x_2}{\Delta x_3/T_3 + \Delta x_2/T_2} \times \frac{h_3 - h_2}{\Delta x_3/2 + \Delta x_2/2} \quad (21)$$

► Tafelbild

Prinzip-Beispiel



$$\Delta x_1 = 100 \text{ m}$$

$$\Delta x_2 = 1000 \text{ m}$$

$$\Delta x_3 = 1000 \text{ m}$$

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

Die Zahlen eingesetzt ergibt sich für

$$Q_{12} = 0.454545 - 0.0454545h_2$$

$$Q_{52} = 0.033333h_5 - 0.033333h_2$$

$$Q_{32} = 0.02500h_3 - 0.02500h_2$$

$$Q_R = R\Delta x_2\Delta y_1 = 0.005$$

$$Q_{P2} = -0.001 \quad (22)$$

$$Q_{12} + Q_{32} + Q_{52} - Q_R - Q_{P2} = 0 \quad (23)$$

- ▶ Bilanzgleichungen für alle Zellen (2,3,4,5):

$$\begin{aligned} 2 &: 0.458545 - 0.103788h_2 + 0.025h_3 + 0.03333h_5 = 0 \\ 3 &: 0.0050 + 0.0250h_2 - 0.0583h_3 + 0.0333h_6 = 0 \\ 5 &: 0.0959 + 0.0333h_2 - 0.0474h_3 + 0.0050h_6 = 0 \\ 6 &: 0.0000 + 0.0333h_3 + 0.0050h_5 - 0.0383h_6 = 0 \end{aligned} \quad (24)$$

- ▶ Gleichungssystem lösen

$$\mathbf{Ax} = \mathbf{b} \quad (25)$$

Ergebnis:

$$h_1 = 10.00$$

$$h_2 = 10.24$$

$$h_3 = 10.41$$

$$h_4 = 10.00$$

$$h_5 = 10.31$$

$$h_6 = 10.39$$

(26)

(27)

- ▶ Gauss-Seidel Verfahren
- ▶ Umstellung des Gleichungssystems

$$h_2 = 0.2408h_3 + 0.3211h_5 + 4.4181$$

$$h_3 = 0.4285h_2 + 0.5714h_6 + 0.0857$$

$$h_5 = 0.7028h_2 + 0.1054h_6 + 2.0223$$

$$h_6 = 0.8695h_3 + 0.1304h_5$$

- ▶ Konstruktion eines iterativen Lösungsverfahrens
- ▶ Pro: Es muss kein Gleichungssystem gelöst werden.
- ▶ Con: Es kann auch mal nicht klappen (keine Konvergenz).

$$h_{2,i+1} = 0.2408h_{3,i} + 0.3211h_{5,i} + 4.4181$$

$$h_{3,i+1} = 0.4285h_{2,i} + 0.5714h_{6,i} + 0.0857$$

$$h_{5,i+1} = 0.7028h_{2,i} + 0.1054h_{6,i} + 2.0223$$

$$h_{6,i+1} = 0.8695h_{3,i} + 0.1304h_{5,i}$$

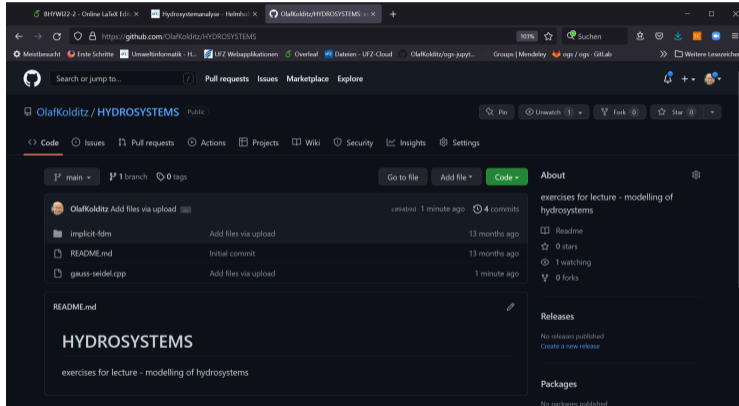
```
1 void Dialog::GaussSeidel()  
2 {  
3     for(int k=0;k<solver_iterations;k++)  
4     {  
5         x[1] = 0.2408 * x[2] + 0.3211 * x[4] + 4.4181;  
6         x[2] = 0.4285 * x[1] + 0.5714 * x[5] + 0.0857;  
7         x[4] = 0.7028 * x[1] + 0.1054 * x[5] + 2.0223 ;  
8         x[5] = 0.8695 * x[2] + 0.1304 * x[4];  
9         TestOutput(x);  
10    }  
11 }
```

Listing: C++ code for Gauss-Seidel solver

Übung

- Prinzipbeispiel Grundwassermodell

Übersicht der Lehrveranstaltung: Übungen (github)



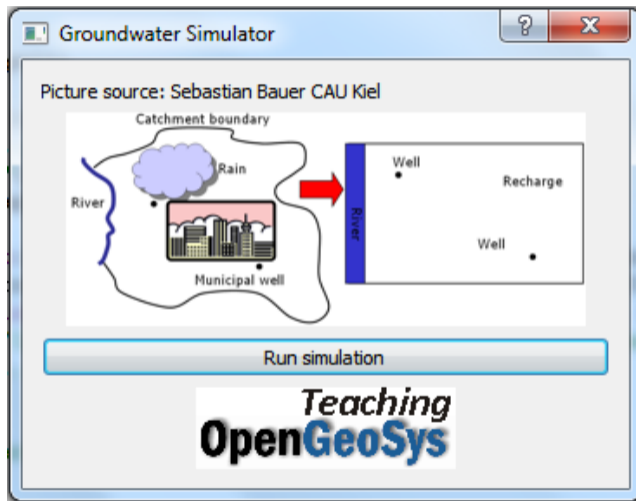
Link:

<https://github.com/OlafKolditz/HYDROSYSTEMS>

- ▶ git clone
- ▶ git fetch -all
- ▶ git pull

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

Übung: Prinzipbeispiel Grundwassermodell



- ▶ Gauss Eliminierungsverfahren (direkte Methode)
- ▶ Gauss-Seidel Verfahren (iterative Methode)

Übung: Prinzipbeispiel Grundwassermodell

BHYWI-22-E1-Qt-Grundwasser-Prinzip

```
1 void Dialog::on_pushButtonRUN_clicked()
2 {
3     switch(solver_method)
4     {
5         case 0: //Gauss
6             AssembleEQS(); //assemble equation system
7             TestOutput(A,b);
8             Gauss(A,b,x,n); //solve EQS via Gauss
9             break;
10        case 1: //Gauss-Seidel
11            //CalculateFluxes();
12            GaussSeidel();
13            msgBox.setText("Gauss-Seidel method finished, \n results in out.txt");
14            break;
15        case 3: // neues Verfahren
16            msgBox.setText("Neues Verfahren vorbereitet");
17            break;
18    }
19 }
```

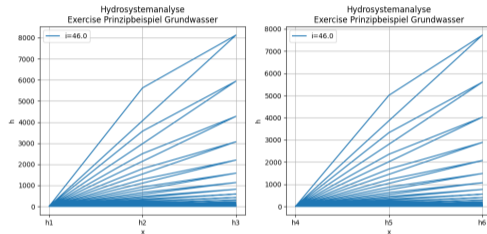
Listing: C++ code for solver selection

Übung: Prinzipbeispiel Grundwassermodell

C++ und Python

```
1 //C++ Programm kompilieren
2 g++ gauss-seidel-plt.cpp
3
4 //Programm ausführen
5 a.exe
6
7 //Ergebnisse mit Python plotten
8 python data_from_file_txt2.py
```

Listing: C++/Python code



Exercise (github repository)

..\HYDROSYSTEMS\grundwasser-prinzip\