

# Modellierung von Hydrosystemen - SoSe 2023

## BHYWI-22-B2-T1.4: Finite-Differenzen-Methode: Implizit

Olaf Kolditz, Lars Bilke, Karsten Rink, Haibing Shao, Erik Nixdorf

<sup>1</sup>Helmholtz Centre for Environmental Research – UFZ, Leipzig

<sup>2</sup>Technische Universität Dresden – TUD, Dresden

<sup>3</sup>Center for Advanced Water Research – CAWR

<sup>4</sup>TUBAF-UFZ Center for Environmental Geosciences – C-EGS, Freiberg / Leipzig

<sup>4</sup>Bundesanstalt für Geowissenschaften und Rohstoffe – BGR, Hannover / Berlin

Dresden, 16.06.2023

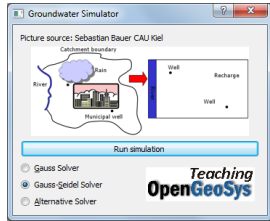
# Zeitplan: Modellierung von Hydrosystemen: Zweiter Block (B2)

Sommersemester 2023: BHYWI-22-B2

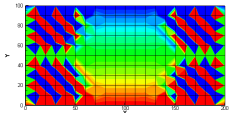
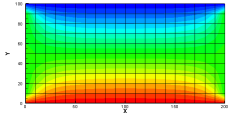
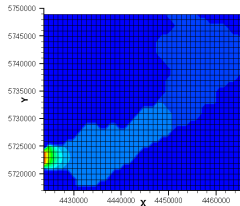
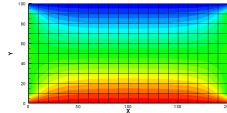
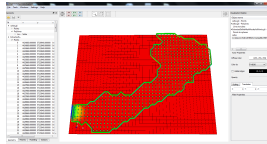
Datum	B2	Thema	Format
16.06.2023	B2-T1.0	Einführung in die Veranstaltung (B2) (Kolditz)	HSZ/403
16.06.2023	B2-T1.1	Hydromechanik und Numerische Methoden (Kolditz)	HSZ/403
16.06.2023	B2-T1.2	Grundwasserhydraulik und Prinzipbeispiel (Kolditz)	HSZ/403
16.06.2023	B2-T1.3	Finite-Differenzen-Methode: Explizit (Kolditz)	HSZ/403
16.06.2023	B2-T1.4	Finite-Differenzen-Methode: Implizit (Kolditz)	HSZ/403
23.06.2023	B2-T2.1	Regionale Grundwassersysteme (Nixdorf)	HSZ/403
23.06.2023	B2-T2.2	Regionale Grundwassersysteme (Nixdorf)	HSZ/403
23.06.2023	B2-T2.3	Regionale Grundwassersysteme (Nixdorf): Übung	HSZ/403
30.06.2023	B2-T3.1	Stofftransport in Hydrosystemen (Shao/Chen)	HSZ/403
30.06.2023	B2-T3.2	Stofftransport in Hydrosystemen (Shao/Chen)	HSZ/403
30.06.2023	B2-T3.3	Stofftransport in Hydrosystemen (Shao/Chen)	GER/37/H
07.07.2023	B2-T4.1	Virtuelle VISLAB Tour - Vorlesung (Rink/Bilke)	VISLAB UFZ
07.07.2023	B2-T4.2	Virtuelle VISLAB Tour - Demo (Rink/Bilke)	VISLAB UFZ
14.07.2023	B2-T1.6	Zusammenfassung der Veranstaltung Numerik (Kolditz)	HSZ/403
14.07.2023	B2-T1.7	Zusammenfassung der Veranstaltung (Hartmann/Kolditz)	HSZ/403
14.07.2023	B2-T1.8	Vorbereitung Klausur (Hartmann/Kolditz)	HSZ/403

# Übersicht: Numerische Verfahren

## explizite FDM

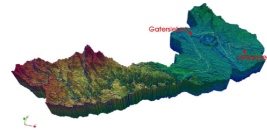
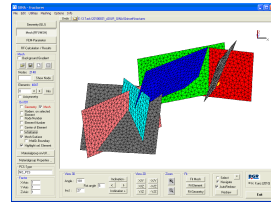


## implizite FDM

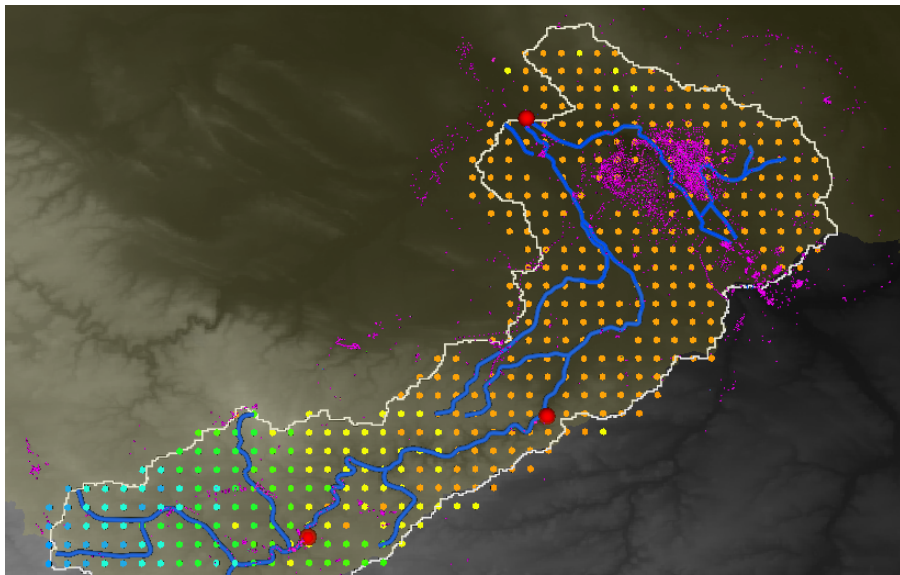


- Pro / Cons
- FDM: einfache Implementierung, starre Geometrien
- FEM: schwieriger zu implementieren (heute), flexible Geometrien

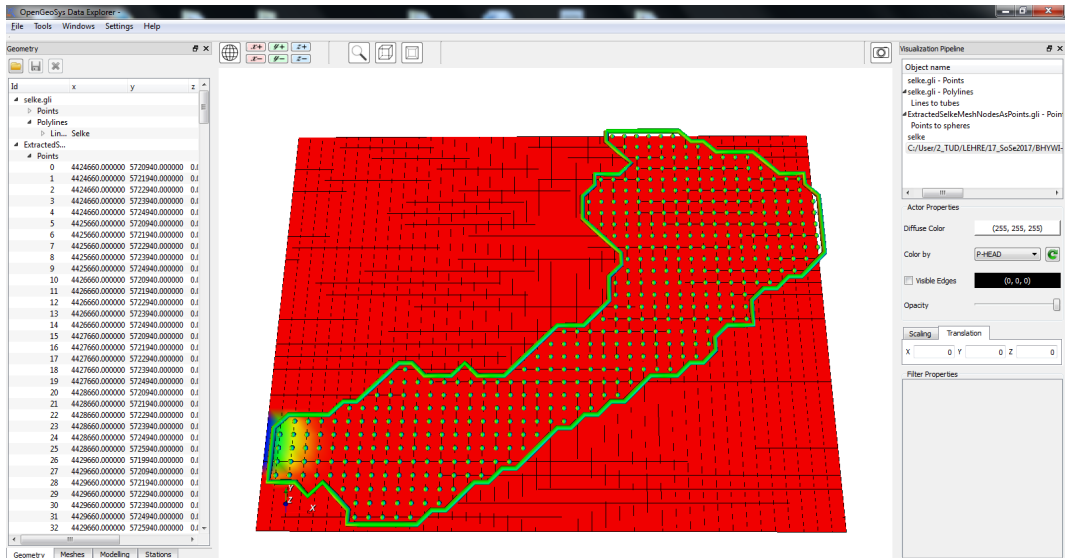
## FEM



# Selke Einzugsgebiet



# Selke Einzugsgebiet



siehe auch Abschn. 4.2 Hydroinformatik II

- ▶ Auswertung der Ableitungen zum neuen Zeitpunkt  $t^{n+1}$

$$\left[ \frac{\partial^2 h}{\partial x^2} \right]_{i,j}^{n+1} \approx \frac{h_{i-1,j}^{n+1} - 2h_{i,j}^{n+1} + h_{i+1,j}^{n+1}}{\Delta x^2} \quad (1)$$

$$\left[ \frac{\partial^2 u}{\partial y^2} \right]_{i,j}^{n+1} \approx \frac{h_{i,j-1}^{n+1} - 2h_{i,j}^{n+1} + h_{i,j+1}^{n+1}}{\Delta y^2} \quad (2)$$

► Differenzen-Schema

$$S_{i,j} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} - K_{i,j}^x \frac{u_{i+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i-1,j}^{n+1}}{\Delta x^2} - K_{i,j}^y \frac{u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}}{\Delta y^2} = Q_{i,j} \quad (3)$$

► Gleichungssystem

$$\begin{aligned} & \left( \frac{S}{\Delta t} + 2\frac{K^x}{\Delta x^2} + 2\frac{K^y}{\Delta y^2} \right) u_{i,j}^{n+1} \\ & - \left( \frac{K^x}{\Delta x^2} \right) (u_{i-1,j}^{n+1} + u_{i+1,j}^{n+1}) - \left( \frac{K^y}{\Delta y^2} \right) (u_{i,j-1}^{n+1} + u_{i,j+1}^{n+1}) \\ & = \frac{S}{\Delta t} u_{i,j}^n + Q_{i,j} \end{aligned} \tag{4}$$



Wir vereinfachen die Gleichung (4), indem wir für den Moment annehmen, dass  $K^x = K^y = K$  (Isotropie) und  $\Delta x = \Delta y = \Delta l$  (gleichförmige Diskretisierung). Die Multiplikation mit  $\Delta t/S$  ergibt dann folgende Beziehung.

$$\begin{aligned} & \left(1 + 4 \frac{K \Delta t}{S \Delta l^2}\right) u_{i,j}^{n+1} \\ & - \left(\frac{K \Delta t}{S \Delta l^2}\right) (u_{i-1,j}^{n+1} + u_{i+1,j}^{n+1} + u_{i,j-1}^{n+1} + u_{i,j+1}^{n+1}) \\ & = u_{i,j}^n + \frac{\Delta t}{S} Q_{i,j} \end{aligned} \tag{5}$$

**K** : Vergleichen Sie die Beziehung (5) mit der Gleichung (4.10, Skript Hydroinformatik II).

Der Ausdruck  $K/S = \alpha$  entspricht dem Diffusivitätskoeffizienten (Überprüfen sie dies anhand der Einheiten). Damit ist die Neumann-Zahl

$$Ne = \frac{K \Delta t}{S \Delta l^2} \quad (6)$$

Nun versuchen wir anhand der Gleichung (5) die Struktur des zu lösenden Gleichungssystems zu beschreiben. Wir gehen wieder ganz genau so vor wie bei der 1D Diffusionsgleichung im Abschn. 4.2 (Hydroinformatik II).

# 2D implizite FDM - Gleichungssystem

$$\underbrace{\begin{bmatrix}
 1 + 4Ne & -Ne & & & & & & & \\
 -Ne & \dots & \dots & & & & & & \\
 & \dots & \dots & \dots & & & & & \\
 & & & -Ne & 1 + 4Ne & & & & \\
 & & & & & 1 + 4Ne & -Ne & & \\
 & & & & & & \dots & \dots & \\
 & & & & & & & -Ne & 1 + 4Ne
 \end{bmatrix}}_{\mathbf{A}}
 \underbrace{\begin{bmatrix}
 u_{0,0}^{n+1} \\
 u_{1,0}^{n+1} \\
 \dots \\
 u_{l-1,0}^{n+1} \\
 u_{0,1}^{n+1} \\
 \dots \\
 u_{l-1,j-1}^{n+1}
 \end{bmatrix}}_{\mathbf{x}}
 =
 \underbrace{\begin{bmatrix}
 u_{0,0}^n + b_{0,0} \\
 u_{1,0}^n + b_{1,0} \\
 \dots \\
 u_{l-1,0}^n + b_{l-1,0} \\
 u_{0,1}^n + b_{0,1} \\
 \dots \\
 u_{l-1,j-1}^n + b_{l-1,j-1}
 \end{bmatrix}}_{\mathbf{b}}$$

Auch was die Programmierung betrifft, können wir auf unsere Erfahrungen in Hydroinformatik II aufbauen. Es gibt praktisch keinen Unterschied, ob wir es mit einem 1D oder 2D Problem zu tun haben. Wir müssen lediglich aufpassen, dass wir die Indizes richtig zählen.

Wir benutzen die Grundstruktur des objekt-orientierten Programms für das explizite FD Verfahren. Die wesentlichen Unterschiede der impliziten zur expliziten FDM sind, dass wir ein Gleichungssystem aufbauen und lösen müssen.

# 2D implizite FDM - die main function

```
1 #include <iostream>
2 #include "fdm.h"
3 #include <time.h>
4 extern void Gauss(double*,double*,double*,int);
5 int main(int argc, char *argv[])
6 {
7     //-----
8     FDM* fdm = new FDM();
9     fdm->SetInitialConditions();
10    fdm->SetBoundaryConditions();
11    //-----
12    int tn = 2;
13    for(int t=0;t<tn;t++)
14    {
15        fdm->AssembleEquationSystem();
16        Gauss(fdm->matrix,fdm->vecb,fdm->vecx,fdm->IJ);
17        fdm->SaveTimeStep();
18        fdm->OutputResults(t);
19    }
20    //-----
21    fdm->out_file.close();
22    return 0;
23 }
```

Listing 1: OOP main function

Dennoch können wir erstaunlich viel wiederverwenden, bis auf

```
1 fdm->AssembleEquationSystem();  
2 Gauss(fdm->matrix,fdm->vecb,fdm->vecx,fdm->IJ);
```

## Listing 2: Rechenschema

Der Gleichungslöser Gauss ist übrigens genau der gleiche, den wir schon für die Lösung des impliziten FD Verfahrens für die Diffusionsgleichung in Hydroinformatik II benutzt haben.

Der Reihe nach. Die Assemblierfunktion soll das Gleichungssystem (7) aufbauen. Vom Prinzip her das Gleiche wie beim 1D FD Verfahren:

- ▶ Die Hauptdiagonale bekommt den Wert  $1 + 4Ne$ ,
- ▶ die Nebendiagonalen haben den Wert  $-Ne$ .

Dies lässt sich programmtechnisch recht einfach bewerkstelligen (sie erinnern sich, wie wir in einer Doppelschleife, die Hauptdiagonale herausfinden können)

```
1 void FDM::AssembleEquationSystem()
2 {
3     // Matrix entries
4     for(i=0;i<IJ;i++)
5     {
6         vecb[i] = u[i];
7         for(j=0;j<IJ;j++)
8         {
9             matrix[i*IJ+j] = 0.0;
10            if(i==j)
11                matrix[i*IJ+j] = 1. + 4.*Ne;
12            else if(abs((i-j))==1)
13                matrix[i*IJ+j] = - Ne;
14        }
15    }
16    // Incorporate boundary conditions
17    IncorporateBoundaryConditions();
18    // Matrix output
19    WriteEquationSystem();
20 }
```

Listing 3: Gleichungssystem aufbauen



# Übung

- Explizite FDM

`https://github.com/OlafKolditz/HYDROSYSTEMS/tree/main/BHYWI-22-E3\_FDM-implizit-Rechteck-python`

```
1 cd ... \HYDROSYSTEMS \BHYWI-22-E3_FDM-implizit-Rechteck-python
```

## Listing 4: Verzeichnis auswaehlen

run.bat

```
1 set PATH=%PATH%;C:\MinGW\bin  
2 g++ main.cpp fdm.cpp solver.cpp  
3 a.exe  
4 python isolines.py
```

## Listing 5: Skript