

# Modellierung von Hydrosystemen

## ”Numerische und daten-basierte Methoden”

### Wdh. Mechanik, Grundwasserhydraulik-Übung

Olaf Kolditz

\*Helmholtz Centre for Environmental Research – UFZ

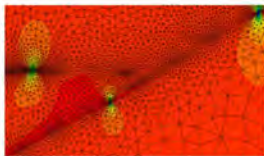
<sup>1</sup>Technische Universität Dresden – TUDD

<sup>2</sup>Centre for Advanced Water Research – CAWR

27.04./04.05.2018 - Dresden

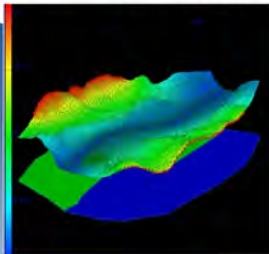
- ▶ Grundlagen (kurze Wiederholung)
- ▶ Grundwassergleichung
- ▶ Prinzip-Beispiel
- ▶ Bilanzierung
- ▶ Berechnungsverfahren
- ▶ Lösung
- ▶ Übung USA1: Programmierung für Prinzip-Beispiel

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

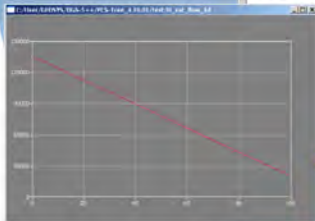
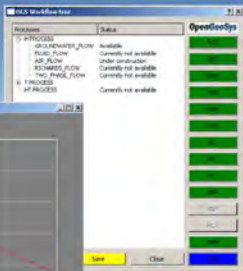


Basics  
Mechanik

Anwendung

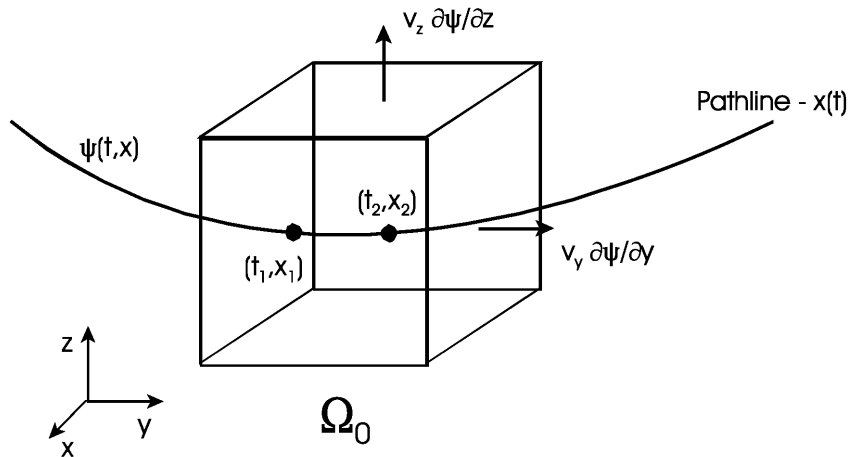


Numerische  
Methoden



Programmierung  
Visual C++

# 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

$$\boldsymbol{\sigma} = -p\mathbf{1} + \boldsymbol{\tau} \quad (3)$$

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

$$0 = \mathbf{f}^e - \frac{1}{\rho} \nabla p + \nu \Delta \mathbf{v} \quad (7)$$

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

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.

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

# Prinzip-Beispiel



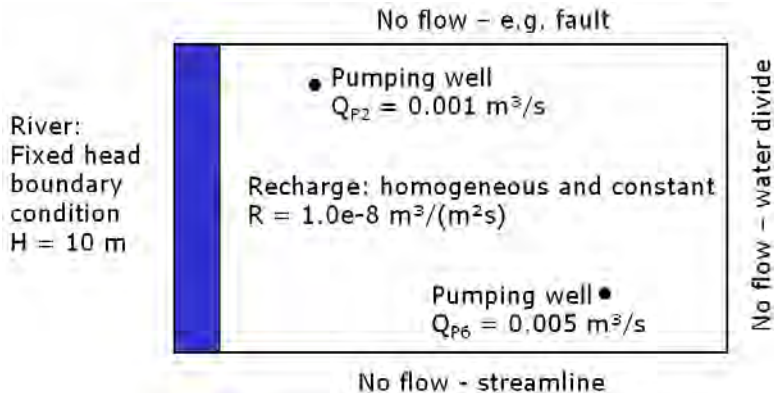
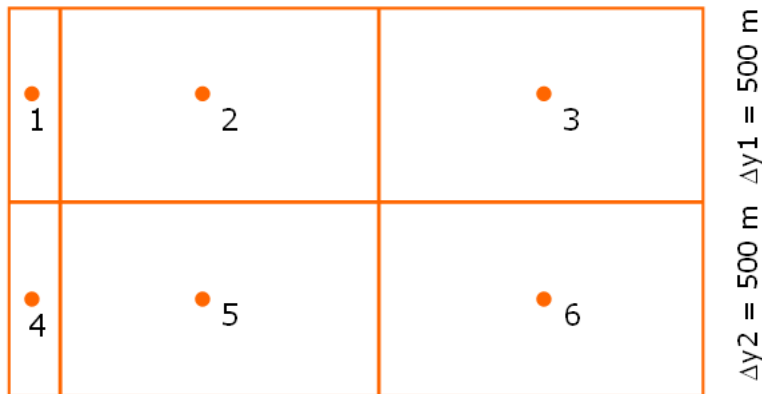


Fig.: Definition der Randbedingungen

# Prinzip-Beispiel



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

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

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

Fig.: Bilanzierungs-Schemata

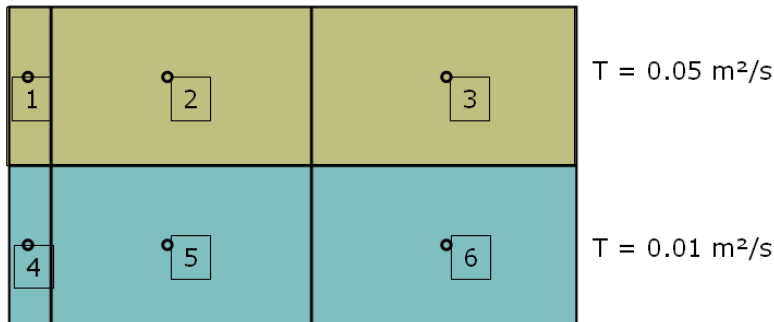


Fig.: Definition der Materialgruppen

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

(14)

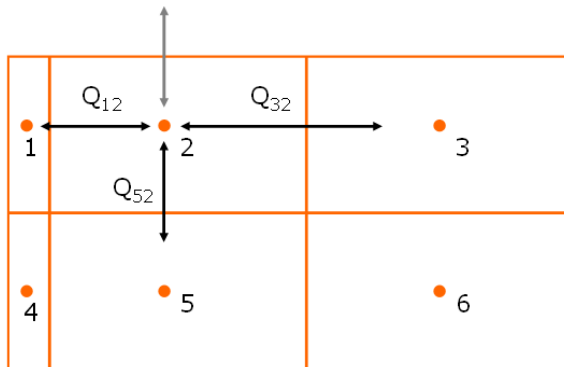


Fig.: Knoten-Bilanz aufstellen

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

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

$$T_{12} = \frac{\Delta x_1 + \Delta x_2}{\Delta x_1 / T_1 + \Delta x_2 / T_2}$$



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

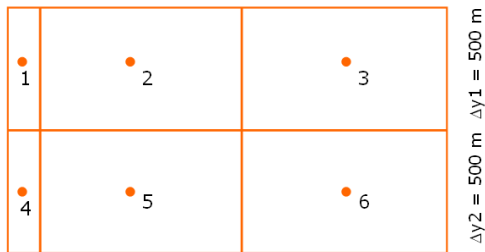
Damit können wir für die Flussterme schreiben.

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

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

- ▶ Tafelbild

# Prinzip-Beispiel



$$\Delta x_1 = 100 \text{ m} \quad \Delta x_2 = 1000 \text{ m} \quad \Delta x_3 = 1000 \text{ m}$$

Die Zahlen eingesetzt ergibt sich für

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

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

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

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

$$Q_{P2} = -0.001$$

- ▶ Bilanzgleichungen für alle Zellen (2,3,5,6):

$$\begin{aligned} 2 : 0.458545 - 0.103788h_2 + 0.025h_3 + 0.03333h_5 &= 0 \\ 3 : 0.0050 + 0.0250h_2 - 0.0583h_3 + 0.0333h_6 &= 0 \\ 5 : 0.0959 + 0.0333h_2 - 0.0474h_3 + 0.0050h_6 &= 0 \\ 6 : 0.0000 + 0.0333h_3 + 0.0050h_5 - 0.0383h_6 &= 0 \end{aligned} \quad (22)$$

- ▶ Gleichungssystem lösen

$$\mathbf{Ax} = \mathbf{b} \quad (23)$$

Ergebnis:

$$h_1 = 10.00$$

$$h_2 = 10.24$$

$$h_3 = 10.41$$

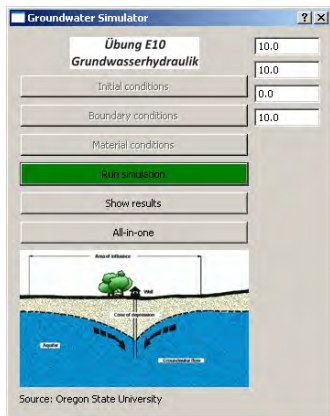
$$h_4 = 10.00$$

$$h_5 = 10.31$$

$$h_6 = 10.39$$

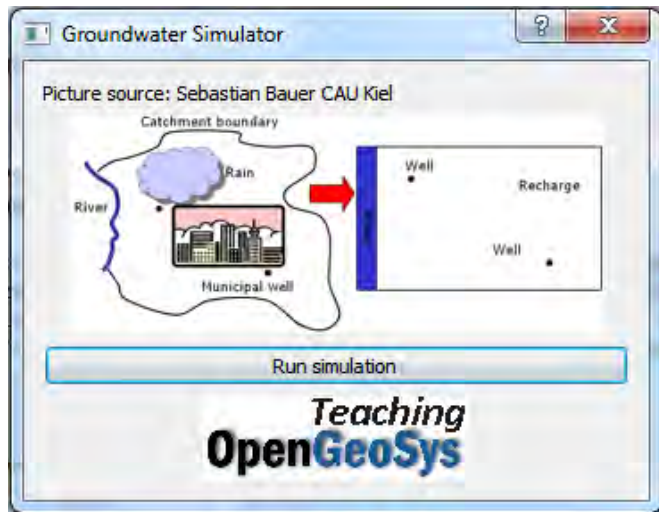
(24)

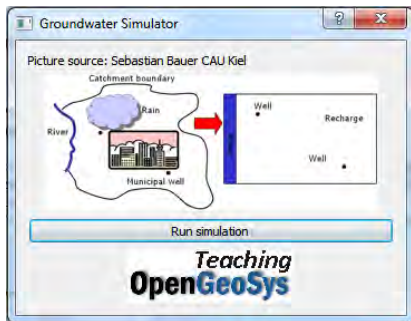
(25)



Eigenes MatLab ...

- ▶ Funktions-Simulator FDM Simulator (explizit und implizit)
- ▶ Newton Simulator
- ▶ ... alles noch 1D, schau'n wir mal (Systemanalyse)
- ▶ 2D FDM Grundwassersimulator



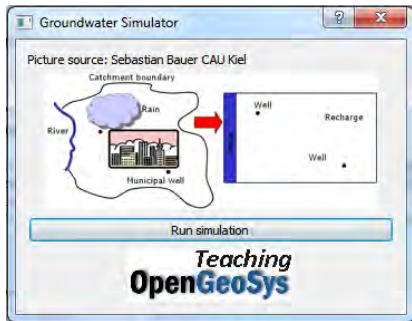


- ▶ Dialog-Gestaltung
- ▶ main Funktion
- ▶ Konstruktor
- ▶ Anfangsbedingungen
- ▶ Randbedingungen
- ▶ Materialparameter
- ▶ Gleichungslöser
- ▶ Ergebnisse plotten
- ▶ ...

# Übung USA#1 - main

```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    //.....
    QPixmap pixmap ("bk-1.png");
    QSplashScreen splash(pixmap);
    splash.show();
    splash.showMessage(QObject::tr("Übung HSA#1.GW wird geladen..."), Qt::black);
    QTime dieTime = QTime::currentTime().addSecs(3); while( QTime::currentTime()
        < dieTime)
        QCoreApplication::processEvents(QEventLoop::AllEvents, 1000);
    // Warte-Funktion (http://lists.trolltech.com/qt-interest/2007-01/thread00133)
    //.....
    Dialog w;
    splash.finish(&w); //test
    w.setWindowTitle("Groundwater Simulator");
    w.setFixedWidth(350);
    w.show();
    //.....
    return a.exec();
}
```





- ▶ Elements
  - ▶ Labels
  - ▶ PushButtons
- ▶ Connect
- ▶ Layout

```
Dialog::Dialog(QWidget *parent) : QDialog(parent)
{
    //elements
    //labels
    QLabel* labelHeader = new QLabel(tr("Picture source: Sebastian Bauer CAU Kiel"));
    QLabel *label_ogs = new QLabel();
    label_ogs->setAlignment(Qt::AlignCenter);
    label_ogs->setPixmap(QPixmap("ogs_teaching_150.png"));
    QLabel *label_exercise = new QLabel();
    label_exercise->setAlignment(Qt::AlignCenter);
    label_exercise->setPixmap(QPixmap("gw1_300.png"));
    //push buttons
    QPushButtonRUN = new QPushButton(tr("Run simulation"));
    //connect
    connect(pushButtonRUN,SIGNAL(clicked()),this,SLOT(on_pushButtonRUN_clicked()));
    //layout
    QVBoxLayout *mainLayout = new QVBoxLayout;
    mainLayout->addWidget(labelHeader);
    mainLayout->addWidget(label_exercise);
    mainLayout->addWidget(pushButtonRUN);
    mainLayout->addWidget(label_ogs);
    setLayout(mainLayout);
}
```

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

$$\begin{aligned} 2 : 0.458545 - 0.103788h_2 + 0.025h_3 + 0.03333h_5 &= 0 \\ 3 : 0.0050 + 0.0250h_2 - 0.0583h_3 + 0.0333h_6 &= 0 \\ 5 : 0.0959 + 0.0333h_2 - 0.0474h_3 + 0.0050h_6 &= 0 \\ 6 : 0.0000 + 0.0333h_3 + 0.0050h_5 - 0.0383h_6 &= 0 \end{aligned} \quad (26)$$

- ▶ Gleichungssystem lösen

$$\mathbf{Ax} = \mathbf{b} \quad (27)$$

```
void Dialog::on_pushButtonRUN_clicked()
{
    double* a;
    a = new double[6*6];
    // h1 is BC
    a[0] = 1.; //a11
    a[1] = 0.; //a12
    a[2] = 0.; //a13
    a[3] = 0.; //a14
    a[4] = 0.; //a15
    a[5] = 0.; //a16
    // h2 to be calculated
    a[6] = 0.; //a21
    a[7] = -0.103788; //a22
    a[8] = 0.025; //a23
    a[9] = 0.; //a24
    a[10] = 0.0333; //a25
    a[11] = 0.; //a26
    ...
}
```

```
void Dialog::TestOutput(double*a,double*b)
{

    for(int i=0;i<6;i++)
    {
        for(int j=0;j<6;j++)
        {
            out_file << a[6*i+j] << "\t";
        }
        out_file << " : " << b[i] << endl;
    }
}
```

# Übung USA#1 - Gleichungssystem schreiben

```
1 0 0 0 0 0 : 10
0 -0.103788 0.025 0 0.0333 0 : -0.459
0 -0.240876 -0.0522781 0 0.00802116 0.0333 : -0.005
0 -0 -0 1 0 0 : 10
0 -0.320846 -0.153432 0 -0.0354851 0.0101093 : -0.0959
0 -0 -0.636978 0 -0.284889 -0.358909 : 0
```

```
out_file.precision(3);
```

```
1      0      0 0      0      0 : 10
0 -0.104  0.025 0  0.0333      0 : -0.459
0 -0.241 -0.0523 0 0.00802 0.0333 : -0.005
0      -0      -0 1      0      0 : 10
0 -0.321 -0.153 0 -0.0355 0.0101 : 0.0959
0      -0  -0.637 0  -0.285 -0.359 : 0
```

```
double x[6];  
Gauss(a,b,x,6);  
TestOutput(x);
```

h0:10

h1:7.68

h2:3.62

h3:10

h4:7.47

h5:0.412

- ▶ Das stimmt doch gar nicht!

# Übung USA#1 - Polymorphismus

```
void Dialog::TestOutput(double*x)
{
    for(int i=0;i<6;i++)
    {
        out_file << "h" << QString::number(i).toString() << ":" << x[i]
    }
}
```

```
void Dialog::TestOutput(double*a,double*b)
{

    for(int i=0;i<6;i++)
    {
        for(int j=0;j<6;j++)
        {
            out_file << a[6*i+j] << "\t";
        }
        out_file << " : " << b[i] << endl;
    }
}
```



- ▶ 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).

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

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

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


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

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

# Übung USA#1 - Alternative Solver

Groundwater Simulator

Picture source: Sebastian Bauer CAU Kiel



Catchment boundary

Rain

River

Municipal well

Well

Recharge

Well

Run simulation

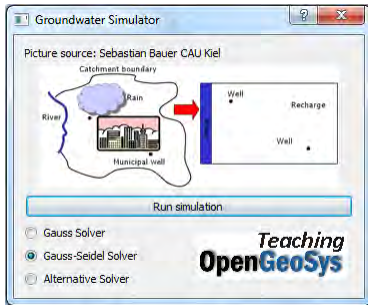
Gauss Solver

Gauss-Seidel Solver

Alternative Solver

**Teaching  
OpenGeoSys**

# Übung USA#1 - Alternative Solver



- ▶ Layout (Tafelbild)
- ▶ QBasics: Radio-Buttons

# Übung USA#1 - Layout

```
//layout
QVBoxLayout *mainLayout = new QVBoxLayout;
QVBoxLayout *upperLayout = new QVBoxLayout;
QHBoxLayout *lowerLayout = new QHBoxLayout;
QVBoxLayout *lowerleftLayout = new QVBoxLayout;
upperLayout->addWidget(labelHeader);
upperLayout->addWidget(label_exercise);
upperLayout->addWidget(pushButtonRUN);
lowerLayout->addLayout(lowerleftLayout);
lowerleftLayout->addWidget(radio1);
lowerleftLayout->addWidget(radio2);
lowerleftLayout->addWidget(radio3);
lowerleftLayout->addStretch(1);
groupBox->setLayout(lowerleftLayout);
lowerLayout->addWidget(label_ogs);
mainLayout->addLayout(upperLayout);
mainLayout->addLayout(lowerLayout);
setLayout(mainLayout);
```

```
#include <QRadioButton>

//elements
QGroupBox *groupBox = new QGroupBox(tr("Exclusive Radio Buttons"));
QRadioButton *radio1 = new QRadioButton(tr("&Gauss Solver"));
QRadioButton *radio2 = new QRadioButton(tr("Gauss-&Seidel Solver"));
QRadioButton *radio3 = new QRadioButton(tr("&Alternative Solver"));
radio2->setChecked(true);

//connect
connect(radio1,SIGNAL(clicked(bool)),this,SLOT(clickedstate(bool)));
```

```
void Dialog::on_pushButtonRUN_clicked()
{
    switch(solver_method)
    {
        case 0: //Gauss
            AssembleEQS(); //assemble equation system
            TestOutput(A,b);
            Gauss(A,b,x,n); //solve EQS via Gauss
            break;
        case 1: //Gauss-Seidel
            GaussSeidel();
            break;
    }
    TestOutput(x);
    pushButtonRUN->setStyleSheet("background-color: green");
}
```



Iteration step: 0

h0:10

h1:10.037

h2:10.101

h3:10

h4:10.13

h5:10.103

years later ...

Iteration step: 18

h0:10

h1:10.238

h2:10.416

h3:10

h4:10.314

h5:10.402