Modellierung von Hydrosystemen

"Numerische und daten-basierte Methoden"

BHYWI-22-V2 @ 2019

Grundwasserhydraulik-Prinzipbeispiel

Olaf Kolditz

*Helmholtz Centre for Environmental Research – UFZ

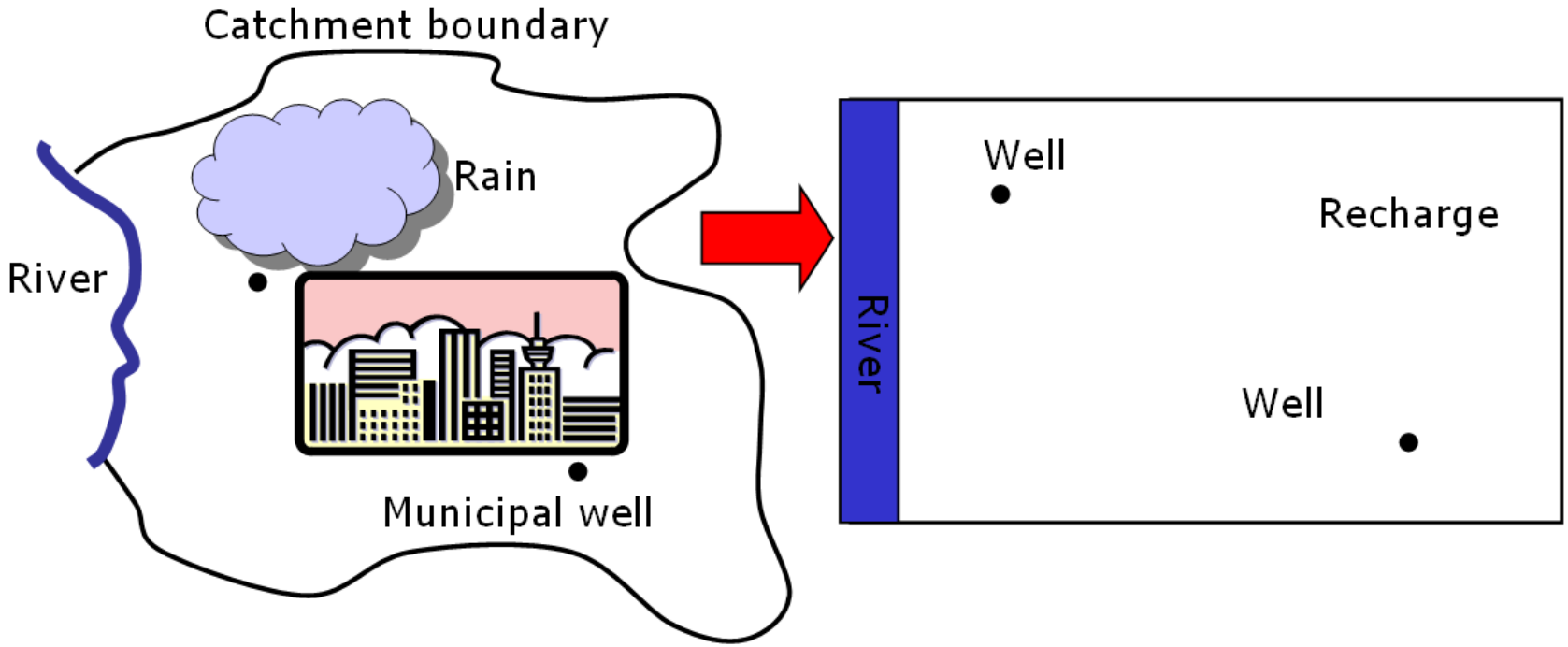
[1]Technische Universität Dresden – TUDD

[2]Centre for Advanced Water Research – CAWR

12.04./19.04.2019 - Dresden

TECHNISCHE UNIVERSITÄT DRESDEN

Prinzip-Beispiel



Quelle: Sebastian Bauer (Uni Kiel)

Prinzip-Beispiel

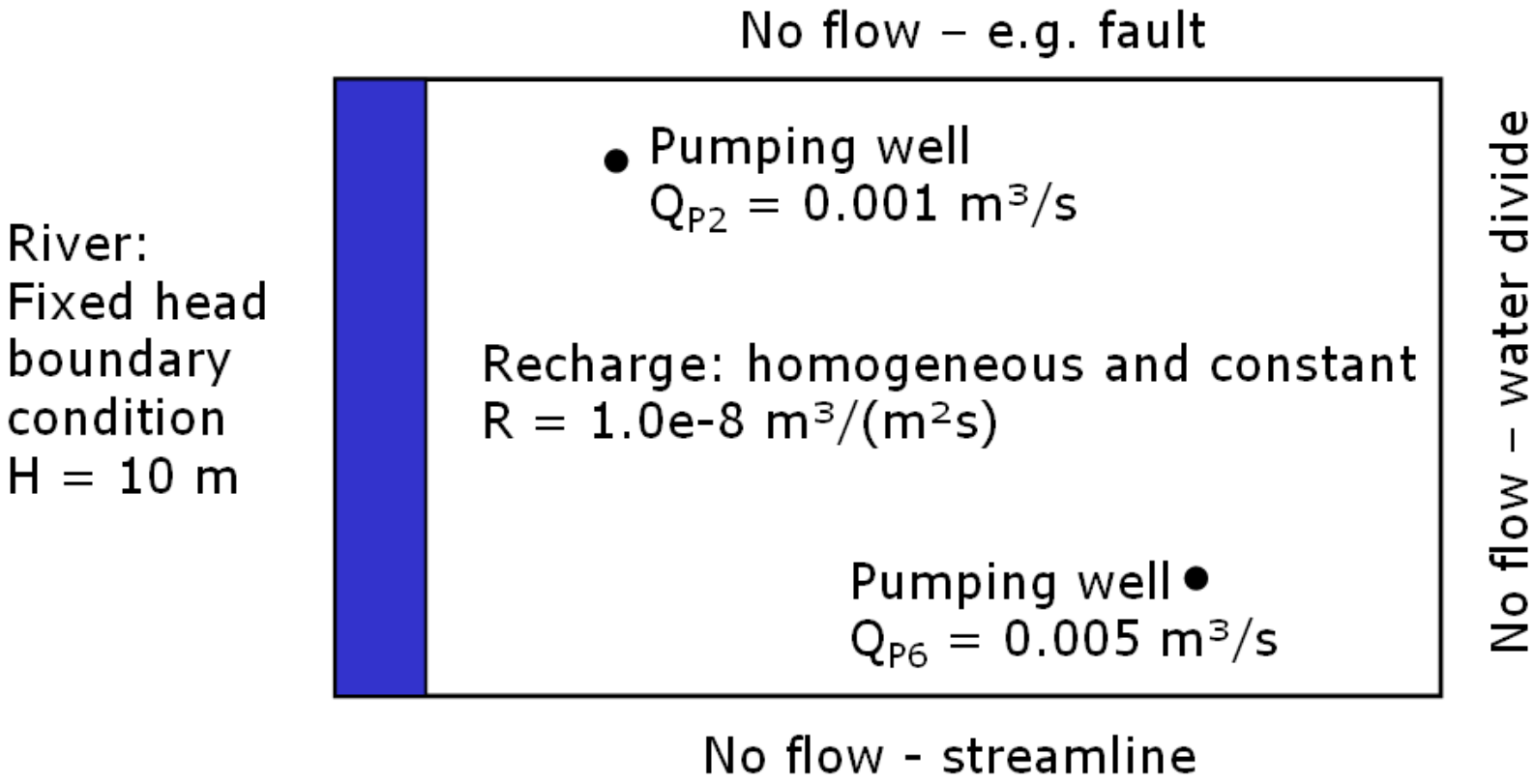


Fig.: Definition der Randbedingungen

Prinzip-Beispiel

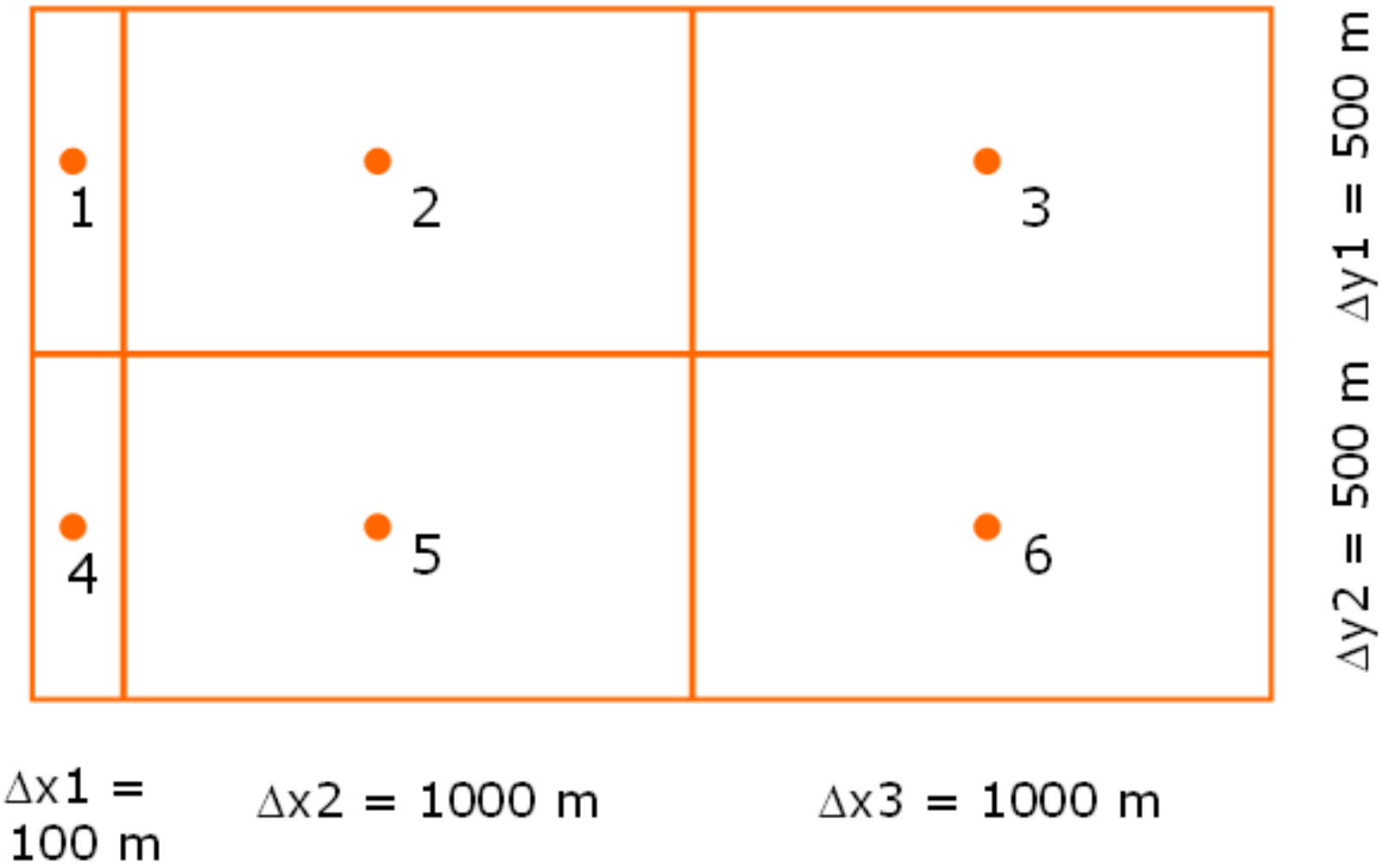


Fig.: Bilanzierungs-Schemata

Prinzip-Beispiel

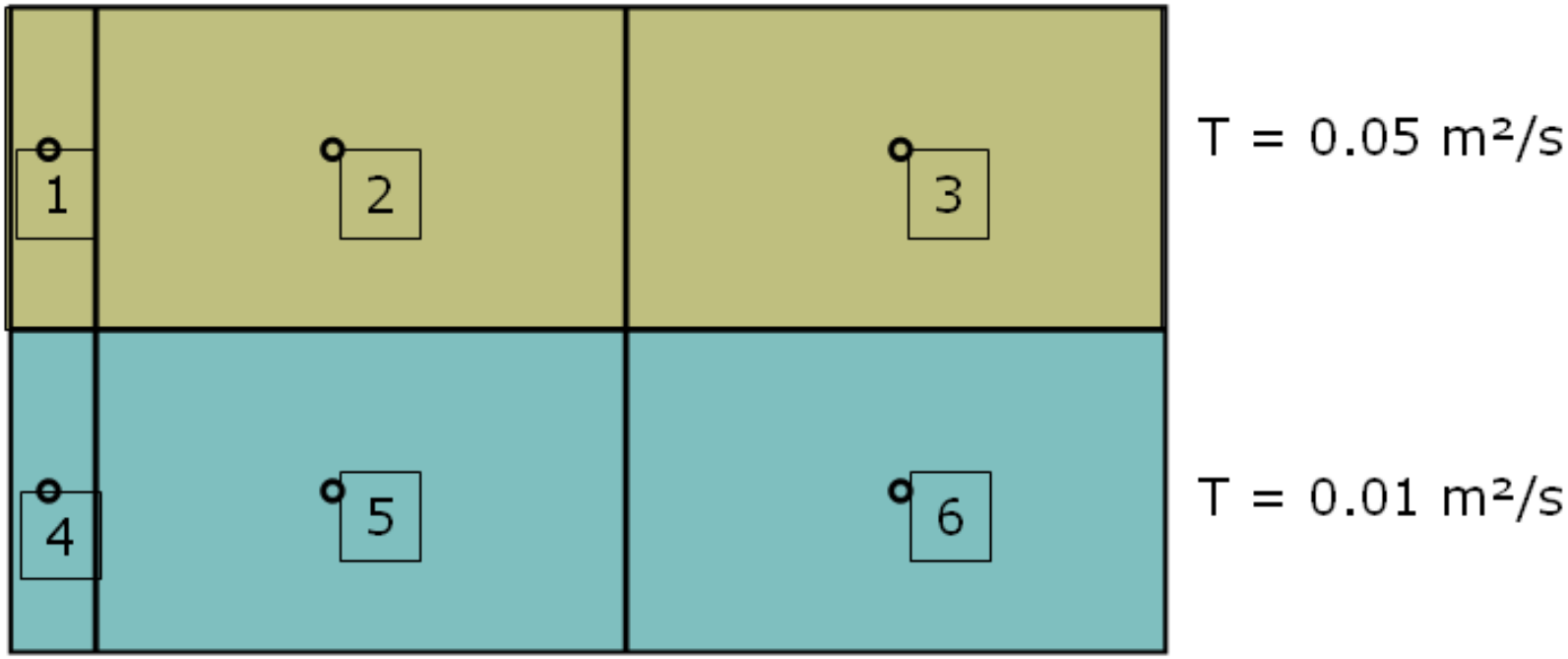


Fig.: Definition der Materialgruppen

$$T = \frac{K}{S} \tag{1}$$

TECHNISCHE UNIVERSITÄT DRESDEN

Prinzip-Beispiel

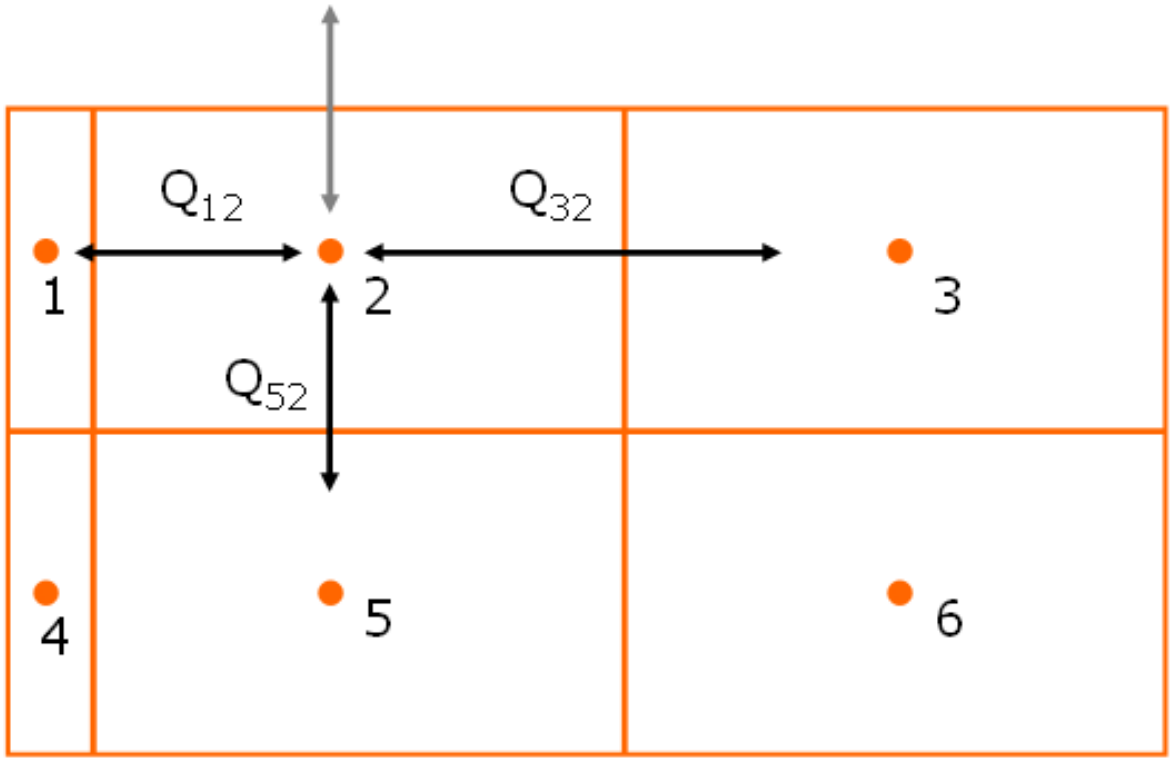


Fig.: Knoten-Bilanz aufstellen

$$Q_{12} + Q_{32} + Q_{52} + Q_R + Q_{P2} = 0 \quad (2)$$

Prinzip-Beispiel

Wir benutzen folgendes Differenzenschema.

$$\frac{\partial h}{\partial x} = \frac{h_i - h_j}{x_i - x_j} \tag{3}$$

$$\frac{\partial h}{\partial y} = \frac{h_i - h_j}{y_i - y_j} \tag{4}$$

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

$$\frac{\partial h}{\partial x}|_{12} = \frac{h_1 - h_2}{\Delta x_1/2 + \Delta x_2/2} \tag{5}$$

TECHNISCHE UNIVERSITÄT DRESDEN

Prinzip-Beispiel - Transmissivität

- Berechnung von (hydraulischen) Widerständen - harmonisches Mittel

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

- Transmissivität

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

TECHNISCHE UNIVERSITÄT DRESDEN

Prinzip-Beispiel

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

Damit können wir für die Flussterme schreiben.

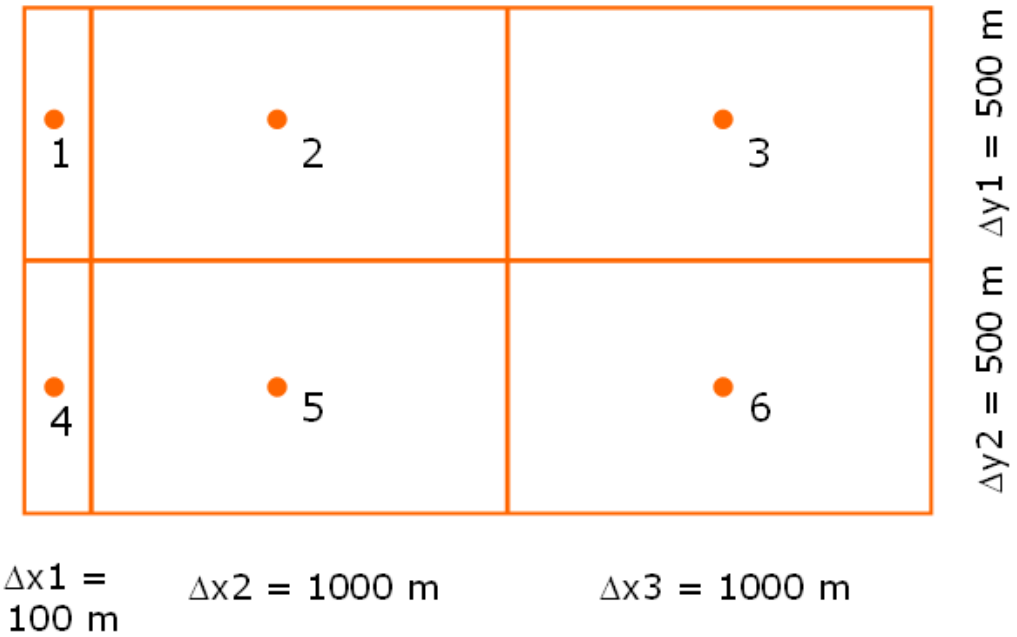
$$Q_{12} = \Delta y_1 \frac{\Delta x_1 + \Delta x_2}{\Delta x_1 / T_1 + \Delta x_2 / T_2} \times \frac{h_1 - h_2}{\Delta x_1 / 2 + \Delta x_2 / 2} \tag{6}$$

$$Q_{52} = \Delta x_2 \frac{\Delta y_5 + \Delta y_2}{\Delta y_5 / T_5 + \Delta y_2 / T_2} \times \frac{h_5 - h_2}{\Delta y_5 / 2 + \Delta y_2 / 2} \tag{7}$$

- Tafelbild

TECHNISCHE UNIVERSITÄT DRESDEN

Prinzip-Beispiel



Die Zahlen eingesetzt ergibt sich für

$$
\begin{aligned}
Q_{12} &= 0.454545 - 0.0454545 h_2 \\
Q_{52} &= 0.03333 h_5 - 0.03333 h_2 \\
Q_{32} &= 0.02500 h_3 - 0.02500 h_2 \\
Q_R &= R \Delta x_2 \Delta y_1 = 0.005 \\
Q_{P2} &= -0.001
\end{aligned}
\tag{8}
$$

Prinzip-Beispiel - Lösungsverfahren

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

$$
\begin{aligned}
2 &: 0.458545 - 0.103788h_2 + 0.025h_3 + 0.03333h_5 = 0 \\
3 &: 0.0050 + 0.0250h_2 - 0.0583h_3 + 0.0333h_6 = 0 \\
5 &: 0.0959 + 0.0333h_2 - 0.0474h_3 + 0.0050h_6 = 0 \\
6 &: 0.0000 + 0.0333h_3 + 0.0050h_5 - 0.0383h_6 = 0
\end{aligned} \tag{9}
$$

- Gleichungssystem lösen

$$\mathbf{Ax} = \mathbf{b} \tag{10}$$

TECHNISCHE UNIVERSITÄT DRESDEN

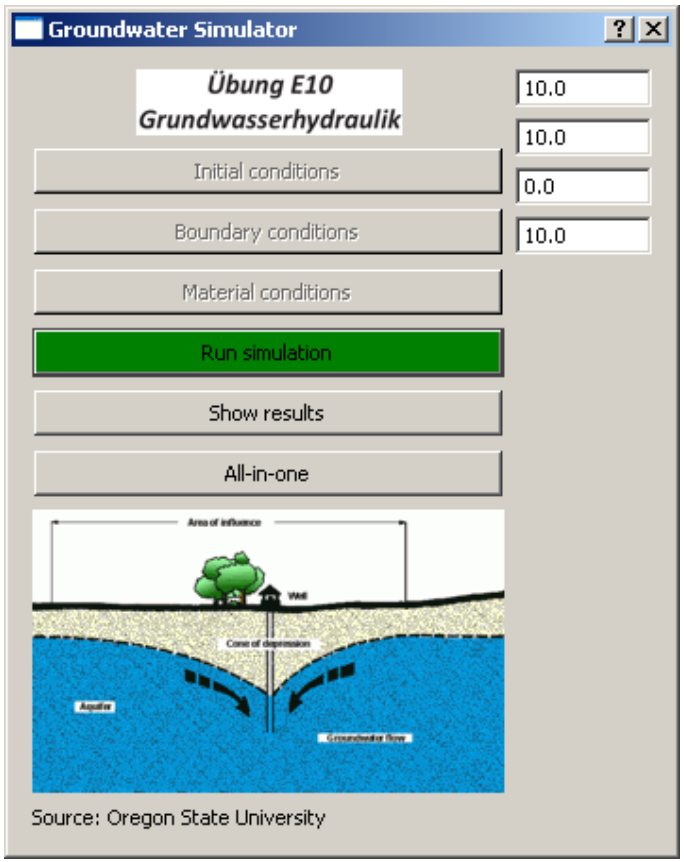
Prinzip-Beispiel - Übung USA#1

Ergebnis:

$$
\begin{aligned}
h_1 &= 10.00 \\
h_2 &= 10.24 \\
h_3 &= 10.41 \qquad (11) \\
h_4 &= 10.00 \\
h_5 &= 10.31 \\
h_6 &= 10.39 \\
& \qquad (12)
\end{aligned}
$$

TECHNISCHE UNIVERSITÄT DRESDEN

Übung USA#1

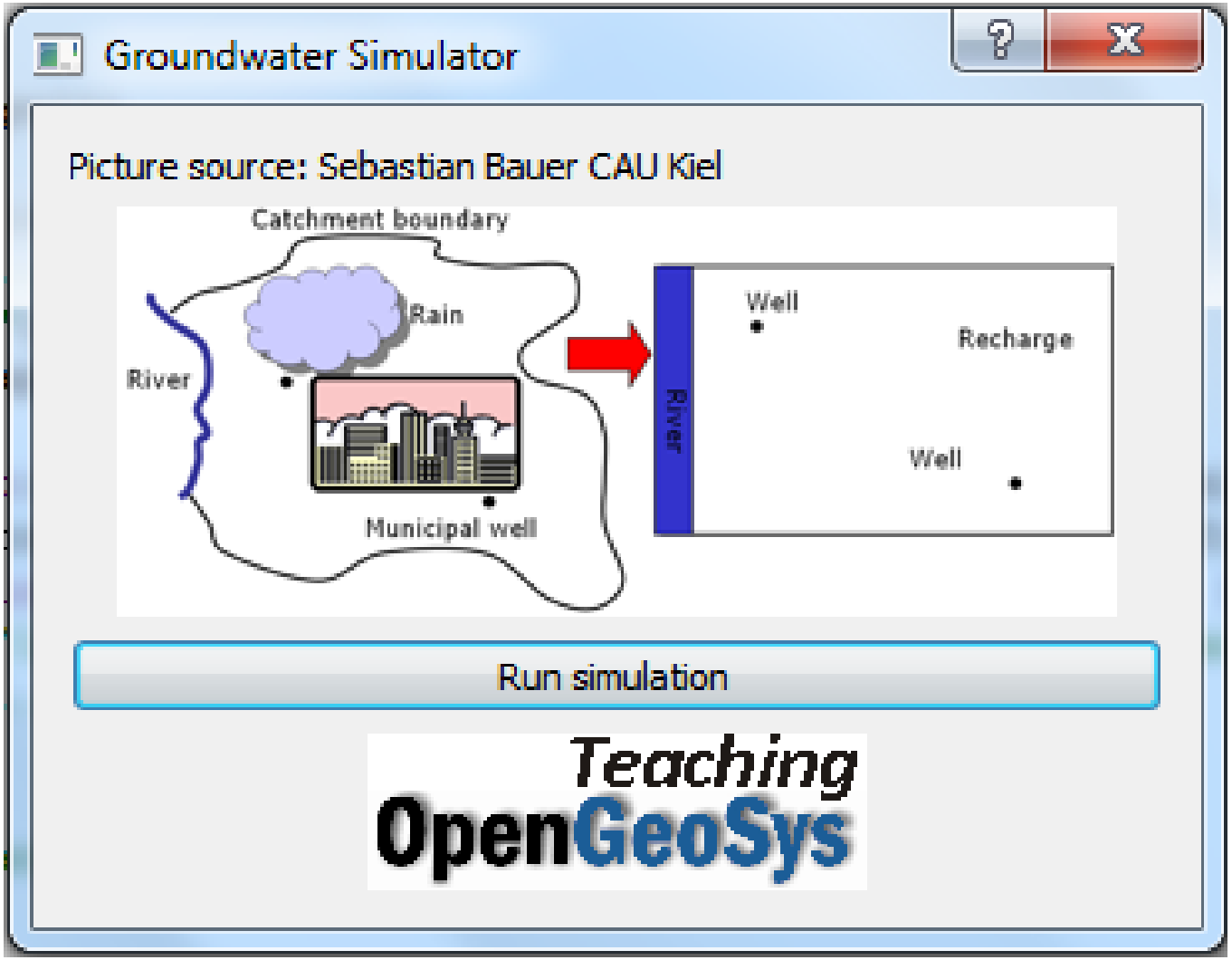


Eigenes MatLab ...

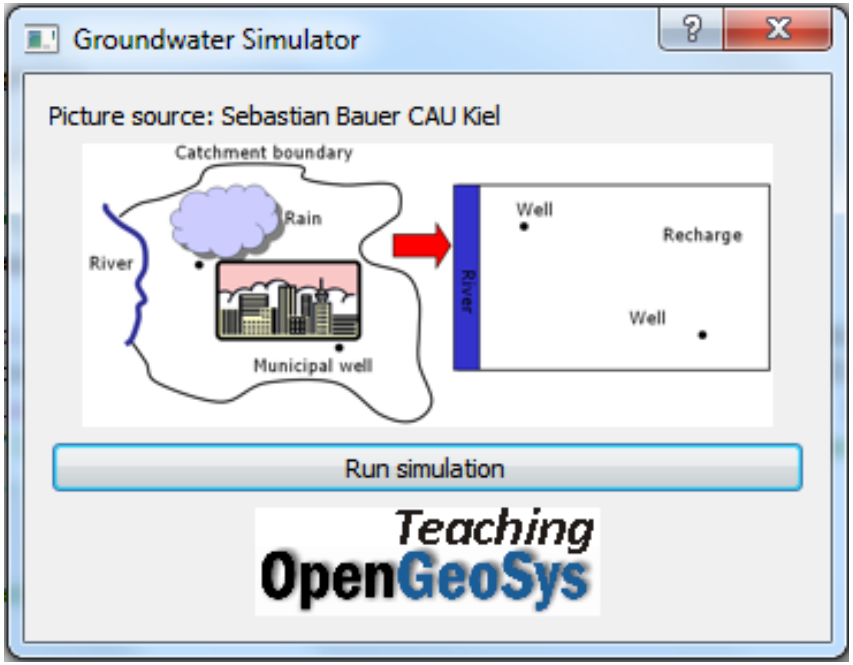
- Funktions-Simulator
- FDM Simulator (explizit und implizit)
- Newton Simulator
- ... alles noch 1D, schau'n wir mal (Systemanalyse)
- 2D FDM Grundwassersimulator

TECHNISCHE UNIVERSITÄT DRESDEN

Übung USA#1


Groundwater Simulator
Picture source: Sebastian Bauer CAU Kiel
Catchment boundary
Rain
River
Municipal well
River
Well
Recharge
Well
Run simulation
Teaching
OpenGeoSys

Übung USA#1 - Reminder



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

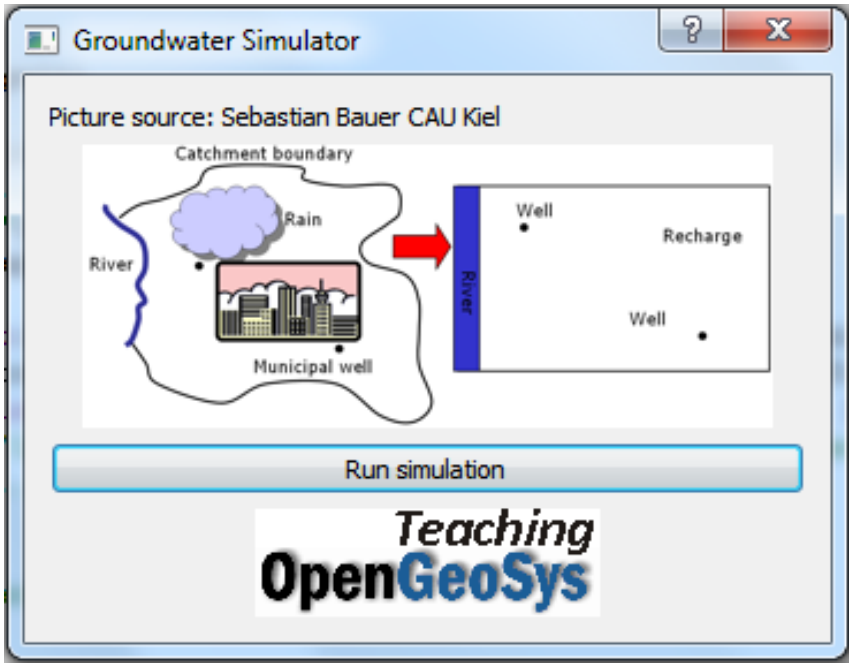
TECHNISCHE UNIVERSITÄT DRESDEN

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()
                  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();
}
```

TECHNISCHE UNIVERSITÄT DRESDEN

Übung USA#1 - Constructor



- Elements
  - Labels
  - PushButtons
- Connect
- Layout

TECHNISCHE UNIVERSITÄT DRESDEN

Übung USA#1 - Constructor

```
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
  pushButtonRUN = 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);
}
```

TECHNISCHE UNIVERSITÄT DRESDEN

Übung USA#1 - Gleichungssystem

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

$$
\begin{aligned}
2 &: 0.458545 - 0.103788h_2 + 0.025h_3 + 0.03333h_5 = 0 \\
3 &: 0.0050 + 0.0250h_2 - 0.0583h_3 + 0.0333h_6 = 0 \\
5 &: 0.0959 + 0.0333h_2 - 0.0474h_3 + 0.0050h_6 = 0 \\
6 &: 0.0000 + 0.0333h_3 + 0.0050h_5 - 0.0383h_6 = 0
\end{aligned}
\tag{13}
$$

- Gleichungssystem lösen

$$
\mathbf{Ax} = \mathbf{b} \tag{14}
$$

TECHNISCHE UNIVERSITÄT DRESDEN

Übung USA#1 - Gleichungssystem lösen

```
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
  ...
```

TECHNISCHE UNIVERSITÄT DRESDEN

Übung USA#1 - Gleichungssystem schreiben

```
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
```

TECHNISCHE UNIVERSITÄT DRESDEN

Übung USA#1 - Gleichungssystem lösen

```
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!

TECHNISCHE UNIVERSITÄT DRESDEN

Übung USA#1 - Polymorphismus

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

TECHNISCHE UNIVERSITÄT DRESDEN

Übung USA#1 - Alternatives Lösungsverfahren

- Gauss-Seidel Verfahren
- Umstellung des Gleichungssystems

$$
\begin{aligned}
h_2 &= 0.2408h_3 + 0.3211h_5 + 4.4181 \\
h_3 &= 0.4285h_2 + 0.5714h_6 + 0.0857 \\
h_5 &= 0.7028h_2 + 0.1054h_6 + 2.0223 \\
h_6 &= 0.8695h_3 + 0.1304h_5
\end{aligned}
$$

TECHNISCHE UNIVERSITÄT DRESDEN

Übung USA#1 - Alternatives Lösungsverfahren

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

$$
\begin{aligned}
h_{2,i+1} &= 0.2408 h_{3,i} + 0.3211 h_{5,i} + 4.4181 \\
h_{3,i+1} &= 0.4285 h_{2,i} + 0.5714 h_{6,i} + 0.0857 \\
h_{5,i+1} &= 0.7028 h_{2,i} + 0.1054 h_{6,i} + 2.0223 \\
h_{6,i+1} &= 0.8695 h_{3,i} + 0.1304 h_{5,i}
\end{aligned}
$$

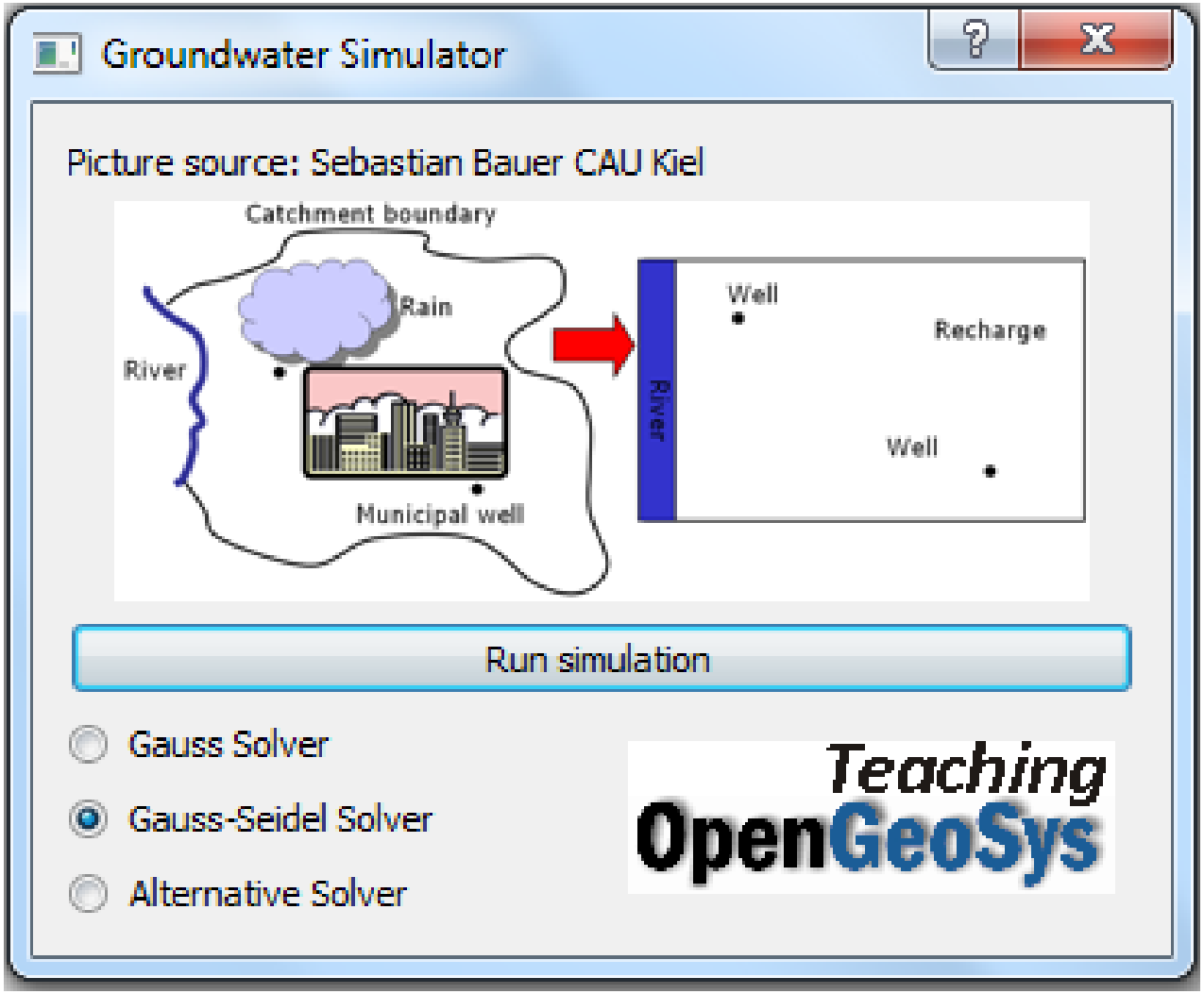
TECHNISCHE UNIVERSITÄT DRESDEN

Übung USA#1

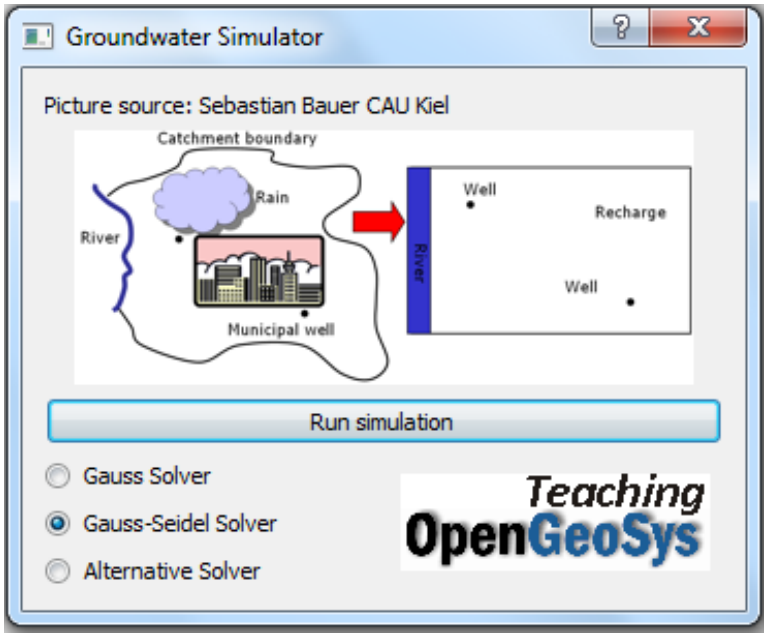
```
void Dialog::GaussSeidel()
{
  for(int k=0;k<solver_iterations;k++)
  {
    x[1] = 0.2408 * x[2] + 0.3211 * x[4] + 4.4181;
    x[2] = 0.4285 * x[1] + 0.5714 * x[5] + 0.0857;
    x[4] = 0.7028 * x[1] + 0.1054 * x[5] + 2.0223 ;
    x[5] = 0.8695 * x[2] + 0.1304 * x[4];
    TestOutput(x);
  }
}
```

TECHNISCHE UNIVERSITÄT DRESDEN

Übung USA#1 - Alternative Solver



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);
```

TECHNISCHE UNIVERSITÄT DRESDEN

QBasics: Radio-Button

```
#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(clickkedstate(bool)));
```

TECHNISCHE UNIVERSITÄT DRESDEN

Übung USA#1 - Run-Function

```
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");
}
```

TECHNISCHE UNIVERSITÄT DRESDEN

Übung USA#1 - Results

```
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
```

TECHNISCHE UNIVERSITÄT DRESDEN