

# BHYWI-08: Semester-Fahrplan 2019

## Vorlesungen

Datum	V	Thema
05.04.2019	01	IT: GitHub / Qt Installation
12.04.2019	02	Grundlagen: Kontinuumsmechanik
19.04.2019	--	Ostern
26.04.2019	03	Grundlagen: Hydromechanik
03.05.2019	04	Grundlagen: Partielle Differentialgleichungen
10.05.2019	05	Grundlagen: Numerik, Qt Übung: Funktionsrechner
17.05.2019	06	Einführung in Python
24.05.2019	07	Numerik: Finite Differenzen Methode I (explizit)
31.05.2019	08	Hausaufgabe
07.06.2019	09	Numerik: Finite Differenzen Methode II (implizit)
14.06.2019		Pfingsten
21.06.2019	10	Gerinnehydraulik: Theorie – Grundlagen
28.06.2019	11	Gerinnehydraulik: Programmierung, Übung
05.07.2019	12	Grundwassermodellierung: Datenbasierte Methoden
12.07.2019	13	Beleg

# Hydroinformatik II

## ”Prozesssimulation und Systemanalyse”

### BHYWI-08-10 @ 2019

### Gerinnehydraulik - Übungen

Olaf Kolditz

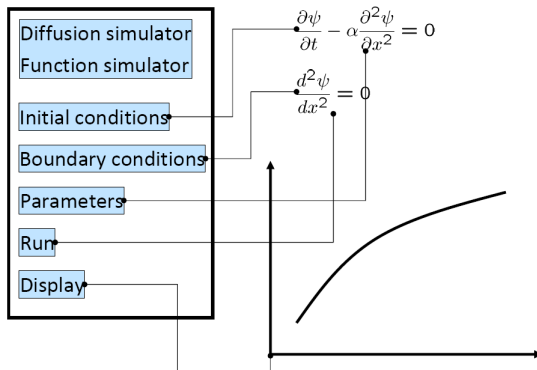
\*Helmholtz Centre for Environmental Research – UFZ

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

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

12.07.2019 - Dresden

# 0 - Zur Erinnerung: ... Konzept und Fazit



Eigenes MatLab ...

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

# 0 - Newton-Schritt Simulator

**Newton Simulator** [?] [X]

**Übung E9**  
**Gerinnehydraulik**

Here we can set text ...

**Initial conditions**

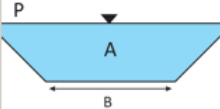
**Boundary conditions**

**Material conditions**

Run Newton step

Show results

All-in-one



Initial water level:

Water level boundary:

Channel discharge:

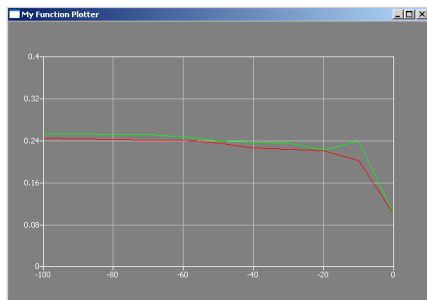
Friction coefficient 1:

Friction coefficient 2:

Bed slope:

Newton error tolerance:

Newton error:



- ▶ **Reminder:**
- ▶ Newton-Schritt Simulator
- ▶ QBasics: `QLineEdit`
- ▶ Plotter (local / global variables)

# 1 - Parameter anzeigen #1

**Newton Simulator** [?] [X]

**Übung E9**  
**Gerinnehydraulik**

Here we can set text ...

Initial conditions

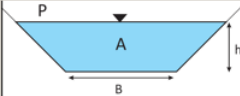
Boundary conditions

Material conditions

Run Newton step

Show results

All-in-one



Initial water level: 0.20000

Water level boundary: 0.15000

Channel discharge: 0.05000

Friction coefficient 1: 0.50000

Friction coefficient 2: 1.00000

Bed slope: 0.00040

Newton error tolerance: 0.00100

Newton error:

Newton iterations:

- ▶ Tafelbild:
- ▶ Datentransfer
- ▶ Beispiel:  
Anfangsbedingungen

# 1 - Parameter anzeigen #2

```
void Dialog::on_pushButtonIC_clicked()
{
    // Anfangsbedingungen setzen
    for(int i=0;i<n;i++)
    {
        u_old[i] = ICValue;
    }
    // Daten im Dialog sichtbar machen
    sDummy.setNum(ICValue,'f',5);
    lineEditIC->setText(sDummy);
    // Schnick-Schnack
    pushButtonIC->setStyleSheet("background-color: green");
}
```

↔ Important: ICValue

# 1 - Parameter anzeigen #3

```
void Dialog::on_pushButtonBC_clicked()
{...
    sDummy.setNum(BCValue, 'f', 5);
    lineEditBCR->setText(sDummy);
}

void Dialog::on_pushButtonMAT_clicked()
{
    sDummy.setNum(friction_law_exponent, 'f', 5);
    lineEditFrictionLawExponent->setText(sDummy);
    sDummy.setNum(m, 'f', 5);
    lineEditFrictionCoefficient->setText(sDummy);
    sDummy.setNum.bed_slope, 'f', 5);
    lineEditBedSlope->setText(sDummy);
    sDummy.setNum(discharge, 'f', 5);
    lineEditDischarge->setText(sDummy);
    sDummy.setNum(error_tolerance, 'f', 5);
    lineEditNewtonTolerance->setText(sDummy);
}
```

# 2 - Iterationszähler #1

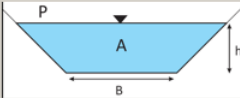
**Newton Simulator** [?] [X]

**Übung E9**  
**Gerinnehydraulik**

Here we can set text ...

Initial conditions  
Boundary conditions  
Material conditions  
Run Newton step  
Show results  
All-in-one

Initial water level: 0.20000  
Water level boundary: 0.15000  
Channel discharge: 0.05000  
Friction coefficient 1: 0.50000  
Friction coefficient 2: 1.00000  
Bed slope: 0.00040  
Newton error tolerance: 0.00100  
Newton error: -0.18206  
Newton iterations: 1



- ▶ Wo einbauen?
- ▶ `while()` Schleife



- ▶ Implementierung: Welche Funktion(en)?

```
int main(int argc, char *argv[])
```

```
public:
```

```
    Dialog(QWidget *parent = 0);  
    ~Dialog();
```

```
private slots:
```

```
    void on_pushButtonIC_clicked();  
    void on_pushButtonBC_clicked();  
    void on_pushButtonMAT_clicked();  
    void on_pushButtonRUN_clicked();  
    void on_pushButtonALL_clicked();  
    double RUN_NewtonStep();
```

- ▶ Implementierung: while()

```
void Dialog::on_pushButtonALL_clicked()
{...
  float error = 1.1*error_tolerance;
  while(error>error_tolerance)
  {
    error = RUN_NewtonStep();
  }
}
```

### ► Implementierung: Komplet

```
void Dialog::on_pushButtonALL_clicked()
{...
    float error = 1.1*error_tolerance;
    while(error>error_tolerance)
    {
        error = RUN_NewtonStep();
        //Plotter
        QVector<QPointF> points0;
        for(int i=0;i<n-1;i++)
        {
            points0.append(QPointF(x[i],u_new[i]));
        }
        points0.append(QPointF(x[10],u_new[10]));
        plotter->setCurveData(k++, points0);
        plotter->show();
        //Datenübergabe Dialog
        sDummy.setNum(error, 'f', 5);
        lineEditNewtonError->setText(sDummy);
        sDummy.setNum(k,5);
        lineEditIterations->setText(sDummy);
    }
}
```

# 3 - Fehlerberechnung #1

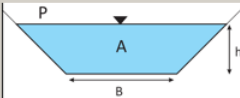
**Newton Simulator** [?] [X]

**Übung E9**  
**Gerinnehydraulik**

Here we can set text ...

Initial conditions  
Boundary conditions  
Material conditions  
Run Newton step  
Show results  
All-in-one

Initial water level: 0.20000  
Water level boundary: 0.15000  
Channel discharge: 0.05000  
Friction coefficient 1: 0.50000  
Friction coefficient 2: 1.00000  
Bed slope: 0.00040  
Newton error tolerance: 0.00100  
Newton error: -0.18206  
Newton iterations: 1



- ▶ Warum nur eine Iteration?

## 3 - Fehlerberechnung #2

```
while(error>error_tolerance)
{...}
```

```
while(error > error_tolerance)  
{...}
```

$$\varepsilon = \sqrt{\sum_i (h^{k+1} - h^k)^2}$$

```
error = sqrt(error*error);
```

## 3 - Fehlerberechnung #2

```
while(error>error_tolerance)  
{...}
```

$$\varepsilon = \sqrt{\sum_i (h^{k+1} - h^k)^2}$$

```
error = sqrt(error*error);
```

$$\varepsilon = |h^{k+1} - h^k|$$

```
error = abs(error);
```

# 3 - Fehlerberechnung #3

**Newton Simulator** [?] [X]

**Übung E9**  
**Gerinnehydraulik**

Here we can set text ...

Initial conditions

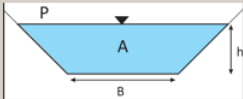
Boundary conditions

Material conditions

Run Newton step

Show results

All-in-one



Initial water level: 0.20000

Water level boundary: 0.15000

Channel discharge: 0.05000

Friction coefficient 1: 0.50000

Friction coefficient 2: 1.00000

Bed slope: 0.00040

Newton error tolerance: 0.00100

Newton error: 0.00080

Newton iterations: 20

► Jetzt klappts.



# 4 - Parameter ändern #1

- ▶ Erstmal überlegen.
- ▶ ↪ Tafelbild

## 4 - Parameter ändern #2

**Newton Simulator** [?] [X]

**Übung E9**  
**Gerinnehydraulik**

Initial water level:

Water level bounda:

Here we can set te: Channel discharge:

**Initial conditions** Friction coefficient 1:

**oundary condition** Friction coefficient 2:

**Material conditions** Bed slope:

Newton error tolera:

Newton error:

Newton iterations:

- Layout: Blöd gelaufen.

## 4 - Parameter ändern #2

**Newton Simulator** [?] [X]

**Übung E9**  
**Gerinnehydraulik**

Here we can set text ...

**Initial conditions**

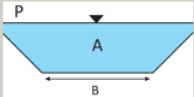
**Boundary conditions**

**Material conditions**

Run Newton step

Show results

All-in-one



Initial water level:

Water level boundary:

Channel discharge:

Friction coefficient 1:

Friction coefficient 2:

Bed slope:

Newton error tolerance:

Newton error:

Newton iterations:

Change IC value

- ▶ Wo ändern?
- ▶ `w.setFixedWidth(400)`
- ▶ Na also.

```
#include <QLineEdit>

//declaration
QLineEdit* lineEditIC;
//definition
lineEditIC = new QLineEdit();
//use
Get- und Set-Funktionen
```

- ▶ Warum auf ein mal Unterscheidung zwischen Deklaration und Definition?

```
QString ICString = lineEditIC->text()  
double ICValue = lineEditIC->text().toDouble()
```

```
#include <QLineEdit>

//declaration
QLineEdit* lineEditIC;
//definition
lineEditIC = new QLineEdit();
//use
    //set
QString ICString;
ICString.setNum(ICValue, 'f', 5);
lineEditIC->setText(sIC);
    //get
QString ICString = lineEditIC->text()
double ICValue = lineEditIC->text().toDouble()
```

K

# 5 - Anfangsbedingungen ändern #1 Wie

**Newton Simulator** [?] [X]

**Übung E9**  
**Gerinnehydraulik**

Here we can set text ...

**Initial conditions**

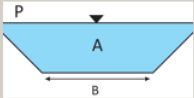
**Boundary conditions**

**Material conditions**

**Run Newton step**

**Show results**

**All-in-one**



Initial water level:

Water level boundary:

Channel discharge:

Friction coefficient 1:

Friction coefficient 2:

Bed slope:

Newton error tolerance:

Newton error:

Newton iterations:

**Change IC value**

► Change IC Value

## 5 - Anfangsbedingungen ändern #2 Programmierung

Deklaration:

```
class Dialog : public QDialog
{...
private slots:
    void on_pushButtonICChange_clicked();
...}
```

Definition:

```
void Dialog::on_pushButtonICChange_clicked()
{
    QString sICValue = lineEditIC->text();
    ICValue = lineEditIC->text().toDouble();
    for(int i=0;i<n-1;i++)
    {
        u_old[i] = ICValue;
    }
}
```



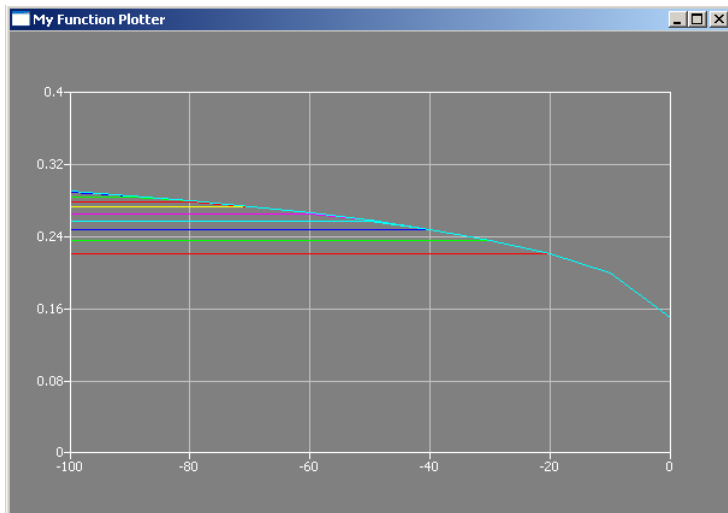
Verknüpfung einer Funktion mit einem Ereignis:

```
connect(pushButtonICChange,  
        SIGNAL(clicked()),  
        this,  
        SLOT(on_pushButtonICChange_clicked()));
```

Typische K-Fragen:

- ▶ Wo werden Member-Funktionen einer Klasse deklariert?
- ▶ Wo werden Member-Funktionen einer Klasse definiert?
- ▶ Schreiben sie die Verknüpfung einer Funktion mit einem Ereignis, wenn eine Schaltfläche gedrückt wurde?

## 5 - Anfangsbedingungen ändern #3 Analyse



▶ ↷ Run simulation.

# 6 - Randbedingungen #1

**Newton Simulator** [?] [X]

**Übung E9**  
**Gerinnehydraulik**

Here we can set text ...

**Initial conditions**

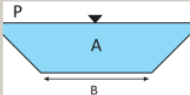
**Boundary conditions**

**Material conditions**

Run Newton step

Show results

All-in-one



Initial water level: 0.20000

Water level boundary: 0.30000

Channel discharge: 0.05000 Change IC value

Friction coefficient 1: 0.50000

Friction coefficient 2: 1.00000

Bed slope: 0.00040 Change BC value

Newton error tolerance: 0.00100

Newton error: 0.00099

Newton iterations: 114

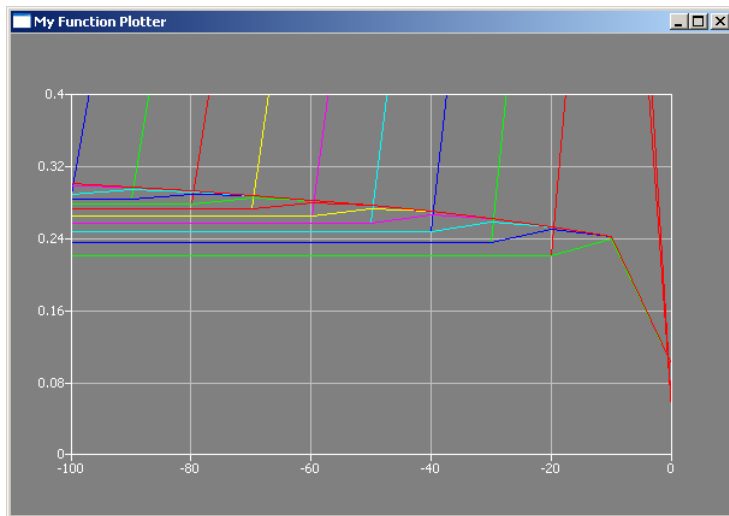
- ▶ the same procedure as ...
- ▶ Wenn Zeit: Implementierung zeigen (step-by-step).

- ▶ Advanced OOP

```
void Dialog::on_pushButtonBCChange_clicked()
{
    QString sBCValue = lineEditBCR->text();
    BCValue = lineEditBCR->text().toDouble();
    on_pushButtonBC_clicked();
}
```

- 1 Funktion deklarieren
- 2 Funktion definieren (leer) [Komplizierbarkeit]
- 3 Funktion schreiben
- 4 Dialogelement anlegen [z.B. Schaltfläche]
  - ▶ Element deklarieren
  - ▶ Element definieren
  - ▶ Element ins Layout einfügen
- 5 Verknüpfung zwischen Funktion und Dialogelement herstellen
- 6 Funktionalität testen
  - ▶ Bei Fehlern: Debuggen
  - ▶ 2. (verbesserte) Implementierung
  - ▶ Namenskonventionen ...

# 6 - Randbedingungen #2



## 7 - Discharge #1

- ▶ Nochmal lokale und globale Variablen ...
- ▶ Nochmal, die wichtige Rolle des Konstruktors ...

```
void Dialog::on_pushButtonMAT_clicked()
{...
    //ab in den Konstruktor! discharge = 0.05; // Volumenflie
...}
```

```
Dialog::Dialog(QWidget *parent) : QDialog(parent)
{...
    //hier bin ich glücklich ...
    discharge = 0.05; // Volumenfließrate [m3/s]
...}
```

# 7 - Discharge #2

**Newton Simulator** [?] [X]

**Teaching OpenGeoSys**

Here we can set text ...

**Initial conditions**

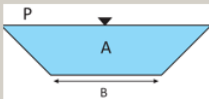
**Boundary conditions**

**Material conditions**

Run Newton step

Show results

All-in-one



Initial water level: 0.20000

Water level boundary: 0.15000

Channel discharge: 0.10000

Friction coefficient 1: 0.50000

Friction coefficient 2: 1.00000

Bed slope: 0.00040

Newton error tolerance: 0.00100

Newton error: 0.00020

Newton iterations: 100

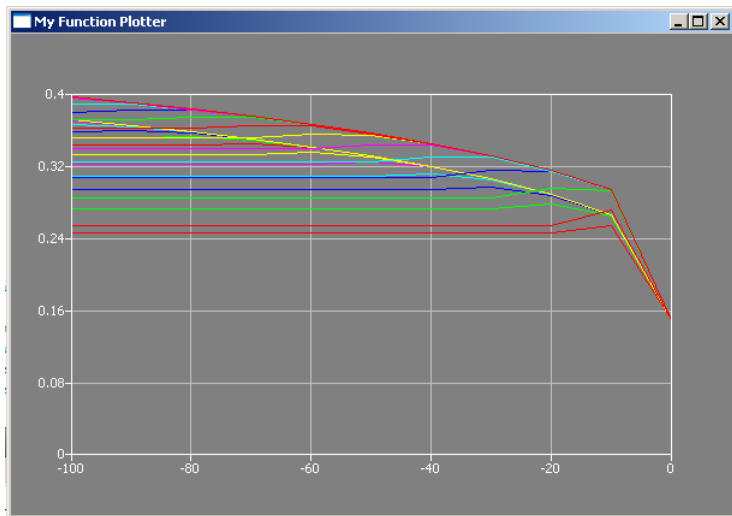
Change IC value

Change BC value

Change discharge value



# 7 - Discharge #3



# 8 - Fast fertig #1

Newton Simulator

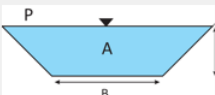
**Teaching OpenGeoSys**

Here we can set text ...

**BHYWI-08: HYDROINFORMATIK II**  
**E8: Newton Simulator**

Initial conditions  
Boundary conditions  
Material conditions

Run Newton step  
Show results  
All-in-one



Initial water level: 0.20000

Water level boundary: 0.15000

Channel discharge: 0.10000

Friction coefficient 1: 0.50000

Friction coefficient 2: 1.00000

Bed slope: 0.00040

Newton error tolerance: 0.01000

Newton error:

Newton iterations:

# Software-Engineering

## GitHub

GitHub, Inc. (US) | <https://github.com/envinf/teaching> Suchen

Search or jump to... Pull requests Issues Marketplace Explore

envinf / teaching Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

This repository is intended for teaching purposes at our partner universities Edit

Add topics

3 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

OlafKolditz	Update dummy.cpp	Latest commit b6c9194 14 days ago
hydrosystems	Update dummy.cpp	14 days ago
.gitignore	Initial commit	14 days ago
README.md	Initial commit	14 days ago

README.md

Search or jump to...
Pull requests Issues Marketplace Explore

envinf / **teaching** Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

This repository is intended for teaching purposes at our partner universities Edit

[Add topics](#)

6 commits
1 branch
0 releases
1 contributor

Branch: master New pull request
Create new file Upload files Find file **Clone or download**

	Olaf Kolditz Add files via upload	Latest commit c107675 22 minutes ago
	hydroinformatics2/channelflow	Add files via upload 22 minutes ago
	hydrosystems	Update dummy.cpp 21 days ago
	.gitignore	Initial commit 21 days ago
	README.md	Initial commit 21 days ago

README.md

The screenshot shows the GitHub interface for the repository 'envinf / teaching'. At the top, there is a search bar and navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The repository name 'envinf / teaching' is displayed, along with 'Unwatch' (1), 'Star' (0), and 'Fork' (0) buttons. Below this, there are tabs for 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Insights', and 'Settings'. The current branch is 'master', and the file path is 'teaching / hydroinformatics2 / channelflow /'. There are buttons for 'Create new file', 'Upload files', 'Find file', and 'History'. A commit by 'OlafKolditz' is shown with the message 'Add files via upload' and a timestamp of '23 minutes ago'. Below the commit, a list of files is displayed with their respective commit messages and timestamps.

File	Commit Message	Time
..		
BHYWI-08-08-E.pro	Add files via upload	23 minutes ago
dialog.cpp	Add files via upload	23 minutes ago
dialog.h	Add files via upload	23 minutes ago
main.cpp	Update main.cpp	24 minutes ago
plotter.cpp	Add files via upload	23 minutes ago
plotter.h	Add files via upload	23 minutes ago

# BHYWI-08: Semester-Fahrplan

## Übungen

Datum	E	Übungen
05.04.2019	00	Git und QT (Lars Bilke)
03.05.2019	01	Qt: Hallo World
10.05.2019	02	Qt: Funktionsrechner
17.05.2019		
31.05.2019	03	Qt: Explizite Finite-Differenzen-Methode
07.06.2019	04	Qt: Implizite Finite-Differenzen-Methode
28.06.2019	05	Qt: Gerinnehydraulik I (QAD)
	06	Qt: Gerinnehydraulik II (OOP)
	08	Qt: Gerinnehydraulik IV (interaktiv)
		...

<https://github.com/envinf/Hydroinformatik-II>