

# Hydroinformatik - SoSe 2026

## UW-BHW-414-16: Gerinnehydraulik

Prof. Dr.-Ing. habil. Olaf Kolditz

<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

Dresden, 03.07.2026

# Zeitplan: Hydroinformatik I+II

Sommersemester 2026: Stand: 06.04.2026

Nr.	KW	Datum	ID	Thema
01+02	16	17.04.2026	UW-BHW-414-01/02	Einführung in die Vorlesung, Umweltinformatik
03	16	17.04.2026	UW-BHW-414-03	Werkzeuge, Hello World (in C++)
05	17	24.04.2026	UW-BHW-414-04	Selbststudium: Software-Installationen
07	19	08.05.2026	UW-BHW-414-05	Objekt-Orientierte Programmierung: C++, Klassen
09	20	15.05.2026	UW-BHW-414-06	Programmiersprache Python
11	21	22.05.2026	UW-BHW-414-07/08	Modellierung, Digitalisierung - Wasser 4.0
00	22	29.05.2026		Vorlesungsfreie Woche
13	23	05.06.2026	UW-BHW-414-09/10	KI, Maschinelles Lernen, Neuronale Netzwerke
15	24	12.06.2026	UW-BHW-414-11/12	Kontinuumsmechanik, Hydromechanik
17	25	19.06.2026	UW-BHW-414-13/14	Differentialgleichungen, Näherungsverfahren
19	26	26.06.2026	UW-BHW-414-15	Finite-Differenzen, explizite Verfahren
21	27	03.07.2026	UW-BHW-414-16	Finite-Differenzen, implizite Verfahren
23	28	10.07.2026	UW-BHW-414-17	Gerinnehydraulik, Grundwasserhydraulik
25	29	17.07.2026	UW-BHW-414-M	Grundwasserhydraulik
27	30	24.07.2026	UW-BHW-414-N	Zusammenfassung, Klausurvorbereitung

- 1 UW-BHW-414-16: Gerinnehydraulik
  - Semesterplan

- 1 Abfallwirtschaft / Wärmeversorgung: Diffusionsprozesse
- 2 **Hydrology: Gerinnehydraulik (→ this)**
- 3 Grundwasserwirtschaft: Grundwasserhydraulik (→ next)

0 Finite-Differenzen Verfahren (explizit und implizit)

---

1 Gerinnehydraulik: Parameter, Symbole ...

2 Energieerhaltung: Bernoulli-Gleichung

3 Impulserhaltung: Saint-Venant-Gleichungen

4 Lösung der Bernoulli-Gleichung

5 Newton-Verfahren

6  $\mapsto$  numerisches Lösungsverfahren (nächste Veranstaltung)

---

7 Abschluss Übungen

8 Klausurvorbereitung

Für die Beschreibung von Strömungsprozessen in Flüssen (d.h. Gerinne) gibt es zwei verschiedene Möglichkeiten basierend auf den sogenannten Bernoulli (Energiebetrachtung) oder Saint-Venant Gleichungen. Wir verwenden die erste für die Implementierung in C++ (Bildquellen: Technische Hydromechanik).



- ▶ History
- ▶ gelebt 1700-1782
- ▶ Freund von L Euler
- ▶ Bernoulli Gleichung
- ▶ kinetische Theorie der Gase
- ▶ russ. Akademie Petersburg
- ▶ Vater/Sohn Beziehung

# Bernoulli-Gleichung #1

$$z_{i+1} + \frac{p_{i+1}}{\rho g} + \frac{v_{i+1}^2}{2g} = z_i + \frac{p_i}{\rho g} + \frac{v_i^2}{2g} + h_f \quad (1)$$

- ▶ Symbole
- ▶ Gesetz von Darcy-Weisbach für streckenabhängige Verluste

$$h_f = \lambda \frac{x_{i+1} - x_i}{d_{hy}} \frac{v^2}{2g} \quad (2)$$

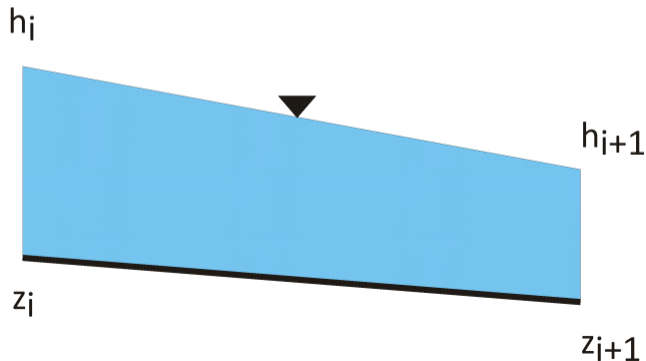
- ▶ Symbole (Zusammenfassung später)
- ▶ Gesetz von Darcy-Weisbach für streckenabhängige Verluste

$$d_{hy} = 4r_{hy} = 4 \frac{A}{P}$$

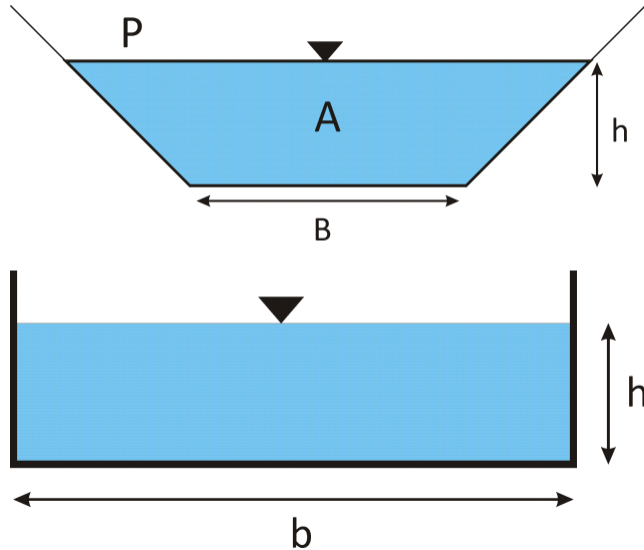
# Bernoulli-Gleichung #2

$$p = \rho gh \quad (4)$$

$$z_{i+1} + h_{i+1} + \frac{v_{i+1}^2}{2g} = z_i + h_i + \frac{v_i^2}{2g} + h_f \quad , E = h + \frac{v^2}{2g} \quad (5)$$



# Channel Geometry



► Durchfluss

$$Q = A v = b h v \quad (6)$$

$$z_{i+1} + h_{i+1} + \frac{Q_{i+1}^2}{2g b^2 h_{i+1}^2} = z_i + h_i + \frac{Q_i^2}{2g b^2 h_i^2} + h_f \quad (7)$$

# Fluid Momentum Balance - Navier-Stokes equation

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = \mathbf{f}^e + \frac{1}{\rho} \nabla \cdot \sigma \quad (8)$$

In index notation the above vector equation is written as

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} &= \frac{1}{\rho} \left( \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} \right) \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} &= \frac{1}{\rho} \left( \frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} \right) \\ \frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} &= g + \frac{1}{\rho} \left( \frac{\partial \sigma_{zx}}{\partial x} + \frac{\partial \sigma_{zy}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} \right) \end{aligned} \quad (9)$$

with  $u = v_x, v = v_y, w = v_z$  and  $\mathbf{f}^e = \mathbf{g}$ .

- ▶ History
- ▶ gelebt 1797-1886
- ▶ 1D open channel flow
- ▶ shallow water equations (Flachwassergleichungen)
- ▶ Stokes equation (is his)
- ▶ Engineer
- ▶ Count

# Saint-Venant-Gleichungen #1

- ▶ Saint-Venant-Gleichungen
- ▶ Vertikal gemittelte Navier-Stokes-Gleichung
- ▶ 2D Flachwasser-Gleichungen (Filmchen)
- ▶ Mittelung über die Querschnittsfläche
- ▶ 1D Gerinnehydraulik
- ▶ Fließgeschwindigkeit  $v = Q/A$  in Flußrichtung

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0 \quad (10)$$

$$\frac{\partial Q}{\partial t} \frac{1}{A} + \frac{\partial}{\partial x} \left( \frac{Q^2}{2A^2} + gh(x, A) \right) = g(S_o - S_f) \quad (11)$$

- ▶ Stationäre Bedingungen

$$\begin{aligned}\frac{d}{dx} (A(x, h)v) &= 0 \\ \frac{d}{dx} \left( \frac{v^2}{2} + gh(x, A) \right) &= g(S_o - S_f)\end{aligned}\quad (12)$$

unter Verwendung der Primärvariablen  $A$  und  $Q$

$$\frac{d}{dx} \left( \frac{Q(h)^2}{2A(h)^2} + gh(x, A) \right) = g(S_o - S_f)\quad (13)$$

- ▶ Nichtlineare Gleichung

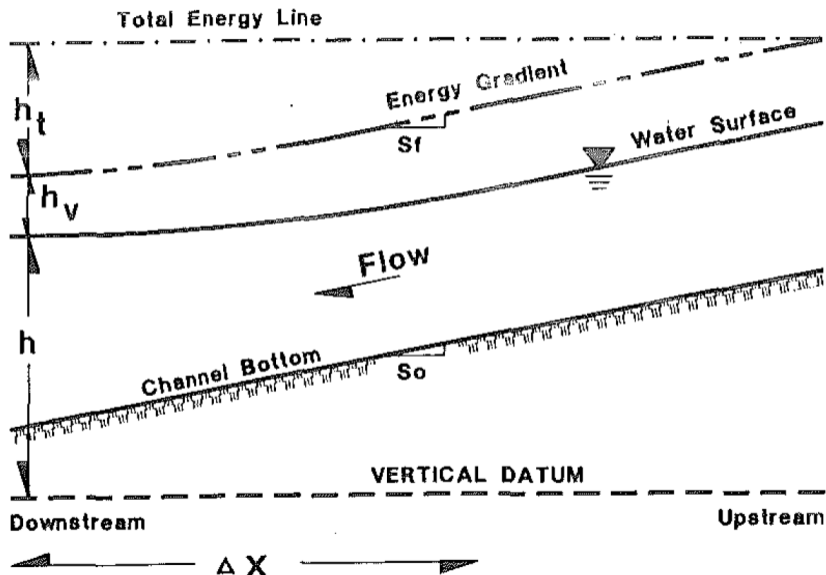
Symbol	Parameter	Englisch	Einheit
$h$	Wasserspiegel	water surface elevation	m
$i$	Sohlenspiegel	channel datum	m
$z$	Wassertiefe	water depth	m
$Z_L, Z_R$	linke / rechte Böschung	left / right channel slope	m
$Q$	Durchfluss	discharge	$\text{m}^3\text{s}^{-1}$
$A$	Querschnittsfläche	channel cross-section	$\text{m}^2$
$B$	Breite	channel width	$\text{m}^2$
$P$	benetzter Umfang	wetted perimeter	m
$R, r_{hy}$	hydraulische Radius	hydraulic radius	m
$U$	stromaufwärts	upstream	-
$D$	stromabwärts	downstream	-
$\Delta x$	Abstand zwischen U und D	distance between U and D	m
$h_v$	kinetische Energie	velocity head	m
$h_f$	Reibungsverluste	friction head	m
$S_o$	Sohlgefälle	bed slope	-
$S_f$	Streckenverlust	friction slope	-

$$z_{i+1} + \frac{p_{i+1}}{\rho g} + \frac{v_{i+1}^2}{2g} = z_i + \frac{p_i}{\rho g} + \frac{v_i^2}{2g} + h_f \quad (14)$$

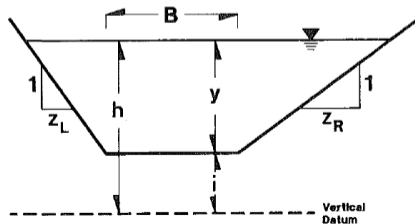
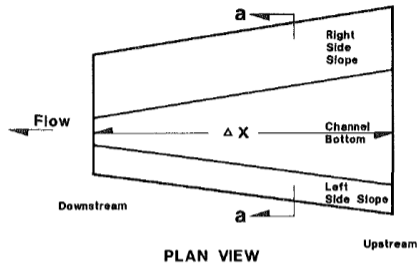
$$h_f = \lambda \frac{x_{i+1} - x_i}{d_{hy}} \frac{v^2}{2g} \quad (15)$$

$$d_{hy} = 4r_{hy} = 4 \frac{A}{p} \quad (16)$$

# Energiebetrachtung #2: Bernoulli/Saint-Venant



# Energiebetrachtung #3: Bernoulli



## Energiebetrachtung #4 Geometrie

- ▶ Die trapezoide Querschnittsfläche des Gerinnes kann folgendermaßen berechnet werden (Abb. 3).

$$A = y \left[ B + \frac{Z_L + Z_R}{2} \right] \quad (17)$$

- ▶ Der benetzte Umfang des Gerinnes ist.

$$P = B + y \left[ \sqrt{1 + Z_L^2} + \sqrt{1 + Z_R^2} \right] \quad (18)$$

- ▶ Der hydraulische Radius ergibt sich dann als:

$$R = \frac{A}{P} \quad (19)$$

- ▶ Letztlich benötigen wir noch den Zusammenhang von Fließgeschwindigkeit und Durchfluss.

$$v = \frac{Q}{A} = \frac{Q}{y \left[ B + \frac{Z_L + Z_R}{2} \right]}$$

Die Streckenverluste lassen sich nach Manning wie folgt beschreiben

$$S_f = \left( \frac{Q}{AR^{2/3}} \right)^2 \quad (21)$$

$$S_f = Q^2 A^{-2} R^{-4/3} \quad (22)$$

mit

$$R = \frac{y(B + C_4)}{B + yC_5} \quad (23)$$

$$C_4 = \frac{Z_L + Z_R}{2} \quad (24)$$

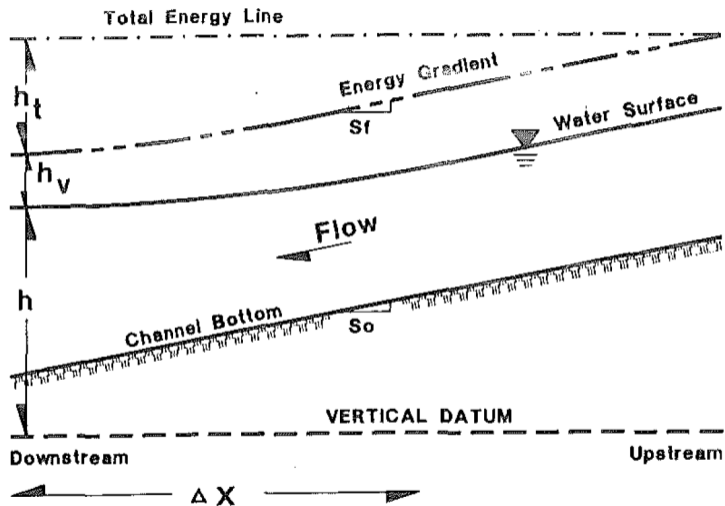
$$C_5 = \sqrt{1 + Z_L^2} + \sqrt{1 + Z_R^2} \quad (25)$$

$$S_f = Q^2 (y(B + C_4))^{-2} \left( \frac{y(B + C_4)}{B + yC_5} \right)^{-4/3} \quad (26)$$

$$S_f = Q^2 (By + C_4y)^{-2} (By + C_4y)^{-4/3} (B + C_5y)^{4/3} \quad (27)$$

$$S_f = Q^2 (By + C_4y)^{-10/3} (B + C_5y)^{4/3}$$

# Lösung einer nichtlinearen Gleichung #1



## Lösung einer nichtlinearen Gleichung #2

$$f(h) = \left( h + \frac{v^2}{2g} \right) |_D - \left( h + \frac{v^2}{2g} \right) |_U + \Delta x \frac{(S_{f,U} + S_{f,D})}{2} \quad (30)$$

$$f(h) = \left( h + \frac{Q^2}{2gA^2} \right) |_D - \left( h + \frac{Q^2}{2gA^2} \right) |_U + \Delta x \frac{(S_{f,U} + S_{f,D})}{2} \quad (31)$$

- ▶ Konstruktion eines Newton-Verfahrens

$$h_{k+1} = h_k + \frac{f(h_k)}{f'(h_k)} \quad (32)$$

# Math: Newton-Verfahren #1

- ▶ Nichtlineare Gleichung

$$\mathbf{A}(\mathbf{x}) \mathbf{x} = \mathbf{b}(\mathbf{x}) \quad (f(h) = 0) \quad (33)$$

- ▶ Residuum

$$\mathbf{R}(\mathbf{x}) = \mathbf{A}(\mathbf{x}) \mathbf{x} - \mathbf{b}(\mathbf{x}) = 0 \quad (34)$$

- ▶ Taylor-Reihe

$$\mathbf{R}_{k+1} = \mathbf{R}_k + \left[ \frac{\partial \mathbf{R}}{\partial \mathbf{x}} \right]_k \Delta \mathbf{x}_{k+1} + 0(\Delta \mathbf{x}_{k+1}^2) \quad (35)$$

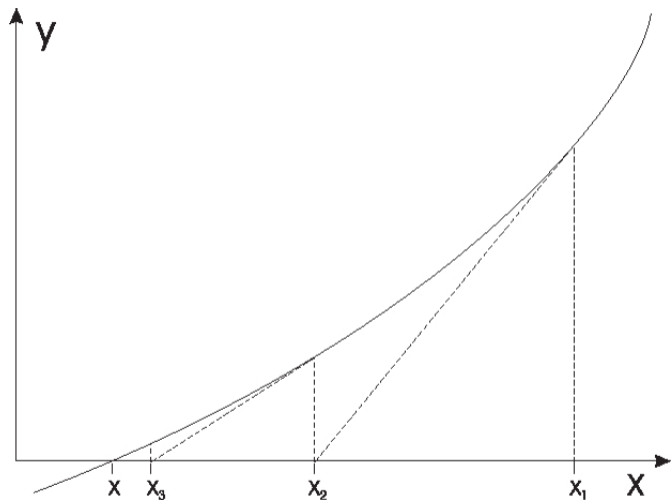
- ▶ Umstellen

$$\Delta \mathbf{x}_{k+1} \approx -\mathbf{J}_k^{-1} \mathbf{R}_k \quad (\mathbf{R}_{k+1} = 0) \quad (36)$$

- ▶ Lösungsvorschrift

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{J}_k^{-1} \mathbf{R}_k$$

# Math: Newton-Verfahren #2



$$y = R(x)$$

- ▶ Newton-Verfahren:  $\mathbf{x} = h$

$$h_{k+1} = h_k + \frac{f(h_k)}{f'(h_k)} = \frac{\mathbf{R}_k}{\mathbf{J}_k} \quad (38)$$

- ▶  $f'(h) = f'(y)$

$$\frac{d}{dh} \left( h + \frac{v^2}{2g} \right) = \frac{d}{dh} \left( h + \frac{Q^2}{2gA^2} \right) = 1 - \frac{Q^2}{2gA^3} \frac{dA}{dh} \quad (39)$$

$$\frac{dA}{dh} = \frac{d}{dh} (y(B + C_4)) = B + C_4 \quad (40)$$

Bleibt noch die Differenzierung der Streckenverluste

$$\frac{dS_f}{dh} = S'_f = \frac{d}{dh} \left( \frac{Q}{AR^{2/3}} \right)^2 \quad (41)$$

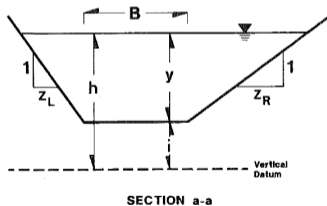
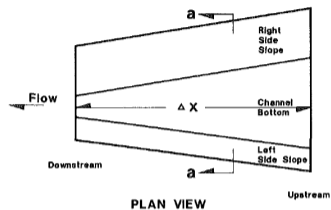
$$\begin{aligned} S'_f &= \left[ Q^2 (By + C_4y)^{10/3} \frac{4}{3} (B + C_5y)^{1/3} C_5 \right] \\ &+ \left[ (B + yC_5)^{4/3} \frac{-10Q^2}{3} (By + C_4y)^{13/3} (B + C_4) \right] \end{aligned} \quad (42)$$

$$\begin{aligned} S'_f &= \frac{4}{3} Q^2 C_5 (By + C_4y)^{-10/3} (B + C_5y)^{1/3} \\ &- \frac{10}{3} Q^2 (B + C_4) (B + C_5y)^{4/3} (By + C_4y)^{-13/3} \end{aligned} \quad (43)$$

# Übung

## Gerinnehydraulik - Lösungsverfahren

# Energiebetrachtung #3: Bernoulli



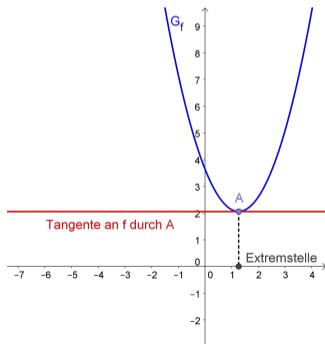
$$y \equiv z$$

Fig.: Trapezgerinne Paine (1992)

# Newton-Verfahren #2

Funktional: Energieerhaltung (potentielle + kinetische + Wärme-Energie (Reibung))

$$f(h) = \left( h + \frac{Q^2}{2gA^2} \right) |_{i+1} - \left( h + \frac{Q^2}{2gA^2} \right) |_i + (x_{i+1} - x_i) \frac{S_{f,i+1} + S_{f,i}}{2} \quad (44)$$



$$\begin{aligned} f'(h) &= \frac{df}{dh} = 0 \\ &\approx \frac{f(h_{k+1}) - f(h_k)}{h_{k+1} - h_k} \\ h_{k+1} &= h_k - \frac{f(h_k)}{f'(h)} = h_k - \frac{N}{D} \end{aligned} \quad (45)$$

Quelle: [https://menzelths.github.io/Mathematik-Kursstufe/Ableitungen\\_Extremstellen.html](https://menzelths.github.io/Mathematik-Kursstufe/Ableitungen_Extremstellen.html)

$$f(h) = \underbrace{\left(h + \frac{Q^2}{2gA^2}\right) \Big|_{i+1}}_{N_1} - \underbrace{\left(h + \frac{Q^2}{2gA^2}\right) \Big|_i}_{N_2} + \underbrace{(x_{i+1} - x_i) \frac{S_{f,i+1} + S_{f,i}}{2}}_{N_3} \quad (46)$$

$$f'(h) = \underbrace{\frac{d}{dh} \left(h + \frac{Q^2}{2gA^2}\right) \Big|_{i+1}}_{D_1} - \underbrace{\frac{d}{dh} \left(h + \frac{Q^2}{2gA^2}\right) \Big|_i}_{D_2} + \underbrace{\frac{x_{i+1} - x_i}{2} \frac{d}{dh} (S_{f,i+1} + S_{f,i})}_{D_3} \quad (47)$$

$$D_1 = \frac{d}{dh} \left( h + \frac{Q^2}{2gA^2} \right) = 1 - \frac{Q^2}{gA^3} \frac{dA}{dh} \quad (48)$$

$$\frac{dA}{dh} = \frac{d}{dh} (y(B + yC_4)) = B + 2yC_4 \quad (49)$$

$$D_1 = 1 - \frac{Q^2}{gA^3} B + 2yC_4 \quad (50)$$

Bleibt noch die Differenzierung der Streckenverluste

$$D_3 = \frac{dS_f}{dh} = S'_f = \frac{d}{dh} \left( \frac{Q}{AR^{2/3}} \right)^2 \quad (51)$$

$$\frac{d}{dh} \left( \frac{Q}{AR^{2/3}} \right)^2 = Q^2 \frac{d}{dh} \left( A^{-2} R^{-4/3} \right) \quad (52)$$

$$\dots = Q^2 \left( \frac{dA^{-2}}{dh} R^{-4/3} \right) + \left( A^{-2} \frac{dR^{-4/3}}{dh} \right) \quad (53)$$

$$\dots = Q^2 \left( -2A^{-3} \frac{dA}{dh} R^{-4/3} \right) + \left( A^{-2} \frac{(-4)}{3} R^{-7/3} \frac{dR}{dh} \right) \quad (54)$$

Bleibt noch die Differenzierung der Streckenverluste

$$\frac{dS_f}{dh} = S'_f = \frac{d}{dh} \left( \frac{Q}{AR^{2/3}} \right)^2 \quad (55)$$

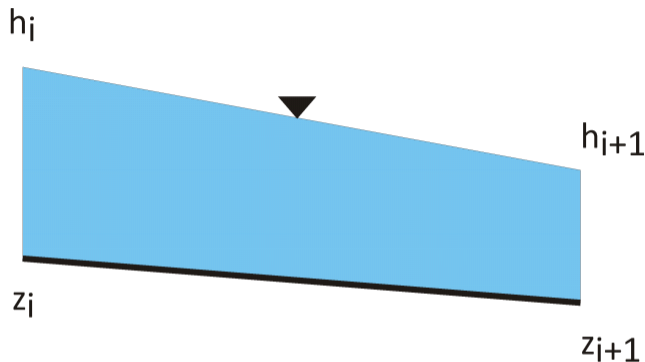
$$\begin{aligned} S'_f &= \left[ Q^2 (By + C_4 y^2)^{10/3} \frac{4}{3} (B + C_5 y)^{1/3} C_5 \right] \\ &+ \left[ (B + yC_5)^{4/3} \frac{-10Q^2}{3} (By + C_4 y^2)^{13/3} (B + 2C_4 y) \right] \end{aligned} \quad (56)$$

$$\begin{aligned} S'_f &= \frac{4}{3} Q^2 C_5 (By + C_4 y^2)^{-10/3} (B + C_5 y)^{1/3} \\ &- \frac{10}{3} Q^2 (B + 2C_4 y) (B + C_5 y)^{4/3} (By + C_4 y^2)^{-13/3} \end{aligned}$$

# Übungen

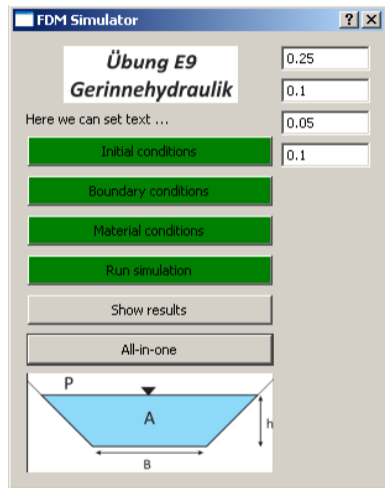
## Gerinnehydraulik (alte Versionen)

- ▶ **Q&D:** Übung 1  
(BHYWI-08-05-E):  
Funktionalität
- ▶ **OOP:** Übung 2  
(BHYWI-08-06-E):  
Modularität
- ▶ **GUI:** Übung 3  
(BHYWI-08-07-E): Interaktion  
(Ausgabe)
- ▶ **GUI:** Übung 4  
(BHYWI-08-08-E): Interaktion  
(Eingabe)



BHYWI-08-05-E > EX29-Gerinnehydraulik  
BHYWI-08-06-E > BHYWI-08-11B  
BHYWI-08-07-E > BHYWI-08-11C  
BHYWI-08-08-E > BHYWI-08-11D

- ▶ **Q&D:** Übung 1  
(BHYWI-08-05-E):  
Funktionalität
- ▶ **OOP:** Übung 2  
(BHYWI-08-06-E):  
Modularität
- ▶ **GUI:** Übung 3  
(BHYWI-08-07-E): Interaktion  
(Ausgabe)
- ▶ **GUI:** Übung 4  
(BHYWI-08-08-E): Interaktion  
(Eingabe)



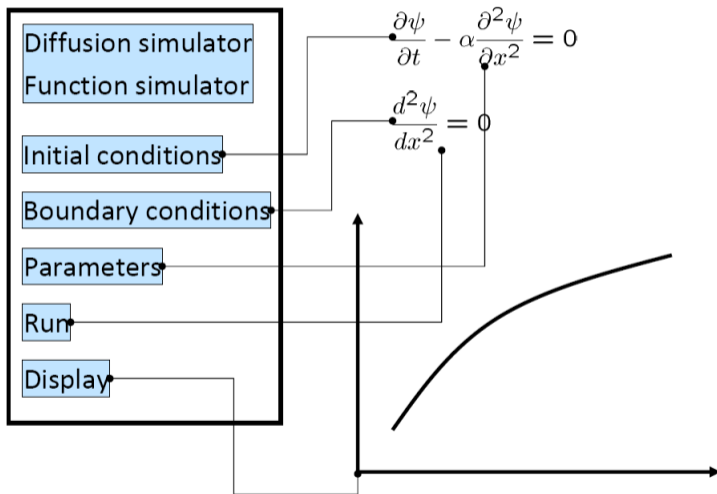
# Übungen

## Gerinnehydraulik

### C++ und Python-Versionen

# Konzept

## Objekt-Orientierung (OOP)



```
1 int main(int argc, char *argv[])
2 {
3     // 1-Geometrie
4     // 2-Anfangsbedingungen
5     // 3-Randbedingungen
6     // 4-Parameter
7     // 5-Berechnungsgroessen
8     // 6-Berechnung (Newton-Verfahren)
9     // 7-Ausgabe der Ergebnisse
10    // File
11    // x-y Plot
12 }
13
14 class channel (::: geo, ic, bc, mat, run, out)
```

**Listing:** Programmstruktur und OOP

$$h_{k+1} = h_k - \frac{f(h_k)}{f'(h_k)} = h_k - \frac{N}{D} \quad (58)$$

```
1 void channel::newton-step
2 {...
3   double N,N1,N2,N3,D,D1,D2,D21,D22;
4   for(int i=0;i<n-1;i++)
5   {
6     N1 = pow(discharge,2)/pow(wetted_cross_section[i+1],2) + gravity*u_old[i+1];
7     N2 = pow(discharge,2)/pow(wetted_cross_section[i],2) + gravity*u_old[i];
8     N3 = gravity*(bed_slope - (friction_slope[i+1]+friction_slope[i])/2.)*(x[i+1]-x[i]);
9     N = N1 - N2 - N3;
10    D1 = pow(discharge,2)/pow(wetted_cross_section[i],3) * (bottom_width+2.*m*u_old[i]) - gravity;
11    D21 = friction_law_exponent*2.*(sqrt(1+m*m))/wetted_perimeter[i];
12    D22 = (1.+friction_law_exponent)/wetted_cross_section[i] * (bottom_width+2.*m*u_old[i]);
13    D2 = gravity*friction_slope[i]*(D21-D22)*(x[i+1]-x[i]);
14    D = D1 + D2;
15    u_new[i] = u_old[i] - N/D;
16  }
17  ...
18 }
```

Listing: Newton-Schritt

# Definition der konstanten Modellparameter

## EX29-Gerinnehydraulik

```
1 // Parameter
2 double discharge = 0.05; // Volumenfließrate [m3/s]
3 double gravity = 9.81; // [m/s2]
4 double friction_law_exponent = 0.5; // Chezy, Manning-Strickler [-]
5 double error_tolerance = 1e-3; // [m]
6 double bed_slope = 0.0004; // [m/m]
7 double bottom_width = 1.; // [m]
8 double m = 1.; //
9 double friction_coefficient = 10.; //
```

# Definition der funktionalen Modellparameter

## EX29-Gerinnehydraulik

```
1 // Berechnungsgroessen
2 // Newton-Verfahren
3 double wetted_perimeter[n];
4 double wetted_cross_section[n];
5 double water_level_elevation[n];
6 // Abgeleitete Groessen (Ausgabe)
7 double hydraulic_radius[n];
8 double flow_velocity[n];
9 double Froude_number[n];
10 double friction_slope[n];
```

# Berechnung der funktionalen Modellparameter

## EX29-Gerinnehydraulik

```
1  for(int i=0;i<n;i++)
2  {
3      wetted_perimeter[i] = bottom_width + 2.*sqrt(1.+m*m)*u_old[i];
4      wetted_cross_section[i] = (bottom_width + m*u_old[i])*u_old[i];
5      hydraulic_radius[i] = wetted_cross_section[i] / wetted_perimeter[i];
6      water_level_elevation[i] = bottom_elevation[i] + u_old[i];
7      flow_velocity[i] = discharge/wetted_cross_section[i];
8      Froude_number[i] = flow_velocity[i]/(sqrt(gravity*
9      wetted_cross_section[i]\
10         /sqrt(bottom_width*bottom_width+4.*m*
11         wetted_cross_section[i])));
12     friction_slope[i] = pow(flow_velocity[i]/(friction_coefficient*
13         pow(hydraulic_radius[i],friction_law_exponent))
14     ,2);
15 }
```

```
1 void channel::out
2 {...
3   ofstream out_file("out.txt");
4   out_file.precision(4);
5   out_file << "Water depth (old):\t";
6   for(int i=0;i<n;i++)
7   {
8     out_file << "\t" << u_old[i] << " ";
9   }
10  out_file << endl;
11  ...
12  out_file << "Water depth (new):\t";
13  for(int i=0;i<n;i++)
14  {
15    out_file << "\t" << u_new[i] << " ";
16  }
17  out_file << endl;
18  out_file.close();
19  ...}
20 }
```

out.txt - Editor												
Datei Bearbeiten Format Ansicht ?												
0.25												
0.1												
Iteration: 0												
water depth (old):		0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.1
wetted perimeter:		1.71	1.71	1.71	1.71	1.71	1.71	1.71	1.71	1.71	1.71	1.28
wetted cross section:	0.313	0.313	0.313	0.313	0.313	0.313	0.313	0.313	0.313	0.313	0.313	0.11
Hydraulic radius:		0.183	0.183	0.183	0.183	0.183	0.183	0.183	0.183	0.183	0.183	0.0857
water level elevation:	0.29	0.286	0.282	0.278	0.274	0.27	0.266	0.262	0.258	0.254	0.1	
Flow velocity:	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.455
Froude number:	0.112	0.112	0.112	0.112	0.112	0.112	0.112	0.112	0.112	0.112	0.112	0.479
Friction slope:	0.0014	0.0014	0.0014	0.0014	0.0014	0.0014	0.0014	0.0014	0.0014	0.0014	0.0014	0.0241
water depth (new):		0.259	0.259	0.259	0.259	0.259	0.259	0.259	0.259	0.259	0.259	0.242
Iteration: 1												
water depth (old):		0.259	0.259	0.259	0.259	0.259	0.259	0.259	0.259	0.259	0.259	0.242
wetted perimeter:		1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.73	1.69
wetted cross section:	0.326	0.326	0.326	0.326	0.326	0.326	0.326	0.326	0.326	0.326	0.301	0.11
Hydraulic radius:		0.188	0.188	0.188	0.188	0.188	0.188	0.188	0.188	0.188	0.188	0.179
water level elevation:	0.299	0.295	0.291	0.287	0.283	0.279	0.275	0.271	0.267	0.264	0.1	
Flow velocity:	0.153	0.153	0.153	0.153	0.153	0.153	0.153	0.153	0.153	0.166	0.166	0.455
Froude number:	0.105	0.105	0.105	0.105	0.105	0.105	0.105	0.105	0.105	0.118	0.118	0.479
Friction slope:	0.00125	0.00125	0.00125	0.00125	0.00125	0.00125	0.00125	0.00125	0.00125	0.00125	0.00125	0.00125
water depth (new):		0.267	0.267	0.267	0.267	0.267	0.267	0.267	0.267	0.267	0.253	0.242
Iteration: 2												
water depth (old):		0.267	0.267	0.267	0.267	0.267	0.267	0.267	0.267	0.267	0.253	0.242
wetted perimeter:		1.76	1.76	1.76	1.76	1.76	1.76	1.76	1.76	1.76	1.72	1.69
wetted cross section:	0.339	0.339	0.339	0.339	0.339	0.339	0.339	0.339	0.339	0.317	0.301	0.11
Hydraulic radius:		0.193	0.193	0.193	0.193	0.193	0.193	0.193	0.193	0.185	0.185	0.179
water level elevation:	0.307	0.303	0.299	0.295	0.291	0.287	0.283	0.279	0.261	0.246	0.1	
Flow velocity:	0.148	0.148	0.148	0.148	0.148	0.148	0.148	0.148	0.158	0.166	0.166	0.455
Froude number:	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.11	0.118	0.118	0.479
Friction slope:	0.00113	0.00113	0.00113	0.00113	0.00113	0.00113	0.00113	0.00113	0.00113	0.00113	0.00113	0.00113
water depth (new):		0.274	0.274	0.274	0.274	0.274	0.274	0.274	0.274	0.262	0.253	0.242
Iteration: 3												
water depth (old):		0.274	0.274	0.274	0.274	0.274	0.274	0.274	0.274	0.262	0.253	0.242
wetted perimeter:		1.78	1.78	1.78	1.78	1.78	1.78	1.78	1.78	1.74	1.72	1.69
wetted cross section:	0.349	0.349	0.349	0.349	0.349	0.349	0.349	0.349	0.331	0.317	0.301	0.11
Hydraulic radius:		0.197	0.197	0.197	0.197	0.197	0.197	0.197	0.19	0.185	0.185	0.179

$$h_{k+1} = h_k - \frac{f(h_k)}{f'(h_k)} = h_k - \frac{N}{D} \quad (59)$$

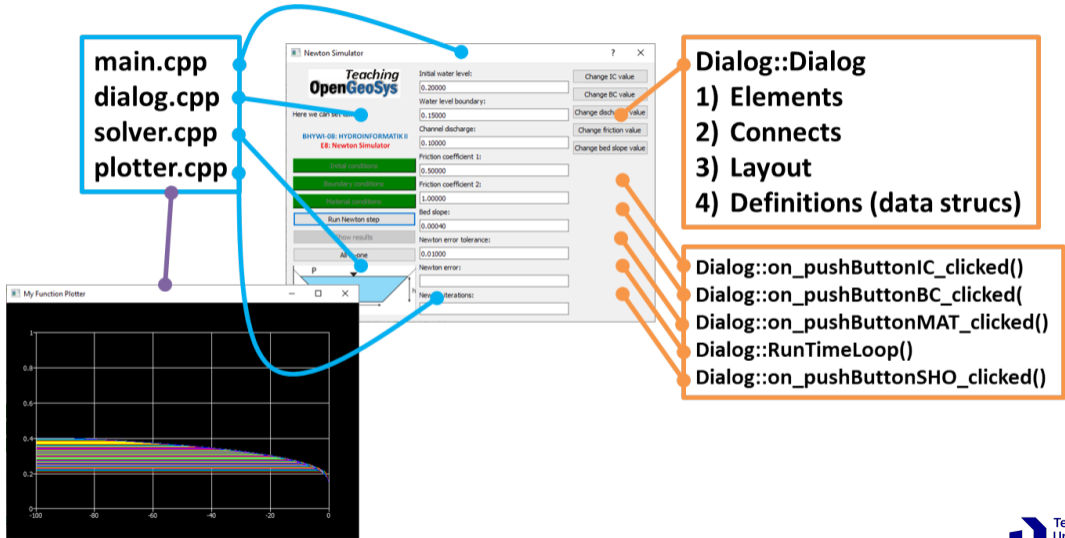
```
1 // Newton iteration loop
2 for(int k=0;k<kn;k++)
3 {...
4   RunNewtonStep();
5 ...}
```

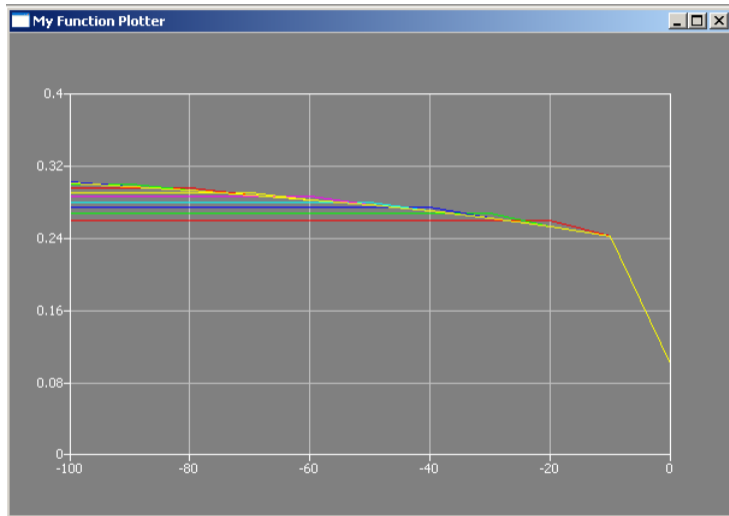
bis  $\| h_{k+1} - h_k \| < \epsilon$  (60)

# Anlage

## Gerinnehydraulik

### GUI (alt)





# Anlage

## Gerinnehydraulik

### Python-Version

