# Hydroinformatik - SoSe 2025

# UW-BHW-414-I1: Partielle Differentialgleichungen

Prof. Dr.-Ing. habil. Olaf Kolditz

[1]Helmholtz Centre for Environmental Research – UFZ, Leipzig

[2]Technische Universität Dresden – TUD, Dresden

[3]Center for Advanced Water Research – CAWR

[4]TUBAF-UFZ Center for Environmental Geosciences – C-EGS, Freiberg / Leipzig

Dresden, 20.06.2025

TECHNISCHE UNIVERSITÄT DRESDEN

# Zeitplan: Hydroinformatik I+II

Sommersemester 2025

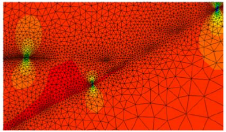
Stand: 08.05.2025

| Nr. | KW | Datum | ID | Thema |
| --- | --- | --- | --- | --- |
| 01 | 15 | 11.04.2025 | UW-BHW-414-A | Einführung, Werkzeuge#1, Hello World |
| 03 | 17 | 25.04.2025 | UW-BHW-414-B | Umweltinformatik, Werkzeuge#2 (git), Datentypen |
| 05 | 18 | 02.05.2025 | UW-BHW-414-C | Selbststudium |
| 07 | 19 | 09.05.2025 | UW-BHW-414-D | Objekt-Orientierte Programmierung: C++, Klassen |
| 09 | 20 | 16.05.2025 | UW-BHW-414-E | Python |
| 11 | 21 | 23.05.2025 | UW-BHW-414-F | Modellierung, Digitalisierung, Wasser 4.0 |
| 13 | 22 | 30.05.2025 | UW-BHW-414-G | KI, Maschinelles Lernen, Neuronale Netzwerke |
| 15 | 23 | 06.06.2025 | UW-BHW-414-H | Kontinuumsmechanik, Hydromechanik |
| 05 | 18 | 13.06.2025 | | Vorlesungsfreie Woche |
| 17 | 25 | 20.06.2025 | UW-BHW-414-I | Differentialgleichungen, Näherungsverfahren |
| 19 | 26 | 27.06.2025 | UW-BHW-414-J | Finite-Differenzen, explizite Verfahren |
| 21 | 27 | 04.07.2025 | UW-BHW-414-K | Finite-Differenzen, implizite Verfahren |
| 24 | 28 | 11.07.2025 | UW-BHW-414-L | Gerinnehydraulik, Grundwasserhydraulik |
| 25 | 29 | 18.07.2025 | UW-BHW-414-M | Zusammenfassung, Klausurvorbereitung |

TECHNISCHE UNIVERSITÄT DRESDEN

# Vorlesungskonzept

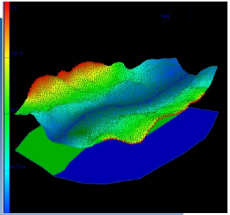
$$\frac{d\psi}{dt} = \frac{\partial\psi}{\partial t} + \mathbf{v}^E \nabla\psi$$



Basics

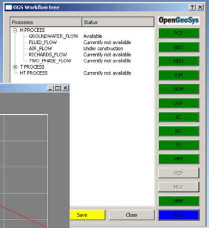
Mechanik

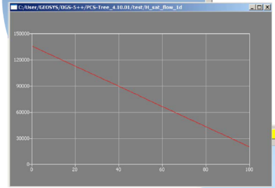
Anwendung



Numerische

Methoden



Programmierung

Visual C++

Prozessverständnis

TECHNISCHE UNIVERSITÄT DRESDEN

# Fahrplan für heute



TECHNISCHE UNIVERSITÄT DRESDEN

# Dimensionsanalyse

Navier-Stokes Equation (NSE)

$$\boxed{\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = \mathbf{f}^e - \frac{1}{\rho}\nabla p + \nu \Delta \mathbf{v}} \tag{1}$$

Ableitungen:

$$\frac{\partial}{\partial t}$$

$$\nabla = \frac{\partial}{\partial x}$$

$$\Delta = \frac{\partial^2}{\partial x^2}$$

*K-Krage: Erläutern Sie die physikalische Bedeutung der Terme der NSE.*

# Mathematical Classification (1.5)

A common formulation of a PDE in $\mathcal{R}^3$ is

$$L(\psi) = F(t, x_i, \psi, \frac{\partial \psi}{\partial x_i}, ..., \frac{\partial^n \psi}{\partial x_i^n}) = 0 \qquad , \qquad i = 3 \tag{2}$$

where $L$ is s differential operator. Second-order PDE with two independent variables are given by

$$A\frac{\partial^2 \psi}{\partial x^2} + B\frac{\partial^2 \psi}{\partial x \partial y} + C\frac{\partial^2 \psi}{\partial y^2} + D\frac{\partial \psi}{\partial x} + E\frac{\partial \psi}{\partial y} + F\psi + G = 0 \tag{3}$$

Second-order PDEs with more independent variables can be classified by examination of the eigenvalues of the matrix $a_{ij}$.

$$\sum_i \sum_j a_{ij} \frac{\partial \psi^2}{\partial x_i \partial x_j} + G = 0 \quad , \quad a_{ii} = \lambda_i \quad \text{Eigenvalues} \tag{4}$$

TECHNISCHE UNIVERSITÄT DRESDEN

# Mathematical Classification (1.5)

| PDE type | Discriminant | Eigenvalues | Canonical form | Example |
|---|---|---|---|---|
| Elliptic | $B^2 - 4AC < 0$<br>complex characteristics | $\forall\lambda > 0$<br>equal signs | $\frac{\partial^2\psi}{\partial\xi^2} + \frac{\partial^2\psi}{\partial\eta^2} = 0$ | Laplace equation |
| Parabolic | $B^2 - 4AC = 0$ | $\exists\lambda = 0$ | $\frac{\partial^2\psi}{\partial\eta^2} = G$ | Diffusion, Burgers equations |
| Hyperbolic | $B^2 - 4AC > 0$<br>real characteristics | $\exists\lambda < 0$<br>different signs | $\frac{\partial^2\psi}{\partial\xi^2} - \frac{\partial^2\psi}{\partial\eta^2} = 0$ | Wave equation |

# General Balance Equation (1.1.7)

- Integral form

$$\int_\Omega \frac{d\psi}{dt} d\Omega = \int_\Omega \frac{\partial\psi}{\partial t} d\Omega + \int_\Omega \nabla \cdot (\mathbf{v}\psi) d\Omega - \int_\Omega \nabla \cdot (\mathbf{D}^\psi \nabla\psi) d\Omega = \int_\Omega Q^\psi d\Omega \tag{5}$$

- Differential form

$$\frac{d\psi}{dt} = \frac{\partial\psi}{\partial t} + \nabla \cdot (\mathbf{v}\psi) - \nabla \cdot (\mathbf{D}^\psi \nabla\psi) = \mathbf{Q}^\psi \tag{6}$$

TECHNISCHE UNIVERSITÄT DRESDEN

# PDE: Definition

$$\frac{d\psi}{dt} = \frac{\partial \psi}{\partial t} + \nabla \cdot (\mathbf{v}\psi) - \nabla \cdot (\mathbf{D}^{\psi} \nabla \psi) = Q^{\psi} \tag{7}$$

---

A common formulation of a PDE in $\mathcal{R}^3$ is

$$L(\psi) = F(t, x_i, \psi, \frac{\partial \psi}{\partial x_i}, ..., \frac{\partial^n \psi}{\partial x_i^n}) = 0 \qquad , \qquad i = 3 \tag{8}$$

where $L$ is s differential operator.

# PDE: Klassifikation

$$\frac{d\psi}{dt} = \frac{\partial \psi}{\partial t} + \nabla \cdot (\mathbf{v}\psi) - \nabla \cdot (\mathbf{D}^{\psi} \nabla \psi) = Q^{\psi} \tag{9}$$

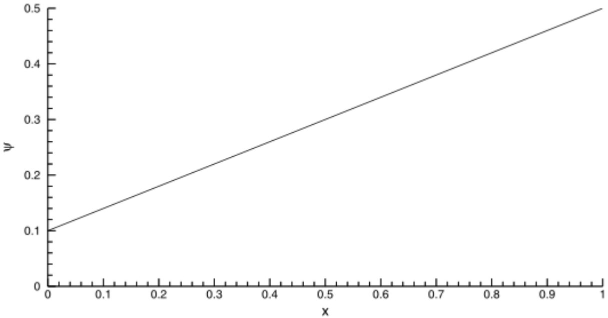
| Physical problem | Math. problem | Examples |
|---|---|---|
| Equilibrium problems | Elliptic equations | Irrotational incompressible flow<br>Inviscid incompressible flow<br>Steady state heat conduction |
| Propagation problems (infinite propagation speed) | Parabolic equations | Unsteady viscous flow<br>Transient heat transfer |
| Propagation problems (finite propagation speed) | Hyperbolic equations | Wave propagation (vibration)<br>Inviscid supersonic flow |

- Parabolisch: Diffusion, Gerinne (nichtlinear)
- Elliptisch: Grundwasser (stationär)

# PDE: Elliptic Equation 1-D

$$\frac{d^2\psi}{dx^2} = 0 \tag{10}$$

$$\psi = ax + b \tag{11}$$



# PDE: Elliptic Equation 2-D

The prototype of an elliptic equation is the Laplace equation.

$$\frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} = 0 \tag{12}$$

By substitution it can be easily verified that the exact solution of the Laplace equation is

$$\psi = sin(\pi x)exp(-\pi y) \tag{13}$$

$$\frac{\partial\psi}{\partial x} = \pi cos(\pi x)exp(-\pi y), \frac{\partial\psi}{\partial y} = ..., \frac{\partial^2\psi}{\partial x^2} = ..., \frac{\partial^2\psi}{\partial y^2} = ... \tag{14}$$

$\Rightarrow$ Hausaufgabe (am Ende)

TECHNISCHE UNIVERSITÄT DRESDEN

# PDE: Elliptic Equation 2-D

The prototype of an elliptic equation is the Laplace equation.

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0 \tag{15}$$

$$\psi(x, y) = \int \left( \frac{x - L}{(x - L)^2 + y^2} - \frac{x + L}{(x + L)^2 + y^2} \right) dx\, dy \tag{16}$$

$$v_x = \frac{\partial \psi}{\partial x} = \frac{x - L}{(x - L)^2 + y^2} - \frac{x + L}{(x + L)^2 + y^2} \tag{17}$$

$$v_y = \frac{\partial \psi}{\partial y} = \frac{y}{(x - L)^2 + y^2} - \frac{y}{(x + L)^2 + y^2} \tag{18}$$

$\Rightarrow$ Randbedingungen sind wichtig

# Divergenzfreie Strömung

## Übung: EX04-divergenzfreie-stroemung.py

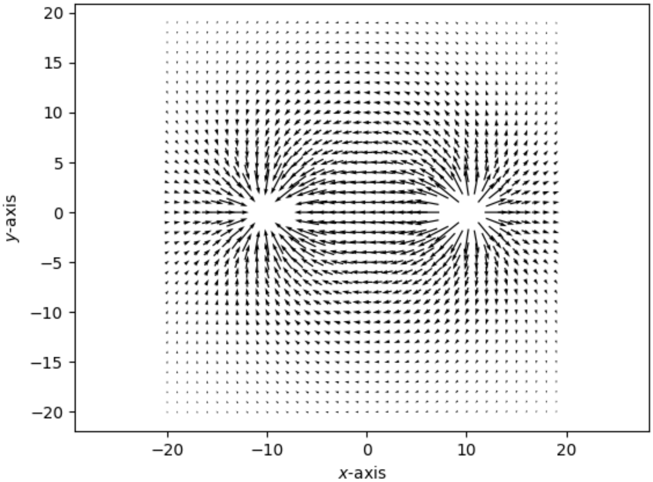
```python
import numpy as np
import matplotlib.pyplot as plt
# set up a normalized grid:
dim= 20
xarray= np.arange(-dim,dim)
yarray= np.arange(-dim,dim)
# (fluid) flow from a source at L to a sink at -L:
L = dim/2
x,y = np.meshgrid(xarray,yarray)
vx = (x-L)/((x-L)**2+y**2) - (x+L)/((x+L)**2 +y**2)
vy = y/((x-L)**2+y**2) - y/((x+L)**2 +y**2)
# plot the flow lines:
plt.figure()
plt.quiver(x,y, vx, vy, pivot='mid')
plt.xlabel("$x$-axis")
plt.ylabel("$y$-axis")
plt.axis('equal')
plt.show()
```

Source: https://auckland.figshare.com/articles/dataset/Chapter_6_Divergence_of_a_vector_field/5732421

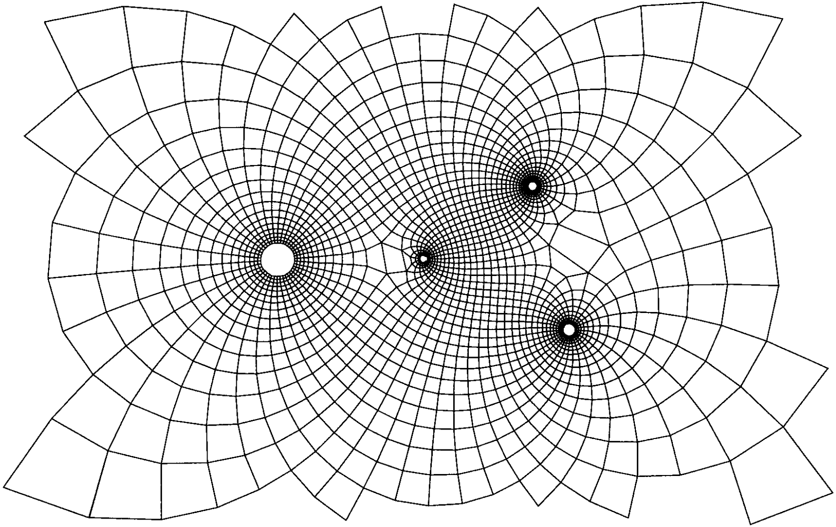
**Listing:** Python code for divergence-free flow (div v equals 0)

# Divergenzfreie Strömung

## Übung: EX04-divergenzfreie-stroemung.py



# PDE: Elliptic Equation 2-D



# PDE: Parabolic Equation 1-D

$$\frac{\partial \psi}{\partial t} = \alpha \frac{\partial^2 \psi}{\partial x^2} \tag{19}$$

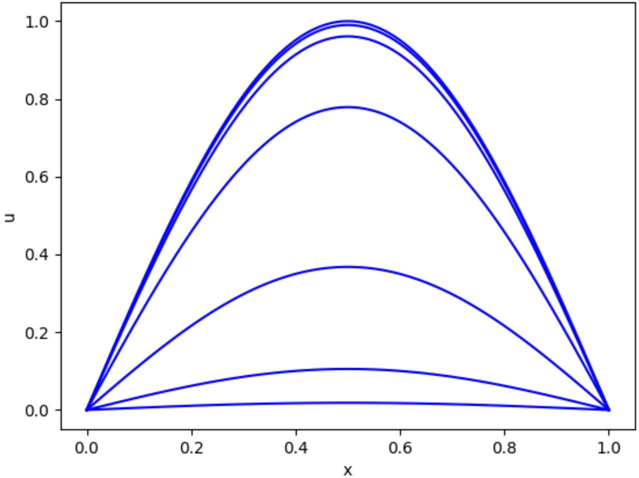
Proof the following solutions on correctness:

$$\psi(t,x) = sin(\sqrt{\pi\alpha}x)exp(-\pi t) \tag{20}$$

$$\psi(t,x) = sin(\frac{\pi}{\sqrt{\alpha}}x)exp(-\pi^2 t) \tag{21}$$

$$\psi(t,x) = sin(\pi x)exp(-\alpha\pi^2 t) \tag{22}$$

# PDE: Parabolic Equation 1-D




https://github.com/OlafKolditz/Hydroinformatik-II/blob/master/EX06-parabolische-gleichung-1D.py

# PDE: Hyperbolic Equation 1-D

$$\frac{\partial \psi}{\partial t} - v_x \frac{\partial \psi}{\partial x} = 0 \tag{23}$$

$\Rightarrow$ Übung

TECHNISCHE UNIVERSITÄT DRESDEN

# PDE: Hyperbolic Equation 1-D

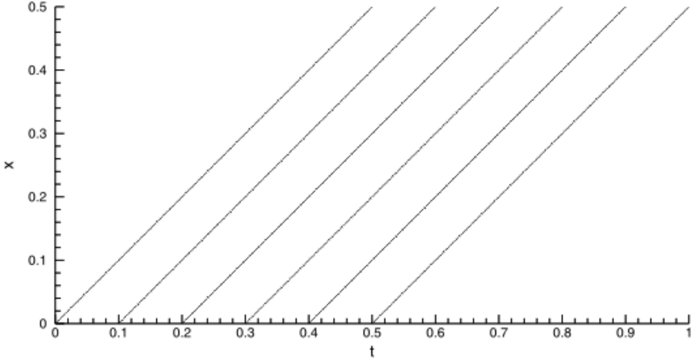
$$\frac{\partial^2 \psi}{\partial^2 t} - c^2 \frac{\partial^2 \psi}{\partial^2 x} = 0 \tag{24}$$

$$\psi(t, x) = a \cos(\frac{\pi c t}{L}) \sin(\frac{\pi x}{L}) \tag{25}$$

TECHNISCHE UNIVERSITÄT DRESDEN

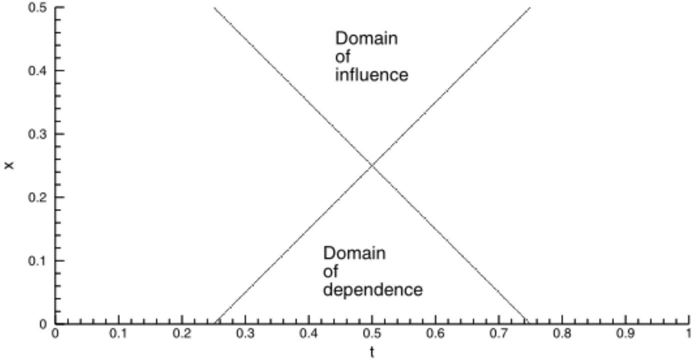
# PDE: Hyperbolic Equation 1-D - Characteristics

$$A\frac{\partial\psi}{\partial t} - B\frac{\partial\psi}{\partial x} = 0 \tag{26}$$



# PDE: Hyperbolic Equation 1-D - Characteristics

$$A\frac{\partial \psi}{\partial t} - B\frac{\partial \psi}{\partial x} = 0 \quad (27)$$



# PDE: Hyperbolic-Parabolic Equation 1-D

Transportgleichung (advection-diffusion equation ADE)

$$\frac{\partial \psi}{\partial t} - v_x \frac{\partial \psi}{\partial x} + D_{xx} \frac{\partial^2 \psi}{\partial x^2} = 0 \qquad (28)$$

$\Rightarrow$ Übung

TECHNISCHE UNIVERSITÄT DRESDEN

# PDE: Hyperbolic-Parabolic Equation 1-D

Transportgleichung (advection-diffusion equation ADE)

$$\frac{\partial \psi}{\partial t} - v_x \frac{\partial \psi}{\partial x} + D_{xx} \frac{\partial^2 \psi}{\partial x^2} = 0 \tag{28}$$

$\Rightarrow$ Übung

$$\frac{d\psi}{dt} = \frac{\partial \psi}{\partial t} + \nabla \cdot (\mathbf{v}\psi) - \nabla \cdot (\mathbf{D}^{\psi} \nabla \psi) = \mathbf{Q}^{\psi} \tag{29}$$

# PDE: Equation Types

The following table gives typical examples of balance equations for the denoted quantities and their PDE types.

| Physics | Equation structure | Examples |
|---|---|---|
| Continuity | $\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0$ | Laplace equation |
| Mass/energy | $\frac{\partial \psi}{\partial t} + u \frac{\partial \psi}{\partial x} - \alpha \frac{\partial^2 \psi}{\partial x^2} = 0$ | Fokker-Planck equation |
| Momentum | $\frac{\partial \psi}{\partial t} + \psi \frac{\partial \psi}{\partial x} - \frac{\partial}{\partial x}[\alpha(\psi) \frac{\partial \psi}{\partial x}] = 0$ | Navier-Stokes equation |

# Boundary Conditions I

The following table gives an overview on common boundary condition types and its mathematical representation.

**Table:** Boundary conditions types

| Type of BC | Mathematical Meaning | Physical Meaning |
|---|---|---|
| Dirichlet type | $\psi$ | prescribed value<br>potential surface |
| Neumann type | $\nabla\psi$ | prescribed flux<br>stream surface |
| Cauchy type | $\psi + A\nabla\psi$ | resistance between<br>potential and stream surface |

TECHNISCHE UNIVERSITÄT DRESDEN

# Boundary Conditions II

To describe conditions at boundaries we can use flux expressions of conservation quantities.

**Table:** Fluxes through surface boundaries

| Quantity | Flux term |
|---|---|
| Mass | $\rho \mathbf{v}$ |
| Momentum | $\rho \mathbf{v}\mathbf{v} - \sigma$ |
| Energy | $\rho e \mathbf{v} - \lambda \nabla \mathbf{T}$ |

# Hausaufgabe 03/BHYWI04-2020 vom 15.05.2020

## Partielle Differentialgleichungen

**Aufgabe**: Prüfen sie die Gültigkeit der Lösungen für die partiellen Differentialgleichungen: (14), (20), (21), (22), (25).

**Lösungsweg**: Berechnen sie hierfür die entsprechenden partiellen Ableitungen und setzen sie diese dann in die entsprechenden Gleichungen (14), (20), (21), (22), (25) ein.

$$\frac{\partial \psi}{\partial t} = \ldots \quad , \quad \frac{\partial \psi}{\partial x} = \ldots \tag{30}$$

$$\frac{\partial^2 \psi}{\partial x^2} = \ldots \quad , \quad \frac{\partial^2 \psi}{\partial y^2} = \ldots \tag{31}$$

TECHNISCHE UNIVERSITÄT DRESDEN

# Farbcode

Vorlesung

Übung

Hausaufgabe

Klausur

# Übungen

# Übungen: Hydroinformatik

Hydroinformatik II - HyBHW-1-02 `https://github.com/OlafKolditz/HYDROINFORMATIK-II`

- EX01: Jupyter Notebook
- EX02: Python: matplotlib
- EX03: Kontinnumsmechanik: Skalarprodukt
- EX04: Hydromechanik: Divergenzfreie Strömung
- EX05: Analytische Lösung: Elliptische Gleichung
- EX06: Analytische Lösung: Parabolische Gleichung (Diffusion)
- EX07: Analytische Lösung: Transportgleichung (ADE)
- EX08: Finite-Differenzen-Methode (FDM) explizit
- EX09: Finite-Differenzen-Methode (FDM) implizit
- EX10: Gerinnehydraulik

```
Quellcode
...
```

**Listing:** Quellcode für Übungen (C++ und Python)

TECHNISCHE UNIVERSITÄT DRESDEN

# Hausaufgaben

1. Skalarprodukt: Schreiben sie das Skalarprodukt $\nabla \cdot \mathbf{v}$ in Komponentenschreibweise.
2. Mechanik: Was ist $\mathbf{v} \cdot \nabla\psi$? Physikalische Bedeutung des Terms
3. Mechanik: Was ist $\Phi^{\psi}$? Physikalische Bedeutung des Terms
4. Hydromechanik: Komponentenschreibweise $\nabla \cdot (\mathbf{v}\psi)$
5. Hydromechanik: Komponentenschreibweise $\nabla \cdot (\mathbf{D}^{\psi}\nabla\psi)$
6. Analytik: Prüfen Sie die Gültigkeit einer der Lösungen für die Diffusionsgleichung: (20), (21), (22) (siehe Vorlesung 5 $>>$ HyBHW-S2-01-V05).
7. Analytik: Stellen Sie die ausgewählte analytische Lösung für die 1-D parabolische Differentialgleichung unter Verwend der Übung EX06-parabolische-gleichung-1D.py dar. Ergänzen Sie Ihren Namen oder Matrikelnummer mit dem Befehl `plt.title("Name oder Matrikelnummer")`.
8. Numerik: Darstellung der numerischen Lösung (explizite FDM) für die 1-D parabolische Differentialgleichung (EX08-fdm-explicit-python). Produzieren Sie eine stabile und instabile Lösung.
9. Numerik: Darstellung der numerischen Lösung (implizite FDM) für die 1-D parabolische Differentialgleichung (EX09-fdm-implicit-python). Produzieren Sie die stationäre Lösung.

# Hausaufgaben: Hydroinformatik

## Beispiel: Aufgaben 7-9

- zum Internet-Repository gehen (Webseite)
- Python-File editieren (Matrikel-Nummer oder Name)
- Programme zum Rechnen und Darstellen ausführen
- Ergebnis (Abbildung) in die Hausaufgaben einfügen

TECHNISCHE UNIVERSITÄT DRESDEN

# Hydroinformatik - SoSe 2025

# UW-BHW-I2: Einführung in Näherungsverfahren

Prof. Dr.-Ing. habil. Olaf Kolditz

[1]Helmholtz Centre for Environmental Research – UFZ, Leipzig

[2]Technische Universität Dresden – TUD, Dresden

[3]Center for Advanced Water Research – CAWR

[4]TUBAF-UFZ Center for Environmental Geosciences – C-EGS, Freiberg / Leipzig

Dresden, 20.06.2025

TECHNISCHE UNIVERSITÄT DRESDEN

# Zeitplan: Hydroinformatik I+II

## Sommersemester 2025

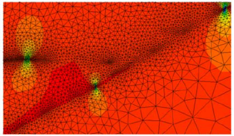
Stand: 08.05.2025

| Nr. | KW | Datum | ID | Thema |
|---|---|---|---|---|
| 01 | 15 | 11.04.2025 | UW-BHW-414-A | Einführung, Werkzeuge#1, Hello World |
| 03 | 17 | 25.04.2025 | UW-BHW-414-B | Umweltinformatik, Werkzeuge#2 (git), Datentypen |
| 05 | 18 | 02.05.2025 | UW-BHW-414-C | Selbststudium |
| 07 | 19 | 09.05.2025 | UW-BHW-414-D | Objekt-Orientierte Programmierung: C++, Klassen |
| 09 | 20 | 16.05.2025 | UW-BHW-414-E | Python |
| 11 | 21 | 23.05.2025 | UW-BHW-414-F | Modellierung, Digitalisierung, Wasser 4.0 |
| 13 | 22 | 30.05.2025 | UW-BHW-414-G | KI, Maschinelles Lernen, Neuronale Netzwerke |
| 15 | 23 | 06.06.2025 | UW-BHW-414-H | Kontinuumsmechanik, Hydromechanik |
| 05 | 18 | 13.06.2025 | | Vorlesungsfreie Woche |
| 17 | 25 | 20.06.2025 | UW-BHW-414-I | Differentialgleichungen, Näherungsverfahren |
| 19 | 26 | 27.06.2025 | UW-BHW-414-J | Finite-Differenzen, explizite Verfahren |
| 21 | 27 | 04.07.2025 | UW-BHW-414-K | Finite-Differenzen, implizite Verfahren |
| 24 | 28 | 11.07.2025 | UW-BHW-414-L | Gerinnehydraulik, Grundwasserhydraulik |
| 25 | 29 | 18.07.2025 | UW-BHW-414-M | Zusammenfassung, Klausurvorbereitung |

TECHNISCHE UNIVERSITÄT DRESDEN

# Konzept

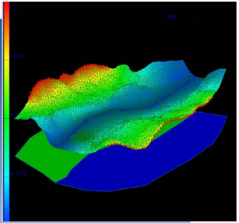
$$\frac{d\psi}{dt} = \frac{\partial\psi}{\partial t} + \mathbf{v}^E \nabla\psi$$



Basics

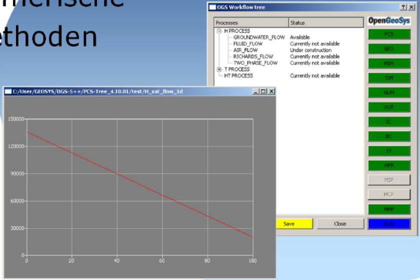
Mechanik

Anwendung



Numerische

Methoden




Programmierung

Visual C++

Prozessverständnis

# Fahrplan für heute

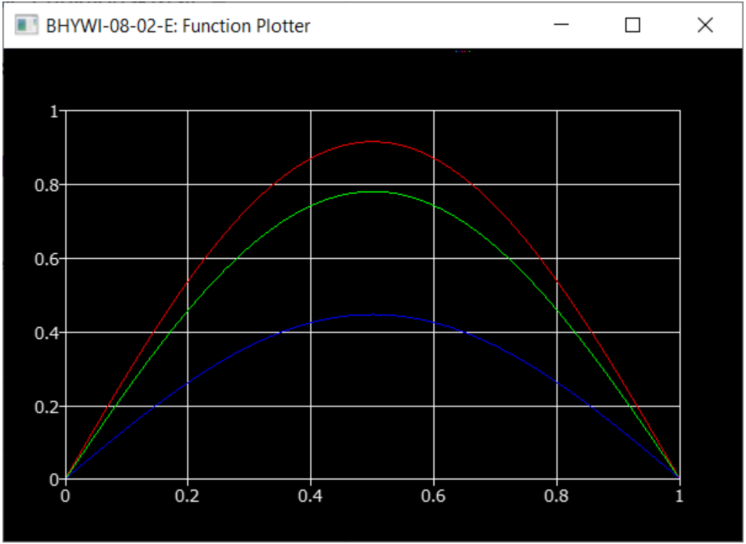


TECHNISCHE UNIVERSITÄT DRESDEN

# Demo: Funktionsrechner

TECHNISCHE UNIVERSITÄT DRESDEN

# Übung: BHYWI-08-02-E: Funktionsrechner




Parabolic equation

$$\frac{\partial \psi}{\partial t} = \alpha \frac{\partial^2 \psi}{\partial x^2} \tag{1}$$

TECHNISCHE UNIVERSITÄT DRESDEN

# Python: EX06-parabolische-Gleichung

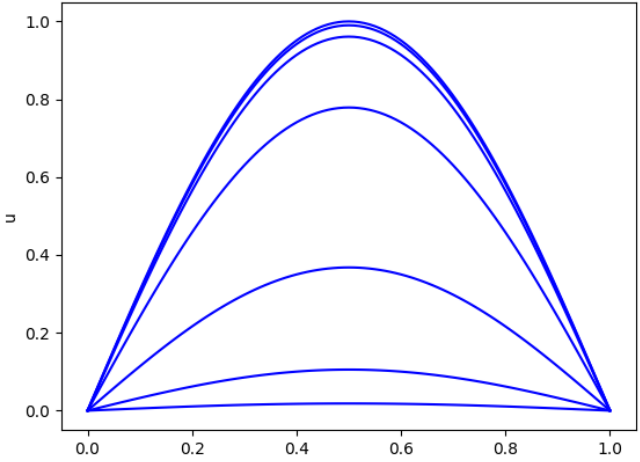
https://github.com/OlafKolditz/Hydroinformatik-II/

```python
import matplotlib.pylab as plt
PI = 3.14159265358979323846
numPoints = 100
alpha = 1.0
t = [0.01,0.1,0.2,0.5,1.0,1.5,2.0]
x = []
y = []
for n in t:
    for i in range(0,numPoints+1):
        x.append(float(i)/float(numPoints))
        #y.append(math.sin(PI*x[i]) * math.exp(-alpha*n*n))
        y.append(math.sin(math.pi*x[i]) * math.exp(-alpha*n*n))
    plt.plot(x,y,color='blue')
    x = []
    y = []
plt.xlabel('x')
plt.ylabel('u')
plt.axis('tight')
plt.savefig("diffusion-equation.png")
plt.show()
```

**Listing:** Python code for parabolic equation

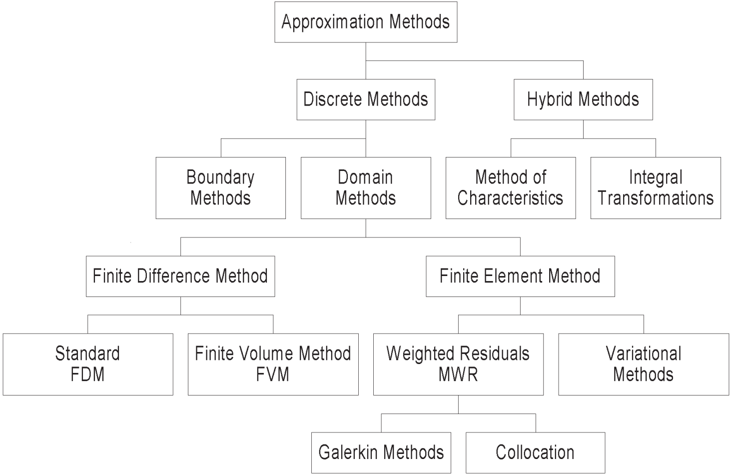
# Python: EX06-parabolische-Gleichung: Funktionsrechner

https://github.com/OlafKolditz/Hydroinformatik-II/

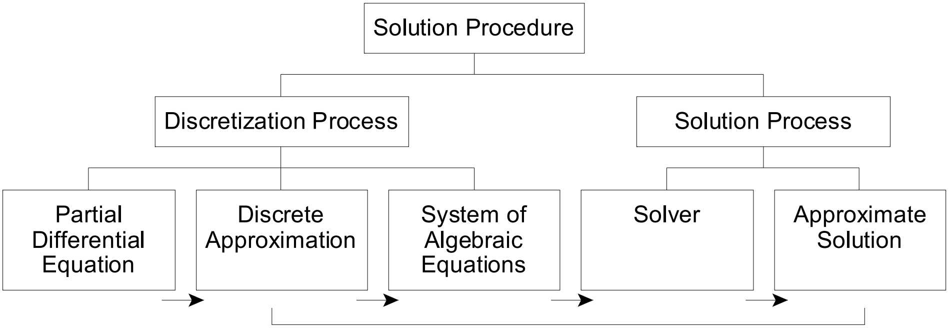


# Näherungsverfahren

# Näherungsverfahren



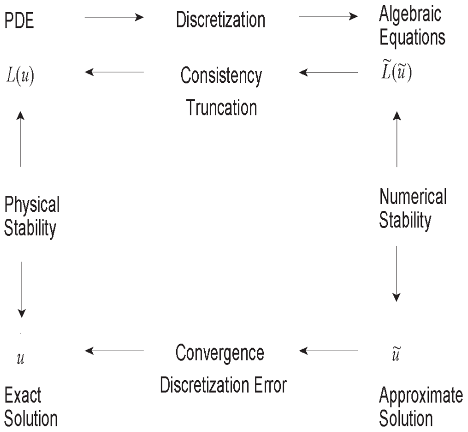
# Lösungsverfahren




Governing Equations
**Continuum Mechanics**

Numerical Methods
**Computational Mechanics**

# Definitionen



# Konvergenz

Definition: A solution of the algebraic equations which approximate a given PDE is said to be convergent if the approximate solution approaches the exact solution of the PDE for each value of the independent variable as the grid spacing tends to zero. Thus we require

$$\lim_{\Delta t, \Delta x \to 0} \mid u_j^n - u(t_n, x_j) \mid = 0 \tag{3}$$

Or in other words, the approximate solution converges to the exact one as the grid sizes becomes infinitely small. The difference between exact and approximate solution is the solution error, denoted by

$$\varepsilon_j^n = \mid u_j^n - u(t_n, x_j) \mid \tag{4}$$

TECHNISCHE UNIVERSITÄT DRESDEN

# Konsistenz

Definition: The system of algebraic equations (SAE) generated by the discretization process is said to be consistent with the original partial differential equation (PDE) if, in the limit that the grid spacing tends to zero, the SAE is equivalent to the PDE at each grid point. Thus we require

$$\lim_{\Delta t, \Delta x \to 0} \mid \tilde{L}(u_j^n) - L(u[t_n, x_j]) \mid = 0 \tag{5}$$

TECHNISCHE UNIVERSITÄT DRESDEN

# Stabilität

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} - \alpha\frac{\partial^2 u}{\partial x^2} = 0 \tag{6}$$

- Courant-Zahl

$$Cr = \frac{v\Delta t}{\Delta x} \leqslant 1 \tag{7}$$

- Peclet-Zahl

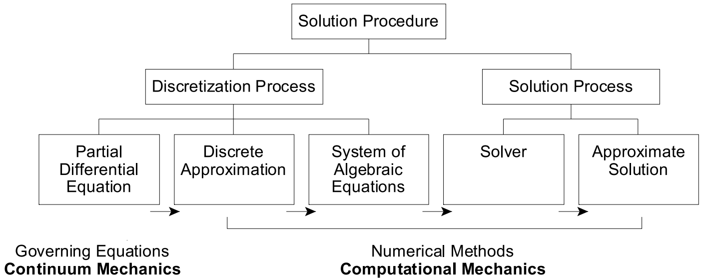
$$Pe = \frac{v\Delta x}{\alpha} \leqslant 2 \tag{8}$$

- Neumann-Zahl

$$Ne = \frac{\alpha\Delta t}{\Delta x^2} \leqslant \frac{1}{2} \tag{9}$$

$$0 < Cr^2 < Ne < 1 \tag{10}$$

# Gleichungssysteme




$$\mathbf{A}(\mathbf{x})\mathbf{x} = \mathbf{b}(\mathbf{x}) \tag{11}$$

TECHNISCHE UNIVERSITÄT DRESDEN

# Gleichungslöser

The following list reveals an overview on existing methods for solving linear algebraic equation systems.

- Direct methods
  - **Gaussian elimination**
  - Block elimination (to reduce memory requirements for large problems)
  - Cholesky decomposition
  - Frontal solver
- Iterative methods
  - Linear steady methods (Jacobian, **Gauss-Seidel**, Richardson and block iteration methods)
  - Gradient methods (CG) (also denoted as Krylov subspace methods)

TECHNISCHE UNIVERSITÄT DRESDEN

# Lösen linearer Gleichungen

Application of direct methods to determine the solution of equation

$$\mathbf{x} = \mathbf{A}^{-1}\,\mathbf{b} \tag{12}$$

requires an efficient techniques to invert the system matrix.
As a first example we consider the Gaussian elimination technique. If matrix $\mathbf{A}$ is not singular (i.e. $\det \mathbf{A} \neq 0$), can be composed in following way.

$$\mathbf{P}\,\mathbf{A} = \mathbf{L}\,\mathbf{U} \tag{13}$$

with a permutation matrix $\mathbf{P}$ and the lower $\mathbf{L}$ as well as the upper matrices $\mathbf{U}$ in triangle forms.

$$\mathbf{L} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ l_{n1} & \cdots & 1 \end{bmatrix} \quad , \quad \mathbf{U} = \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & u_{nn} \end{bmatrix} \tag{14}$$

TECHNISCHE UNIVERSITÄT DRESDEN

## Gauss-Verfahren (Eliminierungsverfahren)

- 1

$$a_{11}u_1 + a_{12}u_2 = b_1 \tag{15}$$
$$a_{21}u_1 + a_{22}u_2 = b_2 \tag{16}$$

- 2

$$a_{21}\frac{a_{11}}{a_{21}}u_1 + a_{22}\frac{a_{11}}{a_{21}}u_2 = \frac{a_{11}}{a_{21}}b_2 \tag{17}$$

- 3: (19) - (17)

$$\left(\frac{a_{22}a_{11}}{a_{21}} - a_{12}\right)u_2 = \frac{a_{11}}{a_{21}}b_2 - b_1 \tag{18}$$

- 4

$$u_2 = \frac{\frac{a_{11}}{a_{21}}b_2 - b_1}{\frac{a_{22}a_{11}}{a_{21}} - a_{12}} \tag{19}$$

TECHNISCHE UNIVERSITÄT DRESDEN

# Lösen linearer Gleichungen - Iterative Verfahren

High resolution FEM leads to large equation systems with sparse system matrices. For this type of problems iterative equation solver are much more efficient than direct solvers. Concerning the application of iterative solver we have to distinguish between symmetrical and non-symmetrical system matrices with different solution methods. The efficiency of iterative algorithms, i.e. the reduction of iteration numbers, can be improved by the use of pre-conditioning techniques).

| Symmetric Matrices | Non-symmetric Matrices |
| --- | --- |
| CG | BiCG |
| Lanczos | CGStab |
| **Gauss-Seidel**, Jacobian, Richards | GMRES |
| SOR and block-iteration | CGNR |

The last two rows of solver for symmetric problems belong to the linear steady iteration methods. The algorithms for solving non-symmetrical systems are also denoted as Krylov subspace methods.

TECHNISCHE UNIVERSITÄT DRESDEN