Hydroinformatik - SoSe 2026

UW-BHW-414-14: 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, 19.06.2026

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-J	Finite-Differenzen, explizite Verfahren
21	27	03.07.2026	UW-BHW-414-K	Finite-Differenzen, implizite Verfahren
23	28	10.07.2026	UW-BHW-414-L	Gerinnehydraulik, Grundwasserhydraulik
25	29	17.07.2026	UW-BHW-414-M	Grundwasserhydraulik
27	30	24.07.2026	UW-BHW-414-N	Zusammenfassung, Klausurvorbereitung

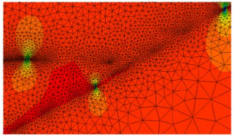
Technische
Universität
Dresden

Fahrplan für heute ...

1. UW-BHW-414-14: Einführung in Näherungsverfahren
    - Semesterplan

Konzept

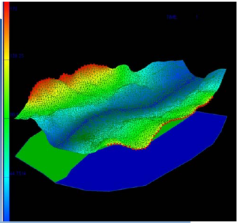
$$\frac{d\psi}{dt} = \frac{\partial\psi}{\partial t} + \mathbf{v}^E \nabla\psi$$



Basics

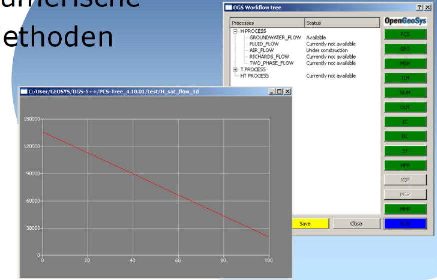
Mechanik

Anwendung



Numerische

Methoden



Processes	Status
H PROCESS	
GROUNDWATER_FLOW	Available
FLUID_FLOW	Currently not available
AIR_FLOW	Under construction
RICHARDS_FLOW	Currently not available
TWO_PHASE_FLOW	Currently not available
T PROCESS	
HT PROCESS	Currently not available

Programmierung

Visual C++

Prozessverständnis

Fahrplan für heute

0 Partielle Differentialgleichungen (PDE)

---

1 Näherungsverfahren
2 Lösungsverfahren
3 Definitionen
4 Fehler
5 Kriterien
6 Lösen von Gleichungssystemen

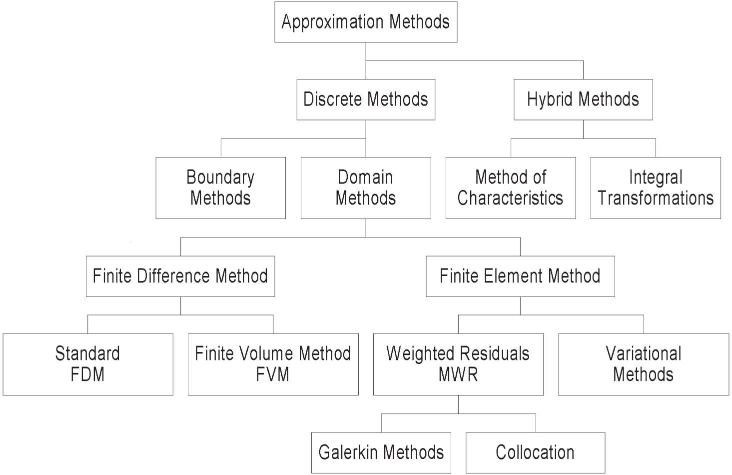
---

7 Finite-Differenzen-Verfahren

Technische Universität Dresden

Näherungsverfahren

Näherungsverfahren



Approximation Methods

Discrete Methods

Hybrid Methods

Boundary Methods

Domain Methods

Method of Characteristics

Integral Transformations

Finite Difference Method

Finite Element Method

Standard FDM

Finite Volume Method FVM

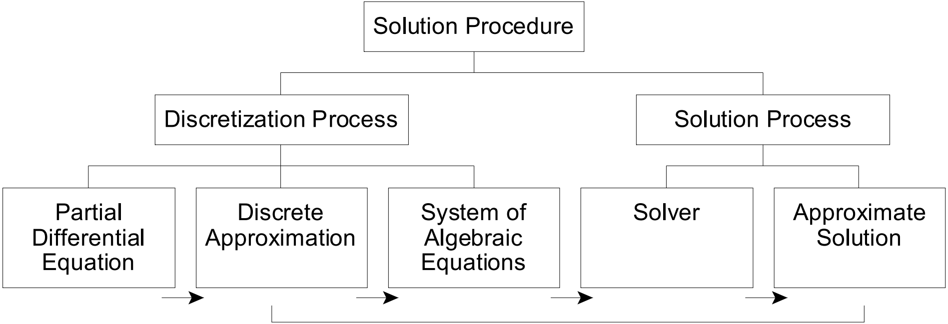
Weighted Residuals MWR

Variational Methods

Galerkin Methods

Collocation

Lösungsverfahren



Solution Procedure

Discretization Process

Solution Process

Partial Differential Equation

Discrete Approximation

System of Algebraic Equations

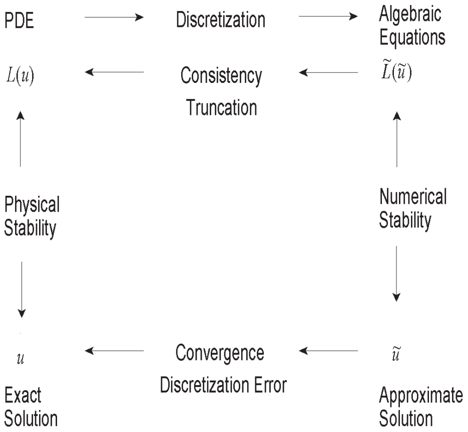
Solver

Approximate Solution

Governing Equations
Continuum Mechanics

Numerical Methods
Computational Mechanics

Definitionen



PDE

Discretization

Algebraic Equations

$L(u)$

Consistency

Truncation

$\tilde{L}(\tilde{u})$

Physical Stability

Numerical Stability

u

Exact Solution

Convergence

Discretization Error

$\tilde{u}$

Approximate Solution

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

$$\lim_{\Delta t, \Delta x \to 0} \mid u_j^n - u(t_n, x_j) \mid = 0 \tag{1}$$

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

$$\varepsilon_j^n = \mid u_j^n - u(t_n, x_j) \mid \tag{2}$$

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

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

Stabilität

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} - \alpha\frac{\partial^2 u}{\partial x^2} = 0 \tag{4}$$

- Courant-Zahl

$$Cr = \frac{v\Delta t}{\Delta x} \leqslant 1 \tag{5}$$

- Peclet-Zahl

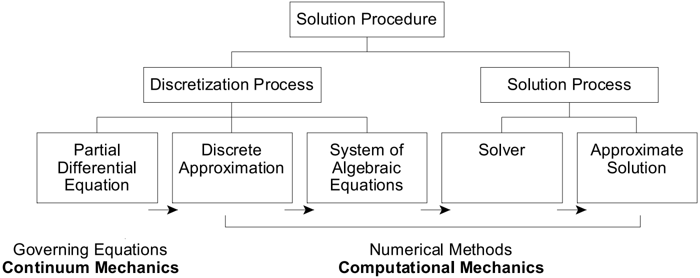
$$Pe = \frac{v\Delta x}{\alpha} \leqslant 2 \tag{6}$$

- Neumann-Zahl

$$Ne = \frac{\alpha\Delta t}{\Delta x^2} \leqslant \frac{1}{2} \tag{7}$$

$$0 < Cr^2 < Ne < 1 \tag{8}$$

Gleichungssysteme



Solution Procedure

Discretization Process

Solution Process

Partial Differential Equation

Discrete Approximation

System of Algebraic Equations

Solver

Approximate Solution

Governing Equations
Continuum Mechanics

Numerical Methods
Computational Mechanics

$$\mathbf{A}(\mathbf{x})\mathbf{x} = \mathbf{b}(\mathbf{x}) \tag{9}$$

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)

Lösen linearer Gleichungen

Application of direct methods to determine the solution of equation

$$\mathbf{x} = \mathbf{A}^{-1}\,\mathbf{b} \tag{10}$$

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

$$\mathbf{P}\,\mathbf{A} = \mathbf{L}\,\mathbf{U} \tag{11}$$

with a permutation matrix **P** and the lower **L** as well as the upper matrices **U** in triangle forms.

$$\mathbf{L} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ l_{n1} & \cdots & 1 \end{bmatrix} \quad , \quad \mathbf{U} = \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & u_{nn} \end{bmatrix} \tag{12}$$

Lösen linearer Gleichungen - Direkte Verfahren

Gauss-Verfahren (Eliminierungsverfahren)

- 1

$$a_{11}u_1 + a_{12}u_2 = b_1 \tag{13}$$

$$a_{21}u_1 + a_{22}u_2 = b_2 \tag{14}$$

- 2

$$a_{21}\frac{a_{11}}{a_{21}}u_1 + a_{22}\frac{a_{11}}{a_{21}}u_2 = \frac{a_{11}}{a_{21}}b_2 \tag{15}$$

- 3: (19) - (17)

$$\left(\frac{a_{22}a_{11}}{a_{21}} - a_{12}\right)u_2 = \frac{a_{11}}{a_{21}}b_2 - b_1 \tag{16}$$

- 4

$$u_2 = \frac{\frac{a_{11}}{a_{21}}b_2 - b_1}{\frac{a_{22}a_{11}}{a_{21}} - a_{12}} \tag{17}$$

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.