Hydroinformatik II:

Grundlagen Numerik

V8 [BHYWI-08-05]

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

[2]Technische Universität Dresden – TUD, Dresden

Leipzig, 25. Mai 2017

Vorlesungsplan Hydroinformatik II SoSe 2017

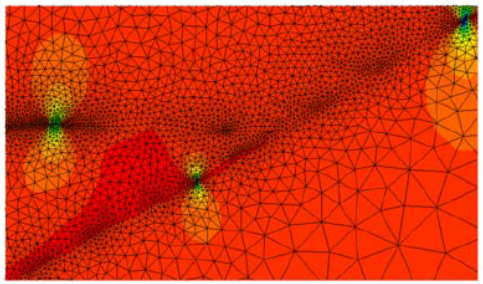
V	Datum	Thema	BHYWI
01	07.04.2017	Einführung: Veranstaltung	08-01
02	07.04.2017	Grundlagen: Kontinuumsmechanik	08-02
–	14.04.2017	Ostern	
03	21.04.2017	Grundlagen: Hydromechanik	08-03
04	28.04.2017	Übung 1: Qt Installation (Robert Schlick)	
05	05.05.2017	Grundlagen: Hydromechanik / PDE / Übung 2	08-04
06	12.05.2017	Literaturstudium: Hydromechanik	
07	19.05.2017	Übung 2: Funktionsrechner	
08	26.05.2017	Numerik: Einführung	08-05
09	02.06.2017	Numerik: (exp) Finite Differenzen Methode	08-06
–	09.06.2017	Pfingsten	
10	16.06.2017	Numerik: (imp) Finite Differenzen Methode	08-07
11	23.06.2017	Gerinnehydraulik: Theorie - Grundlagen	08-08
12	30.06.2017	Gerinnehydraulik: Übungen, Programmierung	08-09+10+11
13	07.07.2017	Grundwassermodellierung: Einführung	08-12
14	14.07.2017	Kurs-Zusammenfassung, Ausblick und Beleg	08-13

Übungsplan Hydroinformatik II SoSe 2017

V	Datum	Thema	BHYWI-E
04	28.04.2017	Übung 1: Qt Installation (Robert Schlick)	
05	05.05.2017	Übung 2: Funktionsrechner, Einführung	08-04
07	19.05.2017	Übung 2: Funktionsrechner, Anwendung	
08	26.05.2017	Numerik: Einführung	
09	02.06.2017	Numerik: (exp) Finite Differenzen Methode	08-06
10	16.06.2017	Numerik: (imp) Finite Differenzen Methode	08-07
12a	30.06.2017	Gerinnehydraulik: Q&D	08-09-E2
12b	30.06.2017	Gerinnehydraulik: OOP, Dialog	08-10
12c	30.06.2017	Gerinnehydraulik: Interaktiv	08-11
13	07.07.2017	Grundwassermodellierung: Einführung	08-12

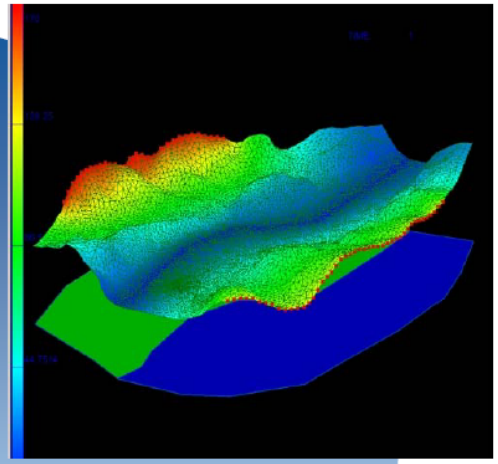
Konzept

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

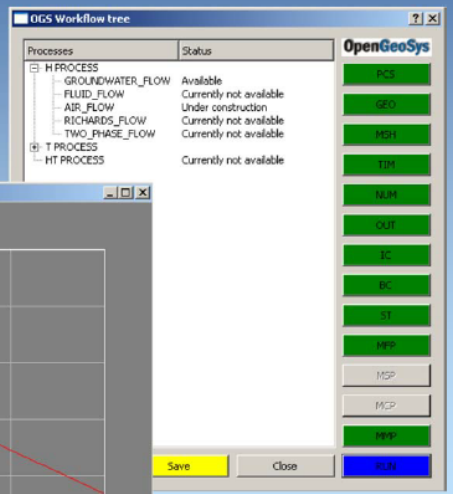


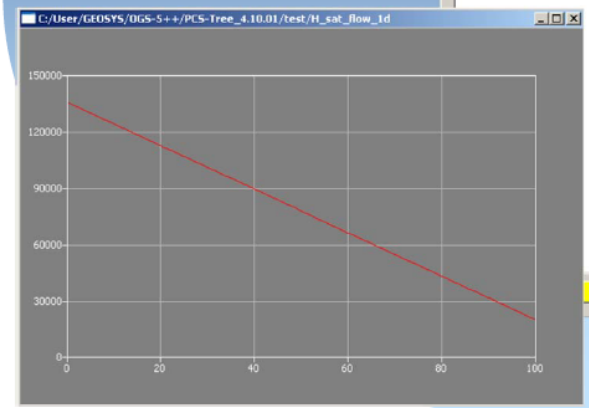
Basics
Mechanik

Anwendung



Numerische
Methoden





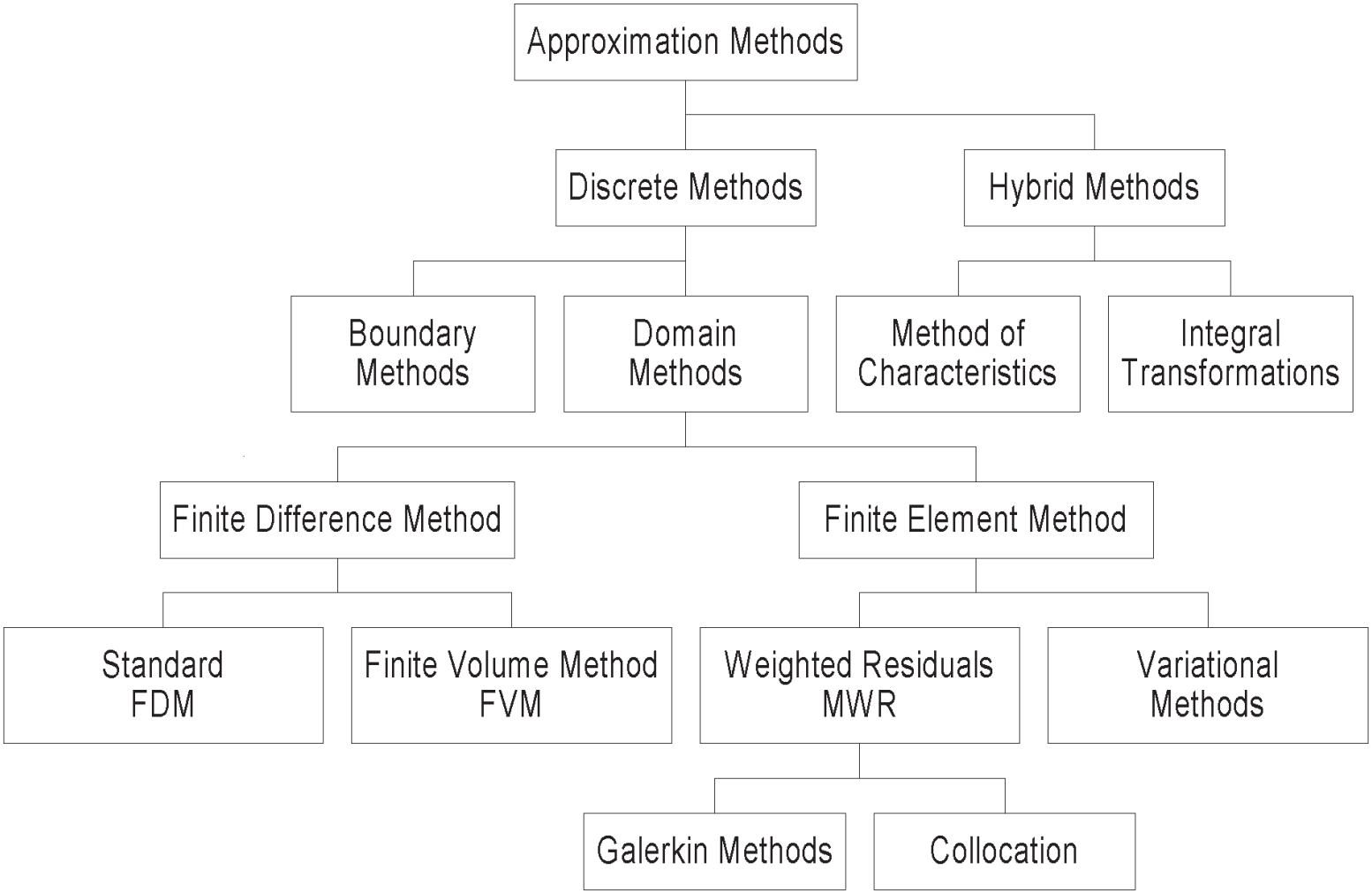
Programmierung
Visual C++

Prozessverständnis

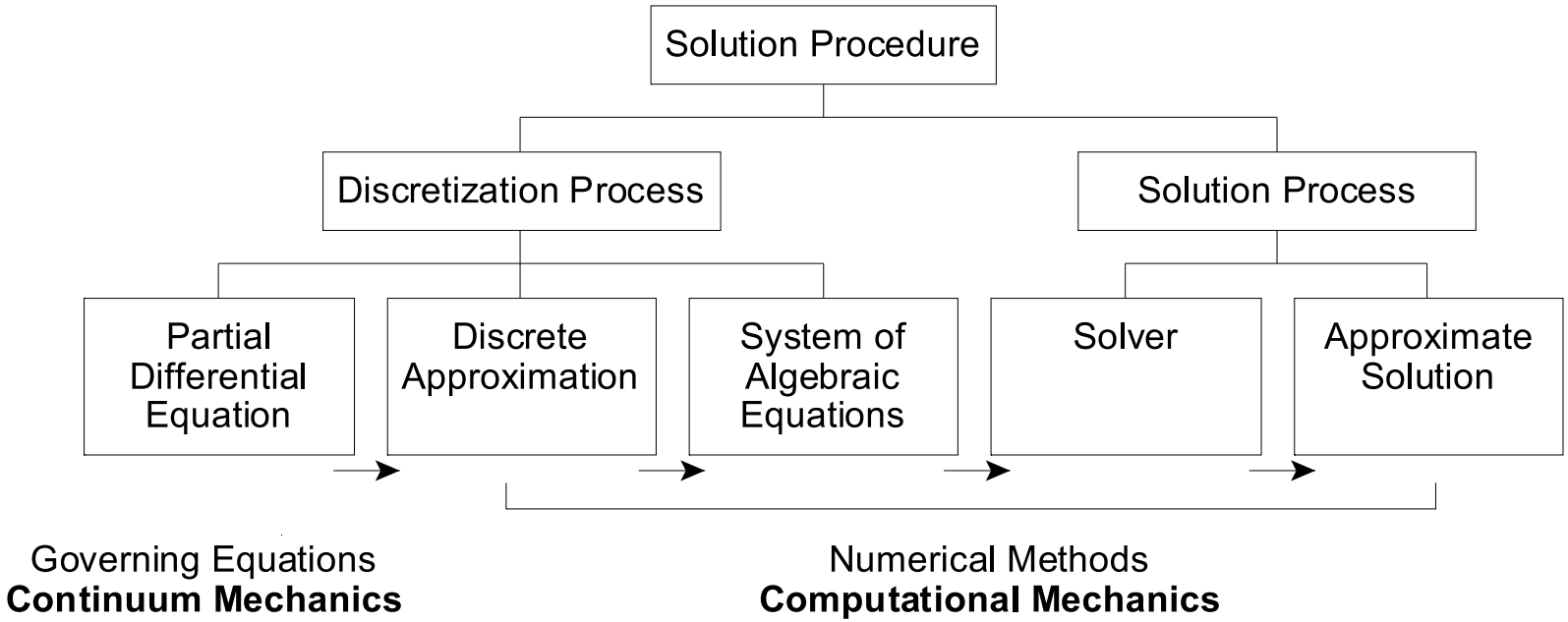
Inhalte

- Konzept
- Näherungsverfahren
- Lösungsverfahren
- Definitionen
- Fehler
- Kriterien
- Lösen von Gleichungssystemen

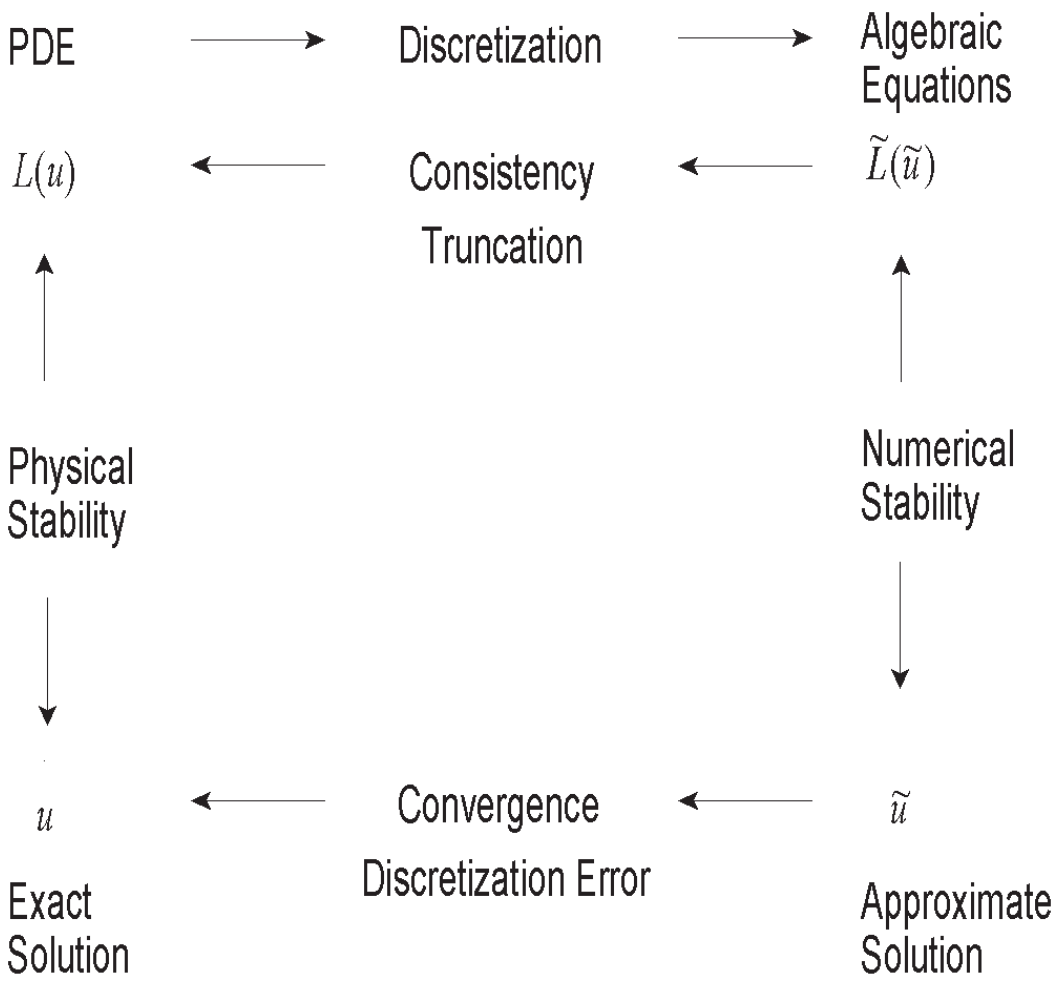
Näherungsverfahren



Lösungsverfahren



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

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

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

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

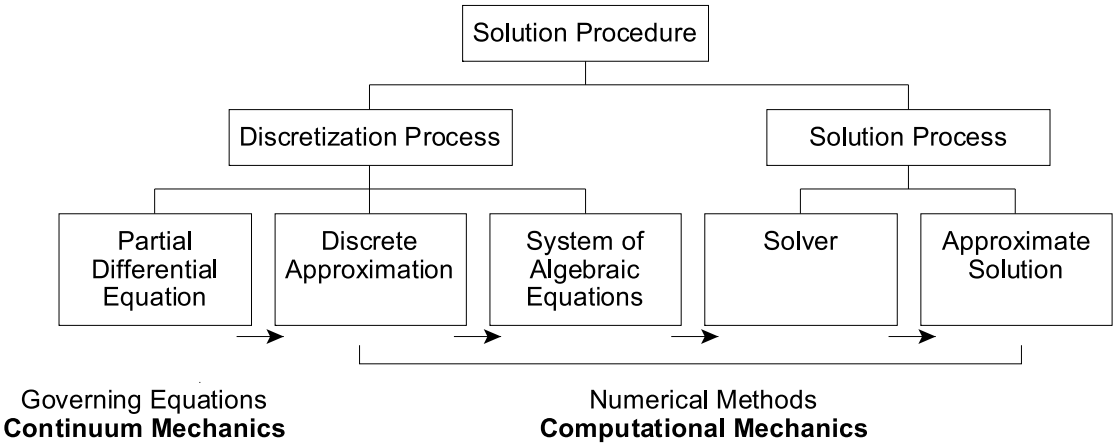
Stabilität

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

- Courant-Zahl
$$Cr = \frac{v\Delta t}{\Delta x} \leq 1 \tag{5}$$
- Peclet-Zahl
$$Pe = \frac{v\Delta x}{\alpha} \leq 2 \tag{6}$$
- Neumann-Zahl
$$Ne = \frac{\alpha\Delta t}{\Delta x^2} \leq \frac{1}{2} \tag{7}$$

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

Gleichungssysteme



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

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 $\mathbf{A}$ is not singular (i.e. $\det \mathbf{A} \neq 0$), can be composed in following way.

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

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

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

Lösen linearer Gleichungen - Direkte Verfahren

If $\mathbf{P} = \mathbf{I}$ the matrix $\mathbf{A}$ has a so-called LU-decomposition: $\mathbf{A} = \mathbf{L}\,\mathbf{U}$. The task reduces then to calculate the lower and upper matrices and invert them. Once $\mathbf{L}$ and $\mathbf{U}$ are determined, the inversion of $\mathbf{A}$ is trivial due to the triangle structure of $\mathbf{L}$ and $\mathbf{U}$.
Assuming that beside non-singularity and existing LU-decomposition, $\mathbf{A}$ is symmetrical additionally, we have $\mathbf{U} = \mathbf{D}\,\mathbf{L}^T$ with $\mathbf{D} = diag(d_i)$. Now we can conduct the following transformations.

$$\mathbf{A} = \mathbf{L}\,\mathbf{U} = \mathbf{L}\,\mathbf{D}\,\mathbf{L}^T = \underbrace{\mathbf{L}\sqrt{\mathbf{D}}}_{\tilde{\mathbf{L}}}\underbrace{\sqrt{\mathbf{D}}\,\mathbf{L}^T}_{\tilde{\mathbf{L}}^T} \tag{13}$$

The splitting of $\mathbf{D}$ requires that $\mathbf{A}$ is positive definite thus that $\forall d_i > 0$. The expression

$$\mathbf{A} = \tilde{\mathbf{L}}\,\tilde{\mathbf{L}}^T \tag{14}$$

is denoted as Cholesky decomposition. Therefore, the lower triangle matrices of both the Cholesky and the Gaussian method are connected via

$$\tilde{\mathbf{L}} = \mathbf{L}^T\sqrt{\mathbf{D}} \tag{15}$$

Lösen linearer Gleichungen - Direkte Verfahren

- Gauss-Verfahren (Eliminierungsverfahren)

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.

Lösen linearer Gleichungen - Iterative Verfahren

- Gauss-Seidel Verfahren (Hydrosystemanalyse)

Lösen nichtlinearer Gleichungen

In this section we present a description of selected iterative methods that are commonly applied to solve non-linear problems.

- Picard method (fixpoint iteration)
- Newton methods
- Cord slope method
- Dynamic relaxation method

All methods call for an initial guess of the solution to start but each algorithm uses a different scheme to produce a new (and hopefully closer) estimate to the exact solution. The general idea is to construct a sequence of linear sub-problems which can be solved with ordinary linear solver

Lösen nichtlinearer Gleichungen

The general algorithm of the Picard method can be described as follows. We consider a non-linear equation written in the form

$$\mathbf{A}(\mathbf{x})\,\mathbf{x} - \mathbf{b}(\mathbf{x}) = 0 \tag{16}$$

We start the iteration by assuming an initial guess $\mathbf{x}_0$ and we use this to evaluate the system matrix $\mathbf{A}(\mathbf{x}_0)$ as well as the right-hand-side vector $\mathbf{b}(\mathbf{x}_0)$. Thus this equation becomes linear and it can be solved for the next set of $\mathbf{x}$ values.

$$\begin{aligned} \mathbf{A}(\mathbf{x}_{k-1})\,\mathbf{x}_k - \mathbf{b}(\mathbf{x}_{k-1}) &= 0 \\ \mathbf{x}_k &= \mathbf{A}^{-1}(\mathbf{x}_{k-1})\,\mathbf{b}(\mathbf{x}_{k-1}) \end{aligned} \tag{17}$$

Lösen nichtlinearer Gleichungen

Repeating this procedure we obtain a sequence of successive solutions for $\mathbf{x}_k$. During each iteration loop the system matrix and the right-hand-side vector must be updated with the previous solution. The iteration is performed until satisfactory convergence is achieved. A typical criterion is e.g.

$$\varepsilon \geq \frac{\| \mathbf{x}_k - \mathbf{x}_{k-1} \|}{\| \mathbf{x}_k \|} \tag{18}$$

where ε is a user-defined tolerance criterion. For the simple case of a non-linear equation $\mathbf{x} = \mathbf{b}(\mathbf{x})$ (i.e. $\mathbf{A} = \mathbf{I}$), the iteration procedure is graphically illustrated in Fig. 1. To achieve convergence of the scheme it has to be guaranteed that the iteration error

Lösen nichtlinearer Gleichungen

$$e_k = \| \mathbf{x}_k - \mathbf{x} \| < C \| \mathbf{x}_{k-1} - \mathbf{x} \|^p = e_{k-1} \tag{19}$$

or, alternatively, the distance between successive solutions will reduce

$$\| \mathbf{x}_{k+1} - \mathbf{x}_k \| < \| \mathbf{x}_k - \mathbf{x}_{k-1} \|^p \tag{20}$$

where p denotes the convergence order of the iteration scheme. It can be shown that the iteration error of the Picard method decreases linearly with the error at the previous iteration step. Therefore, the Picard method is a first-order convergence scheme.

Lösen nichtlinearer Gleichungen

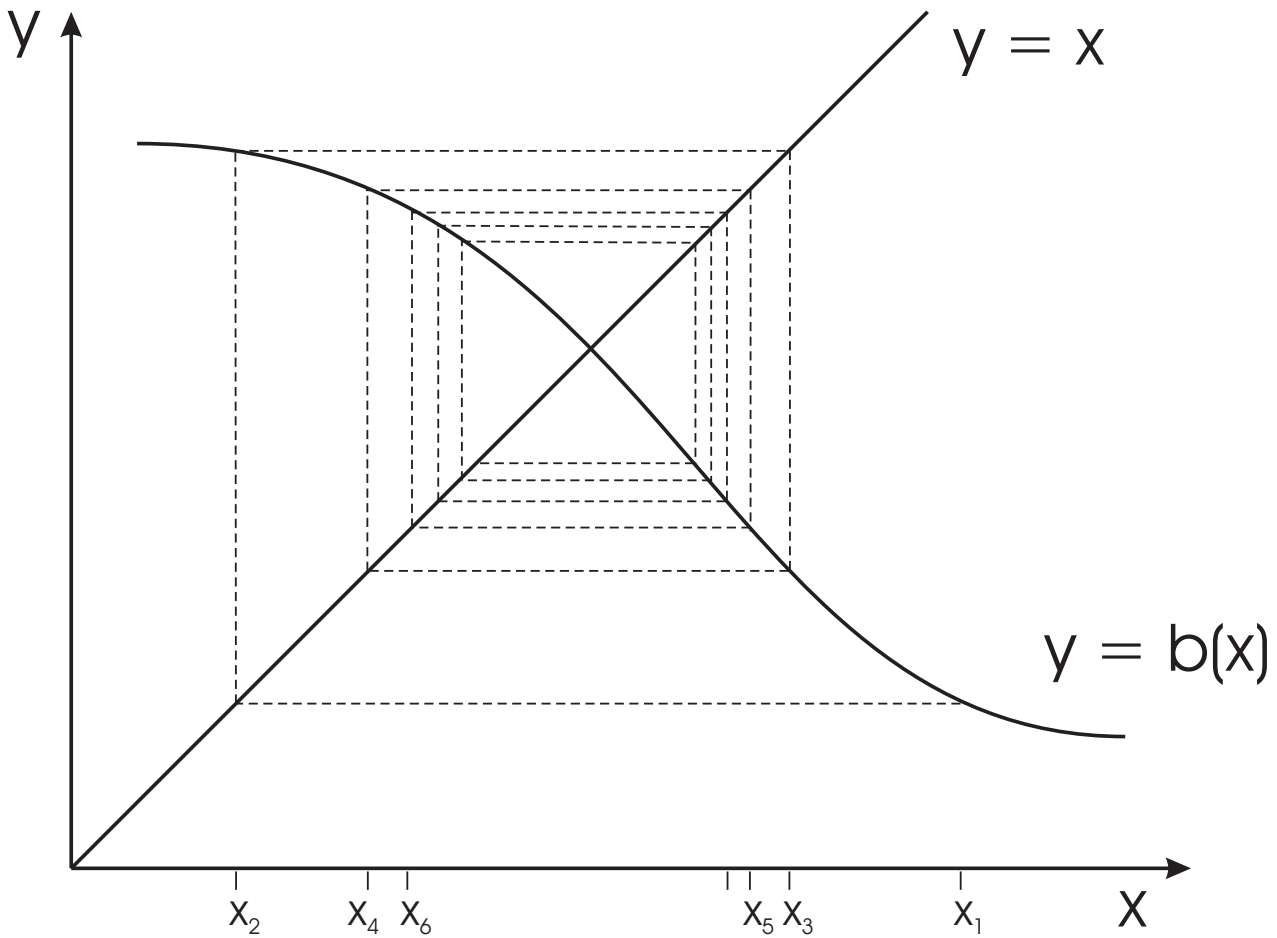


Abbildung: Graphical illustration of the Picard iteration method

Lösen nichtlinearer Gleichungen - Newton-Verfahren

In order to improve the convergence order of non-linear iteration methods, i.e. derive higher-order schemes, the Newton-Raphson method can be employed. To describe this approach, we consider once again the non-linear equation

$$\mathbf{R}(\mathbf{x}) = \mathbf{A}(\mathbf{x})\,\mathbf{x} - \mathbf{b}(\mathbf{x}) = 0 \tag{21}$$

Assuming that the residuum $\mathbf{R}(\mathbf{x})$ is a continuous function, we can develop a Taylor series expansion about any known approximate solution $\mathbf{x}_k$.

$$\mathbf{R}_{k+1} = \mathbf{R}_k + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{x}}\right]_k \Delta\mathbf{x}_{k+1} + 0(\Delta\mathbf{x}_{k+1}^2) \tag{22}$$

Lösen nichtlinearer Gleichungen - Newton-Verfahren

Second- and higher-order terms are truncated in the following. The term $\partial \mathbf{R}/\partial \mathbf{x}$ represents tangential slopes of $\mathbf{R}$ with respect to the solution vector and it is denoted as the Jacobian matrix $\mathbf{J}$. As a first approximation we can assume $\mathbf{R}_{k+1} = 0$. Then the solution increment can be immediately calculated from the remaining terms in equation (22).

$$\Delta \mathbf{x}_{k+1} = -\mathbf{J}_k^{-1}\, \mathbf{R}_k \tag{23}$$

where we have to cope with the inverse of the Jacobian. The iterative approximation of the solution vector can be computed now from the increment.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta \mathbf{x}_{k+1} \tag{24}$$

Lösen nichtlinearer Gleichungen - Newton-Verfahren

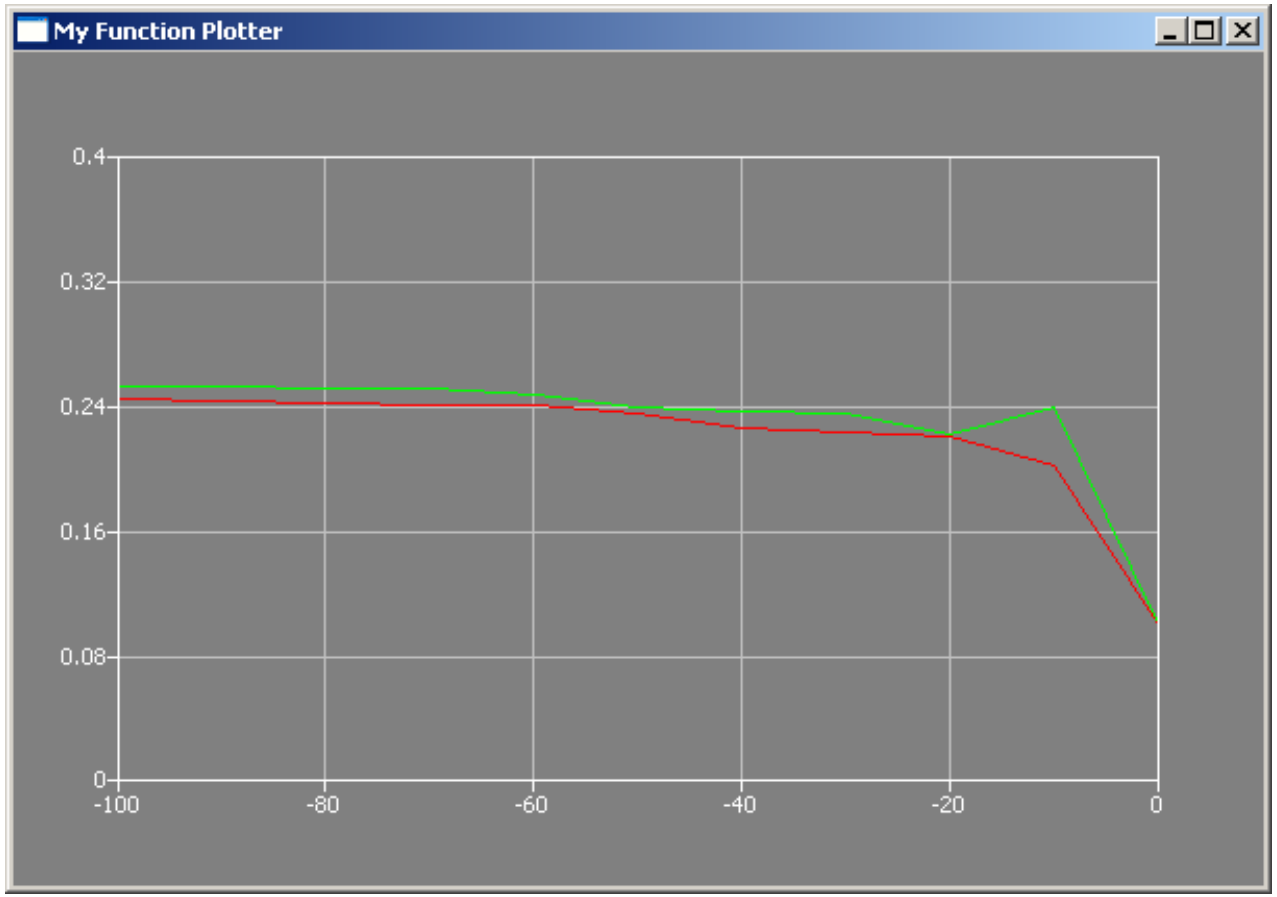


Abbildung: Graphical illustration of the Newton-Raphson iteration method

Lösen nichtlinearer Gleichungen - Newton-Verfahren

Once an initial guess is provided, successive solutions of $\mathbf{x}_{k+1}$ can be determined using equations (23) and (24) (Fig. 2). The Jacobian has to re-evaluated and inversed at every iteration step, which is a very time-consuming procedure in fact. At the expense of slower convergence, the initial Jacobian $\mathbf{J}_0$ may be kept and used in the subsequent iterations. Alternatively, the Jacobian can be updated in certain iteration intervals. This procedure is denoted as modified or 'initial slope' Newton method (Fig. 3).
The convergence velocity of the Newton-Raphson method is second-order. It is characterized by the expression.

$$\| \mathbf{x}_{k+1} - \mathbf{x} \| \leq C \| \mathbf{x}_k - \mathbf{x} \|^2 \tag{25}$$

Lösen nichtlinearer Gleichungen - Newton-Verfahren

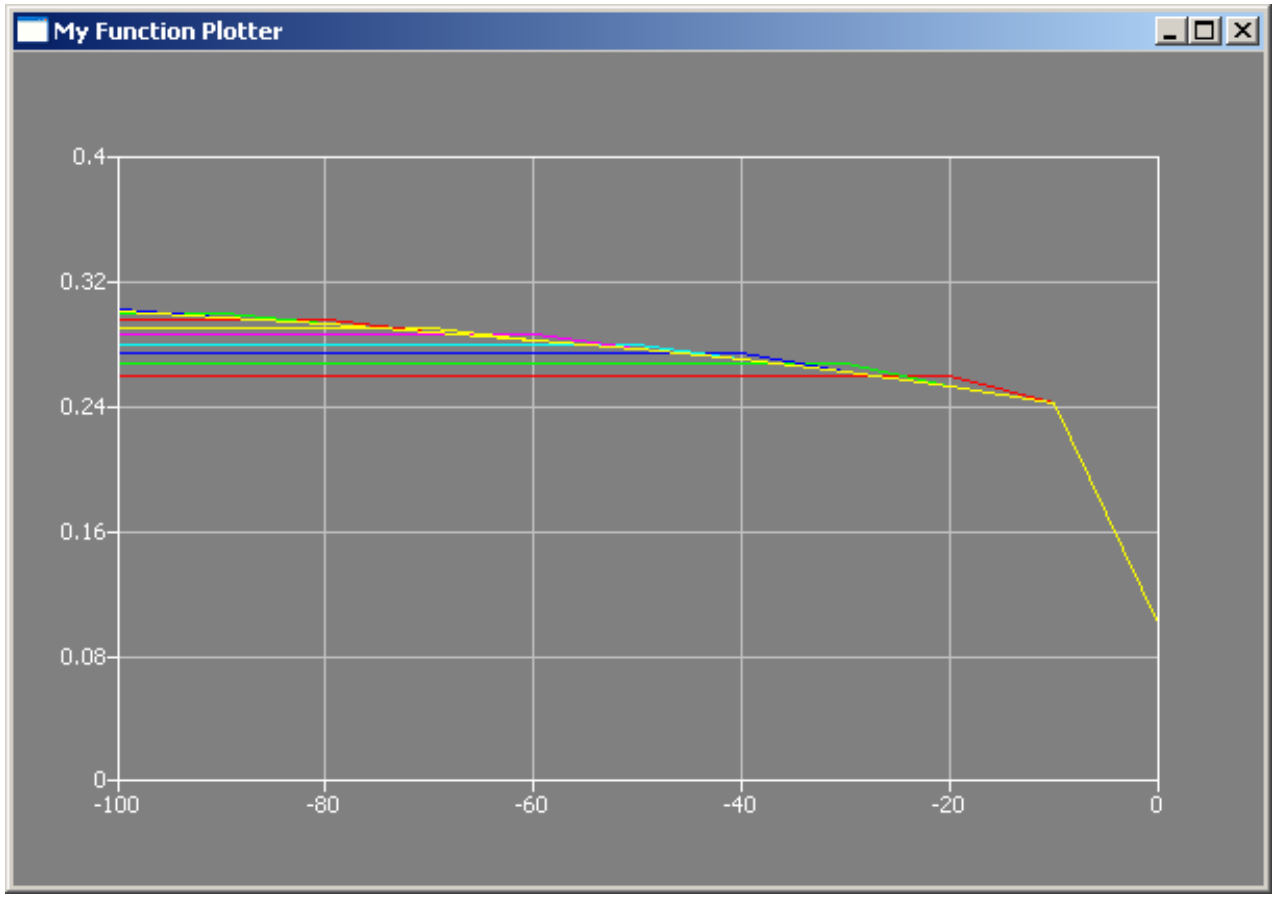


Abbildung: Graphical illustration of the modified Newton-Raphson iteration method