Hydroinformatik II - SoSe 2024

HyBHW-S2-01-V10: Finite-Differenzen-Methode I

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, 02.07.2024

TECHNISCHE UNIVERSITÄT DRESDEN

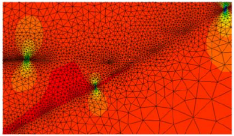
Zeitplan: Hydroinformatik II - SoSe 2024

Datum	HI	II	Thema	Typ
14.06.2024	14	2-01	Einführung in die Lehrveranstaltung - Teil 2	L
14.06.2024	15	2-02	Werkzeuge | Tools	L
14.06.2024	16	2-03	Grundlagen: Kontinuumsmechanik	L
21.06.2024	17	2-04	Grundlagen: Hydromechanik	L
21.06.2024	18	2-05	Grundlagen: Partielle Partialgleichungen	L
21.06.2024	19	2-06	Übung: Analytische Lösungen	E
28.06.2024*	20	2-07	Grundlagen: Näherungsverfahren	L
28.06.2024*	21	2-08	Übung: Jupyter Diffusionsprozess	E
02.07.2024*	22	2-09	Numerik: Finite-Differenzen-Methode (explizit)	L
02.07.2024*	23	2-10	Numerik: Finite-Differenzen-Methode (implizit)	L
12.07.2024	24	2-11	Übung: Finite-Differenzen-Methoden	E
12.07.2024	25	2-12	Grundlagen: Gerinnehydraulik	L
12.07.2024	26	2-13	Übung: Gerinnehydraulik	E
19.07.2024	27	2-14	Ausblick: Grundwassermodellierung	E
19.07.2024	28	2-15	Klausur/Beleg: Besprechung zur Vorbereitung	L

*online Vorlesung

Konzept

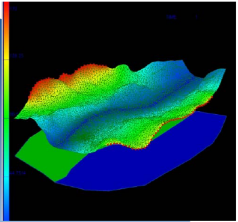
$$\frac{d\psi}{dt} = \frac{\partial\psi}{\partial t} + \mathbf{v}^E \nabla\psi$$



Basics

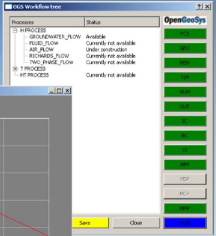
Mechanik

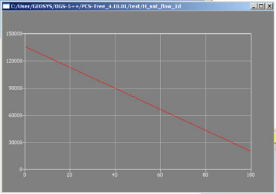
Anwendung



Numerische

Methoden





Programmierung

Visual C++

Prozessverständnis

TECHNISCHE UNIVERSITÄT DRESDEN

Fahrplan

Übung

- Jupyter installation: Probleme? ($\Rightarrow$ 5)

Vorlesung

- Grundlagen der Finite Differenzen Methode
- Approximation methods
- Finite difference method – FDM (Ch. 3)
- Taylor series expansion
- Derivatives
- Diffusion equation
- (Finite element method – FEM $\Rightarrow$ Hydrosystemanalyse)

TECHNISCHE UNIVERSITÄT DRESDEN

Jupyter

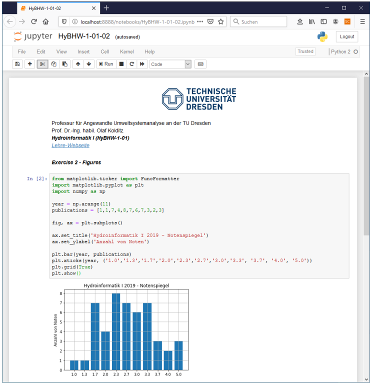
- Jupyter Notebook
- Jupyter Lab
- Browser-basiert

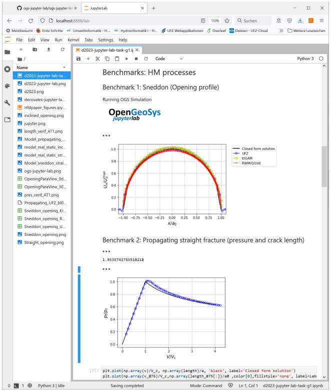
TECHNISCHE UNIVERSITÄT DRESDEN

Jupyter

- "The Jupyter Notebook · The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, ..."
- Webseite: `https://jupyter.org/`
- Vorteil: funktioniert auf allen Rechnern
- ... ein Teil unserer (neuen) Übungen machen wir mit Jupyter Notebooks ($>>$ Demo)

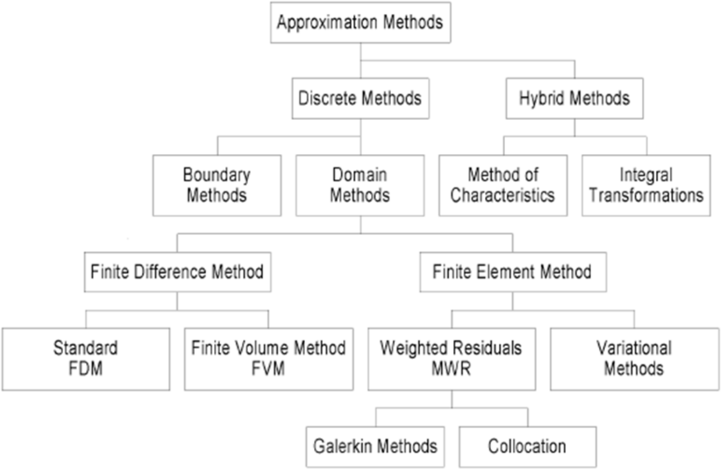
Jupyter: Example





TECHNISCHE UNIVERSITÄT DRESDEN

Näherungsverfahren



FDM Anwendungen - MODFLOW

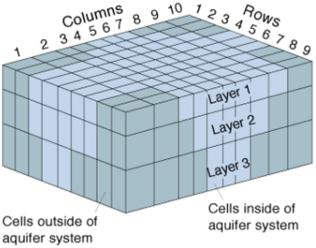


Figure 2. Example of model grid for simulating three-dimensional ground-water flow.

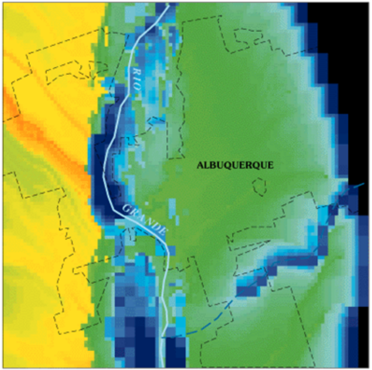
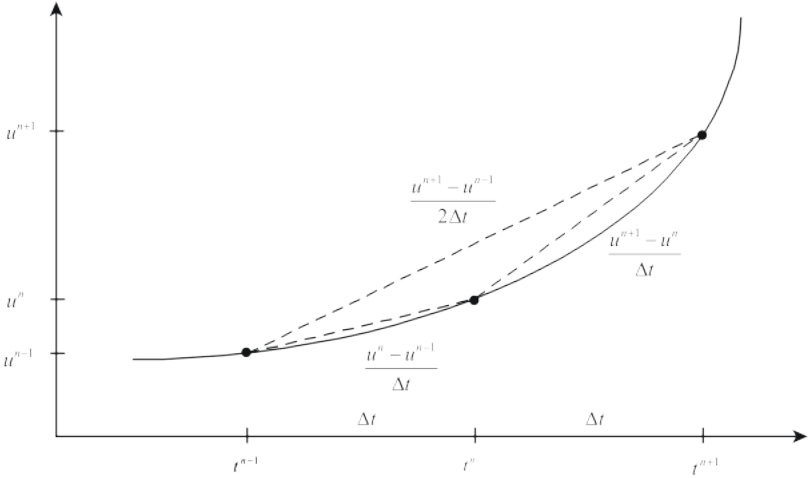


Figure 7. Application of particle tracking to estimate ground-water travel time.

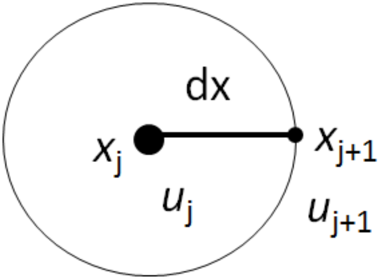
http://water.usgs.gov/pubs/FS/FS-121-97/images/fig7.gif

TECHNISCHE UNIVERSITÄT DRESDEN

Ableitungen



Taylor-Reihe



in time

$$u_j^{n+1} = \sum_{m=0}^{\infty} \frac{\Delta t^m}{m!} \left[\frac{\partial^m u}{\partial t^m} \right]_j^n \qquad (1)$$

$$\Delta t = t^{n+1} - t^n$$

in space

$$u_{j+1}^{n} = \sum_{m=0}^{\infty} \frac{\Delta x^m}{m!} \left[\frac{\partial^m u}{\partial x^m} \right]_j^n \qquad (2)$$

$$\Delta x = x_{j+1} - x_j$$

TECHNISCHE UNIVERSITÄT DRESDEN

Trunkation

$$u_j^{n+1} = u_j^n + \Delta t \left[\frac{\partial u}{\partial t}\right]_j^n + \frac{\Delta t^2}{2}\left[\frac{\partial^2 u}{\partial t^2}\right]_j^n + 0(\Delta t^3) \qquad (3)$$

$$u_{j+1}^{n} = u_j^n + \Delta x \left[\frac{\partial u}{\partial x}\right]_j^n + \frac{\Delta x^2}{2}\left[\frac{\partial^2 u}{\partial x^2}\right]_j^n + 0(\Delta x^3) \qquad (4)$$

siehe auch Trunkationsfehler (letzte Vorlesung)

1. Ableitung

$$\left[\frac{\partial u}{\partial t}\right]_j^n = \frac{u_j^{n+1} - u_j^n}{\Delta t} - \frac{\Delta t}{2}\left[\frac{\partial^2 u}{\partial t^2}\right]_j^n + 0(\Delta t^2) \tag{5}$$

$$\left[\frac{\partial u}{\partial x}\right]_j^n = \frac{u_{j+1}^{n} - u_j^n}{\Delta x} - \frac{\Delta x}{2}\left[\frac{\partial^2 u}{\partial x^2}\right]_j^n + 0(\Delta x^2) \tag{6}$$

Differenzen-Schemata

Forward difference approximation

$$\left[\frac{\partial u}{\partial x}\right]_j^n = \frac{u_{j+1}^n - u_j^n}{\Delta x} + 0(\Delta x) \tag{7}$$

Backward difference approximation

$$\left[\frac{\partial u}{\partial x}\right]_j^n = \frac{u_j^n - u_{j-1}^n}{\Delta x} + 0(\Delta x) \tag{8}$$

Central difference approximation

$$\left[\frac{\partial u}{\partial x}\right]_j^n = \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + 0(\Delta x^2) \tag{9}$$

TECHNISCHE UNIVERSITÄT DRESDEN

Zentrale Differenzen

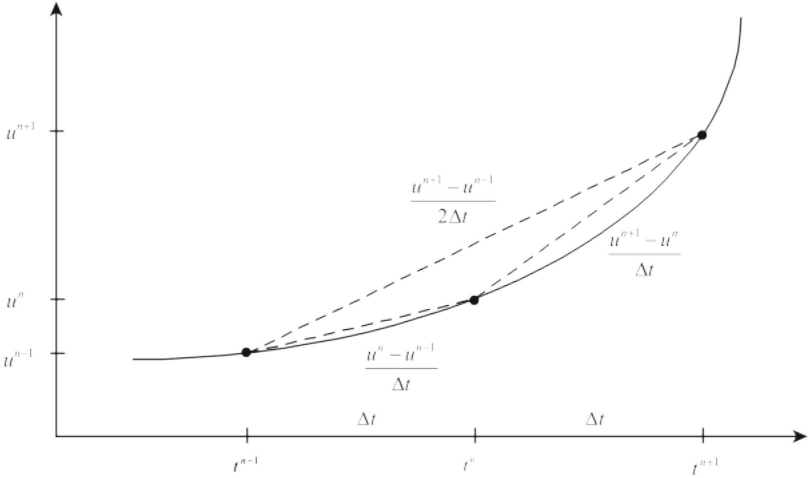
$$
\begin{aligned}
u_{j+1}^n &= u_j^n + \Delta x \left[\frac{\partial u}{\partial x}\right]_j^n + \frac{\Delta x^2}{2}\left[\frac{\partial^2 u}{\partial x^2}\right]_j^n + 0(\Delta x^3) \\
u_{j-1}^n &= u_j^n - \Delta x \left[\frac{\partial u}{\partial x}\right]_j^n + \frac{\Delta x^2}{2}\left[\frac{\partial^2 u}{\partial x^2}\right]_j^n - 0(\Delta x^3)
\end{aligned}
\tag{10}
$$

Central difference approximation

$$
\left[\frac{\partial u}{\partial x}\right]_j^n = \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + 0(\Delta x^2) \tag{11}
$$

TECHNISCHE UNIVERSITÄT DRESDEN

Ableitungen



TECHNISCHE UNIVERSITÄT DRESDEN

2. Ableitung

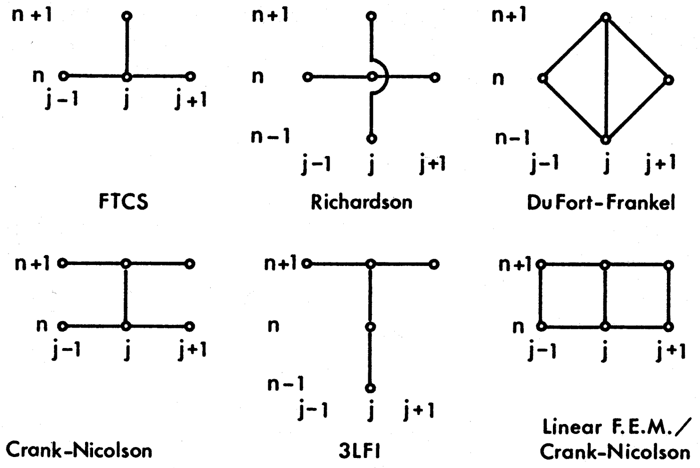
$$\left[\frac{\partial^2 u}{\partial x^2}\right]_j^n \approx \frac{1}{\Delta x}\left(\left[\frac{\partial u}{\partial x}\right]_{j+1}^n - \left[\frac{\partial u}{\partial x}\right]_j^n\right)$$

$$\approx \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} \qquad (12)$$

$$\left[\frac{\partial^2 u}{\partial x^2}\right]_j^n = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} + \frac{\Delta x^2}{12}\left[\frac{\partial^4 u}{\partial x^4}\right]_j^n + \dots \qquad (13)$$

TECHNISCHE UNIVERSITÄT DRESDEN

Übersicht Differenzenverfahren



Diffusionsgleichung

$$\frac{\partial u}{\partial t} - \alpha \frac{\partial^2 u}{\partial x^2} = 0 \tag{14}$$

TECHNISCHE UNIVERSITÄT DRESDEN

Analytical solution for diffusion equation (Skript 5.2.2)

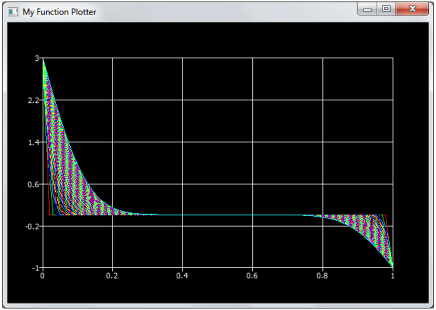
- Diffusion equation

$$\frac{\partial u}{\partial t} - \alpha \frac{\partial^2 u}{\partial x^2} = 0 \qquad (15)$$

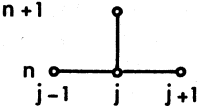
- Analytical solution

  Einschränkungen: Nur für symmetrische Randbedingungen

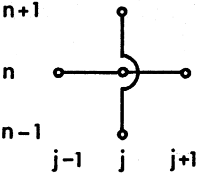
- K: validity



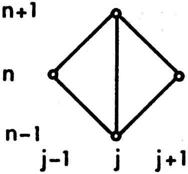
$\Rightarrow$ Übung



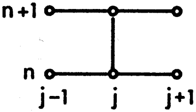
FTCS



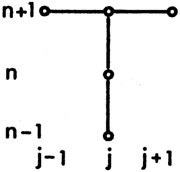
Richardson



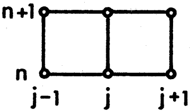
DuFort-Frankel



Crank-Nicolson



3LFI



Linear F.E.M./
Crank-Nicolson

Explizite FDM - FTCS Verfahren (Skript 3.2.2/4.1)

- PDE for diffusion processes

$$\frac{\partial u}{\partial t} - \alpha \frac{\partial^2 u}{\partial x^2} = 0 \tag{16}$$

- forward time / centered space

$$\left[\frac{\partial u}{\partial t}\right]_j^n \approx \frac{u_j^{n+1} - u_j^n}{\Delta t} \quad \left[\frac{\partial^2 u}{\partial x^2}\right]_j^n \approx \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{\Delta x^2} \tag{17}$$

- substitute

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} - \alpha \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{\Delta x^2} = 0 \tag{18}$$

- FTCS scheme for diffusion equations

$$u_j^{n+1} = u_j^n + \frac{\alpha \Delta t}{\Delta x^2}(u_{j-1}^n - 2u_j^n + u_{j+1}^n)\,,\ Ne = \frac{\alpha \Delta t}{\Delta x^2} \tag{19}$$

TECHNISCHE UNIVERSITÄT DRESDEN

Eigenschaften numerischer Verfahren

Analysis of approximation schemes consists of three steps:

- Develop the **algebraic scheme**,
- Check **consistency** of the algebraic approximate equation,
- Investigate **stability** behavior of the scheme.

Eigenschaften numerischer Verfahren

Analysis of approximation schemes consists of three steps:

- Develop the **algebraic scheme**,

$$u_j^{n+1} = u_j^n + \frac{\alpha \Delta t}{\Delta x^2}(u_{j-1}^n - 2u_j^n + u_{j+1}^n) \tag{20}$$

- Check **consistency** of the algebraic approximate equation,

$$\lim_{\Delta t, \Delta x \to 0} |\hat{L}(u_j^n) - L(u[t_n, x_j])| = 0 \tag{21}$$

- Investigate **stability** behavior of the scheme.

$$Ne = \frac{\alpha \Delta t}{\Delta x^2} \leqslant 1/2 \tag{22}$$

TECHNISCHE UNIVERSITÄT DRESDEN

Lösung des FTCS Schemas

Algebraische Schema

$$u_j^{n+1} = u_j^n + \frac{\alpha \Delta t}{\Delta x^2}(u_{j-1}^n - 2u_j^n + u_{j+1}^n) \tag{23}$$

Resultierendes Gleichungssystem

$$\mathbf{u}^{n+1} = \mathbf{A}\mathbf{u}^n \quad , \quad n = 0, 1, 2, \ldots \tag{24}$$

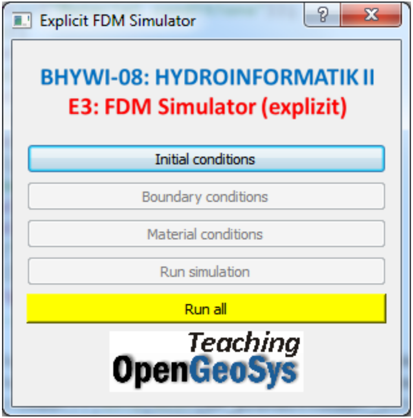
$$\mathbf{A} = \begin{bmatrix} 1 - 2\frac{\alpha \Delta t}{\Delta x^2} & \frac{\alpha \Delta t}{\Delta x^2} & & & \\ \frac{\alpha \Delta t}{\Delta x^2} & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \frac{\alpha \Delta t}{\Delta x^2} \\ & & & \frac{\alpha \Delta t}{\Delta x^2} & 1 - 2\frac{\alpha \Delta t}{\Delta x^2} \end{bmatrix}, \mathbf{u}^n = \begin{bmatrix} u_2^n \\ u_3^n \\ \ldots \\ u_{np-2}^n \\ u_{np-1}^n \end{bmatrix}$$

TECHNISCHE UNIVERSITÄT DRESDEN

Übungen

- (Qt Übung (alt): BHYWI-08-03-E)
- C++ Übung: EX08-fdm-explicit-C++
- Python Übung: EX08-fdm-explicit-python

TECHNISCHE UNIVERSITÄT DRESDEN

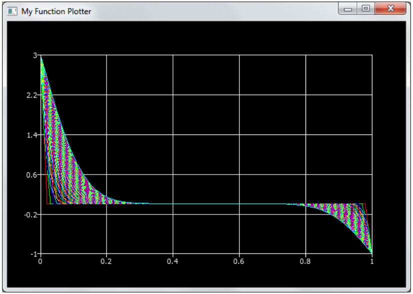


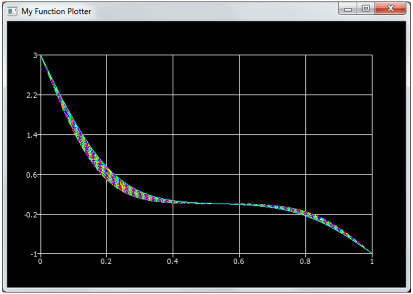
Dialog-Klasse: Konstruktor
`Dialog::Dialog`

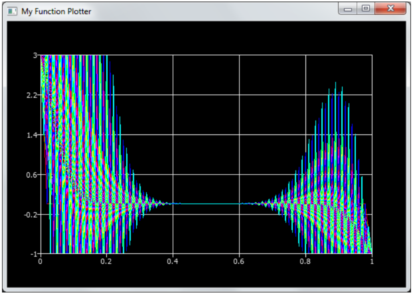
1. Elemente
2. Connects
3. Layout
4. Datenstrukturen (Speicherreservierung)

FDM Übung: Explizit Qt

Ergebnisse







$$Ne = \frac{\alpha \Delta t}{\Delta x^2} \leqslant 0.5 \qquad (25)$$

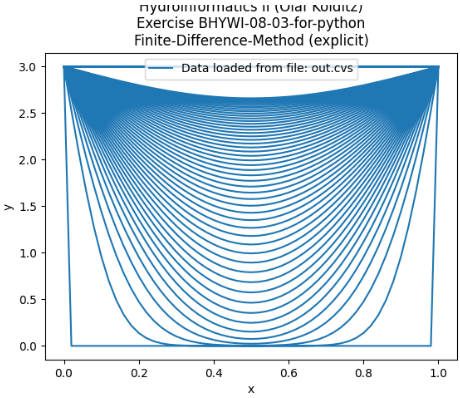
How sensitive ?

TECHNISCHE UNIVERSITÄT DRESDEN

Übung: EX08 (C++): explizite FDM

```
echo Compilation
g++ main.cpp
echo Execution
a.exe
echo Ploting
data_from_file.py
echo End
```

Listing: Skript: C++ Berechnung > Python Grafik



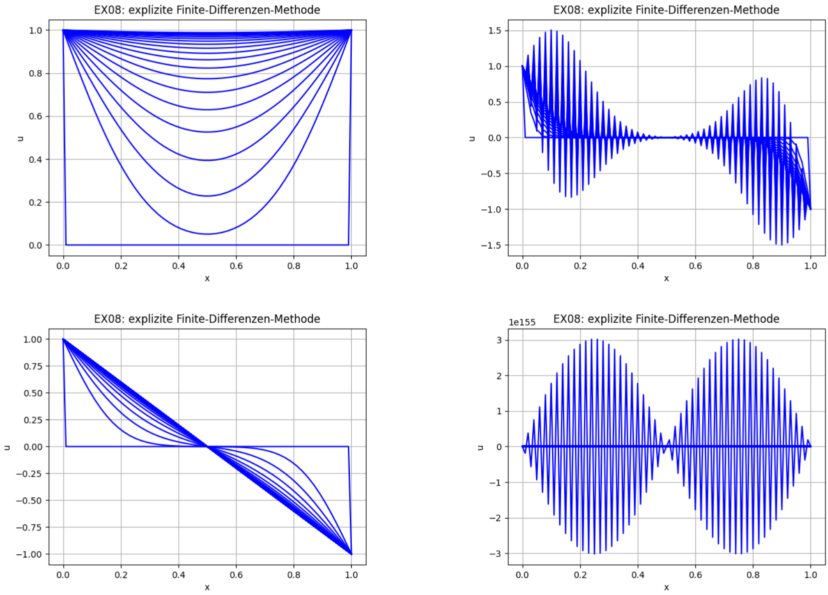
TECHNISCHE UNIVERSITÄT DRESDEN

Übung: EX08 (Python): explizite FDM

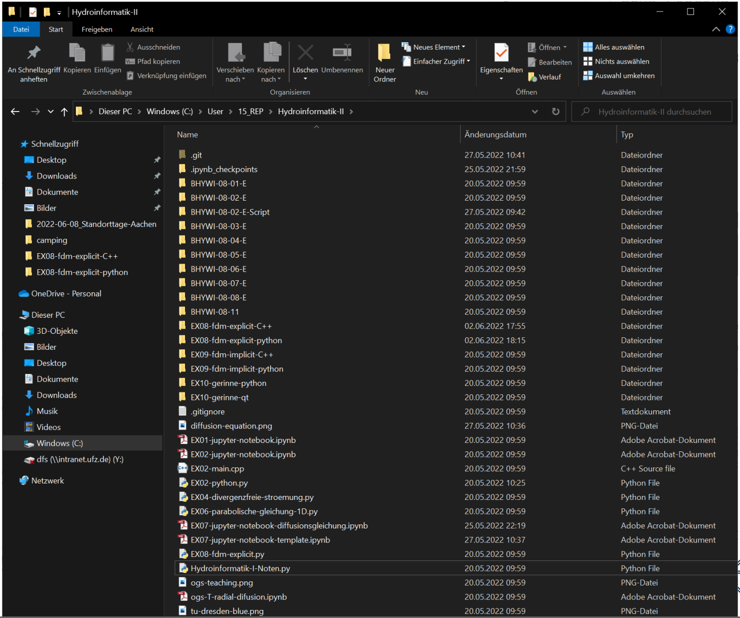
```python
import math
import numpy as np
import matplotlib.pylab as plt
#data structures
##physical parameter
alpha = 1.0
##numerical parameters (discretization)
nx = 10
x = np.zeros(nx+1)
dx = 1./nx
t = [0.01]
nt = 10 #wieviele Zeitschritte bis zum stationaeren Zustand
dt = 0.5 * dx*dx / alpha
Ne = alpha * dt / (dx*dx)
##field function
u = np.zeros(nx+1)
uo = np.zeros(nx+1)
#initial condition
u_ic = 0.
for i in range(nx+1):
  x[i] = 0
  u[i] = 0
  uo[i] = 0
#boundary conditions
u_bc_l = 1.
u_bc_r = 1.
u[0] = uo[0] = u_bc_l
u[nx] = uo[nx] = u_bc_r
#initial state
...
```

```python
...
#initial state
for i in range(0,nx+1):
    x[i] = (float(i)/float(nx))
plt.plot(x,u,color='blue')
#fdm-explicit
for n in range(1,nt):
  for i in range(1,nx):
    u[i] = uo[i] + Ne *(uo[i-1] - 2*uo[i] + uo[i+1])
  plt.plot(x,u,color='blue')
  for i in range(1,nx):
    uo[i] = u[i]
#plots
plt.title('EX08: explizite Finite-Differenzen-Methode')
plt.xlabel('x')
plt.ylabel('u')
plt.axis('tight')
plt.grid()
plt.savefig("fdm-explicit.png")
plt.show()
```

Listing: ...


EX08: explizite Finite-Differenzen-Methode
u
x
EX08: explizite Finite-Differenzen-Methode
u
x
EX08: explizite Finite-Differenzen-Methode
u
x
1e155
EX08: explizite Finite-Differenzen-Methode
u
x

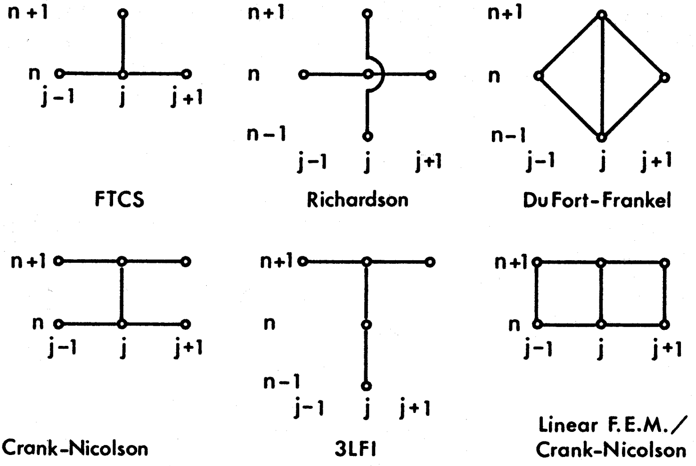
Hydroinformatik II Übungen - Repository



TECHNISCHE UNIVERSITÄT DRESDEN

Alternative Verfahren

Explizite und implizite Differenzenverfahren



TECHNISCHE UNIVERSITÄT DRESDEN

Algebraische Schema:

$$\left[\frac{\partial^2 u}{\partial x^2}\right]_j^{n+1} \approx \frac{u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}}{\Delta x^2} \tag{26}$$

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} - \alpha \frac{u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}}{\Delta x^2} = 0 \tag{27}$$

$$\frac{\alpha \Delta t}{\Delta x^2} \left(-u_{j-1}^{n+1} + 2u_j^{n+1} - u_{j+1}^{n+1}\right) + u_j^{n+1} = u_j^n \tag{28}$$

TECHNISCHE UNIVERSITÄT DRESDEN

Farbcode

Vorlesung

Übung

Hausaufgabe

Beleg

Beleg

Beleg

besteht aus 3 Teilen:

- Hausaufgaben

2 Berichte:

- Beschreiben Sie die Wassersituation in einem Gebiet ihrer Wahl, Literatur-Recherche (ca. 5 Seiten)
- Warum ist die Modellierung ein wichtiges Werkzeug für die Bewertung und das Management von Wasserressourcen ? (ca. 5 Seiten) (siehe auch OpenGeoSys-Project on YouTube)

⇒ `https://www.youtube.com/user/OpenGeoSys`

TECHNISCHE UNIVERSITÄT DRESDEN

Hausaufgaben

Hausaufgaben: Hydroinformatik

Hydroinformatik II - HyBHW-1-02

1. Skalarprodukt: Schreiben sie das Skalarprodukt $\nabla \cdot \mathbf{v}$ in Komponentenschreibweise.
2. Mechanik: Was ist $\mathbf{v} \cdot \nabla\psi$?
3. Mechanik: Was ist Φ^{ψ}?
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 partiellen Differentialgleichungen: (17), (19), (20), (21), (23) (siehe Vorlesung 5).
7. Analytik: Darstellung der analytischen Lösung für die 1-D parabolische Differentialgleichung (BHYWI-08-02-E-Script) (41)
8. Numerik: Darstellung der numerischen Lösung (explizite FDM) für die 1-D parabolische Differentialgleichung (EX08-fdm-explicit-python)
9. Numerik: Darstellung der numerischen Lösung (implizite FDM) für die 1-D parabolische Differentialgleichung (EX09-fdm-implicit-python)
10. Numerik: Darstellung der nichtlinearen Lösung für die Gerinnehydraulik (EX10-gerinne-python)

TECHNISCHE UNIVERSITÄT DRESDEN

Hausaufgaben: Hydroinformatik

Beispiel: Aufgaben 7-10

- zip-File mit Übungen (Hydroinformatik-Beleg-2023-Übungen.zip) herunterladen (Lehre-Webseite)
- Python-File editieren (Matrikel-Nummer oder Name)
- Programme zum Rechnen und Darstellen ausführen
- Ergebnis in den Beleg einfügen

TECHNISCHE UNIVERSITÄT DRESDEN