

Hydrosystemanalyse: Finite-Elemente-Methode (FEM)

Prof. Dr.-Ing. habil. Olaf Kolditz

¹Helmholtz Centre for Environmental Research – UFZ, Leipzig

²Technische Universität Dresden – TUD, Dresden

Dresden, 03. Juli 2015

Vorlesungsplan SoSe 2015: Hydrosystemanalyse

#	Datum	Vorlesung	Übung	Skript
1	17.04.15	Einführung, Systemanalyse		
2	17.04.15	Grundwasserhydraulik: Einzugsgebiet Bilanzierung, Vorlesung		1.1+2
3	24.04.15	Grundwasserhydraulik: Einzugsgebiet Bilanzierung, Übung		1.3
7	22.05.15	Grundwasserhydraulik: Finite-Differenzen-Verfahren, Rechteckaquifer		
11	12.06.15	Grundwasserhydraulik: Finite-Differenzen-Verfahren, Selke	BHYWI-22-E3	1.4, 1.5
13	19.06.15	Grundwasserhydraulik: Finite-Differenzen-Verfahren, OOP, VTK, iFDM	BHYWI-22-E4	1.5, 1.6
15	26.06.15	Grundwasserhydraulik: implizites Finite-Differenzen-Verfahren, Randbedingungen	BHYWI-22-E15	1.5
20	03.07.15	Grundwasserhydraulik: Finite-Elemente-Verfahren		
23	10.07.15	UFZ-Exkursion: VISLAB		
32	17.07.15	Klausurvorbereitung		

Lecture Table of Contents

- ▶ implizites Verfahren - Grundidee
- ▶ implizite FD Gleichungen
- ▶ Implementierung
- ▶ Testbeispiel

Finite-Elemente-Methode (FEM)

Wir haben uns sehr intensiv mit der Methode der finiten Differenzen beschäftigt. Bei der Einführung der numerischen Berechnungsmethoden in der Hydroinformatik II Veranstaltung haben wir gesehen, dass es ein ganzes Arsenal von Verfahren gibt (Abb. 2.1, Hydroinformatik II Skript), welche für bestimmte Problemstellungen geeignet oder ungeeignet sind. In den Visualisierungsübungen im VISLab werden wir sehen, dass FD Verfahren Grenzen haben, wenn es um die exakte Beschreibung komplexer Geometrien geht. Hier sind Verfahren im Vorteil, die sogenannte unstrukturierte Rechengitter benutzen können. Hierzu zählt z.B. die Finite Elemente Methode, mit der wir uns nun etwas näher beschäftigen möchten. Die Abb. 1 zeigt uns ein aktuelles Beispiel aus einem Forschungsvorhaben zusammen mit der Bundesanstalt für Geowissenschaften und Rohstoffe (BGR) in Hannover

Finite-Elemente-Methode (FEM)

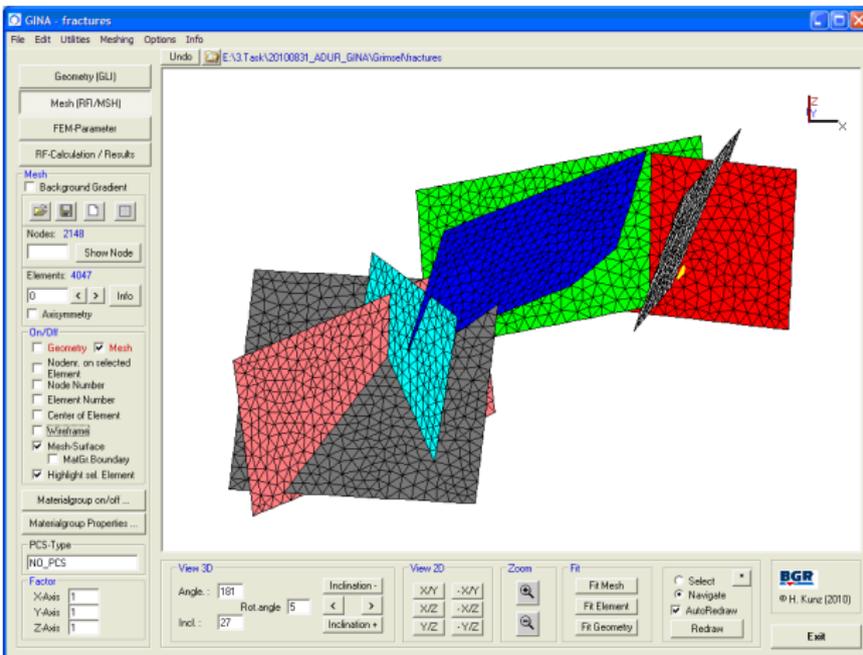


Abbildung: Modellierung eines Kluftsystems im Kristallin (Herbert Kunz, BGR)

Finite-Elemente-Methode (FEM) - Motivation

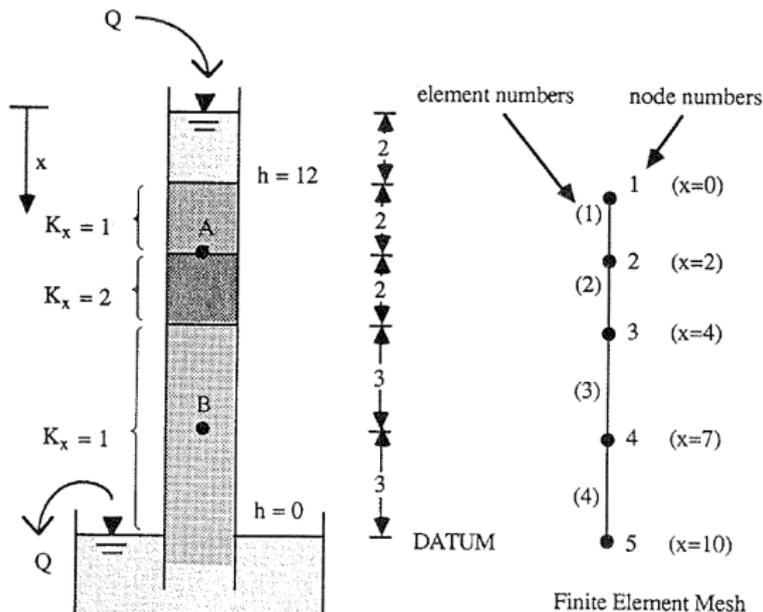


Abbildung: Bodensäulenmodell zur Erläuterung der FE Methode nach Istok (1989)

Finite-Elemente-Methode (FEM) - Konzept

Wir betrachten ein 1D stationäres Grundwasserströmungsproblem.

$$\frac{\partial}{\partial x} \left(K_x \frac{\partial h}{\partial x} \right) = 0 \quad (1)$$

Ein Näherungsverfahren wird uns eine Näherungslösung \hat{h} liefern, welche die Bilanzgleichung (1) nicht mehr ganz korrekt erfüllt.

$$\frac{\partial}{\partial x} \left(K_x \frac{\partial \hat{h}}{\partial x} \right) = R(x) \neq 0 \quad (2)$$

Finite-Elemente-Methode (FEM)

Dabei ist $R(x)$ der Fehler, das sogenannte Residuum. Das Residuum kann in den verschiedenen Gitterpunkten i, j unterschiedliche Wert $R_i \neq R_j$ annehmen. Am Knoten 3 hängen die Elemente (2) und (3) (Abb. 2). Daher ergibt sich das Residuum im Knoten 3 aus den Elementwerten wie folgt.

$$R_3 = R_3^{(2)} + R_3^{(3)} \quad (3)$$

Finite-Elemente-Methode (FEM)

Ohne Beschränkung der Allgemeinheit können wir für jeden Knoten i , das Residuum wie folgt schreiben.

$$R_i = \sum_{e=1}^p R_i^{(e)} \quad (4)$$

Dabei ist p die Anzahl der Elemente, die am Knoten i angebunden sind.

Finite-Elemente-Methode (FEM)

Der Elementbeitrag zum Residuum lässt sich wie folgt berechnen.

$$R_i^{(e)} = \int_{x_i^e}^{x_j^e} N_i^{(e)} \left(K_x^{(e)} \frac{\partial^2 \hat{h}^{(e)}}{\partial x^2} \right) dx \quad (5)$$

Dabei ist $N_i^{(e)} \equiv W_i(x)$ eine Interpolationsfunktion auf dem Element (e) (Abb. 3).

Finite-Elemente-Methode (FEM)

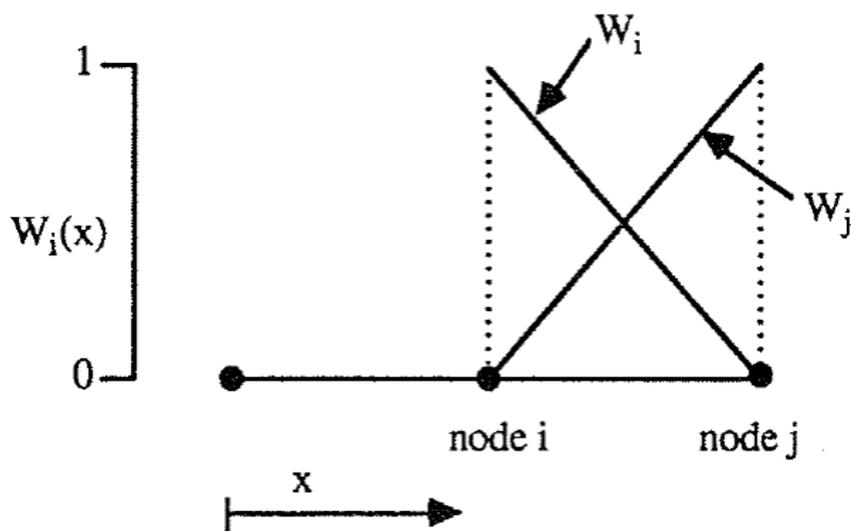


Abbildung: Interpolationsfunktion für die Galerkin-Methode nach Istok (1989)

Finite-Elemente-Methode (FEM)

Die gleiche Beziehung lässt sich den anderen Element-Knoten j schreiben.

$$R_j^{(e)} = \int_{x_i^{(e)}}^{x_j^{(e)}} N_j^{(e)} \left(K_x^{(e)} \frac{\partial^2 \hat{h}^{(e)}}{\partial x^2} \right) dx \quad (6)$$

Finite-Elemente-Methode (FEM)

Die linearen Interpolationsfunktionen für 1D Elemente sind

$$N_i^{(e)}(x) = \frac{x_j^{(e)} - x}{L^{(e)}} \quad , \quad N_j^{(e)}(x) = \frac{x - x_i^{(e)}}{L^{(e)}} \quad (7)$$

Finite-Elemente-Methode (FEM)

Die approximierte Feldgröße h kann nun wie folgt auf dem 1D finiten Element interpoliert werden (Abb. 3).

$$\begin{aligned}\hat{h}^{(e)}(x) &= N_i^{(e)} h_i + N_j^{(e)} h_j \\ &= \frac{x_j^{(e)} - x}{L^{(e)}} h_i + \frac{x - x_i^{(e)}}{L^{(e)}} h_j\end{aligned}\quad (8)$$

Finite-Elemente-Methode (FEM)

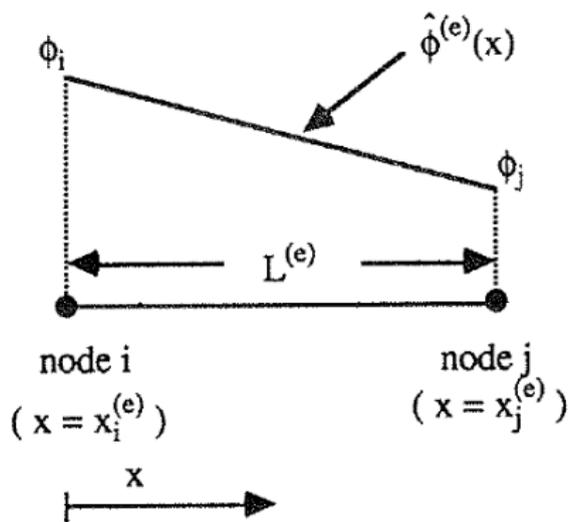


Abbildung: Interpolierte Näherungslösung auf einem 1D Element nach Istok (1989)

Finite-Elemente-Methode (FEM)

Jetzt stoßen wir auf ein ernsthaftes Problem. In den Gleichungen (5) und (6) müssten wir Ableitungen zweiter Ordnung berechnen, unsere Interpolationsfunktionen sind aber linear - also existieren keine zweiten Ableitung - was tun? Wir machen einen mathematischen Trick in diesen Gleichungen. Eine partielle Integration von (5) ergibt.

$$\int_{x_i^e}^{x_j^e} N_i^{(e)} \left(K_x^{(e)} \frac{\partial^2 \hat{h}^{(e)}}{\partial x^2} \right) dx \quad (9)$$

$$= - \int_{x_i^e}^{x_j^e} K_x^{(e)} \frac{\partial N_i^{(e)}}{\partial x} \frac{\partial \hat{h}^{(e)}}{\partial x} dx + N_i^{(e)} K_x^{(e)} \frac{\partial \hat{h}^{(e)}}{\partial x} \Big|_{x_i^e}^{x_j^e}$$

Finite-Elemente-Methode (FEM)

- ▶ Wie können wir die Umformung in Gleichung (10) überprüfen?

Der zweite Term auf der rechten Seite von (10)

$$N_i^{(e)} K_x^{(e)} \frac{\partial \hat{h}^{(e)}}{\partial x} \Big|_{x_i^e}^{x_j^e} \quad (10)$$

entspricht der Vorgabe von Werten auf den Randknoten x_j^e und x_i^e des Elements (e). Handelt es sich um einen Randknoten, dann geht es um Randbedingungen.

Finite-Elemente-Methode (FEM)

- ▶ Um welchen Randbedingungstypen handelt es bei (10)?
- ▶ Welche Randbedingung ist es, wenn der Wert von (10) gleich Null ist?
- ▶ Was passiert mit inneren Knoten?

Finite-Elemente-Methode (FEM)

Nun setzen wir die Beziehung (10) in die Gleichung (5) ein und erhalten.

$$\begin{aligned}
 R_i^{(e)} &= \int_{x_i^e}^{x_j^e} N_i^{(e)} \left(K_x^{(e)} \frac{\partial^2 \hat{h}^{(e)}}{\partial x^2} \right) dx \\
 &= - \int_{x_i^e}^{x_j^e} K_x^{(e)} \frac{\partial N_i^{(e)}}{\partial x} \frac{\partial \hat{h}^{(e)}}{\partial x} dx + N_i^{(e)} K_x^{(e)} \frac{\partial \hat{h}^{(e)}}{\partial x} \Bigg|_{x_i^e}^{x_j^e} \quad (11)
 \end{aligned}$$

Finite-Elemente-Methode (FEM)

Jetzt müssen wir uns um $\partial \hat{h}^{(e)} / \partial x$ kümmern.

$$\frac{\partial \hat{h}^{(e)}}{\partial x} = \frac{\partial}{\partial x} \left(N_i^{(e)} h_i + N_j^{(e)} h_j \right) = \frac{\partial N_i^{(e)}}{\partial x} h_i + \frac{\partial N_j^{(e)}}{\partial x} h_j \quad (12)$$

Finite-Elemente-Methode (FEM)

Nach Einsetzen der Interpolationsfunktionen erhalten wir schließlich.

$$\frac{\partial \hat{h}^{(e)}}{\partial x} = \frac{1}{L^{(e)}}(-h_i + h_j) \quad (13)$$

Finite-Elemente-Methode (FEM)

Für die Ableitungen der linearen Interpolationsfunktionen folgt.

$$\frac{\partial N_i^{(e)}}{\partial x} = \frac{\partial}{\partial x} \left(\frac{x_j^{(e)} - x}{L^{(e)}} \right) = -\frac{1}{L^{(e)}} \quad (14)$$

$$\frac{\partial N_j^{(e)}}{\partial x} = \frac{\partial}{\partial x} \left(\frac{x - x_i^{(e)}}{L^{(e)}} \right) = \frac{1}{L^{(e)}} \quad (15)$$

Finite-Elemente-Methode (FEM)

Setzen wir jetzt die Beziehungen in die Gleichung (11) ein, ergibt sich.

$$\begin{aligned} R_i^{(e)} &= \int_{x_i^e}^{x_j^e} K_x^{(e)} \left(-\frac{1}{L^{(e)}} \right) \left(\frac{1}{L^{(e)}} \right) (-h_i + h_j) \\ &= \frac{K_x^{(e)}}{L^{(e)}} (h_i - h_j) \end{aligned} \quad (16)$$

In gleicher Weise bekommen wir.

$$R_j^{(e)} = \frac{K_x^{(e)}}{L^{(e)}} (-h_i + h_j) \quad (17)$$

Finite-Elemente-Methode (FEM)

Beide Gleichungen (16) und (17) lassen sich zusammen in einer Matrizen-Form schreiben.

$$\begin{Bmatrix} R_i^{(e)} \\ R_j^{(e)} \end{Bmatrix} = \frac{K_x^{(e)}}{L^{(e)}} \underbrace{\begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}}_{2 \times 2} \begin{Bmatrix} h_i \\ h_j \end{Bmatrix} \quad (18)$$

Finite-Elemente-Methode (FEM)

Leitfähigkeitsmatrix

$$[K^{(e)}] = \frac{K_x^{(e)}}{L^{(e)}} \underbrace{\begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}}_{2 \times 2} \quad (19)$$

Finite-Elemente-Methode (FEM)

Aufgrund der Elementgeometrien $L^{(e)}$ (Abb. 2) ergeben sich folgende Elementmatrizen.

$$\begin{aligned} [K^{(1)}] &= \begin{bmatrix} +1/2 & -1/2 \\ -1/2 & +1/2 \end{bmatrix}, & [K^{(2)}] &= \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix} \\ K^{(1)} &= \begin{bmatrix} +1/3 & -1/3 \\ -1/3 & +1/3 \end{bmatrix}, & [K^{(2)}] &= \begin{bmatrix} +1/3 & -1/3 \\ -1/3 & +1/3 \end{bmatrix} \end{aligned} \quad (20)$$

Finite-Elemente-Methode (FEM)

Zusammenbau des Gleichungssystems.

$$\{\mathbf{R}\} = [\mathbf{K}]\{\mathbf{h}\} = 0 \quad (21)$$

$$\{\mathbf{R}\} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \end{bmatrix}, \quad \{\mathbf{h}\} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \end{bmatrix} \quad (22)$$

$$[\mathbf{K}] = [\mathbf{K}^{(1)}] + [\mathbf{K}^{(2)}] + [\mathbf{K}^{(3)}] + [\mathbf{K}^{(4)}] \quad (23)$$

Finite-Elemente-Methode (FEM)

$$\begin{aligned}
 [\mathbf{K}] &= \begin{bmatrix} 1/2 & -1/2 & 0 & 0 & 0 \\ -1/2 & 1 + 1/2 & -1 & 0 & 0 \\ 0 & -1 & 1 + 1/3 & -1/3 & 0 \\ 0 & 0 & -1/3 & 1/3 + 1/3 & -1/3 \\ 0 & 0 & 0 & -1/3 & 1/3 \end{bmatrix} \\
 &= \begin{bmatrix} 1/2 & -1/2 & 0 & 0 & 0 \\ -1/2 & 3/2 & -1 & 0 & 0 \\ 0 & -1 & 4/3 & -1/3 & 0 \\ 0 & 0 & -1/3 & 2/3 & -1/3 \\ 0 & 0 & 0 & -1/3 & 1/3 \end{bmatrix} \quad (24)
 \end{aligned}$$

Finite-Elemente-Methode (FEM)

$$\begin{bmatrix} 1/2 & -1/2 & 0 & 0 & 0 \\ -1/2 & 3/2 & -1 & 0 & 0 \\ 0 & -1 & 4/3 & -1/3 & 0 \\ 0 & 0 & -1/3 & 2/3 & -1/3 \\ 0 & 0 & 0 & -1/3 & 1/3 \end{bmatrix} \begin{Bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \quad (25)$$

Finite-Elemente-Methode (FEM) - Implementierung

Vergleichen wir die Quelltexte der `main` Funktionen für FD und FE Verfahren, sehen wir kaum Unterschiede. Das heisst die Abläufe (Algorithmen) sind sehr ähnlich.

Finite-Elemente-Methode (FEM)

```
extern void Gauss(double*,double*,double*,int);

int main()
{
    //-----
    FEM* fem = new FEM();
    fem->SetInitialConditions();
    fem->SetBoundaryConditions();
    //-----
    int tn = 10;
    for(int t=0;t<tn;t++)
    {
        fem->AssembleEquationSystem();
        Gauss(fem->matrix,fem->vecb,fem->vecx,fem->IJ);
        fem->SaveTimeStep();
        fem->OutputResults(t);
    }
    //-----
    return 0;
}
```