

Modellbildung und Simulation

Kapitel 7: Systeme mit verteilten Parametern

Balázs Pritz
pritz@kit.edu

Institut für Thermische Strömungsmaschinen
Prof. Dr.-Ing. Hans-Jörg Bauer



Quelle: Xflow Product Sheet – www.xflowcf.com

Evaluation zum Kapitel 7

■ 10 Min. 

■ Evaluation nur zum Kapitel „Systeme mit verteilten Parametern“

■ Bewertungen zur

- Saalübung zu Modellreduktion
- Saalübung zu MGR, FDM
- (Poolübung zu FDM)

bitte unter dem Punkt „5. Zusatzfragen“ eingeben!

Lernziele der heutigen Vorlesung

- Die Studierenden
 - kennen die meist verwendeten Lösungsverfahren und ihre wichtigsten Eigenschaften.
 - kennen die Notwendigkeit der Modellierung komplexer Phänomene bei Mehrskalenproblemen.

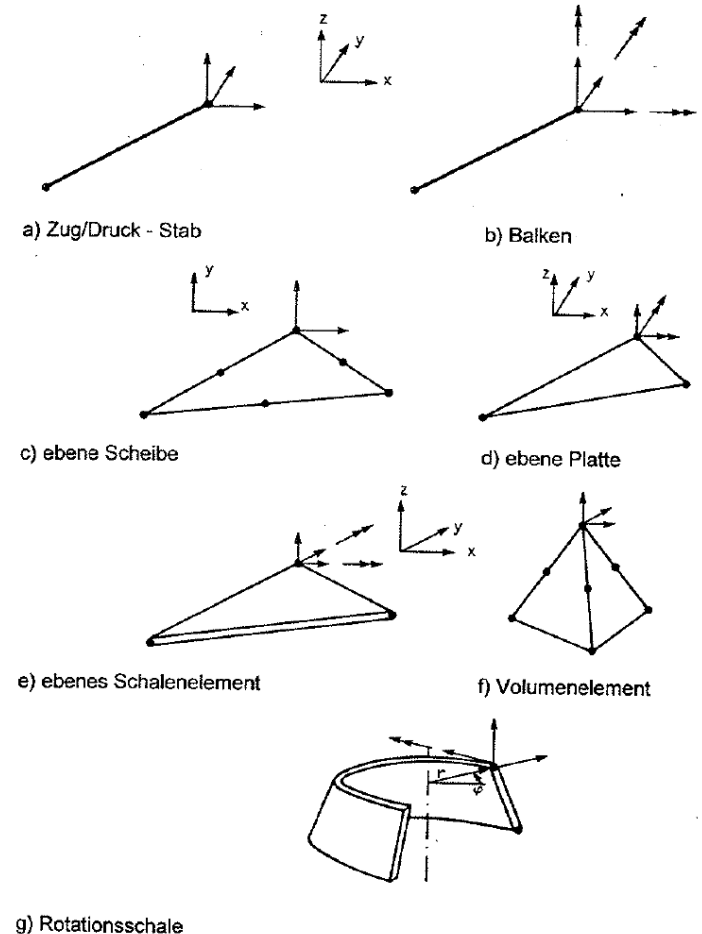
■ Hinweise zur Poolübung

Finite Elemente Methode

- FEM: Unterteilen des Rechengebietes in finite Elemente

- Je nach Problemstellung geeignete Definition der Elemente (Zugstab, Biegebalken, Platte, Volumenelement ...)

- Idee: Suchen nach der Lösung, bei der Energie des System minimal wird: $\delta \Pi = 0$



FEM: Allgemeine Vorgehensweise

FEM: Allgemeine Vorgehensweise

1. Ansatzfunktion $u_i^{approx}(x, y, z)$ für jedes Element i in **lokalem Koordinatensystem** (x_i, y_i, z_i)

(z.B. lineares Zugstabelement mit 2 FHG: $u_i^{approx}(x, y, z) = a_0 + a_1 x$)

2. Anpassung der Koeffizienten an Knotenverschiebung $u_i^{approx}(0, y, z) = u_i^1$,

$$u_i^{approx}(L_i, y, z) = u_i^2$$

3. Diskretisierung der Ansatzfunktion: $u_i^{approx}(x, y, z) = \mathbf{N}_i(x_i, y_i, z_i) \mathbf{u}_i$

Interpolationsmatrix

z. B.: Zugstab: $u_i^{approx}(x, y, z) = [1 - x/L_i, x/L_i] \begin{bmatrix} u_i^1 \\ u_i^2 \end{bmatrix}$

4. Diskretisierung der Dehnung $\varepsilon_i = \mathbf{B}_i(x_i, y_i, z_i) \mathbf{u}_i$ und Spannung $\sigma_i = \mathbf{C}_i \mathbf{B}_i(x_i, y_i, z_i) \mathbf{u}_i$

5. Berechnung der Elementsteifigkeit $\mathbf{K}_i = \int_V \mathbf{B}_i^T(x_i, y_i, z_i) \mathbf{C}_i \mathbf{B}_i(x_i, y_i, z_i) dV$

$$\rightarrow \mathbf{K}_{Ges} = \sum_{i=1}^N (\mathbf{K}_i)_{(X,Y,Z)} \quad \text{im globalen Koordinatensystem } (X, Y, Z)$$

6. Mit $\delta \Pi = 0$ erhält man $\mathbf{f} = \mathbf{K}_{Ges} \mathbf{u}$

7. Lösen des linearen Gleichungssystems $\begin{pmatrix} \mathbf{f}_a \\ \mathbf{f}_b \end{pmatrix} = \begin{pmatrix} \mathbf{K}_{Ges}^{11} & \mathbf{K}_{Ges}^{12} \\ \mathbf{K}_{Ges}^{21} & \mathbf{K}_{Ges}^{22} \end{pmatrix} \begin{pmatrix} \mathbf{u}_a \\ \mathbf{u}_b \end{pmatrix}$ mit den bekannten Verschiebungsrandbedingung \mathbf{u}_a und den äußeren Kräften \mathbf{f}_b

Lokales
KOS

Globales
KOS

Übersicht

7 Systeme mit verteilten Parametern

7.3 Numerische Lösungsverfahren für PDGLen

7.3.6 Lösungsverfahren

Direkte Methoden

Iterative Methoden

...

Zeitintegration

Lösungsverfahren

Es sei ϕ eine der Erhaltungsgrößen. Ersetzt man alle Ableitungen von ϕ in der entsprechenden Erhaltungsgleichung durch eine Differenzenformel, so erhält man ein algebraisches Gleichungssystem (AGS) mit einer Koeffizientenmatrix A und dem Lösungsvektor $\phi = (\dots, \phi_{i-2}, \phi_{i-1}, \phi_i, \phi_{i+1}, \phi_{i+2}, \dots)$, das das räumliche Voranschreiten der Lösungsinformation beschreibt.

AGS mit Koeffizientenmatrix A :
$$A\phi = Q$$

Solch ein AGS müsste für jede zu lösende Variable (Erhaltungsvariable) aufgestellt werden.

Das AGS kann entweder **direkt** und/oder **iterativ** gelöst werden.

Lösungsverfahren

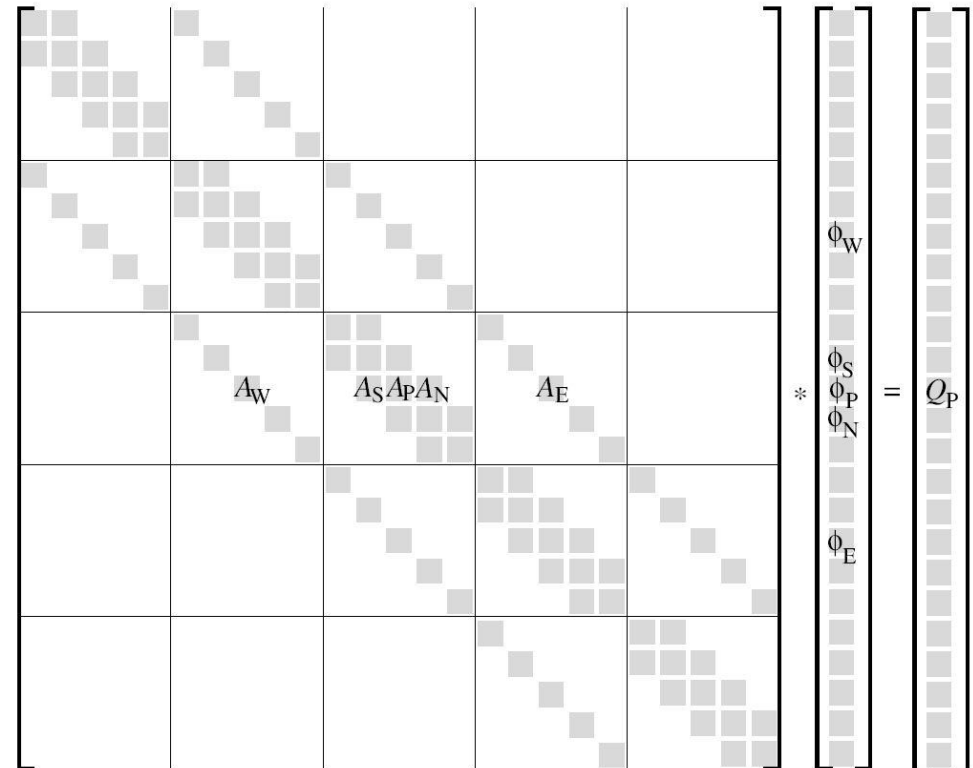
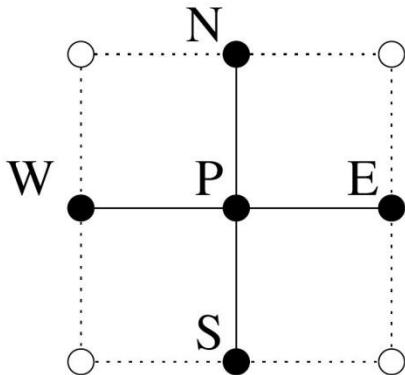
Beispiel: Das algebraische Gleichungssystem für 2D Rechnetz

Graphische Darstellung der Matrix für 2D Netz mit 5 Punkten Rechenstern

A – geometrische Größen, Fluideigenschaften, bei nichtlinearen GL die Variablen auch

Q – Terme ohne unbekannten Variablen

$$A_P \phi_P + \sum_l A_l \phi_l = Q_P$$



Direkte Lösungsverfahren

Die folgenden Matrizenlösungsverfahren bilden eine Basis für die direkten Methoden:

- **Gaußsches Eliminationsverfahren**
- **LU-Zerlegung**
- **Thomas Algorithmus**

Direkte Methoden können für beliebige, vollbesetzte Matrizen verwendet werden und liefern exakte Lösungen.

Die sind aber nicht *effizient*: Operationen zu Lösung des AGS $\sim n^3/3$ (Gauß'sches Verfahren) bzw. $\sim n^2/2$ (LU-Zerlegung).

Direkte Lösungsverfahren

Gaußsches Eliminationsverfahren

Idee: Modifikation des AGS so, dass die volle Matrix A zu einer rechten oberen Dreiecksmatrix U wird (Vorwärtselimination).

Elimination der unteren Dreieckselemente durch Multiplikation der ersten Zeile der Matrix mit A_{i1} / A_{11} und Subtraktion von der i -ten Reihe. Q muss auch angepasst werden.

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1n} \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & A_{n3} & \cdots & A_{nn} \end{pmatrix} \longrightarrow U = \begin{pmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1n} \\ 0 & A_{22} & A_{23} & \cdots & A_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & A_{nn} \end{pmatrix}$$

Dann Rücksubstitution beginnend mit letzter Reihe.

Anzahl der Operationen $\sim n^3$

Direkte Lösungsverfahren

LU-Zerlegung

Modifikation von Gaußschen Eliminationsverfahren.

Finde linke untere bzw. rechte obere Dreieckmatrix L bzw. U mit $A=LU$.

$$A\phi = LU\phi = Q$$

Löse zuerst (~ Rücksubstitution):

$$LY = Q$$

und anschließend (~ Rücksubstitution):

$$U\phi = Y$$

Vorteil: die LU-Faktorisierung kann durchgeführt werden, ohne Q zu kennen.

Basis für einige iterativen Methoden.

Direkte Lösungsverfahren

Thomas-Algorithmus (Tridiagonal Matrix-Algorithmus – TDMA)

Wenn die Matrix A die Form einer tridiagonalen Bandmatrix hat (streng genommen nur bei 1-dimensionalen Problemen) dann ist das Gaußsche Eliminationsverfahren sehr effizient (proportional zu n) und einfach zu programmieren:

$$\text{LGS: } A_W^i \phi_{i-1} + A_P^i \phi_i + A_E^i \phi_{i+1} = Q_i$$

$$\text{VE: } A_P^i = A_P^i - \frac{A_W^i A_E^{i-1}}{A_P^{i-1}} \quad Q_i^* = Q_i - \frac{A_W^i Q_{i-1}^*}{A_P^{i-1}}$$

$$\text{RS: } \phi_i = \frac{Q_i^* - A_E^i \phi_{i+1}}{A_P^i}$$

Direkte Lösungsverfahren

Direkte Methoden finden Einsatz hauptsächlich in der Lösung von Problemen in der Strukturmechanik.

Beispiele bei Open Source Programmen:

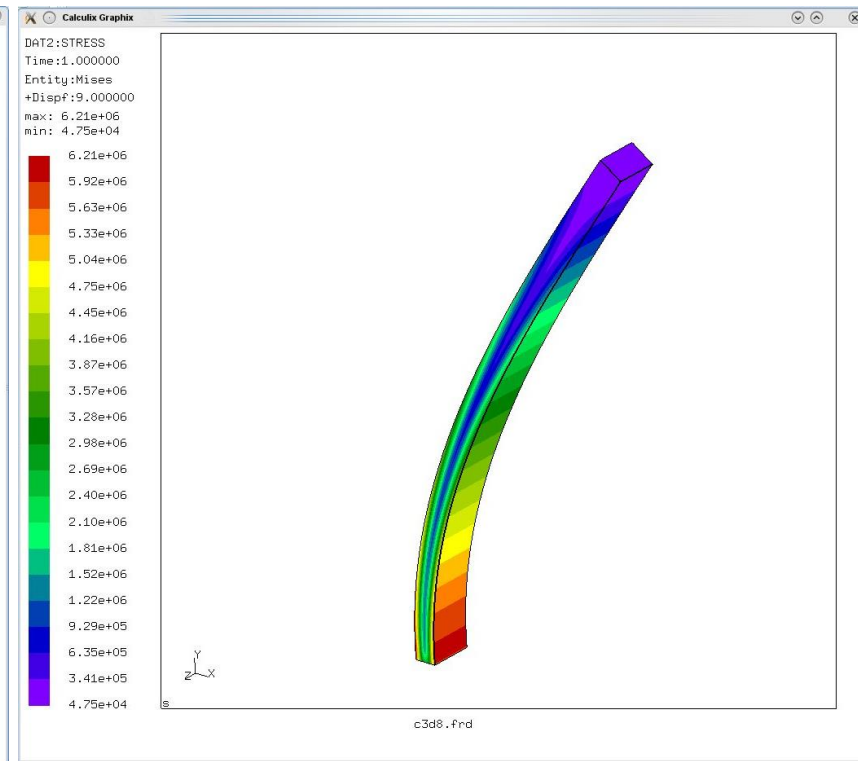
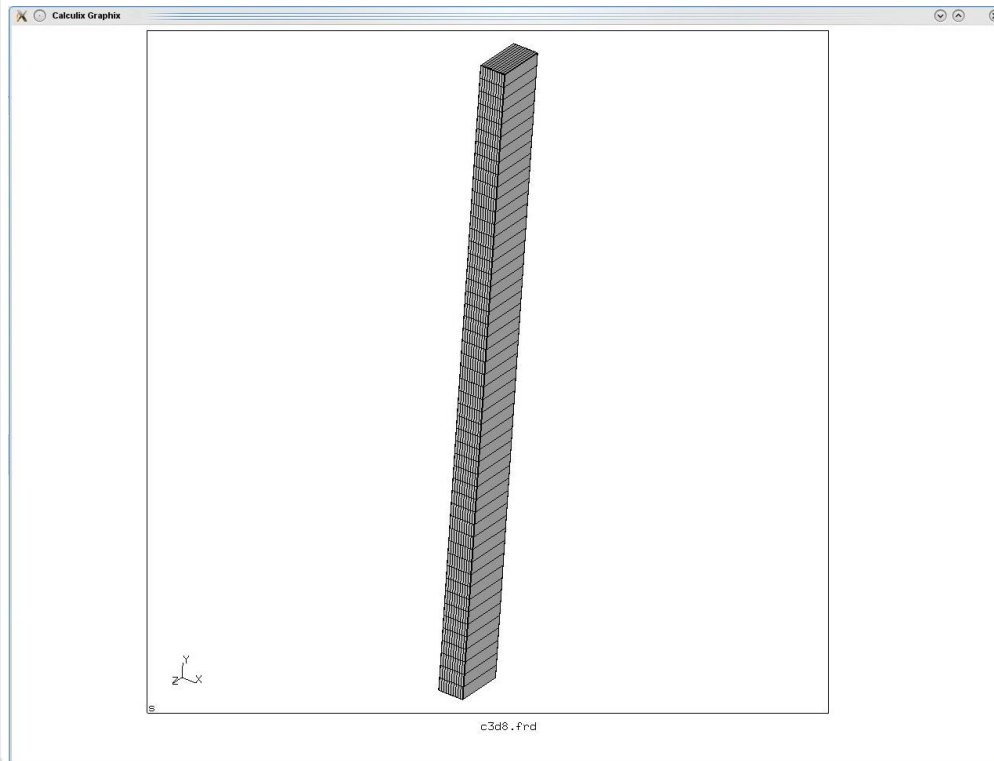
- CalculiX löst das lineare Gleichungssystem mit der QR-Zerlegung Methode, implementiert in der Bibliothek SPOOLES.
- Z88 benutzt das Cholesky Verfahren, welche nur für positiv-definite Matrizen funktioniert.

Weitere Beispiele können Sie unter
<http://web.eecs.utk.edu/~dongarra/etemplates/node388.html>
finden.

Direkte Lösungsverfahren

Beispiel mit Calculix. Statische mechanische Analyse.

Balken: an einem Ende gehalten, an anderem Ende wird durch eine Einzelkraft belastet.



Iterative Lösungsverfahren

Die Zerlegung produziert von dünn besetzten Matrizen nicht dünn besetzten Dreieckmatrizen \Rightarrow hohe Rechenkosten.

Diskretisierungsfehler ist viel größer als die Genauigkeit der Rechenarithmetik \Rightarrow AGS sehr genau zu lösen ist nicht notwendig.

\Rightarrow Iterative Methoden:

- für nichtlineare Probleme (anders geht es nicht, z.B. NS-Gl. konvektiver Term)
- für dünn besetzte lineare Systeme.

Gestartet wird mit einer abgeschätzter Lösung $\phi^{(0)}$, welche iterativ verbessert wird:

$$\phi^{(n+1)} = f(\phi^{(n)})$$

Die exakte Lösung ϕ wird immer besser approximiert.

Iterative Lösungsverfahren

$$A\phi = \mathbf{Q} \quad \Rightarrow \quad A\phi^n = \mathbf{Q} - \rho^n \quad \text{nach n Iterationen die Näherungslösung mit einem Rest}$$

Iterationsfehler: $\varepsilon^n = \phi - \phi^n$

Residuum: $A\varepsilon^n = \rho^n$

Das Residuum bzw. der Iterationsfehler müssen gegen null getrieben werden.

Der Iterationsfehler wird oft durch den Korrektur (Update) $\delta^n = \phi^{n+1} - \phi^n$ approximiert.

Iterative Lösungsverfahren

Grundlegende iterative Methoden:

- Jacobi-Methode
- Gauß-Seidel-Methode
- Sukzessive Überrelaxationsmethode (SOR)
- Unvollständige LU-Zerlegung (ILU)

Eine weit verbreitete Vorgehensweise zur Lösung von **stationären**, elliptischen oder hyperbolischen Problemen ist die Erweiterung der Gleichungen mit einem instationären Term. Bei diesem, dann parabolischen Problem, fällt der Zusatzterm im auskonvergierten Zustand heraus, und man erhält die Lösung des Ausgangsproblems.*

* Poolübung zu FDM

Integration in der Zeit

$$\frac{\partial(\rho\phi)}{\partial t} + \frac{\partial(\rho v_j \phi)}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\Gamma \frac{\partial \phi}{\partial x_j} \right) + q_\phi$$

- s. Vorlesung 6 bzw. Kapitel 5
- Explizite* vs. implizite Methoden
- CFL-Zahl*

* Poolübung zu FDM

Übersicht

7 Systeme mit verteilten Parametern

7.4 Modellierung komplexer Phänomene

Modellierung komplexer Phänomene

Kann das Modell alles beschreiben?

Kann ich alles simulieren? Wie viel Zeit habe ich? Wie schnell ist mein Rechner?

Wie kann man in moderater Zeit ein Ergebnis liefern?

Multiskalenproblem: Verschiedene Vorgänge können eine sehr feine Diskretisierung in der Zeit und/oder im Raum benötigen während das gesamte Integrationsgebiet und/oder die Integrationszeit um mehrere Größenordnung größer ist.

Weitere Modellierung ist nötig:

- Turbulenz: - RANS (Reynolds-averaged Navier-Stokes)
 - LES (Large-Eddy Simulation)
 - DNS (Direct Numerical Simulation)
- Oberflächenrauigkeit
- chemische Reaktionen (Verbrennung)
- Zweiphasenströmungen (freie Oberfläche, Schaum)
- poröse Medien
- ...

Ausblick

- Nächste Übungen:
 - Poolübung zu FDM
 - Klausurvorbereitung II: 13.01.2020
 - Übung zu FEM (ITM)
 - Poolübung zu FEM (ITM)

- Nächste Vorlesungen (IFL):
 - Verifikation und Validierung
 - Versuchsplanung