

# **Präkonditionierer für zell-basierte finiten Elemente Operatoren**

Bachelorarbeit

eingereicht von

**Enes Witwit**

betreut von

**Prof. Dr. Kanschat**

**Fakultät für Mathematik und Informatik**

**Universität Heidelberg**

---

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig verfasst habe. Ich versichere, dass ich keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe, und dass die eingereichte Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen ist.

30. März 2017

Heidelberg

Unterschrift

## Zusammenfassung

# Inhaltsverzeichnis

<b>Notation</b>	<b>V</b>
<b>Abbildungsverzeichnis</b>	<b>VI</b>
<b>Tabellenverzeichnis</b>	<b>VII</b>
<b>1 Einführung</b>	<b>1</b>
<b>2 Theorie</b>	<b>2</b>
2.1 Schwache Lösungen . . . . .	3
2.2 Galerkin Verfahren . . . . .	5
2.3 Methode der finiten Elemente . . . . .	6
2.4 Diskontinuierliche Galerkin-Methode . . . . .	7
2.5 Tensor Dekomposition . . . . .	8
2.6 Summenfaktorisierung . . . . .	9
<b>3 Präkonditionierer für zell-basierte finiten Elemente Operatoren</b>	<b>12</b>
<b>4 Numerische Untersuchungen</b>	<b>12</b>
<b>5 Resultate</b>	<b>12</b>

## Notation

**Abbildungsverzeichnis**

**Abbildungsverzeichnis**

## **Tabellenverzeichnis**

# 1 Einführung



## **2 Theorie**

## 2.1 Schwache Lösungen

Es ist naheliegend, dass wir uns zu erst mit notwendigen Funktionenräumen beschäftigen und uns auf analytischer Ebene eine Umformulierung der Differentialgleichung zu nutze machen, welche uns letztlich die Grundlage für die finiten Elementen Methode liefert.

Dazu schauen wir uns folgendes Randwert Problem an:

$$\begin{aligned} -\Delta u &= f \text{ in } \Omega \\ u &= 0 \text{ in } \partial\Omega \end{aligned} \tag{2.1}$$

Wir sehen von der Form der Differentialgleichung, dass die gesuchte Lösung bestimmte Differenzierbarkeits- und Stetigkeitsbedingungen zu erfüllen hat. Nämlich, dass  $u$  zweimal stetig differenzierbar sein sollte. Dementsprechend legen die Differenzierbarkeitsanforderungen den Raum  $u \in C_0^2$  nahe.

Nun kann es aber sein, dass eine Lösung für dieses Problem garnicht in diesem Raum existiert. Wir können uns dafür als Beispiel die Betragsfunktion anschauen.

$$f(x) = |x| \tag{2.2}$$

Offensichtlich ist diese Funktion in 0 nicht stetig differenzierbar. Trotzdem können wir eine Ableitung finden, die wir schwache Ableitung nennen.

$$f'(x) = \begin{cases} -1 & \text{für } x < 0 \\ 0 & \text{für } x = 0 \\ 1 & \text{für } x > 0 \end{cases} \tag{2.3}$$

Nun um das was wir jetzt für die Betragsfunktion gemacht haben, für unser Randwertproblem zu machen nutzen wir eine funktionalanalytische Idee die aus der Distributionstheorie stammt. Wir multiplizieren mit einer Testfunktion  $\psi$  und integrieren über das Gebiet.

$$\int_{\Omega} -\Delta u \psi dx = \int_{\Omega} f \psi dx \tag{2.4}$$

Der nächste Schritt, welcher durchwegs fundamental für die Herleitung ist, ist die intuitive Nutzung der Struktur und Integrationswerkzeuge um zu erreichen, dass an  $u$  weniger Differenzierbarkeitsanforderungen gebunden sind. Für diesen

Schritt ist der Satz von Green und die partielle Integration von Wichtigkeit.

**Lemma 2.1.** *Satz von Green*

**Lemma 2.2.** *Partielle Integration*

Für unsere Differentialgleichung (2.4) nutzen wir die partielle Integration und erhalten.

$$\int_{\Omega} -\nabla u \nabla \psi dx = \int_{\Omega} f \psi dx \quad (2.5)$$

Dies ist die so genannte Variationsgleichung. Sie ist ein erster Indiz für die später gewünschte Bilinearform der zugrundeliegenden Topologie. Die Lösung  $u$  von (2.5) nennt man *schwache Lösung* für das Problem (2.1). Die Lösung  $u \in C_0^2$  von (2.1) nennt man *klassische Lösung*. Nun wissen wir in welchem Raum die klassische Lösung liegt, aber welche Topologie ist für die schwache Lösung sinnvoll? Ein funktionalanalytischer Ansatz versucht nun die Räume zu definieren, in der die Lösung  $u$  für (2.5) liegt. In unserem Fall wäre folgender Raum ergiebig:

$$H_0^1(\Omega) = \{v \in L_2(\Omega) : \frac{\delta v}{\delta x_i} \in L_2(\Omega), v = 0 \text{ in } \partial\Omega, i = 1, \dots, d\}$$

Diese Räume nennt man Sobolev Räume. Allgemein sind sie definiert durch:

**Definition 2.3.** *Sobolev Raum*

Das heißt Sobolev Räume sind eine Teilmenge von den  $L_2$  Räumen. Von der analytischen Perspektive ist die Wahl des Funktionenraumes essentiell für den Nachweis der Existenz der Lösung. Von der Perspektive der finiten Elementen Methode ist dies für die Fehlerabschätzung wichtig, da wir dann die induzierte Norm des Funktionenraumes benutzen [Joh08, 36]. Beide genannten Themen würden den Rahmen dieser Bachelorarbeit sprengen, daher verweise ich an gegebenen Stellen an weiterführende Literatur.

Die Sobolev Räume wurden mit Skalarprodukten ausgestattet, sodass unsere Variationsgleichung intuitiv als Skalarprodukt der zugrundeliegenden Sobolev Räume geschrieben werden kann.

**Lemma 2.4.** *Sobolev Norm*

**Lemma 2.5.** *Sobolev Skalarprodukt*

## **2.2 Galerkin Verfahren**

## **2.3 Methode der finiten Elemente**

## **2.4 Diskontinuierliche Galerkin-Methode**

## 2.5 Tensor Dekomposition

## 2.6 Summenfaktorisierung

Nun haben wir uns mit Hilfe der Tucker Dekomposition eine Herleitung für die Pseudoinverse erarbeitet. Nun geht es um die effiziente Berechnung dieser Formel. Dazu wollen wir uns die Summenfaktorisierung zu nutze machen, wie sie auch in [Tea, 9-11] vorgeschlagen wird. Sei  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_n}$ . Die Formel für die Pseudoinverse lautet:

$$\mathcal{A}^\dagger = \mathcal{S}^\dagger \times_{n=1}^N \mathbf{U}^{(n)\top} \quad (2.6)$$

Wobei  $\mathcal{S} \in \mathbb{R}^{I_1 \times \dots \times I_n}$  und  $\mathbf{U}^{(n)} \in \mathbb{R}^{J_n \times I_n}$ . Man kann 2.6 nach [TK09, 462] äquivalent umformen zu

$$\begin{aligned} \mathcal{A}_{(n)}^\dagger &= \mathbf{U}^{(n)\top} \mathcal{S}_{(n)}^\dagger (\mathbf{U}^{(N)\top} \otimes \dots \otimes \mathbf{U}^{(n+1)\top} \otimes \mathbf{U}^{(n-1)\top} \otimes \dots \otimes \mathbf{U}^{(1)\top})^\top \\ \iff \mathcal{A}_{(n)}^\dagger &= \mathbf{U}^{(n)\top} \mathcal{S}_{(n)}^\dagger (\mathbf{U}^{(N)} \otimes \dots \otimes \mathbf{U}^{(n+1)} \otimes \mathbf{U}^{(n-1)} \otimes \dots \otimes \mathbf{U}^{(1)}) \end{aligned} \quad (2.7)$$

Nun betrachten wir uns das Matrix-Vektor Produkt und überlegen uns wie wir uns die Strukturen dort zu nutze machen.

$$\mathcal{A}_{(n)}^\dagger \mathbf{v} = \mathbf{U}^{(n)\top} \mathcal{S}_{(n)}^\dagger (\mathbf{U}^{(N)} \otimes \dots \otimes \mathbf{U}^{(n+1)} \otimes \mathbf{U}^{(n-1)} \otimes \dots \otimes \mathbf{U}^{(1)}) \mathbf{v} \quad (2.8)$$

Wir schauen uns die Struktur mal für den Fall, dass  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_4}$ . Das heißt 2.8 reduziert sich auf:

$$\mathcal{A}_{(n)}^\dagger \mathbf{v} = \mathbf{U}^{(n)\top} \mathcal{S}_{(n)}^\dagger (\mathbf{U}^{(N_1)} \otimes \mathbf{U}^{(N_2)} \otimes \mathbf{U}^{(N_3)}) \mathbf{v} \quad (2.9)$$

mit  $N_i \neq n$ .

Zu Zwecken der Veranschaulichung nehmen wir eine Umdefinierung vor:  $A := \mathbf{U}^{(N_3)}$ ,  $B := \mathbf{U}^{(N_2)}$ ,  $C := \mathbf{U}^{(N_1)}$ . Wenn wir uns die Struktur in der Klammer anschauen sieht diese wie folgt aus



$$z := \begin{pmatrix} c_{11}b_{11}A & \dots & c_{11}b_{1n}A & \dots & \dots & c_{1n}b_{11}A & \dots & c_{1n}b_{1n}A \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ c_{11}b_{n1}A & \dots & c_{11}b_{nn}A & \dots & \dots & c_{1n}b_{n1}A & \dots & c_{1n}b_{nn}A \\ c_{21}b_{n1}A & \dots & c_{21}b_{nn}A & \dots & \dots & c_{2n}b_{n1}A & \dots & c_{2n}b_{nn}A \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ c_{n1}b_{n1}A & \dots & c_{n1}b_{nn}A & \dots & \dots & c_{nn}b_{n1}A & \dots & c_{nn}b_{nn}A \end{pmatrix} * v \quad (2.10)$$

Wir sehen hier sich zwei wiederholende Strukturen die wir ausnutzen können um bei einem Matrix-Vektor Produkt operationen zu sparen.

$$z = \begin{pmatrix} c_{11}\textcolor{red}{b}_{11}A & \dots & c_{11}\textcolor{red}{b}_{1n}A & \dots & \dots & c_{1n}\textcolor{red}{b}_{11}A & \dots & c_{1n}\textcolor{red}{b}_{1n}A \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ c_{11}\textcolor{red}{b}_{n1}A & \dots & c_{11}\textcolor{red}{b}_{nn}A & \dots & \dots & c_{1n}\textcolor{red}{b}_{n1}A & \dots & c_{1n}\textcolor{red}{b}_{nn}A \\ c_{21}\textcolor{red}{b}_{n1}A & \dots & c_{21}\textcolor{red}{b}_{nn}A & \dots & \dots & c_{2n}\textcolor{red}{b}_{n1}A & \dots & c_{2n}\textcolor{red}{b}_{nn}A \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ c_{n1}\textcolor{red}{b}_{n1}A & \dots & c_{n1}\textcolor{red}{b}_{nn}A & \dots & \dots & c_{nn}\textcolor{red}{b}_{n1}A & \dots & c_{nn}\textcolor{red}{b}_{nn}A \end{pmatrix} * v \quad (2.11)$$

Nun wollen wir uns dies zu nutze machen. Wir schauen uns erstmal die einzelnen Einträge von  $z$  an und bekommen. Vorher definieren wir unser  $v$  um zu einem Tensor  $\mathcal{V} \in \mathbb{R}^{l_1 \times l_2 \times l_3}$ . Der erste Index repräsentiert in welcher Spalteneintrag von  $C$  wir uns befinden, der zweite in welchem Spalteneintrag von  $B$  und der Dritte in welchem Spalteneintrag von  $A$ .

$$z_1 = \mathcal{V}(1, 1, 1)c_{11}\textcolor{red}{b}_{11}\textcolor{red}{a}_{11} + \dots + \mathcal{V}(1, 1, n)c_{11}\textcolor{red}{b}_{11}\textcolor{red}{a}_{1n} + \dots + \mathcal{V}(1, n, 1)c_{11}\textcolor{red}{b}_{1n}\textcolor{red}{a}_{11} \\ + \dots + \mathcal{V}(n, 1, 1)c_{1n}\textcolor{red}{b}_{11}\textcolor{red}{a}_{11} + \dots + \mathcal{V}(n, n, n)c_{1n}\textcolor{red}{b}_{1n}\textcolor{red}{a}_{1n} \quad (2.12)$$

Definiere  $w_1(i, j) := \mathcal{V}(i, j, 1)\textcolor{red}{a}_{11} + \dots + \mathcal{V}(i, j, n)\textcolor{red}{a}_{1n}$ . Dann erhalten wir:

$$z_1 = w_1(1, 1)c_{11}b_{11} + \dots + w_1(1, n)c_{11}b_{1n} + \dots + w_1(n, 1)c_{1n}b_{11} + \dots + w_1(n, n)c_{1n}b_{1n}$$

Damit haben wir uns die sich wiederholende Struktur von der Matrix  $A$  zu nutze gemacht. Im nächsten Schritt machen wir uns die sich wiederholende Struktur von  $\textcolor{red}{b}_{ij}$  zu nutze. Definiere hierfür  $\mathfrak{w}_{1,k}(i) := w_k(i, 1)\textcolor{red}{b}_{11} + \dots + w_k(i, n)\textcolor{red}{b}_{1n}$ . Damit erhalten wir:

$$z_1 = \mathfrak{w}_{1,1}(1)c_{11} + \cdots + \mathfrak{w}_{1,1}(n)c_{1n} \quad (2.13)$$

Wir wollen nun  $z$  genau so umformen wie wir das auch für  $v$  gemacht haben. Damit erhalten wir für allgemeines  $z_i$  folgende Formel:

$$\mathcal{Z}(i, j, k) = \mathfrak{w}_{j,k}(1)c_{i1} + \cdots + \mathfrak{w}_{j,k}(n)c_{in} \quad (2.14)$$

Wobei  $j$  und  $k$  den Zeilen jeweils in den Matrizen  $B$  und  $C$  entsprechen.

Der komplette Algorithmus würde nun wie folgt aussehen:

```

for k=1 < n do
  for i= 1 < n do
    for j= 1 < n do
       $w_k(i, j) = \mathcal{V}(i, j, 1)a_{k1} + \cdots + \mathcal{V}(i, j, n)a_{kn}$ 
    end for
  end for
end for
for k=1 < n do
  for i= 1 < n do
    for j= 1 < n do
       $\mathfrak{w}_{i,j}(k) := w_k(i, 1)b_{11} + \cdots + w_k(i, n)b_{1n}$ 
    end for
  end for
end for
for k=1 < n do
  for i= 1 < n do
    for j= 1 < n do
       $\mathcal{Z}(i, j, k) = \mathfrak{w}_{j,k}(1)c_{i1} + \cdots + \mathfrak{w}_{j,k}(n)c_{in}$ 
    end for
  end for
end for

```

Wenn wir annehmen, dass die Matrizen  $A, B, C \in \mathbb{R}^{n \times n}$ . Dann haben wir bei 2.11 eine Matrix-Vektor Multiplikation von einer Matrix der Größe  $n^3 \times n^3$ . Dementsprechend hätten wir  $n^6$  Multiplikationen und  $n^6$  Additionen. Die Komplexität des vorgeschlagenen Algorithmuses reduziert sich auf  $3n^4$  Multiplikationen und genau so viele Additionen. Ein enorme Reduktion, vor allem für großes  $n$ .

**3 Präkonditionierer für zell-basierte finiten  
Elemente Operatoren**

**4 Numerische Untersuchungen**

**5 Resultate**

## Literatur

- [Joh08] Claes Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Dover Publications, 2008.
- [Tea] *Efficient evaluation of weak forms in discontinuous Galerkin methods*.
- [TK09] Brett Bader Tamara Kolda. *Tensor Decompositions and Applications*. SIAM, 2009.