Pseudoinverse für zell-basierte finiten Elemente Operatoren

Bachelorarbeit

eingereicht von

Enes Witwit

betreuut von

Prof. Dr. Kanschat

Fakultät für Mathematik und Infromatik

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 Prüfungsverfahren gewesen ist.

30. April 2017 Heidelberg Unterschrift

Zusammenfassung

Inhaltsverzeichnis

Notation **V**

Abbildungsverzeichnis **VI**

Tabellenverzeichnis **VII**

1 Einführung **1**

2 Theorie **2**

2.1 Schwache Lösungen 3
2.2 Galerkin Verfahren 6
2.3 Methode der finiten Elemente 8
2.4 Diskontinuierliche Galerkin-Methode 10
2.5 Tensor Dekomposition 11
2.6 Summenfaktorisierung 12

3 Pseudoinverse für zell-basierte finiten Elemente Operatoren **15**

3.1 Tensorprodukt Struktur 15
3.2 HOSVD 15

4 Numerische Untersuchungen **15**

5 Resultate **15**

Notation

Abbildungsverzeichnis

Abbildungsverzeichnis

1 Ansatzfunktionen φ_i 9

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:

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

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 ψ 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.

$$\int_\Omega -\nabla u \nabla \psi dx = \int_\Omega f\psi dx \tag{2.5}$$

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:

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

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 verweis 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*

Lemma 2.6. *Falls die Bilinearform symmetrisch ist* $u \in V$ *ein Minimierer der Gleichung*

$$J(u) = \min_{v \in V} J(v) = \frac{1}{2} a(u, v) - f(v) \tag{2.6}$$

genau dann wenn u die schwache Formulierung löst.

2.2 Galerkin Verfahren

(Numerik 2 Skript Kanschat) Unser Ausgangspunkt ist nun

$$\text{Finde } u \in \Omega : a(u,v) = f(v) \forall u \in \Omega \text{ und } v \in V \tag{2.7}$$

Da die Sobolev Räume unendlich dimensional sind, ist es schwer sich mit der Konstruktion einer Lösung vertraut. Daher ist die Kernidee des Galerkin Verfahrens unseren Banachraum zu diskretisieren. Dazu führen wir eine sogenannte konforme Approximation durch. Wir wählen $V_n \subset V$, sodass $\dim V_n = n < \infty$. Nun gilt für die Minimierung in (2.6)

$$u = \arg\min_{v \in V} J(v) \text{ (stetig)}$$
$$u_n = \arg\min_{v_n \in V_n} J(v_n) \text{ (diskret)}$$

Daraus folgt

$$J(u_n) \geq J(u)$$

Dies folgt direkt aus der Wahl des Raumes als Teilraum des ursprünglichen Raumes. Man nennt diese Methode konforme Ritz-Galerkin Methode aus dem Grund, dass der diskrete Raum ein Teilraum von dem ursprüngichen Raum is und die Funktion J gleich bleibt. Was hat uns das Ganze gebracht? Nun da, $\dim V_n = n < \infty$ können wir eine Basis für V_n. Das bringt uns den Vorteil, dass wir u_n als Linear Kombination der Basiselemente in V_n approximieren können.

Die schwache Formulierung sieht nun wie folgt aus

Lemma 2.7. *Schwache Formulierung (Galerkin)*
Finde $u_n \in V_n$*, sodass* $a(u_n, v_n) = f(v_n)\ \forall v_n \in V_n$.

Sei nun $e_1, \ldots, e_n$ eine Basis von V_n. Es ist nun ausreichend nur die Basis zum Testen zu nutzen. Die obere Gleichung in Lemma 2.7 reduziert sich auf

$$a(u_n, e_i) = f(e_i) \forall i \in \{1, \ldots, n\} \tag{2.8}$$

Im nächsten Schritt erweitern wir u_n als Linearkombination wie folgt

$$u_n = \sum_{j=1}^{n} u_j e_j \tag{2.9}$$

und setzen dies in 2.8 ein und erhalten.

$$\begin{aligned} &a(\sum_{j=1}^{n} u_j e_j, e_i) = f(e_i) \iff \\ &\sum_{j=1}^{n} u_j a(e_j, e_i) = f(e_i) \end{aligned} \tag{2.10}$$

Das können wir zusammenfassen in einem Linearen Gleichungssystem mit $A_{ij} = a(e_j, e_i)$ und $u = (u_1, \ldots, u_n)^T$. Insgesamt erhalten wir

$$Au = f \tag{2.11}$$

Bemerkung 2.8. *Eigenschaften Galerkin*

1. *Galerking Orthogonalität*
   Eine Kerneigenschaft der Galerkin Methode ist, dass der Fehler orthogonal zu dem Teilraum von V liegt.

2. *Symmetrie*
   Die Matrix A ist genau dann symmetrisch, wenn die Bilinearform symmetrisch ist.

2.3 Methode der finiten Elemente

Im Vorherigen Kapitel haben wir das Ritz-Galerkin Verfahren kennengelernt. Der Kernaspekt dieser konformer Approximation war eine diskretisierung des Raumes und mit einhergehend global einheitlich definierte Funktionen. Nun öffnen wir die letzt genannte Einschränkung und fordern nur noch stückweise definierte Funktionen, in der Regel Polynome. Wo genau eine Funktion definiert ist, hängt von unsererer Gebietszerlegung ab. Das heißt für die Finite Elemente Methode (FEM) ist es zu erst notwendig das Grundgebiet in geometrisch einfache Teilgebiete $\Omega_h = \{\Omega_k\}_{k=1\ldots N}$ z.B. Dreiecke und Rechtecke bei Problemen in der Ebene oder Tetraeder und Quader bei Problemen im dreidimensionalen Raum. Dann folgt eine Definition von Ansatz- und Testfunktionen über Teilgebieten. Da wir zwischen den Teilgebieten eine Stetigkeit fordern, definiert man Übergangsbedigungen die dann die Sicherung der Stetigkeit global sichert. (Grossmann Seite 175). Die Stetigkeit wird gefordert damit wir $V_n \subset V$ bekommen, da beispielsweise gelten kann $V \subset H^1(\Omega)$.

Bemerkung 2.9. *Natürliche Voraussetzungen an Zerlegung (Grossman S.176)*
Die Voraussetzungen an die Zerlegung $\mathscr{Z}=\{\Omega_j\}_{j=1}^m$ *sind*

1. $$\bar{\Omega} = \bigcup_{j=1}^{m} \bar{\Omega}_j$$

2. $$\mathrm{int}\Omega_i \cap \mathrm{int}\Omega_j = \emptyset \text{ , } \textit{falls } i \neq j.$$

Beispiel 2.10. $\Omega = [a, b]$ *(Grossmann S.184)*
Wir definieren Gitterpunkte $\{x_i\}_{i=0}^N$ *über* $\bar{\Omega}$ *beschrieben wie folgt:*

$$a = x_0 < x_1 < x_2 < \cdots < x_{N-1} < x_N = b$$

und eine Zerlegung $\mathscr{Z}=\{\Omega_j\}_{j=1}^m$ *mit* $\Omega_i := (x_{i-1}, x_i)$ $i = 1, \ldots, N$.

Ferner sei $h_i := x_i - x_{i-1}$, $i = 1, \ldots, N$. Wir wählen lineare Ansatzfunktionen damit

gilt $V_h = \text{lin}\{\varphi_i\}_{i=0}^N$ wobei die Ansatzfunktionen definiert sind durch

$$\varphi_i(x) = \begin{cases} \frac{1}{h_i}(x - x_{i-1}), \text{ für } x \in \Omega_i \\ \frac{1}{h_{i+1}}(x_{i+1} - x) \text{ für } x \in \Omega_{i+1} \\ 0, \text{ sonst} \end{cases} \tag{2.12}$$

Es gilt nach Konstruktion $\varphi \in C(\bar{\Omega})$ sowie $\varphi_i\big|_{\Omega_j} \in C^1(\bar{\Omega}_j)$, somit hat man insgesamt $\varphi_i \in H^1(\Omega)$. Die folgende Abbildung stellt die Graphen von Ansatzfunktionen φ_i dar.

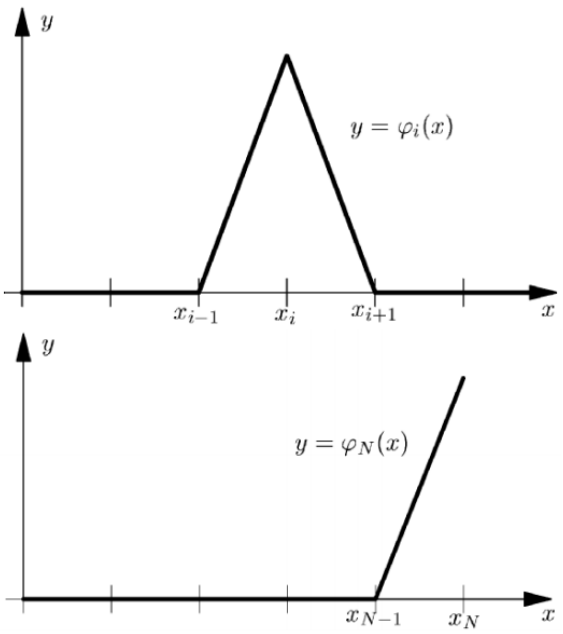

Abbildung 1: Ansatzfunktionen φ_i

Es gilt demnach $\varphi_i(x_k) = \delta_{ik}, i, k = 0, 1, \ldots, N$.

2.4 Diskontinuierliche Galerkin-Methode

2.5 Tensor Dekomposition

Definition 2.11. *Rang Eins Tensor*
Ein Tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ ist von Rang Eins wenn es als äußeres Produkt von N Vektoren

$$\mathcal{X} = a^{(1)} \circ \ldots a^{(N)}$$

geschrieben werden kann.

Bemerkung 2.12. *Symmetrie*

- *Ein Tensor nennt man kubisch genau dann wenn jeder Mode dieselbe Dimension hat.*
- *Einen kubischen Tensor nennt man supersymmetrisch genau dann wenn die Elemente des Tensors konstant bleiben unter jeglicher Permutation der Indizes*
- *Ein Tensor kann stückweise symmetrisch sein wenn die Elemente konstant bleiben unter der Permutation von mindestens 2 Indizes.*

Definition 2.13. *Diagonal*
Einen Tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ nennt man diagonal, wenn $x_{i_1,\ldots,i_N} \neq 0$ genau dann wenn $i_1 = \cdots = i_N$.

Bemerkung 2.14. *Entfaltung*
Einen Tenspr kann man entfalten. Dies impliziert eine Neu Ordnung der Tensorelemnte in eine Matrix. Wir betrachten nur die sogenannte mode-n Entfaltung, da dies die einzig relevante Form der Entfaltung ist. Eine mode-n Entfaltung eines Tensors $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ wird mit $\mathbf{X}_{(n)}$ geschrieben und ordnet die mode-n fibers in die Spalten der Ergebnismatrix.

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 $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_n}$. Die Formel für die Pseudoinverse lautet:

$$\mathcal{A}^{\dagger} = \mathcal{S}^{\dagger} \times_{n=1}^{N} U^{(n)^T} \tag{2.13}$$

Wobei $\mathcal{S} \in \mathbb{R}^{I_1 \times \cdots \times I_n}$ und $U^{(n)} \in \mathbb{R}^{J_n \times I_n}$ Man kann 2.13 nach [TK09, 462] äquivalent umformen zu

$$\begin{aligned} & \mathcal{A}^{\dagger}_{(n)} = U^{(n)^T} \mathcal{S}^{\dagger}_{(n)} (U^{(N)^T} \otimes \cdots \otimes U^{(n+1)^T} \otimes U^{(n-1)^T} \otimes \cdots \otimes U^{(1)^T})^T \\ \iff & \mathcal{A}^{\dagger}_{(n)} = U^{(n)^T} \mathcal{S}^{\dagger}_{(n)} (U^{(N)} \otimes \cdots \otimes U^{(n+1)} \otimes U^{(n-1)} \otimes \cdots \otimes U^{(1)}) \end{aligned} \tag{2.14}$$

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

$$\mathcal{A}^{\dagger}_{(n)} v = U^{(n)^T} \mathcal{S}^{\dagger}_{(n)} (U^{(N)} \otimes \cdots \otimes U^{(n+1)} \otimes U^{(n-1)} \otimes \cdots \otimes U^{(1)}) v \tag{2.15}$$

Wir schauen uns die Struktur mal für den Fall, dass $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_4}$. Das heißt 2.15 reduziert sich auf:

$$\mathcal{A}^{\dagger}_{(n)} v = U^{(n)^T} \mathcal{S}^{\dagger}_{(n)} (U^{(N_1)} \otimes U^{(N_2)} \otimes U^{(N_3)}) v \tag{2.16}$$

mit $N_i \neq n$.

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

$$z := \begin{pmatrix} c_{11}b_{11}A & \dots & c_{11}b_{1n}A & \dots & \dots & c_{1n}b_{11}A & \dots & c_{1n}b_{1n}A \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ c_{11}b_{n1}A & \dots & c_{11}b_{nn}A & \dots & \dots & c_{1n}b_{n1}A & \dots & c_{1n}b_{nn}A \\ c_{21}b_{n1}A & \dots & c_{21}b_{nn}A & \dots & \dots & c_{2n}b_{n1}A & \dots & c_{2n}b_{nn}A \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ c_{n1}b_{n1}A & \dots & c_{n1}b_{nn}A & \dots & \dots & c_{nn}b_{n1}A & \dots & c_{nn}b_{nn}A \end{pmatrix} * v \quad (2.17)$$

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

$$z = \begin{pmatrix} c_{11}b_{11}A & \dots & c_{11}b_{1n}A & \dots & \dots & c_{1n}b_{11}A & \dots & c_{1n}b_{1n}A \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ c_{11}b_{n1}A & \dots & c_{11}b_{nn}A & \dots & \dots & c_{1n}b_{n1}A & \dots & c_{1n}b_{nn}A \\ c_{21}b_{n1}A & \dots & c_{21}b_{nn}A & \dots & \dots & c_{2n}b_{n1}A & \dots & c_{2n}b_{nn}A \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ c_{n1}b_{n1}A & \dots & c_{n1}b_{nn}A & \dots & \dots & c_{nn}b_{n1}A & \dots & c_{nn}b_{nn}A \end{pmatrix} * v \quad (2.18)$$

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 $\mathcal{V} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. 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.

$$\begin{aligned} z_1 = \mathcal{V}(1,1,1)c_{11}b_{11}a_{11} + \dots + \mathcal{V}(1,1,n)c_{11}b_{11}a_{1n} + \dots + \mathcal{V}(1,n,1)c_{11}b_{1n}a_{11} \\ + \dots + \mathcal{V}(n,1,1)c_{1n}b_{11}a_{11} + \dots + \mathcal{V}(n,n,n)c_{1n}b_{1n}a_{1n} \end{aligned} \quad (2.19)$$

Definiere $w_1(i,j) := \mathcal{V}(i,j,1)a_{11} + \dots + \mathcal{V}(i,j,n)a_{1n}$. 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 b_{ij} zu nutze. Definiere hierfür $\mathfrak{w}_{1,k}(i) := w_k(i,1)b_{11} + \dots + w_k(i,n)b_{1n}$. Damit erhalten wir:

$$z_1 = \mathfrak{w}_{1,1}(1)c_{11} + \cdots + \mathfrak{w}_{1,1}(n)c_{1n} \tag{2.20}$$

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

$$\mathcal{Z}(i,j,k) = \mathfrak{w}_{j,k}(1)c_{i1} + \cdots + \mathfrak{w}_{j,k}(n)c_{in} \tag{2.21}$$

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 2.18 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 Pseudoinverse für zell-basierte finiten Elemente Operatoren

3.1 Tensorprodukt Struktur

3.2 HOSVD

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.