



Technische Universität Berlin  
Institut für Mathematik

Masterarbeit  
im Studiengang Scientific Computing

# **Vergleich von Prolongations- und Restriktionsoperatoren für Deflationsmethoden zur Lösung von Gleichungssystemen**

Alessandro Bartsch <ale.bartsch@gmail.com>  
Betreut von Prof. Dr. Reinhard Nabben

Version vom  
14. November 2018

# Eidesstattliche Erklärung zur Masterarbeit

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

*Unterschrift :*

*Berlin, den*

## Zusammenfassung

Die nachfolgende Arbeit befasst sich mit dem Vergleich von Deflations- und Prolongationsoperatoren. Diese werden zur Vorkonditionierung von Projektionsmethoden zum Lösen von linearen Gleichungssystemen genutzt. Als Referenz für die Projektionsmethoden wird die GMRES-Methode verwendet. Die betrachteten Deflationsoperatoren sind zwei simple Algorithmen, welche unabhängig von der Form der Matrix Einträge auswählen, einen Algorithmus, der sich auf die Umordnung durch Grob- und Feingitterpunkte bezieht und einen Algorithmus, der die Matrix als Graph interpretiert um die Einträge durch die Nachbarn zu gruppieren.

# Inhaltsverzeichnis

Abbildungsverzeichnis	5
Tabellenverzeichnis	6
<b>1 Einleitung</b>	<b>7</b>
<b>2 Theoretische Grundlagen</b>	<b>9</b>
<b>3 Deflationsmethoden</b>	<b>12</b>
3.1 diagonale Restriktion . . . . .	12
3.2 gewichtete diagonale Restriktion . . . . .	12
3.3 häufungsbasierte Restriktion . . . . .	13
3.4 reduktionsbasierende Restriktion . . . . .	14
<b>4 Verfahren</b>	<b>15</b>
<b>5 Auswertung</b>	<b>17</b>
5.1 Diagonale Restriktion . . . . .	18
5.2 Gewichtete diagonale Restriktion . . . . .	21
5.3 Häufungsbasierte Restriktion . . . . .	24
5.4 Reduktionsbasierende Restriktion . . . . .	28
5.5 Vergleich simpler Restriktionen . . . . .	32
5.6 Vergleich komplexer Restriktionen . . . . .	32
5.7 Vergleich aller Restriktionen . . . . .	32
5.8 Abweichungen zu dem nicht geglätteten Vorkonditionierer . . . . .	32
<b>6 Schlussbemerkung</b>	<b>33</b>
Literaturverzeichnis	34

## Abbildungsverzeichnis

1	Konvergenz der Matrix JGL009 mit der diagonalen Restriktion . . . . .	19
2	Konvergenz der Matrix NOS7 mit der diagonalen Restriktion . . . . .	19
3	Konvergenz der Matrix BCSPWR09 mit der diagonalen Restriktion . .	20
4	Konvergenz der Matrix SAYLR3 mit der diagonalen Restriktion . . . .	20
5	Konvergenz der Matrizen BCSSTK25 und GEMAT11 (v.l.) mit der diagonalen gewichteten Restriktion . . . . .	22
6	Konvergenz der Matrix BCSPWR07 mit der diagonalen gewichteten Restriktion . . . . .	23
7	Konvergenz der Matrix GR3030 mit der diagonalen gewichteten Restriktion . . . . .	23
8	Konvergenz der Matrix BCSPWR07 mit der häufungsbasierten Restriktion	25
9	Konvergenz der Matrix NOS7 mit der häufungsbasierten Restriktion . .	26
10	Konvergenz der Matrix BCSSTK25 mit der reduktionsbasierten Restriktion . . . . .	29
11	Konvergenz der Matrix WATT1 mit der reduktionsbasierten Restriktion	30
12	Konvergenz der Matrizen DWT234 und DWT992 (v.l.) mit der reduktionsbasierten Restriktion . . . . .	30

## Tabellenverzeichnis

1	Testmatrizen aus der HARWELL-BOEING-Kollektion . . . . .	17
2	Übersicht zur Konvergenz für die diagonale Restriktion . . . . .	18
3	Übersicht zu Zeit und Speicher für die diagonale Restriktion . . . . .	21
4	Übersicht zur Konvergenz für die diagonale gewichtete Restriktion . . .	22
5	Übersicht zu Zeit und Speicher für die diagonale gewichtete Restriktion	24
6	Übersicht zur Konvergenz für die häufungsbasierte Restriktion . . . . .	25
7	Zeitliche Übersicht zur Erstellung der Restriktionsmatrizen für die häufungsbasierte Restriktion . . . . .	27
8	Übersicht zu Zeit und Speicher für das Lösen mit der häufungsbasierten Restriktion . . . . .	28
9	Übersicht zur Konvergenz für die reduktionsbasierte Restriktion . . . .	29
10	Zeitliche Übersicht zur Ermittlungszeit für die reduktionsbasierte Restriktion . . . . .	31
11	Zeitliche Übersicht zur Konvergenz für die reduktionsbasierte Restriktion	32

# 1 Einleitung

Eines der am längsten betrachteten Probleme in der Mathematik ist das Lösen des linearen Gleichungssystems

$$Ax = b, \quad A \in \mathbb{C}^{n \times n}, \quad x, b \in \mathbb{C}^n. \quad (1)$$

Häufig verbergen sich hinter  $A$  sehr große Matrizen, welche die Kapazitäten vieler Rechner übersteigen. Deswegen versucht man mittels eines Operators das Gleichungssystem auf ein Kleineres zu skalieren. Die Dimension  $r \ll n$  des Zielsystems orientiert sich häufig an  $\frac{n}{2}$ .

Der gesuchte Operator bezieht sich jedoch auf sogenannte Prolongations- bzw. Restriktionsmatrizen  $P \in \mathbb{C}^{n \times r}$  und  $R \in \mathbb{C}^{r \times n}$ .

In vielen Fällen ist die Restriktionsmatrix jedoch nur die Transponierte der Prolongationsmatrix, also

$$P = R^T.$$

Für die Wahl solcher Matrizen wären am Besten die Eigenvektoren von  $A$  als Spalten geeignet. Die Berechnung dieser ist in der Regel zu zeitaufwändig und somit keine praktikable Lösung. Deswegen wurden Algorithmen entwickelt um geeignete Matrizen zu konstruieren.

Für die Ermittlung der Matrizen kann man den Operator

$$P_N := I - AP(RAP)^{-1}R + P(RAP)^{-1}R$$

konstruieren. Mit diesem wird die Gleichung 1 überführt in

$$P_N Ax = P_N b. \quad (2)$$

Der Vorteil dieser Gleichung besteht darin, dass die linke Seite stets nicht singulär ist. Für die Lösung der Gleichung 2 nutzen wir den GMRES. Hier wird der Operator  $P_N$  als Vorkonditionierer verwendet.

In dieser Arbeit werden die theoretischen Grundlagen wiederholt und die einzelnen Methoden zur Ermittlung der Restriktionsmatrizen, sowie das Verfahren zum Lösen der Gleichung (2) und deren Auswertungen vorgestellt. Zu Beginn orientiert sich die Arbeit für die Grundlagen stark an dem Paper von NABBEN<sup>1</sup> [Nab17] und dem Paper von NABBEN, GARCÍAS<sup>2</sup> und KEHL<sup>3</sup> [NKR17], da die Erarbeitung der theoretischen Grundlagen nicht der Hauptbestandteil dieser Arbeit seien soll.

---

<sup>1</sup>Er beschäftigt sich hauptsächlich mit der numerischen Mathematik an der TU Berlin.

<sup>2</sup>Er forscht in der angewandten und numerischen Mathematik

<sup>3</sup>Sein Forschungsgebiet ist vor allem die numerische Mathematik

Anschließend werden die vier im Zuge dieser Arbeit verwendeten Algorithmen zur Restriktion bzw. Prolongation vorgestellt. Durch die simple Struktur der ersten beiden Algorithmen ist hier das Hauptaugenmerk auf die zwei komplexeren Algorithmen gelegt. Die genauen Definitionen sind dem Artikel [EN09] und dem Paper [GN12] entnommen.

Im nächsten Kapitel wird, das im Zuge dieser Arbeit entwickelte, Programm vorgestellt, welches durch einen Pseudocode verdeutlicht ist und auch im Anhang zu finden ist.

Das folgende und wohl auch wichtigste Kapitel dieser Arbeit bezieht sich auf die Auswertung des Programms. Hier wird unter anderem der Fehler auf die Iteration aufgetragen und die Geschwindigkeit gemessen. Anhand dieser Werte wird ein ausführlicher Vergleich aller vier Algorithmen vorgenommen. Für die Testdurchläufe wurden ausgewählte Matrizen aus dem Matrix-Markt [mm17] und Referenzprobleme aus anderen Arbeiten verwendet.

Zuletzt erfolgt eine Abschlussbemerkung. Diese umfasst eine kurze Zusammenfassung der ausgewerteten Daten, eine Wertung Dieser und ein Fazit.



## 2 Theoretische Grundlagen

Der Ansatz mittels Deflation ein Gleichungssystem zu lösen ist als erstes in den späten achtziger Jahren aufgetreten und wurde zum Lösen von linearen Gleichungssystemen mit symmetrischen Matrizen verwendet. Diese Methode wurde zu einer neuen Alternative für das schnelle Lösen eines Gleichungssystems mittels KRYLOV-Unterräumen. Man betrachtet auch hier das allgemeine Gleichungssystem (1) und möchte dieses mittels Restriktion in ein singuläres lineares Gleichungssystem mit geringerer Dimension überführen. Das Ziel ist es ein für die KRYLOV-Unterräume geeigneteres Spektrum zu erreichen. Wie in der Einleitung bereits erwähnt werden die erzeugten Deflations-Unterräume idealerweise durch die Eigenvektoren von  $A$  aufgespannt. In der Praxis verwendet man jedoch häufig Approximationen dieser Eigenvektoren.

Im Allgemeinen werden Projektionen eingesetzt um das initiale Gleichungssystem zu überführen. In Anlehnung an [NKR17] werden die Grundzüge der allgemeinen Deflation aufgezeigt.

### Definition 2.1.

Eine Matrix  $P \in \mathbb{C}^{n \times n}$  ist eine Projektion, falls gilt

$$P = P^2.$$

Der  $\mathbb{C}^n$  kann also als direkte Summe der Range ( $\mathcal{R}(P)$ ) und des Null-Raum ( $\mathcal{N}(P)$ ) einer Projektion dargestellt werden. Damit lässt sich eine weitere Definition von Projektionen darstellen. Diese beruht auf eben zwei solchen Unterräumen von  $\mathbb{C}^n$ .

### Definition 2.2.

Für zwei Unterräume  $\mathcal{V}, \mathcal{W} \subseteq \mathbb{C}^n$  mit  $\mathcal{V} \oplus \mathcal{W} = \mathbb{C}^n$  ist der Operator  $P_{\mathcal{V}, \mathcal{W}}$  als eindeutige Projektion mit den Eigenschaften

$$\mathcal{R}(P) = \mathcal{V} \text{ und } \mathcal{N}(P) = \mathcal{W},$$

definiert. Die Projektion  $P_{\mathcal{V}, \mathcal{W}}$  wird Projektion auf  $\mathcal{V}$  entlang  $\mathcal{W}$  genannt.

Deflationen im Allgemeinen werden durch Projektionen der Form  $P_{\mathcal{Y}^\perp, A\mathcal{Z}}$  dargestellt. In vielen Fällen wird hier  $\mathcal{Y} = A\mathcal{Z}$  verwendet. Für die Wohldefiniertheit müssen die Unterräume  $\mathcal{Y}, \mathcal{Z}$  die Eigenschaften des folgenden Theorems erfüllen.

### Theorem 2.3.

Seien  $\mathcal{Y}, \mathcal{Z}$  zwei Unterräume von  $\mathbb{C}^n$  der Dimension  $r$ ,  $Y, Z$  zwei Matrizen, dessen Spalten die Basis der Räume  $\mathcal{Y}, \mathcal{Z}$  bilden und  $A$  eine nicht-singuläre Matrix aus dem Raum  $\mathbb{C}^{n \times n}$ . Dann sind die folgenden Aussagen äquivalent:

1.  $(C)^n = AZ \oplus \mathcal{Y}^\perp$

2.  $Y^H AZ \in (C)^{r \times r}$  ist nicht-singulär und die Projektion  $P_{AZ, \mathcal{Y}^\perp}$  kann durch

$$P_{AZ, \mathcal{Y}^\perp} = AZ(Y^H AZ)^{-1}Y^H,$$

dargestellt werden.

3.  $(C)^n = A^{-1}\mathcal{Y}^\perp \oplus \mathcal{Z}$

4.  $Y^H AZ \in (C)^{r \times r}$  ist nicht-singulär und die Projektion  $P_{A^{-1}\mathcal{Y}^\perp, \mathcal{Z}}$  kann durch

$$P_{A^{-1}\mathcal{Y}^\perp, \mathcal{Z}} = I - Z(Y^H AZ)^{-1}Y^H A,$$

dargestellt werden.

Damit folgt auch die Wohlefiniertheit der von der nicht-symmetrischen Deflation verwendeten Operatoren  $P_D := P_{\mathcal{Y}^\perp, AZ}$  und  $Q_D := P_{A^{-1}\mathcal{Y}^\perp, \mathcal{Z}}$ . Wählt man  $\mathcal{Y} = AZ$ , dann erhält man außerdem die Eigenschaft, dass  $Y^H AZ$  nicht singulär ist, wenn  $A$  nicht singulär ist.

Mit diesen Operatoren kann nun der Deflationsprozess beschrieben werden. Um das Gleichungssystem  $Ax = b$  zu lösen wird zunächst das System

$$P_D Ax = P_D b,$$

gelöst. Die Lösung  $x'$  dieses Systems kann nun verwendet werden um das eigentliche System zu lösen. Durch Umformungen erhält man:

$$A(Q_D x' + Qb) = b.$$

Damit ergibt sich, dass die Lösung des ursprünglichen Gleichungssystem gleich  $Q_D x' + Qb$  ist, also  $x = Q_D x' + Qb$ . Man kann diesen Prozess, so wie das in den hier aufgezogenen Versuchen praktiziert wurde, um einen Glätter  $M^{-1} \approx A^{-1}$  erweitern.

Der folgende und zentrale Satz stellt sicher, dass der GMRES auch für das vorkonditionierte projizierte System

$$M^{-1}P_D Ax = M^{-1}P_D b,$$

konvergiert und die Lösung auch zu der gewünschten Lösung des Ursprungssystems führt.

#### Theorem 2.4.

Seien  $A, M \in \mathbb{C}^{n \times n}$  nicht-singulär. Die Räume  $\mathcal{Z}$  und  $\mathcal{Y}$  sind Unterräume des  $\mathbb{C}^n$  und die Matrizen  $Z$  und  $Y$  sind Basen von  $\mathcal{Z}$  und  $\mathcal{Y}$ . Falls gilt

$$AZ \oplus \mathcal{Y}^\perp = MZ \oplus \mathcal{Y}^\perp = \mathbb{C}^n,$$

dann folgt

1. Die Projektionen  $P_D$  und  $Q_D$  sind wohldefiniert.
2. Der GMRES konvergiert für Startvektor bei dem Gleichungssystem

$$M^{-1}P_D Ax = M^{-1}P_D b. \quad (3)$$

3. Falls  $x'_i$  die approximierte Lösung des GMRES auf die Gleichung (3) ist, dann gelten für

$$x_i = Q_D x'_i + Qb,$$

folgende Eigenschaften:

- Falls  $r_i := b - Ax_i$  und  $r'_i = P_D(b - Ax'_i)$ , dann gilt  $r_i = r'_i$ . Die Fehler sind also gleich. Damit folgt, dass spätestens  $X_n = x$  gelten muss.
- $P_D r'_i = r'_i$
- $P_D r_i = r_i$

Damit haben wir die nötigen Grundlagen für diese Arbeit geschaffen, nämlich die Wohldefiniertheit und die Konvergenz des GMRES-Verfahrens für die Vorkoditionierung mittels Deflations-Methoden.

## 3 Deflationsmethoden

In diesem Kapitel werden die vier verschiedenen Methoden aufgezeigt, die zur Konstruktion der Prolongations- und Restriktionsmatrizen verwendet werden.

### 3.1 diagonale Restriktion

Diese Methode ist simpelste und auch schnellste der vier vorgestellten Methoden. Sie ist vollkommen unabhängig von den Einträgen der ursprünglichen Matrix  $A$ .

Zur Erstellung benötigt man nur die Dimension  $n$  und erstellt mit Dieser eine Matrix deren Einträgen zunächst alle null betragen. Dann geht man jede Zeile  $i$  durch und setzt den Eintrag in der Spalte  $j = 2i - 1$  gleich eins.

Man erhält für ein gerades  $n$  die Form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}.$$

Und für ein ungerades  $n$  erhält man die Form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 \end{bmatrix}.$$

Diese Restriktion beschränkt sich demzufolge auf jeden zweiten Eintrag der Ursprungsmatrix  $A$ .

### 3.2 gewichtete diagonale Restriktion

Die zweite Restriktion ist, genau wie die erste, ebenfalls unabhängig von den Einträgen der vorliegenden Matrix  $A$ . Zum Aufbau wird ebenso zunächst eine Matrix mit den Einträgen null erstellt um dann in jeder Zeile  $i$  in den Spalten  $2i - 1$ -ten und  $2i + 1$ -ten den Wert 0,25 und in der Spalte  $2i$ -ten den Wert 0,5 zu setzen.

Man erhält also Matrizen der Form:

$$\begin{bmatrix} 0,25 & 0,5 & 0,25 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0,25 & 0,5 & 0,25 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0,25 & 0,5 & 0,25 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0,25 & 0,5 & 0,25 \end{bmatrix}.$$

Man erkennt, dass hier eine einfache Gewichtung der Einträge aus  $A$  vorliegt.

### 3.3 häufungsbasierte Restriktion

Der Algorithmus basiert auf dem *Algorithmus 4* aus dem Paper [EN09].

Die Matrix  $A$  wird hier als Graph interpretiert und auf ihre Häufungspunkte analysiert.

#### Definition 3.3.1.

Für eine dünn-besetzte, nicht-symmetrische Matrix  $A = [a_{ij}] \in \mathbb{R}^{n \times n}$  sei  $\mathcal{G}_A(\mathcal{U}, \mathcal{E})$  der Graph. Dieser besteht aus den  $n$  Knoten  $\mathcal{U} = \{u_1, u_2, \dots, u_n\}$  und den Kanten  $\mathcal{E}$ , sodass alle Kanten  $e_{ij}$  existieren. Eine Kante  $e_{ij}$  verbindet zwei Knoten  $u_i$  und  $u_j$  genau dann, wenn  $a_{ij} \neq 0$  und  $i \neq j$  gilt.

Es wird versucht eine Vergrößerung mit einem Faktor von  $\frac{1}{4}$  zu erreichen. Dabei konzentriert man sich zunächst auf die nahen Nachbarn, der Knoten.

#### Definition 3.3.2.

Die nahen Nachbarn  $\mathcal{N}_i^C$  eines Knoten  $u_i$  sind durch

$$\mathcal{N}_i^C := \{u_j \in \mathcal{U} \mid e_{ij} \in \mathcal{E}\},$$

definiert. Die Untermenge der  $\tau$ -starken nahen Nachbarn ist definiert durch

$$\mathcal{N}_i^{C,\tau} := \left\{ u_j \in \mathcal{N}_i^C \mid \sqrt{\frac{a_{ij}^2}{|a_{ii}a_{jj}|}} \geq \tau \right\}.$$

Damit erreicht man zumeist eine Gruppierung von maximal drei Knotenpunkten. Man erreicht dementsprechend eine  $\frac{1}{3}$ -Vergrößerung, die noch nicht genügt. Es sollte jeweils ein weiterer Knoten zu den Gruppen gefügt werden. Hierfür wird der distante Nachbar einbezogen.

#### Definition 3.3.3.

Ein Knoten  $u_k$  ist in der Gruppe der distanten Nachbarn  $\mathcal{N}_i^D$  von  $u_i$ , falls dieser ein gemeinsamer naher Nachbar der nahen Nachbarn  $u_j \in \mathcal{N}_i^{C,\tau}$ . Die  $\tau$ -starken distanten Nachbarn sind eine Untergruppe der distanten Nachbarn, welche zusätzlich die  $\tau$ -Bedingung aus der Definition der nahen Nachbarn erfüllen.

Mit dieser Einteilung kann man die einzelnen Knotenpunkte durchgehen und durch ihre Nachbarn gruppieren um eine gewünschte Vergrößerung zu erzielen. Mit dieser Einteilung schafft man sich disjunkte Untermengen  $\Omega_q$  mit den Indexsets  $\mathcal{I}_q$ . Mit diesen kann die Restriktionsmatrix  $R$  mit den Einträgen  $r_{iq}$  konstruiert werden:

$$r_{iq} = \begin{cases} 1, & i \in \mathcal{I}_q, \\ 0, & i \notin \mathcal{I}_q. \end{cases}$$

### 3.4 reduktionsbasierende Restriktion

Zum Aufbau der Reduktionsmatrix muss zunächst die Matrix  $A$  zunächst umsortiert werden, sodass die Form

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

entsteht. Hierbei muss die Matrix  $A_{11} \in \mathbb{R}^{n_1 \times n_1}$  alle Feingitterpunkte und  $A_{22} \in \mathbb{R}^{n_2 \times n_2}$  alle Grobgitterpunkte der Matrix  $A$  enthalten. Diese Umsortierung erreichen wir durch den "Greedy Coarser" von SCOTT MACLACHLAN aus dem Anhang des Paper [GN12]. Nimmt man den  $F$ -Glätter<sup>4</sup>, so wäre die optimale Interpolation die Matrix

$$R_{opt} := \begin{bmatrix} -A_{11}^{-1} A_{12} \\ I \end{bmatrix}.$$

Die Berechnung der Inversen  $-A_{11}^{-1}$  ist jedoch wiederum sehr rechenintensiv. Deswegen ersetzt man  $A_{11}$  durch eine Approximation  $D \approx A_{11}$ , welche leicht zu invertieren ist und in einer dünn besetzten Inversen resultiert.

Für diese Arbeit wurde  $D$  als Diagonalmatrix von  $A_{11}$  gewählt:

$$D = \text{diag}(A_{11}).$$

Somit erhält man als Reduktionsmatrix

$$R = \begin{bmatrix} -\text{diag}(A_{11})^{-1} A_{12} \\ I \end{bmatrix}.$$

---

<sup>4</sup>Der  $F$ -Glätter arbeitet nur auf dem  $\mathbb{R}^{n_1}$

## 4 Verfahren

Um die Gleichungssysteme zu lösen wird zunächst eine Restriktionsmatrix  $R$  benötigt. Hierzu wird eine der in Kapitel 3 vorgestellten Methoden benutzt.

Für den Vorkonditionierer definiert man den Prolongationsoperator  $P := R^T$ . Mit Hilfe dieser Operatoren kann man nun das erste transformierte Gleichungssystem aufstellen.

$$P_D A \tilde{x} = P_D b,$$

mit dem Operator

$$P_D := I - AP(RAP)^{-1}R.$$

Um eine optimale Lösung zu erhalten wird dieser Operator verschoben um  $Q := P(RAP)^{-1}R$ . Man erhält den aus der Literatur bekannten Operator

$$P_N := P_D + Q, \tag{4}$$

$$= I - AQ + Q. \tag{5}$$

Mit diesem Operator wird die Hälfte der Berechnungen durchgeführt.

Als weitere Variante glättet man den ersten Teil des Operators  $P_N$  um  $M = \text{diag}(A)$  und erhält den Operator

$$P'_N := M^{-1}P_D + Q. \tag{6}$$

Diese beiden Operatoren werden zum Lösen des Gleichungssystems als Vorkonditionierer in den neustartenden GMRES übergeben.

Da zum Teil sehr kleine Zahlen in den Testmatrizen zu finden sind, kann es vorkommen, dass die Inverse von  $RAP$  nicht berechnet werden konnte. War dies der Fall, so wurde hierfür die Pseudo-Inverse der entsprechenden Matrix verwendet. Der Algorithmus verwendet den GMRES aus dem Julia-internen "LinearAlgebra"-Paket. Er wurde so konfiguriert, dass die Toleranz bei  $10^{-22}$  liegt und die maximale Iterationszahl mindestens 10000 beträgt. In der Standardkonfiguration sind bei diesem Algorithmus automatische Restarts konfiguriert. Diese wurden für genauere Ergebnisse auskonfiguriert. Dadurch erhält man natürlich eine deutlich höheren Speicherbelegung.

**Data:** Matrix  $A \in \mathbb{R}^{n \times n}$ , Array  $b \in \mathbb{R}^{n \times 1}$ , Restriktionsmatrix  $R \in \mathbb{R}^{n \times m}$  mit  
 $m \ll n$ , Boolean  $useM3$

**Result:** Approximierte Lösung  $x$ , Historie  $h$

$n = \text{size}(A, 1);$

$P = R^T;$

**try:**

|  $Q = P(RAP)^{-1}R$

**catch SingularException:**

|  $Q = P * \text{pseudoinverse}(RAP) * R$

**end**

*/\* Ermittle den Vorkonditionierer*

*\*/*

**if**  $useM3$  **then**

**try:**

        |  $InvDiagA = \text{diag}(A)^{-1}$

**catch SingularException:**

        |  $InvDiagA = \text{pseudoinverse}(\text{diag}(A))$

**end**

$P_N = InvDiagA(I - AQ) + Q$

**else**

    |  $P_N = I - AQ + Q$

**end**

*/\* maximale Iteration auf  $\max(10000, 2n)$  und Toleranz auf  $10^{-22}$*

*\*/*

return  $\text{gmres}(A, b, P_N)$

**Algorithm 1:** Algorithmus zum Lösen des Gleichungssystems  $Ax = b$  mit GMRES



## 5 Auswertung

In diesem Kapitel erfolgt die Auswertung, des in Kapitel 4 vorgestellten Verfahrens. Zunächst wird jeder Algorithmus aus Kapitel 3 einzeln untersucht. Danach erfolgen dann die Vergleiche der Algorithmen untereinander. Abschließend werden alle Ergebnisse mit denen verglichen, welche mit dem Vorkonditionierer entstanden, der keine Vorglättung erhielt.

Alle Versuche werden auf dem gleichen Computer ausgeführt. Dieser ist mit dem Betriebssystem "Windows 10" ausgestattet und wird 16 RAM und dem Prozessor "Intel® Core™ i7-4771 CPU @ 3,5GHz" angetrieben.

Das Programm wurde vollständig in Julia<sup>5</sup> v.1.0.1 geschrieben und verwendet ausschließlich Matrizen aus dem Matrix-Market ([mm17]).

Im folgenden werden die Auswertungen mit dem vorkonditionierten Operator (6) durchgeführt. Falls es zu großen Abweichungen zu dem in der Literatur häufiger zu findenden Operator (4) vorkommt, so werden diese am Ende dieses Kapitels vorgestellt.

Für die Auswertungen werden Matrizen aus der HARWELL-BOEING-Kollektion verwendet, da man hier ein breites Spektrum finden kann. Die Spanne reicht von kleinen Matrizen, welche nur zur Widerlegung bestimmter Thesen für Sparse-Matrizen konstruiert wurden, bis hin zu großen Matrizen, die man direkt aus realen Testfällen abgeleitet hat.

Matrix	Größe	Konditionszahl
BCSSTK25	15439 x 15439	65
GEMAT11	4929 x 4929	3.74e+08
BCSSTK24	3562 x 3562	65
BCSSTK23	3134 x 3134	6.9e+12
BCSSTM26	1922 x 1922	2.6e+05
WATT1	1856 x 1856	5.38e+09
BCSPWR09	1723 x 1723	-
BCSPWR07	1612 x 1612	-
SAYLR3	1000 x 1000	1e+02
DWT992	992 x 992	-
GR3030	900 x 900	3.8e+02
NOS7	729 x 729	4.1e+09
NNC666	666 x 666	1.8e+11
MBEAUSE	496 x 496	-
DWT234	234 x 234	-
JGL009	9 x 9	-

Tabelle 1: Testmatrizen aus der HARWELL-BOEING-Kollektion

---

<sup>5</sup>siehe <https://julialang.org/>

## 5.1 Diagonale Restriktion

Dieses Kapitel befasst sich mit der Auswertung der im Kapitel 3.1 vorgestellten Restriktionsmethode. Da die Restriktionsmatrix einzig anhand der Dimension der Matrix  $A$  erstellt wird, werden komplexe Strukturen nicht beachtet. Es wird lediglich jeder zweite Eintrag der Matrix betrachtet und somit bezieht dieser Algorithmus am wenigstens Einträge der eigentlichen Matrix mit in die Berechnung des Vorkonditionierers mit ein. Im gröbsten können die Ergebnisse unterteilt werden in konvergiert und nicht-konvergiert.

nicht-konvergiert	konvergiert
GEMAT11	GR3030
BCSSTK25	JGL009
MBEAUSE	WATT1
NNC666	PCSPWR07
	PCSPWR09
	BCSSTK23
	BCSSTK24
	DWT234
	DWT992
	BCSSTM26
	NOS7
	SAYLR3

Tabelle 2: Übersicht zur Konvergenz für die diagonale Restriktion

Eine Eigenschaft, die drei der vier Matrizen auf der nicht-konvergenten Seite eint, ist die hohe Konditionszahl. Für die vierte Matrix liegen hier keine Informationen vor. Drei andere Matrizen dieser Seite werden dadurch geeint, dass man bei der Berechnung des Glätters für den Vorkonditionierer auf die Pseudoinverse ausweichen musste. Dies mag auch der Grund sein, weswegen das Residuum dieser Matrizen für den gesamten Durchlauf bei positiv Unendlich lag. Ausgenommen von dieser Eigenschaft ist die Matrix BCSSTK25. Aber auch bei anderen Matrizen kam es zur Berechnung der Inversen von  $A$  mittels Pseudoinversität, so zum Beispiel die Matrix JGL009. Hier konvergierte das Verfahren aber sehr überzeugend.

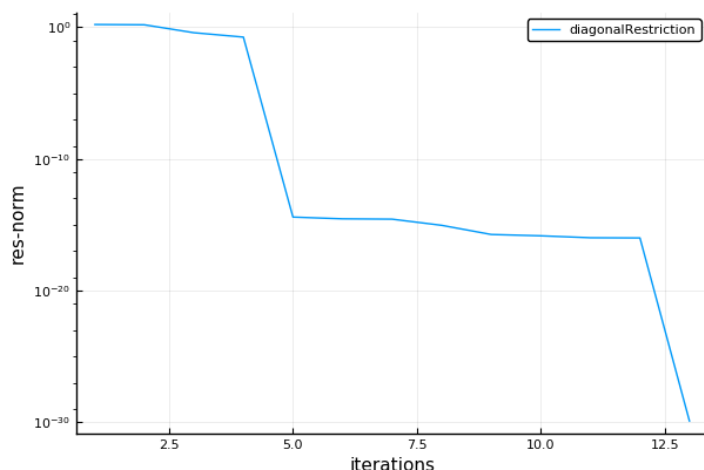


Abbildung 1: Konvergenz der Matrix JGL009 mit der diagonalen Restriktion

Dieser typische Verlauf der konvergenten Durchläufe sieht durch seine geringe Iterationszahl natürlich sehr glatt aus. Doch auch für Durchläufe mit deutlich höheren Iterationszahl ist das Muster wiederzuerkennen. In einer Anfangsphase, welche in einigen Fällen deutlich länger oder kürzer ausfällt, fällt das Residuum nur sehr schwach um dann abrupt in einen deutlich stärkeren Fall überzugen. Im Anschluss wird dieser Verlauf ein zweites mal aufgezeigt und dies auch mit einem abrupten Übergang in ein schwaches Gefälle.

Es gibt jedoch auch verschiedene Formen dieses Musters. So zum Beispiel erkennt man für die Matrix NOS7 eine höhere Wiederholungsrate des Musters und etwas abgeglättete Übergänge.

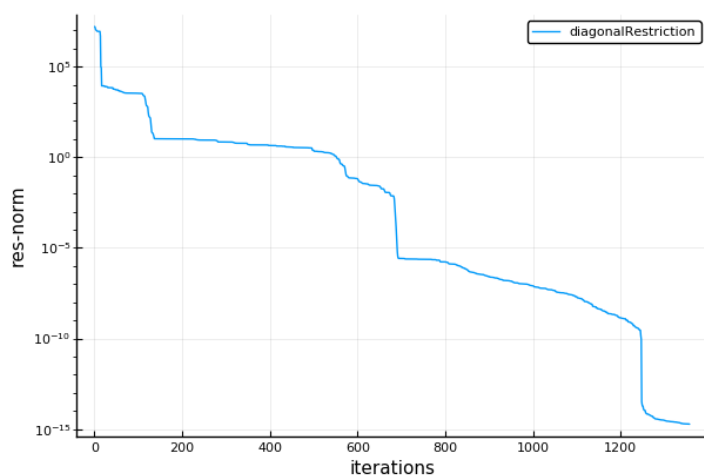


Abbildung 2: Konvergenz der Matrix NOS7 mit der diagonalen Restriktion

Die mit Abstand am meisten abgerundeten Übergänge sind jedoch bei den Matrizen der Reihe BCSPWR zu finden. Man kann auch hier deutlich das Muster des Durchlaufs mit der Matrix JGL009 wiedererkennen, obwohl die anfängliche Phase sich hier deutlich länger aufzeigt. Die Anzahl an Wechslen ist bei diesem Durchlauf aber identisch.

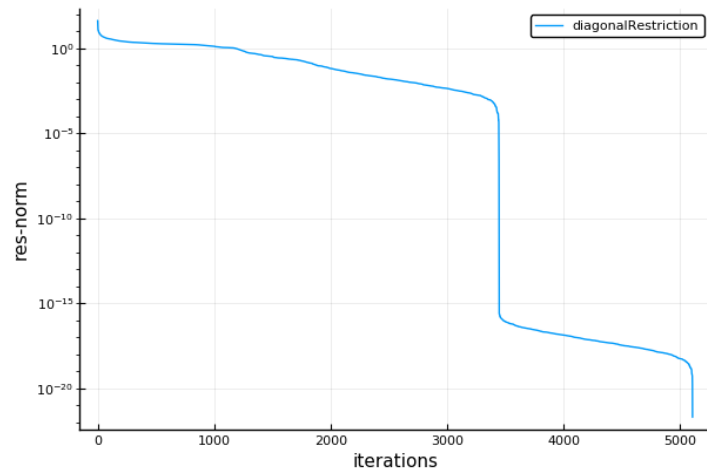


Abbildung 3: Konvergenz der Matrix BCSPWR09 mit der diagonalen Restriktion

Die Matrizen SAYLR3 und WATT1 weichen am meisten von dem Beschriebenen ab. Hier kann man fast einen teilweisen Übergang in eine Linearität erkennen. Im Detail sind dies aber nur sehr schnelle Wiederholungen des oben Beschriebenen Wechsels zwischen starkem und schwachen Fall des Residuums.

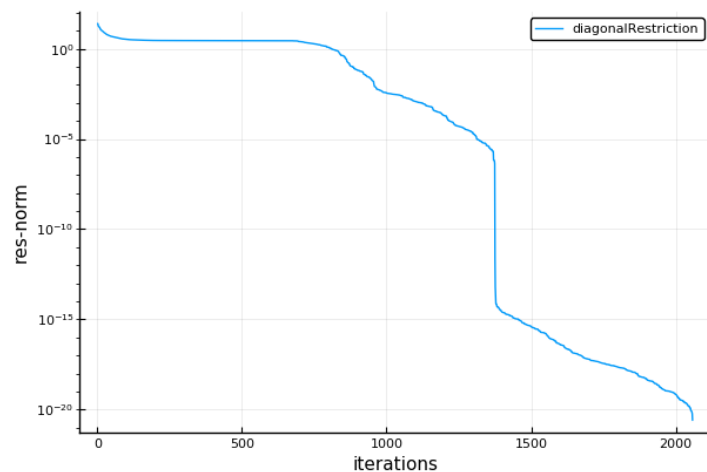


Abbildung 4: Konvergenz der Matrix SAYLR3 mit der diagonalen Restriktion

Schaut man sich die Laufzeit der Durchläufe an, so korreliert diese stark mit der Größe und Komplexität der Matrizen.

Matrix	Zeit	Speicher
BCSSTK25	22183s	45.3GB
GEMAT11	381s	5.21GB
BCSSTK24	141s	2.84GB
BCSSTK23	101s	2.41GB
NNC666	40.6s	3.07GB
BCSPWR09	33.0s	1.69GB
MBEAUSE	30.9s	3.04GB
BCSPWR07	26.7s	1.62GB
WATT1	17.9s	1.35GB
BCSSTM26	12.6s	1.59GB
SAYLR3	3.80s	0.97GB
GR3030	2.79s	0.93GB
DWT992	2.46s	0.94GB
NOS7	1.34s	892MB
JGL009	413ms	764MB
DWT234	365ms	791MB

Tabelle 3: Übersicht zu Zeit und Speicher für die diagonale Restriktion

Man kann deutlich erkennen, dass die Matrix BCCSTK25 die mit Abstand längste Zeit und den meisten Speicher benötigte. Die Tabelle spiegelt außerdem eine grobe Einteilung nach Größe wieder. So sind etwa die ersten drei Matrizen auch in der exakten Anordnung nach Größe. Durch die nicht-konvergenz fallen Matrizen wie die NNC666 aus dieser Reihenfolge heraus. Dies liegt auch daran, dass die maximale Iterationszahl erreicht wurde. Den Abschluss bildet wider Erwartung die Matrix DWT234, trotz der  $9 \times 9$  großen Matrix JGL009. Der verwendete Speicher korreliert sehr stark mit der Größe der Matrizen im Zusammenhang mit der Iterationszahl.

Die zeitliche Übersicht und der verbrauchte Speicher zur Erstellung der Restriktionsmatrix bedarf an dieser Stelle keiner Tabelle, da man hier aus der Theorie und in den Ergebnissen eine direkte Verbindung zwischen Größe der Matrix und dem Ergebnis deuten kann.

## 5.2 Gewichtete diagonale Restriktion

Die im Kapitel 3.2 vorgestellte Restriktion erzielt ähnliche Ergebnisse, wie die im vorherigen Kapitel Vorgestellten. Bei dieser Berechnung werden alle Einträge zur Berechnung des Vorkonditionierers mit Hilfe einer Gewichtung einbezogen. Es ist auch hier die Struktur der Matrix unerheblich und nimmt somit keinen Einfluss auf die Berechnung der Restriktion ein. Die Übersicht zur Konvergenz zeigt, dass es nur für drei Matrizen bei einer nicht-Konvergenz blieb.

nicht-konvergiert	konvergiert
GEMAT11	GR3030
BCSSTK25	JGL009
MBEAUSE	WATT1
	PCSPWR07
	PCSPWR09
	BCSSTK23
	BCSSTK24
	DWT234
	DWT992
	BCSSTM26
	NOS7
	SAYLR3
	NNC666

Tabelle 4: Übersicht zur Konvergenz für die diagonale gewichtete Restriktion

Für die Berechnung des Vorkonditionierers wurde für die Matrix MBEAUSE die Glättung durch die Pseudoinverse von  $A$  dargestellt. Daher wurde hier ein konstantes Residuum von positiv Unendlich verzeichnet. Die Matrizen BCSSTK25 und GEMAT11 weisen überdies eine hohe Konditionszahl auf. Für die Matrix MBEAUSE liegt leider kein Wert für die Konditionszahl vor. Schaut man sich den Verlauf von GEMAT11 und BCSSTK25 an, erkennt man, dass auch für diese Matrizen eine Konvergenz erreicht worden wäre, wenn man die maximale Iterationszahl erhöht hätte.

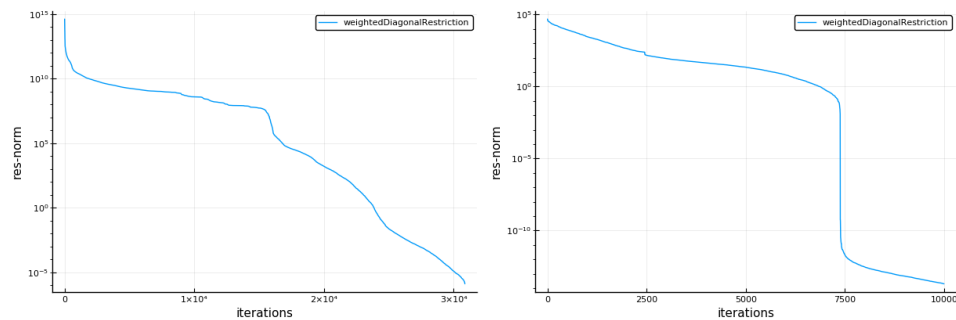


Abbildung 5: Konvergenz der Matrizen BCSSTK25 und GEMAT11 (v.l.) mit der diagonalen gewichteten Restriktion

Der Verlauf von BCSSTK25 ist sehr ähnlich zu den anderen Verläufen der BCSST-Matrizen. Die anderen Matrizen der BCSSTK-Reihe konvergierten sogar bei ähnlichen Residuums-Werten um  $10^{-5}$ . Dies ist relativ ungewöhnlich da die restlichen Matrizen alle bei Residuums-Werten von  $< 10^{-22}$  konvergierten, so wie es auch als relative Toleranz angegeben war. So zum Beispiel war dies der Fall bei der Matrix BCSPWR07.

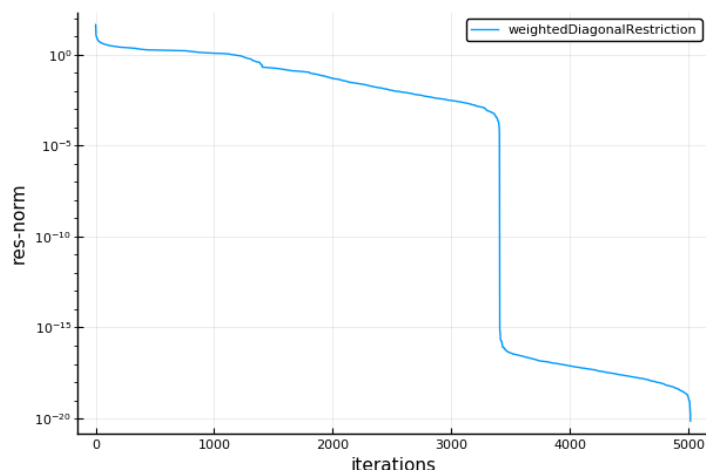


Abbildung 6: Konvergenz der Matrix BCSPWR07 mit der diagonalen gewichteten Restriktion

An diesem Verlauf ist auch sehr gut das typische Konvergenzverhalten zu erkennen. Es gibt eine beliebige Anzahl von Umbrüchen zwischen sehr schwachen und sehr starken Fall des Residuums. Ein Beispiel, welches dies in mehreren Facetten demonstriert ist der Verlauf für den Durchlauf der Matrix GR3030. Man hat hier einen zunächst sehr schnellen Wechsel zwischen den beiden Fallarten um dann in einen nahezu konstanten Verlauf überzugehen.

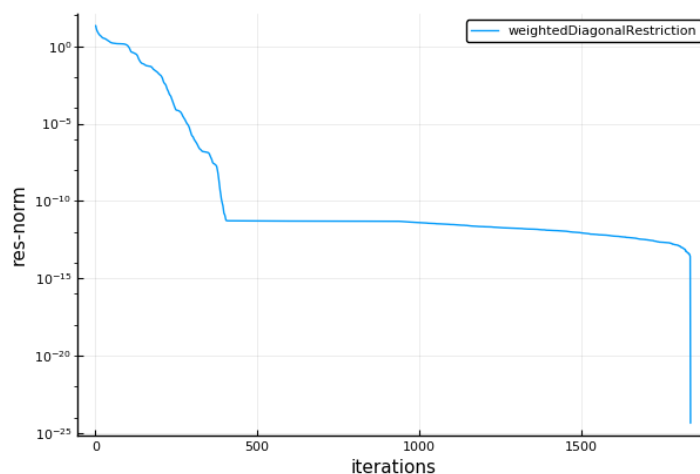


Abbildung 7: Konvergenz der Matrix GR3030 mit der diagonalen gewichteten Restriktion

Den Abschluss bildet jedoch wieder ein sehr starker Fall des Residuums um darin zu konvergieren. Diesen stetigen Wechsel am Anfang des Verlaufes kann man bei vielen anderen Durchläufen ebenfalls beobachten und er lässt den Graphen in einigen Fällen nahezu linear wirken.

Vergleicht man die Geschwindigkeiten der Matrizen, so erkennt man, dass umso größer die Matrix war, umso länger hat die Berechnung gebraucht. Daher entspricht die Anordnung nach Geschwindigkeit fast der Anordnung nach Größe der Matrizen.

Matrix	Zeit	Speicher
BCSSTK25	22188s	45.3GB
GEMAT11	365s	4.71GB
BCSSTK24	138s	2.84GB
BCSSTK23	91.5s	2.37GB
BCSSTM26	37.3s	1.45GB
BCSPWR09	33.7s	1.68GB
MBEAUSE	31.0s	3.04GB
BCSPWR07	27.7s	1.58GB
WATT1	17.9s	1.35GB
SAYLR3	3.90s	0.97GB
GR3030	2.79s	0.93GB
DWT992	2.46s	0.94GB
NOS7	1.71s	905MB
NNC666	1.01s	869MB
JGL009	401ms	764MB
DWT234	337ms	790MB

Tabelle 5: Übersicht zu Zeit und Speicher für die diagonale gewichtete Restriktion

Die Matrix BCSSTK25 benötigt eindeutig die meiste Zeit geschuldet durch ihre mit Abstand größten Dimension, der damit hohen maximalen Iterationszahl von über 30000 und dem Umstand, dass der Durchlauf nicht konvergierte. Etwas anders als eine Reihenfolge, welche durch die Dimension vorgegeben wäre, ordnet sich WATT1 ein, welche eine hohe Konditionszahl hat und eine größere Dimension als die Matrizen BCSPWR07 und BCSPWR07. Trotzdem ist sie erst nach diesen Matrizen zu finden. Außerdem ist die recht kleine Matrix MBEAUSE, geschuldet durch ihre nicht-Konvergenz, auch recht weit oben in der Tabelle zu finden. Und auch wenn die kleinste Matrix den geringsten Speicher in Anspruch genommen hat, so benötigte der Durchlauf für die deutlich größere Matrix DWT234 etwas weniger Zeit. Es handelt sich hier jedoch um einige Millisekunden.

Der Blick auf die Tabelle zur Berechnung der Restriktionsmatrix lohnt sich nicht, da man hier den direkten Zusammenhang zu der Dimension der Matrix sieht.

### 5.3 Häufungsbasierte Restriktion

In diesem Teilkapitel werden die Ergebnisse des geglätteten Vorkonditionierers mit der ermittelten Restriktion aus Kapitel 3.3 vorgestellt. Der Algorithmus ist einer der komplexeren in dieser Arbeit. Man kann erkennen, dass auch hier der Durchlauf für den überwiegenden Teil der Matrizen konvergierte.



nicht-konvergiert	konvergiert
GEMAT11	GR3030
BCSSTK25	JGL009
MBEAUSE	WATT1
NNC666	PCSPWR07
	PCSPWR09
	BCSSTK23
	BCSSTK24
	DWT234
	DWT992
	BCSSTM26
	NOS7
	SAYLR3

Tabelle 6: Übersicht zur Konvergenz für die häufungsbasierte Restriktion

Es fällt besonders auf, dass bei der Berechnung des Vorkonditionierers von drei der nicht-konvergenten Matrizen -alle außer BCSSTK25- es durch eine erhöhte Verwendung der Pseudoinversen zu Einträgen von positiv Unendlich kam. Dies scheint der Grund für ein konstantes Residuum von positiv Unendlich und damit auch für die nicht-Konvergenz. Einzig für die Matrix BCSSTK25 kann man erkennen, dass es vermutlich zu einer Konvergenz gekommen wäre, hätte man die maximale Iterationszahl erhöht. Der typische Verlauf des Residuums aus den vorangehenden Kapiteln findet sich auch bei der häufungsbasierten Restriktion wieder. So kann man zum Beispiel an der Matrix BCSPWR07 sehen, wie sich der starke und sehr schwache Fall des Residuums abrupt abwechseln.

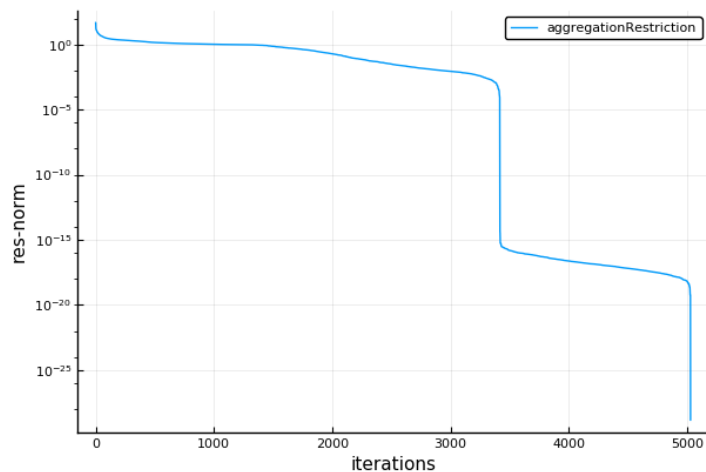


Abbildung 8: Konvergenz der Matrix BCSPWR07 mit der häufungsbasierten Restriktion

Bei genauem Betrachten fällt schnell auf, dass das Residuum fast aller Plots im letzten Iterationsschritt sehr gering ist. Bis auf die Matrizen BCSSTK23, BCSSTK24 und NO7 liegt das Residuum bei  $< 10^{-25}$  und zum Teil sogar  $< 10^{-30}$ . Die Verläufe der BCSSTK-Matrizen ähneln sich auch hier untereinander und denen aus

den vorherigen Kapiteln und für die Matrix NOS7 kann eine sehr schneller Wechsel der oben beschriebenen typischen Struktur der Graphen beobachtet werden.

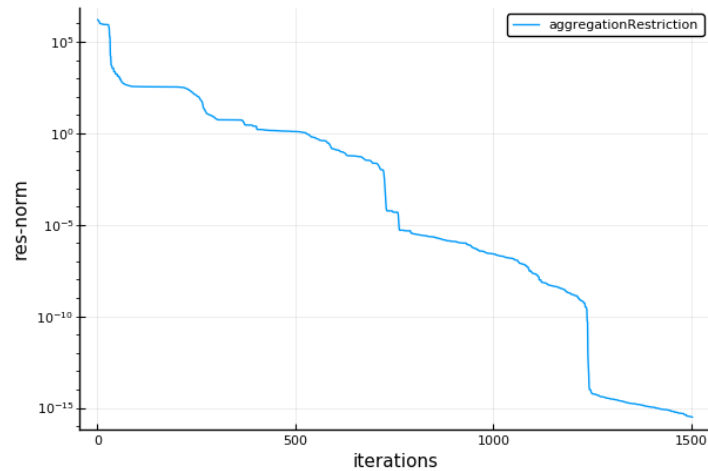


Abbildung 9: Konvergenz der Matrix NOS7 mit der häufungsbasierten Restriktion

Insgesamt kann man kein unerwartetes Verhalten in den Verläufen erkennen. Die Matrix BCSSTK25 ist mit hoher Wahrscheinlichkeit durch ihre hohe Dimension nicht in der angegebenen maximalen Iterationszahl konvergiert, aber der Verlauf lässt eine Konvergenz vermuten. Somit sind auch hier nur die fehlerhaften Berechnungen durch die Pseudoinversen Schuld an der nicht-Konvergenz einiger Matrizen.

Betrachtet man die Geschwindigkeit der Berechnungen, so muss man eine Unterscheidung machen. Durch den komplexeren Aufbau zu der Berechnung der Restriktion wird die Betrachtung unterteilt auf die Zeit die benötigt wurde um die Restriktionsmatrix zu berechnen und die, die benötigt wurde um das Gleichungssystem zu lösen bzw. bis zum letzten Iterationsschritt die Berechnungen durchzuführen.

Bei dem Vergleich zum Aufbau der Restriktionsmatrix fällt auf, dass die Zeit nicht direkt von der Größe der Matrizen abhängt.

Matrix	Zeit
BCSSTK25	65.0s
BCSSTK24	4.47s
BCSSTK23	2.92s
BCSSTM26	1.48s
WATT1	781ms
BCSPWR07	720ms
BCSPWR09	688ms
GEMAT11	561ms
GR3030	373ms
DWT234	300ms
SAYLR3	287ms
DWT992	246ms
NOS7	149ms
MBEAUSE	68.1ms
NNC666	49.8ms
JGL009	71.4 $\mu$ s

Tabelle 7: Zeitliche Übersicht zur Erstellung der Restriktionsmatrizen für die häufungs-basierte Restriktion

Man erkennt, dass für die Matrix BCSSTK25 die längste Zeit in Anspruch genommen werden musste und im Gegensatz zu ihr die Matrix JGL009 innerhalb weniger Mikrosekunden behandelt werden konnte. Dies liegt aber vor allem daran, dass diese beiden Matrizen in der Größenordnung einen wirklich großen Abstand haben.

Jedoch findet man die zweitgrößte Matrix, welche ungefähr ein Drittel der Dimension von BCSSTK25 besitzt, erst im mittleren Teil der Tabelle. Mehrere Matrizen mit weniger als der Hälfte der Dimension haben mehr als das doppelte der Zeit benötigt um die Restriktionsmatrix zu ermitteln. Und für die zweit-kleinste Matrix DWT234 werden nur wenige Millisekunden weniger zur Berechnung benötigt.

Betrachtet man hingegen die benötigte Zeit für die Berechnung der Lösung des Systems, so erhält man eine Anordnung, welche nahezu der Anordnung nach Größe entspricht.

Matrix	Zeit	Speicher
BCSSTK25	22217s	45.4GB
GEMAT11	375s	4.44GB
BCSSTK24	140s	2.88GB
BCSSTK23	107s	2.56GB
NNC666	40.2s	3.06GB
BCSPWR09	33.9s	1.68GB
MBEAUSE	30.8s	3.03GB
BCSPWR07	27.8s	1.58GB
WATT1	18.0s	1.36GB
BCSSTM26	9.40s	1.26GB
SAYLR3	4.15s	1.02GB
GR3030	2.85s	0.95GB
DWT992	2.18s	937MB
NOS7	1.60s	907MB
DWT234	310ms	791MB
JGL009	260ms	764MB

Tabelle 8: Übersicht zu Zeit und Speicher für das Lösen mit der häufungsbasierten Restriktion

Man kann erkennen, dass die größte Matrix auch die meiste Zeit in Anspruch genommen hat und die kleinste am wenigsten. Die Verschiebung der Matrizen MBEAUSE und NNC666 liegt an der nicht-Konvergenz und der damit verbundenen hohen Iterationszahl. Interessant ist, dass die Matrizen WATT1 und BCSSTM26 schneller als die deutlich kleinere Matrix BSCPWR07 waren. Auch mit der Speicherauslastung verhält es sich ähnlich. Man bekommt hier ein durch zum Teil nicht-konvergierte Durchläufe leichte Abwandlung der Anordnung nach der Dimension der Matrizen. Die Matrix mit einer deutlich höheren Konditionszahl und geringfügig niedrigeren Dimension NOS7 hat im Gegensatz zu GR3030 zeitlich und speicherplatztechnisch weniger als die Hälfte benötigt. Dies lässt darauf schließen, dass diese Variablen eher mit der Dimension als der Konditionszahl korrelieren.

## 5.4 Reduktionsbasierende Restriktion

Für den Algorithmus aus 3.4 hält sich die Übersicht der konvergierten und nicht-konvergierten Durchläufe ungefähr die Waage. Man kann feststellen, dass die Seite der Konvergenz deutlich überwiegt. Die Matrix BCSSTK25, welche eindeutig den höchsten Aufwand benötigte konvergierte ebenso. Besonders ist auf der Seite der nicht-Konvergenz, dass die Matrix SAYLR3 erscheint. Wie auch später in dem Vergleich mit dem ungeglätteten Vorkonditionierer zu sehen ist, liegt dies wohl an der reduktionsbasierten Methode. Wir erhalten hier ein konstantes Residuum von positiv Unendlich.

nicht-konvergiert	konvergiert
GEMAT11	GR3030
SAYLR3	JGL009
MBEAUSE	WATT1
NNC666	PCSPWR07
	PCSPWR09
	BCSSTK23
	BCSSTK24
	BCSSTK25
	DWT234
	DWT992
	BCSSTM26
	NOS7

Tabelle 9: Übersicht zur Konvergenz für die reduktionsbasierte Restriktion

Auch auffällig sind die wieder durch die Pseudoinverse gestörten Berechnungen für die restlichen drei Matrizen. Ebenso wie für die Matrix SAYLR3 konnte kein Plot erstellt werden, da man hier dauerhaft einen Wert von positiv Unendlich für das Residuum erhält.

Schaut man sich den Verlauf der Matrix BCSSTK25 an, so erkennt man, dass das Residuum wahrscheinlich auch weiterhin gefallen wäre, hätte man eine geringere Toleranz angesetzt. Es konvergierte jedoch unter Vorgaben bei einem Wert von  $\approx 10^{-7}$  in der Iteration 30251.

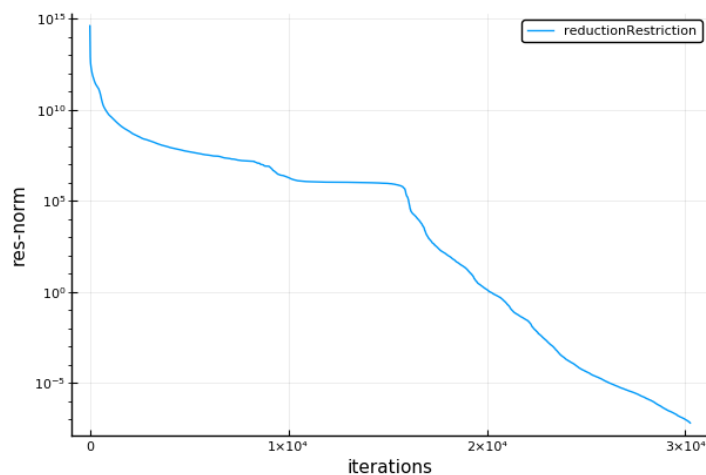


Abbildung 10: Konvergenz der Matrix BCSSTK25 mit der reduktionsbasierten Restriktion

Bei der Betrachtung fällt die relativ lange Konstanz in dem Bereich von  $10^6$  des Residuums. Bei höherer Toleranz wäre der Durchlauf sehr wahrscheinlich hier beendet worden und man hätte eine nicht zufrieden stellende Lösung. Dieses Verhalten lässt sich bei allen Matrizen der BCSSTK-Reihe erkennen.

Auch bei der Matrix WATT1 ist dieses Verhalten zu erkennen. Bei diesem Durch-

lauf geht jedoch ein steiler Fall des Residuums voraus, welcher dann in einen deutlich Geringeren übergeht.

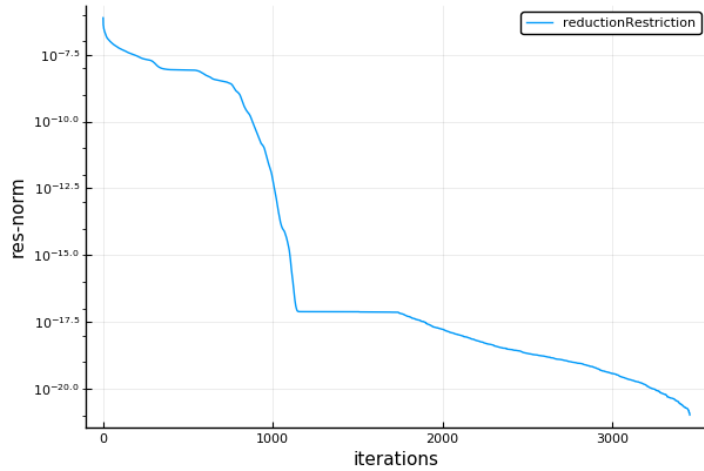


Abbildung 11: Konvergenz der Matrix WATT1 mit der reduktionsbasierten Restriktion

Man kann bei diesem und auch vielen weiteren Verläufen auch erkennen, dass das Residuum meist unter einem Wert von  $10^{-20}$  konvergiert und zum Teil sogar bis auf  $10^{-30}$  sinkt. Dies ist auch der Fall bei den Matrizen DWT234 und DWT992.

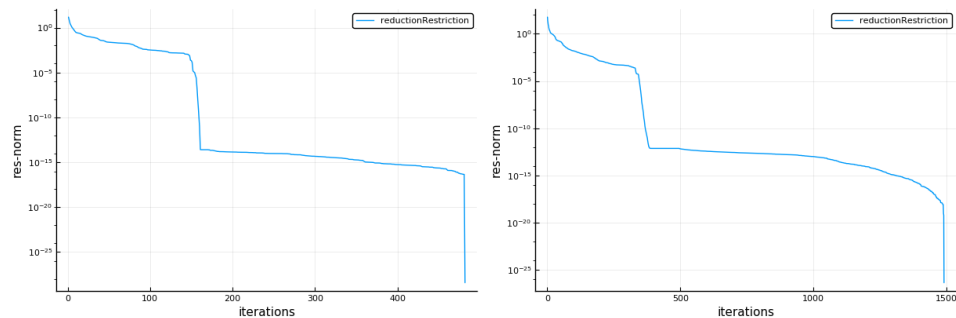


Abbildung 12: Konvergenz der Matrizen DWT234 und DWT992 (v.l.) mit der reduktionsbasierten Restriktion

Es kann wieder der typische Verlauf aufgezeigt werden, der eine Abwechslung zwischen starken und schwachen Fall des Residuums. Man kann an diesem Beispiel sehr gut erkennen, dass mit wachsender Dimension der Matrizen auch die Übergänge zwischen den verschiedenen starken Fällen des Residuums glatter werden. Außerdem steigt mit wachsender Dimension auch der Punkt, an dem der Verlauf konvergiert.

Die restlichen Verläufe ähneln stark denen, die schon in den vorherigen Kapiteln aufgezeigt wurden und werden deswegen hier nicht erneut aufgeführt.

Bei der Betrachtung der benötigten Zeit zur Berechnung wird unterschieden in Berechnungszeit der Restriktionsmatrix und die benötigte Zeit, sodass der Algorithmus zum Lösen der linearen Gleichung endet.

Zunächst wird die Erstellung der Restriktionsmatrix betrachtet. Man erkennt schnell,

dass sich hier die Anordnung stark an der Größe der vorgegebenen Matrizen orientiert. Zwar gibt einige Matrizen, die vor einer größeren kommen, jedoch handelt es sich hauptsächlich um nur kleine Unterschiede in der Dimension. Einzig die Matrix MBEAUSE und DWT234 zeigen ein anderes Verhalten. Es handelt sich bei beiden um, im Vergleich zu den restlichen Matrizen, recht kleine Matrizen.

Matrix	Zeit
BCSSTK25	167s
BCSSTK24	44.3s
GEMAT11	5.57s
BCSSTK23	5.19s
MBEAUSE	1.13s
BCSSTM26	949ms
WATT1	754ms
BCSPWR07	626ms
BCSPWR09	497ms
DWT992	400ms
DWT234	355ms
GR3030	215ms
SAYLR3	181ms
NOS7	112ms
NNC666	98.0ms
JGL009	81.3 $\mu$ s

Tabelle 10: Zeitliche Übersicht zur Ermittlungszeit für die reduktionsbasierte Restriktion

Die beiden deutlich kleineren Matrizen benötigen zum Teil deutlich länger als viele der größeren Matrizen. Vor allem die Matrix MBEAUSE ist überraschender Weise im oberen Drittel der Tabelle zu finden. Die wahrscheinlichste Erklärung hierfür ist, dass sie im Vergleich zu vielen der größeren Matrizen deutlich dichter besetzt ist.

Bei der Betrachtung der Übersicht zu Zeit und Speicher für das Lösen des Gleichungssystems erkennt man schnell einige Strukturen wieder. Die Reihenfolge ähnelt auch hier stark der Reihenfolge, welche durch die Anordnung nach Dimension entsteht. Die mit Abstand größte Matrix benötigt auch hier die meiste Zeit, nämlich gute sechs Stunden und den größten Speicher. Der große Abstand zu den folgenden Matrizen ist wohl mit dem großen Abstand der Dimensionen zu erklären, da die erste folgende Matrix eine Dimension von etwas weniger als einem Drittel von BCSSTK25 besitzt.

Matrix	Zeit	Speicher
BCSSTK25	22183s	45.3GB
GEMAT11	381s	5.21GB
BCSSTK24	141s	2.84GB
BCSSTK23	101s	2.41GB
NNC666	40.6s	3.07GB
BCSPWR09	33.0s	1.69GB
MBEAUSE	30.9s	3.04GB
BCSPWR07	26.7s	1.62GB
WATT1	17.9s	1.35GB
BCSSTM26	12.6s	1.59GB
SAYLR3	3.80s	0.97GB
GR3030	2.79s	0.93GB
DWT992	2.46s	0.94GB
NOS7	1.34s	892MB
JGL009	413ms	764MB
DWT234	365ms	791MB

Tabelle 11: Zeitliche Übersicht zur Konvergenz für die reduktionsbasierte Restriktion

Natürlich reihen sich die Matrizen nicht ein, für die der Algorithmus nicht konvergierte, da hier eine Verzerrung der Zeit und des Speichers stattfand durch die hohen Iterationszahlen, die diese Durchläufe erreichten. Aber auch einige der Matrizen für die eine Konvergenz erreicht wurde, zeigen ein interessantes Verhalten auf. So ist am Ende der Tabelle nicht die mit Abstand kleinste Matrix JGL009 zu finden, sondern die beachtlich größere Matrix DWT234. Die Matrix BCSSTM26 hingegen konnte eine für ihre Dimension vergleichsweise kurze Zeit erreichen und ist somit weiter unten in der Tabelle zu finden.

## 5.5 Vergleich simpler Restriktionen

## 5.6 Vergleich komplexer Restriktionen

## 5.7 Vergleich aller Restriktionen

## 5.8 Abweichungen zu dem nicht geglätteten Vorkonditionierer



## 6 Schlussbemerkung

## Literaturverzeichnis

- [EN09] ERLANGGA, Yogi A.; NABBEN, Reinhard: *SIAM Journal on Scientific Computing* Algebraic Multilevel Krylov Methods 31 (2009), Nr. 5, S. 3417–3437
- [GN12] GOSLLER, Florian; NABBEN, Reinhard: *Electronic Transactions on Numerical Analysis* On AMG methods with F-smoothing based on Chebyshev polynomials and their Relation to AMGr 45 (2012), S. 146–159
- [mm17] *Matrix Market*. <https://math.nist.gov/MatrixMarket/>. Version: 2017, Abruf: 04.05.2018
- [Nab17] NABBEN, Reinhard: *Multilevel methods for solving linear systems of equations*. 7 2017
- [NKR17] NABBEN, Reinhard; KEHL, René ; RAMOS, Luis G.: Projections, deflation and multigrid for non-symmetric matrices. (2017)