



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
7. 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
1 Einleitung	7
2 Theoretische Grundlagen	9
3 Deflationsmethoden	12
3.1 diagonale Restriktion	12
3.2 gewichtete diagonale Restriktion	12
3.3 häufungsbasierte Restriktion	13
3.4 reduktionsbasierende Restriktion	14
4 Verfahren	15
5 Auswertung	17
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	33
5.6 Vergleich komplexer Restriktionen	33
5.7 Vergleich aller Restriktionen	33
5.8 Abweichungen zu dem nicht geglätteten Vorkonditionierer	33
6 Schlussbemerkung	34
Literaturverzeichnis	35

Abbildungsverzeichnis

1	Konvergenz der Matrix SAYLR3 mit der diagonalen Restriktion	18
2	Konvergenz der Matrix NOS7 mit der diagonalen Restriktion	19
3	Konvergenz der Matrix GR3030 mit der diagonalen Restriktion	20
4	Konvergenz der Matrix BCSPWR07 mit der diagonalen gewichteten Restriktion	22
5	Konvergenz der Matrix DWT992 mit der diagonalen gewichteten Restriktion	22
6	Konvergenz der Matrix GEMAT11 mit der diagonalen gewichteten Restriktion	23
7	Konvergenz der Matrix GR3030 mit der diagonalen gewichteten Restriktion	23
8	Konvergenz der Matrix BCSPWR09 mit der häufungsbasierten Restriktion	25
9	Konvergenz der Matrizen DWT234 und GR3030 (v.l.) mit der häufungsbasierten Restriktion	26
10	Konvergenz der Matrix BCSSTK25 mit der reduktionsbasierten Restriktion	29
11	Konvergenz der Matrix BCSSTM26 mit der reduktionsbasierten Restriktion	30
12	Konvergenz der Matrix BCSPWR07 mit der reduktionsbasierten Restriktion	30
13	Konvergenz der Matrizen DWT234 und DWT992 (v.l.) mit der reduktionsbasierten Restriktion	31

Tabellenverzeichnis

1	Testmatrizen aus der HARWELL-BOEING-Kollektion	17
2	Übersicht zur Konvergenz für die diagonale Restriktion	18
3	Zeitliche Übersicht zur Konvergenzzeit für die diagonale Restriktion . .	20
4	Übersicht zur Konvergenz für die diagonale gewichtete Restriktion . . .	21
5	Zeitliche Übersicht zur Konvergenz 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	Zeitliche übersicht zur Konvergenz für die häufungsbasierte Restriktion	28
9	Übersicht zur Konvergenz für die reduktionsbasierte Restriktion	29
10	Zeitliche Übersicht zur Ermittlungszeit für die reduktionsbasierte Restriktion	32
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 NABBEN¹ [Nab17] und dem Paper von NABBEN, GARCÍAS² und KEHL³ [NKR17], da die Erarbeitung der theoretischen Grundlagen nicht der Hauptbestandteil dieser Arbeit seien soll.

¹Er beschäftigt sich hauptsächlich mit der numerischen Mathematik an der TU Berlin.

²Er forscht in der angewandten und numerischen Mathematik

³Sein 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 τ -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 τ -starken distanten Nachbarn sind eine Untergruppe der distanten Nachbarn, welche zusätzlich die τ -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 Ω_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 -Glätter⁴, 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

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

⁴Der 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 "Restarted GMRES" aus dem Julia-internen "LinearAlgebra"-Paket. Er wurde so konfiguriert, dass die Toleranz bei $\frac{1}{100000}$ liegt und die maximale Iterationszahl mindestens 10000 beträgt.

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 $\max(10000, n)$ und Toleranz auf $\frac{1}{100000}$*

**/*

return 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 Julia⁵ 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, welche direkt aus realen Testfällen abgeleitet wurden.

Matrix	Größe
BCSSTK25	15439 x 15439
GEMAT11	4929 x 4929
BCSSTK24	3562 x 3562
BCSSTK23	3134 x 3134
BCSSTM26	1922 x 1922
WATT1	1856 x 1856
BCSPWR09	1723 x 1723
BCSPWR07	1612 x 1612
SAYLR3	1000 x 1000
DWT992	992 x 992
GR3030	900 x 900
NOS7	729 x 729
NNC666	666 x 666
MBEAUSE	496 x 496
DWT234	234 x 234
JGL009	9 x 9

Tabelle 1: Testmatrizen aus der HARWELL-BOEING-Kollektion

⁵siehe <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 erstellt wird, sind die Auswertungen sehr divers. Da nur jeder zweite Eintrag der Matrix betrachtet wurde, bezieht dieser Algorithmus am wenigstens Einträge der eigentlichen Matrix mit in die Berechnung des Vorkonditionierers mit ein. Im größten können die Ergebnisse unterteilt werden in konvergiert und nicht-konvergiert.

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

Tabelle 2: Übersicht zur Konvergenz für die diagonale Restriktion

Eine Eigenschaft, die alle Matrizen auf der nicht-konvergenten Seite mit sehr hohen Residuum zum Ende der Berechnung eint, ist die Symmetrie. Einen beispielhaften Verlauf der nicht-konvergenten Matrizen kann man bei SAYLR3. Das Residuum zum Ende der Berechnung beläuft sich auf $\approx 10^{0,45}$.

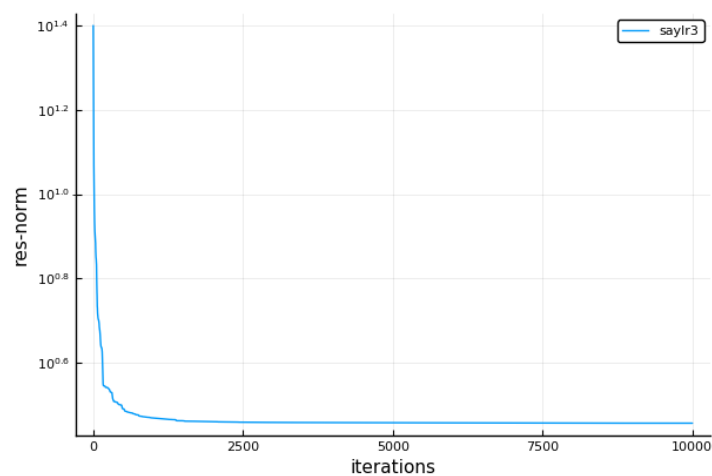


Abbildung 1: Konvergenz der Matrix SAYLR3 mit der diagonalen Restriktion

Dieser typische Verlauf der nicht-konvergenten Matrizen ist leicht zu erkennen. Ein exponentiell fallende Verlauf kann bei den nicht-konvergenten Matrizen sehr häufig beobachtet werden.

Die drei Matrizen aus dem BCSSTK-Set haben einen fast identischen Verlauf. Auch hier ist der exponentiell fallende Verlauf zu erkennen. Jedoch bleibt das Residuum für alle Drei über 10^{10} .

Die Konvergenz von NOS7 ist zwar auch exponentiell fallend, jedoch in einem sehr starken Ausmaß. Hier kann man einen zunächst sehr starken Fall des Residuums innerhalb der ersten Iterationsschritte erkennen, welcher dann abrupt in eine Konstanz umschlägt, welche bis zur maximalen Iterationszahl anhält.

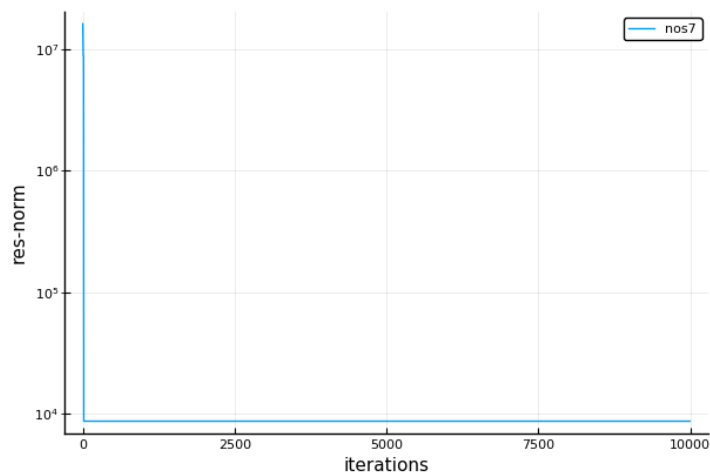


Abbildung 2: Konvergenz der Matrix NOS7 mit der diagonalen Restriktion

Die restlichen Matrizen ähneln sich alle sehr. Das Residuum fällt zunächst relativ stark und geht dann zwischen Iteration 1000 und 2000 in einen sehr schwachen Fall über. Meist liegt das Residuum bei der maximalen Iteration zwischen 10^{-2} und $10^{0,5}$. Die Matrizen für die die Berechnung konvergierte, haben keine klar erkennbare Gemeinsamkeit. Bei der Matrix GR3030 ist ein annähernd linearer Verlauf zu erkennen, welcher jedoch kleinen Schwankungen unterliegt.

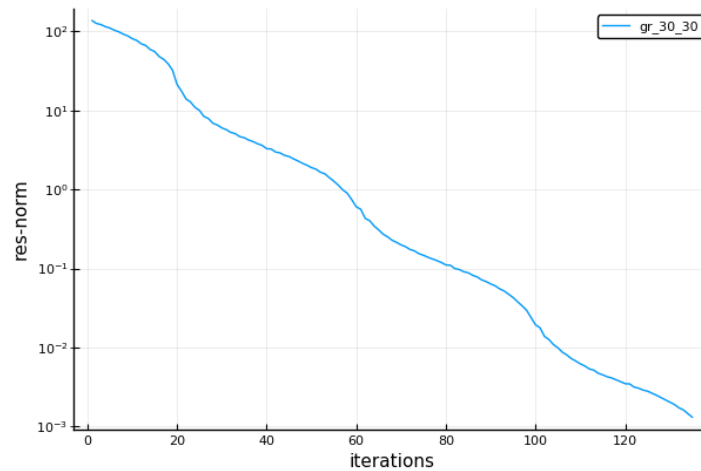


Abbildung 3: Konvergenz der Matrix GR3030 mit der diagonalen Restriktion

Dieses Ergebnis ist jedoch das einzige in diesem Versuch, welches eine solche Struktur aufweist.

Die Matrix JGL009 konvergierte bereits nach dem fünften Iterationsschritt mit einem starken Abfall des Residuums. In den ersten vier Schritten lag das Residuum noch bei ≈ 10 um dann herabzufallen auf $< 10^{-14}$. Besonders verhielt sich die Matrix WATT1, welche direkt nach dem ersten Iterationsschritt konvergierte.

Zusammenfassend kann man sagen, dass nur für wenige Matrizen eine Konvergenz vorliegt und das Schema aller nicht-konvergenten Matrizen exponentiell fallend ist.

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

Matrix	Zeit
BCSSTK25	1865s
GEMAT11	143s
BCSSTK24	68.3s
BCSSTK23	51.4s
BCSSTM26	20.2s
BCSPWR09	17.0s
BCSPWR07	15.0s
SAYLR3	4.15s
DWT992	3.98s
NNC666	2.83s
MBEAUSE	2.23s
NOS7	1.69s
WATT1	520ms
DWT234	491ms
GR3030	251ms
JGL009	1.26ms

Tabelle 3: Zeitliche Übersicht zur Konvergenzzeit für die diagonale Restriktion

Zusätzlich haben alle Durchläufe mit Matrizen bei denen die Berechnung konvergierte sehr geringe Zeiten vorzuweisen. Die Matrix DWT234 kann sich hier durch ihre geringe Größe von 234 x 234 mit bei den konvergenten Matrizen und somit im Millisekundenbereich einordnen. Auffällig ist außerdem die Matrix BCSSTK25, welche auch mit Abstand die größte Matrix war. Auch der beanspruchte Speicherplatz von 14.2GB bildet hier einen kompletten Ausreißer, da durchschnittlich eher Werte zwischen 50MB bis 1GB in Anspruch genommen wurden.

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 einbezogen. Für die gleichen Matrizen konnte auch bei dieser Methode eine Konvergenz erzielt werden.

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

Tabelle 4: Übersicht zur Konvergenz für die diagonale gewichtete Restriktion

Das BCSSTK-Set bleibt in den höchsten Residuums-Werten von über 10^{10} . Beispielfür den Großteil der nicht-konvergenten Matrizen ist auch hier die exponentiell fallende Struktur. Dies kann man gut an der Matrix BCSPWR07 erkennen.

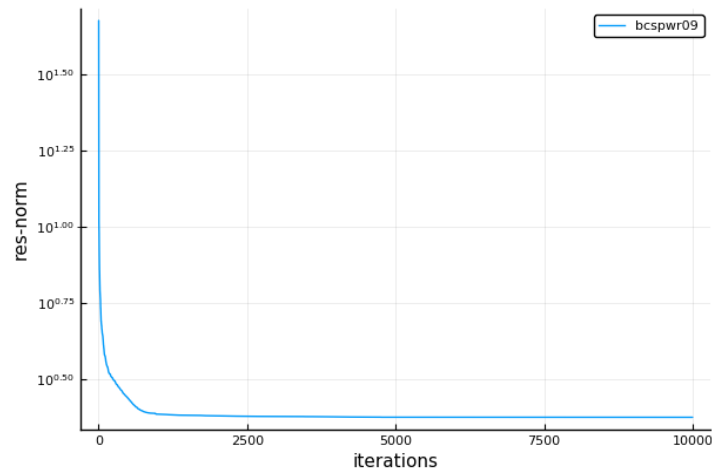


Abbildung 4: Konvergenz der Matrix BCSPWR07 mit der diagonalen gewichteten Restriktion

Man erkennt gut, dass, auch wenn man eine höhere maximale Iterationszahl angegeben hätte, das Residuum nicht konvergiert wäre. Die Matrix NOS7 schlägt auch hier nach einem extrem starken Fall des Residuums in eine spontane Konstanz um. Für die Matrix DWT992 fällt bis zu der letzten Iteration das Residuum ab. Dies war in abgeschwächter Variable auch für die diagonale Restriktion zu beobachten. Nach weiteren Iterationsschritten wäre der Algorithmus hier wohl konvergiert.

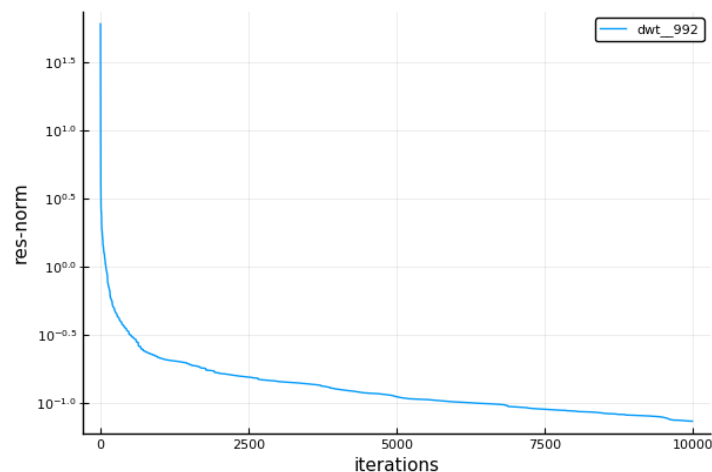


Abbildung 5: Konvergenz der Matrix DWT992 mit der diagonalen gewichteten Restriktion

Eine besondere Rolle nehmen die Matrizen NNC666 und GEMAT11 ein. Hier ist als einziges kein stetiger Fall des Residuums zu erkennen. In beiden Fällen ist eine starke Schwankung zu erkennen. Aber im Durchschnitt fällt das Residuum nach den ersten 100 Iterationsschritten nicht mehr bzw. steigt sogar für die Matrix GEMAT11.

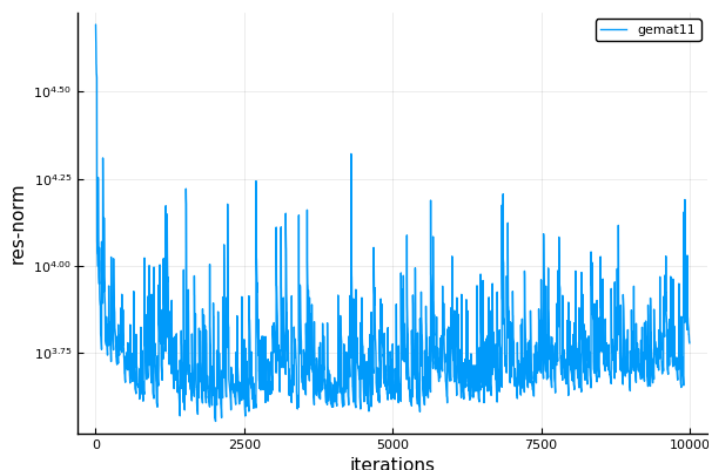


Abbildung 6: Konvergenz der Matrix GEMAT11 mit der diagonalen gewichteten Restriktion

Für die Seite der konvergierenden Matrizen fällt auch hier die Matrix WATT1 auf, welche direkt nach dem ersten Iterationsschritt konvergierte.

Der Verlauf der Matrix JGL009 ist relativ konstant bis zum vierten Iterationsschritt bei einem Residuum von 10^0 . Danach konvergiert er direkt im fünften Schritt.

Die Matrix GR3030 nimmt nach den ersten Iterationsschritten einen nahezu beispielhaften Verlauf eines linearen Falls an. In den ersten Schritten hingegen kann ein sehr starker Fall erkannt werden.

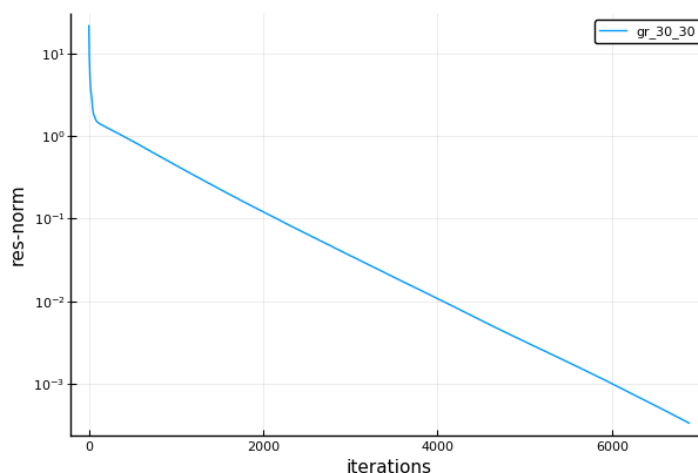


Abbildung 7: Konvergenz der Matrix GR3030 mit der diagonalen gewichteten Restriktion

Bei fast alle Matrizen kann man erkennen, dass der Verlauf in eine Konstanz endet, sich in eine Parallele der X-Achse konvergiert. Die beiden Ausnahmen bilden die Matrizen, die in einen annähernd linearen Verlauf übergehen oder dessen Residuum sehr stark um einen bestimmten Wert schwankt.

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. Die Konvergierten bilden eine Ausnahme und waren zum Teil sogar im Mikrosekunden-Bereich.

Matrix	Zeit
BCSSTK25	1890s
GEMAT11	125s
BCSSTK24	69.6s
BCSSTM26	57.7s
BCSSTK23	53.4s
BCSPWR09	18.1s
BCSPWR07	15.0s
DWT992	4.15s
SAYLR3	4.12s
MBEAUSE	2.29s
GR3030	2.12s
NOS7	1.69s
NNC666	1.35s
WATT1	554ms
DWT234	517ms
JGL009	334 μ s

Tabelle 5: Zeitliche Übersicht zur Konvergenz für die diagonale gewichtete Restriktion

Die Matrix BCSSTK25 benötigt eindeutig die meiste Zeit geschuldet durch ihre mit Abstand größten Struktur. Etwas anders als eine Reihenfolge, die durch die Dimension vorgegeben wäre, ordnet sich BCSSTM26 ein, welche vor BCSSTK23 zu finden ist, obwohl die Dimensionen einen deutlichen Abstand haben. Die konvergierten Matrizen sind am unteren Ende der Tabelle zu finden, wobei z.B. die Matrix GR3030 trotz Konvergenz längere Zeiten aufweist, als so manche nicht-konvergente Matrix.

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 erkennt, dass der Anteil der nicht-konvergenten Durchläufe überwiegt.

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

Tabelle 6: Übersicht zur Konvergenz für die häufungsbasierte Restriktion

Auffällig bei den Matrizen auf der nicht-konvergenten Seite ist, dass viele auch im letzten Iterationsschritt noch ein visuell wahrnehmbares Gefälle des Residuums aufweisen. Dies lässt auf eine mögliche Konvergenz mit genügend Iterationsschritten schließen. Ein Beispiel hierfür ist der Verlauf der Matrix BCSPWR09.

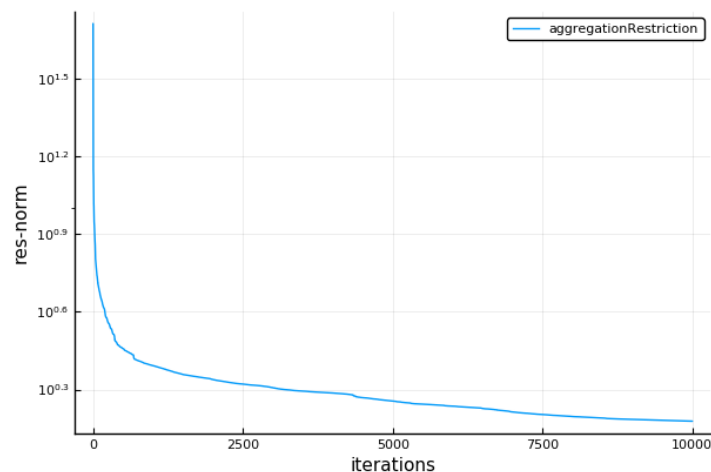


Abbildung 8: Konvergenz der Matrix BCSPWR09 mit der häufungsbasierten Restriktion

Man kann deutlich die Nähe zum exponentiell fallenden Verlauf erkennen, obwohl der bereits erwähnte weiterhin fallende Abschluss nicht auf einen beispielhaften Verlauf schließen lässt. Die andere Hälfte der Matrizen auf der nicht-konvergenten Seite entspricht genau dem exponentiell fallenden Verlauf. Ein extremer Fall dieser Art bietet die Matrix SAYL3. Hier ist zunächst ein sehr starker Fall des Residuums zu erkennen, welche dann in eine Konstanz übergeht. Die Reihe der BCSSTK-Matrizen bleibt in allen Fällen bei einem Residuum zwischen 10^{12} und 10^{14} und haben damit die höchsten Residuumswerte der gesamten Testreihe.

Bei den Matrizen für die der Algorithmus konvergierte sind keine strukturellen Gemeinsamkeiten zu erkennen. Den Ausnahmefall stellt die Matrix WATT1 dar, da diese

bereits nach dem ersten Iterationsschritt konvergierte. Der Verlauf für die beiden Matrizen GR3030 und DWT234 zeigen als einziges einige Gemeinsamkeiten auf. Bei gehen von einem stark fallenden Residuum in einen annähernd linearen Verlauf über. Jedoch ist der Verlauf trotz ähnlich hoher Iterationszahl deutlich instabiler.

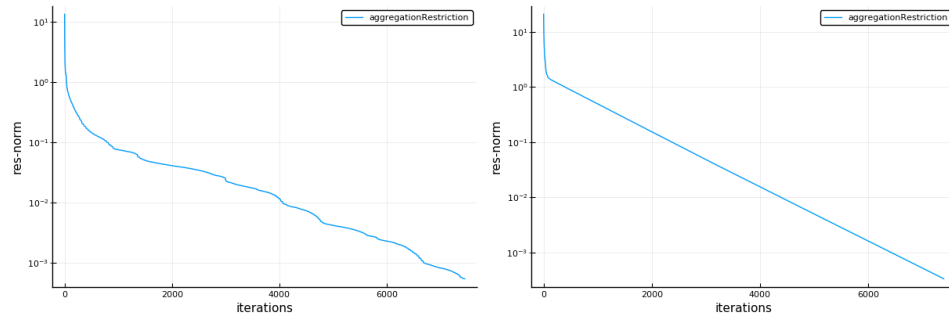


Abbildung 9: Konvergenz der Matrizen DWT234 und GR3030 (v.l.) mit der häufungs-basierten Restriktion

Hervorzuheben ist, dass in keinem der Iterationsschritte das Residuum des Graphen von DWT234. Der Verlauf für die Matrix JGL009 hält sich bis zu dem vierten Iterationsschritt um ein Residuum von 10 um im letzten Schritt zu konvergieren.

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 μ 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
BCSSTK25	1790s
GEMAT11	126s
BCSSTK24	67.1s
BCSSTK23	51.0s
BCSSTM26	20.1s
BCSPWR09	16.7s
BCSPWR07	14.2s
SAYLR3	3.80s
DWT992	3.58s
NNC666	2.63s
MBEAUSE	2.29s
GR3030	2.05s
NOS7	1.61s
WATT1	576ms
DWT234	181ms
JGL009	529 μ s

Tabelle 8: Zeitliche Übersicht zur Konvergenz für die häufungsbasierte Restriktion

Man kann erkennen, dass die größte Matrix auch die meiste Zeit in Anspruch genommen hat und die kleinste am wenigsten. Man sollte hier natürlich nicht außer Acht lassen, dass nicht immer gleich viele Iterationsschritte berechnet wurden. Jedoch sieht auch die Sortierung nach der durchschnittlichen Berechnungsdauer pro Iteration nahezu identisch aus. Einzig die Positionen der Matrizen GR3030 und DWT234 verändern sich um wenige Positionen nach oben. Die kleinste Matrix JGL009 bleibt trotz schneller Konvergenz nach fünf Iterationsschritten an letzter Stelle und auch die restlichen Matrizen ordnen sich weiterhin anhand ihrer Größe ein.

5.4 Reduktionsbasierende Restriktion

Für den Algorithmus aus 3.4 hält sich die Übersicht der konvergierten und nicht-konvergierten Durchläufe ungefähr die Waage. Auf der Seite der Konvergenz findet man zwar etwas weniger Matrizen, jedoch könnte man die Matrizen aus dem BCS-Set zusammenfassend betrachten, was dann die Seite der Konvergenz überwiegen lässt.

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

Tabelle 9: Übersicht zur Konvergenz für die reduktionsbasierte Restriktion

Diese Übersicht könnte auch weiter verschoben werden, wenn man die maximale Iterationszahl erhöhen würde, denn einige der Residuen in einem linearen Fall. Ein Beispiel für diesen Fall ist die Matrix BCSSTK25, welche durch ihre Größe die höchste Iterationszahl von 15439 Iterationen hat.

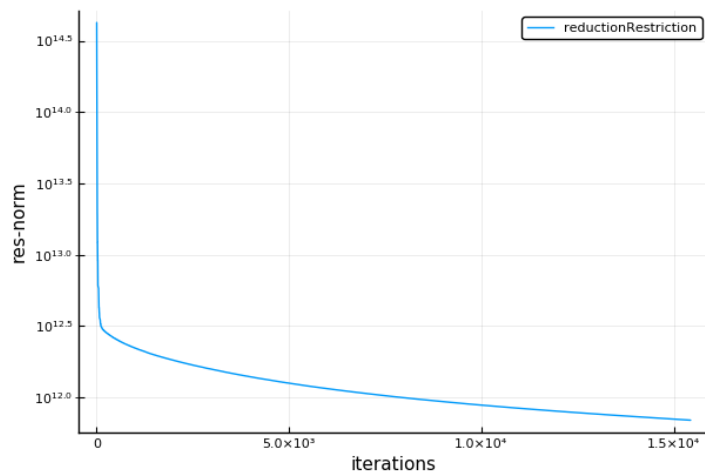


Abbildung 10: Konvergenz der Matrix BCSSTK25 mit der reduktionsbasierten Restriktion

Der hohe Wert des Residuums lässt jedoch vermuten, dass auch dieser Verlauf in eine Konstanz übergeht. Anders ist es für die Matrix BCSSTM26. Hier findet man alle Merkmale die auf eine Konvergenz in höheren Iterationszahlen hinweisen.

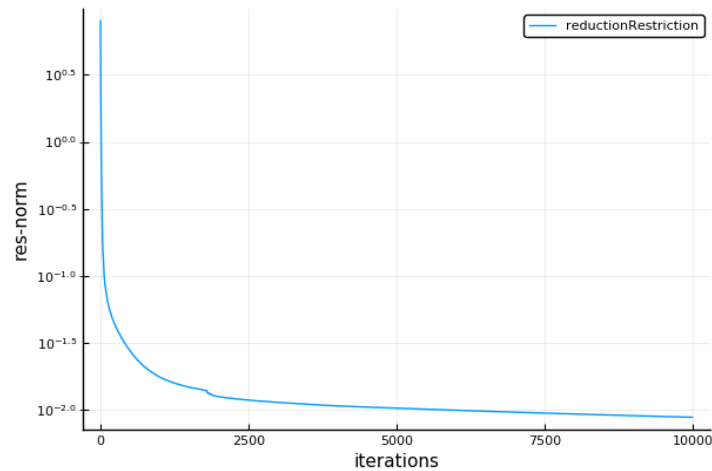


Abbildung 11: Konvergenz der Matrix BCSSTM26 mit der reduktionsbasierten Restriktion

Dies lässt darauf schließen, dass die obige Tabelle mit einer höheren Iterationszahl wirklich noch einmal zu Gunsten der konvergenten Seite verschoben worden wäre. Jedoch würde sich wahrscheinlich für mehrere der Matrizen auch keine große Änderung ergeben, da mehrere den typischen Verlauf eines exponentiellen Falls aufweisen und somit gegen eine Parallele der x-Achse konvergieren. Dies trifft vor allem auf den Verlauf der Matrizen BCSPWR07, BCSPWR09, NOS7 und BCSSTK24 zu. Bei all diesen Matrizen kann der typische Verlauf, welcher hier durch die Matrix BCSPWR07 dargestellt wird, in verschiedenen Residuumsbereichen festgestellt werden.

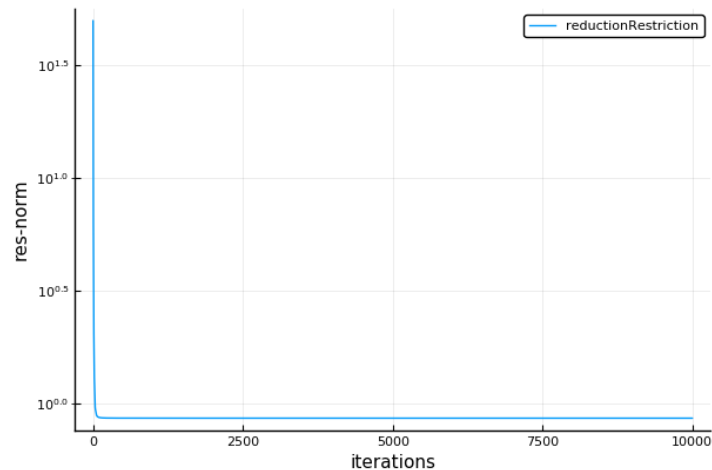


Abbildung 12: Konvergenz der Matrix BCSPWR07 mit der reduktionsbasierten Restriktion

Da alle Matrizen auf der nicht-konvergenten Seite einen relativ ähnlichen Verlauf des Residuums aufweisen, könnte man annehmen, dass dies auch der Fall für die konvergente Seite sei. Man findet auf dieser Seite jedoch eine höhere Varianz in den Formen der Verläufe.

So findet man zum Beispiel bei der Matrix WATT1 eine direkte Konvergenz nach

dem ersten Iterationsschritt. Die Form entspricht hier trivialer Weise einem linearen Graphen. Aber dies ist nicht die einzige Matrix die sich in die Richtung der linearen Konvergenzverläufe einordnen lässt. Auch die Matrizen JGL009 und GR3030 weisen ab einem bestimmten Iterationsschritt einen linearen Verlauf auf. Die Matrix JGL009 ist hier ein ebenso eher trivialer Fall, da sie nach dem vierten Iterationsschritt den abrupten Fall in die direkte Konvergenz aufzeigt und somit ab dem vierten Schritt einen linearen Verlauf aufweist. Theoretisch würde diese Zuordnung zu jedem Verlauf passen, wenn man sich jeweils nur den letzten Iterationsschritt anschaut. Aber diese Zuordnung wurde nur deswegen aufgeführt, da keine angemessene andere Zuordnung festgestellt werden konnte.

Anders verhält es sich mit den Matrizen der DWT-Reihe. Hier kann man wie auch bei alle Matrizen auf der nicht-konvergenten Seite einen annähernd exponentiell fallenden Verlauf erkennen. Die Besonderheit ist jedoch, dass keine der Verläufe eine Konstanz anstrebt, sondern ab einem bestimmten Iterationsschritt in einen leicht fallenden nahezu linearen Verlauf übergeht.

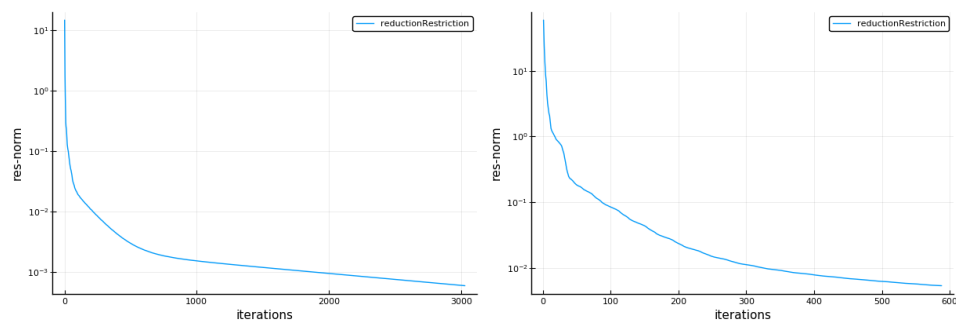


Abbildung 13: Konvergenz der Matrizen DWT234 und DWT992 (v.l.) mit der reduktionsbasierten Restriktion

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 MBE-AUSE 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 μ 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.

Der Vergleich für die Zeiten den Abschluss des Algorithmus zur Berechnung einer Lösung ist hier etwas differenzierter zu betrachten, da mehrere Durchläufe konvergierten und somit weniger Iterationsschritte benötigten.

Matrix	Zeit
BCSSTK25	1972s
GEMAT11	169s
BCSSTK24	70.5s
BCSSTK23	53.1s
BCSSTM26	20.4s
BCSPWR09	19.5s
BCSPWR07	16.5s
SAYLR3	5.72s
GR3030	2.02s
NOS7	1.61s
NNC666	1.58s
MBEAUSE	1.33s
DWT992	821ms
WATT1	732ms
DWT234	81.6ms
JGL009	353 μ s

Tabelle 11: Zeitliche Übersicht zur Konvergenz für die reduktionsbasierte Restriktion

Natürlich wird auch hier die Tabelle angeführt von dem Durchlauf für die nicht-konvergierte Matrix BCSSTK25 und abgeschlossen von der mit Abstand kleinsten Matrix JGL009.

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)