Auf Takagi-Faktoren basierende Präkonditionierung des CSYM-Verfahrens

Zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

am Fachbereich Mathematik und Naturwissenschaften
der Bergischen Universität Wuppertal
genehmigte

Dissertation

von

Dipl.-Math. Sigrid Marion Fischer

Tag der mündlichen Prüfung:	23. März 2006
Referent:	Prof. Dr. A. Frommer
Korreferent:	Prof. Dr. B. Lang

"It remains an open question what the most efficient preconditioner for a complex symmetric system would be. First empirical studies led to unexpected results and more investigations are necessary."

Angelika Bunse-Gerstner and Roland Stöver [5]

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Algorithmen

Vorwort

Komplex symmetrische Matrizen erhielten lange Zeit nur geringe Aufmerksamkeit. Die Notwendigkeit, Gleichungssysteme mit einer großen komplex symmetrischen Matrix zu lösen, wächst in verschiedenen Anwendungsgebieten. Ein Anwendungsbeispiel aus der Elektrodynamik ist die Berechnung der erzeugten Feldverteilung bei vorgegebenen Antennenpositionen. Auf diese Weise kann zum Beispiel die optimale Verteilung der Basisstationen für ein Mobilfunknetz ermittelt werden. Die Größe der zu lösenden Gleichungssysteme wächst mit der Anzahl der berücksichtigten Parameter und somit mit der Realitätsnähe der Simulation.
Im Rahmen dieser Arbeit wird das CSYM-Verfahren untersucht, das die komplexe Symmetrie der Matrix nutzt, und dadurch nur etwa den halben Speicherbedarf und Rechenaufwand hat wie Iterationsverfahren für allgemeine Matrizen. Gegenüber anderen Verfahren mit kurzen Rekursionen hat das CSYM-Verfahren zudem die sehr wichtige Eigenschaft, dass kein verfrühter Abbruch auftreten kann. Damit ist das CSYM-Verfahren von besonderer Bedeutung für das Lösen von größeren Gleichungssystemen mit komplex symmetrischer Matrix.
Gleichungssysteme mit komplex symmetrischen Matrizen können beliebig schwierig zu lösen sein, in dem Sinne, dass die Anzahl der benötigten Iterationsschritte sehr hoch sein kann. Durch eine geeignete Umformung des Gleichungssystems, die sogenannte Präkonditionierung, will man erreichen, dass die Laufzeit des CSYM-Verfahrens wesentlich reduziert wird. Für eine effiziente Präkonditionierung ist ein tieferes Verständnis der Besonderheiten von komplex symmetrischen Matrizen erforderlich.

Zum Lösen von Gleichungssystemen mit komplex symmetrischer Matrix werden häufig Krylov-Unterraum-Verfahren – auch solche, die nicht die Symmetrie der Matrix nutzen – dem CSYM-Verfahren vorgezogen. Gründe dafür sind die bisher beobachteten unerwarteten Ergebnisse, die in Krylov-Unterraum-Verfahren bewährte Präkonditionierer im CSYM-Verfahren liefern. Die vorliegende Arbeit beschäftigt sich deshalb vor allem damit, auch für das CSYM-Verfahren gute Klassen von Präkonditionierern zu finden.

Ich hoffe, mit dieser Arbeit neue Impulse für das Verständnis geeigneter Präkonditionierer für das CSYM-Verfahren zu geben, so dass das CSYM-Verfahren einen größeren Einsatzbereich findet.

◇

Diese Arbeit entstand zwischen Februar 2001 und Dezember 2005 an der Bergischen Universität Wuppertal.

Ich bedanke mich herzlich bei Prof. A. Frommer für seine Betreuung und fachlichen Anregungen und bei Prof. B. Lang für die Übernahme des Korreferats. Mein weiterer Dank gilt den restlichen Mitgliedern der Arbeitsgruppe Angewandte Informatik für das hilfreiche Miteinander.

Sigrid Marion Fischer, Wuppertal im März 2006

Notation und Definitionen

Notation

- Wenn nicht anders angegeben, notieren wir Matrizen mit Großbuchstaben und speziell Vektoren mit Kleinbuchstaben.
- Im Folgenden ist die Dimension n fest.
  Wir betrachten reguläre Matrizen $A \in \mathbb{C}^{n\times n}$ und Vektoren $u \in \mathbb{C}^n$.
- Es ist $e_k \in \mathbb{C}^n,\ 1 \leq k \leq n$, der k-te Einheitsvektor.
- Je nach Kontext steht 0 für den Skalar oder für den Nullvektor.
- Es ist $I_m \in \mathbb{C}^{m\times m}$ die Einheitsmatrix und $0_m \in \mathbb{C}^{m\times m}$ die Nullmatrix. Den Index m lassen wir weg, wenn er den maximalen Wert n annimmt.
- Sei $A = (a_{i,j})$ eine Matrix. Dann schreiben wir

  $A^T = (a_{j,i})$ für die transponierte Matrix,
  $\overline{A} = (\overline{a}_{i,j})$ für die konjugiert-komplexe Matrix und
  $A^H = (\overline{a}_{j,i})$ für die konjugiert-komplexe Transponierte zu A.
- Mit $\sigma_j(A)$ bzw. σ_j bezeichnen wir die Singulärwerte von A, wobei $\sigma_1 \geq \cdots \geq \sigma_n$.
- Die Menge der Singulärwerte von A bezeichnen wir mit $\sigma(A)$.
- Die Eigenwerte von A bezeichnen wir mit $\lambda_j(A)$ bzw. λ_j. In der Regel sortieren wir sie absteigend nach ihrem Betrag, also $|\lambda_1| \geq \cdots \geq |\lambda_n|$.
- Die Menge der Eigenwerte von A bezeichnen wir mit spek (A).
- Für das kanonische Skalarprodukt zweier Vektoren $u,\ v \in \mathbb{C}^n$ schreiben wir $< v, u >= u^H v$ und für die euklidische Norm $\|u\|_2 = \sqrt{u^H u}$.
- Es ist $\|A\|_2 = \sqrt{\lambda_1(A^H A)} = \sigma_1(A)$ die euklidische Matrixnorm von A.

- Es ist $\text{trace}\,(A) = \sum_{i=1}^{n} a_{i,i}$ die Spur von A.
- Es ist $\|A\|_F^2 = \sum_{i,j=1}^{n} |a_{i,j}|^2 = \text{trace}\left(A^H A\right)$, die Frobeniusnorm von A.
- Es ist $\text{cond}\,(A) = \sigma_1(A)/\sigma_n(A)$ die Kondition von A.
- Für ein Polynom $p(t) = \sum_{k=0}^{m} \gamma_k t^k$ bezeichnet $deg(p)$ den Grad von p, also das maximale k, so dass $\gamma_k \neq 0$.
- Es ist $\mathcal{P}_m$ der Vektorraum aller komplexen Polynome $p(t) = \sum_{k=0}^{m} \gamma_k t^k,\ \gamma_k \in \mathbb{C}$ mit $deg(p) \leq m$.
- Für eine Folge von Vektoren $v_1, \ldots, v_m$ ist

$$\text{span}\,(v_1, \ldots, v_m) = \{w \in \mathbb{C}^n |\ w = \sum_{k=1}^{m} \gamma_k v_k, \gamma_k \in \mathbb{C}\}.$$

Definitionen

- A heißt **hermitesch**, falls $A = A^H$.
- A heißt **hermitesch positiv definit**, falls A hermitesch ist und $x^H Ax > 0$ für alle $x \neq 0$ gilt.
- A heißt **symmetrisch**, falls $A = A^T$.

  Ist die Matrix $A \in \mathbb{R}^{n\times n}$, so heißt sie **reell symmetrisch**.

  Ist die Matrix $A \notin \mathbb{R}^{n\times n}$, so heißt sie **komplex symmetrisch**.
- Ein Vektor u heißt **normiert**, wenn $u^H u = 1$ gilt.
- Zwei Vektoren $u,\ v$ heißen **orthogonal**, falls $u^H v = 0$ gilt.
- Zwei Vektoren u und v heißen **orthonormal**, wenn sie normiert und orthogonal sind.
- Die Vektoren einer Menge heißen **orthogonal** bzw. **orthonormal**, wenn die Vektoren paarweise orthogonal bzw. orthonormal sind.
- Eine Matrix Q heißt **orthogonal** oder **unitär**, falls $Q^H Q = I$.
- Eine Matrix A heißt **ähnlich** zu einer Matrix B, wenn es eine reguläre Matrix S gibt, so dass $A = S^{-1}BS$.

- Eine Matrix A heißt **Diagonalmatrix**, wenn $a_{i,j} = 0$ für $i \neq j$ ist.
- Eine Matrix A heißt **Tridiagonalmatrix**, wenn $a_{i,j} = 0$ für $|i - j| > 1$ ist.
- Eine Matrix A heißt **obere Dreiecksmatrix**, wenn $a_{i,j} = 0$ für $i > j$ gilt, und speziell eine **strikte obere Dreiecksmatrix**, wenn zusätzlich $a_{i,i} = 0$ für alle i gilt.
- Eine Matrix A heißt **(strikte) untere Dreiecksmatrix**, wenn A^T eine (strikte) obere Dreiecksmatrix ist.
- Eine Matrix A heißt **diagonalisierbar (tridiagonalisierbar)**, wenn sie ähnlich zu einer Diagonalmatrix D (Tridiagonalmatrix T) ist.
- Eine Matrix A heißt **con-ähnlich** zu einer Matrix B, wenn es eine reguläre Matrix S gibt, so dass $A = SB\overline{S}^{-1}$.
- Eine Matrix A heißt **con-diagonalisierbar (con-tridiagonalisierbar)**, wenn sie con-ähnlich zu einer Diagonalmatrix D (Tridiagonalmatrix T) ist.
- Eine Matrix A heißt **T-kongruent** zu einer Matrix B, wenn es eine reguläre Matrix S gibt mit $A = SBS^T$.
- Eine Matrix A heißt **normal**, wenn $AA^H = A^H A$ gilt.
- Zwei Matrizen A und B **kommutieren**, wenn $AB = BA$ gilt.
- Wir nennen Iterationsverfahren, die die komplexe Symmetrie der Ausgangsmatrix nutzen und beibehalten, **CSYM-Iterationsverfahren**.
- Das **CSYM-Verfahren** bezeichnet dagegen speziell das Iterationsverfahren aus [5].

Kapitel 1

Einleitung

Gegeben sei eine reguläre Matrix A, die komplex symmetrisch ist, d.h. es gilt

$$A = A^T \in \mathbb{C}^{n\times n},\ A \notin \mathbb{R}^{n\times n}.$$

Wir gehen davon aus, dass die betrachtete Matrix groß ist. Sie ist je nach der Art der verwendeten Diskretisierungsmethode entweder dicht oder dünn besetzt.
Wir befassen uns im Folgenden mit Iterationsverfahren zum Lösen von Gleichungssystemen mit komplex symmetrischen regulären Matrizen, d.h. wir suchen eine Lösung x von $A\,x = b$, also $x = A^{-1}b$ für eine rechte Seite b. Häufig ist das Gleichungssystem für verschiedene rechte Seiten zu lösen.
Die wesentlichen Aufwände pro Iterationsschritt sind Matrix-Vektormultiplikationen mit A, also Aufwände von der Größenordnung $\mathcal{O}(n^2)$. Für dünn besetzte Matrizen können Matrix-Vektorprodukte häufig günstiger ausgeführt werden. Iterative Lösungsverfahren sind aber auch für Gleichungssysteme mit dichter Matrix geeignet, solange der Aufwand geringer als der für die direkte Lösung der Gleichungssysteme ist. Dies ist z.B. der Fall, wenn die Anzahl der Iterationsschritte kleiner als n ist. Unter Nutzung der Symmetrie der Matrix kann nur eine Matrix-Vektormultiplikation pro Iterationsschritt erforderlich sein.

Daher untersuchen wir in Kapitel 2 Eigenschaften komplex symmetrischer Matrizen. Wie wir sehen werden, können die Eigenwerte solcher Matrizen beliebig sein. Komplex symmetrische Matrizen müssen auch nicht diagonalisierbar sein – sie können also beliebig nicht-normal sein. Die Singulärwertzerlegung von komplex symmetrischen Matrizen weist dagegen eine Besonderheit auf: Sie kann symmetrisch gewählt werden, als sogenannte Takagi-Faktorisierung [39].

In Kapitel 3 werden wir das CSYM-Verfahren als Iterationsverfahren für lineare Gleichungssysteme mit komplex symmetrischer Matrix kennen lernen. Es gehört, wie alle hier betrachteten iterativen Verfahrenstypen, zu den Unterraum-Verfahren. Ausgehend

von einem Startvektor wird hierbei in jedem Schritt ein Unterraum erweitert und aus dem affinen Unterraum eine neue Approximation der Lösung bestimmt. Die Matrix A wird dabei durch Projektionsmatrizen auf eine Matrix mit vereinfachter Struktur projiziert, im günstigsten Fall auf eine Tridiagonalmatrix. Die Approximation wird dann in jedem Schritt als Lösung eines Geichungssystems mit quadratischer oder rechteckiger projizierter Matrix ermittelt. Anhand des Unterraumes unterscheiden wir zwischen Krylov-Unterraum- und Quasi-Krylov-Unterraum-Verfahren.
Für das Lösen linearer Gleichungssysteme mit großen Matrizen haben sich insbesondere Krylov-Unterraum-Verfahren mit der sogenannten Minimales-Residuum-Eigenschaft als sinnvoll erwiesen. Oft werden zur Lösung von Gleichungssystemen mit komplex symmetrischer Matrix Krylov-Unterraum-Verfahren eingesetzt, die nicht die Symmetrie der Ausgangsmatrix nutzen. Es handelt sich im Wesentlichen um:

- das CGNR-Verfahren, das CG-Verfahren auf die Normalengleichung angewendet,
- das GMRES-Verfahren, das auf allgemeine Matrizen angewendet wird.

Ein anderer Ansatz stellt die Transformation auf ein reelles System doppelter Größe mit reell symmetrischer Matrix dar, das dann mit einem Krylov-Unterraum-Verfahren für reell symmetrische Matrizen gelöst werden kann.
Für hermitesche Matrizen können Speicher- und Aufwand-reduzierte Verfahren angewendet werden. Insbesondere haben diese Matrizen zwei besondere Eigenschaften: Sie haben reelle Eigenwerte und sind mit einer unitären Transformationsmatrix diagonalisierbar. Das bedeutet, dass diese Matrizen in dem Sinne konditioniert sind, dass kleine Änderungen in den Matrixkoeffizienten kleine Änderungen in den Eigenwerten bewirken. Diese beiden Eigenschaften gelten, wie wir in Kapitel 2 sehen werden, nicht für komplex symmetrische Matrizen.
Es ist dennoch sinnvoll, sich mit effizienteren Verfahren bzgl. Speicherplatz und Zeitbedarf, die die spezielle Struktur von komplex symmetrischen Matrizen ausnutzen und beibehalten, zu beschäftigen. Dazu betrachten wir zwei Iterationsverfahren zur Lösung von linearen Gleichungssystemen mit einer komplex symmetrischen Matrix A, die die Symmetrie der Matrix A berücksichtigen, indem sie A auf eine komplex symmetrische Tridiagonalmatrix projizieren.
Das ersten Verfahren ist das CQMR-Verfahren [11], ein Krylov-Unterraum-Verfahren basierend auf einer Variante des klassischen Lanczos-Verfahrens, mit den Eigenschaften:

- Die Eigenwerte der projizierten Matrizen approximieren die Eigenwerte von A.
- Die Norm der Projektionsmatrix muss nicht beschränkt sein, so dass das Verfahren schlecht konditioniert sein kann.

- Außerdem kann das Verfahren versagen, indem ein sogenannter breakdown auftritt.
- Die approximierte Lösung des linearen Gleichungssystems erfüllt eine schwächere Bedingung als die Minimales-Residuum-Eigenschaft.

Das zweite Verfahren ist das CSYM-Verfahren [5], ein Verfahren, das orthogonale Transformationen benutzt und folgende Eigenschaften hat:

- Die Singulärwerte der Ausgangsmatrix werden durch die Singulärwerte der projizierten Matrix approximiert.
- Die approximierte Lösung des linearen Gleichungssystems minimiert das Residuum im Unterraum, hat also die Minimales-Residuum-Eigenschaft.
- Es ist in exakter Arithmetik ein endliches Verfahren.
- Wie auch das CQMR-Verfahren, kommt es mit konstantem Aufwand pro Iterationsschritt aus (im Wesentlichen eine Matrix-Vektormultiplikation).
- Es ist jedoch kein Krylov-, sondern ein Quasi-Krylov-Unterraum-Verfahren.

Mit der effizienten Präkonditionierung des CSYM-Verfahrens beschäftigt sich der übrige Teil dieser Arbeit. Es geht darum, das ursprüngliche Gleichungssystem geeignet umzuformen, so dass das CSYM-Verfahren auf diesem System schneller konvergiert als auf dem ursprünglichen. Um eine komplex symmetrische präkonditionierte Matrix zu erhalten – damit wir das CSYM-Verfahren überhaupt weiter nutzen können – wenden wir einen komplex symmetrischen Präkonditionierer $M = SS^T$ beidseitig an. Dieses ist ein typisches Vorgehen zur Formulierung eines präkonditionierten Verfahrens. Um z.B. das CG-Verfahren für hermitesch positiv definite Matrizen weiter nutzen zu können, wird entsprechend von einer hermitesch positiv definiten präkonditionierten Matrix ausgegangen, die man durch beidseitige Präkonditionierung mit einem faktorisierten Präkonditionierer $\tilde{M} = \tilde{S}\tilde{S}^H$ erhält. Das CG-Verfahren kann dann algorithmisch sogar so formuliert werden, dass nur $\tilde{M}$, nicht jedoch $\tilde{S}$, benötigt wird. Bei dem CSYM-Verfahren ist eine entsprechende Vorgehensweise allerdings nicht möglich, da in der Formulierung statt $M = SS^T$ die Faktoren S in der Form SS^H vorkommen. Es reicht daher nicht aus, einen symmetrischen Präkonditionierer M zu wählen, sondern die Faktorisierung $M = SS^T$ muss immer zusätzlich bestimmt werden.
Zunächst untersuchen wir die Änderung der Singulärwerte der ursprünglichen Matrix bei beidseitiger Präkonditionierung und leiten daraus Forderungen an einen guten Präkonditionierer ab. Die Transformation zwischen verschiedenen Faktorisierungen des gleichen Präkonditionierers hilft uns bei der Beantwortung der Frage, welche Faktorisierung bei

einem festen Präkonditionierer optimal ist. Die Takagi-Faktorisierung liefert die besten Ergebnisse, scheint aber auf den ersten Blick zu aufwändig zu sein.
Abschließend betrachten wir in den Kapiteln 5 und 6 konkrete Anwendungen, die diesen optimalen Präkonditionierer nutzen. Im CSYM-Verfahren mit Deflation gehen wir direkt von einem Takagi-Faktor aus. Der Aufwand zur Anwendung der Präkonditionierung ist moderat, die Berechnung eines guten Faktors dagegen schwierig, insbesondere für große Matrizen. Takagi-faktorisierte Block-Präkonditionierer werden wir in Kapitel 6 untersuchen.

Kapitel 2

Komplex symmetrische Matrizen

Im Folgenden untersuchen wir Eigenschaften von komplex symmetrischen Matrizen. Es gilt zunächst

$$A = A^T \in \mathbb{C}^{n\times n},\ A \notin \mathbb{R}^{n\times n}$$
$$\Rightarrow A^H = \bar{A}^T \neq A^T.$$

Nur für Koeffizienten $a_{i,j} \in \mathbb{R}$ gilt $a_{i,j} = a_{i,j}^T = \overline{a_{i,j}}^T$. Daher können die Diagonalelemente $a_{i,i} = a_{i,i}^T$ von komplex symmetrischen Matrizen komplexe Zahlen sein. Für hermitesche Matrizen gilt dagegen $a_{i,i} = \overline{a_{i,i}}^T$, so dass alle Diagonalelemente reell sind.
Unter den symmetrischen Matrizen sind gerade die reell symmetrischen Matrizen hermitesch und daher immer mit einer unitären Basis aus Eigenvektoren diagonalisierbar. Dies ist für die Klasse der komplex symmetrischen Matrizen nicht der Fall.
Zunächst betrachten wir Aussagen über Eigenwerte und Eigenvektoren von komplex symmetrischen Matrizen. Dabei geht es um die Beantwortung der Frage, wann es eine Basis aus Eigenvektoren zu den Eigenwerten gibt, d.h. wann eine komplex symmetrische Matrix diagonalisierbar ist. Anschließend charakterisieren wir den Spezialfall der normalen komplex symmetrischen Matrix.
Dann wenden wir uns den Singulärwerten und Singulärvektoren zu. Insbesondere die Singulärvektoren von komplex symmetrischen Matrizen können in spezieller Form gewählt werden. Der maximale Singulärwert lässt sich durch eine variationelle Maximum-Beziehung darstellen.

2.1 Eigenwerte

Wir beginnen mit den Spektraleigenschaften von komplex symmetrischen Matrizen. Von hermiteschen und damit auch von reell symmetrischen Matrizen ist bekannt, dass ihre

Eigenwerte reell sind. Was lässt sich über die Eigenwerte von komplex symmetrischen Matrizen sagen? Aussagen hierzu liefert der folgende Satz.

Satz 2.1. *Jede komplexe Matrix ist ähnlich zu einer symmetrischen Matrix.*

Beweis: [23] Jede komplexe Matrix ist ähnlich zu einer oberen Jordanschen Normalform

$$J = J_{n_1}(\lambda_1) \oplus J_{n_2}(\lambda_2) \oplus \cdots \oplus J_{n_k}(\lambda_k) \text{ mit Jordanblöcken}$$
$$J_{n_l}(\lambda_l) = N_{n_l} + \lambda_l I_{n_l}, \text{ wobei } N_{n_l} = \begin{pmatrix} 0 & 1 & & 0 \\ & 0 & \ddots & \\ & & \ddots & 1 \\ 0 & & & 0 \end{pmatrix} \in \mathbb{C}^{n_l \times n_l}.$$

Sei $B_{n_l} \in \mathbb{C}^{n_l \times n_l}$, die Rückwärtseinheitsmatrix, gegeben durch

$$B_{n_l} = \begin{pmatrix} 0 & & 1 \\ & .^{\cdot} & \\ 1 & & 0 \end{pmatrix} \text{ und } T_{n_l} = \frac{1}{\sqrt{2}}(I_{n_l} + iB_{n_l}).$$

Die Matrix T_{n_l} ist komplex symmetrisch und unitär. Sei $S_{n_l}(\lambda_l) = T_{n_l} J_{n_l}(\lambda_l) \bar{T}_{n_l}$. Dann gilt $S_{n_l}(\lambda_l) = T_{n_l} N_{n_l} \bar{T}_{n_l} + T_{n_l} \lambda_l I_{n_l} \bar{T}_{n_l}$ und

$$\begin{aligned} T_{n_l} N_{n_l} \bar{T}_{n_l} &= \frac{1}{2}(N_{n_l} + B_{n_l} N_{n_l} B_{n_l}) + \frac{i}{2}(B_{n_l} N_{n_l} - N_{n_l} B_{n_l}), \\ T_{n_l} \lambda_l I_{n_l} \bar{T}_{n_l} &= \frac{1}{2} \lambda_l I_{n_l}. \end{aligned}$$

Die Matrix $T_{n_l} N_{n_l} \bar{T}_{n_l}$ ist symmetrisch, denn

$$\begin{aligned}
B_{n_l} N_{n_l} B_{n_l} &= \begin{pmatrix} 0 & & & 0 \\ 1 & 0 & & \\ & \ddots & \ddots & \\ 0 & & 1 & 0 \end{pmatrix} = N_{n_l}^T, \\
B_{n_l} N_{n_l} &= \begin{pmatrix} 0 & & & 0 \\ & & 0 & 1 \\ & \iddots & \iddots & \\ 0 & 1 & & 0 \end{pmatrix} = (B_{n_l} N_{n_l})^T \text{ und} \\
N_{n_l} B_{n_l} &= \begin{pmatrix} 0 & & 1 & 0 \\ & \iddots & \iddots & \\ 1 & 0 & & \\ 0 & & & 0 \end{pmatrix} = (N_{n_l} B_{n_l})^T.
\end{aligned}$$

Daher ist $S_{n_l}(\lambda_l)$ auch symmetrisch. Die Anwendung der Ähnlichkeitstransformation mit $T = T_{n_1} \oplus T_{n_2} \oplus \cdots \oplus T_{n_k}$ liefert also die zu J ähnliche symmetrische Matrix

$$S = S_{n_1}(\lambda_1) \oplus S_{n_2}(\lambda_2) \oplus \cdots \oplus S_{n_k}(\lambda_k).$$

□

Aus Satz 2.1 folgt, dass i.A. keinerlei spezielle Eigenschaften des Eigensystems einer komplex symmetrischen Matrix A vorliegen müssen. Insbesondere muss A weder normal noch diagonalisierbar sein.
Eine notwendige Bedingung dafür, dass eine komplexe Matrix ähnlich zu einer reell symmetrischen ist, sind reelle Eigenwerte.

Im Folgenden charakterisieren wir die Spezialfälle diagonalisierbarer und normaler komplex symmetrischer Matrizen genauer.

2.2 Diagonalisierbarkeit

Wir betrachten zunächst die Diagonalisierbarkeit mit der Kernfrage, wann es eine Basis von Eigenvektoren gibt.
Die geometrische Vielfachheit eines Eigenwertes entspricht der Dimension des Raumes, der durch die zugehörigen Eigenvektoren erzeugt wird, wohingegen die algebraische Vielfachheit der Potenz des Eigenwertes im charakteristischen Polynom von A entspricht. Stimmen die beiden Größen jeweils für alle verschiedenen Eigenwerte überein, so ist die Matrix diagonalisierbar. Problemfälle sind somit mehrfache Eigenwerte, die eine geometrische Vielfachheit haben, die kleiner als die algebraische Vielfachheit ist.
Ist eine komplex symmetrische Matrix diagonalisierbar, so hat sie durchaus spezielle Eigenschaften hinsichtlich der Transformationsmatrix der Diagonalisierung, wie der nachfolgende Satz zeigt. Zuvor definieren wir einige Begriffe.

Definition 2.2. Zwei Vektoren u und $v \in \mathbb{C}^n$ heißen **komplex-orthogonal**, falls $u^T v = 0$ gilt.

Definition 2.3. Eine Matrix $P \in \mathbb{C}^{n\times n}$ heißt **komplex-orthogonal**, wenn $P^T P = I$ ist.

Definition 2.4. Ein Vektor $p \in \mathbb{C}^n,\ p \neq 0$, heißt **quasi-null**, wenn $p^T p = 0$ ist.

Bemerkung 2.5. Das innere Produkt $u^T v \in \mathbb{C}$ ist indefinit und damit kein Skalarprodukt.

Satz 2.6. *Sind zwei symmetrische Matrizen A und B ähnlich, dann sind sie über eine komplex-orthogonale Transformationsmatrix P ähnlich.*

Für den Beweis benötigen wir folgenden Satz:

Satz 2.7. *(Polarzerlegung) Eine reguläre Matrix $A \in \mathbb{C}^{n\times n}$ kann in der Form*

$$A = SP \text{ bzw. } A = P_1 S_1$$

dargestellt werden. Dabei sind S und S_1 symmetrische, P und P_1 komplex-orthogonale Matrizen.
Ferner ist

$$S = \sqrt{AA^T} = f(AA^T),\ S_1 = \sqrt{A^T A} = f_1(A^T A),$$

wobei $f(\lambda)$ und $f_1(\lambda)$ Polynome in λ sind.
Es gilt $S = S_1$ und $P = P_1$ genau dann, wenn A mit A^T kommutiert.

Beweis: s. [13].

Zurück zum Beweis von Satz 2.6 [13]:
Sei $C \in \mathbb{C}^{n\times n}$ regulär und $B = C^{-1}AC$. Dann folgt

$$B^T = C^T A C^{-T}. \tag{2.1}$$

Da $B^T = B$ und $A^T = A$, gilt auch:

$$B^T = C^{-1} A^T C. \tag{2.2}$$

Aus (2.1) und (2.2) folgt damit $CC^T A = ACC^T$. Da also CC^T mit A kommutiert, kommutiert auch $S = \sqrt{CC^T}$ mit A. Setzen wir die Polarzerlegung $C = SP$ in (2.2) ein, so erhalten wir daher

$$B = P^{-1}S^{-1}ASP = P^{-1}AP.$$

□

Da eine Diagonalmatrix insbesondere eine symmetrische Matrix ist, folgt unmittelbar aus Satz 2.6 das folgende Lemma.

Lemma 2.8. *Eine komplex symmetrische Matrix A ist genau dann diagonalisierbar, wenn sie mit einer komplex-orthogonalen Matrix P diagonalisierbar ist.*

Folgerung 2.9. Aus $P^{-1}AP = D$ mit einer Diagonalmatrix D und einer komplex-orthogonalen Matrix P folgt $P^TAP = D$. Die T-Kongruenz erhält also die Symmetrie der ursprünglichen Matrix. Die Eigenwerte ändern sich bei einer T-Kongruenz höchstens dann, wenn die Transformationsmatrix nicht komplex-orthogonal ist.

Weiter lässt sich nach [9] zeigen, dass eine nicht diagonalisierbare komplex symmetrische Matrix A mindestens einen Eigenvektor p hat, der quasi-null ist. Dieser Fall kann nur bei mehrfachen Eigenwerten eintreten.

Beispiel 2.10. [23]

Die Matrix $\begin{pmatrix} 2i & 1 \\ 1 & 0 \end{pmatrix}$ hat den doppelten Eigenwert i und nur einen Eigenvektor $\begin{pmatrix} i \\ 1 \end{pmatrix}$.

2.3 Normalität

Äquivalente Bedingungen für die Normalität einer Matrix A sind [23]:

- $A^H A = AA^H$
- $A = U\Lambda U^H$ mit einer unitären Matrix U, die als Spalten die Eigenvektoren von A enthält, und einer Diagonalmatrix Λ mit den Eigenwerten von A.

Nun zur Normalität von komplex symmetrischen Matrizen. Wann gibt es eine Orthogonalbasis aus Eigenvektoren? Aus Lemma 2.8 folgt, dass eine komplex symmetrische Matrix A genau dann normal ist, wenn die komplex-orthogonale Transformationsmatrix P unitär und damit eine **reelle** Matrix ist. Weitere Aussagen über diese unitäre Matrix liefert Teil ii) des Beweises des folgenden Lemmas.

Lemma 2.11. *Eine komplex-orthogonale Matrix $A = B + iC$ mit reell symmetrischen Matrizen B und C ist genau dann normal, wenn B und C kommutieren.*

Beweis: [23] Wir zeigen in i), dass aus A normal $BC = CB$ folgt und in ii) die umgekehrte Richtung.

i) Sei A normal und damit $A^H A = AA^H$. Da

$$AA^H = (B + iC)(B - iC) = B^2 - iBC + iCB + C^2 \text{ und}$$
$$A^H A = (B - iC)(B + iC) = B^2 + iBC - iCB + C^2$$

folgt, dass $-iBC + iCB = iBC - iCB$ gilt und damit $CB = BC$.

ii) Sei $BC = CB$. Da die Matrizen B und C reell symmetrisch sind, sind sie normal. Wenn zwei normale Matrizen miteinander kommutieren, so sind sie mit der gleichen unitären Transformationsmatrix Q diagonalisierbar [23]. Daher gilt

$$QAQ^H = Q(B + iC)Q^H = QBQ^H + iQCQ^H \text{ ist diagonal.}$$

Es gibt also eine orthonormale Basis aus Eigenvektoren zu Eigenwerten von A, d.h. A ist normal.

□

Eine normale komplex symmetrische Matrix hat daher mindestens einen nicht-reellen Eigenwert. Wir werden auf die Normalität noch im folgenden Abschnitt über Singulärwerte zurückkommen.

2.4 Symmetrische Singulärwertzerlegung

Zunächst führen wir Singulärwerte und Singulärvektoren für allgemeine Matrizen ein.

Satz 2.12. ***(Singulärwertzerlegung)*** *Sei $A \in \mathbb{C}^{m\times n}$ und $q = \min(m,n)$. Dann gibt es eine Matrix $\Sigma = (\sigma_{i,j}) \in \mathbb{R}^{m\times n}$ mit $\sigma_{i,j} = 0$ für $i \neq j$ und $\sigma_{1,1} \geq \sigma_{2,2} \geq \cdots \geq \sigma_{q,q} \geq 0$, und es gibt zwei unitäre Matrizen $V \in \mathbb{C}^{m\times m}$ und $W \in \mathbb{C}^{n\times n}$, so dass $A = V\Sigma W^H$ gilt. Die reellen Zahlen $\sigma_i = \sigma_{i,i}$ sind identisch mit den nichtnegativen Wurzeln der Eigenwerte von AA^H.*
Die Spalten von V sind die Eigenvektoren von AA^H und die Spalten von W sind die Eigenvektoren von A^HA.

Beweis: s. [24].

Definition 2.13. Die Zahlen σ_i heißen **Singulärwerte**, die Spalten von V **linke Singulärvektoren** und die Spalten von W **rechte Singulärvektoren** von A.

Die Singulärwerte können wie folgt charakterisiert werden [24] (S. 148):

$$\begin{aligned}
&\sigma_1(A) = \max\{\|Ax\|_2 : x \in \mathbb{C}^n, \|x\|_2 = 1\} \text{ und } \sigma_1 = \|Aw_1\|_2\\
&\sigma_2(A) = \max\{\|Ax\|_2 : x \in \mathbb{C}^n, \|x\|_2 = 1, x \perp w_1\} \text{ und } \sigma_2 = \|Aw_2\|_2\\
&\vdots\\
&\sigma_k(A) = \max\{\|Ax\|_2 : x \in \mathbb{C}^n, \|x\|_2 = 1, x \perp w_1, \ldots, w_{k-1}\} \text{ und } \sigma_k = \|Aw_k\|_2
\end{aligned}$$

Mit Hilfe der Singulärwertzerlegung (SVD für Singular Value Decomposition) und der Schurform kann ein Maß für die Abweichung von der Normalität einer Matrix definiert werden [16].

Satz 2.14. *Sei $A \in \mathbb{C}^{n\times n}$. Die Eigenwerte von A seien gegeben durch $\lambda_1, \ldots, \lambda_n$ und die Singulärwerte durch $\sigma_1, \ldots, \sigma_n$. Sei $A = QRQ^H$ die Schurzerlegung von A mit einer unitären Matrix Q und einer oberen Dreiecksmatrix $R = \Lambda + N$, zerlegt in die Diagonalmatrix Λ aus Eigenwerten von A und eine echte obere Dreiecksmatrix N. Dann gilt*

$$\|A\|_F^2 = \sum_{k=1}^{n} \sigma_k^2 = \|\Lambda\|_F^2 + \|N\|_F^2.$$

Die Matrix A ist genau dann normal, wenn $\|N\|_F^2 = 0$ *und damit*

$$\|A\|_F^2 = \sum_{k=1}^{n} |\lambda_k|^2 = \sum_{k=1}^{n} \sigma_k^2 \text{ gilt.}$$

Beweis: s. [16].

Komplex symmetrische Matrizen haben eine Besonderheit hinsichtlich der Singulärwertzerlegung, ein Analogon zu der Eigenwertzerlegung von hermiteschen Matrizen, wie der folgende Satz zeigt.

Satz 2.15. ***(Symmetrische SVD)*** *Jede komplex symmetrische Matrix A hat eine symmetrische Singulärwertzerlegung (SSVD), eine sogenannte Takagi-Faktorisierung, der Form*

$$A = V\Sigma V^T$$

mit einer unitären Matrix $V \in \mathbb{C}^{n\times n}$ *und einer Diagonalmatrix* $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$ *mit den Singulärwerten* $\sigma_i \geq 0$.

Beweis: s. [23].

Jeder Singulärwert σ_k und der zugehörige Singulärvektor v_k aus der symmetrischen Singulärwertzerlegung erfüllen daher die Gleichung $A\overline{v}_k = \sigma_k v_k$.

Notation. O.B.d.A. wird $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ angenommen.

Die symmetrische Singulärwertzerlegung für komplex symmetrische Matrizen $A \in \mathbb{C}^{n\times n}$ ist ein Spezialfall der Singulärwertzerlegung für eine beliebige Matrix $A = U\Sigma W^H$. Bei der symmetrischen Singulärwertzerlegung ist $W = \overline{U}$. Sie ist eine con-Ähnlichkeitstransformation mit unitärer Matrix, denn es gilt

$$A = V\Sigma V^T = V\Sigma\overline{V}^{-1}.$$

Es gilt sogar die Äquivalenz: Eine Matrix ist genau dann symmetrisch, wenn A unitär con-diagonalisierbar ist [23].

Gilt $V \in \mathbb{R}^{n\times n}$, so erhalten wir den Spezialfall einer reell symmetrischen Matrix.

Definition 2.16. Wir nennen eine Matrix $S = V\Sigma^{1/2}$ aus der Takagi-Faktorisierung $A = V\Sigma V^T$ einer komplex symmetrischen Matrix A einen **Takagi-Faktor**.

Folgendes Lemma verdeutlicht den Zusammenhang zwischen der Eigenwertzerlegung von AA^H und der symmetrischen Singulärwertzerlegung von A für den Spezialfall, dass es keine mehrfachen Singulärwerte größer 0 gibt.

Lemma 2.17. *Sei $A \in \mathbb{C}^{n\times n}$ symmetrisch. Weiter seien alle positiven Eigenwerte von $A\overline{A}$ einfach und sei $A\overline{A} = V\Sigma^2 V^H$ die unitäre Diagonalisierung von $A\overline{A}$ mit*

$$\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n),\ \sigma_i \geq 0,\ i = 1, \ldots, n.$$

Dann gibt es eine Diagonalmatrix $D = \operatorname{diag}(e^{i\theta_1}, \ldots, e^{i\theta_n})$ mit $\theta_j \in \mathbb{R}$, so dass $A = U\Sigma U^T$ mit $U = VD$. Die Diagonaleinträge von D zu entsprechenden Nichtnull-Diagonaleinträgen von Σ sind durch die Beziehung $A\overline{V} = V\Sigma D^2$ festgelegt; die Diagonaleinträge von D, die den Nullelementen von Σ entsprechen, können als 1 gewählt werden.

Beweis: s. [23].

Die Singulärvektoren zu einfachen Singulärwerten sind bis auf Drehungen eindeutig. Jedoch folgt im Allgemeinen nicht aus der Eigenwertzerlegung $AA^H = V\Sigma^2 V^H$, dass die symmetrische Singulärwertzerlegung von A durch $A = V\Sigma V^T$ gegeben ist.

Aus der Singulärwertzerlegung für allgemeine Matrizen erhalten wir wie folgt die symmetrische Singulärwertzerlegung.

Lemma 2.18. *Sei A komplex symmetrisch und regulär und $A = V\Sigma W^H$ eine Singulärwertzerlegung. Seien dann in V_k die linken und W_k die rechten Singulärvektoren zum Singulärwert σ_k mit Vielfachheit $m_k \geq 1$ (auch Spezialfall eines einfachen Singulärwertes) enthalten. Dann ist die symmetrische Singulärwertzerlegung von $A = U\Sigma U^T$ gegeben durch $U = V\overline{S}$ mit einer Blockdiagonalmatrix S bestehend aus Blöcken S_k aus der symmetrischen Singulärwertzerlegung von $V_k^T W_k = S_k S_k^T$.*

Beweis: s. [4].

Bemerkung 2.19. Zusammenhänge zwischen Diagonalelementen und Singulärwerten komplex symmetrischer Matrizen werden in [40] untersucht.

2.5 Charakterisierung der Singulärwerte

Satz 2.20. *Für den größten Singulärwert $\sigma_1(A)$ einer komplex symmetrischen Matrix A gilt ([23], S. 213, Exercise 5., analog dem Rayleigh-Ritz Theorem für hermitesche Matrizen):*

$$\sigma_1(A) = \max_{x^H x=1} |x^T A x|.$$

Beweis: Sei $A = V\Sigma V^T$ eine Takagi-Faktorisierung von A. Wir zeigen zunächst, dass $|x^T A x| \leq \sigma_1 x^H x$ ist. Sei $x \in \mathbb{C}^n$ beliebig und $y = V^T x$, dann gilt $\|x\|_2 = \|y\|_2$ und damit

$$\begin{aligned} |x^T A x| &= |x^T V \Sigma V^T x| = |y^T \Sigma y| \\ &= |\sum_{k=1}^{n} \sigma_k y_k^2| \leq \sum_{k=1}^{n} \sigma_k |y_k^2| \\ &\leq \sum_{k=1}^{n} \sigma_1 |y_k|^2 = \sum_{k=1}^{n} \sigma_1 |x_k|^2 = \sigma_1 x^H x. \end{aligned}$$

Für $x \neq 0$ erhalten wir insbesondere

$$\frac{|x^T A x|}{x^H x} \leq \sigma_1.$$

Wählen wir $x \in \mathbb{C}^n$, so dass $Ax = \sigma_k \overline{x}$ gilt, so erhalten wir

$$|x^T A x| = |x^T \sigma_k \overline{x}| = \sigma_k x^T \overline{x} = \sigma_k x^H x.$$

Das Maximum wird also für den Vektor x, der konjugierter Singulärvektor der symmetrischen Singulärwertzerlegung zum größten Singulärwert ist, angenommen.
Schließlich gilt

$$\frac{|x^T A x|}{x^H x} = \left| \left(\frac{x}{\sqrt{x^H x}} \right)^T A \left(\frac{x}{\sqrt{x^H x}} \right) \right|$$

und

$$\left(\frac{x}{\sqrt{x^H x}} \right)^H \left(\frac{x}{\sqrt{x^H x}} \right) = 1,$$

daher gilt auch $\max_{x^H x=1} |x^T A x| = \sigma_1$. □

Eine andere Darstellung für den maximalen Singulärwert ist somit auch

$$\begin{aligned}\max_{x^Hx=1} |x^TAx| &= \max_{x^Hx=1} \sqrt{(x^TAx)\overline{(x^TAx)}} \\ &= \max_{x^Hx=1} \sqrt{(x^TAx)(\overline{x}^T\overline{A}\overline{x})} \\ &= \max_{x^Hx=1} \sqrt{(\overline{x}^HAx)(x^HA^H\overline{x})}.\end{aligned}$$

Für den kleinsten Singulärwert von komplex symmetrischen Matrizen muss keine analoge Extremaleigenschaft wie für den größten Singulärwert gelten. Es kann ein Vektor x auftreten, so dass $p = \Sigma^{1/2}V^Tx$ quasi-null ist. In diesem Fall ist $|x^TAx|/(x^Hx)$ minimal, nämlich 0, aber kein Singulärwert, wenn A regulär ist.

2.6 Ungleichungen für Singulärwerte

Die in diesem Abschnitt aufgeführten Sätze sind in [24] für rechteckige Matrizen bewiesen. Da wir sie für komplex symmetrische Matrizen anwenden wollen, beschränken wir uns hier in den Aussagen auf quadratische Matrizen C und $D \in \mathbb{C}^{n\times n}$.

Die folgenden Sätze über Produkte und Summen von Singulärwerten benötigen wir in Kapitel 4.

Satz 2.21. *Für zwei Matrizen $C \in \mathbb{C}^{n\times n}$ und $D \in \mathbb{C}^{n\times n}$ und die Matrix CD mit entsprechenden Singulärwerten $\sigma_k(C)$, $\sigma_k(D)$ bzw. $\sigma_k(CD)$ gilt:*

$$\prod_{k=1}^{l} \sigma_k(CD) \leq \prod_{k=1}^{l} \sigma_k(C)\sigma_k(D) \text{ für } l = 1, \ldots, n \tag{2.3}$$

und

$$\prod_{k=1}^{n} \sigma_k(CD) = \prod_{k=1}^{n} \sigma_k(C)\sigma_k(D). \tag{2.4}$$

Beweis: s. [24].

Satz 2.22. *Gilt für die Singulärwerte $\sigma_k(C)$ und $\sigma_k(D)$ zweier Matrizen $C \in \mathbb{C}^{n\times n}$ und $D \in \mathbb{C}^{n\times n}$*

$$\prod_{k=1}^{l} \sigma_k(C) \leq \prod_{k=1}^{l} \sigma_k(D) \text{ für } l = 1, \ldots, n, \tag{2.5}$$

so folgt

$$\sum_{k=1}^{l} \sigma_k(C) \leq \sum_{k=1}^{l} \sigma_k(D) \text{ für } l = 1, \ldots, n. \tag{2.6}$$

Beweis: s. [24].

Die Produktungleichungen der Singulärwerte in (2.5) stellen eine stärkere Bedingung als die Summenungleichungen in (2.6) dar [24].
Der folgende Satz liefert Ungleichungen zwischen den Summen aufeinander folgender Eigenwerte und Singulärwerte.

Satz 2.23. *Sei $p > 0$ und seien $\sigma_k(C)$, $\lambda_k(C)$ die Singulärwerte bzw. Eigenwerte der Matrix C. Dann gilt für $l = 1, \ldots, n$:*

$$\sum_{k=1}^{l} |\lambda_k(C)|^p \leq \sum_{k=1}^{l} \sigma_k^p(C). \tag{2.7}$$

Beweis: s. [24].

Wir werden nur die Spezialfälle $p = 1$ und $p = 2$ benötigen. Insbesondere lässt sich der folgende Satz über die Beziehung zwischen Singulärwerten und Eigenwerten bei normalen Matrizen herleiten.

Satz 2.24. *Eine Matrix C mit Singulärwerten $\sigma_k(C)$ und Eigenwerten $\lambda_k(C)$ ist genau dann normal, wenn $|\lambda_k(C)| = \sigma_k(C)$ für $k = 1, \ldots, n$ gilt.*

Beweis: Eine Matrix C ist nach Satz 2.14 genau dann normal, wenn für $k = 1, \ldots, n$ gilt:

$$\sum_{k=1}^{n} |\lambda_k(C)|^2 = \sum_{k=1}^{n} \sigma_k^2(C).$$

Ist C normal, so ist obige Gleichung erfüllt, und zusammen mit Satz 2.23 für $p = 2$ folgt (2.7). Dies beweist die eine Richtung.
Gilt $|\lambda_k(C)| = \sigma_k(C)$ für $k = 1, \ldots, n$, so ist die Summengleichung aus Satz 2.14 erfüllt und damit C normal. □

Kapitel 3

CSYM-Iterationsverfahren

Nach einer kurzen Einführung in Iterationsverfahren stellen wir zunächst Krylov-Unterraum-Verfahren vor.
Von diesen Verfahren zeichnet sich zum Beispiel das MINRES-Verfahren durch kurze Rekursionen und die sogenannte Minimales-Residuum-Eigenschaft aus. Es setzt dafür aber eine hermitesche Matrix A des zu lösenden Gleichungssystems voraus. Mit dem CG-Verfahren und dem CGNR-Verfahren, das sich aus diesem ableitet, stellen wir zwei dieser Krylov-Unterraum-Verfahren vor. Sie werden sich für das Verständnis des später vorgestellten CSYM-Verfahrens als nützlich erweisen. Dann werden wir zwei Verfahren für komplex symmetrische Matrizen genauer untersuchen. Wir werden sehen, dass das CSYM-Verfahren die Minimales-Residuum-Eigenschaft erfüllt, obwohl es kein Krylov-Unterraum-Verfahren ist. Einige Eigenschaften des CSYM-Verfahrens lassen sich jedoch von den Eigenschaften des CGNR-Verfahrens ableiten. Das CQMR-Verfahren (**C**SYM **QMR**), das zu den Krylov-Unterraum-Verfahren gehört, erfüllt eine schwächere Bedingung als die Minimales-Residuum-Eigenschaft.

3.1 Iterationsverfahren

Gegeben sei ein lineares Gleichungssystem $Ax = b$ mit einer regulären Matrix $A \in \mathbb{C}^{n\times n}$.

Notation. Es ist $x_* = A^{-1}b$ die exakte Lösung des linearen Gleichungssystems $Ax = b$.

Ein Iterationsverfahren (V) startet mit einem Vektor $x_0 \in \mathbb{C}^n$, häufig $x_0 = 0$.
Für $m = 1, 2, \ldots$ bis zur gewünschten Genauigkeit wird in dem Iterationsschritt m eine Iterierte x_m aus den Vorgängern $x_{m-1}, \ldots, x_{m-t}$ berechnet. Wir werden nur Iterationsverfahren mit $t = 1$, sogenannte Iterationsverfahren erster Stufe, die keine zusätzlichen Parameter zur Verfahrenssteuerung benötigen, betrachten.

Definition 3.1. Sei $m \in \mathbb{N}_0$. Im m-ten Iterationsschritt ist das **Residuum** r_m durch $r_m = b - Ax_m$ und der **Fehler** $e_m(x)$ durch $e_m(x) = A^{-1}b - x_m$ gegeben.

Insbesondere gilt für jede Norm $\|\cdot\|$ die Ungleichung

$$\|e_m(x)\| = \|A^{-1}r_m\| \leq \|A^{-1}\|\|r_m\|.$$

Das Verfahren wird abgebrochen, sobald die Größe $\|r_m\|$ eine vorgegebene Schranke unterschreitet. Die Norm des Fehlers $\|e_m(x)\|$ ist höchstens um den Faktor $\|A^{-1}\|$ größer als $\|r_m\|$.

3.1.1 Wahl der Iterierten

Wir betrachten Iterationsverfahren, die in jedem Iterationsschritt einen Unterraum U vergrößern und aus dem affinen Unterraum x_0+U eine optimale Iterierte bezüglich einer „geeigneten“ Norm $\|\cdot\|$ bestimmen. Die neue Approximation x_m soll $\|r_m\| \leq \|r_{m-1}\|$ erfüllen, d.h. die Folge $\{\|r_m\|\}$ soll eine Nullfolge sein. Dies bedeutet, dass dann auch die Folge $\{\|e_m(x)\|\}$ monoton fallend ist.

Definition 3.2. Eine Iterierte $x_m \in x_0+U$ erfüllt die **Min**imales-**Res**iduum-Eigenschaft (MINRES) bezüglich eines Unterraumes U in einer Norm $\|\cdot\|$, wenn gilt

$$\|r_m\| = \|b - Ax_m\| = \min_{x \in x_0+U} \|b - Ax\|.$$

Bemerkung 3.3. Wir wählen als Norm, in der die MINRES-Eigenschaft gelten soll, i.A. die euklidische Norm.

Wir werden hauptsächlich Verfahren betrachten, die eine solche Optimalitätseigenschaft der Iterierten erfüllen.
Eine schwächere Forderung ist, dass die Iterierte x_m die sogenannte Galerkin-Bedingung erfüllen soll, d.h.

$$(b - Ax_m) \perp U.$$

3.1.2 Konvergenz

Definition 3.4. Ein Iterationsverfahren heißt **konvergent**, wenn für jeden Startvektor x_0 gilt:

$$\lim_{m\to\infty} r_m = 0 \Longleftrightarrow \lim_{m\to\infty} x_m = x_* = A^{-1}b.$$

Als Qualitätsmaß für die Konvergenz werden für die hier zu betrachtenden Verfahren typischerweise Abschätzungen der Gestalt

$$\|r_m\| \leq \alpha c^m \|r_0\|$$

mit α klein und $c \in (0, 1)$ herangezogen.

3.1.3 Krylov-Unterraum-Verfahren

Eine häufig genutzte Klasse von Iterationsverfahren sind Krylov-Unterraum-Verfahren, die wir hier kurz einführen.

Definition 3.5. Unter einem **Krylov-Unterraum der Stufe m** bzgl. A und einem Vektor $r \in \mathbb{C}^n$ verstehen wir den Unterraum

$$K_m(A, r) = \text{span}\left(r, Ar, A^2r, \ldots, A^{m-1}r\right) = \{q(A)r, q \in \mathcal{P}_{m-1}\}.$$

Natürlich kann die Dimension eines Krylov-Unterraumes nicht größer als n werden. Auskunft über die maximale Dimension eines Krylov-Unterraumes gibt das folgende Lemma.

Lemma 3.6. *Zu $r \in \mathbb{C}^n$ existiert ein $m_* \in \mathbb{N}_0$, $m_* \leq n$, so dass*

$$\dim(K_m(A, r)) = m \text{ für } m = 1, \ldots, m_*,$$
$$\dim(K_m(A, r)) = m_* \text{ für } m \geq m_*$$

und ein Polynom $p_{m_*} \in \mathcal{P}_{m_*}$ mit $p_{m_*}(A)r = 0$. Dabei ist $m_* \leq n$ der minimale Grad aller Polynome mit $p(A)r = 0$.

Beweis: s. [34].

Lemma 3.7. *Die Lösung $x_* = A^{-1}b$ erfüllt mit $r_0 = b - Ax_0$*

$$x_* \in x_0 + K_{m_*}(A, r_0) \text{ und}$$
$$x_* \notin x_0 + K_m(A, r_0) \text{ für alle } m < m_*.$$

Beweis: Sei $p_{m_*}(\cdot)$ aus Lemma 3.6 gegeben durch $p_{m_*}(t) = \sum_{k=0}^{m_*} \gamma_k t^k$. Wir zeigen zunächst durch einen Widerspruchsbeweis, dass γ_0 nicht verschwindet.
Angenommen $p_{m_*}(\cdot)$ ist gegeben durch $p_{m_*}(t) = \sum_{k=1}^{m_*} \gamma_k t^k$, dann gilt

$$A\left(\sum_{k=1}^{m_*} A^{k-1}r\right) = 0.$$

Da A eine reguläre Matrix ist, folgt

$$\left(\sum_{k=1}^{m_*} A^{k-1}r\right) = 0.$$

Dann würde aber $\tilde{p}(A)r = 0$ für ein Polynom $\tilde{p} \in \mathcal{P}_{m_*-1}$ gelten. Dieses widerspricht der Minimalität von p_{m_*}.
Aus $p_{m_*}(A)r_0 = 0$ folgt

$$A^{-1} \sum_{k=0}^{m_*} \gamma_k A^k (b - Ax_0) = 0.$$

Wir ziehen $\gamma_0(b - Ax_0)$ aus der Summe, lösen nach $x_* = A^{-1}b$ auf und erhalten

$$x_* = x_0 - \frac{1}{\gamma_0} \sum_{k=1}^{m_*} \gamma_k A^{k-1} r_0.$$

Somit ist

$$x_* \in x_0 + K_{m_*}(A, r_0).$$

Angenommen $x_* \in x_0 + K_m(A, r_0)$ mit $m < m_*$, dann existiert ein Polynom p_m mit einem Grad $m < m_*$. Dies widerspricht der Minimalität von p_{m_*}. □

Mit Hilfe der Krylov-Unterräume können wir nun die Verfahren definieren.

Definition 3.8. Ein **Krylov-Unterraum-Verfahren** (KUV) zur Lösung von $Ax = b$ ist ein Verfahren, für dessen Iterierte x_m gilt:

$$x_m \in x_0 + K_m(A, r_0),\ m = 0, 1, \ldots.$$

Lemma 3.9. *Gilt $x_m = x_0 + q_{m-1}(A)r_0$ mit einem $q_{m-1} \in \mathcal{P}_{m-1}$, dann gilt*

$$r_m = p_m(A)r_0 \text{ mit } p_m(t) = 1 - tq_{m-1}(t) \text{ und } p_m(0) = 1.$$

Beweis: Wir setzen $x_m = x_0 + q_{m-1}(A)r_0$ in die Darstellung von r_m ein und erhalten

$$\begin{aligned} r_m &= b - A(x_0 + Aq_{m-1}(A)r_0) \\ &= r_0 - Aq_{m-1}(A)r_0 \\ &= (I - Aq_{m-1}(A))r_0. \end{aligned}$$

Somit folgt

$$r_m = p_m(A) \text{ mit } p_m(t) = 1 - tq_{m-1}(t),\ p_m(0) = 1.$$

□

Aus Lemma 3.9 folgt, dass $r_m \in K_{m+1}(A, r_0)$ gilt.

Definition 3.10. Die Polynome p_m und q_{m-1} heißen **Verfahrenspolynome**.

Insbesondere legen die Verfahrenspolynome ein spezielles Krylov-Unterraum-Verfahren fest. Eine Optimalitätseigenschaft für die x_m definiert die Verfahrenspolynome.

Bemerkung 3.11. Für ein Krylov-Unterraum-Verfahren, das die MINRES-Eigenschaft in der euklidischen Norm erfüllt, werden die Verfahrenspolynome so gewählt, dass

$$\|r_m\|_2 = \min_{p \in \mathcal{P}_m} \|p(A)r_0\|_2 = \min_{q \in \mathcal{P}_{m-1}} \|r_0 - Aq(A)r_0\|_2.$$

Konvergenz

Das Konvergenzverhalten eines konkreten Krylov-Unterraum-Verfahrens hängt von der Wahl der Verfahrenspolynome, der Matrix A des zu lösenden linearen Gleichungssystems und r_0 und damit vom Startvektor x_0 ab. Wir müssen zwischen der Konvergenz in exakter Arithmetik und der Konvergenz in finiter Arithmetik unterscheiden. Die Konvergenz von Krylov-Unterraum-Verfahren in exakter Arithmetik ist allgemein gut untersucht. Für normale Matrizen können wir scharfe obere Schranken angeben. Das Konvergenzverhalten der Verfahren lässt sich auf die Approximation von 0 durch die Verfahrenspolynome p_m auf der Menge der Eigenwerte zurückführen. In die Konvergenzgeschwindigkeit geht die Kondition $\operatorname{cond}(A)$ der Matrix A ein.

Basis des Krylov-Unterraumes

In diesem Abschnitt beschäftigen wir uns mit der Bestimmung einer geeigneten Basis eines Krylov-Unterraumes, d.h. der Wahl von Vektoren $q_1, \ldots, q_{m+1}$, so dass

$$K_{m+1}(A, r_0) = \operatorname{span}(q_1, \ldots, q_{m+1}) = \operatorname{span}\left(r_0, Ar_0, A^2 r_0, \ldots, A^m r_0\right).$$

Zur Expansion des Krylov-Unterraumes wird im Iterationsschritt $m+1$ ein neuer Vektor

$$q_{m+1} \in K_{m+1}(A, r_0) \setminus K_m(A, r_0)$$

berechnet. Da wir dazu $\tilde{q}_{m+1} = Aq_m$ berechnen müssen, beträgt der Rechenaufwand auf jeden Fall eine Matrix-Vektormultiplikation mit A.
Die Wahl der Basisvektoren eines Krylov-Unterraumes hat zum einen Einfluss auf die Konvergenz des Verfahrens in finiter Arithmetik. Andererseits beeinflusst sie aber auch den Aufwand zur Berechnung der Basisvektoren. Nicht zu vernachlässigen ist auch der Speicherbedarf für die Basisvektoren. Die Wahl der Basis bestimmt auch, welche Optimalitätsbedingung für x_{m+1} mit konstantem Aufwand pro Iterationsschritt realisierbar

ist.
Wir beschränken uns bei allen folgenden Ausführungen über die Basisgenerierung auf die Darstellung des Prozesses in Matrixnotation. Dass das Verfahren die gewünschten Eigenschaften der Basisvektoren liefert, wird jeweils in den angegebenen Quellen gezeigt.
Orthonormale Basisvektoren des Krylov-Unterraumes erhalten wir durch Anwendung einer der Varianten des Orthogonalisierungsverfahrens von E. Schmidt, das sogenannte Arnoldi-Verfahren [2]. Seien die orthonormalen Basisvektoren als Spalten in einer Matrix $Q_m = [q_1, \ldots, q_m]$ angeordnet und $Q_{m+1} = [Q_m, q_{m+1}]$.

Definition 3.12. Eine Matrix A heißt eine **obere Hessenbergmatrix**, falls $a_{i,j} = 0$ für $i > j + 1$.

Sei $H_{m,m} \in \mathbb{C}^{m\times m}$ eine obere Hessenbergmatrix und

$$H_{m+1,m} = \begin{bmatrix} H_{m,m} \\ \beta e_m^T \end{bmatrix} \in \mathbb{C}^{(m+1)\times m} \text{ mit } \beta \in \mathbb{C}.$$

Der Orthogonalisierungsprozess kann in Matrixschreibweise dargestellt werden als

$$AQ_m = Q_{m+1}H_{m+1,m} \tag{3.1}$$

und liefert eine Projektion der Matrix A durch Q_m auf die obere Hessenbergmatrix $H_{m,m}$, denn aus (3.1) folgt

$$Q_m^H AQ_m = [I_m|0]H_{m+1,m} = H_{m,m}. \tag{3.2}$$

Bemerkung 3.13. Die Matrix $H_{m,m}$ ist im schlechtesten Fall eine voll besetzte obere Hessenbergmatrix, was für die Berechnung von q_{m+1} den Zugriff auf alle vorherigen Basisvektoren $q_1, \ldots, q_m$ bedeutet. Dies ist der Fall beim GMRES-Verfahren (Generalized MINRES)[35] für allgemeine Matrizen. Der Aufwand zur Berechnung des neuen Vektors nimmt dabei in jedem Iterationsschritt zu.
Der neue Basisvektor soll sich aus einigen wenigen s Basisvektoren $q_m, \ldots, q_{m-s+1}$, also über eine möglichst kurze Rekursion, berechnen lassen. Damit beschränken wir sowohl den Speicherbedarf als auch den Rechenaufwand. Krylov-Unterraum-Verfahren, die dies erfüllen, basieren auf Varianten des Lanczos-Verfahrens [26]. Die Kürze der Rekursion hängt insbesondere von dem Typ der Matrix A ab. Hermitesch positiv definite Matrizen erweisen sich als vorteilhaft, da sich besonders kurze Rekursionen finden lassen. Die Wahl der Basisvektoren sollte auch in Hinblick auf die Optimalität der Iterierten $x_m \in x_0 + K_m(A, r_0)$ erfolgen. Sie soll mit wenig Aufwand aus der Vorgänger-Iterierten berechenbar sein.

Lemma 3.14. *Sei $q_1 = r_0/\|r_0\|_2$. Gilt mit orthonormalen Basisvektoren $q_1, \ldots, q_m, q_{m+1}$ in der Notation $Q_m = [q_1, \ldots, q_m]$, $Q_{m+1} = [Q_m, q_{m+1}]$ die Darstellung*

$$AQ_m = Q_{m+1}H_{m+1,m},$$

so erfüllt $x_m = x_0 + Q_m z_m$ bezüglich des von den Spalten von Q_m aufgespannten Unterraumes die MINRES-Eigenschaft, wenn der Vektor $z_m \in \mathbb{C}^n$ die Minimierungsaufgabe $\min_{z\in\mathbb{C}^m} \|\|r_0\|_2 e_1 - H_{m+1,m}z\|_2$ löst.

Beweis: Sei $x_m = x_0 + Q_m z_m$ gewählt, so dass $\|r_m\|_2$ minimal in dem von den Basisvektoren $q_1, \ldots, q_m$ aufgespannten Unterraum ist. Dann erhalten wir

$$\begin{aligned}
\|r_m\|_2 &= \|b - Ax_m\|_2 \\
&= \|b - Ax_0 - A(Q_m z_m)\|_2 \\
&= \|r_0 - Q_{m+1}H_{m+1,m}z_m\|_2 \\
&= \|Q_{m+1}(Q_{m+1}^H r_0 - H_{m+1,m}z_m)\|_2 \\
&= \|\|r_0\|_2 e_1 - H_{m+1,m}z_m\|_2 .
\end{aligned}$$

□

Die Minimierungsaufgabe $\min_{z\in\mathbb{C}^m} \|\|r_0\|_2 e_1 - H_{m+1,m}z\|_2$ ist ein kleinstes Quadrate-Problem bei überbestimmtem Gleichungssystem. Der minimierende Vektor z_m lässt sich mit Hilfe einer QR-Zerlegung von $H_{m+1,m}$ berechnen. Die Faktoren der QR-Zerlegung von $H_{m+1,m}$ können unter Verwendung von Givens-Rotationen [14] aus der QR-Zerlegung von $H_{m,m-1}$ des vorigen Iterationsschrittes aufdatiert werden.

Das MINRES-Verfahren

Unter der Bedingung, dass A hermitesch ist, kann das MINRES-Verfahren [31] durchgeführt werden. Hierbei wird eine Orthonormalbasis über das hermitesche Lanczos-Verfahren konstruiert. Die Aufwände zur Berechnung der Basisvektoren und zur Aktualisierung der Iterierten mit MINRES-Eigenschaft sind pro Iterationsschritt konstant. Der konstante Aufwand rührt daher, dass die Matrix $H_{m,m}$ in (3.2) im MINRES-Verfahren eine Tridiagonalmatrix ist. Auch hier ist nur eine Matrix-Vektormultiplikation pro Iterationsschritt erforderlich.

Diese günstigen Eigenschaften können auch für Verfahren mit komplex symmetrischen Matrizen gelten, wobei das entwickelte Verfahren, wie wir sehen werden, aus der Klasse der Krylov-Unterraum-Verfahren herausführt.

Das MINRES-Verfahren kann insbesondere für reell symmetrische Matrizen angewendet werden, und damit auch für komplex symmetrische. Dazu wird das komplexe Gleichungssystem $Ax = b$ mit komplex symmetrischer Matrix $A = B + iC$ in ein reelles Gleichungssystem $\tilde{A}\tilde{x} = \tilde{b}$ mit reell symmetrischer Matrix $\tilde{A}$ doppelter Größe umgeformt:

$$\tilde{A} = \begin{pmatrix} B & C \\ C & -B \end{pmatrix}, \ \tilde{x} = \begin{pmatrix} \mathrm{Re}\,(x) \\ -\mathrm{Im}\,(x) \end{pmatrix}, \ \tilde{b} = \begin{pmatrix} \mathrm{Re}\,(b) \\ \mathrm{Im}\,(b) \end{pmatrix}.$$

Die Matrix $\tilde{A}$ ist symmetrisch, da B und C reell symmetrisch sind. In der Praxis ist das Verfahren nicht immer zu empfehlen; Analysen hierzu liefert Freund [11].

Das CGNR-Verfahren

Im Folgenden führen wir das CGNR-Verfahren ein. Es ist von Relevanz insofern, als dass sich Konvergenzaussagen dieses Verfahrens auf die des CSYM-Verfahrens übertragen lassen.
Dazu führen wir zunächst eine spezielle Norm auf hermitesch positiv definiten Matrizen und das CG-Verfahren ein.

Notation. Sei A eine hermitesch positiv definite Matrix. Wir notieren die A-Norm, die gegeben ist durch $\sqrt{v^H A v}$, mit $\|\cdot\|_A$.

Definition 3.15. Das **CG-Verfahren** ist ein Krylov-Unterraum-Verfahren, das die Iterierte x_m bestimmt, so dass $\|e_m(x)\|_A$ minimal ist.

Satz 3.16. *Das CG-Verfahren kann durch kurze Rekursionen aufgebaut werden.*

Beweis: s. [18] durch Angabe des konkreten Algorithmus.

Bemerkung 3.17. Die Matrix $H_{m,m}$ in (3.2) ist im CG-Verfahren eine hermitesch positiv definite Tridiagonalmatrix.

Im Folgenden geben wir zwei Sätze über die Konvergenz des CG-Verfahrens an.

Satz 3.18. *Es sei M die Anzahl der verschiedenen Eigenwerte der hermitesch positiv definiten Matrix A. Dann gilt für die $\overline{M}$-te Iterierte der CG-Verfahrens $x_{\overline{M}} = x_* = A^{-1}b$ mit $\overline{M} \leq M$.*

Beweis: Seien $\lambda_1, \ldots, \lambda_M$ die Eigenwerte von A. Das Minimalpolynom von A ist gegeben durch $p(t) = \prod_{k=1}^{M}(t - \lambda_k)$. Da $p(A) = 0$, folgt insbesondere $p(A)r_0 = 0$ und damit $\|r_{\overline{M}}\|_2 = 0$ und $x_{\overline{M}} = x_*$ für ein $\overline{M} \leq M$ auf Grund der Optimalitätseigenschaft. □

In finiter Arithmetik wird dies nicht erreicht. Der folgende Satz liefert eine Aussage über die Konvergenzgeschwindigkeit.

Satz 3.19. *Für die Iterierten des CG-Verfahrens gilt mit* $c = \frac{\sqrt{cond(A)}-1}{\sqrt{cond(A)}+1}$ *in der A-Norm die Abschätzung*

$$\|x_m - x_*\|_A \leq \frac{2c^m}{1 + c^{2m}}\|x_0 - x_*\|_A \leq 2c^m\|x_0 - x_*\|_A.$$

Beweis: über Chebyscheff-Polynome [34].

Wir können das CG-Verfahren im Prinzip für beliebige Matrizen anwenden [16], indem wir vom Gleichungssystem $Ax = b$ auf die Normalengleichung $A^HAx = A^Hb$ übergehen.

Definition 3.20. Das implizit auf ein Gleichungssystem

$$\hat{A}x = \hat{b} \text{ mit } \hat{A} = A^HA \text{ und } \hat{b} = A^Hb$$

angewendete CG-Verfahren heißt **CGNR**-Verfahren (**C**onjugate **G**radient **N**ormal equation for the **R**esidual).

Bemerkung 3.21. Die Matrix A^HA braucht im CGNR-Verfahren nicht berechnet zu werden.

Bemerkung 3.22. Der Unterraum des CGNR-Verfahrens ist gegeben durch

$$K_m(A^HA, A^Hr_0) = \text{span}\left(A^Hr_0, (A^HA)A^Hr_0, \ldots, (A^HA)^{m-1}A^Hr_0\right).$$

Lemma 3.23. *Für das CGNR-Verfahren gilt im m-ten Iterationsschritt*

$$x_m = x_0 + q_{m-1}(A^HA)A^Hr_0 \text{ mit } q_{m-1} \in \mathcal{P}_{m-1} \text{ und}$$
$$r_m = p_m(AA^H)r_0 \text{ mit } p_m(t) = 1 - tq_{m-1} \in \mathcal{P}_m.$$

Beweis: Aus Lemma 3.9 folgt zusammen mit Bemerkung 3.22

$$x_m = x_0 + q_{m-1}(A^HA)A^Hr_0$$

und damit

$$\begin{aligned} r_m &= b - Ax_0 - Aq_{m-1}(A^HA)A^Hr_0 \\ &= r_0 - (AA^H)q_{m-1}(AA^H)r_0. \end{aligned}$$

□

Bemerkung 3.24. Man beachte, dass in Lemma 3.23 in der Polynom-Darstellung von r_m die Matrix AA^H statt A^HA steht.

Lemma 3.25. *Die Iterierte x_m des CGNR-Verfahrens besitzt die MINRES-Eigenschaft bezüglich des Unterraumes $K_m(A^HA, A^Hr_0)$.*

Beweis: Das CG-Verfahren angewendet auf die Normalengleichung minimiert nach der Definition 3.15 die Norm des Fehlers in der A^HA- Norm. Nun gilt immer

$$\begin{aligned} \|e_m(x)\|_{A^HA} &= \sqrt{e_m(x)^HA^HAe_m(x)} \\ &= \sqrt{(Ae_m(x))^H(Ae_m(x))} \\ &= \|r_m\|_2. \end{aligned}$$

Also wird implizit die Größe $\|r_m\|_2$ minimiert. □

Bemerkung 3.26. Im CGNR-Verfahren wird ein Polynom $p \in \mathcal{P}_m$ bestimmt, so dass $\|r_m\|_2 = \|p(AA^H)r_0\|_2$ minimal ist. Aus den Konvergenzaussagen über das CG-Verfahren lassen sich folgende Konvergenzaussagen über das CGNR-Verfahren ableiten. Aus Satz 3.18 folgt:

Lemma 3.27. *Sei M die Anzahl der verschiedenen Eigenwerte von A^HA, d.h. der verschiedenen Singulärwerte von A. Dann gilt für die $\overline{M}$-te Iterierte des CGNR-Verfahrens $x_{\overline{M}} = x_* = A^{-1}b$ mit $\overline{M} \leq M$.*

Lemma 3.28. *Für die Iterierten des CGNR-Verfahrens gilt mit $c = \frac{cond(A)-1}{cond(A)+1}$ die Abschätzung*

$$\|r_m\|_2 \leq \frac{2c^m}{1+c^{2m}}\|r_0\|_2 \leq 2c^m\|r_0\|_2.$$

Beweis: Zu zeigen ist, dass $c = \frac{\sqrt{\text{cond}\,(A^HA)}-1}{\sqrt{\text{cond}\,(A^HA)}+1}$ obige Darstellung besitzt. Nun gilt

$$\begin{aligned}\text{cond}\,(A^HA) &= \frac{\sigma_1(A^HA)}{\sigma_n(A^HA)} = \frac{\sigma_1(A)^2}{\sigma_n(A)^2}\\ \Longrightarrow \sqrt{\text{cond}\,(A^HA)} &= \sqrt{\text{cond}\,(A)^2} = \text{cond}\,(A).\end{aligned}$$

Daher gilt $c = \frac{\text{cond}\,(A)-1}{\text{cond}\,(A)+1}$ und mit Satz 3.19 und Lemma 3.25 folgt die Behauptung. □

Die Anwendung ist in der Praxis nicht empfehlenswert, einmal wegen des erhöhten Rechenaufwandes von zwei Matrix-Vektormultiplikationen pro Iterationsschritt, andererseits weil die Konvergenz von den Eigenwerten von A^HA abhängt.

Bemerkung 3.29. Im CGNE-Verfahren (**C**onjugate **G**radient **N**ormal equation for the **E**rror) wird das CG-Verfahren implizit auf das lineare Gleichungssystem $AA^Hy = b$ angewendet. Sei $y_* = (AA^H)^{-1}b$ die exakte Lösung des Gleichungssystems $AA^Hy = b$. Die Iterierte y_m wird nicht explizit berechnet, sondern direkt $x_m = A^Hy_m$. Das CGNE-Verfahren minimiert den Fehler $e_m(x)$, denn mit $e_m(y) = y_* - y_m$ und $A^He_m(y) = e_m(x)$ gilt

$$\begin{aligned}\|e_m(y)\|_{AA^H} &= \sqrt{(e_m(y))^H(AA^H)(e_m(y))}\\ &= \sqrt{(A^He_m(y))^H(A^He_m(y))}\\ &= \|e_m(x)\|_2.\end{aligned}$$

3.2 Das CSYM-Verfahren

Im Folgenden betrachten wir zwei Verfahren, die die komplexe Symmetrie der Ausgangsmatrix beibehalten, indem sie bei dem sukzessiven Aufbau der Basis die Ausgangsmatrix auf eine komplex symmetrische Tridiagonalmatrix projizieren. Dies geschieht jedoch auf unterschiedliche Weise.
Für eine Projektion auf eine Tridiagonalmatrix durch T-Kongruenz mit einer „günstigen“ regulären Matrix bieten sich folgende zwei Möglichkeiten an.

- Man nimmt eine unitäre Matrix Q, so dass $T = Q^TAQ = \overline{Q}^{-1}AQ$. Es handelt sich also um eine con-Ähnlichkeitstransformation mit einer unitären Matrix. Die Singulärwerte von T sind daher mit den Singulärwerten von A identisch. Eine solche Matrix Q lässt sich immer finden [5].

- Man nimmt eine komplex-orthogonale Matrix P, so dass $\tilde{T} = P^T AP = P^{-1}AP$. Es handelt sich um eine Ähnlichkeitstransformation, was dem üblichen Vorgehen bei Krylov-Unterraum-Verfahren entspricht. Die Eigenwerte von $\tilde{T}$ sind mit den Eigenwerten von A identisch. Eine solche Matrix P muss es nicht geben.

Vollständige Tridiagonalisierungen sind für die iterative Lösung von Gleichungssystemen nicht notwendig. Wichtig ist, dass der Tridiagonalisierungsprozess für den gewählten Startvektor so lange durchgeführt werden kann, bis eine ausreichend gute Approximation der Lösung im aufgebauten Unterraum ermittelt werden kann.
Das CSYM-Verfahren wurde von Bunse-Gerstner und Stöver [5] 1999 entwickelt. Obwohl der Aufbau des CSYM-Verfahrens große Ähnlichkeit mit dem der Krylov-Unterraum-Verfahren hat, wird die Iterierte aus einem komplizierter aufgebauten Unterraum ermittelt. Dennoch besitzt es einige der gewünschten Eigenschaften des MINRES-Verfahrens, wie die MINRES-Eigenschaft, eine 3-Term-Rekursion und nur eine Matrix-Vektormultiplikation pro Iterationsschritt. Das CSYM-Verfahren setzt sich aus folgenden Schritten zusammen:

- Wahl eines Startvektors $x_0 \in \mathbb{C}^n$.
- Setzen von $q_1 = \overline{r}_0/\|r_0\|_2$.
- Sukzessive Konstruktion der Orthogonalbasis über Tridiagonalisierung durch „unvollständige“ T-Kongruenz mit Matrizen Q_m, die orthonormale Spalten haben, d.h.

$$Q_m^T A Q_m = T_m.$$

- Konstruktion der Iterierten $x_m = x_0 + Q_m z_m$, so dass x_m die MINRES-Eigenschaft im konstruierten Unterraum erfüllt. Insbesondere ist für die Berechnung eine QR-Zerlegung mit Update durch Givens-Rotationen durchführbar.

Wir werden zuerst Tridiagonalisierungen einer komplex symmetrischen Matrix betrachten, wobei wir zunächst das komplex symmetrische Lanczos-Verfahren vorstellen, auf dem das CQMR-Verfahren beruht. Diesem stellen wir dann das Tridiagonalisierungsverfahren gegenüber, auf dem das CSYM-Verfahren basiert.
Das CQMR-Verfahren hat keine Minimales-Residuum-Eigenschaft, sondern die schwächere QMR-Eigenschaft (Quasi-Minimales-Residuum-Eigenschaft).
Eine andere Möglichkeit, das CSYM-Verfahren einzuführen, ergibt sich aus den von Saunders, Simon und Yip [37] entwickelten Quasi-Krylov-Unterraum-Verfahren für allgemeine Matrizen. Daraus lässt sich das CSYM-Verfahren als Spezialfall für komplex symmetrische Matrizen ableiten.

3.2.1 Tridiagonalisierungen

Zunächst stellen wir die Tridiagonalisierung vor, die im CQMR-Verfahren [11] durchgeführt wird. Sie stellt eine Variante des Lanczos-Verfahrens [26] für hermitesche Matrizen dar und ist für eine reell symmetrische Matrix A mit diesem identisch. Anschließend stellen wir das Lanczos-ähnliche Verfahren vor. Bei beiden Verfahren beschränken wir uns auch hier auf die Vorstellung der zugehörigen Algorithmen, der Darstellung in Matrixschreibweise und wichtiger Eigenschaften. Herleitungen und Begründungen, dass die Verfahren die gewünschte Basis erzeugen, sind in [5] bzw. [11] zu finden.

Das komplex symmetrische Lanczos-Verfahren

Ziel ist es, sukzessive eine Folge von komplex-orthogonalen Vektoren $p_1, \ldots, p_m$ zu konstruieren, die jeweils die Basisvektoren von $K_m(A, r_0)$ bilden.
Sei $\hat{p}_{m+1}$ der nächste unnormierte Vektor der Basis und seien die bisherigen Lanczos-Vektoren in einer Matrix $P_m = [p_1, p_2, ..., p_m]$ angeordnet, so ist diese Forderung äquivalent zu $P_m^T \hat{p}_{m+1} = 0$ und $P_m^T P_m = I_m$.
Sei die Tridiagonalmatrix $\tilde{T}_m$ mit $\beta_k = \sqrt{\hat{p}_k^T \hat{p}_k}$ und $\alpha_k = p_k^T A p_k$ für $k = 1, \ldots, m$ gegeben durch:

$$\tilde{T}_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_m \\ & & & \beta_m & \alpha_m \end{pmatrix}. \tag{3.3}$$

Das komplex symmetrische Lanczos-Verfahren lässt sich in Matrixschreibweise darstellen als [11]:

$$AP_m = P_m \tilde{T}_m + \hat{p}_{m+1} e_m^T. \tag{3.4}$$

Die Matrix $\tilde{T}_m$ in (3.3) ist offensichtlich symmetrisch und es gilt $\tilde{T}_m = P_m^T A P_m$. Wenn wir das Verfahren bis m_* (s. Lemma 3.6) laufen lassen können, würde $\tilde{T}_{m_*}$ genau m_* Eigenwerte von A haben. Wie wir nachfolgend sehen werden, kann sich die Tridiagonalisierung bis m_* als nicht durchführbar erweisen. Brechen wir die Tridiagonalisierung für ein $m < m_*$ ab, so approximieren die Eigenwerte von $\tilde{T}_m$ Eigenwerte von A. Daher ist dieses Verfahren auch zur Berechnung von Eigenwerten und Eigenvektoren geeignet [9].

Mit $\hat{p}_{m+1} = \beta_{m+1} p_{m+1}$, $P_{m+1} = [P_m, p_{m+1}]$ und

$$\tilde{T}_{m+1,m} = \begin{bmatrix} \tilde{T}_m \\ \beta_{m+1} e_m^T \end{bmatrix}$$

erhalten wir aus (3.4) die Darstellung

$$AP_m = P_{m+1} \tilde{T}_{m+1,m}. \tag{3.5}$$

Wenn wir in der Matrixschreibweise (3.4) eine Spalte k betrachten und nach dem neuen Vektor $\hat{p}_{k+1}$ auflösen, so erhalten wir

$$\beta_{k+1} p_{k+1} = Ap_k - \alpha_k p_k - \beta_k p_{k-1}.$$

Dieses Konstruktionsverfahren wird in Algorithmus 3.1 umgesetzt.

Algorithmus 3.1 Das komplex symmetrische Lanczos-Verfahren

wähle Vektor $\hat{p}_1 = r_0$ mit $\hat{p}_1^T \hat{p}_1 \neq 0$
setze $p_0 = 0,\ \beta_1 = \sqrt{\hat{p}_1^T \hat{p}_1},\ p_1 = \hat{p}_1 / \beta_1$
for $m = 1, 2, 3 \ldots$ **do**
  $\alpha_m = p_m^T A p_m$
  $\hat{p}_{m+1} = Ap_m - \alpha_m p_m - \beta_m p_{m-1}$
  $\beta_{m+1} = \sqrt{\hat{p}_{m+1}^T \hat{p}_{m+1}}$
  if $\beta_{m+1} = 0$ **then**
    STOP
  else
    $p_{m+1} = \hat{p}_{m+1} / \beta_{m+1}$
  end if
end for

Wir erhalten auch hier eine 3-Term-Rekursion und den Aufwand von einer Matrix-Vektormultiplikation pro Iterationsschritt.
Der wesentliche Unterschied zum Lanczos-Verfahren für hermitesche Matrizen ist, dass die erzeugten Basisvektoren in P_m komplex-orthogonal sind. Bedingt durch diese Eigenschaft der Basisvektoren ist β_{m+1} das Ergebnis eines indefiniten Bilinearproduktes. Tritt in Algorithmus 3.1 ein Vektor $\hat{p}_{m+1} \neq 0$ auf, der quasi-null ist, so bricht das Verfahren ab.

Definition 3.30. Ein **breakdown** bedeutet, dass das Lanczos-Verfahren für ein $m_0 < m_*$ abbricht.

Bemerkung 3.31. Beim Lanczos-Verfahren für hermitesche Matrizen kann ein solcher frühzeitiger Abbruch nicht vorkommen. Das Verfahren kann immer bis $m = m_*$ durchgeführt werden und bricht dann mit einem sogenannten **lucky breakdown** ab, was bedeutet, dass der Krylov-Unterraum die maximale Dimension erreicht hat.

Das komplex symmetrische Lanczos-Verfahren kann im Fall eines breakdowns nicht weitergeführt werden, obwohl noch keine ausreichende Approximation der Lösung x_* berechnet wurde.
Man unterscheidet dabei folgende breakdown-Arten:

- *Near breakdown*, d.h. $\beta_{m+1} \approx 0$,
- *Exact breakdown*, d.h. $\beta_{m+1} = 0$ und $\hat{p}_{m+1} \neq 0$

Mit zusätzlichem Aufwand, der Erweiterung zu einem Look-Ahead-Lanczos-Verfahren [11], lassen sich manchmal diese breakdowns umgehen. Es gibt aber sogenannte *incurable breakdowns*, die mit der Look-Ahead-Variante nicht umgangen werden können.

Satz 3.32. *Angenommen, A ist diagonalisierbar und der Startvektor $\hat{p}_1$ enthält Anteile von allen Eigenvektoren von A. Ein **incurable breakdown** tritt genau dann auf, wenn mindestens ein Eigenvektor von A quasi-null ist.*

Beweis: s. [11].

Eine Möglichkeit, doch noch eine Orthogonalbasis mit gleichem Aufwand wie im symmetrischen Lanczos-Verfahren zu konstruieren und breakdowns zu vermeiden, liefert das folgende Lanczos-ähnliche Verfahren, das speziell für komplex symmetrische Ausgangsmatrizen konstruiert wurde.

Das Lanczos-ähnliche Verfahren

Wir konstruieren orthogonale Vektoren $q_1, \ldots, q_m$, so dass mit $Q_m = [q_1, q_2, ..., q_m]$ und dem unnormierten Vektor $w_{m+1} \in \text{span}\,(\overline{q_1}, \ldots, \overline{q_m})^{\perp}$ die Beziehungen

$$Q_m^H w_{m+1} = 0 \text{ und } Q_m^H Q_m = I_m$$

gelten.

Der Lanczos-ähnliche Prozess lässt sich in Kurznotation in folgender Form darstellen [5]:

$$AQ_m = \overline{Q}_m T_m + w_{m+1} e_m^T \text{ mit} \tag{3.6}$$

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_m \\ & & & \beta_m & \alpha_m \end{pmatrix}, \tag{3.7}$$

$$\text{wobei } \beta_k = \|w_k\|_2,\ \alpha_k = q_k^T A q_k \text{ für } k = 1, \ldots, m. \tag{3.8}$$

Es gilt mit $w_{m+1} = \beta_{m+1}\overline{q_{m+1}}$, $Q_{m+1} = [Q_m, q_{m+1}]$ und

$$T_{m+1,m} = \begin{bmatrix} T_m \\ \beta_{m+1} e_m^T \end{bmatrix}$$

die Darstellung

$$AQ_m = \overline{Q}_{m+1} T_{m+1,m}. \tag{3.9}$$

Die Matrix T_m in (3.7) ist offensichtlich symmetrisch, und aus (3.6) folgt $T_m = Q_m^T A Q_m$. Wenn wir das Verfahren solange laufen lassen bis $w_{m+1} = 0$ ist, so wären die Singulärwerte von T_m auch Singulärwerte von A. So aber approximieren die m Singulärwerte von T_m Singulärwerte von A. Daher ist dieses Verfahren auch zur Berechnung der Singulärwerte und Singulärvektoren geeignet [38]. Die komplette Tridiagonalisierung ist immer durchführbar.
Wenn wir in (3.6) die Spalte k betrachten und nach dem neuen Vektor $w_{k+1} = \beta_{k+1}\overline{q_{k+1}}$ auflösen, erhalten wir

$$\beta_{k+1}\overline{q_{k+1}} = Aq_k - \alpha_k \overline{q_k} - \beta_k \overline{q_{k-1}}.$$

Dieses Verfahren wird in Algorithmus 3.2 umgesetzt.

Algorithmus 3.2 Das komplex symmetrische Lanczos-ähnliche Verfahren

wähle Vektor $w_1 = r_0$
setze $q_0 = 0, \beta_1 = \|w_1\|_2, q_1 = \overline{w}_1/\beta_1$
for $m = 1, 2, 3 \ldots$ **do**
  $\alpha_m = q_m^T A q_m$
  $w_{m+1} = Aq_m - \alpha_m \overline{q_m} - \beta_m \overline{q_{m-1}}$
  $\beta_{m+1} = \|w_{m+1}\|_2$
  if $\beta_{m+1} = 0$ **then**
    STOP
  else
    $q_{m+1} = \overline{w}_{m+1}/\beta_{m+1}$
  end if
end for

Wir erhalten auch hier eine 3-Term-Rekursion und den Aufwand von einer Matrix-Vektormultiplikation pro Iterationsschritt.
Der wesentliche Unterschied zum komplex symmetrischen Lanczos-Verfahren ist, dass die erzeugten Basisvektoren in Q_m orthogonal sind, also die gewünschte Orthogonalität besitzen. Auffallend ist im Vergleich zu Algorithmus 3.1, dass für die Berechnung von w_{m+1} bzw. q_{m+1} auf konjugierte Vektoren zugegriffen wird. Ein weiterer Unterschied ist, dass β_{m+1} das Ergebnis einer Vektornorm ist und daher nur 0 werden kann, wenn w_{m+1} der Nullvektor ist. In diesem Fall hat der generierte Unterraum die maximale Dimension erreicht, die Lösung ist in diesem Unterraum enthalten, und wir sprechen daher, wie auch im Lanczos-Verfahren, von einem lucky breakdown.
Mit Hilfe von Householder-Transformationen wird in [5] gezeigt, dass Algorithmus 3.2 solange durchführbar ist, bis ein lucky breakdown eintritt. Den im Lanczos-ähnlichen Verfahren erzeugten Unterraum werden wir im Folgenden untersuchen.

3.2.2 Der Unterraum des CSYM-Verfahrens

Der Unterraum des CQMR-Verfahrens ist ein Krylov-Unterraum $K_m(A, r_0)$. Das CSYM-Verfahren konstruiert jedoch keinen Krylov-Unterraum, denn es gilt:

Satz 3.33. *Der durch Algorithmus 3.2 konstruierte Unterraum ist gegeben durch*

$$\tilde{K}_m(A, r_0) = K_{\lfloor \frac{m+1}{2} \rfloor}(\overline{A}A, \overline{r}_0) + K_{\lfloor \frac{m}{2} \rfloor}(\overline{A}A, \overline{A}r_0).$$

Beweis: Wir zeigen, dass gilt

$$\tilde{K}_m(A, r_0) = \text{span}\left(\overline{r}_0, \overline{A}r_0, \ldots, (\overline{A}A)^{k-1}\overline{r}_0, (\overline{A}A)^{k-1}\overline{A}r_0\right) \text{ für } m = 2k, \quad (3.10)$$

$$\tilde{K}_m(A, r_0) = \text{span}\left(\overline{r}_0, \overline{A}r_0, \ldots, (\overline{A}A)^{k-1}\overline{A}r_0, (\overline{A}A)^{k}\overline{r}_0\right) \text{ für } m = 2k+1. \quad (3.11)$$

Es gilt $q_1 \in \text{span}\,(\overline{r}_0) = \tilde{K}_1(A, r_0)$ und $q_2 \in \text{span}\,\left(\overline{r}_0, \overline{A}r_0\right) = \tilde{K}_2(A, r_0)$.
Die Darstellung des Unterraumes beweisen wir per Induktion über m. Sei die Darstellung des Unterraumes $\tilde{K}_m(A, r_0)$ gegeben durch (3.10) bzw. (3.11). Algorithmus 3.2 konstruiert den neuen Basisvektor q_{m+1} über die Zuweisung

$$\begin{aligned} \overline{q_{m+1}} &= (Aq_m - \alpha_m\overline{q_m} - \beta_m\overline{q_{m-1}})/\beta_{m+1} \\ &\Rightarrow q_{m+1} \in \overline{A}\tilde{K}_m(\overline{A}, \overline{r}_0) + \tilde{K}_m(A, r_0). \end{aligned}$$

Wir betrachten nun den Unterraum $\overline{A}\tilde{K}_m(\overline{A}, \overline{r}_0)$ genauer. Es gilt für $l \in \mathbb{N}_0$:

$$\begin{aligned} \overline{A}(A\overline{A})^l r_0 &= (\overline{A}A)^l\overline{A}r_0 \text{ und} \\ \overline{A}(A\overline{A})^l A\overline{r}_0 &= (\overline{A}A)^{l+1}\overline{r}_0. \end{aligned}$$

Der Unterraum $\overline{A}\tilde{K}_m(\overline{A}, \overline{r}_0)$ enthält also den Unterraum $\tilde{K}_m(A, r_0) \setminus \text{span}\,(\overline{r}_0)$.
Betrachten wir nun den Term mit den Höchstkoeffizienten in der Darstellung (3.10) bzw. (3.11), so gilt:

$$\begin{aligned} \overline{A}(A\overline{A})^{k-1}A\overline{r}_0 &= (\overline{A}A)^k\overline{r}_0, \text{ für } m = 2k, \\ \overline{A}(A\overline{A})^k r_0 &= (\overline{A}A)^k\overline{A}r_0 \text{ für } m = 2k+1. \end{aligned}$$

Damit ist die Darstellung für den Unterraum $\tilde{K}_{m+1}(A, r_0) = \overline{A}\tilde{K}_m(\overline{A}, \overline{r}_0) + \tilde{K}_m(A, r_0)$ bewiesen. □

Bemerkung 3.34. Ein **Quasi-Krylov-Unterraum** [37] wird für allgemeine Matrizen eingeführt durch

$$\tilde{K}_m(A, c, d) = K_{\lfloor \frac{m+1}{2} \rfloor}(A^H A, c) + K_{\lfloor \frac{m}{2} \rfloor}(A^H A, A^H d)$$

mit zwei Vektoren c und d . Den Unterraum des CSYM-Verfahrens erhalten wir hieraus durch die Wahl $c = \overline{r}_0$ und $d = \overline{c} = r_0$.

Es lässt sich folgender Zusammenhang mit Krylov-Unterräumen herstellen:

- Der Unterraum $K_{\lfloor \frac{m}{2} \rfloor}(\overline{A}A, \overline{A}r_0)$ ist der Unterraum des CGNR-Verfahrens (s. Bemerkung 3.22 mit $A^H = \overline{A}$).
- Für gerades m, $l = m/2$ ist

$$\tilde{K}_m(A, r_0) = K_l(B, Q_0) \text{ mit } B = \overline{A}A,\ Q_0 = (\overline{r}_0, \overline{A}r_0),$$

  d.h. $\tilde{K}_m(A, r_0)$ entspricht einem Unterraum eines Block-Lanczos-Verfahrens mit der Blockgröße 2.

Die Iterierte x_m wird aus dem Unterraum $x_0 + \tilde{K}_m(A, r_0)$ bestimmt. Für die Verfahrenspolynome gilt folgender Zusammenhang:

Lemma 3.35. *Ist* $x_m = x_0 + q_1(\overline{A}A)\overline{r}_0 + q_2(\overline{A}A)\overline{A}r_0$ *mit* $q_1 \in \mathcal{P}_{\lfloor \frac{m-1}{2} \rfloor}$ *und* $q_2 \in \mathcal{P}_{\lfloor \frac{m-2}{2} \rfloor}$ *gegeben, so gilt*

$$r_m = p_1(A\overline{A})A\overline{r}_0 + p_2(A\overline{A})r_0 \text{ mit } p_1(t) = -q_1(t),\ p_2(t) = 1 - tq_2(t),\ p_1,\ p_2 \in \mathcal{P}_{\lfloor \frac{m-1}{2} \rfloor}.$$

Beweis: Wir setzen die Darstellung von x_m in die von r_m ein und erhalten

$$r_m = r_0 - Aq_1(\overline{A}A)\overline{r}_0 - Aq_2(\overline{A}A)\overline{A}r_0 = r_0 - q_1(A\overline{A})A\overline{r}_0 - A\overline{A}q_2(A\overline{A})r_0.$$

Klammern wir noch die beiden Terme mit r_0, so erhalten wir die Darstellung für r_m. □

3.2.3 Die Optimalitätseigenschaft

Im CSYM-Verfahren bestimmen wir die Iterierte x_m so, dass sie die MINRES-Eigenschaft bezüglich des Unterraumes $\tilde{K}_m(A, r_0)$ erfüllt.
In Lemma 3.14 haben wir gezeigt, dass die Iterierte $x_m \in x_0 + Q_m z_m$ mit MINRES-Eigenschaft bezüglich des Krylov-Unterraumes $K_m(A, r_0)$ bestimmt werden kann, indem die Minimierungsaufgabe $\min_{z \in \mathbb{C}^m} \| \|r_0\|_2 e_1 - H_{m+1,m} z\|_2$ mit einer „erweiterten“ Hessenbergmatrix $H_{m+1,m}$ gelöst wird. Voraussetzung war, dass die Basisvektoren in Q_m orthonormal sind und $AQ_m = Q_{m+1}H_{m+1,m}$ gilt. Dies war für das GMRES-Verfahren und auch für das CG/CGNR und MINRES-Verfahren immer der Fall.
Im CG- und MINRES-Verfahren ist die Berechnung vom Aufwand her besonders günstig, weil $H_{m,m}$ speziell eine Tridiagonalmatrix ist.
Auch im CQMR- und CSYM-Verfahren wird eine Tridiagonalmatrix berechnet, das

obige Vorgehen lässt sich aber nicht direkt übertragen. Das dem CQMR-Verfahren zu Grunde liegende Lanczos-Verfahren hat in Matrixschreibweise die gewünschte Form $AP_m = P_{m+1}\tilde{T}_{m+1,m}$, konstruiert jedoch keine Orthogonalbasis. Das Lanczos-ähnliche Verfahren, auf dem das CSYM-Verfahren basiert, generiert eine Orthogonalbasis, jedoch keinen Krylov-Unterraum. Die zugehörige Matrixschreibweise ist gegeben durch $AQ_m = \overline{Q}_{m+1}T_{m+1,m}$ und weicht von der typischen Darstellung für Krylov-Unterräume ab.

Wieso kann eine Iterierte mit Minimales-Residuum-Eigenschaft nicht auch im CQMR-Verfahren mit konstantem Aufwand berechnet werden?
Die Matrix P_m aus der Darstellung (3.5) hat komplex-orthogonale Spalten. Deshalb gilt

$$\begin{aligned}
\|r_m\|_2 &= \|b - Ax_m\|_2 \\
&= \|b - Ax_0 - A(P_m z_m)\|_2 \\
&= \|r_0 - P_{m+1}\tilde{T}_{m+1,m}z_m\|_2 \\
&= \|P_{m+1}(P_{m+1}^T r_0 - \tilde{T}_{m+1,m}z_m)\|_2 \\
&= \|P_{m+1}(\|r_0\|_2 e_1 - \tilde{T}_{m+1,m}z_m)\|_2.
\end{aligned}$$

Die Basisvektoren in P_{m+1} sind nicht normiert, der Wert $\|P_{m+1}\|_2$ unbeschränkt. Deshalb werden die Basisvektoren noch normiert, d.h. es wird $P_{m+1}\Omega_{m+1}$ statt P_{m+1} betrachtet mit einer Diagonalmatrix Ω_{m+1}, wobei $\mathrm{diag}(\Omega_{m+1}) = (1/\|p_1\|_2, \ldots, 1/\|p_{m+1}\|_2)$. Dann erhalten wir

$$\begin{aligned}
\|r_m\|_2 &= \|P_{m+1}\Omega_{m+1} \cdot (\|r_0\|_2 e_1 - \Omega_{m+1}^{-1}\tilde{T}_{m+1,m}z_m)\|_2 \\
&\leq \|P_{m+1}\Omega_{m+1}\|_2 \cdot \left\|\|r_0\|_2 e_1 - \Omega_{m+1}^{-1}\tilde{T}_{m+1,m}z_m\right\|_2 .
\end{aligned}$$

Das kleinste Quadrate-Problem

$$\min_{z\in\mathbb{C}^m} \left\|\|r_0\|_2 e_1 - \Omega_{m+1}^{-1}\tilde{T}_{m+1,m}z\right\|_2$$

kann mit konstantem Aufwand pro Iterationsschritt berechnet werden, liefert aber nur eine Iterierte mit schwächerer Optimalitätseigenschaft als die Minimales-Residuum-Eigenschaft. Die Iterierte wird auf diese Weise im CQMR-Verfahren ermittelt, und erfüllt daher die schwächere sogenannte Quasi-Minimales-Residuum-Eigenschaft.

Zurück zum CSYM-Verfahren. Die Darstellung aus (3.9) eignet sich dazu, eine Iterierte mit MINRES-Eigenschaft mit konstantem Aufwand zu berechnen, wie der nachfolgende Satz zeigt.

Satz 3.36. *Sei $q_1 = \overline{r}_0/\|r_0\|_2$. Gilt mit orthonormalen Basisvektoren $q_1, \ldots, q_m, q_{m+1}$ in der Notation $Q_m = [q_1, \ldots, q_m]$, $Q_{m+1} = [Q_m, q_{m+1}]$ die Darstellung*

$$AQ_m = \overline{Q}_{m+1} T_{m+1,m},$$

so besitzt $x_m = x_0 + Q_m z_m$ bezüglich des durch die Spalten von Q_m aufgespannten Raumes die MINRES-Eigenschaft, wenn z_m die folgende Minimierungsaufgabe löst:

$$\min_{z \in \mathbb{C}^m} \left\| \|r_0\|_2 e_1 - T_{m+1,m} z \right\|_2 .$$

Beweis:

$$\begin{aligned} \|r_m\|_2 &= \|b - Ax_m\|_2 \\ &= \|b - Ax_0 - A(Q_m z_m)\|_2 \\ &= \|r_0 - \overline{Q}_{m+1} T_{m+1,m} z_m\|_2 \\ &= \|\overline{Q}_{m+1}(Q_{m+1}^T r_0 - T_{m+1,m} z_m)\|_2 \\ &= \left\| \|r_0\|_2 e_1 - T_{m+1,m} z_m \right\|_2 . \end{aligned}$$

□

Was bedeutet nun ein lucky breakdown in dem Lanczos-ähnlichen Verfahren für die Iterierte mit MINRES-Eigenschaft bezüglich des Unterraumes $\tilde{K}_m(A, r_0)$? Der folgende Satz liefert eine Antwort.

Satz 3.37. *Sei m^* minimal, so dass*

$$\begin{aligned} \dim(\tilde{K}_m(A, r_0)) &< m^* \text{ für } m < m^*, \\ \dim(\tilde{K}_m(A, r_0)) &= m^* \text{ für } m \geq m^*. \end{aligned}$$

Dann gilt $x_* \in x_0 + \tilde{K}_{m^*}(A, r_0)$ und $x_* \notin x_0 + \tilde{K}_m(A, r_0)$ für $m < m^*$.

Beweis: [37] Angenommen im m^*-ten Schritt des Lanczos-ähnlichen Verfahrens gilt $\beta_{m^*+1} = 0$. Der Quasi-Krylov-Unterraum ist dann maximal, da dies nur eintreten kann, wenn w_{m^*+1} der Nullvektor ist. Da die letzte Zeile von T_{m^*+1,m^*} eine Nullzeile ist, kann die Minimierungsaufgabe aus Satz 3.36 exakt gelöst werden, d.h. $\|r_{m^*}\|_2 = 0$, und somit gilt $x_{m^*} = x_*$. □

Das CSYM-Verfahren generiert mit dem Lanczos-ähnlichen Verfahren (Algorithmus 3.2) eine orthogonale Basis von $\tilde{K}_m(A, r_0)$. In jedem Iterationsschritt wird die Iterierte mit

MINRES-Eigenschaft aus dem Unterraum bestimmt, indem das kleinste Quadrate-Problem aus Satz 3.36 gelöst wird. Die Lösung des kleinsten Quadrate-Problems erfolgt durch sukzessiv aufdatierte QR-Zerlegungen, so dass der Aufwand pro Iterationsschritt konstant ist. Die Norm des Residuums kann dabei mit geringem zusätzlichem Aufwand auch aufdatiert werden. Die Herleitung der Aufdatierung erfolgt analog dem GMRES-Verfahren in [35] und ist in [5] skizziert.
Für das CSYM-Verfahren erhalten wir somit den folgenden Algorithmus. Hierbei sind s_m und c_m die entsprechenden Werte aus der m-ten Givens-Rotation [14].

Algorithmus 3.3 Das CSYM-Verfahren

wähle $x_0,\ r_0 = b - Ax_0$
Setze $q_0 = 0,\ \tau_1 = \|r_0\|_2,\ q_1 = \overline{r}_0/\tau_1,\ z_1 = Aq_1,\ \alpha_1 = q_1^T z_1,\ \beta_1 = 0,$
$c_{-1} = 0,\ s_{-1} = 0,\ c_0 = 1,\ s_0 = 0,\ p_{-1} = 0,\ p_0 = 0$
for $m = 1, 2 \ldots$ **do**
  $\eta_m = c_{m-2}c_{m-1}\beta_m + \overline{s_{m-1}}\alpha_m$
  $\theta_m = \overline{s_{m-2}}\beta_m$
  $\gamma_m = c_{m-1}\alpha_m - c_{m-2}s_{m-1}\beta_m$
  $w = z_m - \alpha_m\overline{q_m} - \beta_m\overline{q_{m-1}}$
  $\beta_{m+1} = \|w\|_2$
  $q_{m+1} = \overline{w}/\beta_{m+1}$
  $z_{m+1} = Aq_{m+1}$
  $\alpha_{m+1} = q_{m+1}^T z_{m+1}$
  if $\gamma_m \neq 0$ **then**
    $\psi = \sqrt{|\gamma_m|^2 + \beta_{m+1}^2}$
    $c_m = |\gamma_m|/\psi$
    $s_m = (\overline{\gamma_m}/|\gamma_m|)(\beta_{m+1}/\psi)$
    $\xi_m = \gamma_m/|\gamma_m|\psi$
  else
    $c_m = 0,\ s_m = 1,\ \xi_m = \beta_{m+1}$
  end if
  $p_m = (q_m - \eta_m p_{m-1} - \theta_m p_{m-2})/\xi_m$
  $x_m = x_{m-1} + \tau_m c_m p_m$
  $\tau_{m+1} = -s_m\tau_m$
  $\|r_m\|_2 = |\tau_{m+1}|$
end for

3.2.4 Konvergenzbetrachtungen

Satz 3.38. *Es sei M die Anzahl der mehrfachen und N die Anzahl der einfachen Singulärwerte der komplex symmetrischen Matrix A. Dann gilt für die m-te Iterierte des CSYM-Verfahrens $x_m = x_* = A^{-1}b$ mit $m \leq 2M + N$.*

Beweisidee: [5] Wir setzen die SSVD von A, $A = V\Sigma V^T$, in folgende Gleichung ein:

$$r_m = r_0 - q_1(A\overline{A})A\overline{r}_0 - q_2(A\overline{A})A\overline{A}r_0.$$

Dann stellen wir ein Gleichungssystem mit den Koeffizienten der Polynome für $r_m = 0$ auf und zeigen, dass es für ein $m \leq 2M + N$ lösbar ist. In Abhängigkeit davon, ob Teilvektoren von $V^H r_0$ und entsprechende Teilvektoren von $V^T\overline{r}_0$ linear unabhängig sind, können pro mehrfachem Singulärwert zwei Iterationsschritte notwendig sein.

Im Vergleich dazu werden im CGNR-Verfahren (Lemma 3.27) nur $M+N$ Iterationsschritte benötigt. Zu beachten ist, dass in jedem Schritt zwei Matrix-Vektormultiplikationen durchzuführen sind.

Satz 3.39. *Für die Iterierten des CSYM-Verfahrens gilt mit $c = \frac{cond(A)-1}{cond(A)+1}$ die Abschätzung*

$$\|r_m^{CSYM}\|_2 \leq 2c^{\lfloor \frac{m}{2} \rfloor}\|r_0\|_2.$$

Beweisidee: [5] Wir schätzen $\|r_{2k}^{CSYM}\|_2 = \|r_0 - q_1(A\overline{A})A\overline{r}_0 - q_2(A\overline{A})A\overline{A}r_0\|_2$ mit $q_1,\ q_2 \in \mathcal{P}_{k-1}$ nach oben ab, indem wir für $q_2(\cdot)$ speziell das Verfahrenspolynom aus dem k-ten Schritt des CGNR-Verfahrens wählen und $q_1 = 0$ setzen. Auf Grund der MINRES-Eigenschaft des CSYM-Verfahrens erhalten wir

$$\|r_{2k+1}^{CSYM}\|_2 \leq \|r_{2k}^{CSYM}\|_2 \leq \|r_k^{CGNR}\|_2.$$

Die Abschätzung von $\|r_k^{CGNR}\|_2$ aus Lemma 3.28 liefert die Behauptung.

In der Abschätzung der Norm des Residuums im m-ten Iterationsschritt des CSYM-Verfahrens kommt $c^{\lfloor \frac{m}{2} \rfloor}$ vor, in der des CGNR-Verfahrens in Lemma 3.28 c^m. In jedem Iterationsschritt sind im CGNR-Verfahren zwei Matrix-Vektormultiplikationen, im CSYM-Verfahren ist aber nur eine Matrix-Vektormultiplikation notwendig. Betrachtet man daher die Abschätzung in Hinblick auf den Aufwand, so liefert das CSYM-Verfahren bei gleicher Anzahl an Matrix-Vektormultiplikationen eine mindestens so gute Approximation der Lösung wie das CGNR-Verfahren.
Der folgende Satz liefert eine Konvergenzabschätzung des Block-Lanczos-Verfahrens für eine beliebige Blockgröße p.

Satz 3.40. *Es sei $B \in \mathbb{C}^{n\times n}$ hermitesch und (λ_i, z_i) seien die Eigenpaare von B, die absteigend nach den Eigenwerten sortiert sind. Weiter seien $\mu_1 \geq \ldots \geq \mu_p$ die größten Eigenwerte der projizierten Matrix im Block-Krylov-Unterraum $K_m(B, Q_0)$ mit Blockgröße p und $Q_0 \in \mathbb{C}^{n\times p}$. Dann sind folgende Ungleichungen für $k = 1, \ldots, p$ erfüllt:*

$$\begin{aligned}
\lambda_k \geq \mu_k \geq \lambda_k - \delta_k^2, &\text{ wobei}\\
\delta_k^2 = (\lambda_1 - \lambda_n)\tan^2(\theta_p)/ &\left(C_{m-1}^2\left(\frac{1+\gamma_k}{1-\gamma_k}\right)\right),\\
\gamma_k = (\lambda_k - \lambda_{p+1})/(\lambda_k - \lambda_n),&\\
\cos(\theta_p) = \sigma_p([z_1, \ldots,\ z_p]^H Q_0) > 0,&
\end{aligned}$$

C_{m-1} ist das Chebyscheff-Polynom vom Grad $m-1$.

Beweis: s. [15].

Bemerkung 3.41. Die projizierte Matrix aus dem Block-Lanczos-Verfahren für hermitesche Matrizen mit Blockgröße p ist eine hermitesche Bandmatrix mit halber Bandbreite p.

Satz 3.40 kann für Blockgröße 2 auf das CSYM-Verfahren übertragen werden [38] und wir erhalten folgenden Satz.

Satz 3.42. *Es sei $B = \overline{A}A$, $Q_0 = (\overline{r}_0, \overline{A}r_0)$ und (λ_i, z_i) seien die Eigenpaare von B, die absteigend nach den Eigenwerten sortiert sind. Weiter seien $\mu_1 \geq \mu_2$ die größten Eigenwerte der Pentadiagonalmatrix im Block-Krylov-Unterraum $K_m(B, Q_0)$. Dann sind folgende Ungleichungen für $k = 1,\ 2$ erfüllt:*

$$\begin{aligned}
\lambda_k \geq \mu_k \geq \lambda_k - \delta_k^2,\ &\text{ wobei}\\
\delta_k^2 = (\lambda_1 - \lambda_n)\tan^2(\theta_2)/ &\left(C_{m-1}^2\left(\frac{1+\gamma_k}{1-\gamma_k}\right)\right),\\
\gamma_k = (\lambda_k - \lambda_3)/(\lambda_k - \lambda_n),&\\
\cos(\theta_2) = \sigma_2([z_1, z_2]^H Q_0) > 0,&
\end{aligned}$$

C_{m-1} ist das Chebyscheff-Polynom vom Grad $m-1$.

Satz 3.42 liefert eine Abschätzung über die Konvergenz der größten Singulärwerte der projizierten Matrix gegen Singulärwerte der Matrix A.

3.3 Weitere CSYM-Iterationsverfahren

Wir haben uns, mit Ausnahme des CQMR-Verfahrens, auf Unterraum-Verfahren beschränkt, die die Minimales-Residuum-Eigenschaft besitzen.
Zu den Unterraum-Verfahren, die die komplexe Symmetrie der Matrix nutzen, gehören u.a. folgende Verfahren, die eine sogenannte Galerkin-Bedingung erfüllen:

- Das COCG-Verfahren (**C**omplex **O**rthogonal **CG**) [43], auch CBCG-Verfahren genannt, konstruiert eine komplex-orthogonale Basis für $K_m(A, r_0)$ wie das CQMR-Verfahren und bestimmt die Iterierte x_m so, dass $r_m^T P_m = 0$ gilt.
- Das CCG-Verfahren (**C**SYM **CG**) [44] konstruiert mit dem Lanczos-ähnlichen Verfahren eine orthonormale Basis für $\tilde{K}_m(A, r_0)$ wie das CSYM-Verfahren und bestimmt die Iterierte x_m so, dass $\overline{r}_m^T Q_m = 0$ gilt.

Beide Verfahren können vorzeitig abbrechen, wenn es keine sogenannte Galerkin-Iterierte im jeweiligen Unterraum gibt. Im COCG-Verfahren ist also ein breakdown, zusätzlich zu dem durch quasi-null Vektoren bedingten, möglich. Die Norm der Residuen weist häufig ein oszillierendes Verhalten auf, da die Optimalitätseigenschaft i.A. keine Minimales-Residuum-Eigenschaft nach sich zieht.

3.4 Zusammenfassung

In diesem Kapitel haben wir das CSYM-Verfahren vor- und dem CQMR-Verfahren gegenübergestellt. Beide Verfahren haben Gemeinsamkeiten wie die Beibehaltung der Symmetrie der projizierten Operatoren, einen Aufwand von einem Matrix-Vektorprodukt je Iterationsschritt und eine Reduktion auf Tridiagonalgestalt. Während das CQMR-Verfahren zu den Krylov-Unterraum-Verfahren zählt und nur ein Verfahrenspolynom involviert wird, ist das CSYM-Verfahren ein Quasi-Krylov-Unterraum-Verfahren, wobei aber Schnittstellen zum CGNR- und zum Block-Lanczos-Verfahren der Größe 2 bestehen. Das CSYM-Verfahren ist, wie auch das CQMR-Verfahren, in exakter Arithmetik ein endliches Verfahren. Singulärwerte mit einer Mehrfachheit größer 2 wirken sich nach Satz 3.38 günstig auf die Konvergenzgeschwindigkeit aus. Satz 3.39 liefert eine Abschätzung der Residuum-Norm in Abhängigkeit von der Kondition der Matrix. Das Hauptaugenmerk zur Konvergenzbeschleunigung liegt somit auf einer Verbesserung der Verteilung der Singulärwerte.

Kapitel 4

Präkonditionierung

Das Ziel einer Präkonditionierung ist eine Konvergenzbeschleunigung. Hierzu lösen wir ein äquivalentes Gleichungssystem mit einer komplex symmetrischen Matrix $\hat{A}$. Diese präkonditionierte Matrix $\hat{A}$ wird in Abhängigkeit von einem Präkonditionierer $M \approx A$ bestimmt und soll günstiger verteilte Singulärwerte als die ursprüngliche Matrix haben. Für Krylov-Unterraum-Verfahren ist ein Ziel möglichst viele Eigenwerte um 1 zu clustern. Mit Blick auf die Sätze 3.38 und 3.39 ist das Ziel für das CSYM-Verfahren einerseits mehrfache Singulärwerte (Mehrfachheit größer zwei) zu erzeugen und andererseits die Kondition zu verkleinern.
In diesem Kapitel betrachten wir zunächst die praktische Anwendung des noch abstrakten Präkonditionierers im CSYM-Verfahren. Das CSYM-Verfahren wird nicht explizit auf das Gleichungssystem mit der präkonditionierten Matrix angewendet sondern implizit. Zusätzliche Aufwände im Vergleich zum unpräkonditionierten Verfahren sind hierbei je Iterationsschritt das Lösen eines Gleichungssystems und eine zusätzliche Matrix-Vektormultiplikation mit einer von der Faktorisierung von M abhängenden Matrix. Dazu werden wir zunächst die notwendige Symmetrie eines Präkonditionierers im CSYM-Verfahren betrachten. Anschließend beschäftigen wir uns mit der Wirkung eines Präkonditionierers auf die Singulärwerte und mit den Eigenschaften, die ein Präkonditionierer des CSYM-Verfahrens haben sollte, um effektiv zu sein.

4.1 Das präkonditionierte CSYM-Verfahren

Die Form des Präkonditionierers

Zunächst wählen wir eine reguläre Matrix $M = M_1 M_2$, den sogenannten Präkonditionierer in faktorisierter Form, und gehen dann auf ein zu $Ax = b$ äquivalentes präkonditioniertes Gleichungssystem über. Wir betrachten

$$M_1^{-1} A M_2^{-1} (M_2 x) = M_1^{-1} b.$$

Im Folgenden sei $\hat{A} = M_1^{-1} A M_2^{-1}$ die präkonditionierte Matrix und es seien $\hat{x} = M_2 x$ und $\hat{b} = M_1^{-1} b$ die entsprechenden Größen des präkonditionierten Systems.
Der Präkonditionierer $M = M_1 M_2$ kann beidseitig wie in der obigen Form angewendet werden. Einseitig präkonditionierte Systeme lassen sich wie folgt als Spezialfälle des beidseitig präkonditionierten Systems ableiten:

- Ist $M_2 = I$ und damit $M = M_1$, so erhalten wir das linksseitig präkonditionierte System

$$M_1^{-1} A x = M_1^{-1} b.$$

- Wählen wir dagegen $M_1 = I$ und damit $M = M_2$, so erhalten wir das rechtsseitig präkonditionierte System

$$A M_2^{-1} (M_2 x) = b.$$

Da zur Nutzung des CSYM-Verfahrens $\hat{A} = \hat{A}^T$ Voraussetzung ist, fordern wir:

$$M_2^{-T} A M_1^{-T} = M_1^{-1} A M_2^{-1}.$$

Also bietet sich bei echter beidseitiger Präkonditionierung die Wahl $M_2 = M_1^T$ an, d.h. es ist ein symmetrischer Präkonditionierer $M = M_1 M_1^T$ zu wählen.
Bei linksseitiger Präkonditionierung müsste für $M = M_1$ die Beziehung

$$\begin{aligned} A M^{-T} &= M^{-1} A \\ \Leftrightarrow M A &= A M^T \end{aligned} \tag{4.1}$$

erfüllt sein, was z.B. der Fall ist, wenn $M = M^T$ ist und A und M kommutieren. Dies bedeutet die Diagonalisierbarkeit von A und M mit derselben regulären Matrix S, was ein absoluter Ausnahmefall ist. Erfüllt A die Beziehung (4.1) mit einem M, so nennt man A auch M-symmetrisch. Analog wäre bei rechtsseitiger Präkonditionierung zu fordern:

$$A M = M^T A.$$

Bemerkung 4.1. Es gibt eine QMR-Variante für M-symmetrische Matrizen A. In jedem Schritt sind dabei zum Aufbau der Krylov-Unterräume $K_m(A, r_0)$ und $K_m(A^H, w_1)$ im unsymmetrischen Lanczos-Verfahren eine Multiplikation mit A und eine mit M durchzuführen. Durch die Nutzung der M-Symmetrie entfällt pro Iterationsschritt eine Multiplikation mit A^H.

Die Anwendung der Präkonditionierung

Im Folgenden betrachten wir die beidseitige Präkonditionierung mit komplex symmetrischem Präkonditionierer $M = SS^T$. Wie bei dem präkonditionierten CG-Verfahren wollen wir jetzt den Algorithmus so umformulieren, dass die Faktorisierung $M = SS^T$ des Präkonditionierers nicht mehr nötig wird. Und da erleben wir eine Überraschung. Es wird sich herausstellen, dass wir im implizit präkonditionierten Verfahren die Faktorisierung des Präkonditionierers $M = SS^T$ benötigen, da der Präkonditionierer in der Form $\tilde{M} = SS^H$ vorkommt.

Wir betrachten zunächst das CSYM-Verfahren bezüglich des neuen Systems $\hat{A}\hat{x} = \hat{b}$:

$$\hat{A}\hat{x} = \hat{b},\ \hat{A} = \hat{A}^T$$

mit $\hat{A} = S^{-1}AS^{-T}$, $\hat{x} = S^T x$ und $\hat{b} = S^{-1}b$.

Algorithmus 4.1 Das explizit präkonditionierte CSYM-Verfahren

(1) wähle $\hat{x}_0,\ \hat{r}_0 = \hat{b} - \hat{A}\hat{x}_0$
(2) $\hat{q}_0 = 0,\ \hat{\tau}_1 = \|\hat{r}_0\|_2,\ \hat{q}_1 = \overline{\hat{r}_0}/\hat{\tau}_1$
(3) $\hat{z}_1 = \hat{A}\hat{q}_1,\ \hat{\alpha}_1 = \hat{q}_1^T\hat{z}_1,\ \hat{\beta}_1 = 0$
Initialisierung $\hat{c}_0,\ \hat{s}_0,\ \hat{c}_{-1},\ \hat{s}_{-1},\ \hat{p}_{-1},\ \hat{p}_0$
for $m = 1, 2\ldots$ **do**
(4) $\hat{w} = \hat{z}_m - \hat{\alpha}_m\overline{\hat{q}_m} - \hat{\beta}_m\overline{\hat{q}_{m-1}}$
(5) $\hat{\beta}_{m+1} = \|\hat{w}\|_2$
(6) $\hat{q}_{m+1} = \overline{\hat{w}}/\hat{\beta}_{m+1}$
(7) $\hat{z}_{m+1} = \hat{A}\hat{q}_{m+1}$
(8) $\hat{\alpha}_{m+1} = \hat{q}_{m+1}^T\hat{z}_{m+1}$
Berechnung $\hat{\eta}_m,\ \hat{\theta}_m, \hat{\gamma}_m,\ \hat{c}_m,\ \hat{s}_m,\ \hat{\xi}_m$
(9) $\hat{p}_m = (\hat{q}_m - \hat{\eta}_m\hat{p}_{m-1} - \hat{\theta}_m\hat{p}_{m-2})/\hat{\xi}_m$
(10) $\hat{x}_m = \hat{x}_{m-1} + \hat{\tau}_m\hat{c}_m\hat{p}_m$
$\hat{\tau}_{m+1} = -\hat{s}_m\hat{\tau}_m$
(11) $\|\hat{r}_m\|_2 = |\hat{\tau}_{m+1}|$
end for

Da wir einerseits $\hat{A}$ nicht berechnen, andererseits statt $\hat{x}_m$ direkt $x_m = S^{-T}\hat{x}_m$ erhalten wollen, transformieren wir die Größen zurück, so dass die Iterierten des ursprünglichen Systems berechnet werden. Nun gilt

$$r_m = S\hat{r}_m.$$

Außerdem setzen wir

$$\tilde{p}_m = S^{-T}\hat{p}_m, \quad \tilde{q}_m = S^{-T}\hat{q}_m \text{ und } \tilde{w} = S^{-H}\hat{w}.$$

Wir betrachten zunächst die Anweisungen in Algorithmus 4.1, in denen $\hat{x}_m$ berechnet wird, und anschließend die Berechnung der involvierten Vektoren.
Multiplikation von links mit S^{-T} von (10), (9) und (6) ergibt:

(10)' $x_m = x_{m-1} + \hat{\tau}_m \hat{c}_m \tilde{p}_m$,
(9)' $\tilde{p}_m = (\tilde{q}_m - \hat{\eta}_m \tilde{p}_{m-1} - \hat{\theta}_m \tilde{p}_{m-2})/\hat{\xi}_m$,
(6)' $\tilde{q}_{m+1} = \overline{S^{-H}\hat{w}}/\hat{\beta}_{m+1} = \overline{\tilde{w}}/\hat{\beta}_{m+1}$.

Insbesondere gilt bei der Berechnung in (2) mit $\tilde{r}_0 = S^{-1}S^{-H}r_0$:

(2)' $\tilde{q}_1 = \overline{S^{-H}S^{-1}r_0} = \overline{\tilde{r}_0}$.

Nun zur Berechnung des Vektors $\tilde{w}$, den wir aus (4) durch Multiplikation mit S^{-H} erhalten:

(4)' $\tilde{w} = S^{-H}\hat{z}_m - \hat{\alpha}_m \overline{S^{-T}\hat{q}_m} - \hat{\beta}_m \overline{S^{-T}\hat{q}_{m-1}}$.

Multiplizieren wir (7) mit S^{-H}, so erhalten wir mit $\tilde{z}_m = S^{-H}\hat{z}_m$:

(7)' $\tilde{z}_{m+1} = S^{-H}S^{-1}A\tilde{q}_{m+1}$, d.h. wir müssen ein Gleichungssystem mit SS^H lösen.

Für die Skalare, die aus den transformierten Vektoren berechnet werden, gilt:

(5) $\hat{\beta}_{m+1} = \|\hat{w}\|_2 = \|S^H\tilde{w}\|_2$,
(8) $\hat{\alpha}_{m+1} = \hat{q}_{m+1}^T\hat{z}_{m+1} = \hat{q}_{m+1}^T\hat{A}\hat{q}_{m+1} = \tilde{q}_{m+1}^T A\tilde{q}_{m+1}$.

Weiter gilt $\|\hat{r}_m\|_2 = \|S^{-1}r_m\|_2 = |\hat{\tau}_{m+1}|$.

Für $\tilde{q}_m$ schreiben wir im folgenden Algorithmus wieder q_m und analog für die anderen Größen (außer $\tilde{z}_m$ und $\tilde{r}_0$) und erhalten so das implizit präkonditionierte Verfahren.

Algorithmus 4.2 Das implizit präkonditionierte CSYM-Verfahren (Variante 1)

wähle $x_0,\ r_0 = b - Ax_0$
löse $\mathbf{SS^H}\tilde{r}_0 = r_0$
$q_0 = 0,\ \tau_1 = \sqrt{r_0^H \tilde{r}_0},\ q_1 = \overline{\tilde{r}_0}/\tau_1$
$z_1 = Aq_1,\ \alpha_1 = q_1^T z_1, \beta_1 = 0$
Initialisierung $c_0,\ s_0,\ c_{-1},\ s_{-1},\ p_{-1},\ p_0$
for $m = 1, 2 \ldots$ **do**
  löse $\mathbf{SS^H}\tilde{z}_m = z_m$
  $w = \tilde{z}_m - \alpha_m \overline{q_m} - \beta_m \overline{q_{m-1}}$
  $\beta_{m+1} =< w, \mathbf{SS^H} w >^{1/2}$
  $q_{m+1} = \overline{w}/\beta_{m+1}$
  $z_{m+1} = Aq_{m+1}$
  $\alpha_{m+1} = q_{m+1}^T z_{m+1}$
  Berechnung $\eta_m,\ \theta_m,\ \gamma_m,\ c_m,\ s_m,\ \xi_m$
  $p_m = (q_m - \eta_m p_{m-1} - \theta_m p_{m-2})/\xi_m$
  $x_m = x_{m-1} + \tau_m c_m p_m$
  $\tau_{m+1} = -s_m \tau_m$
  $\|r_m\|_{(SS^H)^{-1}} = |\tau_{m+1}|$
end for

Algorithmus 4.2 hat folgende Eigenschaften:

- Die Norm bezüglich des präkonditionierten Residuums wird in jedem Iterationsschritt aktualisiert.
- Für die Berechnung von β_{m+1} ist ein Produkt mit SS^H bzw. S nötig.
- In jedem Iterationsschritt ist ein Gleichungssystem mit SS^H zu lösen.

Wir sind von einem komplex symmetrischen Präkonditionierer M ausgegangen. Die Faktoren S aus $M = SS^T$ treten in der Form SS^H auf. Das Verfahren hängt davon ab, welche Faktorisierung von M gewählt wird, d.h. M allein bestimmt **nicht** den Präkonditionierer. Wir haben also folgende drei Möglichkeiten, einen Präkonditionierer zu wählen:

- Jede Wahl eines Faktors S führt zu einem Präkonditionierer $M = SS^T$, im Algorithmus wird aber SS^H verwendet.
- Jede symmetrische Matrix M ist ein möglicher Präkonditionierer für das CSYM-Verfahren, sofern eine Faktorisierung $M = SS^T$ bestimmt wird.

- Jede hermitesch positiv definite Matrix $\tilde{M}$ ist ein Präkonditionierer. Dieser lässt sich immer in der Form $\tilde{M} = SS^H$ faktorisieren. Nehmen wir z.B. $\tilde{S} = V\Sigma^{1/2}$ mit Matrizen V und Σ aus der Singulärwertzerlegung $M = V\Sigma V^H$, so folgt $M = \tilde{S}\tilde{S}^T = V\Sigma V^T$. Diese Faktorisierung muss nicht explizit durchgeführt werden. Die Frage ist jedoch, wie und mit welchem Aufwand wir einen guten Präkonditionierer $\tilde{M}$ erhalten, so dass $\hat{A} = \tilde{S}^{-1}A\tilde{S}^{-T}$ besser verteilte Singulärwerte hat als A.

Die praktischen Ziele hinsichtlich eines Präkonditionierers sind daher:

- wenig Aufwand zur Lösung des Gleichungssystems mit $\tilde{M} = SS^H$,
- wenig Aufwand zur Multiplikation mit S^H oder $\tilde{M} = SS^H$,
- möglichst wenig Aufwand zur Berechnung von S bzw. $\tilde{M}$,
- möglichst geringer Speicherplatzbedarf für S bzw. $\tilde{M}$.

Je nach Präkonditionierer kann eine andere Variante des impliziten Verfahrens günstiger sein. Wir werden jetzt eine weitere implizite Variante entwickeln, die einen sogenannten SGS-Präkonditionierer voraussetzt.

Die Anwendung der SGS-Präkonditionierung

Eine einfach zu bestimmende symmetrische Faktorisierung liefert die Zerlegung aus dem symmetrischen Gauß-Seidel- oder kurz SGS-Verfahren [16].
Sei $A = D - L - L^T$ zerlegt mit einer Diagonalmatrix D und einer linken unteren Dreiecksmatrix L, dann setzen wir

$$M = (D - L)D^{-1}(D - L^T)$$

und als Faktor erhalten wir

$$S = (D - L)D^{-1/2}.$$

Hier kann die Anwendung des Eisenstat-Tricks [28] zu einer Reduktion des Rechenaufwandes führen. Wir faktorisieren den SGS-Präkonditionierer $M = SS^T = (D - L)D^{-1}(D - L)^T$ zunächst in der Form

$$M = \tilde{S}T\tilde{S}^T \text{ mit } \tilde{S} = (D - L),\ T = D^{-1}.$$

Wir verwenden $S = \tilde{S}T^{1/2}$ und erhalten dann

$$SS^H = \tilde{S}|T|\tilde{S}^H = (D - L)|D|^{-1}(D - L)^H.$$

Als Zwischenschritt formulieren wir zunächst ein implizit präkonditioniertes CSYM-Verfahren, in dem statt SS^H die Faktoren S und S^T vorkommen.

Algorithmus 4.3 Das implizit präkonditionierte CSYM-Verfahren (Variante 2)

wähle $x_0,\ r_0 = b - Ax_0$
$\hat{r}_0 = \mathbf{S^{-1}} r_0$
$q_0 = 0,\ \tau_1 = \sqrt{\hat{r}_0^H \hat{r}_0},\ q_1 = \overline{\hat{r}_0}/\tau_1$
$t_1 = \mathbf{S^{-T}} q_1$
$z_1 = \mathbf{S^{-1}}(At_1),\ \alpha_1 = q_1^T z_1, \beta_1 = 0$
Initialisierung $c_0,\ s_0,\ c_{-1},\ s_{-1},\ p_{-1},\ p_0$
for $m = 1, 2 \ldots$ **do**
  $w = z_m - \alpha_m \overline{q_m} - \beta_m \overline{q_{m-1}}$
  $\beta_{m+1} = \sqrt{w^H w}$
  $q_{m+1} = \overline{w}/\beta_{m+1}$
  $t_{m+1} = \mathbf{S^{-T}} q_{m+1}$
  $z_{m+1} = \mathbf{S^{-1}}(At_{m+1})$
  $\alpha_{m+1} = q_{m+1}^T z_{m+1}$
  Berechnung $\eta_m,\ \theta_m,\ \gamma_m,\ c_m,\ s_m,\ \xi_m$
  $p_m = (t_m - \eta_m p_{m-1} - \theta_m p_{m-2})/\xi_m$
  $x_m = x_{m-1} + \tau_m c_m p_m$
  $\tau_{m+1} = -s_m \tau_m$
end for

Bemerkung 4.2. In Algorithmus 4.3 ist pro Iterationsschritt jeweils ein Gleichungssystem mit S und eines mit S^T zu lösen. Je nach Präkonditionierer kann der Aufwand günstiger als in Algorithmus 4.2 sein.

Wir setzen nun

$$B = \tilde{S} + \tilde{S}^T - A = D = T^{-1},$$
$$\Rightarrow A = \tilde{S} + \tilde{S}^T - B.$$

Sei

$$\hat{r}_0 = \tilde{S}^{-1} r_0, \quad \tilde{r}_0 = |T|^{-1} \hat{r}_0, \quad \hat{q}_m = \tilde{S}^T q_m,$$
$$\hat{z}_m = \tilde{S}^{-1} z_m, \quad \tilde{z}_m = |T|^{-1} \hat{z}_m, \quad \hat{w} = \tilde{S}^H w,$$

dann gilt

$$\tau_1 = \sqrt{\hat{r}_0^H \tilde{r}_0}, \quad \alpha_m = \hat{q}_m^T \hat{z}_m, \quad \beta_{m+1} = \sqrt{\hat{w}^H |T| \hat{w}}.$$

Setzen wir $\hat{t}_m = \tilde{S}^{-T} \hat{q}_m$, so erhalten wir

$$\begin{aligned} \hat{z}_{m+1} &= \tilde{S}^{-1} A \tilde{S}^{-T} \hat{q}_{m+1} \\ &= \tilde{S}^{-1} (\tilde{S} + \tilde{S}^T - B) \tilde{S}^{-T} \hat{q}_{m+1} \\ &= \hat{t}_{m+1} + \tilde{S}^{-1} (\hat{q}_{m+1} - B\hat{t}_{m+1}). \end{aligned}$$

Für $\hat{q}_m$ schreiben wir im folgenden Algorithmus wieder q_m und analog für die anderen Größen und erhalten so Algorithmus 4.4.

Algorithmus 4.4 Das SGS-präkonditionierte CSYM-Verfahren mit Eisenstat-Trick

wähle $x_0,\ r_0 = b - Ax_0$
$\hat{r}_0 = \tilde{\mathbf{S}}^{-\mathbf{1}} r_0,\ \tilde{r}_0 = |\mathbf{T}|^{-\mathbf{1}} \hat{r}_0$
$q_0 = 0,\ \tau_1 = \sqrt{\hat{r}_0^H \tilde{r}_0},\ q_1 = \overline{\tilde{r}_0}/\tau_1$
$t_1 = \tilde{\mathbf{S}}^{-\mathbf{T}} q_1$
$z_1 = t_1 + \tilde{\mathbf{S}}^{-\mathbf{1}}(q_1 - \mathbf{B}t_1),\ \alpha_1 = q_1^T z_1, \beta_1 = 0$
Initialisierung $c_0,\ s_0,\ c_{-1},\ s_{-1},\ p_{-1},\ p_0$
for $m = 1, 2 \ldots$ **do**
  $\tilde{z}_m = |\mathbf{T}|^{-\mathbf{1}} z_m$
  $w = \tilde{z}_m - \alpha_m \overline{q_m} - \beta_m \overline{q_{m-1}}$
  $\beta_{m+1} = \sqrt{w^H |\mathbf{T}| w}$
  $q_{m+1} = \overline{w}/\beta_{m+1}$
  $t_{m+1} = \tilde{\mathbf{S}}^{-\mathbf{T}} q_{m+1}$
  $z_{m+1} = t_{m+1} + \tilde{\mathbf{S}}^{-\mathbf{1}}(q_{m+1} - \mathbf{B}t_{m+1})$
  $\alpha_{m+1} = q_{m+1}^T z_{m+1}$
  Berechnung $\eta_m,\ \theta_m,\ \gamma_m,\ c_m,\ s_m,\ \xi_m$
  $p_m = (t_m - \eta_m p_{m-1} - \theta_m p_{m-2})/\xi_m$
  $x_m = x_{m-1} + \tau_m c_m p_m$
  $\tau_{m+1} = -s_m \tau_m$
end for

In Algorithmus 4.2 wären folgende Anweisungen durchzuführen:

$$\text{löse } \tilde{S}|T|\tilde{S}^H\tilde{r}_0 = r_0, \tag{4.2}$$

$$\text{löse } \tilde{S}|T|\tilde{S}^H\tilde{z}_m = z_m, \tag{4.3}$$

$$\beta_{m+1} = \sqrt{w^H\tilde{S}|T|\tilde{S}^H w}. \tag{4.4}$$

Vergleich Aufwand Algorithmus 4.2 und 4.4

- Das Lösen der Gleichungssysteme mit $\tilde{S}$, $\tilde{S}^T$ und $|T|$ in Algorithmus 4.4 entspricht in etwa dem Lösen eines Gleichungssystems mit $SS^H = \tilde{S}|T|\tilde{S}^H$ in Algorithmus 4.2 in (4.2) und (4.3). Die Matrizen S und $\tilde{S}$ sind untere Dreiecksmatrizen und $|T|$ ist eine Diagonalmatrix.
- Statt einer Matrix-Vektormultiplikation mit S^H bzw. SS^H zur Berechnung von β_m in Algorithmus 4.2 (4.4) ist in Algorithmus 4.4 eine Multiplikation mit der Diagonalmatrix $|T|$ bzw. $|T|^{1/2}$ (i.A. billiger) durchzuführen.
- Eine Multiplikation mit A in Algorithmus 4.2 wird in Algorithmus 4.4 durch eine Multiplikation mit der Diagonalmatrix $B = T^{-1}$ (und zwei zusätzliche Vektoradditionen) ersetzt.

Algorithmus 4.4 ist vom Aufwand her wesentlich billiger als Algorithmus 4.2 und vergleichbar mit dem des unpräkonditionierten CSYM-Verfahrens.

Wir werden in Kapitel 6 im Zusammenhang mit der Block-SGS-Präkonditionierung auf diese Anwendung einer ungeblockten SGS-Präkonditionierung zurückkommen.

Bemerkung 4.3. Bei dem ungeblockten Jacobi-Präkonditionierer ist $S = D^{1/2}$ eine Diagonalmatrix. Die Aufwände in beiden implizit präkonditionierten CSYM-Verfahren, Algorithmus 4.2 und Algorithmus 4.3, sind identisch, nämlich pro Iterationsschritt zwei zusätzliche Matrix-Vektormultiplikationen mit einer Diagonalmatrix. Dies entspricht in etwa dem Aufwand des CSYM-Verfahrens mit SGS-Präkonditionierung und Eisenstat-Trick.

4.2 Der faktorisierte Präkonditionierer

Wir gehen in diesem Abschnitt von einem festen Präkonditionierer $M = M^T$ aus. Wie wir festgestellt haben, hängt das Verfahren von der gewählten Faktorisierung $M = SS^T$ ab. Welche Möglichkeiten bieten sich an?

4.2.1 Symmetrische Faktorisierungen

Die Takagi-Faktorisierung

Zunächst folgt aus Satz 2.15, dass sich jede komplex symmetrische Matrix symmetrisch faktorisieren lässt. Insbesondere gilt der folgende Satz [23].

Satz 4.4. *Sei $A \in \mathbb{C}^{n\times n}$. Dann ist A symmetrisch genau dann, wenn es eine Matrix $X \in \mathbb{C}^{n\times n}$ gibt mit $A = XX^T$.*

Beweis: [23] Man kann $X = UD$ wählen mit U unitär, $D = \mathrm{diag}(\sqrt{\sigma_1}, \sqrt{\sigma_2}, \ldots, \sqrt{\sigma_n})$ mit den Singulärwerten σ_i von A. □

Die Cholesky-Zerlegung

Definition 4.5. Sei $A = A^T$. Eine **Cholesky-Zerlegung** ist eine Zerlegung $A = LL^T$ mit einer regulären oberen Dreiecksmatrix L.

Wann existiert nun eine solche symmetrische Zerlegung in Dreiecksmatrizen?

Lemma 4.6. *Sei A symmetrisch und regulär. Dann gibt es eine nichtsinguläre untere Dreiecksmatrix L, so dass $A = LL^T$ genau dann, wenn alle führenden Untermatrizen von A nichtsingulär sind.*

Beweis: Die Gauß-Elimination für allgemeine Matrizen liefert genau dann eine eindeutige Zerlegung $A = LU$ in eine untere Dreiecksmatrix L mit Einheitsdiagonale und eine reguläre rechte obere Dreiecksmatrix U, wenn alle führenden Untermatrizen von A nichtsingulär sind [19]. Die Dreiecksmatrix U ist darstellbar als $U = D\tilde{U}$ mit einer oberen Dreiecksmatrix $\tilde{U}$ mit Einheitsdiagonale und einer Diagonalmatrix D. Auf Grund der Symmetrie von A gilt auch $A = U^T L^T = \tilde{U}^T D L^T$ und, da die Zerlegung eindeutig ist, folgt $L = \tilde{U}^T$ bzw. $DL^T = U$. Insbesondere ist die Cholesky-Zerlegung durch $A = \hat{L}\hat{L}^T$ mit $\hat{L} = LD^{1/2}$ gegeben. □

Selbst wenn die Cholesky-Zerlegung existiert, kann sie numerisch schlecht konditioniert und instabil sein.

Definition 4.7. Der **Wachstumsfaktor** ist gegeben durch $\rho = \max_{i,j} |\rho_{i,j}|$, wobei $|\rho_{i,j}| = \max_{k=1}^{n} |a_{i,j}^{(k)}|$ gilt.

Satz 4.8. *Wird eine Cholesky-Zerlegung berechnet und ist ϵ die Maschinengenauigkeit, so existiert eine Matrix $H \in \mathbb{C}^{n\times n}$, so dass*

$$A + H = LL^T,\ |h_{i,j}| \leq 5.01\epsilon|\rho_{i,j}|,\ i,j = 1,\ldots,n.$$

Beweis: s. [19].

Eine Pivotisierung hat das Ziel, den Wachstumsfaktor, das Maß für die Rückwärtsstabilität der Gauß-Elimination, zu beschränken. Dabei wird statt der Cholesky-Zerlegung von A die von $\tilde{A} = \Pi A \Pi^T$ mit einer Permutationsmatrix Π berechnet. Die Symmetrie der Matrix A bleibt erhalten und es ändert sich die Reihenfolge der Diagonalelemente.
Bei hermiteschen Matrizen sind zum Beispiel Matrizen mit ausschließlich positiven Eigenwerten, also die positiv definiten Matrizen, besonders günstig, da die Cholesky-Zerlegung ohne Pivotisierung durchgeführt werden kann. Der Wachstumsfaktor ist hier unabhängig von der gewählten Permutation.
Dies gilt auch für die Cholesky-Zerlegung komplex symmetrischer Matrizen, deren Eigenwerte alle positiven Real- und Imaginärteil haben, sogenannte CSPD (complex symmetric positive definite) Matrizen [20].
Vor allem bei indefiniten Matrizen sollte eine Block-Cholesky-Zerlegung benutzt werden [19], d.h.

$$A = LDL^T \text{ mit Blockdiagonalmatrix } D,$$
$$L \text{ linke untere Dreiecksmatrix.}$$

Die Matrix D hat nur (1×1)- und (2×2)-Blöcke. Bei der Bestimmung einer Block-Cholesky-Zerlegung gibt es unterschiedliche Strategien zur Wahl des Pivotelements [19]. Bei der sogenannten Rook-Pivotisierung [1], benannt nach den Zügen des Turmes beim Schach, wird das (1×1)- bzw. (2×2)-Pivotelement in einer iterativen Phase bestimmt. Insbesondere ist $\operatorname{cond}\left(|D||L|^T\right)$ bei dieser Strategie exponentiell in n beschränkt. Hieraus lässt sich auch eine Beschränkung des Vorwärtsfehlers durch $\operatorname{cond}(A)$ herleiten.
Andere Strategien, z.B. die partielle Pivotisierung, garantieren überhaupt keine Beschränkung von $\operatorname{cond}\left(|D||L|^T\right)$.
Von den indefiniten Matrizen haben sogenannte quasi-definite Matrizen die Eigenschaft [42], dass für jede symmetrische Permutation eine Zerlegung LDL^T existiert. Die Zerlegungen haben aber nicht alle die gleiche numerische Stabilität.

Definition 4.9. Wir nennen die Matrix L aus der Cholesky-Zerlegung einen **Cholesky-Faktor** und den Faktor $LD^{1/2}$ mit $D^{1/2}$ aus einer Faktorisierung $D = D^{1/2}(D^{1/2})^T$ mit L und D aus der Rook-Pivotisierung einen **R-Cholesky-Faktor**.

Eine symmetrische Faktorisierung einer komplex symmetrischen Matrix, nämlich die Takagi-Faktorisierung, lässt sich also immer finden, eine symmetrische Faktorisierung in Dreiecksmatrizen oder Block-Dreiecksmatrizen dagegen nicht. Häufig ist jedoch der Rechenaufwand zu hoch, um eine Faktorisierung zu bestimmen.

Da es um Präkonditionierer geht, suchen wir eine einfach zu berechnende symmetrische Faktorisierung einer Matrix M, welche A approximiert. Eine erste, besonders einfache Möglichkeit haben wir bereits bei der Anwendung der Präkonditionierung kennen gelernt, die SGS-Faktorisierung. Zu der Zerlegung $A = D - L - L^T$ mit einer Diagonalmatrix D und einer linken oberen Dreiecksmatrix L ist der Präkonditionierer gegeben durch

$$M = (D - L)D^{-1}(D - L^T)$$

und der Faktor ist somit

$$S = (D - L)D^{-1/2}.$$

Die Approximationsgüte von M ist gegeben durch die Restmatrix

$$R = A - M = LD^{-1}L^T.$$

4.2.2 Änderung der Singulärwerte

Im Folgenden untersuchen wir, wie sich die Singulärwerte der Matrix A durch T-Kongruenz mit $X = S^{-1}$ ändern, d.h. durch den Übergang auf die präkonditionierte Matrix $\hat{A} = S^{-1}AS^{-T}$ bei Präkonditionierung mit $M = SS^T$. Mit Hilfe von Satz 2.20, der variationellen Maximum-Beziehung des größten Singulärwertes, erhalten wir:

$$\sigma_1(S^{-1}AS^{-T}) \quad = \quad \max_{x^Hx=1} \frac{|x^T(S^{-1}AS^{-T})x|}{x^Hx}.$$

Der Maximalwert wird für den konjugierten Singulärvektor $\hat{v}_1$ zum Singulärwert $\sigma_1(\hat{A})$ angenommen, also

$$\sigma_1(S^{-1}AS^{-T}) = \frac{|\hat{v}_1^T(S^{-1}AS^{-T})\hat{v}_1|}{\hat{v}_1^H\hat{v}_1}.$$

Setzen wir $y_1 = S^{-T}\hat{v}_1$ in diese Gleichung ein, so ergibt sich:

$$\begin{aligned}\sigma_1(S^{-1}AS^{-T}) &= \frac{|y_1^T A y_1|}{y_1^H y_1}\frac{y_1^H y_1}{y_1^H \overline{S} S^T y_1} \\ &= \frac{|y_1^T A y_1|}{y_1^H y_1}\frac{\overline{\hat{v}}_1^H X X^H \overline{\hat{v}}_1}{\overline{\hat{v}}_1^H \overline{\hat{v}}_1}.\end{aligned}$$

In der Gleichung kommt also ein Rayleigh-Ritz-Quotient der hermitesch positiv definiten Matrix $XX^H = S^{-1}S^{-H}$ vor. Wir werden später eine ähnliche Darstellung, dann aber für alle Singulärwerte, herleiten.
Außerdem können wir $\sigma_1(A)$ in Abhängigkeit von $\hat{A}$ darstellen:

$$\begin{aligned}\sigma_1(A) &= \max_{x^H x=1} |\overline{x}^T A \overline{x}| \\ &= \max_{x^H x=1} |\overline{x}^T (SS^{-1}) A (S^{-T}S^T)\overline{x}| \\ &= \max_{x^H x=1} |(S^T\overline{x})^T S^{-1} A S^{-T} (S^T\overline{x})| \\ &= \max_{y^H S^{-1}S^{-H} y=1} |\overline{y}^T \hat{A} \overline{y}|,\end{aligned}$$

und umgekehrt $\sigma_1(\hat{A})$ in Abhängigkeit von A:

$$\begin{aligned}\sigma_1(\hat{A}) &= \max_{x^H x=1} |\overline{x}^T \hat{A} \overline{x}| \\ &= \max_{x^H x=1} |\overline{x}^T (S^{-1}AS^{-T})\overline{x}| \\ &= \max_{x^H x=1} |(S^{-T}\overline{x})^T A (S^{-T}\overline{x})| \\ &= \max_{y^H SS^H y=1} |\overline{y}^T A \overline{y}|.\end{aligned}$$

Den folgenden Satz benötigen wir, um einen Satz über die Änderung der Singulärwerte von A bei der Transformation XAX^T mit einer regulären Matrix X zu beweisen.

Satz 4.10. *(Ostrowski)*
Seien B, $X \in \mathbb{C}^{n\times n}$ und B hermitesch und X regulär.
Die Eigenwerte $\lambda_i(B)$ bzw. $\lambda_i(XX^H)$ von B und XX^H seien absteigend geordnet.
Für jedes $k = 1, 2, \ldots, n$ gibt es eine positive reelle Zahl θ_k mit

$$\lambda_n(XX^H) \leq \theta_k \leq \lambda_1(XX^H),$$

so dass

$$\lambda_k(XBX^H) = \theta_k \lambda_k(B).$$

Beweis: [23] über den Trägheitssatz von Sylvester und die Weyl-Ungleichung.

Satz 4.11. *Seien* $A,\ X \in \mathbb{C}^{n\times n}$, $A = A^T$, X *regulär und*

$$\hat{A} = XAX^T.$$

Seien Takagi-Faktorisierungen (s. Satz 2.15) gegeben durch

$$\begin{aligned} A &= V\Sigma V^T,\ \Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n),\ \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0, \\ \hat{A} &= UMU^T,\ M = \operatorname{diag}(\mu_1, \mu_2, \ldots, \mu_n),\ \mu_1 \geq \mu_2 \geq \cdots \geq \mu_n \geq 0. \end{aligned}$$

Seien die Eigenwerte $\lambda_i(XX^H)$ *absteigend sortiert.*
Dann gibt es für jedes $k = 1, 2, \ldots, n$ *eine reelle Zahl* $\theta_k \in [\lambda_n(XX^H), \lambda_1(XX^H)]$, *so dass*

$$\mu_k = \theta_k \sigma_k.$$

Beweis: [23] Die Singulärwerte μ_i^2 von $\hat{A}$ sind die Eigenwerte von $\hat{A}\hat{A}^H = \hat{A}\overline{\hat{A}}$. Also gilt mit Anwendung von Satz 4.10 in (4.5) und (4.6) mit $\tilde{\theta}_k, \hat{\theta}_k \in [\lambda_n(XX^H), \lambda_1(XX^H)]$:

$$\begin{aligned} \mu_k^2 &= \lambda_k(XAX^T\overline{XA}X^H) \\ &= \lambda_k(X[AX^T\overline{XA}]X^H) && (4.5) \\ &= \tilde{\theta}_k \lambda_k(AX^T\overline{XA}) \\ &= \tilde{\theta}_k \lambda_k(\overline{XA}AX^T) \\ &= \tilde{\theta}_k \lambda_k(XA\overline{A}X^H) && (4.6) \\ &= \tilde{\theta}_k \hat{\theta}_k \lambda_k(A\overline{A}) \\ &= \tilde{\theta}_k \hat{\theta}_k \sigma_k^2. \end{aligned}$$

Mit $\mu_k = \sqrt{\tilde{\theta}_k \hat{\theta}_k}\sigma_k$ und $\theta_k = \sqrt{\tilde{\theta}_k \hat{\theta}_k}$ folgt die Behauptung. □

Nach Satz 4.11 gilt $\sigma_k(\hat{A}) = \theta_k \sigma_k(A)$, wobei der maximale Wertebereich der θ_k auf Grund der Beziehung $X = S^{-1}$ mit dem Faktor S der Präkonditionierung gegeben ist durch:

$$[\lambda_{min}(S^{-1}S^{-H}), \lambda_{max}(S^{-1}S^{-H})].$$

Eine θ_k-Wert ist aber schon durch den Vorgänger und Nachfolger eingegrenzt, wie folgendes Lemma zeigt.

Lemma 4.12. *Jedes $\theta_k \in [\lambda_{min}(XX^H), \lambda_{max}(XX^H)]$ aus Satz 4.11 ist durch θ_{k-1} bzw. θ_{k+1} und die Singulärwerte von A in folgender Weise eingeschränkt.*

$$\theta_k \geq \theta_{k+1} \frac{\sigma_{k+1}(A)}{\sigma_k(A)} \text{ für } k = 1, \ldots, n-1 \tag{4.7}$$

$$\theta_k \leq \theta_{k-1} \frac{\sigma_{k-1}(A)}{\sigma_k(A)} \text{ für } k = 2, \ldots, n \tag{4.8}$$

Beweis: Folgt direkt aus der absteigenden Sortierung der $\sigma_k(\hat{A})$ und $\sigma_k(A)$ in Satz 4.11 und der Darstellung

$$\sigma_k(\hat{A}) = \theta_k \sigma_k(A).$$

□

4.2.3 Gewünschte Eigenschaften

Welche Eigenschaften sollte nun ein effizienter Präkonditionierer $M = SS^T$ im CSYM-Verfahren haben? Eine gewünschte Eigenschaft von $\hat{A}$ ist, dass $\hat{A}$ mehr mehrfache Singulärwerte hat als A (s. Satz 3.38.) Diese Eigenschaft kann man nun über die Größe θ_k aus Satz 4.11 charakterisieren.

Lemma 4.13. *Sei $m \geq 1$. Es gilt*

$$\sigma_k(\hat{A}) = \sigma_{k+1}(\hat{A}) = \cdots = \sigma_{k+m}(\hat{A}),$$

genau dann, wenn

$$\theta_{k+l} = \theta_k \frac{\sigma_k(A)}{\sigma_{k+l}(A)} \text{ für } l = 1, \ldots, m.$$

Beweis: Analog Lemma 4.12.

Bemerkung 4.14. In diesem Fall ist also das Verhältnis θ_{k+l}/θ_k maximal (s. Lemma 4.12 (4.8)). Es gilt insbesondere $\theta_k \leq \theta_{k+1} \leq \cdots \leq \theta_{k+m}$.

Damit alle Singulärwerte um den Wert 1 geclustert sind, ist insbesondere notwendig, dass $\theta_{k+l} \approx \frac{1}{\sigma_{k+l}(A)}$ für $l = 0, \ldots, m$.
Eine andere Forderung ist, dass die Kondition von $\hat{A}$ kleiner als die von A ist (s. Satz 3.39).

Notation. Wir schreiben $\theta_{1/n}$ für den Quotienten $\frac{\theta_1}{\theta_n}$ mit θ_1 und θ_n aus Satz 4.11.

Lemma 4.15. *Seien A und $\hat{A}$ wie in Satz 4.11 gegeben. Dann gilt*

$$cond\left(\hat{A}\right) < cond(A) \Leftrightarrow \qquad\qquad \theta_{1/n} < 1.$$

Beweis: Die Behauptung folgt direkt mit der Darstellung von $\sigma_1(\hat{A})$ und $\sigma_n(\hat{A})$ aus Satz 4.11. □

Bemerkung 4.16. Der Veränderungsfaktor θ_1 des größten Singulärwertes muss also kleiner als der Veränderungsfaktor θ_n des kleinsten Singulärwertes sein.
Der Idealfall ist natürlich cond $\left(\hat{A}\right) = 1$. Nach Lemma 4.13 bedeutet dies dann insbesondere, dass die Folge der θ_k monoton wächst.

4.2.4 Die Minimaler-Bereich-Eigenschaft

Wählen wir eine hermitesch positiv definite Matrix $\tilde{M}$ als Präkonditionierer, so entspricht dies implizit der Präkonditionierung mit einer Matrix $M = SS^T$, wobei die Faktoren nicht explizit berechnet werden. Diese Faktoren haben eine besondere Eigenschaft, die wir im Folgenden einführen werden. Wir charakterisieren aus der Menge aller möglichen Faktorisierungen zu einem festen Präkonditionierer $M = M^T$ Zerlegungen $M = SS^T$, die diese Eigenschaft erfüllen.

Satz 4.17. *Sei die Singulärwertzerlegung von S gegeben durch*

$$S = V_S \Sigma_S W_S^H. \tag{4.9}$$

Dann gilt

$$[\sigma_n(M), \sigma_1(M)] \subseteq [\sigma_n(S^H S), \sigma_1(S^H S)]. \tag{4.10}$$

Im Spezialfall $W_S^H \overline{W}_S = I$ sind die Singulärwerte von $M = SS^T$ und $S^H S$ identisch und es ist insbesondere

$$[\sigma_n(M), \sigma_1(M)] = [\sigma_n(S^H S), \sigma_1(S^H S)]. \tag{4.11}$$

Zum Beweis des Satzes benötigen wir folgendes Lemma.

Lemma 4.18. *Sei $S \in \mathbb{C}^{n\times n}$, dann gilt*

$$\sigma_k(S) = \sigma_k(S^T) = \sigma_k(\overline{S}) = \sigma_k(S^H), \tag{4.12}$$

$$\sigma_k^2(S) = \sigma_k(S^H S). \tag{4.13}$$

Beweis: Betrachten wir die Singulärwertzerlegung von S, so ist (4.12) erfüllt.
Da $\sigma^2(S) = \text{spek}\left(S^H S\right) = \text{spek}\left(S^H S\right) = \sigma(S^H S)$ gilt, folgt (4.13). □

Nun zurück zum Beweis von Satz 4.17.
Es gilt nach Lemma 4.18 (4.13)

$$\sigma_k(S^H S) = \sigma_k^2(S).$$

Weiter gilt mit der Darstellung (4.9):

$$\begin{aligned} \sigma_k(M) &= \sigma_k(SS^T) \\ &= \sigma_k(V_S \Sigma_S W_S^H \overline{W}_S \Sigma_S V_S^T) \\ &= \sigma_k(\Sigma_S W_S^H \overline{W}_S \Sigma_S) \end{aligned} \tag{4.14}$$

Da die Matrix $W_S^H \overline{W}_S$ unitär ist, gilt $\sigma_k(W_S^H \overline{W}_S) = 1$ und weiter mit Satz 4.11

$$\sigma_k(M) = \theta_k(\Sigma_S^2)\sigma_k(W_S^H \overline{W}_S) = \theta_k(\Sigma_S^2).$$

Die Werte $\theta_k(\Sigma_S^2)$ sind insbesondere absteigend sortiert und stimmen mit den Singulärwerten $\sigma_k(SS^T)$ überein.
Gilt $W_S^H \overline{W}_S = I$, so folgt für alle $k = 1, \ldots, n$ aus (4.14):

$$\sigma_k(\Sigma_S^2) = \sigma_k(M) \Rightarrow \sigma_k(S^H S) = \sigma_k(M).$$

Dies beweist (4.11).
Wir schätzen die extremalen Singulärwerte von SS^T mit Hilfe der euklidischen Norm ab:
Es gilt

$$\begin{aligned}\sigma_1(M) &= \sigma_1(SS^T)\\ &= \|SS^T\|_2\\ &\leq \|S\|_2\|S^T\|_2.\end{aligned}$$

Mit Lemma 4.18 (4.12) erhalten wir so:

$$\begin{aligned}\sigma_1(M) &\leq \|S\|_2\|S^T\|_2\\ &= \sigma_1(S)\sigma_1(S^T)\\ &= \sigma_1^2(S)\\ &= \sigma_1(S^HS).\end{aligned}$$

Dann gilt analog für M^{-1}:

$$\sigma_1(M^{-1}) \leq \sigma_1(S^{-1}S^{-H}),$$

also

$$\frac{1}{\sigma_n(M)} \leq \frac{1}{\sigma_n(S^HS)}$$

und damit

$$\sigma_n(M) \geq \sigma_n(S^HS).$$

Insgesamt haben wir also gezeigt:

$$[\sigma_n(M), \sigma_1(M)] \subseteq [\sigma_n(S^HS), \sigma_1(S^HS)].$$

□

Folgerung 4.19. Faktorisiert man einen komplex symmetrischen Präkonditionierer M als $M = SS^T$ so, dass die Singulärwerte von M mit denen von S^HS übereinstimmen, so gilt mit $X = S^{-1}$ lt. Satz 4.11:

$$\theta_k \in [\sigma_n(X^TX), \sigma_1(X^TX)] = [\sigma_n(XX^H), \sigma_1(XX^H)].$$

Faktoren mit dieser Eigenschaft scheinen eine natürliche Wahl für den Faktor S in $M = SS^T$ zu sein, wenn wir bedenken, dass ein solcher implizit angewendet wird, wenn wir von einem hermitesch positiv definiten Präkonditionierer $\tilde{M}$ ausgehen. Erste numerische Ergebnisse zeigen, dass bei der Wahl eines solchen Faktors $\theta_{1/n}$ minimal unter allen möglichen Faktorisierungen des Präkonditionierers M ist, und daher die Kondition der präkonditionierten Matrix am günstigsten ist.

Definition 4.20. Ein Faktor S von $M = SS^T$ hat die **Minimaler-Bereich-Eigenschaft**, wenn folgende Gleichung erfüllt ist:

$$[\sigma_n(M), \sigma_1(M)] = [\sigma_n(S^H S), \sigma_1(S^H S)].$$

Die Minimaler-Bereich-Eigenschaft wird insbesondere durch eine Singulärwertzerlegung von S mit $W_S \in \mathbb{R}^{n \times n}$ garantiert. Daher folgt:

Lemma 4.21. *Der Takagi-Faktor $S = V_S \Sigma_S$ von M erfüllt die Minimaler-Bereich-Eigenschaft und es gilt sogar*

$$\sigma_k(SS^T) = \sigma_k(S^H S), \text{ für } k = 1, \ldots, n$$

Beweis: Wählen wir den Takagi-Faktor, so gilt $W_S = I$. □

Daher könnten sich Präkonditionierer $M \approx A$, die in der Takagi-Faktorisierung vorliegen, als Präkonditionierer eignen.

Mit der in Satz 4.17 definierten Singulärwertzerlegung von S erhalten wir:

$$\hat{A} = S^{-1} A S^{-T} = W_S \Sigma_S^{-1} V_S^H A \overline{V_S} \Sigma_S^{-1} W_S^T$$

d.h. die Singulärwerte von $\hat{A}$ werden nicht offensichtlich durch die Matrizen W_S beeinflusst. Dagegen ist es sinnvoll zu fordern, dass $V_S^H \approx V$, mit V aus $A = V \Sigma V^T$.

4.2.5 Transformationen

Sei im Folgenden der Präkonditionierer M fest. Wir untersuchen den Zusammenhang zwischen unterschiedlichen Faktorisierungen.

Lemma 4.22. *Sei M symmetrisch faktorisiert in der Form $M = S_1 S_1^T$. Eine weitere Matrix S_2 erfüllt $M = S_2 S_2^T$ genau dann, wenn es eine komplex-orthogonale Matrix P gibt, so dass $S_2 = S_1 P$ gilt.*

Beweis:
i) Seien S_1 und S_2 symmetrische Faktoren von M. Dann gilt

$$\begin{aligned} & S_1 S_1^T = S_2 S_2^T \\ & \Rightarrow S_1^{-1} S_2 S_2^T S_1^{-T} = (S_1^{-1} S_2)(S_1^{-1} S_2)^T = I, \end{aligned}$$

d.h. $P = S_1^{-1} S_2$ ist eine komplex-orthogonale Matrix und $S_2 = S_1 P$.
ii) Wir betrachten $S_2 = S_1 P$. Da P komplex-orthogonal ist, gilt

$$S_2 S_2^T = (S_1 P)(S_1 P)^T = S_1 S_1^T = M.$$

Damit ist auch S_2 symmetrischer Faktor von M. □

Satz 4.23. *Seien zwei symmetrische Zerlegungen von M gegeben, einmal mit dem Takagi-Faktor S und eine weitere mit $\tilde{S} = SP$ mit einer komplex-orthogonalen Matrix P. Dann gilt*

$$\lambda_k(\tilde{S}^H \tilde{S}) = \theta_k \lambda_k(S^H S),\ \theta_k \in [\frac{1}{\sigma_1^2(P)}, \sigma_1^2(P)]$$

und speziell

$$\theta_k = \frac{\lambda_k(\tilde{S}^H \tilde{S})}{\sigma_k(\tilde{S}\tilde{S}^T)} = \frac{\lambda_k(\tilde{S}^H \tilde{S})}{\sigma_k(M)}, \tag{4.15}$$

$$\theta_1 \in [1, \sigma_1^2(P)],\ \theta_n \in [\frac{1}{\sigma_1^2(P)}, 1]. \tag{4.16}$$

Für den Beweis von Satz 4.23 benötigen wir folgendes Lemma.

Lemma 4.24. *Für jede komplex-orthogonale Matrix P gilt*

$$\sigma_n(P) = \frac{1}{\sigma_1(P)}, \tag{4.17}$$

$$cond(P) = \sigma_1^2(P). \tag{4.18}$$

Beweis: Es gilt

$$\sigma_n(P) = \frac{1}{\sigma_1(P^{-1})} = \frac{1}{\sigma_1(P^T)} = \frac{1}{\sigma_1(P)}.$$

Damit haben wir (4.17) gezeigt und erhalten außerdem

$$\text{cond}\,(P) = \frac{\sigma_1(P)}{\sigma_n(P)} = \sigma_1^2(P).$$

□

Zurück zum Beweis von Satz 4.23: Es ist

$$M = \tilde{S}\tilde{S}^T = (SP)(SP)^T = SS^T.$$

Für die Veränderung der Singulärwerte nach Satz 4.11 benötigen wir Aussagen über das Spektrum von S^HS bzw. $\tilde{S}^H\tilde{S}$. Mit Satz 4.10 (Ostrowski) erhalten wir

$$\begin{aligned}&\lambda_k(\tilde{S}^H\tilde{S}) = \lambda_k(P^HS^HSP) = \theta_k\lambda_k(S^HS)\\ &\text{mit } \theta_k \in [\lambda_{min}(P^HP), \lambda_{max}(P^HP)].\end{aligned}$$

Es gilt mit Lemma 4.24 (4.17)

$$\theta_k \in [\sigma_n^2(P), \sigma_1^2(P)] = [\frac{1}{\sigma_1^2(P)}, \sigma_1^2(P)].$$

Aus der Gleichung (4.18) folgt, dass $\sigma_1^2(P) \geq 1$ ist.
Nach Lemma 4.21 gilt für jeden Takagi-Faktor S:

$$\lambda_k(S^HS) = \sigma_k(SS^T) = \sigma_k(M) \text{ für } k = 1, \ldots, n.$$

Für jeden Faktor $\tilde{S}$ in einer symmetrischen Zerlegung $M = \tilde{S}\tilde{S}^T,\ \tilde{S} = SP$ gilt also

$$\theta_k = \frac{\lambda_k(\tilde{S}^H\tilde{S})}{\sigma_k(M)}.$$

Weiter gilt nach Satz 4.17

$$[\sigma_n(\tilde{S}\tilde{S}^T), \sigma_1(\tilde{S}\tilde{S}^T)] \subseteq [\lambda_{min}(\tilde{S}^H\tilde{S}), \lambda_{max}(\tilde{S}^H\tilde{S})].$$

Somit gilt $\theta_1 \geq 1$, $\theta_n \leq 1$ und damit $\theta_1 \in [1, \sigma_1^2(P)]$ und $\theta_n \in [\frac{1}{\sigma_1^2(P)}, 1]$. □

Folgerung 4.25. Ist $P \in \mathbb{R}^{n\times n}$, so gilt $P^HP = P^TP = I$ und somit $\theta_k = 1$, d.h. die Singulärwerte von $\tilde{S}$ und S unterscheiden sich nicht.

Folgerung 4.26. Im implizit präkonditionierten CSYM-Verfahren sind Gleichungssysteme mit $\tilde{S}\tilde{S}^H$ zu lösen. Nun gilt

$$\lambda_k(\tilde{S}\tilde{S}^H) = \lambda_k(\tilde{S}^H\tilde{S}).$$

Ist S der Takagi-Faktor, so gilt für die Kondition

$$\text{cond}\left(SS^H\right) \leq \text{cond}\left(\tilde{S}\tilde{S}^H\right).$$

Unter allen möglichen symmetrischen Faktorisierungen von M minimiert der Takagi-Faktor also die Kondition von $\tilde{M} = \tilde{S}\tilde{S}^H$.

Im Folgenden untersuchen wir den Zusammenhang zwischen präkonditionierten Matrizen, die wir durch unterschiedliche Faktorisierungen des Präkonditionierers M erhalten. Aus Lemma 4.22 folgt:

Lemma 4.27. *Seien $M = SS^T = \tilde{S}\tilde{S}^T$ zwei Faktorisierungen, davon S der Takagi-Faktor. Dann gilt:*

$$\sigma_k(\tilde{S}^{-1}A\tilde{S}^{-T}) = \sigma_k(P^TS^{-1}AS^{-T}P) = \theta_k\sigma_k(S^{-1}AS^{-T})$$

mit $\theta_k \in [\lambda_{min}(P^HP), \lambda_{max}(P^HP)]$.

Sei $\hat{A} = S^{-1}AS^{-T}$ mit einem Takagi-Faktor S. Die Matrix $\tilde{A} = P^T\hat{A}P$ mit P aus Lemma 4.27 erhalten wir durch T-Kongruenz der Matrix $\hat{A}$ mit komplex-orthogonaler Matrix P. Es handelt sich insbesondere um eine Ähnlichkeitstransformation, so dass die Eigenwerte von $\tilde{A}$ und $\hat{A}$ identisch sind. Eine weitere Invariante liefert Satz 4.29. Dazu beweisen wir zunächst folgendes Lemma, welches in dieser Form (Produkt der Singulärwerte) in der Literatur nicht gefunden werden konnte. Der Beweisansatz über die Singulärwerte von komplex-orthogonalen Matrizen ist zu finden in [13].

Lemma 4.28. *Ist P eine komplex-orthogonale Matrix, so gilt $\prod_{k=1}^{n}\sigma_k(P) = 1$.*

Beweis: Es gilt $\sigma_k(P) = \sigma_k(P^T)$ für $k = 1, \ldots, n$. Da zusätzlich $P^{-1} = P^T$ gilt, folgt $\sigma_k(P) = \frac{1}{\sigma_{n-k+1}(P^T)}$ und damit $\sigma_k(P) = \frac{1}{\sigma_{n-k+1}(P)}$. Somit gilt $\sigma_k(P)\sigma_{n-k+1}(P) = 1$ und daraus folgt $\prod_{k=1}^{n}\sigma_k(P) = \prod_{k=1}^{\lfloor n/2\rfloor}\sigma_k(P)\sigma_{n-k+1}(P) = 1$.
Ist n ungerade, so gilt für $k = \lfloor n/2\rfloor+1$ insbesondere $n-k+1 = k$ und damit $\sigma_k = \frac{1}{\sigma_k} = 1$.
□

Satz 4.29. *Seien $\sigma_k(P)$, $\sigma_k(\hat{A})$ und $\sigma_k(P^T\hat{A}P)$ die jeweils absteigend sortierten Singulärwerte einer komplex-orthogonalen Matrix P, einer Matrix $\hat{A}$ und der Matrix $P^T\hat{A}P$. Dann gilt*

$$\prod_{k=1}^{l} \sigma_k(P^T\hat{A}P) \leq \prod_{k=1}^{l} \sigma_k^2(P)\sigma_k(\hat{A}) \text{ für } l = 1, \ldots, n-1, \tag{4.19}$$

$$\prod_{k=1}^{n} \sigma_k(P^T\hat{A}P) = \prod_{k=1}^{n} \sigma_k(\hat{A}). \tag{4.20}$$

Beweis: Zum Beweis nutzen wir Satz 2.21, der Aussagen über die Produkte aufeinander folgender Singulärwerte einer Produktmatrix CD macht. Wenden wir (2.3) aus Satz 2.21 für $l \in \{1, \ldots, n\}$ beliebig mit $C = P^T$ und $D = \hat{A}P$ und nochmals mit $C = \hat{A}$ und $D = P$ an, so ist die Ungleichung (4.19) bewiesen.
Mit Lemma 4.28 folgt aus (2.4) (Satz 2.21) die Gleichheit für $l = n$ und damit (4.20). □

Setzen wir $\sigma_k(P^T\hat{A}P) = \theta_k\sigma_k(\hat{A})$ mit $\theta_k \in [\lambda_{min}(P^HP), \lambda_{max}(P^HP)]$ in (4.19) und (4.20) (s. Satz 4.11) ein, so erhalten wir

$$\prod_{k=1}^{l} \theta_k \leq \prod_{k=1}^{l} \sigma_k^2(P) \text{ für } l = 1, \ldots, n.$$

Daraus folgt mit Satz 2.22

$$\sum_{k=1}^{l} \theta_k \leq \sum_{k=1}^{l} \sigma_k^2(P) \text{ für } l = 1, \ldots, n.$$

Aus Gleichung (4.20) folgt sogar

$$\prod_{k=1}^{n} \theta_k = \prod_{k=1}^{n} \sigma_k^2(P)$$

und aus Lemma 4.28

$$\prod_{k=1}^{n} \theta_k = 1.$$

Haben nicht alle θ_k den Wert 1, so muss es sowohl Werte kleiner als auch größer 1 geben.

Bemerkung 4.30. Sei l_*, so dass $\sigma_{l_*}(P)$ der kleinste Singulärwert größer 1 ist.
Die obere Schranke $\tau(l) = \prod_{k=1}^{l} \sigma_k^2(P)\sigma_k(\hat{A})$ für $\prod_{k=1}^{l} \sigma_k(P^T\hat{A}P)$ ist für $l = 1, \ldots, l_*$ streng monoton wachsend und für $l = l_* + 1, \ldots, n$ monoton fallend. Die Abschätzung in (4.19) kann aber eine starke Überschätzung sein.

Sei nun $M = SS^T$ mit dem Takagi-Faktor S und $\hat{A} = S^{-1}AS^{-T}$. Jede präkonditionierte Matrix $\tilde{A} = \tilde{S}^{-1}A\tilde{S}^{-T}$ liegt in der Äquivalenzklasse von $\hat{A}$ bezüglich der T-Transformation von $\hat{A}$ mit komplex-orthogonaler Transformationsmatrix P und hat damit folgende Eigenschaften:

- Das Spektrum von $\tilde{A}$ ist durch das von $M^{-1}A$ gegeben, denn
$$\text{spek}\left(S^{-1}AS^{-T}\right) = \text{spek}\left(S^{-T}S^{-1}A\right).$$
- Aus Satz 2.23 für $p = 1$ folgt
$$\sum_{k=1}^{l} |\lambda_k(M^{-1}A)| \leq \sum_{k=1}^{l} \sigma_k(\tilde{A}) \text{ für } l = 1, \ldots, n.$$
- Der Wert $\prod_{k=1}^{n} \sigma_k(\tilde{A})$ ist für alle Matrizen der Äquivalenzklasse gleich.

Eine Matrix $\hat{A}$ mit minimalem Wert $\tau(l) = \prod_{k=1}^{l} \sigma_k(\hat{A})$ für $l = 1, \ldots, n-1$ unter allen Matrizen der Äquivalenzklasse garantiert nach Satz 2.22, dass $\sigma_1(\hat{A})$ möglichst klein und $\sigma_n(\hat{A})$ möglichst groß, und damit die Kondition möglichst klein ist. Die schwächere minimale Summenbedingung aus Satz 2.22 würde dafür bereits ausreichen. In der Ähnlichkeitsklasse zu $\hat{A}$ wird die stärkere minimale Produktbedingung durch jede normale Matrix erreicht, da in diesem Fall die Absolutbeträge der Eigenwerte mit den Singulärwerten übereinstimmen (s. Satz 2.24). Die normale Matrix wird aber i.A. nicht in der Äquivalenzklasse liegen.
Aus Satz 2.23 folgt für $p = 2$:

$$\sum_{k=1}^{l} |\lambda_k(\tilde{A})|^2 \leq \sum_{k=1}^{l} \sigma_k^2(\tilde{A}) \text{ für } l = 1, \ldots, n.$$

Mit Satz 2.14 (Abweichung von der Normalität) folgt mit der echten oberen Dreiecksmatrix N aus der Schurzerlegung

$$\|N\|_F^2 = \sum_{k=1}^{n} \sigma_k^2(\tilde{A}) - \sum_{k=1}^{n} |\lambda_k(\tilde{A})|^2.$$

Sei

$$N_l = \sum_{k=1}^{l} \sigma_k^2(\tilde{A}) - \sum_{k=1}^{l} |\lambda_k(\tilde{A})|^2.$$

Bisherige numerische Untersuchungen legen nahe, dass N_l für $l = 1, \ldots, n$ für die Matrix $\hat{A}$ minimal unter allen Matrizen $\tilde{A}$ der Äquivalenzklasse zu $\hat{A}$ ist. Insbesondere würde hieraus folgen, dass $\sigma_1(\hat{A}) \geq |\lambda_1(\hat{A})|$ minimal ist unter allen Matrizen $\tilde{A}$. Die Matrix $\hat{A}^{-1}$ liegt analog in einer Äquivalenzklasse mit allen Matrizen $\tilde{A}^{-1} = PA^{-1}P^T$ mit komplex-orthogonaler Matrix P. Hier würde auch gelten, dass $\sigma_1(\hat{A}^{-1}) \geq |\lambda_1(\hat{A}^{-1})|$ minimal unter allen Matrizen $\tilde{A}^{-1}$ der Äquivalenzklasse zu $\hat{A}^{-1}$ ist. Daraus würde dann folgen, dass $\sigma_n(\hat{A}) \leq |\lambda_n(\hat{A})|$ maximal unter allen Matrizen $\tilde{A}$ der Äquivalenzklasse zu $\hat{A}$ ist. Daher wäre die Kondition von $\hat{A}$ minimal in der Äquivalenzklasse.

Bemerkung 4.31. Es gibt aber Fälle, in denen eine Präkonditionierung mit dem Takagi-Faktor S trotz der Minimaler-Bereich-Eigenschaft nicht besser als die Präkonditionierung mit einem beliebigen Faktor $\tilde{S} = SP$ ist. Dies ist z.B. der Fall, wenn $\hat{A} = S^{-1}AS^{-T}$ mit P kommutiert, denn in diesem Fall gilt

$$\begin{aligned}\sigma(\tilde{A}) &= \sigma(P^T\hat{A}P) \\ &= \sigma(P^TP\hat{A}) \\ &= \sigma(\hat{A}).\end{aligned}$$

In exakter Arithmetik sind in diesem Fall die Singulärwerte der präkonditionierten Matrix unabhängig von der gewählten Faktorisierung. Bei der Rechnung in Gleitpunktarithmetik ist dies natürlich kaum der Fall.

4.3 Zusammenfassung

Wir sind zuerst von einem explizit präkonditionierten CSYM-Verfahren mit einem komplex symmetrischen Präkonditionierer M ausgegangen. Anschließend haben wir ein implizit präkonditioniertes CSYM-Verfahren formuliert, und dabei festgestellt, dass die Faktorisierung von $M = SS^T$ in der Form $\tilde{M} = SS^H$ in den Algorithmus eingeht. Anschließend haben wir die Änderung der Singulärwerte bei T-Kongruenz untersucht. Unter den möglichen Faktorisierungen zeichnet sich der Takagi-Faktor S dadurch aus, dass er die Minimaler-Bereich-Eigenschaft besitzt. Daraus folgt, dass die Kondition von $\tilde{M} = SS^H$ minimal ist. Jeder Faktor $\tilde{S}$, der nicht die Minimaler-Bereich-Eigenschaft hat, ist darstellbar als $\tilde{S} = PS$ mit einer „echt“ komplex-orthogonalen Matrix P. Das Produkt

der Singulärwerte der präkonditionierten Matrix ist unabhängig von der gewählten Faktorisierung. Wir vermuten, dass eine mit einem Takagi-Faktor präkonditionierte Matrix Singulärwerte hat, die die geringsten quadratischen Abweichungen von den Eigenwerten haben, also „möglichst“ normal ist unter allen mit einem festen Präkonditionierer M präkonditionierten Matrizen. In einem solchen Fall ist die Kondition der präkonditionierten Matrix bei festem Präkonditionierer M minimal.

Kapitel 5

Deflation

In diesem Kapitel setzen wir voraus:

Voraussetzung 5.1. Die Matrix A ist regulär und damit $\sigma_k > 0$, für $k = 1, \ldots, n$.

Wir definieren einen Präkonditionierer über einen Takagi-Faktor, der aus k Singulärwerten und den zugehörigen k Singulärvektoren aus der Takagi-Faktorisierung von A besteht. Im Folgenden schreiben wir $\sigma_1, \ldots, \sigma_k$ und $v_1, \ldots, v_k$ für die so gewählten Singulärpaare. Dann setzen wir

$$V_k = [v_1, \ldots, v_k] \in \mathbb{C}^{n\times k},\ \Sigma_k = \operatorname{diag}(\sigma_1, \ldots, \sigma_k) \in \mathbb{C}^{k\times k},$$
$$\tilde{W} = [\tilde{w}_1, \ldots, \tilde{w}_{n-k}] \in \mathbb{C}^{n\times(n-k)},\ \text{wobei } V_k^H \tilde{W} = 0,\ \tilde{W}^H \tilde{W} = I_{n-k}.$$

Mit diesen Größen setzen wir als Faktor des Präkonditionierers $M = SS^T$

$$S = \begin{pmatrix} V_k & \tilde{W} \end{pmatrix} \begin{pmatrix} \Sigma_k^{1/2} & 0 \\ 0 & I_{n-k} \end{pmatrix}. \tag{5.1}$$

Der folgende Satz gibt Auskunft darüber, wie sich die Singulärwerte bei dieser Präkonditionierung ändern.

Satz 5.2. *Die präkonditionierte Matrix*

$$\hat{A} = S^{-1} A S^{-T}$$

mit dem Takagi-Faktor S aus (5.1), der die Singulärwerte $\sigma_1, \ldots, \sigma_k$ berücksichtigt, hat mindestens k-mal den Singulärwert 1 und die Singulärwerte $\sigma_{k+1}, \ldots, \sigma_n$.

Beweis: Sei die Takagi-Faktorisierung von A gegeben durch

$$A = V\Sigma V^T = (V\Sigma^{1/2})(V\Sigma^{1/2})^T \text{ und damit} \tag{5.2}$$
$$\hat{A} = S^{-1} V \Sigma^{1/2} (V\Sigma^{1/2})^T S^{-T} = (S^{-1} V \Sigma^{1/2})(S^{-1} V \Sigma^{1/2})^T. \tag{5.3}$$

Wir nehmen an, dass in der Takagi-Faktorisierung $A = V\Sigma V^T$ die ersten k Singulärwerte und Singulärvektoren genau diejenigen sind, die auch in S verwendet werden, d.h.

$$A = \begin{pmatrix} V_k & W_{n-k} \end{pmatrix} \begin{pmatrix} \Sigma_k & 0 \\ 0 & \Sigma_{n-k} \end{pmatrix} \begin{pmatrix} V_k^T \\ W_{n-k}^T \end{pmatrix}. \tag{5.4}$$

Damit erhalten wir für $B = S^{-1}V\Sigma^{1/2}$

$$B = \begin{pmatrix} \Sigma_k^{-1/2} & 0 \\ 0 & I_{n-k} \end{pmatrix} \begin{pmatrix} V_k^H \\ \tilde{W}^H \end{pmatrix} V\Sigma^{1/2}.$$

Sei $C = \begin{pmatrix} V_k^H \\ \tilde{W}^H \end{pmatrix} V$, dann gilt

$$\begin{aligned} C &= \begin{pmatrix} V_k^H \\ \tilde{W}^H \end{pmatrix} \begin{pmatrix} V_k & W_{n-k} \end{pmatrix} \\ &= \begin{pmatrix} V_k^H V_k & V_k^H W_{n-k} \\ \tilde{W}^H V_k & \tilde{W}^H W_{n-k} \end{pmatrix}. \end{aligned}$$

Es gilt $\tilde{W}^H V_k = 0 = V_k^H W_{n-k}$, da die Spaltenvektoren von $\tilde{W}$ und W_{n-k} aus dem Orthogonalraum zu dem durch die Spaltenvektoren von V_k aufgespannten Unterraum sind.

Damit gilt $C = \begin{pmatrix} I_k & 0 \\ 0 & \tilde{W}^H W_{n-k} \end{pmatrix}$ und, eingesetzt in B,

$$B = \begin{pmatrix} \Sigma_k^{-1/2} & 0 \\ 0 & I_{n-k} \end{pmatrix} \begin{pmatrix} I_k & 0 \\ 0 & \tilde{W}^H W_{n-k} \end{pmatrix} \Sigma^{1/2} = \begin{pmatrix} I_k & 0 \\ 0 & \hat{B}_{2,2} \end{pmatrix}$$

mit $\hat{B}_{2,2} = \tilde{W}^H W_{n-k} \Sigma_{n-k}^{1/2}$.

Damit erhalten wir für $\hat{A} = BB^T$

$$\hat{A} = \begin{pmatrix} I_k & 0 \\ 0 & \hat{B}_{2,2} \end{pmatrix} \begin{pmatrix} I_k & 0 \\ 0 & \hat{B}_{2,2} \end{pmatrix}^T = \begin{pmatrix} I_k & 0 \\ 0 & \hat{B}_{2,2}\hat{B}_{2,2}^T \end{pmatrix}.$$

Die Singulärwerte von $\hat{A}$ bestehen also aus den Singulärwerten von I_k und den Singulärwerten von

$$\hat{B}_{2,2}\hat{B}_{2,2}^T = \tilde{W}^H(W_{n-k}\Sigma_{n-k}W_{n-k}^T)\overline{\tilde{W}}.$$

Es gilt

$$\sigma(W_{n-k}\Sigma_{n-k}W_{n-k}^T) = \{\sigma_{k+1}, \ldots, \sigma_n\}.$$

Die Matrix $W_{n-k}\Sigma_{n-k}W_{n-k}^T$ ist komplex symmetrisch.
Nach Satz 4.11 ist mit $X = \tilde{W}^H$, $XX^H = \tilde{W}^H\tilde{W} = I_{(n-k)}$ dann also der Faktor θ_k identisch 1, und damit

$$\sigma(\tilde{W}^H W_{n-k}\Sigma_{n-k}W_{n-k}^T\overline{\tilde{W}}) = \{\sigma_{k+1}, \ldots, \sigma_n\}.$$

□

Da diese Art von Präkonditionierung bewirkt, dass einige Singulärwerte „entfernt“ werden, indem sie auf den Wert 1 abgebildet werden, verwenden wir hierfür den Begriff Deflation.

Bemerkung 5.3. Die Eigenwerte von $S^{-1}S^{-H}$ sind gegeben durch

$$\begin{aligned}\text{spek}\left((S^HS)^{-1}\right) &= \text{spek}\left(\begin{pmatrix}\Sigma_k & 0\\ 0 & I_{n-k}\end{pmatrix}^{-1}\right)\\ &= \{1, \sigma_1^{-1}, \ldots, \sigma_k^{-1}\}.\end{aligned}$$

Nach Satz 4.11 liegen die Veränderungsfaktoren θ_k für die Singulärwerte $\sigma_k(A)$ bei T-Transformation mit X alle zwischen dem minimalen und maximalen Eigenwert von $XX^H = S^{-1}S^{-H} = (S^HS)^{-1}$. Eine Folgerung aus Satz 5.2 ist, dass bei Deflation die θ_k-Werte mit den Eigenwerten von XX^H übereinstimmen.

5.1 Anwendung

Bei der Anwendung der Deflation im CSYM-Verfahren nutzen wir, dass die Darstellung der Projektionsmatrix $P_{\tilde{W}}$, welche die orthogonale Projektion auf den Unterraum $\tilde{W}$ repräsentiert, eindeutig ist [3]. Es gilt nämlich:

$$P_{\tilde{W}} = \tilde{W}\tilde{W}^H = I_n - V_kV_k^H.$$

Damit erhalten wir für die Matrizen SS^H und $(SS^H)^{-1}$, die wir im implizit präkonditionierten CSYM-Verfahren (Algorithmus 4.2) benötigen:

$$\begin{aligned} SS^H &= \begin{pmatrix} V_k & \tilde{W} \end{pmatrix} \begin{pmatrix} \Sigma_k & 0 \\ 0 & I_{n-k} \end{pmatrix} \begin{pmatrix} V_k & \tilde{W} \end{pmatrix}^H \\ &= V_k \Sigma_k V_k^H + \tilde{W}\tilde{W}^H \\ &= V_k(\Sigma_k - I_k)V_k^H + I_n, \\ (SS^H)^{-1} &= \begin{pmatrix} V_k & \tilde{W} \end{pmatrix} \begin{pmatrix} \Sigma_k^{-1} & 0 \\ 0 & I_{n-k} \end{pmatrix} \begin{pmatrix} V_k & \tilde{W} \end{pmatrix}^H \\ &= V_k \Sigma_k^{-1} V_k^H + \tilde{W}\tilde{W}^H \\ &= V_k(\Sigma_k^{-1} - I_k)V_k^H + I_n. \end{aligned}$$

Eingesetzt in Algorithmus 4.2 erhalten wir:

Algorithmus 5.1 Das CSYM-Verfahren mit Deflation

wähle $x_0,\ r_0 = b - Ax_0$
$\tilde{r}_0 = r_0 + \mathbf{V_k}(\mathbf{\Sigma_k^{-1}} - \mathbf{I_k})\mathbf{V_k^H} r_0$
$q_0 = 0,\ \tau_1 = \sqrt{r_0^H \tilde{r}_0},\ q_1 = \overline{\tilde{r}_0}/\tau_1$
$z_1 = Aq_1,\ \alpha_1 = q_1^T z_1, \beta_1 = 0$
Initialisierung $\hat{c}_0,\ \hat{s}_0, \hat{c}_{-1},\ \hat{s}_{-1},\ \hat{p}_{-1},\ \hat{p}_0$
for $m = 1, 2 \ldots$ **do**
  $\tilde{z}_m = z + \mathbf{V_k}(\mathbf{\Sigma_k^{-1}} - \mathbf{I_k})\mathbf{V_k^H} z$
  $w = \tilde{z}_m - \alpha_m \overline{q_m} - \beta_m \overline{q_{m-1}}$
  $\beta_{m+1} = \sqrt{(w + \mathbf{V_k}(\mathbf{\Sigma_k} - \mathbf{I_k})\mathbf{V_k^H} w)^H w}$
  $q_{m+1} = \overline{w}/\beta_{m+1}$
  $z_{m+1} = Aq_{m+1}$
  $\alpha_{m+1} = q_{m+1}^T z_{m+1}$
  Berechnung $\hat{\eta}_m,\ \hat{\theta}_m,\ \hat{\gamma}_m,\ \hat{c}_m,\ \hat{s}_m,\ \hat{\xi}_m$
  $p_m = (q_m - \eta_m p_{m-1} - \theta_m p_{m-2})/\xi_m$
  $x_m = x_{m-1} + \tau_m c_m p_m$
  $\tau_{m+1} = -s_m \tau_m$
end for

Der größte zusätzliche Aufwand pro Iteration im Vergleich zum unpräkonditionierten Verfahren steckt in den Matrix-Vektormultiplikationen mit den Matrizen $V_k^H \in \mathbb{C}^{k \times n}$

und $V_k \in \mathbb{C}^{n\times k}$. Ein weiterer Vorteil von Algorithmus 5.1 ist, dass nur die Berechnung von V_k und Σ_k erforderlich ist. Da die Matrix $\tilde{W}$ nicht berechnet werden muss, nehmen wir für spätere theoretische Betrachtungen insbesondere $\tilde{W} = W_{n-k}$ mit W_{n-k} aus (5.4) an. Zusammen mit Satz 5.2 erhalten wir $\hat{A} = \mathrm{diag}(I_k, \Sigma_{n-k})$, d.h. $\hat{A}$ ist eine reelle Diagonalmatrix und insbesondere normal. Insbesondere folgt, da dies auch für die Wahl $k = 0$, d.h. $SS^H = I$, gilt:

Lemma 5.4. *Das Konvergenzverhalten des CSYM-Verfahrens zur Lösung von $Ax = b$ ist unabhängig von der Normalität der Matrix A.*

Voraussetzung ist, dass V_k orthogonal ist. In der Praxis muss dies nicht gegeben sein. In diesem Fall kann durch Anwendung der Sherman-Morrison-Woodbury-Formel [30] zur Inversion von SS^H eine Verbesserung erreicht werden.

Satz 5.5. *Sei $A \in \mathbb{C}^{n\times n}$ regulär und $U,\ V \in \mathbb{C}^{n\times k},\ k \leq n$. Dann ist $A + UV^H$ invertierbar genau dann, wenn $I_k + V^H A^{-1} U$ invertierbar ist, und es gilt*

$$(A + UV^H)^{-1} = A^{-1} - A^{-1}U(I_k + V^H A^{-1}U)^{-1}V^H A^{-1}.$$

Beweis: durch Nachrechnen. □

Wir haben gezeigt

$$SS^H = I_n + V_k(\Sigma_k - I_k)V_k^H.$$

Wenden wir mit $A = I$, $U = V_k$ und $V = V_k(\Sigma_k - I_k)$ die Sherman-Morrison-Woodbury-Formel an, so erhalten wir

$$(SS^H)^{-1} = I_n - V_k(I_k + (\Sigma_k - I_k)V_k^H V_k)^{-1}(\Sigma_k - I_k)V_k^H.$$

5.2 Wahl der Startvektoren

Wir betrachten im Folgenden zwei spezielle Startvektoren x_0. Der Startvektor $\mathbf{x_p}$ als Projektion der rechten Seite b auf den von k Singulärvektoren aufgespannten Raum sei gegeben durch

$$\mathbf{x_p} = \overline{V}_k \Sigma_k^{-1} V_k^H b.$$

Der andere Startvektor sei $\mathbf{x_0} = 0$. Wie wir mit folgendem Satz zeigen werden, hängt die Konvergenz des CSYM-Verfahrens mit Deflation stark von der Wahl des Startvektors x_0 ab.

Satz 5.6. *Sei* $\mathbf{r_m}$ *das Residuum des präkonditionierten Systems zu dem Startvektor* $\mathbf{x_0}$ *und* $\mathbf{r_{m_p}}$ *das Residuum des präkonditionierten Systems zum Startvektor* $\mathbf{x_p}$ *im* m*-ten Iterationsschritt des CSYM-Verfahrens mit Deflation. Dann gilt*

$$\|\mathbf{r_{m_p}}\|_2 \quad \leq \|\mathbf{r_m}\|_2.$$

Im Folgenden seien V, V_k, W_{n-k} und Σ, Σ_k, Σ_{n-k} wie in dem Beweis von Satz 5.2 in (5.2) bzw. (5.4) definiert.

Alle weiteren analog partitionierten Teilvektoren eines Vektors $\hat{\alpha} \in \mathbb{C}^n$ schreiben wir als

$$\hat{\alpha} = \begin{pmatrix} \hat{\alpha}_1 \\ \hat{\alpha}_2 \end{pmatrix} \in \mathbb{C}^n.$$

Für den Beweis von Satz 5.6 benutzen wir die folgenden zwei Lemmata.

Lemma 5.7. *Seien* $x_0 = \overline{V}\hat{\alpha}$ *und* $b = V\hat{\beta}$*. Dann ist der Unterraum des CSYM-Verfahrens bezüglich des unpräkonditionierten Systems für* $m = 2l+1$ *gegeben durch*

$$\tilde{K}_{2l+1}(A, r_0) = span\left(\overline{V\hat{\gamma}},\ \overline{V}\Sigma\hat{\gamma},\ \ldots,\ \overline{V}\Sigma^{2l-1}\hat{\gamma},\ \overline{V}\Sigma^{2l}\overline{\hat{\gamma}}\right)$$
$$mit\ \hat{\gamma} = \hat{\beta} - \Sigma\hat{\alpha}.$$

Beweis: Der Unterraum $\tilde{K}_{2l+1}$ ist gegeben durch

$$\tilde{K}_{2l+1}(A, r_0) = \text{span}\left(\overline{r_0},\ \overline{A}r_0\ \ldots,\ (\overline{A}A)^{l-1}\overline{A}r_0,\ (\overline{A}A)^l\overline{r}_0\right).$$

Es ist $r_0 = b - Ax_0 = V\hat{\beta} - A\overline{V}\hat{\alpha}$. Mit $A\overline{V} = V\Sigma$ erhalten wir weiter

$$r_0 = V(\hat{\beta} - \Sigma\hat{\alpha}) = V\hat{\gamma}.$$

Wir beweisen die Darstellung mit Induktion über l:
Es ist $\overline{r}_0 = \overline{V\hat{\gamma}}$ und $\overline{A}r_0 = \overline{A}V\hat{\gamma} = \overline{V}\Sigma\hat{\gamma}$.
Sei l beliebig aber fest. Laut Induktionsvoraussetzung gilt

$$(\overline{A}A)^{l-1}\overline{r_0} = \overline{V}\Sigma^{2l-2}\overline{\hat{\gamma}}.$$

Somit erhalten wir

$$\begin{aligned}(\overline{A}A)^l\overline{r_0} &= (\overline{A}A)(\overline{V}\Sigma^{2l-2})\overline{\hat{\gamma}} \\ &= \overline{A}V\Sigma^{2l-1}\overline{\hat{\gamma}} \\ &= \overline{V}\Sigma^{2l}\overline{\hat{\gamma}}.\end{aligned}$$

Mit der Darstellung

$$(\overline{A}A)^{l-1}\overline{A}r_0 = \overline{V}\Sigma^{2l-1}\hat{\gamma}$$

erhalten wir analog

$$(\overline{A}A)^{l}\overline{A}r_0 = \overline{V}\Sigma^{2l+1}\hat{\gamma}.$$

Damit ist die Darstellung für alle l bewiesen. □

Aus Lemma 5.7 folgt

Lemma 5.8. *Im unpräkonditionierten CSYM-Verfahren gilt mit den Bezeichnungen aus Lemma 5.7 und* $\hat{\beta} = \begin{pmatrix}\hat{\beta}_1\\ \hat{\beta}_2\end{pmatrix}$*:*

a) für den Startvektor $\mathbf{x_p}$ *:* $\hat{\gamma} = \begin{pmatrix}0\\ \hat{\beta}_2\end{pmatrix}$, $\hat{\beta}_2 \in \mathbb{C}^{n-k}$,

b) für den Startvektor $\mathbf{x_0}$ *:* $\hat{\gamma} = \hat{\beta}$.

Beweis: a) Es gilt

$$\begin{aligned} A\mathbf{x_p} &= \begin{pmatrix}V_k & V_{n-k}\end{pmatrix}\begin{pmatrix}\Sigma_k & 0\\ 0 & \Sigma_{n-k}\end{pmatrix}\begin{pmatrix}V_k^T\\ V_{n-k}^T\end{pmatrix}\overline{V}_k\Sigma_k^{-1}V_k^H\begin{pmatrix}V_k & V_{n-k}\end{pmatrix}\hat{\beta}\\ &= \begin{pmatrix}V_k & V_{n-k}\end{pmatrix}\begin{pmatrix}\Sigma_k & 0\\ 0 & \Sigma_{n-k}\end{pmatrix}\begin{pmatrix}I_k\\ 0_{n-k}\end{pmatrix}\Sigma_k^{-1}\begin{pmatrix}I_k & 0_{n-k}\end{pmatrix}\hat{\beta}\\ &= V_k\hat{\beta}_1 \end{aligned}$$

und damit

$$r_0 = b - A\mathbf{x_p} = \begin{pmatrix}V_k & V_{n-k}\end{pmatrix}\begin{pmatrix}\hat{\beta}_1\\ \hat{\beta}_2\end{pmatrix} - V_k\hat{\beta}_1 = V_{n-k}\hat{\beta}_2.$$

b) Da $\mathbf{x_0} = 0$ ist, gilt $\hat{\alpha} = 0$. □

Wir betrachten nun das CSYM-Verfahren mit Deflation mit expliziter Präkonditionierung und setzen insbesondere $\tilde{W} = W_{n-k}$.

Lemma 5.9. *Sei $m = 2l+1$ (der Fall $m = 2l$ geht analog). Der Unterraum des CSYM-Verfahrens bezüglich der Matrix $\hat{A}$ des explizit präkonditionierten Systems mit $\tilde{W} = W_{n-k}$ bei Startvektor $\mathbf{x_0}$ ist gegeben durch*

$$\tilde{\hat{K}}_{2l+1}(\hat{A}, \hat{r}_0) = span\left(\overline{\hat{\delta}(1)},\ \hat{\delta}(2), \ldots,\ \hat{\delta}(2l),\ \overline{\hat{\delta}(2l+1)}\right)$$
$$mit\ \hat{\delta}(j) = \begin{pmatrix} \Sigma_k^{-1/2}\hat{\beta}_1 \\ \Sigma_{n-k}^{j-1}\hat{\beta}_2 \end{pmatrix}.$$

Beweis: Der Zusammenhang zwischen r_0, dem Startresiduum des ursprünglichen Systems, und dem des präkonditionierten Systems ist gegeben durch $\hat{r}_0 = S^{-1}r_0$ mit S lt. (5.1). Daher gilt

$$\begin{aligned}\overline{\hat{r}}_0 &= \overline{S}^{-1}\overline{r}_0 = \overline{S}^{-1}\overline{V\hat{\beta}} \\ &= \begin{pmatrix} \Sigma_k^{-1/2} & 0 \\ 0 & I_{n-k} \end{pmatrix} \begin{pmatrix} V_k^T \\ W_{n-k}^T \end{pmatrix} \overline{V\hat{\beta}} = \begin{pmatrix} \Sigma_k^{-1/2}\overline{\hat{\beta}}_1 \\ \overline{\hat{\beta}}_2 \end{pmatrix}.\end{aligned}$$

Mit dem Startvektor $\mathbf{x_0}$ und mit $V = I_n$, den Singulärvektoren der Matrix $\hat{A}$ und $\Sigma_k = I_k$ wenden wir nun Lemma 5.7 an. □

Folgerung aus Lemma 5.9
Für $x_0 = \mathbf{x_p}$ gilt die obige Darstellung des Unterraumes mit

$$\hat{\delta}(j) = \begin{pmatrix} 0 \\ \Sigma_{n-k}^{j-1}\hat{\beta}_2 \end{pmatrix}.$$

Zurück zum Beweis von Satz 5.6
Nach Lemma 3.35 lässt sich jedes Residuum r_m mit Hilfe von Polynomen $p_1(\cdot)$ und $p_2(\cdot)$ schreiben als

$$r_m = \dot{r}_m(r_0) = r_0 - p_1(A\overline{A})A\overline{r_0} - p_2(A\overline{A})A\overline{A}r_0. \tag{5.5}$$

Das CSYM-Verfahren konstruiert die Polynome $p_1(\cdot)$ und $p_2(\cdot)$, so dass $\|\dot{r}_m(r_0)\|_2$ minimiert wird.

Sei der Vektor $\mathbf{r_0}$ nun speziell das Anfangsresiduum und $\mathbf{r_m}$ das Residuum in dem m-ten Iterationsschritt zu dem Startvektor $\mathbf{x_0}$.
Dann gilt für $\mathbf{r_m}$ mit speziellen Polynomen $p_1(\cdot)$ und $p_2(\cdot)$:

$$\mathbf{r_m} = \dot{\mathbf{r}}_\mathbf{m}(\mathbf{r_0}) = \mathbf{r_0} - p_1(A\overline{A})A\overline{\mathbf{r_0}} - p_2(A\overline{A})A\overline{A}\mathbf{r_0}. \tag{5.6}$$

Nun betrachten wir das präkonditionierte System.
Zusammen mit der Struktur von $\hat{A}$ (Satz 5.2 und Folgerung) und der Darstellung der Vektoren $\mathbf{r_0}$, $A\overline{\mathbf{r}_0}$, $A\overline{A}\mathbf{r_0}$ aus $\tilde{\hat{K}}_{2l+1}(\hat{A}, \hat{r}_0)$ (Lemma 5.9) erhalten wir nach m Schritten des CSYM-Verfahrens einen Vektor

$$\begin{aligned}\mathbf{r_m} &= \begin{pmatrix} \Sigma_k^{-1/2}(\hat{\beta}_1 - p_2(I_k)\hat{\beta}_1 - p_1(I_k)\overline{\hat{\beta}}_1) \\ \hat{\beta}_2 - p_2(\Sigma_{n-k}^2)\Sigma_{n-k}^2\hat{\beta}_2 - p_1(\Sigma_{n-k}^2)\Sigma_{n-k}\overline{\hat{\beta}}_2 \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{r_{m_1}} \\ \mathbf{r_{m_2}} \end{pmatrix} \text{ mit } \mathbf{r_{m_1}} \in \mathbb{C}^k,\ \mathbf{r_{m_2}} \in \mathbb{C}^{n-k}.\end{aligned} \tag{5.7}$$

Der Vektor $\mathbf{r_{m_p}}$ mit $\mathbf{x_p}$ als Startvektor und dem Anfangsresiduum $\mathbf{r_p}$ hat die Struktur:

$$\begin{aligned}\mathbf{r_{m_p}} &= \begin{pmatrix} 0 \\ \hat{\beta}_2 - \tilde{p}_2(\Sigma_{n-k}^2)\Sigma_{n-k}^2\hat{\beta}_2 - \tilde{p}_1(\Sigma_{n-k}^2)\Sigma_{n-k}\overline{\hat{\beta}}_2 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ \mathbf{r_{m_{p_2}}} \end{pmatrix} \text{ mit } \mathbf{r_{m_{p_2}}} \in \mathbb{C}^{n-k}.\end{aligned}$$

Die Darstellung von $\dot{\mathbf{r}}_{\mathbf{m_p}}(\cdot)$ enthält neben anderen Vektoren i.A. auch andere Minimierungspolynome $\tilde{p}_1(\cdot)$ und $\tilde{p}_2(\cdot)$ als die in der Darstellung von $\dot{\mathbf{r}}_{\mathbf{m}}(\cdot)$ in (5.6). Wir erhalten damit

$$\begin{aligned}\|\mathbf{r_m}\|_2^2 &= \|\mathbf{r_{m_1}}\|_2^2 + \|\mathbf{r_{m_2}}\|_2^2 \\ &\geq \|\mathbf{r_{m_2}}\|_2^2 \\ &\geq \|\mathbf{r_{m_{p_2}}}\|_2^2 \\ &= \|\mathbf{r_{m_p}}\|_2^2.\end{aligned}$$

Die letzte Ungleichung folgt, da auch bei Startvektor $\mathbf{x_p}$ das minimale Residuum ermittelt wird. □

In exakter Arithmetik erwarten wir lediglich zwei Iterationen mehr, wenn wir mit dem Nullvektor statt mit dem projizierten Vektor starten, da 1 mehrfacher Singulärwert ist. Brechen wir das Verfahren aber vorher ab, so werden sich die Normen der Residuen unterscheiden. In finiter Arithmetik, wo nur ein genähertes $\tilde{I}_k \approx I_k$ erreicht wird, erwarten wir eine Verstärkung dieses Unterschieds im Konvergenzverhalten.

Bemerkung 5.10. Es gelten folgende Äquivalenzen auf Grund der gleichen Startresiduen und somit gleichen Unterräume beim CSYM-Verfahren:

- Das Lösen des Gleichungssystems $Ax = b$ mit dem Startvektor $\mathbf{x_p}$ entspricht dem Lösen des Gleichungssystems $A\tilde{x} = b - A\mathbf{x_p}$ mit dem Startvektor $\tilde{x}_0 = 0$.
- Das Lösen des Gleichungssystems $Ax = b$ mit dem Startvektor $\mathbf{x_0}$ entspricht dem Lösen des Gleichungssystems $A\tilde{x} = b - A\mathbf{x_p}$ mit dem Startvektor $\tilde{x}_0 = -\mathbf{x_p}$.

5.3 Berechnung des Faktors

Bei der Einführung des CSYM-Verfahrens wurde bereits erwähnt, dass die durchgeführte partielle Tridiagonalisierung auch ein Verfahren für die Approximation der Singulärwerte von A liefert. Daher bietet sich das folgende Vorgehen an, insbesondere wenn wir Gleichungssysteme mit der gleichen Matrix A und mehreren rechten Seiten lösen wollen.

- m Iterationsschritte des CSYM-Verfahrens für $Ax = b$, im Folgenden CSYM(m), liefern
  - eine Matrix $Q_m \in \mathbb{C}^{n\times m}$ mit orthonormalen Spalten
  - eine Tridiagonalmatrix $T_m \in \mathbb{C}^{m\times m}$, wobei $T_m = Q_m^T A Q_m$ gilt.
- Berechnung der SSVD $T_m = U_m \Sigma_m U_m^T$
- Berechnung von approximativen Singulärvektoren von A durch

$$V_m = \overline{Q}_m U_m \in \mathbb{C}^{n\times m}$$

- Übernahme der k „besten“ Singulärvektoren aus V_m in V_k und der entsprechenden k Singulärwerte aus Σ_m in Σ_k

5.4 Numerische Ergebnisse

Alle numerischen Ergebnisse wurden mit MATLAB Version 7.1.0.183 (R14) Service Pack 3 unter Linux SuSE 10.0 auf einem AMD 64 X2 4200+ (Dual Core) Prozessor mit einem Arbeitsspeicher von 4 GByte ermittelt.

Wir werden das CSYM-Verfahren mit Deflation zur Lösung folgender Gleichungssysteme durchführen:

1.) Eine komplex symmetrische Matrix **A1** der Größe $n = 1176$, voll besetzt, aus der Diskretisierung über Oberflächenintegrale [41] von 49 Multibeam-Antennen. Die rechte Seite b ist der Nullvektor bis auf einen Eintrag.

2.) Die dicht besetzte Matrix **C2** von CERFACS [6] ($n = 1299$) aus der Diskretisierung nach der BEM (Boundary Element Method) aus dem Gebiet der Elektrodynamik.

3.) Die dünn besetzte Matrix *youngc*2 (**Y2**) aus der Harwell-Boeing Sammlung [10] (Akustische Wellenausbreitung, Diskretisierung der Helmholtz-Gleichung) mit $n = 841$.

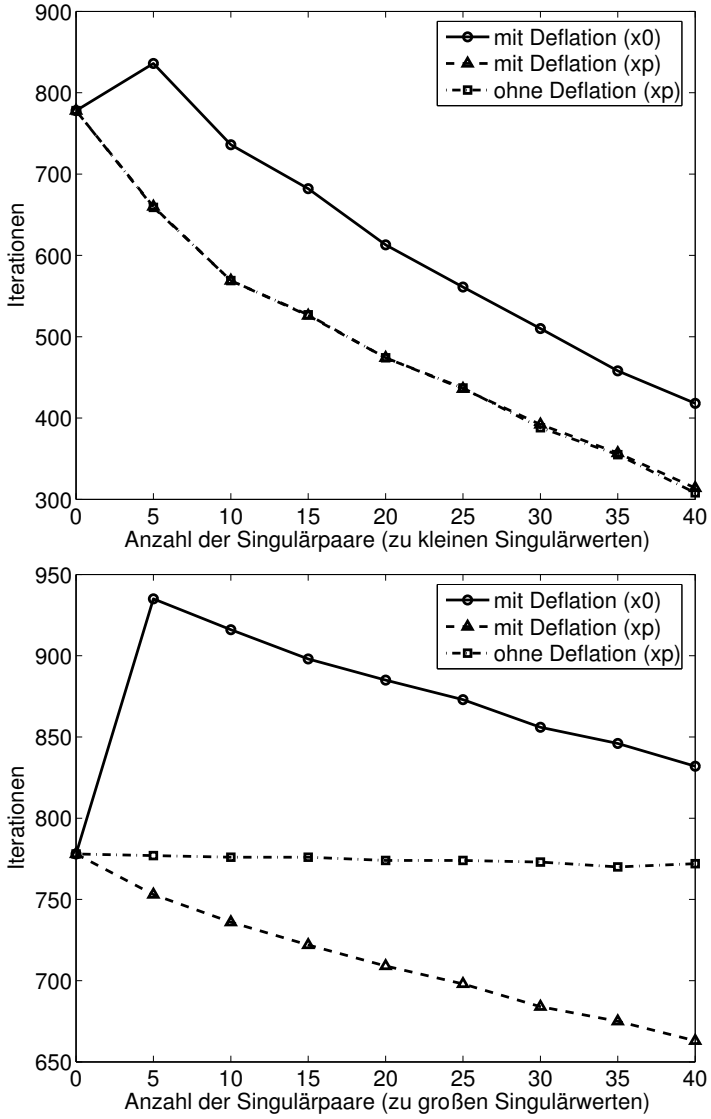
Als rechte Seiten zu 2.) und 3.) wurde der Ergebnisvektor aus dem Matrix-Vektorprodukt der jeweiligen Matrix mit einem zufällig erzeugten Vektor gewählt. Dieses sind auch die Gleichungssysteme, die wir u. a. auch in Kapitel 6 verwenden werden.
Das CSYM-Verfahren wird für **C2** und **A1** beendet, sobald die Norm des relativen Residuums des präkonditionierten Systems kleiner als 10^{-9} ist oder 1000 Iterationen durchgeführt wurden, für **Y2** analog für 10^{-8} und 800 Iterationen.

Zunächst werden wir das CSYM-Verfahren mit Deflation mittels Singulärpaaren testen, die wir mit Hilfe der MATLAB Funktion *svd.m* und anschließender symmetrischer Faktorisierung erhalten. Dabei untersuchen wir Singulärpaare zu kleinsten und größten Singulärwerten und das CSYM-Verfahren mit und ohne Deflation für die Startvektoren $\mathbf{x_p}$ und $\mathbf{x_0}$.

Für das Multibeam-Antennenproblem werden wir das CSYM-Verfahren mit Deflation bzw. projiziertem Startvektor für Singulärpaare untersuchen, die, wie in Kapitel 5.3 dargestellt, über das CSYM-Verfahren ermittelt wurden. Dabei wählen wir zwei verschiedene rechte Seiten, eine für die ersten Schritte des unpräkonditionierten CSYM-Verfahrens zur Ermittlung der Singulärpaare und eine andere rechte Seite für das Gleichungssystem, das wir mit Deflation lösen wollen.

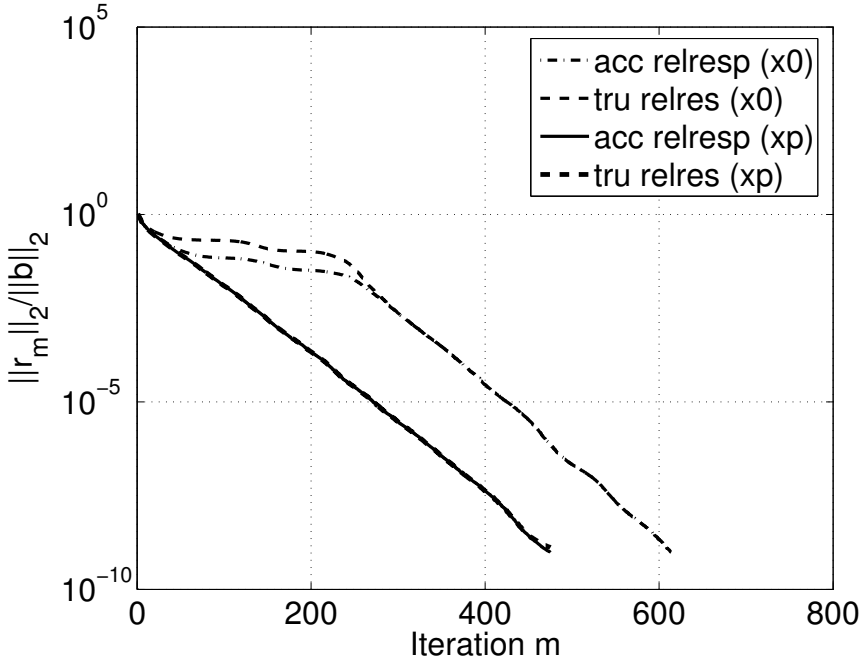
Deflation mit vorberechneten Singulärpaaren

Die folgenden Abbildungen zeigen für das Multibeam-Antennenproblem die Ergebnisse des CSYM-Verfahrens mit und ohne Deflation (bei bis zu 40 Singulärpaaren zu kleinen bzw. großen Singulärwerten) mit unterschiedlichen Startvektoren.



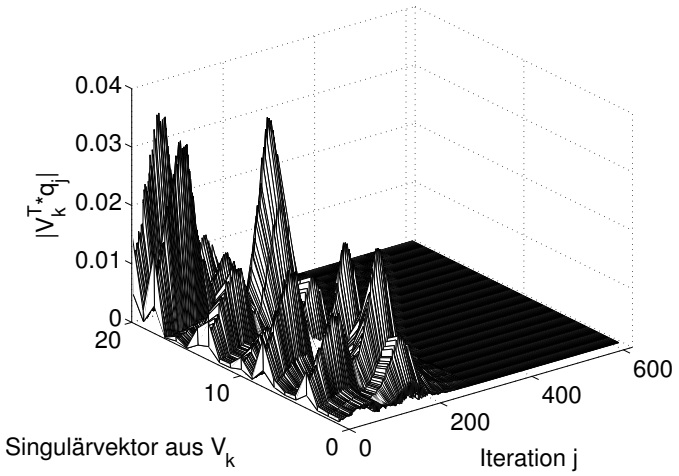
Das CSYM-Verfahren mit Deflation und Startvektor $\mathbf{x_0}$ liefert die schlechtesten Ergebnisse. Für Singulärpaare zu großen Singulärwerten ist es sogar schlechter als das unpräkonditionierte Verfahren mit Startvektor $\mathbf{x_0}$ (778 Iterationen, $k = 0$). Mit Startvektor $\mathbf{x_p}$ dagegen liefert das CSYM-Verfahren mit Deflation immer eine Verbesserung gegenüber dem unpräkonditionierten CSYM-Verfahren, die mit der Zahl der berücksichtigten Singulärpaare zunimmt, am deutlichsten bei Singulärpaaren zu kleinsten Singulärwerten. Mit Startvektor $\mathbf{x_p}$ basierend auf Singulärpaaren zu kleinsten Singulärwerten spielt es keine Rolle, ob wir das CSYM-Verfahren mit oder ohne Deflation durchführen. Das unpräkonditionierte CSYM-Verfahren mit Startvektor $\mathbf{x_p}$ aus Singulärpaaren zu größten Singulärwerten bringt nahezu keinen Vorteil gegenüber dem mit Startvektor $\mathbf{x_0}$.

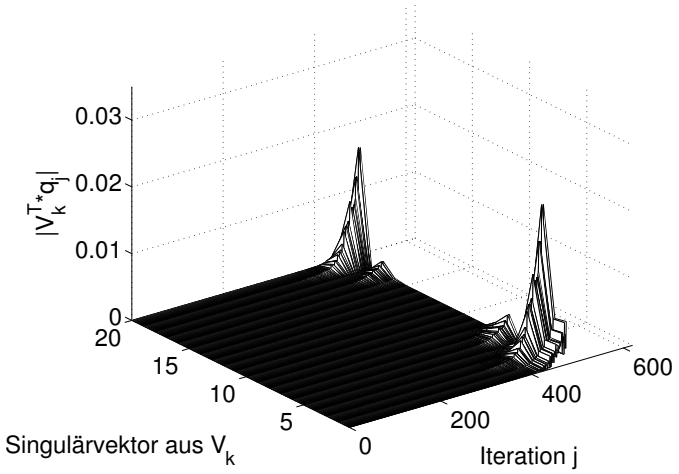
Den Fall mit 20 Singulärpaaren zu kleinsten Singulärwerten untersuchen wir exemplarisch genauer. Das Konvergenzverhalten des CSYM-Verfahrens mit Deflation zeigt die folgende Abbildung, in der jeweils die Norm des akkumulierten relativen präkonditionierten Residuums (acc relresp) und des berechneten (tru relres) bezüglich der rechten Seite b angegeben ist.



Auffallend ist die Lücke zwischen akkumuliertem und berechnetem Residuum für Startvektor $\mathbf{x_0}$.

Die Größe $|v_k^T q_j|$, d.h. die Komponente des jeweiligen Singulärvektors v_k in dem neuen Basisvektor in dem j-ten Iterationsschritt der CSYM-Verfahren, zeigen die folgenden Abbildungen für das CSYM-Verfahren mit Deflation für Startvektor $\mathbf{x_0}$ (oben) bzw. $\mathbf{x_p}$ (unten).

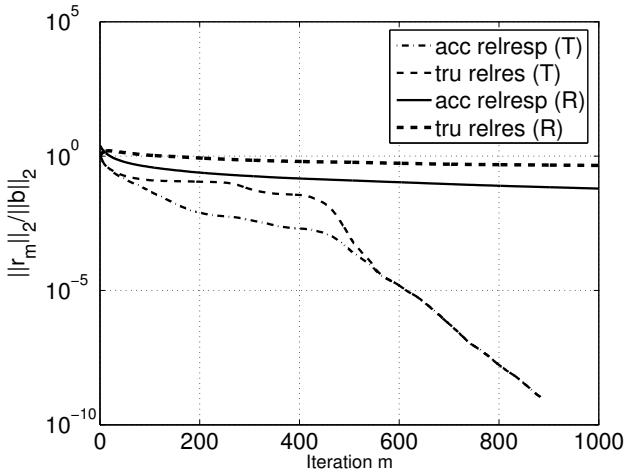




Die Lücken zwischen der Norm des akkumulierten und des berechneten Residuums fallen mit dem Auftreten von starken Komponenten der entfernten Vektoren in der neuen Basis zusammen.

Für das CSYM-Verfahren mit Deflation mit den k größten Singulärwerten erhalten wir ähnliche Bilder. Also ist bei dem CSYM-Verfahren mit Deflation mit Startvektor $\mathbf{x_0}$ der Teilvektor $\mathbf{r_{m_1}} \in \mathbb{C}^k$ in (5.7), aus dem Beweis zu Satz 5.6, der Grund für die langsamere Reduktion der Norm des Residuums. Seine Größe hängt von dem Anteil der Singulärvektoren aus V_k in b ab und dem Faktor $\Sigma_k^{-1/2}$. Daher wird er durch kleine Singulärwerte vergrößert, und wird so schneller reduziert, und durch große verkleinert.
Beim unpräkonditionierten CSYM-Verfahren mit Startvektor $\mathbf{x_p}$ traten starke Anteile der Singulärvektoren aus V_k zu großen Singulärwerten schnell in der Basis auf. Dies war für Singulärvektoren aus V_k zu kleinen Singulärwerten nicht der Fall, sie verhielten sich auch in der Praxis analog Lemma 5.8.

Zum Vergleich testen wir den Präkonditionierer bestehend aus 20 Singulärpaaren zu großen Singulärwerten für eine weitere Faktorisierung.
Die folgende Abbildung zeigt das Konvergenzverhalten des CSYM-Verfahrens mit Präkonditionierung mit dem Takagi-Faktor aus (5.1) (dies entspricht dem CSYM-Verfahren mit Deflation) und der Präkonditionierung basierend auf einer Block-Cholesky-Zerlegung mit Rook-Pivotisierung. In beiden Fällen war der Startvektor $\mathbf{x_0}$.



Das CSYM-Verfahren mit Block-R-Cholesky-Präkonditionierung basierend auf dem gleichen unfaktorisierten Präkonditionierer wie die Deflation stagniert.

Das CSYM-Verfahren mit Deflation liefert auch für die getesteten Gleichungssysteme mit **Y2** und **C2** eine Verbesserung, vor allem für Singulärpaare zu kleinen Singulärwerten (Ergebnisse in Tabellenform, s. Anhang).

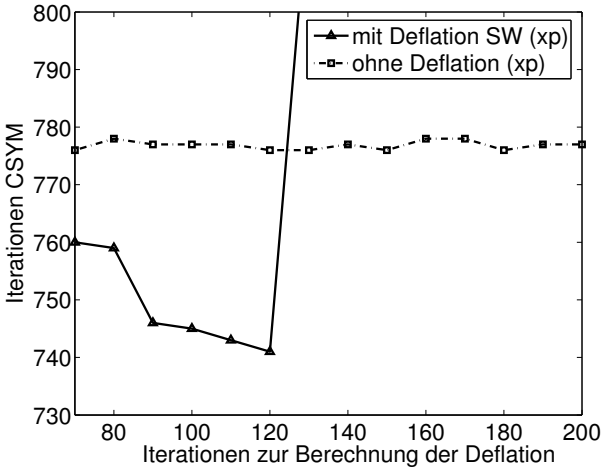
Deflation mit Singulärpaaren aus CSYM

Der Takagi-Faktor wurde nach m Iterationsschritten des CSYM-Verfahrens mit rechter Seite b ermittelt. Als Auswahlkriterium für gute Singulärpaare wählen wir das absolute bzw. relative Qualitätskriterium [38], d.h. wir übernehmen den Singulärvektor u_l und den Singulärwert σ_l der projizierten Matrix T_m, und somit $v_l = \overline{Q}_m u_l$, wenn für seine m-te Komponente $u_{l,m}$ gilt

$$\beta_{m+1}|u_{l,m}| \leq 10^{-5} \text{ bzw. } \frac{\beta_{m+1}|u_{l,m}|}{\sigma_l} \leq 10^{-5}.$$

Die besseren Ergebnisse erhalten wir bei Auswahl der Singulärpaare nach dem absoluten Qualitätskriterium. Anschließend lösen wir das Gleichungssystem mit der gleichen rechten Seite b und führen die Deflation im CSYM-Verfahren unter Nutzung der Sherman-Morrison-Woodbury-Formel (im Folgenden SW) durch.

Abbildung 5.1: Anzahl der Iterationen des CSYM-Verfahrens für Startvektor $\mathbf{x_p}$ nach m Schritten des CSYM-Verfahrens ohne Reorthogonalisierung



Erst nach 70 Iterationsschritten des CSYM-Verfahrens konnten erste Singulärpaare ermittelt werden. Im CSYM-Verfahren mit Deflation wird die Zahl der Iterationen zunächst reduziert, bei Berechnung des Takagi-Faktors nach ca. 120 Iterationen des CSYM-Verfahrens gehen die Iterationen über die des unpräkonditionierten CSYM-Verfahrens hinaus. Das CSYM-Verfahren ohne Deflation mit projiziertem Startvektor verhält sich wie bei vorberechneten Singulärpaaren, liefert also kaum eine Verbesserung.

Die Verschlechterung beim CSYM-Verfahren mit Deflation ist auf den Orthogonalitätsverlust der Singulärvektoren zurückzuführen, so dass $\mathbf{x_p}$ nicht mehr sauber projiziert wird und bald starke Anteile der Singulärvektoren aus V_k in der Basis auftreten.
Auf Grund des Orthogonalitätsverlustes der Spalten Q_m (im Folgenden schreiben wir für die Vektoren aus dem Lanczos-ähnlichen Verfahren kurz Lanczos-Vektoren) verwenden wir zusätzlich Reorthogonalisierungsstrategien [32] zur Verbesserung des CSYM-Verfahrens zur Ermittlung der Singulärpaare:

a) *Complete Reorthogonalization (CR):*
Reorthogonalisierung des neuen Lanczos-Vektors gegen alle vorigen Lanczos-Vektoren, wenn die Abweichung von der Orthogonalität eine feste Schranke überschreitet.

b) *Partial Reorthogonalization (PR):*
Der neue Lanczos-Vektor wird nur gegen die vorigen Lanczos-Vektoren reorthogonalisiert, bei denen der Orthogonalitätsverlust eine Schranke überschreitet.

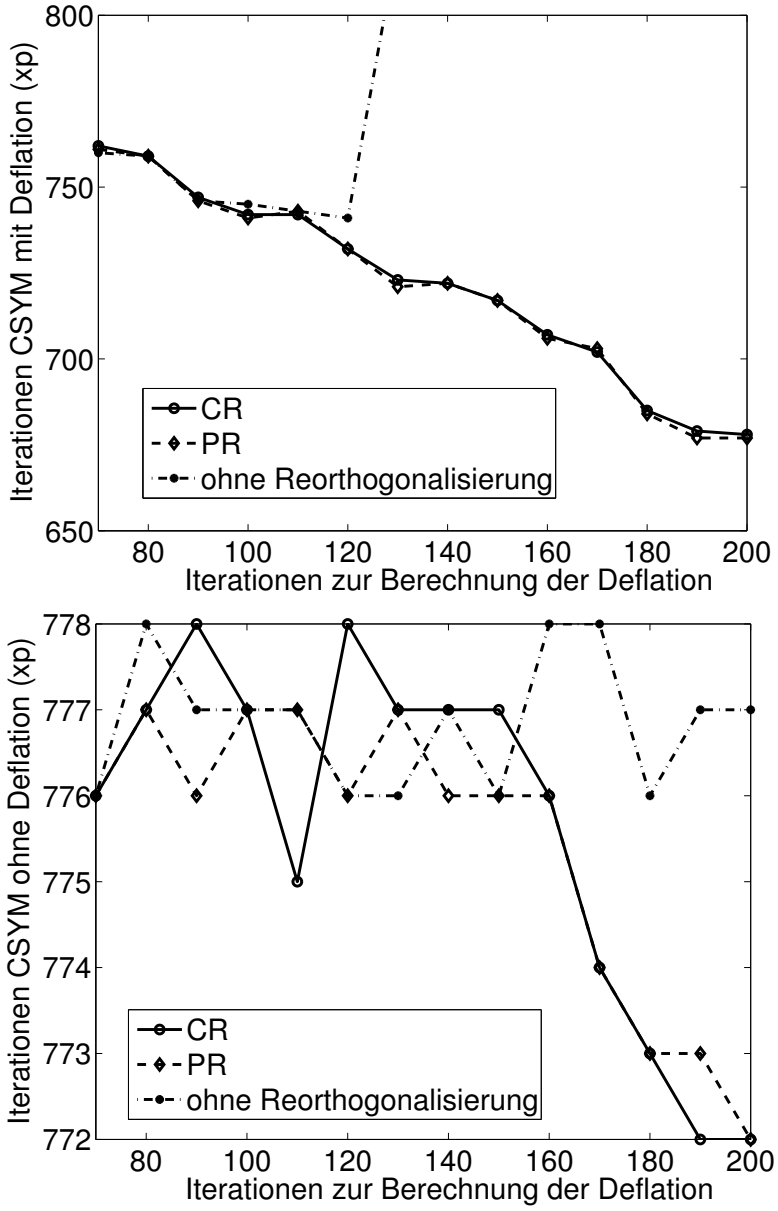
Bei beiden Strategien können die Orthogonalitätsmaße sukzessive aus den vorigen Werten bestimmt werden [38]. Wir haben sie aber immer direkt berechnet, da die iterative Berechnung schnell numerisch instabil wurde. Als Schranke wählen wir im Folgenden $\sqrt{\varepsilon} \approx 1.5 \cdot 10^{-8}$ und speziell als Kriterium für das Durchführen einer Reorthogonalisierung für

$$\begin{aligned} \text{CR:} \quad & \|Q_m^H Q_m - I_m\|_2 > \sqrt{\varepsilon}, \\ \text{PR:} \quad & \|Q_{m-1} q_m\|_\infty > \sqrt{\varepsilon}. \end{aligned}$$

Bemerkung 5.11. Das Verfahren *cssvdn* von Guo und Qiao baut auf demselben Tridiagonalisierungsverfahren wie das CSYM-Verfahren auf. Es wurde speziell für die Berechnung einer SSVD analog [4] entwickelt. Zur Beibehaltung der Orthogonalität der Lanczos-Vektoren im Lanczos-ähnlichen Verfahren wird eine von zwei Methoden angewendet, *LanPO.m* [17] (Lanczos Partial Orthogonalization) oder die modifizierte Variante hiervon, *LanMPO.m* [36]. In beiden Verfahren wird das Orthogonalitätsmaß sukzessive aufdatiert. Die erste Methode entspricht eher einer Mischung aus CR- und PR-, die zweite einer PR-Strategie. Tests mit diesen beiden Reorthogonalisierungsstrategien führten bei geringerem Aufwand ($\mathcal{O}(m)$ statt $\mathcal{O}(nm)$ bei PR) für die Ermittlung des Orthogonalitätsmaßes zu vergleichbaren Ergebnissen im CSYM-Verfahren mit Deflation wie mit CR/PR.

Die folgenden Abbildungen zeigen die Ergebnisse der CSYM-Verfahren ohne und mit Reorthogonalisierung CR bzw. PR im Vergleich.

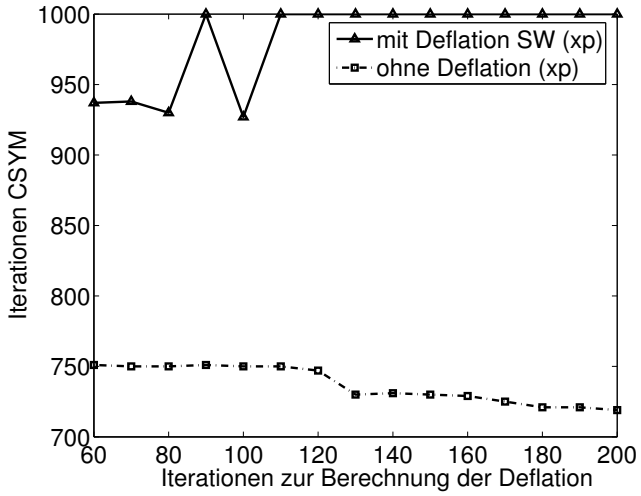
Abbildung 5.2: Anzahl der Iterationen des CSYM-Verfahrens für Startvektor $\mathbf{x_p}$ nach m Schritten des CSYM-Verfahrens mit Deflation (oben) und ohne



Das CSYM-Verfahren mit Deflation aus berechneten Werten des CSYM-Verfahrens mit Reorthogonalisierung ist auch nach mehr als 120 Schritten stabil. Ob wir die CR- oder PR-Strategie benutzen, hat hier keine Auswirkungen. Die Iterationszahlen der unpräkonditionierten CSYM-Verfahren mit projizierten Startvektoren schwanken zwischen 778 und 772 Iterationen, auch hier führt eine Reorthogonalisierung im CSYM-Verfahren zu einer Reduktion der Iterationszahlen nach 160 Schritten, wenn auch in geringerem Maße.

Es lohnt sich ja nur dann, das CSYM-Verfahren mit Deflation anzuwenden, wenn das Gleichungssystem mit verschiedenen rechten Seiten gelöst werden soll. Deshalb wählen wir in dem nun folgenden numerischen Experiment als erste rechte Seite einen Zufallsvektor und als zweite den bisherigen Vektor. Zur Auswahl der Singulärpaare wählen wir das relative Qualitätskriterium, das zu besseren Ergebnissen führt als das absolute.

Abbildung 5.3: Anzahl der Iterationen des CSYM-Verfahrens für Startvektor $\mathbf{x_p}$ mit und ohne Deflation nach m Schritten des CSYM-Verfahrens ohne Reorthogonalisierung

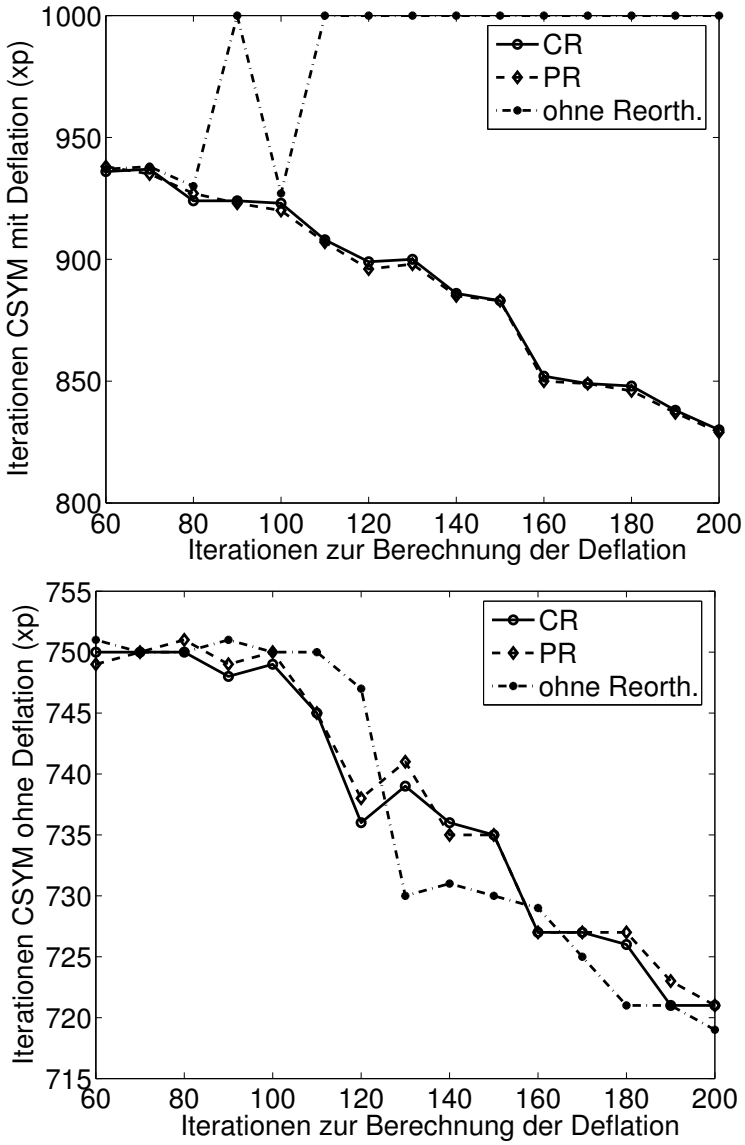


Das Konvergenzverhalten des präkonditionierten ist schlechter als das des unpräkonditionierten CSYM-Verfahrens. Nur das unpräkonditionierte CSYM-Verfahren mit projiziertem Startvektor liefert eine Verbesserung gegenüber dem mit Startvektor $\mathbf{x_0}$.

Bemerkung 5.12. Ohne Anwendung der Sherman-Morrison-Woodbury-Formel im CSYM-Verfahren mit Deflation erhalten wir noch höhere Iterationszahlen.

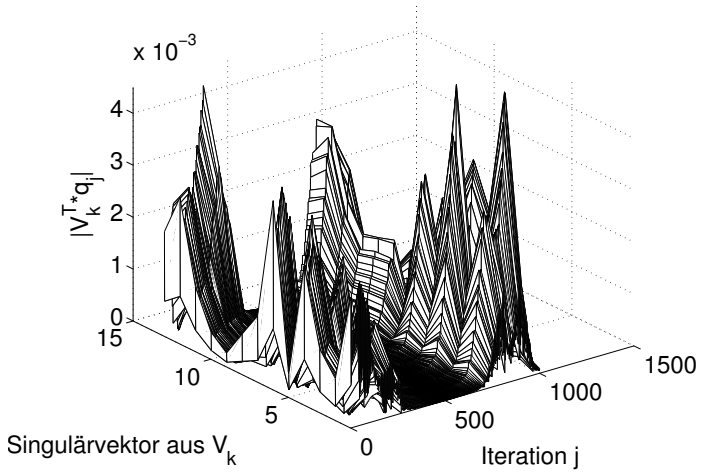
Die folgenden Abbildungen zeigen die Ergebnisse der CSYM-Verfahren (mit Singulärpaaren aus dem CSYM-Verfahren ohne und mit Reorthogonalisierung) im Vergleich.

Abbildung 5.4: Anzahl der Iterationen des CSYM-Verfahrens für Startvektor $\mathbf{x_p}$ nach m Schritten des CSYM-Verfahrens mit Deflation (oben) und ohne (unten)



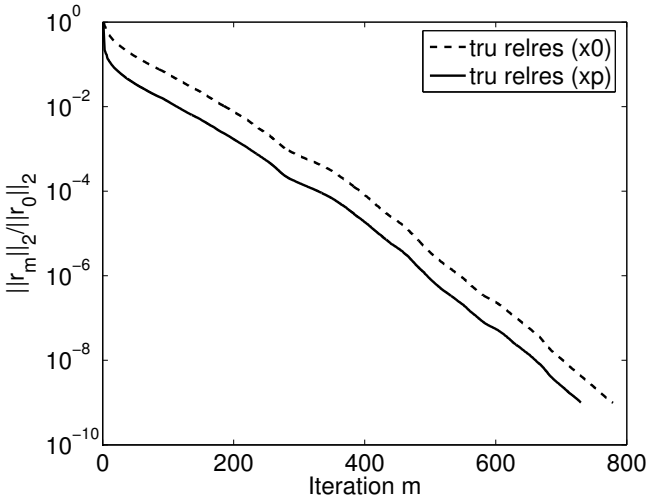
Auch eine Reorthogonalisierung im CSYM-Verfahren vor Berechnung der Singulärpaare liefert keine Reduktion der Iterationszahlen auf unter 778 Iterationen. Abgesehen von einigen Ausnahmen bietet beim CSYM-Verfahren ohne Deflation die Berechnung der Singulärpaare aus dem CSYM-Verfahren mit Reorthogonalisierung eine leichte Reduktion der Iterationszahlen.

Die Größe $|v_k^T q_j|$ für das CSYM-Verfahren mit Deflation für Startvektor $\mathbf{x_p}$ bei 13 Singulärpaaren aus dem CSYM-Verfahren ($m = 120$) ohne Reorthogonalisierung zeigt die folgende Abbildung.



Obwohl wir mit $\mathbf{x_p}$ starten, treten im CSYM-Verfahren mit Deflation schnell Komponenten von V_k in der Basis auf. Dies war für das CSYM-Verfahren mit Deflation mit vorberechneten Singulärpaaren und mit Startvektor $\mathbf{x_0}$ ähnlich. Zusätzlich kommt es zu einem zweiten Auftreten von starken Komponenten in der Basis. Mit Anwendung der Reorthogonalisierung im CSYM-Verfahren zur Ermittlung der Singulärpaare wird nur das zweite Auftreten der Komponenten in der Basis verhindert. Dies liegt daran, dass der projizierte Startvektor $\mathbf{x_p}$ nicht gut genug ist. Dagegen erhalten wir aus dem CSYM-Verfahren mit Reorthogonalisierung bessere Singulärpaare und damit einen etwas besseren Präkonditionierer.

Das unpräkonditionierte CSYM-Verfahren mit Startvektor $\mathbf{x_p}$ liefert dagegen überraschenderweise eine Verbesserung, obwohl permanent Komponenten von V_k in der Basis auftreten. Die Norm der relativen Residuen für das CSYM-Verfahren mit Startvektor $\mathbf{x_0}$ und $\mathbf{x_p}$ bei 25 Singulärpaaren aus dem CSYM-Verfahren ($m = 160$) ohne Reorthogonalisierung zeigt die folgende Abbildung.



Das CSYM-Verfahren mit besser projiziertem Vektor $\mathbf{x_p}$ als Startvektor liefert nicht bessere Ergebnisse als mit Startvektor $\mathbf{x_0}$. Sind dagegen geringe Anteile von V_k gleich zu Anfang des CSYM-Verfahrens in der Basis, so wirken sie konvergenzbeschleunigend.

Welche Singulärpaare?

- Die Singulärpaare zu den größten Singulärwerten werden schnell durch die der Tridiagonalmatrix von CSYM(m) approximiert. Verwenden wir statt dem absoluten das relative Qualitätskriterium, d.h. $\frac{\beta_{m+1}|u_{l,m}|}{\sigma_l} \leq 10^{-5}$, so akzeptieren wir mehr Singulärpaare zu großen Singulärwerten. Wie wir gesehen haben, besteht eine starke Abhängigkeit vom Startvektor der Tridiagonalisierung. Bei gleicher rechter Seite für CSYM(m) und für das CSYM-Verfahren mit Deflation war das absolute Kriterium besser. Das relative Kriterium führte hier zu einer Verschlechterung beim CSYM-Verfahren mit Deflation und kaum zu einer Verbesserung bei dem unpräkonditionierten Verfahren mit Startvektor $\mathbf{x_p}$. Bei verschiedenen rechten Seiten

dagegen war das relative Kriterium etwas besser, insbesondere war hier der Vektor **xp** ein guter Startvektor für das unpräkonditionierte Verfahren.

- Die kleinsten Singulärwerte sind besser als Präkonditionierer geeignet. Auf Grund großer relativer Fehler bei Ermittlung durch CSYM(m) [38] können andere Verfahren zur Ermittlung von Singulärpaaren von A sinnvoller sein. Auch hier bleibt die Ermittlung kleinster Singulärwerte mit guten Singulärvektoren jedoch schwierig [22], [25], [38].

5.5 Zusammenfassung

In diesem Kapitel haben wir einen auf Takagi-Faktoren basierenden Präkonditionierer vorgestellt, in dem die Änderungen der Singulärwerte in exakter Arithmetik berechenbar sind. Der Präkonditionierer erfüllt die Minimaler-Bereich-Eigenschaft und die Anwendung kann mit moderatem Aufwand erfolgen, setzt dazu aber eine ausreichende Orthogonalität der Singulärvektoren des Takagi-Faktors voraus. Für große Matrizen scheint er daher nicht mehr geeignet zu sein, da die berechneten Singulärvektoren an Orthogonalität verlieren. Ungenauigkeiten in der Berechnung des Präkonditionierers scheinen sich hier zu akkumulieren. Ist die Orthogonalität unzureichend, so kann durch Anwendung der Sherman-Morrison-Woodbury-Formel im CSYM-Verfahren mit Deflation noch eine Verbesserung erreicht werden.
Eine weiteres Ziel ist, mit vertretbarem Aufwand gute Singulärpaare zu erhalten. Vor allem die Berechnung der Singulärpaare zu kleinen Singulärwerten ist problematisch. Eine Möglichkeit Singulärpaare zu großen Singulärwerten zu erhalten ist, diese aus der projizierten Tridiagonalmatrix des CSYM-Verfahrens zu berechnen. Die Deflation mit den so erhaltenen Singulärpaaren liefert i.A. nicht die gewünschte Reduktion der Iterationszahlen, unter Umständen kann das CSYM-Verfahren mit projiziertem Startvektor hier sinnvoller sein. Auch zusätzliche Reorthogonalisierungen im CSYM-Verfahren, das auch als Vorstufe zur Berechnung der Singulärpaare dient, brachten bislang nicht den gewünschten Erfolg. In der Praxis hängt das Konvergenzverhalten stark von den rechten Seiten ab (ideales Verhalten für nahezu gleiche rechte Seiten, problematisch wenn zwei rechte Seiten kaum linear abhängig sind).

Kapitel 6

Block-Präkonditionierung

Wir verwenden symmetrische Faktorisierungen von Diagonalblöcken einer komplex symmetrischen Matrix als Präkonditionierer im CSYM-Verfahren. Als Faktorisierungen der Diagonalblöcke untersuchen wir Faktorisierungen in linke untere Dreiecksmatrizen und die Takagi-Faktorisierung.

Die erhofften Vorteile einer Block-Takagi-Präkonditionierung gegenüber der Deflation sind:

- Gute Approximation der SSVD von A durch ausreichend betragsmäßig große Einträge der Diagonalblöcke.
- Genauere Berechnung der Singulärvektoren, da geringere Problemgröße (geringerer Orthogonalitätsverlust).
- Berechnung der SSVD auf kleineren Blöcken, dadurch geringerer Aufwand ($\mathcal{O}(m^2 n)$ bei Blockgröße m).
- Parallelisierung bei der Berechnung der SSVD-Zerlegung für verschiedene Blöcke möglich.

Benutzen wir ausschließlich die Blockdiagonale für den Präkonditionierer, so handelt es sich um einen Block-Jacobi-Präkonditionierer. Diesen Block-Präkonditionierer können wir zu einem Block-SGS-Präkonditionierer erweitern, indem wir die Koeffizienten außerhalb der Blockdiagonalen mit berücksichtigen.

Der erste Schritt ist jedoch, eine geeignete Blockeinteilung zu finden, so dass die Diagonalblöcke viele betragsmäßig große Einträge der Matrix enthalten und möglichst dicht besetzt sind.

6.1 Umordnung mittels TPABLO

Da wir TPABLO später einsetzen werden, erläutern wir im Folgenden die Grundzüge der Strategie, die TPABLO anwendet.
Ziel ist es, Diagonalblöcke mit möglichst vielen betragsmäßig großen Einträgen zu erhalten. Dabei ist wichtig, einen Mittelweg zwischen vielen kleinen Blöcken (im Extremfall der Diagonalen) und wenigen großen Blöcken (im Extremfall die ganze Matrix) zu finden. Viele kleine Blöcke ergeben i.A. einen weniger guten Präkonditionierer als wenige große. Die Berechnungen auf wenigen großen Blöcken sind allerdings aufwändiger.

Das Verfahren PABLO (**Pa**rametrized **Bl**ock **O**rdering) [29] bestimmt in Abhängigkeit von Parametern eine Permutation Π und eine Blockeinteilung, so dass die Diagonalblöcke von $\Pi A \Pi^T$ bestimmte Eigenschaften erfüllen. Es wurde für dünn besetzte Matrizen entwickelt und berücksichtigt nicht die Größe der Einträge der Matrix.

Das Verfahren TPABLO (**T**hreshold **PABLO**) [7] erweitert PABLO, indem es zusätzlich die Einträge der Matrix berücksichtigt, deren Absolutbeträge größer als ein vorgegebener Schwellenwert $\gamma \geq 0$ sind.

Beide Verfahren nutzen graphentheoretische Ansätze für Matrizen, um die Umordnung zu bestimmen. Im Folgenden beschränken wir uns auf symmetrische Matrizen.

Definition 6.1. Der zu einer Matrix A gehörende Kanten gewichtete Graph $G(A) = (V, E, w)$ ist gegeben durch Knoten V, Kanten E und Gewichte w, die in folgender Weise definiert sind:

$$\begin{aligned} V &= \{1, \ldots, n\}, \\ E &= \{\{i, j\} \in V \times V : a_{i,j} \neq 0\}, \\ w(i, j) &= |a_{i,j}|. \end{aligned}$$

Der Grad $d(v)$ eines Knotens $l \in V$, ist gegeben durch

$$d(l) = |\{a_{l,k} \neq 0,\ k \neq l\}|.$$

Definition 6.2. Eine **Clique** ist ein Graph mit einer maximalen Kantenmenge.

Der Graph einer voll besetzten Matrix ist somit eine Clique.

In PABLO geht ein Fülleparameter $\alpha > 0$ in Form einer sogenannten Füllebedingung (FB) und ein Verbundenheitsparameter $\beta \in [0, 1]$ in einer Verbundenheitsbedingung

(VB) ein. Je größer α gewählt wird, desto dichter sollen die Diagonalblöcke sein, wobei u.U. die Blockgröße klein ausfallen kann. Je größer β gewählt wird, desto weniger Einträge sind außerhalb der Diagonalblöcke zugelassen. Die Knoten des Graphen $G(A)$ werden in PABLO in eine Gruppe P (die Koeffizienten in einen Diagonalblock) aufgenommen, wenn FB oder VB erfüllt ist.

In TPABLO muss, in Abhängigkeit vom gewählten TPABLO-Typ, zusätzlich eine sogenannte Schwellenbedingung erfüllt sein.

Definition 6.3. Ein Knoten $j \in V$ erfüllt die **TPABLO1-Bedingung**, falls gilt

$$w(i,j) = |a_{i,j}| > \gamma \text{ für mindestens ein } i \in P.$$

Ein Knoten $j \in V$ erfüllt die **TPABLO2-Bedingung**, falls gilt

$$w(i,j) = |a_{i,j}| > \gamma \text{ für alle } i \in P \text{ mit } \{i,j\} \in E.$$

Der TPABLO-Typ bestimmt, ob die Beträge der Koeffizienten außerhalb der Blockdiagonalen alle kleiner gleich γ (TPABLO1 mit $\alpha = 0$) oder ob die Beträge der Koeffizienten, ausgenommen Diagonalkoeffizienten, in der Blockdiagonalen alle größer als γ (TPABLO2) sein sollen.
In [12] wird eine allgemeinere Form der Bedingung, eine sogenannte XPABLO-Bedingung, definiert, die beide TPABLO-Bedingungen als Spezialfälle enthält.

Definition 6.4. Sei die Matrix $A^{>\gamma}$ definiert durch

$$(a_{i,j}^{>\gamma}) = \begin{cases} a_{i,j} & \text{falls } |a_{i,j}| > \gamma, \\ 0 & \text{sonst.} \end{cases}$$

Der Aufwand von TPABLO ist $\mathcal{O}(n+\nu)$, wobei ν die Anzahl der Kanten von $G(A)$ ist. Wird die Symmetrie der Matrix A berücksichtigt, indem mit dem ungerichteten Graphen der Matrix gearbeitet wird, so kann der Aufwand weiter reduziert werden, aber nicht die Größenordnung.

Sowohl für TPABLO1 als auch für TPABLO2 haben sich die Werte $\alpha = 1$ und $\beta = 0.5$ für viele Problemstellungen als sinnvoll erwiesen. In [12] wird eine spezielle XPABLO-Variante, im Folgenden (Standard)-XPABLO, mit den Werten $\alpha = 1.1$, $\beta = 0.6$ empfohlen. Wesentlich ist, zu einer gegebenen Matrix ein „gutes“ γ zu finden.
Die TPABLO2-Bedingung stellt wesentlich stärkere Anforderungen an die Diagonalblöcke als die TPABLO1-Bedingung, d.h. die Diagonalblöcke werden kleiner. Die (Standard)-XPABLO-Bedingung stellt schwächere Anforderungen als TPABLO1 und TPABLO2 an die Blöcke. Kann eine Blockeinteilung zu gegebenen Parametern bestimmt

werden, so kann sie aus zu vielen kleinen Blöcken bestehen. Die Blockgröße kann durch die Angabe der minimalen und maximalen Blockgröße gesteuert werden.
Die Diagonalblöcke sollen insbesondere regulär sein, was wir im Folgenden voraussetzen.

6.2 Block-Präkonditionierer

Sei die Matrix A (eventuell nach Permutation mit TPABLO) zerlegt in

$$A = B + K, \tag{6.1}$$

mit der Blockdiagonalmatrix B und einer Matrix K.
Sei die Matrix K wiederum zerlegt in untere Block-Dreiecksmatrizen L_K:

$$K = -L_K - L_K^T. \tag{6.2}$$

Mit diesen Zerlegungen definieren wir nun unsere Block-Präkonditionierer.

Definition 6.5. Ein **Block-Jacobi-Präkonditionierer mit Schwellenwert** $\gamma \geq 0$ ist gegeben durch den Präkonditionierer $M = B^{>\gamma}$ mit der Blockdiagonalmatrix B aus (6.1) und einer speziellen Wahl des Faktors S in $M = SS^T$.
Wir unterscheiden anhand der Faktorisierung des Block-Jacobi-Präkonditionierers folgende Präkonditionierer:

- **Block-Takagi-Präkonditionierer** aus einer Takagi-Faktorisierung, also der Zerlegung
$$B^{>\gamma} = S_T S_T^T \text{ mit } S_T,$$
dem Takagi-Faktor $V_{B^{>\gamma}} \Sigma_{B^{>\gamma}}^{1/2}$ von $B^{>\gamma}$.

- **Block-Cholesky-Präkonditionierer** aus der komplex symmetrischen Cholesky-Zerlegung ohne Pivotisierung, d.h. der Zerlegung
$$B^{>\gamma} = S_L S_L^T$$
mit einer linken unteren Dreiecksmatrix S_L.

- **Block-R-Cholesky-Präkonditionierer**. Die Zerlegung basiert auf einer komplex symmetrischen Block-Cholesky-Zerlegung mit Rook-Pivotisierung [1], d.h.
$$\Pi B^{>\gamma} \Pi^T = L_R D_R L_R^T$$

mit einer Permutationsmatrix Π, die nur innerhalb der Diagonalblöcke von $B^{>\gamma}$ permutiert, einer linken unteren Dreiecksmatrix L_R und einer Blockdiagonalmatrix D_R. Die Faktorisierung des Präkonditionierers ist gegeben durch

$$\Pi B^{>\gamma}\Pi^T = S_R S_R^T \text{ mit } S_R = L_R D_R^{1/2},$$

mit einer Blockdiagonalmatrix $D_R^{1/2}$ aus einer Zerlegung $D_R = (D_R^{1/2})(D_R^{1/2})^T$. Für die Zerlegung der (2×2)-Blöcke der Blockdiagonalen D_R nehmen wir entweder die Takagi-Faktorisierung oder eine komplex symmetrische Cholesky-Zerlegung.

Bemerkung 6.6. Wie in Kapitel 4 behandelt, ist nur die Existenz einer Takagi-Faktorisierung gesichert. Existiert eine Cholesky-Faktorisierung, so kann sie numerisch instabil sein, so dass eine Block-Cholesky-Zerlegung z.B. mit Rook-Pivotisierung die bessere Wahl ist.

Definition 6.7. Ein **Block-SGS-Präkonditionierer mit Schwellenwert** $\gamma \geq 0$ ist gegeben durch den Faktor S des Block-Jacobi-Präkonditionierers mit Schwellenwert γ und den Präkonditionierer $M_{SGS} = S_{SGS}S_{SGS}^T$, wobei der Faktor S_{SGS} gegeben ist durch:

$$S_{SGS} = (SS^T - L_K)S^{-T} = S - L_K S^{-T}.$$

Bemerkung 6.8. Die SGS-Faktorisierung wurde in Kapitel 4 eingeführt. Sie ist für Blockgröße 1 identisch mit dem Faktor des Block-SGS-Präkonditionierers mit $\gamma = 0$. Für Blockgröße 1 ist der Faktor des Block-Jacobi-Präkonditionierers aus der Takagi-Faktorisierung derselbe wie in der Cholesky-Zerlegung; er entspricht der Diagonalmatrix und hat die Wurzeln der Diagonalelemente von A als Einträge.

Notation. Betrachten wir gerade eine spezielle Wahl von S bzw. S_{SGS}, so schreiben wir für die Jacobi-präkonditionierte Matrix $\hat{A}_T$, $\hat{A}_L$ oder $\hat{A}_R$ und $\hat{A}_{T-SGS}$, $\hat{A}_{L-SGS}$ oder $\hat{A}_{R-SGS}$ für die SGS-präkonditionierte Matrix, die anderen Matrizen analog.

6.2.1 Eigenschaften der Block-Jacobi-Präkonditionierung

Mit Satz 4.17, der angibt, wann $\sigma_k(M) = \sigma_k(S^H S)$ garantiert wird, gilt für die Block-Jacobi-Präkonditionierer:

- Block-Takagi-Präkonditionierer:
  Da die rechten Singulärvektoren eines Takagi-Faktors $S_T = V_{B^{>\gamma}}\Sigma_{B^{>\gamma}}$ Einheitsvektoren sind, ist die Minimaler-Bereich-Eigenschaft immer erfüllt, d.h.

$$[\sigma_n(S_T S_T^T), \sigma_1(S_T S_T^T)] = [\sigma_n(S_T^H S_T), \sigma_1(S_T^H S_T)].$$

- Block-Cholesky-Präkonditionierer:
  In der Singulärwertzerlegung $S_L = V_{S_L}\Sigma_{S_L}W_{S_L}^H$ ist i.A. $W_{S_L} \notin \mathbb{R}^{n\times n}$. Die Minimaler-Bereich-Eigenschaft liegt i.A. also nicht vor, d.h. es ist

$$[\sigma_n(S_L S_L^T), \sigma_1(S_L S_L^T)] \subseteq [\sigma_n(S_L^H S_L), \sigma_1(S_L^H S_L)].$$

- Block-R-Cholesky-Präkonditionierer:
  Auch der Faktor $S_R = L_R D_R^{1/2}$ erfüllt i.A. nicht die Minimaler-Bereich-Eigenschaft, wobei die Grösse des Bereichs $[\sigma_n(S_R^H S_R), \sigma_1(S_R^H S_R)]$ auch von der Faktorisierung von D_R abhängt.

Bemerkung 6.9. Für den Takagi-Faktor S_T gilt

$$S_T^H S_T = \Sigma_{B^{>\gamma}}.$$

Ist der Block-Jacobi-Präkonditionierer M in einer symmetrischen Faktorisierung $M = SS^T$ gegeben, dann setzen wir analog (6.1)

$$\begin{aligned}\hat{A} &= S^{-1}AS^{-T} = \hat{B} + \hat{K},\\ &\text{mit } \hat{B} = S^{-1}BS^{-T} \text{ und } \hat{K} = S^{-1}KS^{-T}.\end{aligned}$$

Bei Block-Jacobi-Präkonditionierung mit Schwellenwert $\gamma = 0$ gilt $\hat{B} = I$. Wird S in Gleitpunktarithmetik berechnet, so gilt lediglich $\hat{B} \approx I$.

Wie hängen nun die Singulärwerte von $\hat{A}$ von $\hat{K}$ ab?

Für den k-ten Singulärwert von $\hat{A}$ gilt:

$$\begin{aligned}\sigma_k(\hat{A})^2 &= \lambda_k(\hat{A}\hat{A}^H)\\ &= \lambda_k(\hat{K}^H\hat{K} + \hat{K} + \overline{\hat{K}} + I).\end{aligned}$$

Sei $\mathcal{H}(\hat{K}) = \frac{1}{2}(\hat{K} + \overline{\hat{K}})$ der hermitesche Teil von $\hat{K}$, dann gilt:

$$\begin{aligned}\sigma_k(\hat{A})^2 &= \lambda_k(\hat{K}^H\hat{K} + 2\mathcal{H}(\hat{K}) + I)\\ &= \lambda_k(\hat{K}^H\hat{K} + 2\mathcal{H}(\hat{K})) + 1.\end{aligned}$$

Weiter zu untersuchen sind also die Eigenwerte von $\hat{H} = \hat{K}^H\hat{K} + 2\mathcal{H}(\hat{K})$.
Da $\hat{A}\hat{A}^H$ eine hermitesch positiv definite Matrix ist, gilt:

$$\lambda_k(\hat{H}) > 0 \text{ oder } -1 < \lambda_k(\hat{H}) \leq 0.$$

Weiter lässt sich $\sigma_1(\hat{A})$ mit Hilfe von Satz 2.20 abschätzen:

$$\begin{aligned}\sigma_1(\hat{A}) &= \max_{x^H x=1} |x^T \hat{A} x| \\ &= \max_{x^H x=1} |x^T \hat{K} x + x^T x| \\ &\geq |x_1^T \hat{K} x_1 + x_1^T x_1| \text{ für ein } x_1 \text{ mit } x_1^H x_1 = 1.\end{aligned}$$

Wählen wir z.B. $x_1 = e_1 \in \mathbb{C}^n$, so ist $x_1^T \hat{K} x_1 = 0$ und damit $\sigma_1(\hat{A}) \geq 1$, d.h. $\hat{H} = \hat{A}\hat{A}^H - I$ ist entweder positiv definit oder indefinit mit negativen Eigenwerten, die betragsmäßig kleiner als 1 sind.

Damit $\text{cond}\left(\hat{A}\right) \leq \text{cond}\,(A)$ erfüllt ist, muss gelten:

$$\begin{gathered}\frac{\sqrt{\sigma_1(\hat{H})+1}}{\sqrt{\sigma_n(\hat{H})+1}} \leq \text{cond}\,(A) \\ \Leftrightarrow \\ (\sigma_1(\hat{H})+1)\sigma_n^2(A) \leq (\sigma_n(\hat{H})+1)\sigma_1^2(A).\end{gathered}$$

6.2.2 Anwendung der Block-SGS-Präkonditionierung

Wir haben bereits in Kapitel 4 das CSYM-Verfahren mit Eisenstat-Trick für einen ungeblockten SGS-Präkonditionierer vorgestellt. Wir werden Algorithmus 4.4 nun auf die Block-Variante übertragen.

Ein Block-SGS-Präkonditionierer mit Schwellenwert $\gamma = 0$ ist gegeben durch

$$M_{SGS} = (B - L_K)S^{-T}S^{-1}(B - L_K)^T$$

mit der linken unteren Block-Dreiecksmatrix L_K und der faktorisierten Blockdiagonalen $B = SS^T$ aus der Zerlegung $A = B - L_K - L_K^T$.
Wir gehen analog wie bei der Anwendung der ungeblockten SGS-Präkonditionierung vor mit dem einzigen Unterschied, dass wir statt einer Diagonalmatrix D eine Blockdiagonalmatrix B haben.
In Algorithmus 4.4 ist dann $\tilde{S} = B - L_K$ eine untere Blockdreiecksmatrix und B eine Blockdiagonalmatrix. Ist B eine echte Blockdiagonalmatrix, so geht die symmetrische Faktorisierung von B in die Faktorisierung von $T = B^{-1}$ ein. Wählen wir die symmetrische Cholesky-Zerlegung $B = S_L S_L^T$, so gilt $|T|^{-1} = S_L S_L^H$, bei der Takagi-Faktorisierung mit $S_T = V_B \Sigma_B^{1/2}$ dagegen $|T|^{-1} = V_B \Sigma_B V_B^H$.

Aufwand Algorithmus 4.2 und 4.4 bei Block-SGS-Präkonditionierung

- Das Lösen der Gleichungssysteme mit $\tilde{S}$, $\tilde{S}^T$ und $|T|$ in Algorithmus 4.4 entspricht in etwa dem Lösen des Gleichungssystems mit SS^H in Algorithmus 4.2, denn S und $\tilde{S}$ sind untere Block-Dreiecksmatrizen und $|T|$ ist eine Blockdiagonalmatrix.
- Statt einer Matrix-Vektormultiplikation mit S^H bzw. SS^H zur Berechnung von β_m in Algorithmus 4.2 ist in Algorithmus 4.4 eine Multiplikation mit der Blockdiagonalmatrix $|T|$ bzw. $|T|^{1/2}$ durchzuführen.
- Eine Multiplikation mit A in Algorithmus 4.2 wird in Algorithmus 4.4 durch eine Multiplikation mit der Blockdiagonalmatrix $B = T^{-1}$ (und zwei zusätzliche Vektoradditionen) ersetzt.

Es lassen sich analoge Aussagen über den Aufwand wie bei der ungeblockten SGS-Präkonditionierung mit Eisenstat-Trick machen. Auch für Algorithmus 4.4 in der geblockten Variante kann der Aufwand wesentlich billiger als Algorithmus 4.2 und vergleichbar mit dem des unpräkonditionierten CSYM-Verfahrens sein. Das ist besonders dann der Fall, wenn die Diagonalblöcke relativ klein sind.

Bemerkung 6.10. Die Faktoren in der Block-SGS-Präkonditionierung erfüllen i.A. keine Minimaler-Bereich-Eigenschaft.

6.3 Numerische Ergebnisse

Wir werden zuerst das CSYM-Verfahren mit ungeblockter Präkonditionierung testen, d.h. die Blockdiagonale ist durch die Diagonale gegeben, und somit die Takagi-Faktorisierung mit der Cholesky-Zerlegung identisch. Anschließend betrachten wir Block-Präkonditionierungen. Dazu starten wir mit dem Modellproblem aus Kapitel 5, das wir ausführlich untersuchen werden. Dann werden wir Block-Präkonditionierer für fünf weitere dicht besetzte Matrizen von CERFACS und anschließend für dünn besetzte Matrizen aus der Harwell-Boeing Sammlung [10] testen. Zum Abschluss betrachten wir noch Gleichungssysteme mit größeren dünn besetzten Matrizen aus der Elektrodynamik [33], die von U. van Rienen, Universität Rostock, zur Verfügung gestellt wurden. Zur Ermittlung der Blockeinteilungen nutzen wir das mex-Programm *xpablo* von David Fritzsche aus [12] in einer aktuelleren Version. Die komplex symmetrischen Cholesky-Zerlegungen basieren auf den MATLAB-Programmen für hermitesche Cholesky-Zerlegungen aus [21].

Das Modellproblem

Wir betrachten wieder die Testmatrix der Größe $n = 1176$ aus dem ersten Modellproblem (49 Multibeam-Antennen). Tabelle 6.1 zeigt, dass ca. 97% der Einträge einen Betrag von maximal 5 haben.

Tabelle 6.1: Größe der Beträge der Matrix A aus dem ersten Modellproblem

γ	$nnz(A^{>\gamma})$	$\frac{nnz(A^{>\gamma})}{nnz(A^{>0})}$ in %
0	1.382.976	100
5	43.736	3
10	27.496	2
15	14.496	1
20	11.676	1
25	10.388	0.8
30	8.036	0.6
35	6.272	0.5
40	2.744	0.2
45	1.568	0.1
75	0	0

Das CSYM-Verfahren wird abgebrochen, sobald die Norm des akkumulierten relativen Residuums auf 10^{-9} reduziert wurde bzw. nach maximal 1000 Iterationen.

Zunächst betrachten wir als Extremfall in der Blockeinteilung einen Diagonalblock der Größe n und berechnen von $B = A^{>\gamma}$ eine Takagi-Faktorisierung (T), die Cholesky-Zerlegung (L) und die Block-R-Cholesky-Zerlegung mit Takagi-Faktorisierungen der (2×2)-Blöcke (R). Die folgende Tabelle listet die dabei erhaltenen Iterationszahlen und Normen des akkumulierten (acc relres), des berechneten (tru relres) relativen Residuums und des Fehlers (error) auf.

Tabelle 6.2: Werte des CSYM-Verfahrens mit $A^{>\gamma}$-Präkonditionierung

S	γ	*iter*	acc relres	tru relres	error
–	–	778	9.828176e-10	9.828174e-10	7.973643e-09
T	0	4	1.234871e-13	1.950593e-13	2.892056e-13
R	0	2	3.212752e-14	4.799331e-13	9.981166e-13
L	0	2	3.348857e-14	6.036779e-13	1.004047e-12
T	5	304	9.023035e-10	1.168773e-09	5.594766e-09
R	5	1000	6.992610e-04	6.834253e-03	3.377862e-02
L	5	1000	3.127006e-03	4.422883e-02	2.026673e-01
T	10	419	9.610667e-10	1.407212e-09	4.557044e-09
R	10	1000	1.324115e-03	1.321118e-02	6.832372e-02
L	10	1000	3.523434e-03	4.964402e-02	2.420519e-01
T	15	519	9.939461e-10	1.438728e-09	7.494410e-09
R	15	1000	1.523864e-03	1.433436e-02	5.752459e-02
L	15	1000	5.522610e-03	7.403535e-02	2.977569e-01
T	20	343	9.327289e-10	1.091716e-09	6.457723e-09
R	20	1000	4.028252e-03	5.285879e-02	2.516252e-01
L	20	1000	1.084289e-02	2.780295e-01	1.313024e+00
T	25	389	9.981671e-10	1.048945e-09	7.382883e-09
R	25	1000	1.636558e-05	7.846385e-05	4.057621e-04
L	25	1000	1.003332e-03	8.905602e-03	5.587534e-02
T	30	413	9.220310e-10	9.810301e-10	5.528216e-09
R	30	1000	1.959250e-03	1.453157e-02	5.578952e-02
L	30	1000	3.997279e-03	4.383721e-02	1.903991e-01
T	35	435	9.179243e-10	1.032177e-09	4.654036e-09
R	35	1000	3.726670e-07	1.328367e-06	5.183642e-06
L	35	1000	2.394398e-05	1.341715e-04	5.559123e-04
T	40	1000	3.172684e-06	3.651058e-06	2.536517e-05
R	40	683	9.609364e-10	1.636405e-09	6.667212e-09
L	40	1000	5.127897e-06	1.811909e-05	9.681735e-05

Das CSYM-Verfahren mit einer Präkonditionierung, die auf einer Zerlegung der vollen Matrix basiert, konvergiert – wie zu erwarten – nach zwei bis vier Iterationen.
Werden bei der Berechnung des Präkonditionierers nur Werte berücksichtigt, deren Betrag größer als der Schwellenwert γ ist, so zeigt sich bei der Takagi-Präkonditionierung bis $\gamma = 35$ noch eine Verbesserung gegenüber dem unpräkonditionierten CSYM-Verfahren. Der Präkonditionierer aus der Cholesky-Zerlegung liefert schon ab $\gamma = 5$ keine Konvergenz mehr innerhalb von 1000 Iterationsschritten. Nur für $\gamma = 40$ ist das CSYM-Verfahren mit Block-R-Cholesky- besser als mit Takagi-Präkonditionierung.

In der folgenden Tabelle sind als Eigenschaften des Präkonditionierers $A^{>40}$ enthalten:

- Die mittlere quadratische Abweichung der Singulärwerte von XX^H und M^{-1},

$$abw = \|\Sigma(M^{-1}) - \Sigma(XX^H)\|_2,$$

  wobei Σ der Vektor mit den Singulärwerten ist.

- Als Maß für die Minimaler-Bereich-Eigenschaft der Wert

$$mbe = |\sigma_1(M^{-1}) - \sigma_1(XX^H)| + |\sigma_n(M^{-1}) - \sigma_n(XX^H)|.$$

Tabelle 6.3: Eigenschaften von $\hat{A}$ aus der Präkonditionierung mit $A^{>40}$

1176 / 40.00	$\hat{A}_T$	$\hat{A}_R$	$\hat{A}_L$
$iter$	1000	683	1000
abw	2.582350e+00	4.359279e-01	7.674994e-01
mbe	5.774478e-01	6.521743e-02	5.111188e-02
cond $(\cdot)$	5.105973e+02	5.674429e+01	1.357642e+02
$\theta_{1/n}$	7.860117e+00	8.735196e-01	2.089950e+00
θ_1	1.304796e-01	1.226104e-02	1.945230e-02
θ_n	1.660022e-02	1.403636e-02	9.307546e-03
$\sigma_1(XX^H)$	1.314944e+00	3.478918e-02	6.134555e-02
$\sigma_n(XX^H)$	5.260072e-03	4.927359e-03	1.739724e-03
$\|\hat{A} - I\|_2$	5.154710e+01	4.972855e+00	7.814290e+00

Der große Wert mbe in Tabelle 6.3 zeigt, dass der Takagi-Präkonditionierer numerisch keine Minimaler-Bereich-Eigenschaft hat. Die beiden anderen Block-Präkonditionierer

liefern einen etwas kleineren Bereich. Nur für den Block-R-Cholesky-Präkonditionierer hat $\hat{A}$ eine kleinere Kondition als die unpräkonditionierte Matrix. Auf $A_{\Pi}^{>40}$ bzw. $A_{\Pi}^{>25}$ nach einer weiteren Permutation Π werden wir bei der Block-Präkonditionierung zurückkommen.

Als nächstes nehmen wir n Diagonalblöcke der Größe 1 als Präkonditionierer.
Wir invertieren nur die Diagonaleinträge (eigentlicher Jacobi- bzw. symmetrischer Gauß-Seidel-Präkonditionierer (SGS)). Dazu berechnen wir die Wurzeln der Diagonaleinträge.

Die folgende Tabelle listet die erhaltenen Iterationszahlen bei SGS- und Jacobi-Präkonditionierung auf.

Tabelle 6.4: Iterationszahlen des CSYM-Verfahrens mit Jacobi- und SGS-Präkonditionierung

Typ	*iter*	acc relres	tru relres	error
–	778	9.828176e-10	9.828174e-10	7.973643e-09
Jac	779	9.863979e-10	9.863984e-10	8.082115e-09
SGS	1000	7.501209e-02	1.018347e+00	3.098141e+00

Die Iterationszahlen ändern sich kaum mit Jacobi-Präkonditionierer gegenüber denen des unpräkonditionierten Verfahrens. Da die Beträge der Diagonalelemente nicht stark variieren, ist dies auch nicht überraschend. Das Verfahren mit SGS-Präkonditionierung divergiert.

Wir nehmen die durch Umordnung mittels TPABLO1 [7] ($\alpha = 1,\ \beta = 0.5$) erzeugten Diagonalblöcke als Basis für die Takagi-Faktorisierung (alle Einträge, die betragsmäßig größer als die vorgegebene Schranke γ sind, liegen in den Diagonalblöcken). Als weitere Eingabewerte geben wir *minbs* und *maxbs*, die gewünschte minimale bzw. die gewünschte maximale Blockgröße, vor.

Wir betrachten die in der folgenden Tabelle angegebenen Blockeinteilungen, die wir durch Anwendung von TPABLO1 erhalten. Um Verwechslungen mit dem Schwellenwert γ der Block-Präkonditionierer zu vermeiden, notieren wir den Schwellenwert für das TPABLO-Verfahren mit γ_T. Die restlichen Größen der Tabelle, die Ausgabewerte von TPABLO sind, haben folgende Bedeutung:

- *bn* gibt die Zahl der Blöcke an,

- *min* ist die Größe des kleinsten Blocks,
- *avg* ist die durchschnittliche Blockgröße,
- *max* ist die Größe des größten Blocks.

Die tatsächliche minimale Blockgröße *min* kann kleiner als die gewünschte Blockgröße *minbs* sein, da der letzte ermittelte Block kleiner ausfallen kann.

Tabelle 6.5: Blockeinteilungen für **A1** mit TPABLO1

Bez.	γ_T	*minbs*	*maxbs*	*bn*	*min*	*avg*	*max*
bo1	25.00	1	1176	49	24	24.00	24
bo2	40.00	1	1176	392	3	3.00	3

Im Folgenden werden wir für diese beiden Blockeinteilungen (bo1 und bo2) Präkonditionierer für verschiedene Schwellenwerte γ testen. Damit wir immer den gleichen Präkonditionierer M anwenden, gehen wir dabei, wenn nicht anders angegeben, stets von der Matrix aus, die wir durch weitere beidseitige Permutation auf Grund der Rook-Pivotisierung erhalten. Hierbei wird nur innerhalb der einzelnen Blöcke permutiert.

(24×24)-Blöcke

Für $\gamma_T = 25, 30, 35$ erhalten wir 49 (24×24)-Blöcke, was auf Grund der Tatsache, dass die Matrix 49 Multibeam-Antennen modelliert, plausibel ist. Durch geschickte Anordnung in den Blöcken kann z.B. der Aufwand für die Berechnung der Takagi-Faktoren reduziert werden. Wir berechnen Faktorisierungen der (24×24)-Blockdiagonalen $B^{>\gamma}$ für $\gamma = 0$ und $\gamma = 25$ und nutzen diese direkt als Block-Jacobi-Präkonditionierer oder für den Block-SGS-Präkonditionierer im CSYM-Verfahren.

Bemerkung 6.11. Da in der unpermutierten Matrix alle Diagonalblöcke bis auf den ersten gleich sind, können hier die Faktoren der Blöcke, mit Ausnahme des ersten Blocks, gleich gewählt werden. Der erste Block hat die gleichen Singulärwerte wie die restlichen Blöcke.

Die folgenden Abbildungen zeigen Besetztheitsmuster der Blockdiagonalen $B^{>25}$ für einen Ausschnitt der Größe 96×96 bei natürlicher Anordnung, der TPABLO1 Anordnung und zusätzlichen Umordnungen innerhalb jedes Blocks.

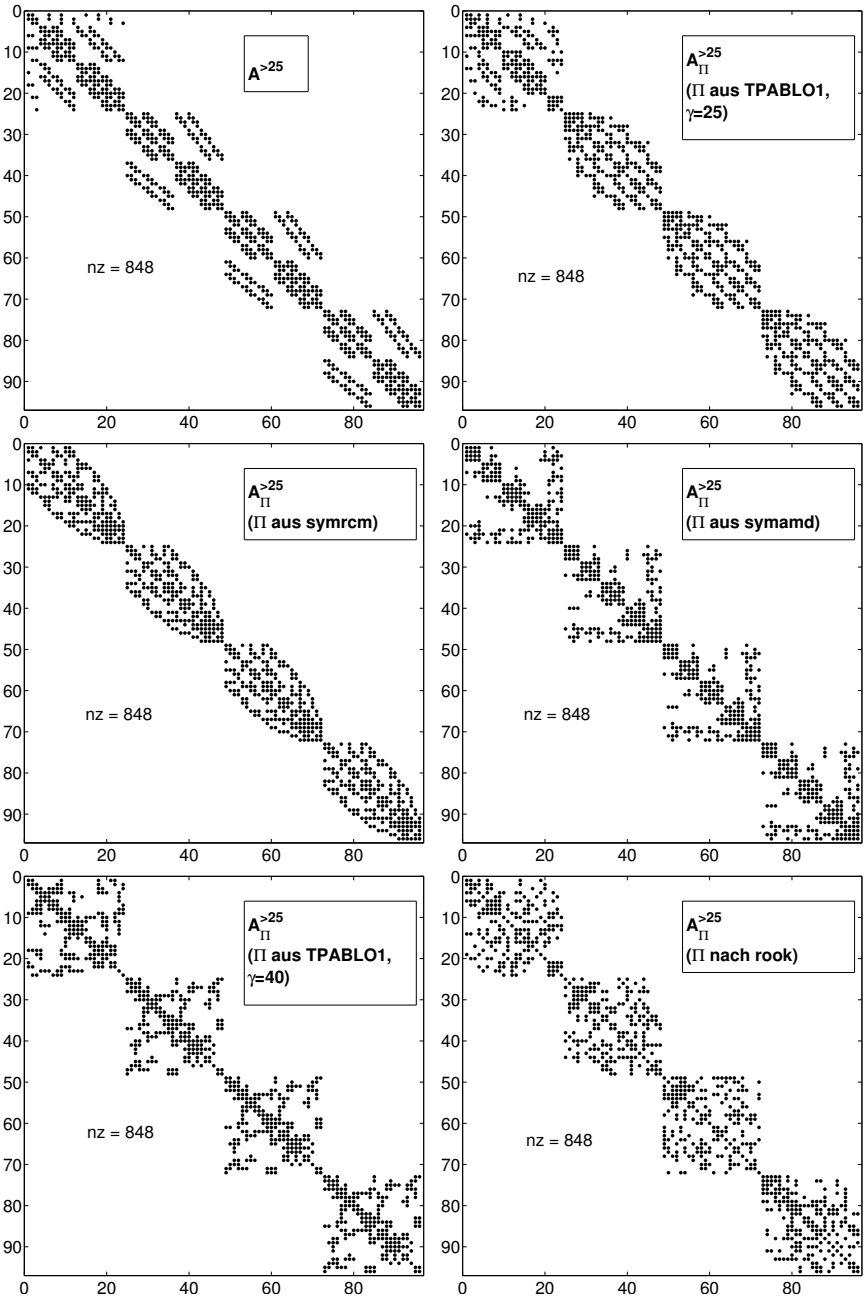

A>25
nz = 848
A>25 Π (Π aus TPABLO1, γ=25)
nz = 848
A>25 Π (Π aus symrcm)
nz = 848
A>25 Π (Π aus symamd)
nz = 848
A>25 Π (Π aus TPABLO1, γ=40)
nz = 848
A>25 Π (Π nach rook)
nz = 848

Tabelle 6.6: Iterationszahlen des CSYM-Verfahrens mit (24×24)-Block-Präkonditionierung (bo1 in Tabelle 6.5)

Typ	S	γ	*iter*	acc relres	tru relres	error
–	–	–	778	9.828176e-10	9.828174e-10	7.973643e-09
Jac	T	0	322	9.594965e-10	1.156090e-09	6.113651e-09
Jac	R	0	1000	1.107789e-04	8.135333e-04	4.584055e-03
Jac	L	0	1000	2.281005e-04	2.347216e-03	8.506999e-03
SGS	T	0	252	9.915293e-10	2.270033e-09	8.021491e-09
SGS	R	0	1000	1.376054e-06	1.562676e-05	4.055869e-05
SGS	L	0	1000	8.324621e-06	1.225266e-04	3.156219e-04
Jac	T	25	389	9.934474e-10	1.039219e-09	7.270858e-09
Jac	R	25	1000	5.994724e-04	3.428669e-03	1.813533e-02
Jac	L	25	1000	2.207047e-03	2.042464e-02	1.411834e-01
SGS	T	25	297	9.229581e-10	1.206481e-09	6.491309e-09
SGS	R	25	1000	2.029989e-05	1.405052e-04	4.972923e-04
SGS	L	25	1000	2.425095e-04	2.668563e-03	1.292306e-02

Nur die CSYM-Verfahren mit Block-Takagi-Präkonditionierung führen zu einer Verbesserung im Vergleich zum unpräkonditionierten CSYM-Verfahren. Wir erhalten folgende Ergebnisse für die CSYM-Verfahren mit Block-Takagi-Präkonditionierung:

- **Ohne Schwellenwert**
  Die Präkonditionierung mit der vollen Blockdiagonalen liefert eine Reduktion der Iterationszahl auf 32.4% (SGS) bzw. auf 41.3% (Jacobi) gegenüber dem unpräkonditionierten Verfahren.

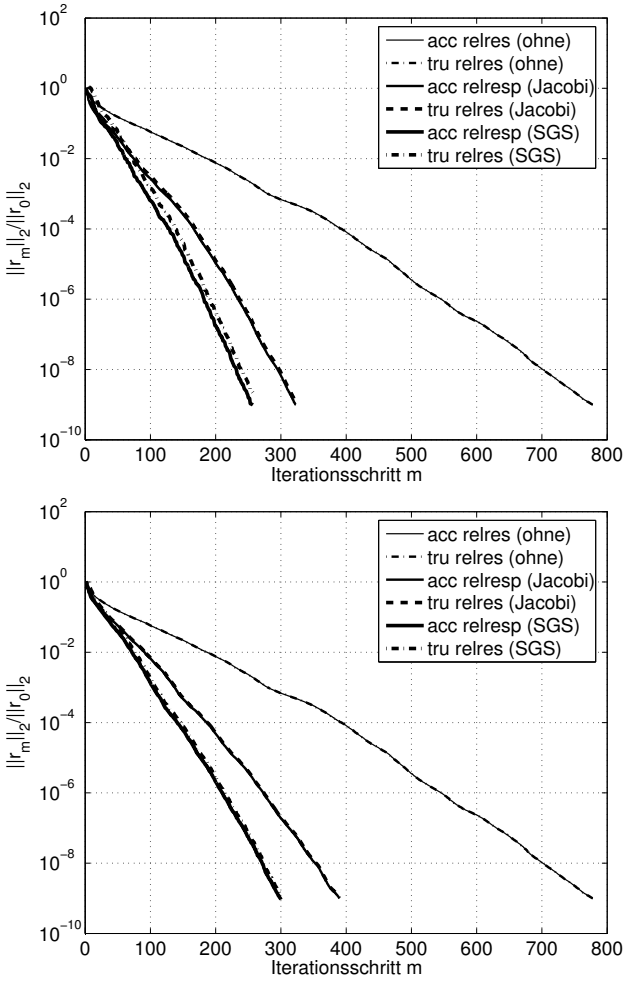
- **Mit Schwellenwert 25**
  Nehmen wir nur die Werte der Blockdiagonalen mit einem Betrag größer 25 für den Präkonditionierer, so ist die Reduktion wie erwartet geringer (38.2% bei SGS, 50% bei Jacobi).

- **Block-SGS- vs. Block-Jacobi-Präkonditionierung**
  Das CSYM-Verfahren mit Block-SGS-Präkonditionierung konvergiert für beide Schwellenwerte schneller als mit Block-Jacobi-Präkonditionierung.

Die Ergebnisse für $\gamma = 25$ sind vergleichbar mit denen aus Tabelle 6.2, wo wir $A^{>25}$ als Präkonditionierer gewählt haben.

Das Konvergenzverhalten des CSYM-Verfahrens mit (24×24)-Block-Präkonditionierung zeigen die folgenden Abbildungen.

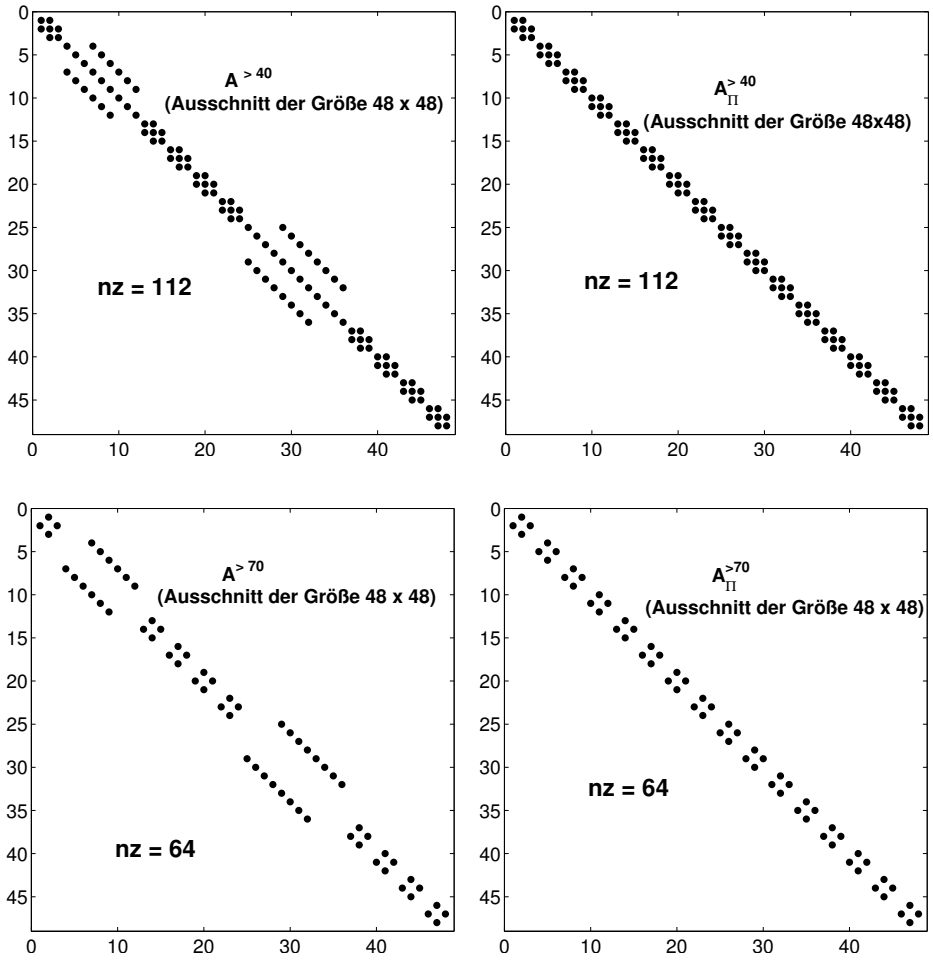
Abbildung 6.1: CSYM-Verfahren mit (24×24)-Block-Präkonditionierung ($B^{>0}$ (oben), $B^{>25}$ (unten), relative Residuen)



Die akkumulierten relativen Residuen (präkonditioniertes System) und die berechneten relativen Residuen (ursprüngliches System) weichen in allen Fällen nicht wesentlich voneinander ab.

(3×3)-Blöcke

Ab dem Wert $\gamma_T = 40$ erhalten wir mit TPABLO1 392 Blöcke der Größe 3. Die folgenden Abbildungen zeigen die Einträge größer 40 (oben) und größer 70 (unten), jeweils für die ursprüngliche Matrix (links) und nach Umordnung durch TPABLO (rechts).



Nach Umordnung mit TPABLO sind die Blöcke der Blockdiagonalen $B^{>40}$ Tridiagonalmatrizen mit Koeffizienten größer 70 in den Nebendiagonalen.

Wir testen das CSYM-Verfahren mit (3×3)-Block-Präkonditionierung mit Faktorisierungen wie in der vorigen Blockzerlegung und Schwellenwert $\gamma = 0$ und $\gamma = 40$.

Die folgende Tabelle listet die dabei erhaltenen Iterationszahlen und Normen des akkumulierten (acc relres), des berechneten (tru relres) relativen Residuums und des Fehlers auf.

Tabelle 6.7: Iterationszahlen des CSYM-Verfahrens mit (3×3)-Block-Präkonditionierung (bo2 in Tabelle 6.5)

Typ	S	γ	*iter*	acc relres	tru relres	error
–	–	–	778	9.828176e-10	9.828174e-10	7.973643e-09
Jac	T	0	585	9.406139e-10	1.464961e-09	5.985254e-09
Jac	R	0	719	9.863902e-10	1.896739e-09	7.417915e-09
Jac	L	0	1000	5.593385e-07	2.083278e-06	7.490243e-06
SGS	T	0	1000	1.101065e-04	1.186816e+00	3.125908e+00
SGS	R	0	1000	1.027621e-04	1.209393e+00	3.124922e+00
SGS	L	0	1000	2.497546e-04	1.170059e+00	3.114367e+00
Jac	T	40	451	9.204476e-10	1.054349e-09	4.478079e-09
Jac	R	40	671	9.987706e-10	1.707426e-09	7.865064e-09
Jac	L	40	671	9.700952e-10	1.660359e-09	7.639649e-09
SGS	T	40	1000	1.381241e-03	6.828877e-03	4.751416e-02
SGS	R	40	1000	5.583869e-03	3.494748e-02	1.914083e-01
SGS	L	40	1000	5.868448e-03	3.667236e-02	2.005701e-01

Von den Block-Jacobi-Präkonditionierern liefern auch hier die Takagi-Präkonditionierer die besten Ergebnisse, genauer:

- **Mit Schwellenwert 40**
  Nehmen wir nur die Werte der Blockdiagonalen mit einem Betrag größer als 40 für den Präkonditionierer, beträgt die Iterationszahl etwa 58% der des unpräkonditionierten Verfahrens.
- **Ohne Schwellenwert**
  Die Präkonditionierung mit der vollen Blockdiagonalen (die Einträge mit Beträgen kleiner als 40, genauer zwischen 23 und 25, werden mitberücksichtigt) liefert nur eine Reduktion der Iterationszahl auf 75% gegenüber dem unpräkonditionierten Verfahren.
- **Block-Cholesky-Präkonditionierung**
  Bei Präkonditionierung mit der vollen Blockdiagonalen konvergiert das CSYM-

Verfahren nicht innerhalb von 1000 Iterationen. Nur mit dem Jacobi-Präkonditionierer mit Schwellenwert 40 ergibt sich eine Konvergenzbeschleunigung im Vergleich zum unpräkonditionierten Verfahren. Die Cholesky-Zerlegung und die Block-R-Cholesky-Zerlegung stimmen für die Blockdiagonale mit Schwellenwert 40 nahezu überein.

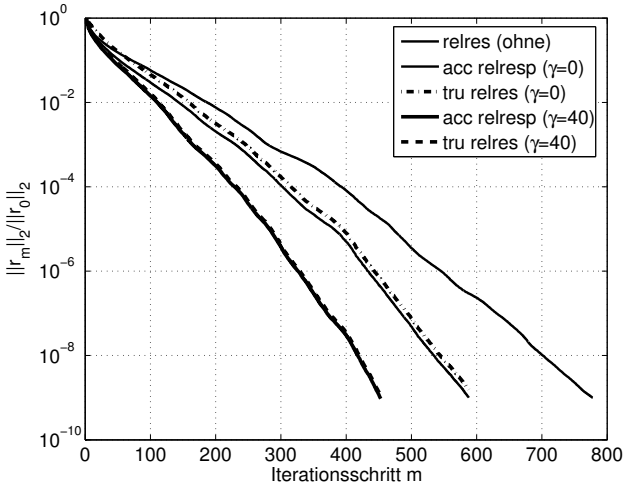
- **Block-R-Cholesky-Präkonditionierung**
  Das CSYM-Verfahren mit Block-R-Cholesky-Präkonditionierung ist für die volle Blockdiagonale schon bei einer Blockgröße von 3×3 dem mit Cholesky-Präkonditionierung überlegen. Es ist allerdings langsamer als mit (3×3)-Block-Takagi-Präkonditionierung (671 Iterationen statt 451 bei $\gamma = 40$, und 729 statt 587 Iterationen mit $\gamma = 0$).

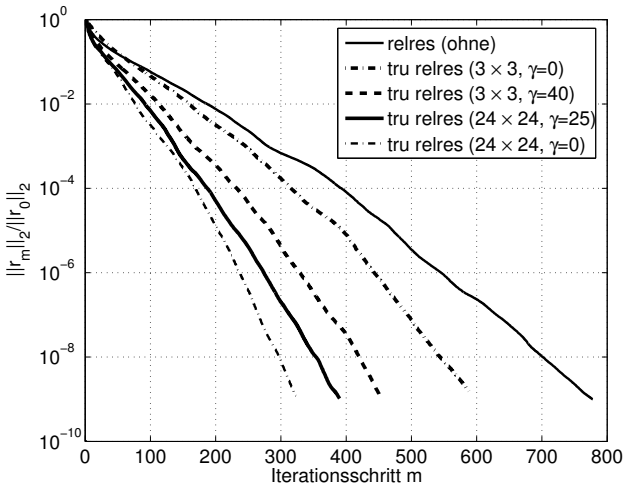
Die besseren Ergebnisse mit Block-Jacobi-Präkonditionierer für den Schwellenwert 40 gegenüber dem ohne Schwellenwert, hängen mit der speziellen Struktur von $B^{>\gamma}$ zusammen. Insbesondere sind die Blöcke tridiagonal. Eine weitere Untersuchung ergab, dass die Real- und Imaginärteile der Eigenwerte beide entweder positiv bzw. negativ sind.

3 / 40.00	$\hat{A}_T$	$\hat{A}_R$	$\hat{A}_L$
iter	451	671	671
abw	2.150987e-16	2.409401e-01	2.409401e-01
mbe	2.168404e-18	7.159374e-03	7.159374e-03
$\mathrm{cond}\,(\cdot)$	4.119162e+01	5.670951e+01	5.670951e+01
$\theta_{1/n}$	6.341023e-01	8.729843e-01	8.729843e-01
θ_1	1.034699e-02	1.226147e-02	1.226147e-02
θ_n	1.631754e-02	1.404546e-02	1.404546e-02
$\sigma_1(XX^H)$	2.286002e-02	3.478918e-02	3.478918e-02
$\sigma_n(XX^H)$	7.316941e-03	4.927359e-03	4.927359e-03
$\|\hat{A} - I\|_2$	3.651313e+00	4.966632e+00	4.966632e+00

Zu Beginn haben wir für den Extremfall eines Blocks, der der Ausgangsmatrix entspricht, auch die Block-Präkonditionierer basierend auf unterschiedlichen Faktorisierungen für $\gamma = 40$ getestet (s. Tabelle 6.2). Dabei lieferte das CSYM-Verfahren mit R-Präkonditionierung die besten Ergebnisse (683 Iterationen), das Verfahren konvergierte sowohl mit (L) als auch mit (T)-Präkonditionierung nicht innerhalb von 1000 Iterationen. Dies weist darauf hin, dass auch die Qualität des berechneten Takagi-Faktors stark von der Anordnung abhängt.

Die CSYM-Verfahren mit (3×3)-Block-SGS-Präkonditionierung konvergieren nicht innerhalb von 1000 Iterationen. Das Konvergenzverhalten des CSYM-Verfahrens mit (3×3)-Block-Takagi-Präkonditionierung bzw. mit den bisher betrachteten Block-Jacobi-Präkonditionierungen zeigen die folgenden Abbildungen.





Die folgende Tabelle listet die Besten der betrachteten Präkonditionierer zusammen mit $\theta_{1/n}$ und *mbe* der präkonditionierten Matrix auf.

Tabelle 6.8: Ranking der Block-Präkonditionierer für das CSYM-Verfahren

Nr.	*iter*	$\hat{A}$	*bo*	γ	$\theta_{1/n}$	*mbe*
1	252	$\hat{A}_{T-SGS}$	bo1	0	1.343708e+00	4.073151e-01
2	297	$\hat{A}_{T-SGS}$	bo1	25	4.676341e-01	4.561023e-02
3	322	$\hat{A}_T$	bo1	0	3.976859e-01	8.890458e-18
4	389	$\hat{A}_T$	bo1	25	4.533795e-01	2.234966e-13
5	451	$\hat{A}_T$	bo2	40	6.341023e-01	2.168404e-18
6	585	$\hat{A}_T$	bo2	0	8.475269e-01	2.602085e-17
7	671	$\hat{A}_R$	bo2	40	8.729843e-01	7.159374e-03
8	719	$\hat{A}_R$	bo2	0	9.792408e-01	3.876266e-03
9	778	A	–	–	1	–

Die Tabelle zeigt, dass das Konvergenzverhalten des CSYM-Verfahrens mit Block-Jacobi-Präkonditionierung um so besser ist, je kleiner $\theta_{1/n}$, und damit die Kondition der präkonditionierten Matrix, ist. Ein kleiner Wert *mbe* ist hinreichend, aber nicht notwendig für die Reduzierung der Kondition, was an den Plätzen 2, 7 und 8 deutlich wird.
Für den Erfolg der Block-SGS-Präkonditionierung (Plätze 1 und 2) sind andere Kriterien ausschlaggebend, sie sind effektiver als die entsprechenden Jacobi-Varianten, die jeweils kleinere $\theta_{1/n}$-Werte haben.

Analyse der Block-Jacobi-Präkonditionierung

Wir betrachten exemplarisch die Werte der Block-Jacobi-Präkonditionierung für bo1 (Blockeinteilung mit (24×24)-Blöcken ohne Schwellenwert), die unter den betrachteten Block-Jacobi-Präkonditionierern die besten Ergebnisse liefert.

Für die ursprünglich Matrix A gilt $\mathrm{cond}\,(A) = 64.96$, $iter = 778$ und $\|A - I\|_2 = 399.69$.

Tabelle 6.9: Eigenschaften von $\hat{A}$ aus der (24×24)-Block-Jacobi-Präkonditionierung

24 / 0.00	$\hat{A}_T$	$\hat{A}_R$	$\hat{A}_L$
$iter$	322	1000	1000
abw	2.640657e-16	1.908772e+00	2.113956e+00
mbe	8.890458e-18	1.244750e-01	1.348586e-01
$\mathrm{cond}\,(\cdot)$	2.583388e+01	2.711572e+02	3.084682e+02
$\theta_{1/n}$	3.976859e-01	4.174185e+00	4.748550e+00
θ_1	6.469899e-03	2.324571e-02	2.424661e-02
θ_n	1.626887e-02	5.568922e-03	5.106109e-03
$\sigma_1(XX^H)$	3.331974e-02	2.791479e-01	2.998123e-01
$\sigma_n(XX^H)$	3.598973e-03	4.771040e-04	3.744092e-04
$\|\hat{A} - I\|_2$	2.521056e+00	9.298469e+00	9.692580e+00

Zunächst untersuchen wir den maximalen Wertebereich der θ_k und anschließend die tatsächlich angenommenen Werte θ_k für die betrachteten Faktorisierungen.
Die tatsächlich angenommenen θ_k-Werte sind jedoch auch für $\hat{A}_R$ alle in dem Intervall

$$[\sigma_n(X_T X_T^H), \sigma_1(X_T X_T^H)] \subset [\sigma_n(X_R X_R^H), \sigma_1(X_R X_R^H)]$$

enthalten.
Die Relation

$$[\sigma_n(S_T)^{-1}, \sigma_1(S_T)^{-1}] \subset [\sigma_n(S_R)^{-1}, \sigma_1(S_R)^{-1}]$$

bleibt für die θ_k-Werte erhalten, d.h.

$$[\min(\theta_k(\hat{A}_T)), \max(\theta_k(\hat{A}_T))] \subset [\min(\theta_k(\hat{A}_R)), \max(\theta_k(\hat{A}_R))].$$

Abbildung 6.2: Absteigend sortierte Singulärwerte von XX^H

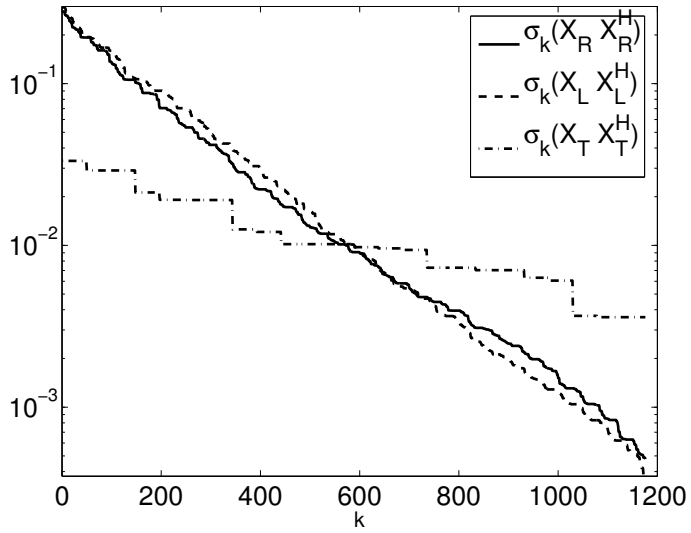
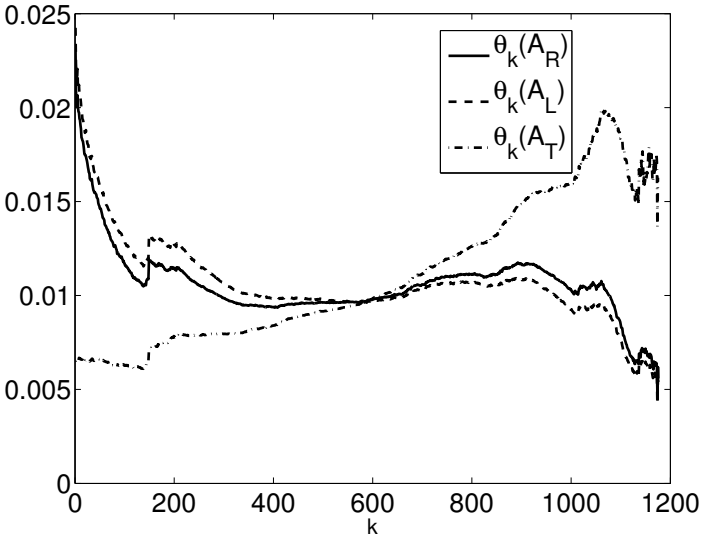


Abbildung 6.3: Angenommene θ_k-Werte für $\hat{A}$



Für $\hat{A}_R$ und $\hat{A}_L$ werden die globalen Extrema der θ_k-Werte zu Singulärwerten nahe der extremalen Singulärwerte angenommen. In θ_1 wird das globale Maximum angenommen, in θ_{n-2} das globale Minimum. Die Grundtendenz der θ_k-Werte ist fallend, während die Grundtendenz bei den θ_k-Werten von $\hat{A}_T$ steigend ist. Ein Anstieg aufeinander folgender θ_k-Werte ist ein notwendiges aber nicht hinreichendes Kriterium für mehrfache Singulärwerte (s. Bemerkung zu Lemma 4.13.)
Tatsächlich hat $\hat{A}_T$ auch mehr geclusterte Singulärwerte als $\hat{A}$, wie folgende Tabelle zeigt. Dabei gelten zwei Singulärwerte als identisch, wenn ihre absolute Differenz weniger als 10^{-6} beträgt. Die gleichen Ergebnisse erhielten wir für alle betrachteten Block-Jacobi-Präkonditionierer.

Tabelle 6.10: Vielfachheit der Singulärwerte der Block-Jacobi-präkonditionierten Matrizen im Vergleich zur unpräkonditionierten Matrix

m	A	$\hat{A}_T$	$\hat{A}_R$	$\hat{A}_L$
1	1046	588	1176	1176
2	65	294	0	0

Tabelle 6.11: Eigenschaften von $\hat{A}$ aus der (24 × 24)-Block-Präkonditionierung (mit Schwellenwert 25)

24 / 25.00	$\hat{A}_T$	$\hat{A}_R$	$\hat{A}_L$
iter	389	1000	1000
abw	4.127181e-09	1.569258e+00	1.973587e+00
mbe	2.234966e-13	1.313658e-01	1.781226e-01
$\mathrm{cond}\,(\cdot)$	2.945177e+01	2.768472e+02	4.093526e+02
$\theta_{1/n}$	4.533795e-01	4.261776e+00	6.301560e+00
θ_1	6.079259e-03	1.908503e-02	2.297809e-02
θ_n	1.340876e-02	4.478186e-03	3.646412e-03
$\sigma_1(XX^H)$	1.817702e-02	2.771982e-01	3.705944e-01
$\sigma_n(XX^H)$	4.154536e-03	4.441597e-04	3.266150e-04
$\|\hat{A} - I\|_2$	1.762977e+00	7.501210e+00	9.081155e+00

Für die Block-Jacobi-Präkonditionierung mit Schwellenwert 25 ergeben sich ähnliche Abbildungen bzw. Eigenschaften.

Abbildung 6.4: Angenommene θ_k-Werte für $\hat{A}_T$ und $\hat{A}_L$

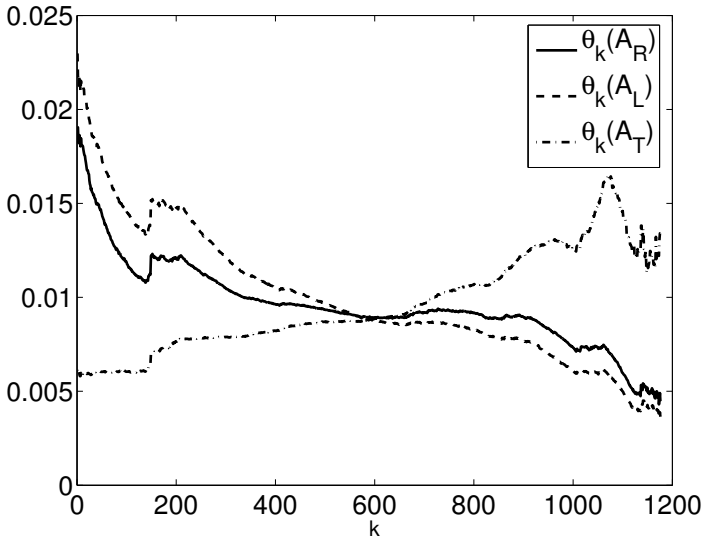
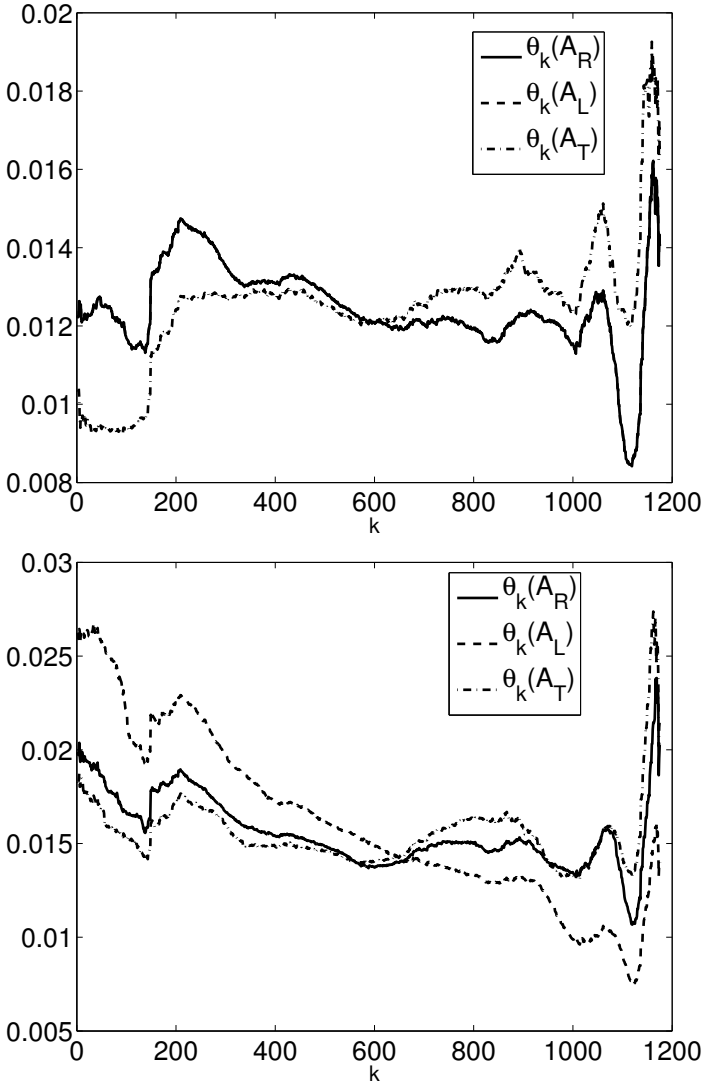


Tabelle 6.12: Eigenschaften von $\hat{A}$ aus der (3×3)-Block-Präkonditionierung

3 / 0.00	$\hat{A}_T$	$\hat{A}_R$	$\hat{A}_L$
iter	585	719	1000
abw	6.870425e-16	1.321837e-01	6.124973e-01
mbe	2.602085e-17	3.876266e-03	1.702480e-02
cond $(\cdot)$	5.505578e+01	6.361199e+01	1.220800e+02
$\theta_{1/n}$	8.475269e-01	9.792408e-01	1.879296e+00
θ_1	1.828008e-02	1.986388e-02	2.624608e-02
θ_n	2.156873e-02	2.028498e-02	1.396591e-02
$\sigma_1(XX^H)$	4.481119e-02	5.129668e-02	7.544960e-02
$\sigma_n(XX^H)$	6.701379e-03	5.434333e-03	3.290182e-03
$\|\hat{A} - I\|_2$	7.017347e+00	7.673933e+00	1.022865e+01

Abschließend betrachten wir noch den Fall der (3×3)-Block-Präkonditionierung mit Schwellenwert 40 (oben) und ohne (unten).

Abbildung 6.5: Angenommene θ_k-Werte für $\hat{A}_T$ und $\hat{A}_L$



Analyse der Block-SGS-Präkonditionierung

Im Folgenden untersuchen wir die Plätze 1 und 2, d.h. die (24×24)-Block-SGS-Präkonditionierer.

Tabelle 6.13: Eigenschaften von $\hat{A}$ aus der (24×24)-Block-SGS-Präkonditionierung ohne Schwellenwert

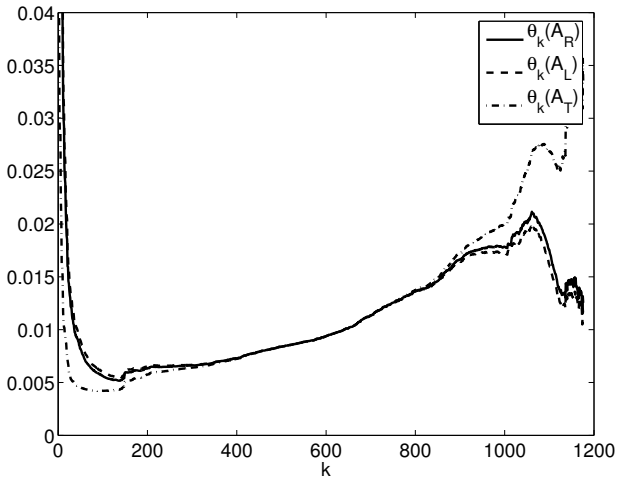
24 / 0.00	$\hat{A}_{T-SGS}$	$\hat{A}_{R-SGS}$	$\hat{A}_{L-SGS}$
iter	252	1000	1000
abw	1.569993e+00	5.473317e+00	5.880565e+00
mbe	4.073151e-01	1.273926e+00	1.349466e+00
cond $(\cdot)$	8.728796e+01	6.649770e+02	7.846195e+02
$\theta_{1/n}$	1.343708e+00	1.023663e+01	1.207841e+01
θ_1	4.510422e-02	1.284010e-01	1.372888e-01
θ_n	3.356698e-02	1.254328e-02	1.136647e-02
$\sigma_1(XX^H)$	8.462293e-01	2.578024e+00	2.729067e+00
$\sigma_n(XX^H)$	1.774240e-03	3.461201e-04	3.097551e-04
$\|\hat{A} - I\|_2$	1.784167e+01	5.117716e+01	5.472322e+01

Der SGS-Takagi-Präkonditionierer ohne Schwellenwert liefert die besten Ergebnisse, obwohl sich die Kondition der präkonditionierten Matrix im Vergleich zur unpräkonditionierten Matrix leicht erhöht.

Die folgende Tabelle zeigt, dass die präkonditionierte Matrix mehr einfache Singulärwerte hat als die unpräkonditionierte, aber einen zusätzlichen Cluster von 31 Singulärwerten.

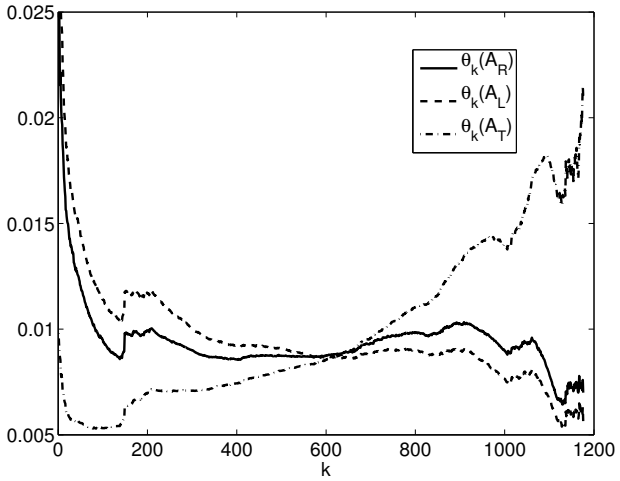
m	A	$\hat{A}_T$	$\hat{A}_R$	$\hat{A}_L$
1	1046	1131	1144	1145
2	65	7	2	3
3	0	0	1	0
25	0	0	1	1
31	0	1	0	0

Abbildung 6.6: Angenommene θ_k-Werte für $\hat{A}_T$ und $\hat{A}_L$



Die Abbildung zeigt, dass bis auf die θ_k-Werte für ein paar größte Singulärwerte, auch hier ein Aufwärtstrend zu erkennen ist. Ein ähnliches Bild liefert die Darstellung für den Block-SGS-Präkonditionierer mit Schwellenwert 25.

Abbildung 6.7: Angenommene θ_k-Werte für $\hat{A}$



Weitere dicht besetzte Matrizen

Wir testen die dicht besetzten Matrizen **C1**, ..., **C5** von CERFACS [6] aus der Diskretisierung nach der BEM (Boundary Element Method) aus dem Gebiet der Elektrodynamik.

Die rechte Seite der Gleichungssysteme wurde bestimmt, indem die Matrix mit der Lösung, einem zufällig erzeugten Vektor, multipliziert wurde.

Die nächste Tabelle listet die Iterationszahlen des unpräkonditionierten CSYM-Verfahrens und der CSYM-Verfahren mit ungeblockten Präkonditionierern auf. Die Iterationen wurden beendet, sobald die Norm des akkumulierten relativen Residuums um mehr als 10^{-9} reduziert wurde oder die maximale Anzahl der zugelassenen Iterationen (*maxiter*), in Abhängigkeit von der Größe der Matrix, erreicht wurde.

Tabelle 6.14: Ungeblockte Präkonditionierung der Matrizen von CERFACS

Matrix	n	Typ	*iter*	acc relres	tru relres	error
C1	1080	–	1000	1.988195e-05	1.988195e-05	1.944102e-04
	1080	Jac	1000	8.618854e-06	7.406696e-06	6.792290e-05
	1080	SGS	830	9.683099e-10	1.041052e-09	6.842675e-09
C2	1299	–	1000	1.004092e-06	1.004092e-06	3.336948e-02
	1299	Jac	1000	3.842894e-07	4.336303e-07	3.588160e-03
	1299	SGS	933	9.979045e-10	1.616860e-09	9.834019e-07
C3	1701	–	1500	7.733890e-06	7.733890e-06	1.604213e-02
	1701	Jac	1500	2.445651e-06	3.110950e-06	7.607192e-03
	1701	SGS	1151	9.843037e-10	8.120389e-10	1.510457e-06
C4	2016	–	888	9.905856e-10	9.905856e-10	1.186593e-06
	2016	Jac	832	9.999101e-10	9.977709e-10	1.170635e-06
	2016	SGS	434	9.865238e-10	6.746381e-10	5.583928e-07
C5	2430	–	1215	9.891208e-10	9.891208e-10	5.658888e-07
	2430	Jac	1148	9.989217e-10	1.011936e-09	5.896385e-07
	2430	SGS	1959	9.858895e-10	6.815891e-10	3.377554e-07

Trotz der relativ kleinen Dimensionen der Matrizen **C1**, **C2** und **C3** sind die zugehörigen Systeme mit dem unpräkonditionierten CSYM-Verfahren relativ schwer zu lösen.

Nur für die Gleichungssysteme mit den Matrizen **C4** und **C5** konvergiert auch das unpräkonditionierte CSYM-Verfahren relativ gut ($iter < n/2$ für **C4** bzw. $iter \approx n/2$ für **C5**). Die ungeblockte Jacobi-Präkonditionierung liefert in allen Fällen eine Verbesserung gegenüber dem unpräkonditionierten CSYM-Verfahren, die SGS-Präkonditionierung reduziert die Iterationszahlen für das Lösen der Gleichungssysteme mit **C1** bis **C4** effektiver. Für **C5** liefert die SGS-Päkonditionierung eine Verschlechterung gegenüber dem unpräkonditionierten und dem CSYM-Verfahren mit Jacobi-Präkonditionierung.

Die folgende Tabelle listet Werte des CSYM-Verfahrens mit Block-Präkonditionierung auf, mit denen bei mindestens einer der Faktorisierungen weniger Iterationen nötig sind als mit obigen Favoriten der CSYM-Verfahren mit ungeblockten Präkonditionierern ($xiter$). Es wurden jeweils die nach Umordnung mit TPABLO1 bzw. XPABLO bei gleichen Werten $minbs$ und $maxbs$ (bis 16) besten ermittelten Block-Präkonditionierer ausgewählt. Der L-Block-Präkonditionierer wurde nicht aufgeführt, wenn er mit dem R-Block-Präkonditionierer übereinstimmte, d.h. wenn kein (2×2)-Pivot bestimmt wurde.

Tabelle 6.15: Block-Präkonditionierung der Matrizen von CERFACS

C1, $xiter$ =830, TPABLO1, $minbs = 3$, $maxbs = 16$ γ_T=1.50, $bn = 180$, $min = 3$, $avg = 6.00$, $max = 16$					
Typ	S	$iter$	acc relres	tru relres	error
Jac	T	1000	2.139887e-06	2.461819e-06	1.977242e-05
Jac	R	1000	1.970678e-06	2.282711e-06	1.456523e-05
SGS	T	778	9.994910e-10	1.568976e-09	7.995340e-09
SGS	R	861	9.854068e-10	1.556492e-09	7.821060e-09

C2, $xiter = 933$, XPABLO, $minbs = 3$, $maxbs = 9$ $\gamma_T = 0.800$, $bn = 209$, $min = 3$, $avg = 6.22$, $max = 9$					
Typ	S	$iter$	acc relres	tru relres	error
Jac	T	1000	1.269207e-06	1.279102e-06	6.316146e-02
Jac	R	1000	2.689900e-06	3.211206e-06	1.484012e-01
Jac	L	1000	2.792015e-06	3.320562e-06	1.532766e-01
SGS	T	740	9.859299e-10	8.100653e-10	4.670049e-07
SGS	R	843	9.823368e-10	9.114632e-10	5.130619e-07
SGS	L	843	9.775627e-10	9.056072e-10	5.127753e-07

C3, $xiter = 1151$, TPABLO1, $minbs = 1$, $maxbs = 15$ $\gamma_T = 6.000$, $bn = 1514$, $min = 1$, $avg = 1.12$, $max = 14$					
Typ	S	iter	acc relres	tru relres	error
Jac	T	1500	6.396666e-07	7.340408e-07	1.514198e-03
Jac	R	1500	5.702158e-04	1.743515e-03	2.860051e+00
Jac	L	1500	6.540043e-07	7.502011e-07	1.537079e-03
SGS	T	811	9.922977e-10	8.246213e-10	1.214579e-06
SGS	R	1500	6.781239e-03	9.993876e-01	3.390605e+01
SGS	L	818	9.609105e-10	8.038005e-10	1.175430e-06

Trotz einer relativ kleinen durchschnittlichen Blockgröße (6, 6.22 und 1.12) konnte mit einer Block-Takagi-SGS-Präkonditionierung für **C1** eine Verbesserung um 9.6%, für **C2** eine Verbesserung um 21% und für **C3** eine Verbesserung um 30% gegenüber *xiter* erreicht werden. In den betrachteten Fällen ist das SGS-präkonditionierte CSYM-Verfahren mit Takagi-Faktorisierung denen mit anderer Blockzerlegung (R und L) überlegen. Dagegen konvergieren die CSYM-Verfahren mit Block-Jacobi-Präkonditionierung nicht innerhalb der vorgegebenen maximalen Anzahl von Iterationen. Für **C4** und **C5** konnte keine Blockeinteilung ermittelt werden, die zu einer ausreichenden Reduktion der Iterationszahlen gegenüber dem SGS-präkonditionierten CSYM-Verfahren führt.

Dünn besetzte Matrizen

Als nächstes testen wir das CSYM-Verfahren mit Block-Präkonditionierern für die drei dünn besetzten Matrizen *youngc*2, *youngc*3 und *youngc*4 (im Folgenden **Y2**, **Y3** und **Y4**) der Größe $n = 841$ aus der Harwell-Boeing Sammlung.
Auch hier wurde die rechte Seite der Gleichungssysteme bestimmt, indem die Matrix mit der Lösung, einem zufällig erzeugten Vektor, multipliziert wurde.

Die nächste Tabelle listet wieder die Iterationszahlen des unpräkonditionierten CSYM-Verfahrens und der CSYM-Verfahren mit den ungeblockten Präkonditionierern auf. Die Verfahren wurden beendet, sobald die Norm des akkumulierten relativen Residuums um mehr als 10^{-8} reduziert wurde oder die maximale Anzahl von 800 Iterationen erreicht wurde.

Tabelle 6.16: Ungeblockte Präkonditionierung der Young-Matrizen

Matrix	Typ	*iter*	acc relres	tru relres	error
Y2	–	800	4.585001e-08	4.585001e-08	1.196174e-05
	Jac	737	9.705238e-09	9.899616e-09	2.637682e-06
	SGS	800	2.023955e-04	7.731309e-03	1.369998e+00
Y3	–	800	2.116993e-05	2.116993e-05	3.400438e-01
	Jac	800	2.691623e-05	2.758404e-05	4.464024e-01
	SGS	800	7.976882e-06	2.180668e-03	1.269484e+00
Y4	–	800	3.320078e-05	3.320078e-05	2.235051e-02
	Jac	800	2.220222e-07	2.618266e-07	7.113665e-05
	SGS	800	5.233868e-08	8.671287e-01	1.765686e+01

Nur für **Y2** und **Y4** erhalten wir mit Jacobi-Präkonditionierung eine Verbesserung gegenüber dem unpräkonditionierten Verfahren, für **Y2** sogar eine Konvergenz nach weniger als 800 Iterationen. Mit SGS-Präkonditionierung erhalten wir dagegen stets eine Verschlechterung gegenüber dem unpräkonditionierten Verfahren (man beachte die Norm des Fehlers nach 800 Iterationen).

Zunächst ermitteln wir für **Y2** mit XPABLO Blockeinteilungen, so dass das CSYM-Verfahren mit Block-Präkonditionierung in weniger als $xiter = 737$ Iterationen konvergiert. Die besten Ergebnisse für jeweils $\gamma_T = 20,\ 40,\ 60,\ 80$ bis einschließlich $bsmax = 32$ mit Iterationszahlen, die sogar teilweise unter 500 liegen, sind im Folgenden aufgelistet.

Tabelle 6.17: CSYM-Verfahren mit Block-Präkonditionierern (**Y2**)

Y2, $xiter = 737$, XPABLO , $minbs = 1$, $maxbs = 30$					
$\gamma_T = 20.000$, $bn = 49$, $min = 1$, $avg = 17.16$, $max = 30$					
Typ	S	*iter*	acc relres	tru relres	error
Jac	T	503	9.857197e-09	9.911379e-09	1.827802e-06
Jac	R	800	1.861119e-03	8.398796e-03	1.205259e+00
Jac	L	800	4.772301e-03	6.969419e-02	6.254628e+00
Y2, $xiter = 737$, XPABLO , $minbs = 1$, $maxbs = 30$					
$\gamma_T = 40.000$, $bn = 59$, $min = 1$, $avg = 14.25$, $max = 30$					
Typ	S	*iter*	acc relres	tru relres	error
Jac	T	444	9.473258e-09	9.461840e-09	1.667497e-06
Jac	R	800	1.492660e-03	7.164173e-03	9.848039e-01
Jac	L	800	2.779645e-03	6.574301e-02	4.961302e+00
Y2, $xiter = 737$, XPABLO , $minbs = 1$, $maxbs = 32$					
$\gamma_T = 60.000$, $bn = 67$, $min = 1$, $avg = 12.55$, $max = 32$					
Typ	S	*iter*	acc relres	tru relres	error
Jac	T	429	9.892197e-09	9.644811e-09	1.658009e-06
Jac	R	800	2.088949e-03	1.165372e-02	1.512318e+00
Jac	L	800	3.836235e-03	7.110440e-02	6.017361e+00
$\gamma_T = 80.000$, $bn = 79$, $min = 1$, $avg = 10.65$, $max = 32$					
Typ	S	*iter*	acc relres	tru relres	error
Jac	T	461	9.846498e-09	9.708393e-09	1.654121e-06
Jac	R	800	1.766059e-03	9.499339e-03	1.360263e+00
Jac	L	800	5.329118e-03	9.826841e-02	7.960599e+00

Tabelle 6.18: CSYM-Verfahren mit Block-Präkonditionierern (**Y3**)

Y3, $xiter = 800$, XPABLO, $minbs = 1$, $maxbs = 50$ $\gamma_T = 128$, $bn = 122$, $min = 1$, $avg = 6.89$, $max = 50$					
Typ	S	*iter*	acc relres	tru relres	error
Jac	T	752	9.270488e-09	9.577498e-09	6.227676e-07
Jac	R	800	1.205406e-03	7.391138e-03	4.850496e+00
Jac	L	800	5.686563e-03	1.545219e-01	1.503812e+01
SGS	T	800	3.138825e-07	1.740965e-01	1.785229e+01
SGS	R	800	1.331888e-04	4.387783e-02	8.647837e+00
SGS	L	800	1.983264e-06	1.586233e+00	6.953067e+01

Für das Gleichungssystem mit **Y3** erhalten wir erst für eine relativ große Blockgröße (50) eine Konvergenz (und zwar nur mit Takagi-Präkonditionierung). Die folgende Abbildung zeigt den zugehörigen Verlauf der relativen Residuen.

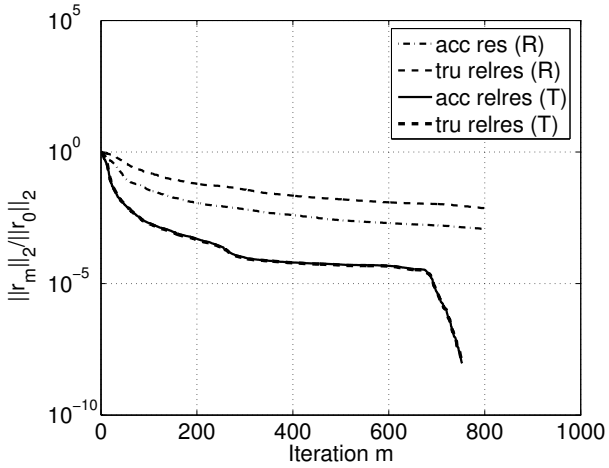


Tabelle 6.19: Eigenschaften des Block-Jacobi-Präkonditionierers (**Y3**)

6.89/ 0.00	$\hat{A}_T$	$\hat{A}_R$	$\hat{A}_L$
abw	5.341724e-15	9.811678e+00	6.111373e+01
mbe	2.442924e-15	5.741670e-01	2.765336e+01
cond $(\cdot)$	2.813223e+03	1.647254e+05	6.958886e+05
$\theta_{1/n}$	8.217600e-01	4.811733e+01	2.032734e+02
θ_1	1.346899e-02	3.185934e-01	1.743214e-01
θ_n	1.639041e-02	6.621178e-03	8.575712e-04
$\sigma_1(XX^H)$	3.581940e-01	1.505223e+00	5.566346e+01
$\sigma_n(XX^H)$	1.456992e-03	1.520925e-04	2.980140e-06
$\|\hat{A} - I\|_2$	9.255773e+00	2.257603e+02	1.230118e+02

Nur die (T)-Jacobi-präkonditionierte Matrix hat eine kleinere Kondition als die unpräkonditionierte Matrix ($\theta_{1/n} < 1$).

Tabelle 6.20: Vielfachheit der Singulärwerte der Jacobi-präkonditionierten Matrix (**Y3**)

m	A	$\hat{A}_T$	$\hat{A}_R$	$\hat{A}_L$
1	746	392	355	352
2	2	0	0	0
90	0	0	1	0
91	1	0	0	0
396	0	0	1	0
449	0	1	0	0
489	0	0	0	1

Die präkonditionierten Matrizen haben alle eine größere Anzahl mehrfacher Singulärwerte als die unpräkonditionierte Matrix. Obwohl sowohl die (L)- als auch die (R)-präkonditionierte Matrix die meisten vielfachen Singulärwerte haben, konvergiert das CSYM-Verfahren mit diesen Präkonditionierern auf Grund der schlechteren Kondition langsamer.

Als nächstes ermitteln wir für **Y4** mit XPABLO und den Standardwerten Blockeinteilungen, so dass das CSYM-Verfahren mit Block-Präkonditionierung innerhalb von 800 Iterationen konvergiert. Die besten Ergebnisse für jeweils $\gamma_T = 40,\ 60,\ 80$ bis einschließlich $bsmax = 32$, die im Folgenden aufgelistet sind, werden für $minbs = 3$ angenommen.

Tabelle 6.21: CSYM-Verfahren mit Block-Präkonditionierern (**Y4**)

Y4, $xiter = 800$, XPABLO, $minbs = 3$, $maxbs = 32$ $\gamma_T = 40.000$, $bn = 61$, $min = 3$, $avg = 13.79$, $max = 32$					
Typ	S	*iter*	acc relres	tru relres	error
Jac	T	599	9.906746e-09	8.822589e-09	2.544152e-06
Jac	R	800	1.357658e-04	4.984783e-04	8.189847e-02
Jac	L	800	3.969561e-03	5.726169e-02	4.973267e+00
Y4, $xiter = 800$, XPABLO, $minbs = 3$, $maxbs = 32$ $\gamma_T = 60.000$, $bn = 73$, $min = 1$, $avg = 11.52$, $max = 32$					
Typ	S	*iter*	acc relres	tru relres	error
Jac	T	644	9.817440e-09	9.743680e-09	2.910734e-06
Jac	R	800	6.188836e-04	2.885251e-03	4.462348e-01
Jac	L	800	4.736115e-03	8.007881e-02	5.852935e+00
Y4, $xiter = 800$, XPABLO, $minbs = 3$, $maxbs = 32$ $\gamma_T = 80.000$, $bn = 78$, $min = 1$, $avg = 10.78$, $max = 32$					
Typ	S	*iter*	acc relres	tru relres	error
Jac	T	623	9.841722e-09	1.021319e-08	3.012568e-06
Jac	R	800	5.665610e-04	2.676758e-03	4.028678e-01
Jac	L	800	5.044400e-03	8.513487e-02	6.293983e+00

Die CSYM-Verfahren mit Block-SGS-Präkonditionierung lieferten keine Konvergenz innerhalb von 800 Iterationen oder brachen auf Grund der Reduktion des präkonditionierten Residuums ab, ohne eine Reduktion des berechneten Residuums zu liefern.

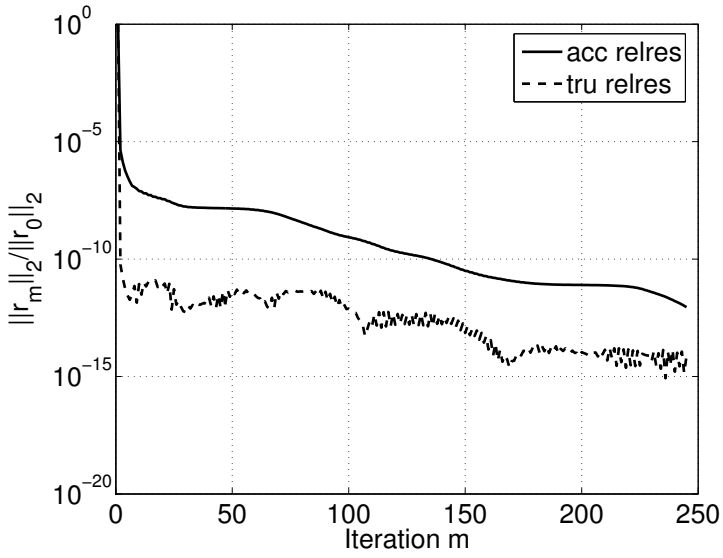
Weitere dünn besetzte Matrizen

Im Folgenden untersuchen wir die Konvergenz des CSYM-Verfahrens bei der Lösung von Gleichungssystemen aus dem Gebiet der Elektrodynamik aus der Diskretisierung mit FIT [33]. Wir werden uns auf drei Matrizen mit einer Dimension von $n > 10^4$ konzentrieren, bei Matrizen kleinerer Dimension erhalten wir ähnliche Ergebnisse. Die komplex symmetrischen Matrizen sind dünn besetzt, relativ schlecht konditioniert und die Beträge der Koeffizienten sind maximal 1. Darüber hinaus sind die Realteile der Eigenwerte alle positiv, zum Teil handelt es sich sogar um CSPD Matrizen.
Wir brechen die Verfahren ab, sobald die Norm des akkumulierten Residuums kleiner als 10^{-12} ist oder die maximale Anzahl der Iterationen ($maxiter$), die abhängig von der Größe n festgelegt wird, durchgeführt wurde.

Tabelle 6.22: Ergebnisse des CSYM-Verfahrens mit ungeblockter Präkonditionierung

Matrix	n	Typ	*iter*	acc relres	tru relres	error
Alt8	11781	–	659	9.994419e-13	9.989883e-13	2.117021e+05
		Jac	1504	9.792926e-13	3.497387e-15	7.294616e-01
		SGS	223	9.182531e-13	1.531143e-15	3.436887e-01
Alt4	12635	–	1202	9.990194e-13	1.000410e-12	1.224469e+05
		Jac	879	9.903551e-13	1.016023e-14	2.644096e-01
		SGS	244	9.253138e-13	7.582759e-15	5.688265e-01
Ex3	16588	–	1670	9.999611e-13	9.978956e-13	1.568090e+05
		Jac	1754	9.861451e-13	4.453444e-15	4.453701e-01
		SGS	449	9.662705e-13	8.566201e-15	1.578186e+00

Das unpräkonditionierte CSYM-Verfahren wird nach weniger als 700 bzw. weniger als 1700 Iterationen auf Grund der kleinen Norm des akkumulierten relativen Residuums beendet. Auffallend ist die noch große Norm des Fehlers der Größenordnung 10^5. Mit Jacobi- bzw. SGS-Präkonditionierung erhalten wir eine wesentlich stärkere Reduktion der Norm des Fehlers, bei der Jacobi-Präkonditionierung auch unter starker Erhöhung der Iterationszahl (von 659 auf 1504). Die Norm des akkumulierten relativen Residuums ist stets größer als die des berechneten. Das CSYM-Verfahren mit SGS-Präkonditionierung konvergiert bereits innerhalb von 500 bzw. 250 Iterationen. Die folgende Abbildung zeigt den zugehörigen Verlauf der relativen Residuen für **Alt4**.



Durch Block-SGS-Präkonditionierung kann die Iterationszahl weiter reduziert werden. Überraschenderweise liegt bei den hier untersuchten Matrizen der Fall vor, dass es kaum qualitative Unterschiede zwischen den gewählten Faktorisierungen der Blockdiagonalen gibt. Die Analyse der extremalen Singulärwerte der Faktoren zeigt, dass der maximale Singulärwert immer 1 und die kleinsten Singulärwerte nahezu identisch sind. Die Blockdiagonale M weist jedoch keine Besonderheiten hinsichtlich der Eigenwerte auf, da sie sowohl Eigenwerte mit positivem als auch mit negativem Realteil hat. Vermutlich ist diese Eigenschaft auf die Besonderheit der Matrizen zurückzuführen, dass die Eigenwerte nur positiven Realteil haben, d.h. positiv stabil sind [33].

Tabelle 6.23: CSYM-Verfahren mit Block-Präkonditionierern (**Alt8**)

$xiter = 223$, XPABLO, $minbs = 3$, $maxbs = 32$ $\gamma_T = 0.50$, $bn = 3927$, $min = 3.00$, $avg = 3.00$, $max = 3.00$					
Typ	S	*iter*	acc relres	tru relres	error
SGS	T	194	9.883266e-13	4.081858e-17	4.014611e-01
SGS	R	193	9.993613e-13	4.479715e-16	4.104614e-01
$xiter = 223$, XPABLO, $minbs = 6$, $maxbs = 16$ $\gamma_T = 0.50$, $bn = 1964$, $min = 3$, $avg = 6.00$, $max = 6$					
Typ	S	*iter*	acc relres	tru relres	error
SGS	T	183	9.863719e-13	1.543797e-17	3.747222e-01
SGS	R	183	8.913438e-13	7.590961e-16	3.332905e-01
$xiter = 223$, XPABLO, $minbs = 9$, $maxbs = 16$ $\gamma_T = 0.50$, $bn = 1309$, $min = 9$, $avg = 9.00$, $max = 9$					
Typ	S	*iter*	acc relres	tru relres	error
SGS	T	177	9.724651e-13	3.411713e-17	3.909462e-01
SGS	R	176	9.937468e-13	4.556003e-16	3.952333e-01
$xiter = 223$, XPABLO, $minbs = 12$, $maxbs = 16$ $\gamma_T = 0.50$, $bn = 982$, $min = 9$, $avg = 12.00$, $max = 12$					
Typ	S	*iter*	acc relres	tru relres	error
SGS	T	175	8.166226e-13	6.070836e-16	3.018517e-01
SGS	R	173	9.745868e-13	4.547991e-16	3.501752e-01

Tabelle 6.24: CSYM-Verfahren mit Block-Präkonditionierern (**Alt4**)

$xiter = 244$, XPABLO, $minbs = 3$, $maxbs = 16$ γ_T= 0.50, $bn = 4212$, $min = 2.00$, $avg = 3.00$, $max = 3.00$					
Typ	S	*iter*	acc relres	tru relres	error
SGS	T	205	9.793308e-13	1.061516e-15	3.496493e-01
SGS	R	206	9.674812e-13	3.042956e-16	3.222892e-01
$xiter = 244$, XPABLO, $minbs = 6$, $maxbs = 16$ $\gamma_T = 0.50$, $bn = 2106$, $min = 5$, $avg = 6.00$, $max = 6$					
Typ	S	*iter*	acc relres	tru relres	error
SGS	T	191	8.594898e-13	1.221992e-15	2.465129e-01
SGS	R	190	9.934150e-13	1.348938e-15	3.404366e-01
$xiter = 244$, XPABLO , $minbs = 9$, $maxbs = 16$ $\gamma_T = 0.500$, $bn = 1404$, $min = 8$, $avg = 9.00$, $max = 9$					
Typ	S	*iter*	acc relres	tru relres	error
SGS	T	196	9.823089e-13	1.770706e-15	2.931990e-01
SGS	R	185	9.150401e-13	5.012928e-16	3.174184e-01
$xiter = 244$, XPABLO, $minbs = 12$, $maxbs = 16$ $\gamma_T = 0.500$, $bn = 1053$, $min = 11$, $avg = 12.00$, $max = 12$					
Typ	S	*iter*	acc relres	tru relres	error
SGS	T	193	8.889664e-13	1.108140e-15	3.184680e-01
SGS	R	183	9.689548e-13	1.910382e-15	3.144576e-01

Tabelle 6.25: CSYM-Verfahren mit Block-Präkonditionierern (**Ex3**)

$xiter = 449$, XPABLO, $minbs = 3$, $maxbs = 16$ $\gamma_T = 0.50$, $bn = 5530$, $min = 1$, $avg = 3.00$, $max = 3$					
Typ	S	*iter*	acc relres	tru relres	error
SGS	T	404	9.987190e-13	2.966426e-15	9.420502e-01
SGS	R	404	9.474097e-13	9.598020e-16	8.675857e-01
$xiter = 449$, XPABLO, $minbs = 6$, $maxbs = 16$ $\gamma_T = 0.50$, $bn = 2765$, $min = 4$, $avg = 6.00$, $max = 6$					
Typ	S	*iter*	acc relres	tru relres	error
SGS	T	393	9.644440e-13	1.994019e-15	6.870221e-01
SGS	R	392	9.107212e-13	8.190863e-16	6.273635e-01
$xiter = 449$, XPABLO, $minbs = 9$, $maxbs = 16$ $\gamma_T = 0.50$, $bn = 1844$, $min = 1$, $avg = 9.00$, $max = 9$					
Typ	S	*iter*	acc relres	tru relres	error
SGS	T	390	9.976364e-13	1.522470e-16	5.957424e-01
SGS	R	385	9.571538e-13	8.169828e-16	3.946797e-01
$xiter = 449$, XPABLO, $minbs = 12$, $maxbs = 16$ $\gamma_T = 0.50$, $bn = 1383$, $min = 4$, $avg = 11.99$, $max = 12$					
Typ	S	*iter*	acc relres	tru relres	error
SGS	T	384	9.863125e-13	6.655348e-16	8.744018e-01
SGS	R	383	9.880437e-13	4.667517e-15	8.880145e-01

Die mit wenig Aufwand zu berechnende Präkonditionierung mit einer durchschnittlichen Blockgröße von 3 liefert nur eine Reduktion um 10% gegenüber der Iterationszahl der ungeblockten SGS-Präkonditionierung. Die größer gewählten minimalen Blockgrößen führen nur zu einer Reduktion um weitere 5%. Der zusätzliche Aufwand rentiert sich also nicht.

6.4 Zusammenfassung

Wir haben das CSYM-Verfahren mit verschiedenen Block-Präkonditionierungen (Blockeinteilungen, Block-Jacobi- und Block-SGS) getestet, basierend auf unterschiedlichen Zerlegungen der Blockdiagonalen. Bei der Erklärung der Ergebnisse haben wir sowohl den Aspekt der Reduktion der Kondition als auch die Erhöhung der Mehrfachheit der Singulärwerte der präkonditionierten Matrix berücksichtigt, die nach Satz 3.39 bzw. Satz 3.38 beide das Konvergenzverhalten des CSYM-Verfahrens bestimmen.
Für die Block-Jacobi-Präkonditionierung mit Takagi-Faktorisierung ist die Minimaler-Bereich-Eigenschaft in exakter Arithmetik garantiert. Diese ist aber in der Praxis nicht immer ausreichend erfüllt, insbesondere hängt auch hier die Qualität der Berechnung von der Anordnung der Unbekannten ab. Ist die Minimaler-Bereich-Eigenschaft erfüllt, d.h. bei sehr kleinen Werten von *mbe*, so ist dies in den untersuchten Fällen ein hinreichendes Kriterium für die Reduktion der Kondition der Matrix durch Präkonditionierung. Die Präkonditionierer basierend auf einer Takagi-Faktorisierung sind den entsprechenden Varianten basierend auf einer Cholesky-Zerlegung überlegen, soweit eine Konvergenzverbesserung durch eine Block-Jacobi-Präkonditionierung erreicht wurde (s. Ergebnisse für **A1**, **Y2**, **Y3** und **Y4**). Die Faktorisierungen der Blockdiagonalen der betrachteten positiv stabilen Matrizen dagegen unterscheiden sich unwesentlich hinsichtlich der Qualität. Wir nehmen an, dass die Singulärwerte der Matrix mit Block-Takagi-Präkonditionierung die kleinste quadratische Abweichung von den Beträgen der Eigenwerte haben. Selbst dann liefert dies aber keine Begründung dafür, dass die so präkonditionierte Matrix eine günstigere Singulärwertverteilung habe sollte als bei einer anderen Faktorisierung. Die Takagi-Faktorisierung hat auch die Besonderheit, dass alle Singulärwerte des unfaktorisierten Präkonditionierers M mit den Singulärwerten der hermiteschen Matrix $\tilde{M} = SS^H$ übereinstimmen. Für eine größere Mehrfachheit der Singulärwerte bei Takagi-Präkonditionierung haben wir auch praktisch keine Anhaltspunkte gefunden.
Für die Block-SGS-Präkonditionierung mit Takagi-Faktorisierung ist die Minimaler-Bereich-Eigenschaft nicht garantiert. In numerischen Experimenten, in denen die Zerlegung der Blockdiagonalen eine ausgeprägte Minimaler-Bereich-Eigenschaft hat, liefert die SGS-Präkonditionierung bessere Ergebnisse als entsprechende Präkonditionierer auf Basis von Faktorisierungen, wo dies nicht erfüllt ist (s. Ergebnisse zu **C1**, **C2** und **C3**). Selbst wenn die Kondition sich erhöht, erhalten wir in Einzelfällen eine schnellere Konvergenz (**A1**). Hier scheinen andere, noch nicht untersuchte numerische Kriterien einzugehen, zum Beispiel zusätzliche Cluster von Singulärwerten. Eine Reduktion der Kondition scheint also hinreichend aber nicht notwendig für die schnellere Konvergenz des CSYM-Verfahrens zu sein.

Es stellen sich noch folgende Fragen:

- Wann ist eine Block-SGS- besser als die entsprechende Block-Jacobi-Präkonditionierung? Offensichtlich erhalten wir trotz erhöhter Kondition der präkonditionierten Matrix bei der SGS- eine Verbesserung gegenüber der Jacobi-Präkonditionierung.
- Welches sind die besten Parameter für XPABLO bzw. TPABLO, um für eine Präkonditionierung günstige Blöcke zu erhalten? Wann ist welches γ und *minbs* bzw. *maxbs* sinnvoll?
- In welchen Fällen erfüllt eine Cholesky-Zerlegung der Blockdiagonalen die Minimaler-Bereich-Eigenschaft so gut, dass auf die aufwändigere Berechnung des Takagi-Faktors verzichtet werden kann? Dies ist für die getesteten CSPD Matrizen der Fall.
- Welches ist die beste Anordnung der Unbekannten um eine Takagi-Faktorisierung stabil berechnen zu können?
- Kann mit weniger Aufwand ein guter hermitesch positiv definiter Block-Präkonditionierer bestimmt werden? Erste numerische Ergebnisse mit Blockeinteilungen von AA^H liefern nicht zufrieden stellende Ergebnisse.

Kapitel 7

Zusammenfassung und Ausblick

Bei der Präkonditionierung des CSYM-Verfahrens spielt die Faktorisierung des Präkonditionierers $M = SS^T$ eine entscheidende Rolle. Ein Takagi-Faktor S zeichnet sich unter den möglichen Faktorisierungen durch die Minimaler-Bereich-Eigenschaft aus. Darüber hinaus liefert er die „günstigsten“ Singulärwerte der präkonditionierten Matrix bei festem Präkonditionierer M, insofern, als die Singulärwerte von M mit denen von $\tilde{M} = SS^H$ übereinstimmen. Wir haben folgende zwei Anwendungen von Takagi-faktorisierten Präkonditionierern betrachtet.

Die erste Anwendung, das CSYM-Verfahren mit Deflation, ist nur in Betracht zu ziehen, wenn das Gleichungssystem mit verschiedenen rechten Seiten gelöst werden soll. Bei der Deflation werden einige Singulärpaare (Singulärwerte und zugehörige -vektoren) berechnet, wobei die Berechnung der Singulärpaare aus der Tridiagonalisierung des CSYM-Verfahrens nur bedingt sinnvoll ist. Es besteht eine starke Abhängigkeit vom Startvektor, zusätzliche Reorthogonalisierungen bringen nicht den gewünschten Erfolg und nur Singulärpaare zu großen Singulärwerten werden gut approximiert. Der Aufwand für die Berechnung ist im Vergleich zur Qualität der Präkonditionierer, d.h. der Zahl an gesparten Iterationsschritten, zu hoch. Die Deflation kann mit geringem zusätzlichem Rechenaufwand angewendet werden (Nutzung der Projektion). Sie liefert jedoch nicht immer eine Verbesserung. Das CSYM-Verfahren ohne Deflation mit projiziertem Startvektor kann besser sein. Das Konvergenzverhalten hängt stark vom Startvektor (s. Satz 5.6) ab, daher sollte immer der projizierte Vektor als Startvektor gewählt werden.
Eine Block-Präkonditionierung basierend auf Takagi-Faktoren ist günstig bei Matrizen mit relativ wenigen betragsmäßig großen Einträgen, die alle in Diagonalblöcke permutiert werden können (TPABLO1). Die Anordnung der Unbekannten spielt für die Takagi-Präkonditionierung wie auch bei Präkonditionierern, die auf Cholesky-Zerlegungen basieren, für die Qualität des berechneten Block-Jacobi-Präkonditionierers eine große Rolle. Sie bestimmt, wie gut die Minimaler-Bereich-Eigenschaft erfüllt ist und wie gut die Faktorisierung den unfaktorisierten Präkonditionierer approximiert. Es kann vorkommen,

dass ein auf Cholesky-Faktorisierung basierender Präkonditionierer qualitativ vergleichbar oder besser ist als ein auf Takagi-Faktorisierung basierender. Ist dies nicht der Fall, so lohnt sich die aufwändigere Berechnung des Takagi-Faktors. Der Aufwand relativiert sich auch hier insbesondere, wenn das Gleichungssystem mit verschiedenen rechten Seiten gelöst werden soll. Ein Block-Takagi-Präkonditionierer liefert i.A. bei ausreichender Blockgröße eine Reduktion der Kondition gegenüber der unpräkonditionierten Matrix und gegenüber den Block-Cholesky-Präkonditionierern, auch mit Rook-Pivotisierung. Eine Block-SGS-Präkonditionierung kann sich lohnen und mit Eisenstat-Trick ohne zusätzlichen Aufwand im Vergleich zur Block-Jacobi-Präkonditionierung angewendet werden.

Als Anregung stellen wir einige, noch ungeklärte Fragen:

- Ist eine Spezialisierung von Satz 4.11 möglich (schwächere Bedingungen als in Satz 5.2)?
- Sind noch genauere Vorhersagen über die θ_k-Werte aus Satz 4.11 möglich?
- Kann eine Faktorisierung umgangen werden, indem gleich die hermitesch positiv definite Matrix zu A aus der Polarzerlegung approximiert wird?
- Ist eine schnelle und gute Approximation von kleineren Singulärwerten möglich (z.B. über Restarts und Shifts s. [3], [38], [4])?
- Wie kann der Aufwand zur Berechnung/Anwendung des Takagi-Präkonditionierers weiter reduziert werden?
- Wann lohnt sich ein Block-SGS-Präkonditionierer?

Anhang A

Tabellen zu den numerischen Ergebnissen

Das CSYM-Verfahren wurde beendet, sobald das relative akkumulierte Residuum bezüglich des präkonditionierten Systems kleiner als die Genauigkeit (*tol*) war oder die maximale Anzahl der Iterationen (*maxiter*) durchgeführt wurde. Die folgenden Tabelle listet die Werte der in Kapitel 5.5 und Kapitel 6.3 betrachteten Matrizen auf.

Tabelle A.1: Werte im Abbruchkriterium

Matrix	n	tol	$maxiter$
A1	1176	10^{-9}	1000
C1	1080	10^{-9}	1000
C2	1299	10^{-9}	1000
C3	1701	10^{-9}	1500
C4	2016	10^{-9}	2000
C5	2430	10^{-9}	2400
Y2	841	10^{-8}	800
Y3	841	10^{-8}	800
Y4	841	10^{-8}	800
Alt8	11781	10^{-12}	10000
Alt4	12635	10^{-12}	10000
Ex3	16588	10^{-12}	15000

A.1 Zu Kapitel 5.5

A.1.1 Deflation mit vorberechneten Singulärpaaren

Die folgenden sechs Tabellen A.3 - A.8 enthalten Werte des CSYM-Verfahrens für verschiedene Startvektoren mit Deflation und ohne. Die erste Zeile enthält zum Vergleich jeweils die Werte des unpräkonditionierten Verfahrens mit dem Nullvektor als Startvektor.

- Die Spalte D gibt an, ob ohne (–) oder mit Deflation (D) gerechnet wurde.
- Wurde der Präkonditionierer nicht in Takagi-faktorisierter Form angewendet, so steht in Spalte D (L) für Block-Cholesky-Zerlegung bzw. (R) für die Zerlegung mit Rook-Pivotisierung des Präkonditionierers.
- Die Spalte k enthält für k Singulärpaare zu größten Singulärwerten den positiven Wert k bzw. zu kleinsten Singulärwerten den negativen Wert $-k$.

A.1.2 Deflation mit Singulärpaaren aus CSYM

Hierbei enthält die zusätzliche Spalte m die Anzahl der durchgeführten Schritte des CSYM-Verfahrens, aus dem die Takagi-Faktoren ermittelt wurden. Jeweils drei Tabellen (ohne Reorthogonalisierung, CR, PR) enthalten die Werte für die zwei betrachteten Fälle

- A.9 - A.11 bei gleicher rechter Seite b,
- A.12 - A.14 bei verschiedenen rechten Seiten.

Tabelle A.2: **A1**: Anzahl der ermittelten Singulärpaare nach m Schritten,
b Nullvektor bis auf eine Komponente (oben), Zufallsvektor (unten)

m	70	80	90	100	110	120	130	140	150	160	170	180	190	200
–	3	5	6	7	7	9	12	18	19	19	22	23	25	27
CR	3	5	6	7	7	12	15	15	17	21	23	30	33	33
PR	3	5	6	7	7	12	15	15	17	21	23	30	33	33
–	5	7	7	7	9	13	19	19	24	25	27	39	41	43
CR	5	7	7	8	13	16	15	20	21	32	33	34	38	41
PR	5	7	7	8	13	16	15	20	21	32	33	34	38	41

Tabelle A.3: **A1**: CSYM-Verfahren mit Singulärpaaren zu kleinsten Singulärwerten (*svd.m*)

D	k	x_0	*iter*	acc relres	tru relres	error
–	–	$\mathbf{x_0}$	778	9.828176e-10	9.828174e-10	7.973643e-09
D	-40	$\mathbf{x_0}$	418	8.366723e-10	8.052429e-10	2.355993e-09
D	-40	$\mathbf{x_p}$	314	9.669014e-10	2.862389e-09	1.759925e-08
–	-40	$\mathbf{x_p}$	308	9.765935e-10	9.765929e-10	3.955983e-09
D	-35	$\mathbf{x_0}$	458	9.518013e-10	9.221775e-10	2.970768e-09
D	-35	$\mathbf{x_p}$	357	9.928731e-10	2.379389e-09	1.592902e-08
–	-35	$\mathbf{x_p}$	355	9.387305e-10	9.387303e-10	3.700946e-09
D	-30	$\mathbf{x_0}$	510	9.462893e-10	9.191236e-10	3.372200e-09
D	-30	$\mathbf{x_p}$	392	9.528371e-10	2.271001e-09	1.538209e-08
–	-30	$\mathbf{x_p}$	388	9.339044e-10	9.339044e-10	3.878751e-09
D	-25	$\mathbf{x_0}$	561	9.649587e-10	9.457441e-10	3.387307e-09
D	-25	$\mathbf{x_p}$	436	9.830926e-10	1.403318e-09	1.004597e-08
–	-25	$\mathbf{x_p}$	437	9.481316e-10	9.481312e-10	3.954031e-09
D	-20	$\mathbf{x_0}$	613	9.812746e-10	9.627126e-10	3.704960e-09
D	-20	$\mathbf{x_p}$	474	9.674736e-10	1.310409e-09	9.836494e-09
–	-20	$\mathbf{x_p}$	474	9.901974e-10	9.901973e-10	5.396382e-09
D	-15	$\mathbf{x_0}$	682	9.699509e-10	9.544855e-10	4.995242e-09
D	-15	$\mathbf{x_p}$	526	9.875827e-10	1.032229e-09	6.784257e-09
–	-15	$\mathbf{x_p}$	527	9.243406e-10	9.243405e-10	4.383651e-09
D	-10	$\mathbf{x_0}$	736	9.922718e-10	9.811002e-10	5.307193e-09
D	-10	$\mathbf{x_p}$	569	9.957306e-10	1.036649e-09	7.278689e-09
–	-10	$\mathbf{x_p}$	569	9.925755e-10	9.925757e-10	5.338709e-09
D	-5	$\mathbf{x_0}$	836	9.880051e-10	9.793973e-10	5.839055e-09
D	-5	$\mathbf{x_p}$	660	9.755250e-10	1.016668e-09	8.274244e-09
–	-5	$\mathbf{x_p}$	659	9.896864e-10	9.896862e-10	6.802576e-09

Das CSYM-Verfahren konvergiert in allen Fällen innerhalb von 1000 Iterationen, lediglich das CSYM-Verfahren mit Deflation aus 5 Singulärpaaren zu kleinsten Singulärpaaren mit Startvektor $\mathbf{x_0}$ konvergiert langsamer als das unpräkonditionierte Verfahren.

Tabelle A.4: **A1**: CSYM-Verfahren mit Singulärpaaren zu größten Singulärwerten (*svd.m*)

D	k	x_0	*iter*	acc relres	tru relres	error
–	–	$\mathbf{x_0}$	778	9.828176e-10	9.828174e-10	7.973643e-09
D	5	$\mathbf{x_0}$	935	9.883414e-10	9.880716e-10	6.543609e-09
D	5	$\mathbf{x_p}$	753	9.729399e-10	9.984001e-10	7.921157e-09
–	5	$\mathbf{x_p}$	777	9.821464e-10	9.821463e-10	8.043341e-09
D	10	$\mathbf{x_0}$	916	9.510845e-10	9.505754e-10	6.253492e-09
D	10	$\mathbf{x_p}$	736	9.517199e-10	2.532014e-09	7.675830e-09
–	10	$\mathbf{x_p}$	776	9.942294e-10	9.942296e-10	8.059693e-09
D	15	$\mathbf{x_0}$	898	9.794002e-10	9.740436e-10	6.424760e-09
D	15	$\mathbf{x_p}$	722	9.877108e-10	3.087663e-09	7.903020e-09
–	15	$\mathbf{x_p}$	776	9.994986e-10	9.994983e-10	8.093483e-09
D	20	$\mathbf{x_0}$	885	9.971029e-10	9.894035e-10	6.558996e-09
D	20	$\mathbf{x_p}$	709	9.987364e-10	3.102260e-09	8.042408e-09
–	20	$\mathbf{x_p}$	774	9.887805e-10	9.887810e-10	7.954242e-09
L	20	$\mathbf{x_0}$	1000	4.398637e-02	1.034410e+00	3.232195e+00
L	20	$\mathbf{x_p}$	1000	1.741944e-02	2.993817e-01	7.378206e+01
R	20	$\mathbf{x_0}$	1000	2.580601e-02	4.465343e-01	1.574076e+00
R	20	$\mathbf{x_p}$	1000	6.068937e-03	2.041903e-01	2.410583e+01
D	25	$\mathbf{x_0}$	873	9.983347e-10	9.900151e-10	6.573274e-09
D	25	$\mathbf{x_p}$	698	9.669074e-10	3.280043e-09	7.748474e-09
–	25	$\mathbf{x_p}$	774	9.732456e-10	9.732458e-10	7.819123e-09
D	30	$\mathbf{x_0}$	856	9.930121e-10	9.818521e-10	6.475635e-09
D	30	$\mathbf{x_p}$	684	9.807763e-10	3.302412e-09	7.801116e-09
–	30	$\mathbf{x_p}$	773	9.773029e-10	9.773028e-10	7.906856e-09
D	35	$\mathbf{x_0}$	846	9.748103e-10	9.628172e-10	6.322233e-09
D	35	$\mathbf{x_p}$	675	9.840354e-10	3.309447e-09	7.878105e-09
–	35	$\mathbf{x_p}$	770	9.960909e-10	9.960911e-10	7.973066e-09
D	40	$\mathbf{x_0}$	832	9.917294e-10	9.781215e-10	6.446514e-09
D	40	$\mathbf{x_p}$	663	9.837450e-10	3.380077e-09	7.895450e-09
–	40	$\mathbf{x_p}$	772	9.950652e-10	9.950652e-10	7.973940e-09

Tabelle A.5: **Y2**: CSYM-Verfahren mit Singulärpaaren zu kleinsten Singulärwerten (*svd.m*)

D	k	x_0	*iter*	acc relres	tru relres	error
–	–	$\mathbf{x_0}$	800	4.834961e-08	4.834961e-08	1.261569e-05
D	-40	$\mathbf{x_0}$	677	9.880589e-09	9.880219e-09	1.521966e-06
D	-40	$\mathbf{x_p}$	467	9.662771e-09	9.662771e-09	1.385392e-06
–	-40	$\mathbf{x_p}$	466	9.709502e-09	9.709502e-09	1.406220e-06
D	-35	$\mathbf{x_0}$	709	9.759201e-09	9.758867e-09	1.534878e-06
D	-35	$\mathbf{x_p}$	487	9.603846e-09	9.603846e-09	1.476135e-06
–	-35	$\mathbf{x_p}$	487	9.804448e-09	9.804448e-09	1.494576e-06
D	-30	$\mathbf{x_0}$	768	9.931986e-09	9.931792e-09	1.508457e-06
D	-30	$\mathbf{x_p}$	541	9.824448e-09	9.824448e-09	1.705878e-06
–	-30	$\mathbf{x_p}$	542	9.841021e-09	9.841021e-09	1.698720e-06
D	-25	$\mathbf{x_0}$	800	1.273046e-08	1.273022e-08	2.206343e-06
D	-25	$\mathbf{x_p}$	567	9.807537e-09	9.807537e-09	1.606975e-06
–	-25	$\mathbf{x_p}$	567	9.737626e-09	9.737626e-09	1.597015e-06
D	-20	$\mathbf{x_0}$	800	3.893999e-08	3.893965e-08	7.552572e-06
D	-20	$\mathbf{x_p}$	600	9.983445e-09	9.983445e-09	2.005688e-06
–	-20	$\mathbf{x_p}$	600	9.991662e-09	9.991662e-09	2.007725e-06
D	-15	$\mathbf{x_0}$	800	1.064051e-07	1.064043e-07	2.125901e-05
D	-15	$\mathbf{x_p}$	621	9.734875e-09	9.734875e-09	1.823970e-06
–	-15	$\mathbf{x_p}$	621	9.835567e-09	9.835567e-09	1.850869e-06
D	-10	$\mathbf{x_0}$	800	2.916776e-07	2.916775e-07	5.842165e-05
D	-10	$\mathbf{x_p}$	663	9.793794e-09	9.793794e-09	2.133133e-06
–	-10	$\mathbf{x_p}$	665	9.774681e-09	9.774681e-09	2.126700e-06
D	-5	$\mathbf{x_0}$	800	2.119146e-06	2.119890e-06	5.049684e-04
D	-5	$\mathbf{x_p}$	745	9.851539e-09	9.851539e-09	2.208920e-06
–	-5	$\mathbf{x_p}$	743	9.971961e-09	9.971962e-09	2.237054e-06

Bereits bei 5 Singulärpaaren zu kleinsten Singulärwerten erhalten wir eine Konvergenz innerhalb von 800 Iterationen für Startvektor $\mathbf{x_p}$.

Tabelle A.6: **Y2**: CSYM-Verfahren mit Singulärpaaren zu größten Singulärwerten (*svd.m*)

D	k	x_0	*iter*	acc relres	tru relres	error
–	–	$\mathbf{x_0}$	800	4.834961e-08	4.834961e-08	1.261569e-05
D	5	$\mathbf{x_0}$	800	1.648696e-05	1.942429e-04	4.734059e-03
D	5	$\mathbf{x_p}$	800	3.539771e-08	3.539771e-08	9.315725e-06
–	5	$\mathbf{x_p}$	800	4.833785e-08	4.833785e-08	1.254277e-05
D	10	$\mathbf{x_0}$	800	1.307342e-05	9.292528e-05	3.743394e-03
D	10	$\mathbf{x_p}$	800	2.739344e-08	2.739344e-08	7.347594e-06
–	10	$\mathbf{x_p}$	800	4.666529e-08	4.666529e-08	1.204028e-05
D	15	$\mathbf{x_0}$	800	1.072907e-05	5.904197e-05	2.995728e-03
D	15	$\mathbf{x_p}$	800	2.093450e-08	2.093449e-08	5.683311e-06
–	15	$\mathbf{x_p}$	800	4.533648e-08	4.533648e-08	1.163812e-05
D	20	$\mathbf{x_0}$	800	8.711442e-06	3.648110e-05	2.398638e-03
D	20	$\mathbf{x_p}$	800	1.596806e-08	1.596806e-08	4.454498e-06
–	20	$\mathbf{x_p}$	800	4.688039e-08	4.688039e-08	1.201322e-05
D	25	$\mathbf{x_0}$	800	6.572574e-06	1.749096e-05	1.758421e-03
D	25	$\mathbf{x_p}$	800	1.247033e-08	1.247033e-08	3.574789e-06
–	25	$\mathbf{x_p}$	800	4.526531e-08	4.526531e-08	1.152127e-05
D	30	$\mathbf{x_0}$	800	5.222869e-06	9.351201e-06	1.342315e-03
D	30	$\mathbf{x_p}$	800	1.017535e-08	1.017535e-08	2.937229e-06
–	30	$\mathbf{x_p}$	800	4.460588e-08	4.460588e-08	1.130441e-05
D	35	$\mathbf{x_0}$	800	3.564914e-06	4.481929e-06	8.829489e-04
D	35	$\mathbf{x_p}$	787	9.822688e-09	9.822688e-09	2.810938e-06
–	35	$\mathbf{x_p}$	800	4.312191e-08	4.312191e-08	1.091451e-05
D	40	$\mathbf{x_0}$	800	2.606514e-06	2.744987e-06	6.665933e-04
D	40	$\mathbf{x_p}$	774	9.982061e-09	9.982061e-09	2.848234e-06
–	40	$\mathbf{x_p}$	800	4.493447e-08	4.493447e-08	1.129484e-05

Erst bei 35 Singulärpaaren zu größten Singulärwerten erhalten wir eine Konvergenz innerhalb von 800 Iterationen für Startvektor $\mathbf{x_p}$.

Tabelle A.7: **C2**: CSYM-Verfahren mit Singulärpaaren zu kleinsten Singulärwerten (*svd.m*)

D	k	x_0	*iter*	acc relres	tru relres	error
–	–	$\mathbf{x_0}$	1000	1.004092e-06	1.004092e-06	3.336948e-02
D	-40	$\mathbf{x_0}$	933	9.879881e-10	9.880370e-10	6.212766e-07
D	-40	$\mathbf{x_p}$	927	9.748872e-10	9.748872e-10	6.044106e-07
–	-40	$\mathbf{x_p}$	928	9.843306e-10	9.843306e-10	6.111813e-07
D	-35	$\mathbf{x_0}$	954	9.959169e-10	9.959595e-10	6.292694e-07
D	-35	$\mathbf{x_p}$	949	9.617176e-10	9.617177e-10	6.037919e-07
–	-35	$\mathbf{x_p}$	949	9.256889e-10	9.256889e-10	5.829235e-07
D	-30	$\mathbf{x_0}$	973	9.632621e-10	9.632928e-10	6.530611e-07
D	-30	$\mathbf{x_p}$	969	9.993884e-10	9.993882e-10	6.765661e-07
–	-30	$\mathbf{x_p}$	970	9.296926e-10	9.296926e-10	6.269317e-07
D	-25	$\mathbf{x_0}$	993	9.898665e-10	9.898945e-10	6.684393e-07
D	-25	$\mathbf{x_p}$	989	9.431045e-10	9.431045e-10	6.388669e-07
–	-25	$\mathbf{x_p}$	989	9.859343e-10	9.859343e-10	6.658910e-07
D	-20	$\mathbf{x_0}$	1000	1.283554e-09	1.283575e-09	9.168168e-07
D	-20	$\mathbf{x_p}$	1000	1.226199e-09	1.226199e-09	8.673054e-07
–	-20	$\mathbf{x_p}$	1000	1.244817e-09	1.244817e-09	8.832402e-07
D	-15	$\mathbf{x_0}$	1000	2.530868e-09	2.530900e-09	1.844624e-06
D	-15	$\mathbf{x_p}$	1000	2.407381e-09	2.407381e-09	1.746181e-06
–	-15	$\mathbf{x_p}$	1000	2.422360e-09	2.422360e-09	1.762750e-06
D	-10	$\mathbf{x_0}$	1000	3.737312e-09	3.737342e-09	2.950914e-06
D	-10	$\mathbf{x_p}$	1000	3.561126e-09	3.561126e-09	2.799111e-06
–	-10	$\mathbf{x_p}$	1000	3.653290e-09	3.653290e-09	2.884091e-06
D	-5	$\mathbf{x_0}$	1000	7.368231e-09	7.368258e-09	5.995098e-06
D	-5	$\mathbf{x_p}$	1000	7.097330e-09	7.097330e-09	5.715169e-06
–	-5	$\mathbf{x_p}$	1000	7.374797e-09	7.374798e-09	5.985632e-06

Erst bei 25 Singulärpaaren zu kleinsten Singulärwerten erhalten wir eine Konvergenz innerhalb von 1000 Iterationen für Startvektor $\mathbf{x_p}$. Die Wahl des Startvektors beeinflusst die Iterationszahlen kaum, d.h. wir erhalten nahezu vergleichbare Ergebnisse für Startvektor $\mathbf{x_0}$.

Tabelle A.8: **C2**: CSYM-Verfahren mit Singulärpaaren zu größten Singulärwerten (*svd.m*)

D	k	x_0	*iter*	acc relres	tru relres	error
–	–	$\mathbf{x_0}$	1000	1.004092e-06	1.004092e-06	3.336948e-02
D	5	$\mathbf{x_0}$	1000	3.651223e-07	3.618032e-07	4.340865e-03
D	5	$\mathbf{x_p}$	1000	3.638633e-07	3.646894e-07	4.297435e-03
–	5	$\mathbf{x_p}$	1000	9.908821e-07	9.908821e-07	3.182932e-02
D	10	$\mathbf{x_0}$	1000	1.414564e-07	1.396415e-07	6.546267e-04
D	10	$\mathbf{x_p}$	1000	1.451011e-07	1.451025e-07	6.846191e-04
–	10	$\mathbf{x_p}$	1000	9.882049e-07	9.882049e-07	3.126711e-02
D	15	$\mathbf{x_0}$	1000	8.419784e-08	8.340248e-08	2.376317e-04
D	15	$\mathbf{x_p}$	1000	8.566485e-08	8.566486e-08	2.442694e-04
–	15	$\mathbf{x_p}$	1000	9.881311e-07	9.881311e-07	3.102543e-02
D	20	$\mathbf{x_0}$	1000	4.200906e-08	4.136027e-08	6.633700e-05
D	20	$\mathbf{x_p}$	1000	4.903624e-08	4.903624e-08	8.586800e-05
–	20	$\mathbf{x_p}$	1000	9.778842e-07	9.778842e-07	2.990152e-02
D	25	$\mathbf{x_0}$	1000	2.623944e-08	2.567200e-08	3.113882e-05
D	25	$\mathbf{x_p}$	1000	2.590530e-08	2.590530e-08	3.029472e-05
–	25	$\mathbf{x_p}$	1000	9.664934e-07	9.664934e-07	2.908428e-02
D	30	$\mathbf{x_0}$	1000	1.550523e-08	1.506970e-08	1.520711e-05
D	30	$\mathbf{x_p}$	1000	1.502459e-08	1.502459e-08	1.450105e-05
–	30	$\mathbf{x_p}$	1000	9.721409e-07	9.721409e-07	2.922355e-02
D	35	$\mathbf{x_0}$	1000	9.474167e-09	9.159574e-09	8.419319e-06
D	35	$\mathbf{x_p}$	1000	9.418114e-09	9.418114e-09	8.324999e-06
–	35	$\mathbf{x_p}$	1000	9.776118e-07	9.776118e-07	2.918531e-02
D	40	$\mathbf{x_0}$	1000	6.199878e-09	5.986477e-09	5.179355e-06
D	40	$\mathbf{x_p}$	1000	6.309248e-09	6.309248e-09	5.232075e-06
–	40	$\mathbf{x_p}$	1000	9.446054e-07	9.446054e-07	2.714781e-02

Wir erhalten keine Konvergenz innerhalb von 1000 Iterationen bei Singulärpaaren zu größten Singulärwerten. Für Startvektor $\mathbf{x_p}$ nimmt die Norm des relativen Residuums und des Fehlers mit der Zahl der berücksichtigten Singulärpaare ab. Die Ungleichung $\|\mathbf{r_{m_p}}\|_2 \leq \|\mathbf{r_m}\|_2$ (s. Satz 5.6) ist hier nicht mehr erfüllt.

Tabelle A.9: **A1**: CSYM-Verfahren mit Singulärpaaren aus CSYM(m) ohne Reorthogonalisierung (gleiches b)

D	m	k	x_0	*iter*	acc relres	tru relres	error
–	–	–	$\mathbf{x_0}$	778	9.828176e-10	9.828174e-10	7.973643e-09
D	70	3	$\mathbf{x_p}$	760	9.977463e-10	2.925333e-09	8.037728e-09
–	70	3	$\mathbf{x_p}$	776	9.991843e-10	9.991843e-10	8.111312e-09
D	80	5	$\mathbf{x_p}$	759	9.823349e-10	1.104133e-08	7.202408e-09
–	80	5	$\mathbf{x_p}$	778	9.634183e-10	9.634184e-10	7.799751e-09
D	90	6	$\mathbf{x_p}$	746	9.906792e-10	3.391104e-09	7.967737e-09
–	90	6	$\mathbf{x_p}$	777	9.936988e-10	9.936987e-10	8.072359e-09
D	100	7	$\mathbf{x_p}$	745	9.918866e-10	5.953714e-09	7.911838e-09
–	100	7	$\mathbf{x_p}$	777	9.912510e-10	9.912509e-10	8.120767e-09
D	110	7	$\mathbf{x_p}$	743	9.925606e-10	2.603981e-09	8.086314e-09
–	110	7	$\mathbf{x_p}$	777	9.976254e-10	9.976254e-10	8.087272e-09
D	120	9	$\mathbf{x_p}$	741	9.984615e-10	6.519841e-09	7.846993e-09
–	120	9	$\mathbf{x_p}$	776	9.860152e-10	9.860150e-10	7.986843e-09
D	130	12	$\mathbf{x_p}$	821	9.863033e-10	9.047896e-09	7.170547e-09
–	130	12	$\mathbf{x_p}$	776	9.980435e-10	9.980436e-10	8.043835e-09
D	140	18	$\mathbf{x_p}$	1000	2.324643e-09	1.655144e-08	1.228738e-08
–	140	18	$\mathbf{x_p}$	777	9.842548e-10	9.842547e-10	8.011977e-09
D	150	19	$\mathbf{x_p}$	1000	2.895767e-09	5.643157e-08	1.762940e-08
–	150	19	$\mathbf{x_p}$	776	9.927578e-10	9.927578e-10	8.030549e-09
D	160	19	$\mathbf{x_p}$	1000	5.033817e-09	1.307298e-07	3.426188e-08
–	160	19	$\mathbf{x_p}$	778	9.574075e-10	9.574078e-10	7.786386e-09
D	170	22	$\mathbf{x_p}$	1000	2.206748e-09	3.621833e-08	1.191120e-08
–	170	22	$\mathbf{x_p}$	778	9.929547e-10	9.929551e-10	8.009412e-09
D	180	23	$\mathbf{x_p}$	1000	3.929898e-09	1.008940e-07	2.602437e-08
–	180	23	$\mathbf{x_p}$	776	9.989608e-10	9.989608e-10	8.045250e-09
D	190	25	$\mathbf{x_p}$	1000	2.104500e-08	9.123487e-08	1.102287e-07
–	190	25	$\mathbf{x_p}$	777	9.989752e-10	9.989753e-10	8.081362e-09
D	200	27	$\mathbf{x_p}$	1000	7.304793e-09	2.018582e-07	5.073841e-08
–	200	27	$\mathbf{x_p}$	777	9.815986e-10	9.815984e-10	7.932794e-09

Tabelle A.10: **A1**: CSYM-Verfahren mit Singulärpaaren aus CSYM(m) mit CR (gleiches b)

D	m	k	x_0	*iter*	acc relres	tru relres	error
–	–	–	$\mathbf{x_0}$	778	9.828176e-10	9.828174e-10	7.973643e-09
D	70	3	$\mathbf{x_p}$	762	9.862179e-10	2.803672e-09	8.021891e-09
–	70	3	$\mathbf{x_p}$	776	9.885401e-10	9.885401e-10	8.023537e-09
D	80	5	$\mathbf{x_p}$	759	9.816432e-10	1.103246e-08	7.199484e-09
–	80	5	$\mathbf{x_p}$	777	9.949353e-10	9.949353e-10	8.137004e-09
D	90	6	$\mathbf{x_p}$	747	9.791379e-10	2.840637e-09	7.898680e-09
–	90	6	$\mathbf{x_p}$	778	9.900233e-10	9.900237e-10	8.024850e-09
D	100	7	$\mathbf{x_p}$	742	9.959151e-10	1.125494e-09	8.068386e-09
–	100	7	$\mathbf{x_p}$	777	9.860617e-10	9.860618e-10	8.044766e-09
D	110	7	$\mathbf{x_p}$	742	9.986070e-10	1.406938e-09	8.085568e-09
–	110	7	$\mathbf{x_p}$	775	9.707923e-10	9.707927e-10	7.937186e-09
D	120	12	$\mathbf{x_p}$	732	9.977844e-10	6.737910e-09	7.757888e-09
–	120	12	$\mathbf{x_p}$	778	9.898971e-10	9.898973e-10	8.000455e-09
D	130	15	$\mathbf{x_p}$	723	9.800250e-10	3.048306e-09	7.919313e-09
–	130	15	$\mathbf{x_p}$	777	9.734334e-10	9.734335e-10	7.955584e-09
D	140	15	$\mathbf{x_p}$	722	9.823315e-10	2.269542e-09	7.895201e-09
–	140	15	$\mathbf{x_p}$	777	9.731168e-10	9.731176e-10	7.943977e-09
D	150	17	$\mathbf{x_p}$	717	9.820050e-10	2.576653e-09	7.952913e-09
–	150	17	$\mathbf{x_p}$	777	9.644085e-10	9.644082e-10	7.809968e-09
D	160	21	$\mathbf{x_p}$	707	9.865580e-10	2.225513e-09	7.945285e-09
–	160	21	$\mathbf{x_p}$	776	9.678498e-10	9.678497e-10	7.847661e-09
D	170	23	$\mathbf{x_p}$	702	9.863097e-10	3.507260e-09	7.888399e-09
–	170	23	$\mathbf{x_p}$	774	9.707358e-10	9.707357e-10	7.819341e-09
D	180	30	$\mathbf{x_p}$	685	9.902371e-10	1.922417e-09	8.010092e-09
–	180	30	$\mathbf{x_p}$	773	9.937890e-10	9.937886e-10	8.075994e-09
D	190	33	$\mathbf{x_p}$	679	9.945415e-10	5.974731e-09	7.842735e-09
–	190	33	$\mathbf{x_p}$	772	9.911429e-10	9.911426e-10	7.942213e-09
D	200	33	$\mathbf{x_p}$	678	9.651456e-10	2.425103e-09	7.701924e-09
–	200	33	$\mathbf{x_p}$	772	9.921929e-10	9.921930e-10	7.993179e-09

Tabelle A.11: **A1**: CSYM-Verfahren mit Singulärpaaren aus CSYM(m) mit PR (gleiches b)

D	m	k	x_0	*iter*	acc relres	tru relres	error
–	–	–	$\mathbf{x_0}$	778	9.828176e-10	9.828174e-10	7.973643e-09
D	70	3	$\mathbf{x_p}$	761	9.864692e-10	2.895917e-09	7.969150e-09
–	70	3	$\mathbf{x_p}$	776	9.826303e-10	9.826300e-10	7.982497e-09
D	80	5	$\mathbf{x_p}$	759	9.951562e-10	1.117142e-08	7.316412e-09
–	80	5	$\mathbf{x_p}$	777	9.943770e-10	9.943772e-10	8.141449e-09
D	90	6	$\mathbf{x_p}$	746	9.629127e-10	2.882365e-09	7.844009e-09
–	90	6	$\mathbf{x_p}$	776	9.863315e-10	9.863314e-10	8.040247e-09
D	100	7	$\mathbf{x_p}$	741	9.932981e-10	1.192964e-09	8.102337e-09
–	100	7	$\mathbf{x_p}$	777	9.822263e-10	9.822261e-10	8.020984e-09
D	110	7	$\mathbf{x_p}$	743	9.819709e-10	1.049122e-09	8.038585e-09
–	110	7	$\mathbf{x_p}$	777	9.838400e-10	9.838403e-10	8.067664e-09
D	120	12	$\mathbf{x_p}$	732	9.797499e-10	6.480723e-09	7.694791e-09
–	120	12	$\mathbf{x_p}$	776	9.859174e-10	9.859176e-10	7.959135e-09
D	130	15	$\mathbf{x_p}$	721	9.888988e-10	3.193729e-09	7.980365e-09
–	130	15	$\mathbf{x_p}$	777	9.909493e-10	9.909494e-10	8.080670e-09
D	140	15	$\mathbf{x_p}$	722	9.806445e-10	1.378893e-09	7.936258e-09
–	140	15	$\mathbf{x_p}$	776	9.938215e-10	9.938217e-10	8.003022e-09
D	150	17	$\mathbf{x_p}$	717	9.956181e-10	2.594070e-09	8.042841e-09
–	150	17	$\mathbf{x_p}$	776	9.765825e-10	9.765827e-10	7.861383e-09
D	160	21	$\mathbf{x_p}$	706	9.736922e-10	2.945047e-09	7.828145e-09
–	160	21	$\mathbf{x_p}$	776	9.939707e-10	9.939708e-10	8.057107e-09
D	170	23	$\mathbf{x_p}$	703	9.935201e-10	5.590145e-09	7.840436e-09
–	170	23	$\mathbf{x_p}$	774	9.955307e-10	9.955304e-10	8.024376e-09
D	180	30	$\mathbf{x_p}$	684	9.868911e-10	2.163599e-09	7.914586e-09
–	180	30	$\mathbf{x_p}$	773	9.833048e-10	9.833048e-10	7.949875e-09
D	190	33	$\mathbf{x_p}$	677	9.973122e-10	3.527088e-09	7.970941e-09
–	190	33	$\mathbf{x_p}$	773	9.849918e-10	9.849917e-10	7.914001e-09
D	200	33	$\mathbf{x_p}$	677	9.800236e-10	1.918848e-09	7.905887e-09
–	200	33	$\mathbf{x_p}$	772	9.872861e-10	9.872860e-10	7.912025e-09

Tabelle A.12: **A1**: CSYM-Verfahren mit Singulärpaaren aus CSYM(m) ohne Reorthogonalisierung

D	m	k	x_0	*iter*	acc relres	tru relres	error
–	–	–	$\mathbf{x_0}$	778	9.828176e-10	9.828174e-10	7.973643e-09
D	60	5	$\mathbf{x_p}$	937	9.673222e-10	4.177113e-10	6.426839e-09
–	60	5	$\mathbf{x_p}$	751	9.844757e-10	9.844757e-10	1.758404e-08
D	70	5	$\mathbf{x_p}$	938	9.994695e-10	6.728257e-10	6.655603e-09
–	70	5	$\mathbf{x_p}$	750	9.999380e-10	9.999379e-10	1.754507e-08
D	80	7	$\mathbf{x_p}$	930	9.838421e-10	4.426687e-09	6.034898e-09
–	80	7	$\mathbf{x_p}$	750	9.922783e-10	9.922784e-10	1.762429e-08
D	90	7	$\mathbf{x_p}$	1000	1.327379e-09	3.930936e-09	8.911254e-09
–	90	7	$\mathbf{x_p}$	751	9.576018e-10	9.576017e-10	1.709897e-08
D	100	7	$\mathbf{x_p}$	927	9.949131e-10	2.643493e-09	6.458182e-09
–	100	7	$\mathbf{x_p}$	750	9.678829e-10	9.678826e-10	1.725941e-08
D	110	9	$\mathbf{x_p}$	1000	2.263847e-08	1.385851e-08	1.467599e-07
–	110	9	$\mathbf{x_p}$	750	9.834447e-10	9.834447e-10	1.764755e-08
D	120	13	$\mathbf{x_p}$	1000	1.385463e-08	4.324493e-08	9.144860e-08
–	120	13	$\mathbf{x_p}$	747	9.681045e-10	9.681045e-10	2.012036e-08
D	130	19	$\mathbf{x_p}$	1000	7.666736e-07	2.701732e-06	4.672033e-06
–	130	19	$\mathbf{x_p}$	730	9.834467e-10	9.834469e-10	3.055773e-08
D	140	19	$\mathbf{x_p}$	1000	5.185681e-07	3.402549e-06	3.683590e-06
–	140	19	$\mathbf{x_p}$	731	9.755997e-10	9.755996e-10	3.048257e-08
D	150	24	$\mathbf{x_p}$	1000	8.563066e-07	2.781229e-06	5.222651e-06
–	150	24	$\mathbf{x_p}$	730	9.675661e-10	9.675660e-10	3.168156e-08
D	160	25	$\mathbf{x_p}$	1000	5.312546e-07	2.770974e-06	3.550528e-06
–	160	25	$\mathbf{x_p}$	729	9.988401e-10	9.988402e-10	3.301607e-08
D	170	27	$\mathbf{x_p}$	1000	2.139414e-06	8.978272e-06	1.374792e-05
–	170	27	$\mathbf{x_p}$	725	9.738009e-10	9.738009e-10	3.569416e-08
D	180	39	$\mathbf{x_p}$	1000	1.392029e-06	6.282284e-06	1.026127e-05
–	180	39	$\mathbf{x_p}$	721	9.504458e-10	9.504458e-10	4.191368e-08
D	190	41	$\mathbf{x_p}$	1000	2.266755e-05	1.773374e-05	1.300226e-04
–	190	41	$\mathbf{x_p}$	721	9.928090e-10	9.928089e-10	4.280674e-08
D	200	43	$\mathbf{x_p}$	1000	1.815045e-04	1.015169e-03	1.574316e-03
–	200	43	$\mathbf{x_p}$	719	9.812683e-10	9.812683e-10	4.271654e-08

Tabelle A.13: **A1**: CSYM-Verfahren mit Singulärpaaren aus CSYM(m) mit CR

D	m	k	x_0	*iter*	acc relres	tru relres	error
–	–	–	$\mathbf{x_0}$	778	9.828176e-10	9.828174e-10	7.973643e-09
D	60	5	$\mathbf{x_p}$	936	9.835910e-10	4.245463e-10	6.523767e-09
–	60	5	$\mathbf{x_p}$	750	9.868637e-10	9.868636e-10	1.746936e-08
D	70	5	$\mathbf{x_p}$	937	9.956847e-10	4.335866e-10	6.641256e-09
–	70	5	$\mathbf{x_p}$	750	9.926423e-10	9.926423e-10	1.741411e-08
D	80	7	$\mathbf{x_p}$	924	9.927831e-10	4.295995e-10	6.593877e-09
–	80	7	$\mathbf{x_p}$	750	9.949619e-10	9.949618e-10	1.779082e-08
D	90	7	$\mathbf{x_p}$	924	9.703245e-10	4.199849e-10	6.440307e-09
–	90	7	$\mathbf{x_p}$	748	9.986914e-10	9.986915e-10	1.787846e-08
D	100	8	$\mathbf{x_p}$	923	9.922378e-10	5.005222e-10	6.616701e-09
–	100	8	$\mathbf{x_p}$	749	9.753134e-10	9.753135e-10	1.761192e-08
D	110	13	$\mathbf{x_p}$	908	9.948776e-10	9.152562e-10	6.587844e-09
–	110	13	$\mathbf{x_p}$	745	9.965541e-10	9.965542e-10	2.074867e-08
D	120	16	$\mathbf{x_p}$	899	9.762871e-10	7.241093e-10	6.503490e-09
–	120	16	$\mathbf{x_p}$	736	9.965867e-10	9.965867e-10	2.621996e-08
D	130	15	$\mathbf{x_p}$	900	9.639590e-10	9.619288e-10	6.367000e-09
–	130	15	$\mathbf{x_p}$	739	9.884907e-10	9.884908e-10	2.429797e-08
D	140	20	$\mathbf{x_p}$	886	9.541616e-10	9.662942e-10	6.292133e-09
–	140	20	$\mathbf{x_p}$	736	9.789884e-10	9.789884e-10	2.648260e-08
D	150	21	$\mathbf{x_p}$	883	9.923750e-10	9.997731e-10	6.597888e-09
–	150	21	$\mathbf{x_p}$	735	9.956411e-10	9.956412e-10	2.730455e-08
D	160	32	$\mathbf{x_p}$	852	9.848470e-10	1.086499e-09	6.534564e-09
–	160	32	$\mathbf{x_p}$	727	9.974004e-10	9.974003e-10	3.603037e-08
D	170	33	$\mathbf{x_p}$	849	9.944968e-10	1.090508e-09	6.648136e-09
–	170	33	$\mathbf{x_p}$	727	9.790699e-10	9.790699e-10	3.524709e-08
D	180	34	$\mathbf{x_p}$	848	9.638291e-10	1.157150e-09	6.341229e-09
–	180	34	$\mathbf{x_p}$	726	9.779637e-10	9.779638e-10	3.580711e-08
D	190	38	$\mathbf{x_p}$	838	9.803783e-10	1.047683e-09	6.544337e-09
–	190	38	$\mathbf{x_p}$	721	9.909429e-10	9.909429e-10	4.078952e-08
D	200	41	$\mathbf{x_p}$	830	9.770400e-10	9.987927e-10	6.539771e-09
–	200	41	$\mathbf{x_p}$	721	9.807278e-10	9.807278e-10	4.107050e-08

Tabelle A.14: **A1**: CSYM-Verfahren mit Singulärpaaren aus CSYM(m) mit PR

D	m	k	x_0	*iter*	acc relres	tru relres	error
–	–	–	$\mathbf{x_0}$	778	9.828176e-10	9.828174e-10	7.973643e-09
D	60	5	$\mathbf{x_p}$	934	9.668100e-10	4.172885e-10	6.406584e-09
–	60	5	$\mathbf{x_p}$	750	9.986173e-10	9.986176e-10	1.764534e-08
D	70	5	$\mathbf{x_p}$	934	9.821771e-10	4.272401e-10	6.513306e-09
–	70	5	$\mathbf{x_p}$	750	9.673189e-10	9.673189e-10	1.701099e-08
D	80	7	$\mathbf{x_p}$	925	9.941981e-10	4.275522e-10	6.634431e-09
–	80	7	$\mathbf{x_p}$	749	9.971128e-10	9.971130e-10	1.788046e-08
D	90	7	$\mathbf{x_p}$	925	9.679822e-10	4.163183e-10	6.434015e-09
–	90	7	$\mathbf{x_p}$	751	9.761055e-10	9.761055e-10	1.746343e-08
D	100	8	$\mathbf{x_p}$	924	9.667694e-10	4.135161e-10	6.421985e-09
–	100	8	$\mathbf{x_p}$	749	9.828700e-10	9.828700e-10	1.775380e-08
D	110	13	$\mathbf{x_p}$	906	9.918872e-10	3.757855e-10	6.588771e-09
–	110	13	$\mathbf{x_p}$	747	9.907850e-10	9.907849e-10	2.062306e-08
D	120	16	$\mathbf{x_p}$	898	9.899574e-10	3.038295e-10	6.614593e-09
–	120	16	$\mathbf{x_p}$	737	9.925457e-10	9.925456e-10	2.639516e-08
D	130	15	$\mathbf{x_p}$	899	9.673822e-10	3.059565e-10	6.455083e-09
–	130	15	$\mathbf{x_p}$	739	9.785983e-10	9.785984e-10	2.406832e-08
D	140	20	$\mathbf{x_p}$	885	9.852829e-10	3.107728e-10	6.591278e-09
–	140	20	$\mathbf{x_p}$	737	9.558094e-10	9.558094e-10	2.620051e-08
D	150	21	$\mathbf{x_p}$	883	9.925079e-10	3.220949e-10	6.649448e-09
–	150	21	$\mathbf{x_p}$	737	9.664292e-10	9.664293e-10	2.624848e-08
D	160	32	$\mathbf{x_p}$	850	9.979437e-10	3.098624e-10	6.766829e-09
–	160	32	$\mathbf{x_p}$	727	9.712481e-10	9.712481e-10	3.524163e-08
D	170	33	$\mathbf{x_p}$	847	9.950093e-10	3.278383e-10	6.763132e-09
–	170	33	$\mathbf{x_p}$	726	9.847088e-10	9.847088e-10	3.538691e-08
D	180	34	$\mathbf{x_p}$	847	9.830188e-10	3.507492e-10	6.681407e-09
–	180	34	$\mathbf{x_p}$	725	9.991130e-10	9.991131e-10	3.675232e-08
D	190	38	$\mathbf{x_p}$	838	9.761406e-10	3.320999e-10	6.657363e-09
–	190	38	$\mathbf{x_p}$	722	9.801794e-10	9.801793e-10	4.009708e-08
D	200	41	$\mathbf{x_p}$	828	9.990252e-10	3.722863e-10	6.845166e-09
–	200	41	$\mathbf{x_p}$	721	9.664638e-10	9.664638e-10	4.050988e-08

Literaturverzeichnis

[1] C. Ashcraft, R. G. Grimes, J. G. Lewis, Accurate symmetric indefinite linear equation solvers, SIAM J. Matrix Anal. Appl. 20, No.2, 513-561 (1998)

[2] W. E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, Q. appl. Math. 9, 17-29 (1951)

[3] J. Baglama, D. Calvetti, G. H. Golub, L. Reichel, Adaptively preconditioned GMRES algorithms, SIAM J. Sci. Comput. 20, No.1, 243-269 (1999)

[4] A. Bunse-Gerstner, W. B. Gragg, Singular value decompositions of complex symmetric matrices, J. Comput. Appl. Math. 21, 41-54 (1988)

[5] A. Bunse-Gerstner, R. Stöver, On a conjugate gradient-type method for solving complex symmetric linear systems, Linear Algebra Appl. 287, No.1-3, 105-123 (1999)

[6] B. Carpentieri, I. S. Duff, L. Giraud, M. Mangolu monga Made, Sparse symmetric preconditioners for dense linear systems in electromagnetism, CERFACS Technical Report TR/PA/01/35

[7] H. Choi, D. B. Szyld, Application of threshold Partitioning of Sparse Matrices to Markov Chains, Proceedings of the IEEE International Computer Performance and Dependability Symposium IPDS'96, IEEE Computer Society Press, Los Alamitos, CA, 158-165 (1996)

[8] B. D. Craven, Complex symmetric matrices, J. Aust. Math. Soc. 10, 341-354 (1969)

[9] J. K. Cullum, R. A. Willoughby, Lanczos algorithms for large symmetric eigenvalue computations, Volume I. Theory, Birkhäuser, Boston, 1985

[10] Iain S. Duff, Roger G. Grimes, John G. Lewis, Users' Guide for the Harwell-Boeing Sparse Matrix Collection (Release I), TR/PA/92/86

[11] R. W. Freund, Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices, SIAM J. Sci. Statist. Comput. 13, 425-448 (1992)

[12] David Fritzsche, Graph Theoretical Methods For Preconditioners, Diplomarbeit, Bergische Universität Wuppertal, Mai 2004

[13] F. R. Gantmacher, The Theory of Matrices, Vol. 2., Chelsea Publ., New York,1974

[14] G. H. Golub, C. F. van Loan, Matrix Computations (North Oxford Academic, Oxford, 1983)

[15] G. H. Golub, R. Underwood, The Block Lanczos method for computing eigenvalues. Mathematical software III, Proc. Symp., Madison 1977, 361-377 (1977)

[16] Anne Greenbaum, Iterative methods for solving linear systems, Frontiers in Applied Mathematics. 17. Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics (1997)

[17] Chengshu Guo, Sanzheng Qiao, A Stable Lanczos Tridiagonalization of Complex Symmetric Matrices, Technical Report No. CAS 03-08-SQ, Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada. L8S 4K1. July 2003.

[18] M. R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. 49, 409-436 (1952)

[19] Nicholas J. Higham, Accuracy and stability of numerical algorithms. 2nd ed. Philadelphia, PA: SIAM (2002)

[20] Nicholas J. Higham, Factorizing complex symmetric matrices with positive definite real and imaginary parts, Math. Comput. 67, No.224, 1591-1599 (1998)

[21] N. J. Higham, The Matrix Computation Toolbox,
http://www.ma.man.ac.uk/ higham/mctoolbox

[22] M. E. Hochstenbach, Harmonic and refined extraction methods for the singular value problem, with applications in least squares problems, BIT 44, No.4, 721-754 (2004)

[23] Roger A. Horn, Charles R. Johnson, Matrix Analysis, Cambridge University Press, Reprinted 1999

[24] Roger A. Horn, Charles R. Johnson, Topics in Matrix Analysis, Cambridge University Press, Reprinted 1995

[25] E. Kokiopoulo, C. Bekas, E. Gallopoulos, Computing Smallest Singular Triplets with Implicitly Restarted Lanczos Bidiagonalization, Appl. Numer. Math. 49, No.1, 39-61 (2004)

[26] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Nat. Res. Bur. Standards 45, 255-282 (1959)

[27] Using MATLAB. The Math Works, Inc.,24 Prime Park Way Natick, MA 01760-1500, 1999

[28] W. Niethammer, Relaxation bei komplexen Matrizen, Math. Zeitschr. 86, 34-40 (1964)

[29] J. O'Neil, D. B. Szyld: A block ordering method for sparse matrices, SIAM J. Sci. Stat. Comput. 11, 811-823 (1990)

[30] J. M. Ortega, W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in several Variables, academic Press, Inc., Orlando, Florida 32887, 1970

[31] C. C. Paige, M. A. Saunders, Solution of sparse indefinite systems of linear equations, SIAM J. Numer. Anal., 11, 197-209 (1974)

[32] B. N. Parlett, The Symmetric Eigenvalue Problem, Prentice-Hall, Englewood Cliffs, NJ, 1980

[33] U. van Rienen, Numerical Methods in Computational Electrodynamics: Linear Systems in Practical Applications, (Lecture notes in computational science and engineering; 12), Springer-Verlag Berlin Heidelberg, 2001

[34] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS PUBLISHING COMPANY, 20 Park Plaza, Boston, MA 02116-4324, 1996

[35] Y. Saad and M. H. Schultz, GMRES a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comp., 7, 856–869 (1986).

[36] Sanzheng Qiao, Orthogonalization Techniques for the Lanczos Tridiagonalization of Complex Symmetric Matrices, Advanced Signal Processing Algorithms, Architectures, and Implementations XIV, edited by Franklin T. Luk, Proc. of SPIE Vol. 5559, 423–434 (2004)

[37] M.A. Saunders, H.D. Simon, E. L. Yip, Two Conjugate-Gradient-Type Methods for Unsymmetric Linear Equations, SIAM J.Numer.Anal. 25, 927-940 (1988)

[38] V. Simoncini, E. Sjöström, An algorithm for approximating the singular triplets of complex symmetric matrices. Numer. Linear Algebra Appl. 4, No.6, 469-489 (1997).

[39] T. Takagi, On an algebraic problem related to an analytic theorem of Caratheodory and Feyer and on allid theorem of Landau, Japanese J. Math., 1 (1927), 83-93

[40] R. C. Thompson, Singular values and diagonal elements of complex symmetric matrices, Lin. Alg. Appl. 26, 65-106 (1978)

[41] T. Vaupel, Convergence properties of linear equation solvers applied to iterative spectral domain integral equation methods, SCEE-2000 Scientific Computing in Electrical Engineering, 20.-23.08.2000, Warnemünde, Germany

[42] R. J. Vanderbei, Symmetric quasidefinite matrices. SIAM J. Optim. 5, No.1, 100-113 (1995)

[43] H. A. van der Vorst, J. B. M. Melissen, A Petrov-Galerkin type method for solving Ax = b, where A is symmetric complex, IEEE Trans. Magnetics 26, 706-708 (1990).

[44] J. W. L. Wan, Solving Complex Symmetric Linear Systems with Multiple Right-Hand Sides, Seventh SIAM Conference on Applied Linear Algebra, Oct. 23-25, 2000