ÜBER QUADRATISCH KONVERGENTE JACOBI-ÄHNLICHE BLOCKVERFAHREN FÜR BELIEBIGE REELLE MATRIZEN MIT KOMPLEXEN EIGENWERTEN

DISSERTATION

zur

Erlangung des Grades eines Dr. rer. nat.

des Fachbereichs Mathematik und Informatik

der FERNUNIVERSITÄT -Gesamthochschule- Hagen

vorgelegt von

HANS JOACHIM WENZEL

aus Dortmund

Hagen 1983

Berichterstatter: Prof.Dr. K. Veselić

Prof.Dr. M. Reimer

Tag der mündlichen Prüfung: 5.Juli 1983

I_n_h_a_1_t_s_v_e_r_z_e_i_c_h_n_i_s

Einlei	tung	1
1.	Grundlagen	1
1.1	Grundlegende Definitionen und Sätze	1
1.2	Charakteristisches Polynom, Nullstellen	8
1.2.1	Minima eines Polynoms vierten Grades	8
1.2.2	Nullstellen eines Polynoms vierten Grades	9
1.2.3	Nullstellen eines kubischen Polynoms	10
1.2.4	Charakteristisches Polynom einer reellen schiefsymmetrischen 4×4-Matrix	15
1.2.5	Charakteristisches Polynom einer reellen symmetrischen 4×4-Matrix	16
2.	Elementare Matrizen für Jacobi-ähnliche Verfahren	22
2.1	Die elementare Johnsen-Veselić-Voevodin- Matrix	24
2.1.1	Definition und normreduzierende Eigenschaften	24
2.1.2	Der Parameter X ₂ für quadratische Konvergenz	28
2.1.3	Ein verbesserter Parameter X_3 für quadratische Konvergenz	34
2.2	Die elementare Paardekooper-Jacobi-Matrix	40
2.3	Die elementare Veselić-Householder-Matrix	50
2.4	Die elementare Eberlein-Jacobi-Matrix	53
3.	Konvergenzbeweise	55
3.1	Zur globalen Konvergenz eines Jacobi- ähnlichen normreduzierenden Verfahrens bei optimaler Pivotwahl	56
3.2	Zur asymptotisch quadratischen Konvergenz eines Jacobi-ähnlichen normreduzierenden Verfahrens bei zeilenzyklischer Pivotstrategie	58

3.2.1	Beschreibung des Verfahrens	58
3.2.2	Vorbereitende Lemmata	61
3.2.3	Der zweite Hauptsatz	75
3.2.4	Normreduzierung der Blockdiagonalen	84
4.	Jacobi-ähnliche Verfahren für die praktische Berechnung der Eigenwerte und Eigenvektoren	96
4.1	Ein Verfahren für reelle Matrizen mit kom- plexen Eigenwerten, deren Imaginärteile ge- trennt sind (Verfahren I)	102
4.2	Ein Verfahren für allgemeine reelle Matrizen (Verfahren II)	104
4.3	Numerische Resultate	110
Anha	a n g	
A) K (onstantenverzeichnis	127
	L G O L 60 - Programm für Verfahren II, riante 2	130
C) E 1	rgebnisse-(Beispiel)	165
Lite	eraturverzeichnis	169

In der Mathematik begegnet es mir nicht selten: Ich habe das Resultat; ich weiß nur nicht, auf welchem Wege ich es finden werde.

Carl Friedrich Gauß

EINLEITUNG

Die vorliegende Arbeit beschäftigt sich mit der Konstruktion Jacobi-ähnlicher Verfahren (s.Kap.2 und 4), sowie deren Konvergenz (s.Kap.3) und numerischen Anwendbarkeit (s.Kap.4). Jacobi-ähnliche Verfahren sind Verallgemeinerungen des klassischen Jacobi-Verfahrens [16]. Dieses im Jahre 1846 von C.G.F.Jacobi angegebene Verfahren zur Berechnung aller Eigenwerte und Eigenvektoren reeller symmetrischer Matrizen diagonalisiert die gegebene Matrix mittels Jacobi-Rotationen (s.§2.4) in einem infiniten Prozess. Wegen des hohen Rechenaufwandes blieb das Verfahren lange Zeit unpraktikabel, bis ihm über 100 Jahre nach der Veröffentlichung die elektronischen Rechenautomaten zu einer späten Renaissance verhalfen.

Dabei wurden die Pivotelemente zur Rotation, aus Gründen der Rechenzeitoptimierung, in zyklischer Folge (s.[29] und Kap.2) und nicht, wie im klassischen Jacobi-Verfahren, der Betragsgröße nach (d.h. mit optimaler Pivotstrategie) gewählt. Das war zwar numerisch erfolgreich, verursachte aber Probleme beim Beweis der Konvergenz. Die Konvergenz und ihre Geschwindigkeit wurde von Henrici ([14], (1958)), Schönhage ([30], (1961), [31] (1964)) und Wilkinson ([40], (1962)) untersucht, wobei Henrici [14] auch eine Verallgemeinerung für Hermitesche Matrizen angab.

Paardekooper ([26], (1971)) gab ein Jacobi-ähnliches Verfahren für schiefsymmetrische Matrizen an, bei dem mittels vier nacheinander ausgeführten Jacobi-Rotationen ein Pivotblock *) annulliert wird (s.§ 2.2, Kap.3, § 4.1).

*) Ein Block ist hier eine 2×2 -Untermatrix, vergl. (1.1.2).

Die Konvergenz dieses Verfahrens wurde von Hari ([10],[11],[12], (1982)) untersucht.

Zur Konstruktion Jacobi-ähnlicher Verfahren für nicht normale Matrizen werden Ähnlichkeitstransformationen benötigt, die die Euklidische Norm der Matrix reduzieren, da nach einem Satz von I. Schur [32] (s. Satz 1.1.14) das Quadrat der Euklidischen Norm nach unten beschränkt ist durch die Summe der Betragsquadrate der Eigenwerte und Gleichheit genau für normale Matrizen gilt. Voevodin [39], (1966)) führte dazu Matrizen der Form ein (s. § 2.1), während Eberlein und Boothroyd ([6],(1968)) ein Jacobiähnliches Verfahren für reelle Matrizen konstruierten, welches mit Scherungen (s. § 2.4) zur Normreduzierung und Jacobi-Rotationen zur Annullierung des Pivotelements des symmetrischen Teils der Matrix arbeitet. Weiterhin gab Eberlein ([5],(1970)) ein Verfahren für komplexe Matrizen an, das komplexe Scherungen zur Normreduzierung und komplexe Rotationen zur wahlweisen Annullierung des Pivotelements des hermiteschen- oder schiefhermiteschen Teils der Matrix verwendet (vergl.auch § 4.2).

Für das reelle Verfahren von Eberlein bewies Veselic ([33],(1976)) die Konvergenz für eine optimale Variante, Dollinger ([3],(1981)) für eine quasizyklische Variante und Hari ([9],(1982)) für das originale zyklische Verfahren. Für das komplexe Verfahren von Eberlein [5] ist das Problem der globalen Konvergenz noch offen. Trotzdem gilt dieses Verfahren als eines der zuverlässigsten zur Berechnung aller Eigenwerte. Allgemein ist die Entwicklung numerisch erfolgreicher Verfahren erheblich weiter als der Beweis deren Konvergenz. So werden alle hier zitierten Verfahren erfolgreich mit zyklischer Pivotstrategie verwendet, obwohl für viele Verfahren (noch) kein entsprechender Konvergenzbeweis existiert. Wir leisten hier einen Beitrag in beiden Richtungen; für ein Verfahren führen wir einen Konvergenzbeweis, und wir konstruieren ein neues Verfahren, das auf einer größeren Klasse von Matrizen erfolgreich arbeitet.

Die Konvergenzgeschwindigkeit herkömmlicher reeller Jacobiähnlicher Verfahren, wie des Verfahrens von Eberlein und verwandter Verfahren, ist beim Auftreten komplexer Eigenwerte langsam (nicht asymptotisch quadratisch), was darin begründet liegt, daß

$$\tau(A) = \left(\sum_{i \neq j} |a_{ij}|^2\right)^{1/2}$$

nicht klein wird, wie es bei ausschließlich reellen, nicht defektiven Eigenwerten der Fall ist. Reelle Matrizen mit komplexen Eigenwerten spielen aber in der Technik eine bedeutende Rolle. Komplexe Jacobi-ähnliche Verfahren, die auch bei komplexen Eigenwerten asymptotisch quadratische Konvergenz ermöglichen, haben gegenüber den reellen Verfahren den Nachteil des doppelten Speicherplatzbedarfs und 2- bis 11-fachen Aufwands der arithmetischen Operationen. Johnsen ([17],(1978)) und Veselić ([34],[38],(1979)) führten reelle Jacobi-ähnliche Blockverfahren ein, deren Konvergenzeigenschaften mit komplexen Verfahren vergleichbar sind (s.[38]). In der vorliegenden Arbeit beschäftigen wir uns mit solchen Verfahren.

In Kap. 2 beweisen wir Eigenschaften der von Johnsen [17] und Veselić [34] verallgemeinerten Voevodin-Matrix (s. § 2.1) und der Paardekooper-Jacobi-Matrix (s. § 2.2,[26],[10]), mit deren Hilfe in Kap. 3 Konvergenzbeweise für Varianten des in [38] angegebenen Verfahrens, welches die o.a. Matrizen benutzt, geführt werden. Für ein Jacobi-ähnliches Verfahren, das mit optimaler Pivotstrategie, Paardekooper-Jacobi-Matrizen und Scherungen arbeitet, hat Hari [7] die globale Konvergenz bewiesen. Analog zu diesem Resultat läßt sich auch die globale Konvergenz unseres Verfahrens bei optimaler Pivotstrategie beweisen (s. § 3.1).

In § 3.2 wird für unser Verfahren die asymptotisch quadratische Konvergenz bei zeilenzyklischer Pivotstrategie gezeigt. Der erste Teil des Beweises (s.Abschn. 3.2.3) kann in Anlehnung an einen Beweis von Ruhe [28] geführt werden. Der Beweis erhält hier eine neue Qualität, weil bei Blockverfahren mit

$$S(A) = \left(\sum_{i \neq j} \|A_{ij}\|^2\right)^{1/2}$$

anstelle von $\tau(A)$ gearbeitet wird und eine zusätzliche Behandlung der Diagonalblöcke A_{ii} erforderlich ist (s. Abschnitte 3.2.2 und 3.2.4).

Die durch dieses Verfahren erzeugte Matrixfolge konvergiert nur dann gegen Blockdiagonalgestalt, wenn die Imaginärteile der Eigenwerte getrennt sind. Soll ein Verfahren für alle Matrizen Konvergenz gegen Blockdiagonalgestalt erzeugen, so ist auch der symmetrische Teil zu diagonalisieren (vergl.[5]).Dazu beschreiben wir in § 2.3 eine orthogonale Matrix, die bei Verwendung in Ähnlichkeitstransformationen die wahlweise Annullierung des symmetrischen bzw. schiefsymmetrischen Teils einer Matrix gestattet. Zur Konstruktion dieser Matrix müssen die Eigenwerte des symmetrischen bzw. schiefsymmetrischen Teils einer 4×4-Matrix numerisch bekannt sein; ein stabiles Verfahren zu deren Berechnung wird in § 1.2 angegeben.

In § 4.2 geben wir ein Jacobi-ähnliches Blockverfahren an, das mit diesen und den verallgemeinerten Voevodin-Matrizen arbeitet und für alle reellen Matrizen Konvergenz gegen Blockdiagonalgestalt erzeugt. Wir führen für dieses Verfahren keinen formalen Konvergenzbeweis, deuten aber an, daß sich die asymptotisch quadratische Konvergenz mit der in § 3.2 entwickelten Technik analog beweisen läßt.

In § 4.3 haben wir numerische Resultate der o.a. Verfahren bei Anwendung auf eine Reihe von Testmatrizen zusammengestellt, aus denen die Vorzüge und Grenzen der Verfahren zu erkennen sind. Alle entwickelten Verfahren sind in ALGOL 60 formuliert und mit dem "Fast ALGOL Compiler of Delft, Release of 1/2/1977" übersetzt worden. Die numerischen Berechnungen wurden auf der Rechenanlage IBM 3031/4MB unter OS/MVS des Rechenzentrums der Fernuniversität -Gesamthochschule - Hagen durchgeführt.

Das Ende eines Beweises wird in dieser Arbeit durch das Symbol **a** am rechten Rand gekennzeichnet, und zur besseren Orientierung ist im Anhang ein alphabetisches Verzeichnis der verwendeten Konstanten angeführt.

Wir hoffen, mit den vorliegenden Untersuchungen einen Beitrag zur Entwicklung Jacobi-ähnlicher Blockverfahren geleistet zu haben.

Ich möchte mich an dieser Stelle bei allen bedanken, die mich bei der vorliegenden Arbeit durch kritische, hilfreiche Diskussionen unterstützt haben. Mein besonderer Dank gilt Herrn Prof.Dr.K.Veselić, der mich zu diesen Untersuchungen angeregt und als kritischer Gesprächspartner den Fortgang der Arbeit durch vielfältige wertvolle Hinweise und Anregungen gefördert hat, und Herrn Prof.Dr. M. Reimer, der mich schon während meines Studiums und bei der Ausfertigung meiner Diplomarbeit intensiv unterstützte und auch hier gerne bereit war, das Korreferat zu übernehmen. Außerdem danke ich Frau H.R.Cramer für die sorgfältige Reinschrift des Manuskripts und die große Geduld bei den Korrekturen.

GRUNDLAGEN

1.1 GRUNDLEGENDE DEFINITIONEN UND SÄTZE

In diesem Abschnitt werden Bezeichnungen eingeführt, Definitionen und Bemerkungen gegeben und Sätze zitiert, die für die späteren Kapitel von grundlegender Bedeutung sind.

Wir gehen im Folgenden stets aus von einer reellen n×n-Matrix

(1.1.1)
$$A = (a_{k1})_{k,1=1(1)n}$$
, $n = 2p, p \in \mathbb{N}$.

Matrizen mit ungerader Dimension können natürlich auch behandelt werden, wenn man z.B. eine Nullzeile und Nullspalte anfügt (d.h. einen Eigenwert "Null" hinzufügt). Die Matrix (1.1.1) wird blockgeteilt in der Form

$$(1.1.2) \qquad A = (A_{ij})_{i,j=1(1)p},$$

wobei die Blöcke A_{ij} reelle 2×2 Matrizen sind. Wir schreiben für A_{ij} auch $[A]_{ij}$. Matrizen dieser Form bezeichnen wir als Blockmatrizen.

Definition 1.1.1

Die Matrix A (1.1.2) hat Blockdiagonalgestalt, wenn

$$A_{ij} = 0$$
, $i \neq j$

gilt. Man schreibt dann abkürzend

(1.1.3)
$$A = diag(A_{11}, A_{22}, ..., A_{pp}) = \bigoplus_{i=1}^{p} A_{ii}$$

wobei die direkte Summe bedeutet.

Definition 1.1.2

Der Kommutator der Matrix A wird definiert durch

$$(1.1.4) C(A) := A^{T}A - AA^{T},$$

wobei A^{T} die Transponierte der Matrix A ist.

Bemerkung 1.1.3

Setzt man

wie in (1.1.1), (1.1.2), so gilt für $1 \le r, s \le p$:

(1.1.6)
$$C_{rs} = \sum_{i=1}^{p} (A_{ir}^{T} A_{is} - A_{ri} A_{si}^{T})$$
,

und damit $C_{sr} = C_{rs}^{T}$, d.h. C(A) ist symmetrisch.

Definition 1.1.4

Eine Ähnlichkeitstransformation einer Matrix A ist eine Abbildung

$$(1.1.7) A \to A' := R^{-1} A R ,$$

wobei R eine reguläre Matrix ist. Sie heißt orthogonal, wenn R orthogonal ist. Matrizen A, A', für die (1.1.7) gilt, heißen $\ddot{a}hnlich$.

Definition 1.1.5

Mit I bezeichnen wir für jede (quadratische) Dimension die Einheitsmatrix

(1.1.8)
$$I := (\delta_{ij}) , \text{ wo } \delta_{ij} := \begin{cases} 1 & \text{für } i = j \\ 0 & \text{für } i \neq j \end{cases}$$

das Kronecker-Symbol bezeichnet.

Satz 1.1.6

Die Eigenwerte *) λ_i , i=1(1)n einer Matrix A sind invariant unter Ähnlichkeitstransformationen.

Bemerkung 1.1.7

Auf dem Raum der quadratischen Matrizen (fester Dimension) wird durch

(1.1.9)
$$\langle A, B \rangle := Spur(A^TB) = Spur(AB^T)$$

ein skalares Produkt definiert.

Definition und Bemerkung 1.1.8

Durch

$$(1.1.10) ||A||_{E} := \langle A, A \rangle^{1/2} = \left(\sum_{i,j=1}^{n} |a_{ij}|^{2}\right)^{1/2}$$

wird eine Matrixnorm definiert.

 $||A||_E$ heißt die *Euklidische Norm* der Matrix A. Da im Folgenden weitgehend die Euklidische Norm verwendet wird, lassen wir den Index weg und schreiben ||A||; ist eine andere als die Euklidische Norm gemeint, so wird dieses jeweils deutlich gemacht.

Eine weitere Matrixnorm, die hier zuweilen Verwendung findet, ist die Spektralnorm

(1.1.11)
$$||A||_{sp} := (spr(A^TA))^{1/2}$$
,

wobei

(1.1.12)
$$\operatorname{spr}(A) := \max\{|\lambda| | \exists X \neq 0, AX = \lambda X\}$$

den Spektralradius bezeichnet.

^{*)} Unter der Gesamtheit aller Eigenwerte verstehen wir die Äquivalenzklasse aller Permutationen des n-Tupels $(\lambda_1, \lambda_2, \ldots, \lambda_n)$, in dem die Nullstellen des charakteristischen Polynoms $\chi(\lambda) := \det(A-\lambda I)$

⁽vgl.Def. 1.1.5) in der Häufigkeit ihrer algebraischen Vielfachheit auftreten. Der Begriff "Eigenwerte von A" ist stets in diesem Sinne zu verstehen.

Die Spektralnorm ist interessant, weil die Ungleichungen

(1.1.13)
$$\|AB\| \le \|A\|_{sp} \|B\|$$
, $\|AB\| \le \|A\| \|B\|_{sp}$ gelten, die i.A. schärfer sind, als

$$(1.1.14)$$
 $||AB|| \le ||A|| ||B||$.

Satz 1.1.9

Die Euklidische Norm einer Matrix A ist invariant unter orthogonalen Ähnlichkeitstransformationen, d.h.

$$\bigcup \bigcup^{\mathbf{T}} = I \Rightarrow ||\bigcup^{\mathbf{T}} A \bigcup || = ||A|| .$$

Definition 1.1.10

Eine reelle Matrix M ist in Murnaghan-Form, wenn sie sich als direkte Summe von Matrizen der Form (λ) für jeden reellen Eigenwert *) λ und $2\times 2-B$ 1öcken der Form

$$\begin{pmatrix} \operatorname{Re} \lambda & \operatorname{Im} \lambda \\ -\operatorname{Im} \lambda & \operatorname{Re} \lambda \end{pmatrix}$$

für jedes Paar konjugiert komplexer Eigenwerte*) $(\lambda,\overline{\lambda})$ darstellen läßt. M heißt Murnaghan-Form einer Matrix A, wenn es eine reguläre Matrix R gibt, so daß

$$M = R^{-1} A R$$

in Murnaghan-Form ist.

Definition 1.1.11

Eine reelle (quadratische) Matrix A ist normal, wenn ihr Kommutator verschwindet, d.h.

$$C(A) = A^{T}A - AA^{T} = 0$$

gilt.

*) Vgl. Fußnote S.3

Satz 1.1.12

Eine reelle Matrix ist genau dann normal, wenn sie orthogonal ähnlich einer Matrix in Murnaghan-Form ist ([25], S.9f).

Satz 1.1.13

Sei A eine normale reelle Matrix. Die Realteile der Eigenwerte *) von A sind die Eigenwerte des symmetrischen Teils

(1.1.15)
$$A^{+} = \frac{1}{2} (A + A^{T}) =: (A_{ij}^{+})_{i,j=1(1)p}$$

von A; die Imaginärteile der Eigenwerte von A sind die Eigenwerte des schiefsymmetrischen Teils

(1.1.16)
$$A^{-} = \frac{1}{2} (A - A^{T}) =: (A_{ij}^{-})_{i,j=1(1)p}$$

von A.

Satz 1.1.14 (I.Schur)
Die Matrix A habe die Eigenwerte *) $\lambda_1, \lambda_2, \ldots, \lambda_n$. Dann gilt

(1.1.17)
$$\sum_{i=1}^{n} |\lambda_{i}|^{2} \leq ||A||^{2} .$$

Das Gleichheitszeichen gilt genau dann, wenn A normal ist ([32]).

Definition 1.1.15

Für jede Matrix A wird die Abweichung von der Normalität $\Delta(A) \geq 0$ (Henrici-Maß) definiert durch

(1.1.18)
$$\Delta^{2}(A) := ||A||^{2} - \sum_{i=1}^{n} |\lambda_{i}|^{2},$$

 λ_i die Eigenwerte *) von A sind (vergl.[13]).

Satz 1.1.16

Zwischen der Abweichung von der Normalität $\Delta(A)$ und dem Kommutator C(A) gelten folgende Beziehungen:

(1.1.19)
$$\Delta^{2}(A) \leq \sqrt{\frac{n^{3}-n}{12}} ||C(A)||$$

*) Vergl. Fußnote S.3

und

$$(1.1.20)$$
 $||C(A)|| \le 2||A|| \Delta(A)$

Den Beweis von (1.1.19) findet man bei Henrici [13], den von (1.1.20) in [18].

Satz 1.1.17

Für jede Matrix A gilt

$$\inf_{\text{T regular}} \Delta(\text{T}^{-1}\text{AT}) = 0.$$

([22]).

Satz 1.1.18

Für jede reelle Matrix A existiert eine reelle normale Matrix N (gleicher Dimension), die dieselben Eigenwerte wie A hat, und für die

$$||A - N|| \leq \Delta(A)$$

gilt ([25],S.14) .

Definition 1.1.19

Eine Folge $(A_k)_{k \in \mathbb{N}}$ zu $A_o := A$ ähnlicher Matrizen konvergiert gegen Normalität, wenn es eine Folge $(N_k)_{k \in \mathbb{N}}$ normaler Matrizen gibt, die dieselben Eigenwerte *) wie A haben, und für die

$$\lim_{k \to \infty} || A_k - N_k || = 0$$

ist.

Satz 1.1.20

Eine Folge $(A_k)_{k\in\mathbb{N}}$ ähnlicher Matrizen konvergiert genau dann gegen Normalität, wenn

$$\lim_{k\to\infty} \Delta(A_k) = 0$$

gilt ([25], S.16), und das ist nach Satz 1.1.16 äquivalent mit

$$\lim_{k\to\infty} ||C(A_k)|| = 0.$$

*) Vgl. Fußnote S.3

Definition 1.1.21

Für eine beliebige reelle Matrix A ist die Abweichung von der Blockdiagonalgestalt definiert durch

(1.1.21)
$$S = S(A) := \left(\sum_{\substack{i,j=1\\i\neq j}}^{p} ||A_{ij}||^{2} \right)^{1/2}$$

Bemerkung 1.1.22

Eine beliebige reelle Matrix A sei zerlegt in

(1.1.22)
$$A = D + E$$
, mit $D = \bigoplus_{j=1}^{p} A_{jj}$.

Dann gilt

(1.1.23)
$$C(A) = C(D) + F, ||F|| \le 4 ||A|| S(A),$$

$$(1.1.24) S(C(A)) \leq 4 ||A|| S(A) und$$

$$(1.1.25)$$
 $||C(A_{jj})|| \le ||C(D)||$, $j = 1(1)p$.

Definition 1.1.23

Eine beliebige reelle Matrix A habe die Eigenwerte *) λ_i , i = 1(1)n. Wir definieren

(1.1.26)
$$\delta = \delta(A) := \min\{|\lambda_i - \lambda_i| | i \neq j\}$$

$$(1.1.27) \qquad \delta^- = \delta^-(A) := \min\{|\operatorname{Im} \lambda_i - \operatorname{Im} \lambda_j| \mid i \neq j\} .$$

Lemma 1.1.24

Für die Lösung der Operator-Gleichung

$$LX = C$$
, mit $LX = AX - XB$,

wobei X, A, B, C 2×2 -Matrizen sind und A und B getrennte Spektren (α_1,α_2) bzw. (β_1,β_2) haben, gilt

$$\|X\| \le F_4(y) \frac{\|C\|}{\min\{|\alpha_i - \beta_i| | i, j \in \{1,2\}\}}$$

$$F_{4}(y) = 1 + y + y^{2} + y^{3} + y^{4} , \quad y = \frac{\Delta(A) + \Delta(B)}{\min\{|\alpha_{i} - \beta_{j}| | i, j \in \{1,2\}\}}$$
 ([20]).

*) Vgl.Fußnote S.3.

1.2 CHARAKTERISTISCHES POLYNOM, NULLSTELLEN

In diesem Paragraphen werden das charakteristische Polynom einer schiefsymmetrischen, sowie symmetrischen 4×4-Matrix aufgestellt und Lösungsverfahren für die Berechnung deren Nullstellen angegeben. Diese Algorithmen werden zur Berechnung der elementaren Veselić-Householder-Matrix in § 2.3 benötigt. Zur Bestimmung eines Parameters der elementaren Johnson-Veselić-Voevodin-Matrix in § 2.1 ist ein Polynom vierten Grades zu minimieren; da dieses Problem auf ein Nullstellenproblem führt, wird es an dieser Stelle vorab behandelt.

1.2.1 Minima eines Polynoms vierten Grades

Gegeben sei ein reelles Polynom

(1.2.1)
$$p(\alpha) = a_4 \alpha^4 + a_3 \alpha^3 + a_2 \alpha^2 + a_1 \alpha + a_0 \ge 0 , \alpha \in \mathbb{R} ,$$

$$a_0, a_1, a_2, a_3, a_4 \in \mathbb{R} .$$

Damit gilt $p(\alpha) \rightarrow \infty$, $\alpha \rightarrow \pm \infty$, oder p ist konstant. Gesucht sind die Minimalstellen von $p(\alpha)$. Sei p nicht konstant; dann sind die Minimalstellen die kleinste und die größte Nullstelle der Ableitung

(1.2.2)
$$p'(\alpha) = 4 a_4 \alpha^3 + 3 a_3 \alpha^2 + 2 a_2 \alpha + a_1$$
.

Falls $a_4 = 0$, folgt wegen $p(\alpha) \ge 0$ auch $a_3 = 0$, und man erhält die einzige Lösung als eindeutige Nullstelle

$$\alpha_{0} = -\frac{a_{1}}{2 a_{2}}$$

von p'. Für $a_4 \neq 0$ sind die kleinste und größte Nullstelle α_1 und α_2 des kubischen Polynoms (1.2.2) zu berechnen *), was in Abschnitt 1.2.3 behandelt wird. Wenn in diesem Fall zwei Minima existieren **), ist das tiefere der beiden zu wählen:

(1.2.4)
$$\alpha_{o} = \begin{cases} \alpha_{1} & \text{falls } p(\alpha_{1}) < p(\alpha_{2}) \\ \alpha_{2} & \text{sonst.} \end{cases}$$

- *) α_1 und α_2 können zusammenfallen
- **) d.h. $\alpha_1 = \alpha_2$ und weder α_1 noch α_2 doppelt.

1.2.2 Nullstellen eines Polynoms vierten Grades

Vorgegeben sei ein reelles Polynom vierten Grades mit Leitkoeffizient eines und nur reellen Nullstellen:

(1.2.5)
$$P_{A}(x) = x^{4} + bx^{3} + cx^{2} + dx + e .$$

Um die Lösungen des Problems

$$(1.2.6) P_{\underline{a}}(\mathbf{x}) = 0$$

zu gewinnen, macht man nach Ferrari den Ansatz ([2],S.45)

$$(1.2.7) 4 P_{\Lambda} (x) + (px+q)^{2} = (2x^{2} + bx + r)^{2} ,$$

wobei p,q,r noch zu bestimmende (reelle) Konstanten sind. Ausmultiplizieren und Koeffizientenvergleich liefert die Bestimmungsgleichungen

$$p^{2} = 4r + b^{2} - 4c ,$$

$$pq = br - 2d ,$$

$$q^{2} = r^{2} - 4e .$$

Daraus erhält man eine kubische Gleichung in r, die sogenannte "Ferrari-Resolvente"

$$(1.2.9) r3 - cr2 + (bd - 4e)r + 4ce - eb2 - d2 = 0 ,$$

deren Wurzeln r_o , ebenso wie p und q nach obiger Voraussetzung über P_4 reell sind (vergl.[1], S.11). Die numerische Berechnung einer Wurzel r_o der Ferrari-Resolvente wird im nächsten Abschnitt behandelt; ist sie gefunden, so lassen sich p und q aus (1.2.8) berechnen, und die Nullstellen x_i , i = 1(1)4 von P_4 ergeben sich mit (1.2.7) zu :

$$x_{1,2} = -\frac{b+p}{4} + \sqrt{\frac{(b+p)^2}{16} - \frac{r_o+q}{2}}$$

$$(1.2.10)$$

$$x_{3,4} = -\frac{b-p}{4} + \sqrt{\frac{(b-p)^2}{16} - \frac{r_o-q}{2}}$$

Da die Nullstellen x_i nach Voraussetzung reell sind, kann ein negativer Radikand nur auf Rundungsfehlern basieren und ist durch Null zu ersetzen. Ergibt sich weiterhin der Radikand als Differenz zweier etwa gleicher Zahlen zu "fast Null", so würde durch das Radizieren die Genauigkeit halbiert; wir setzen deshalb statt $x = \sqrt{y-z}$ den Ausdruck *) $y,z \in \mathbb{R}$:

$$x := \begin{cases} 0 & \text{falls} & y - z < 10^{-13} | y + z | \\ \sqrt{y - z} & \text{sonst.} \end{cases}$$

Entsprechend erfolgt auch die Berechnung von p und q aus (1.2.8). Falls $4r_0 + b^2 - 4c > 10^{-14}$ oder $r_0^2 - 4e > 10^{-14}$, dann falls

$$(4r_o + b^2 - 4c) |r_o^2 + 4e| > (r_o^2 - 4e) |4r_o - b^2 + 4c|$$

$$dann p := \sqrt{4r_o + b^2 - 4c} , q := (br_o - 2d)/p ,$$

$$sonst q := \sqrt{r_o^2 - 4e} , p := (br_o - 2d)/q ,$$

sonst p := 0, q := 0.

1.2.3 Nullstellen eines kubischen Polynoms

Vorgegeben sei ein reelles Polynom dritten Grades mit dem Leitkoeffizenten eins, dessen Nullstellen zu bestimmen sind; d.h. es ist die folgende kubische Gleichung zu lösen:

$$(1.2.11) y3 + by2 + cy + d = 0 ;$$

die Ferrari-Resolvente (1.2.9) hat auch diese Form. Nach ([2],S.10 ff) erhält man daraus mit x = 3y + b die reduzierte Gleichung

$$(1.2.12) x^3 = 3px + q ,$$

mit $p = b^2 - 3c$, $q = 9bc - 2b^3 - 27d$. Eine Wurzel \mathbf{x}_o ergibt sich zu

*) Alle hier auftretenden Schwellen beziehen sich auf die Maschinengenauigkeit $\epsilon_0 = 2^{-52} \approx 0.22 \cdot 10^{-15}$ (vergl.Kap.4) .

$$x_{o} = \sqrt[3]{U} + \sqrt[3]{V},$$

$$U = \frac{1}{2} \left(q + \sqrt{q^{2} - 4p^{3}} \right), \quad V = \frac{1}{2} \left(q - \sqrt{q^{2} - 4p^{3}} \right).$$

Die beiden anderen Wurzeln x_{12} ergeben sich als Wurzeln der quadratischen Gleichung $x^2 + x_0x + (x_0^2 - 3p) = 0$ zu:

$$(1.2.14) x_{12} = -\frac{x_0}{2} \pm \sqrt{3p - \frac{3x_0^2}{4}}$$

Damit sind die Wurzeln der reduzierten kubischen Gleichung (1.2.12) und somit auch die von (1.2.11) bekannt. Es bleibt die Frage, wie man die reellen Wurzeln - denn nur die sind für uns interessant - unter ausschließlicher Verwendung von reeller Arithmetik mit kleinstem Aufwand berechnen kann. Einen ersten Hinweis liefert der folgende Satz von Dörrie ([2],S.31).

Satz 1.2.1

Die kubische Gleichung (1.2.12) hat

drei verschiedene reelle Wurzeln,

eine Doppelwurzel und eine weitere reelle Wurzel *)

oder ein konjugiert komplexes Wurzelpaar und eine reelle Wurzel, je nachdem die Diskriminante

$$(1.2.15) D = q^2 - 4p^3 \begin{cases} < 0 \\ = 0 \\ > 0 \end{cases}$$

ist.

Bereits im Fall ausschließlich reeller Wurzeln (D < 0) wird schon in (1.2.13) komplexe Arithmetik erforderlich. Diese kann man zwar durch Einführung von Polarkoordinaten umgehen, was jedoch die Berechnung eines Arcuscosinus und für jede gesuchte Wurzel die eines weiteren Cosinus erfordert.

Die Wurzeln können zusammenfallen, so daß eine Dreifachwurzel vorliegt.

Ist D>0, so erhält man die einzige reelle Wurzel aus (1.2.13) durch Berechnung von 2 dritten Wurzeln. Sei nun D=0; ist weiter q=0, so ist nach (1.2.15) auch p=0 und die reduzierte kubische Gleichung hat die dreifache Wurzel $\mathbf{x}_0=0$. Für $q\neq 0$ folgt aus (1.2.15) p>0 und nach (1.2.13) ergibt sich die einfache reelle Wurzel zu

(1.2.16)
$$x_0 = 2 \quad \sqrt[3]{\frac{q}{2}} = 2 \sqrt{p} \operatorname{sgn} q$$

Bis auf den Fall mehrfacher Nullstellen sind numerisch aufwendige mathematische Funktionen zu berechnen, die sowohl im Hinblick auf numerische Stabilität als auch auf Rechenaufwand den Einsatz eines Iterationsverfahrens sinnvoll erscheinen lassen, wenn geeignete Startwerte zur Verfügung stehen. Solche lassen sich aus dem folgenden Einschließungssatz für die Wurzeln von (1.2.12) ([2], S.38 f) gewinnen.

Satz 1.2.2

Ist der Koeffizient p positiv, so hat (1.2.12) bei negativer Diskriminante D je eine Wurzel in den Intervallen

$$(-2\sqrt{p}, -\sqrt{p})$$
 , $(-\sqrt{p}, \sqrt{p})$, $(\sqrt{p}, 2\sqrt{p})$

und bei positiver Diskriminante liegt die einzige reelle Wurzel für positives q im Intervall

$$(2 \sqrt{p}, \infty)$$

für negatives q im Intervall

$$(-\infty, -2\sqrt{p})$$
.

Ist p negativ, so liegt die einzige reelle Wurzel zwischen 0 und - q/(3p) .

Betrachtet man (1.2.12) als Polynom

$$(1.2.17) P(x) = x^3 - 3px - q ,$$

und bildet die Ableitung

$$(1.2.18) P'(x) = 3(x^2 - p) ,$$

so sieht man, daß P nur für p>0 relative Extrema hat, und diese liegen bei $\pm\sqrt{p}$. Für p<0 konvergiert also die durch das Newton-Verfahren

(1.2.19)
$$x_{n+1} = x_n - \frac{P(x_n)}{P'(x_n)}$$
 , $n = 0,1,2, ...$

gebildete Folge für $x_0 = -q/(3p)$ monoton und quadratisch gegen die einzige Nullstelle von P . Für p>0 und D>0 konvergiert die entsprechende Folge x_1 , x_2 , ... für $x_0 = 2\sqrt{p}$ sgn q monoton und quadratisch gegen die einzige Nullstelle von P . Für D<0 konvergiert die entsprechende Folge für den Startwert $x_0 = -2\sqrt{p}$ ($x_0 = 2\sqrt{p}$) monoton und quadratisch gegen die kleinste (größte) der drei Nullstellen von P .

Ist nur eine Nullstelle gesucht, so wird diejenige Nullstelle am stabilsten berechnet, die von den beiden anderen den größten Abstand hat. Das Polynom P (1.2.17) ist nach Addition von q eine ungerade Funktion; diejenigen benachbarten Nullstellen von P haben also den größten Abstand, deren zwischenliegendes Extremum den größten Betrag hat. Die Extrema liegen bei $\pm\sqrt{p}$, und es ist

$$P(-\sqrt{p}) = 2 p^{3/2} - q$$
,
 $P(\sqrt{p}) = -2 p^{3/2} - q$.

Für q>0 ist $|P(-\sqrt{p})|<|P(\sqrt{p})|$, also hat die Nullstelle $\widetilde{x}\in(\sqrt{p}$, $2\sqrt{p}$) den größten Abstand von den beiden anderen, und für q<0 ist $|P(-\sqrt{p})|>|P(\sqrt{p})|$, und die Nullstelle $\widetilde{x}\in(-2\sqrt{p}$, $-\sqrt{p})$ hat von den beiden anderen den größten Abstand. Als Startwert für das Newton-Verfahren ist, sofern nur eine beliebige Nullstelle von (1.2.12) gesucht ist, $x=2\sqrt{p}$ sgn q zu wählen.

In Abbildung 1 ist der Programmablaufplan für den Algorithmus zur Bestimmung einer Wurzel der Ferrari-Resolvente (1.2.9) aufgeführt. Damit werden die in Abschnitt 1.2.5 angeführten Ergebnisse erzielt.

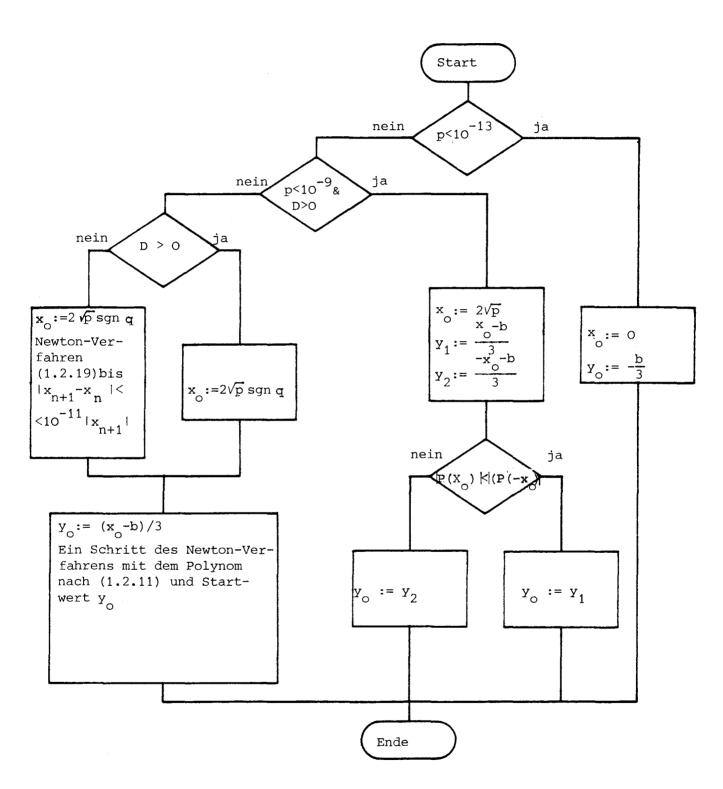


Abb.1 Berechnen einer Wurzel der Ferrari-Resolvente

1.2.4 Charakteristisches Polynom einer reellen schiefsymmetrischen 4×4 - Matrix

Eine schiefsymmetrische reelle 4×4-Matrix hat die allgemeine Form

$$(1.2.20) \quad A = \begin{pmatrix} 0 & a_{12} & a_{13} & a_{14} \\ -a_{12} & 0 & a_{23} & a_{24} \\ -a_{13} & -a_{23} & 0 & a_{34} \\ -a_{14} & -a_{24} -a_{34} & 0 \end{pmatrix}$$

mit dem charakteristischen Polynom

$$(1.2.21) P(\lambda) = \det(A - \lambda I)$$

$$= \lambda^4 + a \lambda^2 + b^2 ,$$

$$a = a_{12}^2 + a_{13}^2 + a_{14}^2 + a_{23}^2 + a_{24}^2 + a_{34}^2 ,$$

$$b = a_{12}a_{34} - a_{13}a_{24} + a_{14}a_{23} .$$

Zur Lösung der Gleichung

$$(1.2.22)$$
 $P(\lambda) = 0$

substituiert man $z = \lambda^2$ und erhält die Lösungen

$$(1.2.23) z_{12} = -\frac{a}{2} \pm \sqrt{\frac{a^2}{4} - b^2}$$

Die Wurzel ist stets reell und nicht größer als a/2, wodurch die Lösungen z_{12} stets negativ und die Lösungen von (1.2.22) $\lambda_1 = \overline{\lambda}_2$, $\lambda_3 = \overline{\lambda}_4$ stets imaginär sind. Für die betragskleineren Eigenwerte steht in (1.2.23) das Pluszeichen, für die betragsgrößeren das Minuszeichen. Ein doppeltes konjugiert komplexes Paar tritt als Lösung von (1.2.23) genau dann auf, wenn

$$(1.2.24)$$
 $a^2 = (2b)^2$

ist, und das ist identisch mit genau einer der beiden folgenden

Bedingungen:

(1.2.25)
$$a_{34} = -a_{12}$$
, $a_{24} = a_{13}$, $a_{23} = -a_{14}$
oder $a_{34} = a_{12}$, $a_{24} = -a_{13}$, $a_{23} = a_{14}$.

Die doppelten, konjugiert komplexen Eigenwerte ergeben sich in diesem Fall zu

$$(1.2.26) \lambda_{1234} = \pm i \sqrt{a_{12}^2 + a_{13}^2 + a_{14}^2} .$$

In der numerischen Praxis wird natürlich keine oder beiden Bedingungen (1.2.25) exakt erfüllt sein, obwohl ein doppelter Eigenwert im Rahmen der Rechengenauigkeit vorliegt. Wenn etwa die Diskriminante

$$D = \frac{a^2}{4} - b^2$$

kleiner als $10^{-13} \left(\frac{a^2}{4} + b^2\right)$ ist, so wird man von einem doppelten Eigenwert sprechen. Den Eigenwert berechnet man aber nicht nach (1.2.26) sondern aus

$$(1.2.27) \quad \lambda_{1234} = \pm i \quad \left\{ \sqrt{\left(\frac{a_{12} - a_{34}}{2}\right)^2 + \left(\frac{a_{13} + a_{24}}{2}\right)^2 + \left(\frac{a_{14} - a_{23}}{2}\right)^2}, a_{12}^{\text{falls}} a_{34} < 0 \right.$$

$$\left(\sqrt{\left(\frac{a_{12} + a_{34}}{2}\right)^2 + \left(\frac{a_{13} - a_{24}}{2}\right)^2 + \left(\frac{a_{14} + a_{23}}{2}\right)^2}, a_{12}^{\text{falls}} a_{34} > 0,$$

weil das arithmetische Mittel bei zufällig gestörten Eingangsvariablen eine bessere Näherung liefert.

1.2.5 Charakteristisches Polynom einer reellen symmetrischen 4×4 - Matrix

Eine symmetrische reelle 4×4-Matrix hat die allgemeine Form

$$(1.2.28) \quad A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{pmatrix}$$

mit dem charakteristischen Polynom

Bei den Koeffizienten sind die Quadrate in b_2 und die eckige Klammer in b_1 invariant unter Spektralverschiebungen *).

Haben mindestens drei Nullstellen von (1.2.29) einen relativ zu ihrem Betrag kleinen Abstand voneinander, so ist die Berechnung nach Abs. 1.2.3 und Abs. 1.2.4 numerisch instabil. Besonders kritisch wird die Berechnung, wenn alle vier Nullstellen relativ zu ihrem Betrag dicht beieinander liegen. Die in [1] durchgeführte Berechnung mit direkten Formeln nach [2] liefert bei einer "fast dreifachen" Nullstelle z.T. nur 5 Dezimalstellen richtig, bei einer "fast vierfachen" sogar nur 2 Dezimalstellen, bei einer Mantissenlänge von 16 Dezimalstellen. Das hier in den vorigen Abschnitten vorgestellte Verfahren liefert demgegenüber zwar wesentlich bessere Ergebnisse, die aber nicht voll befriedigen können.

^{*)} Unter Spektralverschiebung (shift) einer Matrix mit μ verstehen wir die Addition des μ -fachen der Einheitsmatrix : A' = A + μ I .

Wir nehmen deshalb zunächst eine Spektralverschiebung *) mit der Spur der Matrix vor. Damit sind die Probleme bei "fast vierfachen" Nullstellen beseitigt, da der Betrag der Nullstellen von der gleichen Größenordnung wie deren Abstand ist. Lediglich beim Auftreten einer "fast dreifachen" Nullstelle ist nach der Berechnung eine weitere Spektralverschiebung *) mit dem arithmetischen Mittel dieser Nullstellen mit auschließender erneuter Berechnung erforderlich.

Mit diesen Maßnahmen werden alle einfachen und doppelten Nullstellen mit 14 Dezimalstellen genau berechnet, es sei denn, der Abstand von Nullstellen ist kleiner als die Wurzel aus der Rechengenauigkeit; dann ist der Fehler etwa um den Faktor 10 kleiner als der Abstand der Nullstellen. Beim Auftreten einer dreifachen Nullstelle kann es unter ungünstigen Umständen passieren, daß eine doppelte Nullstelle und eine einfache berechnet werden, die um die Wurzel aus der Rechengenauigkeit voneinander entfernt sind.

Beim Auftreten von Nullstellen der Form $\pm c$ und $\pm d$ hat das Polynom eine spezielle Gestalt und es kann einfacher gerechnet werden. Ist in (1.2.29) $b_1 = b_3 = 0$ und $b_2 < 0$, so hat (1.2.29) die Form

$$(1.2.30) P(\lambda) = \lambda^4 + b_2 \lambda^2 + b_0,$$

und das Nullstellenproblem $P(\lambda) = 0$ lautet nach der Substitution $\lambda^2 = x$:

$$(1.2.31) x2 + b2 x + b0 = 0 ,$$

mit den Lösungen

(1.2.32)
$$x_{12} = -\frac{b_2}{2} + \sqrt{\frac{b_2^2}{4} - b_0}$$
, $\lambda_1 = -\sqrt{x_2}$, $\lambda_2 = -\sqrt{x_1}$, $\lambda_3 = \sqrt{x_1}$ $\lambda_4 = \sqrt{x_2}$.

*) Siehe Fußnote S. 17

Da es sich bei $P(\lambda)$ um das charakteristische Polynom einer symmetrischen Matrix handelt, die bekanntlich nur reelle Eigenwerte hat, sind alle Wurzeln in (1.2.32) reell, und es gilt

$$(1.2.33) \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \lambda_4 .$$

In Abb.2 und Abb.3 sind der globale Ablauf zur Berechnung der Nullstellen des charakteristischen Polynoms einer reellen symmetrischen $4\times4-$ Matrix, sowie die Abfrage auf mehrfache Nullstellen und die Auswahl der am dichtesten benachbarten Nullstellen, bzw. der Nullstellen, die den Eigenwerten von A_{11} am nähsten liegen in einem Programmablaufplan dargestellt (vergl.Kap.4).

Diese Form der Berechnung - Wurzel der Ferrari-Resolvente mit Hilfe des Newton-Verfahrens und daraus die Nullstellen des charakteristischen Polynoms mit expliziten Formeln - wird gewählt, weil so die besten numerischen Resultate bei vertretbarem Aufwand erzielt werden können. Bei der Ferrari-Resolvente bietet sich das Newton-Verfahren an, weil geeignete Startwerte zur Verfügung stehen und der Rechenaufwand kleiner ist als bei den expliziten Formeln bei gleichzeitig größerer Genauigkeit, was für die anschließende explizite Berechnung der Nullstellen des charakteristischen Polynoms von großer Wichtigkeit ist. Eine Berechnung der Nullstellen des charakteristischen Polynoms durch ein Iterationsverfahren stößt neben dem größeren Rechenaufwand auf Schwierigkeiten, weil keine geeigneten Startwerte bekannt sind und bei mehrfachen Nullstellen numerische Konvergenzprobleme nicht auszuschließen sind.

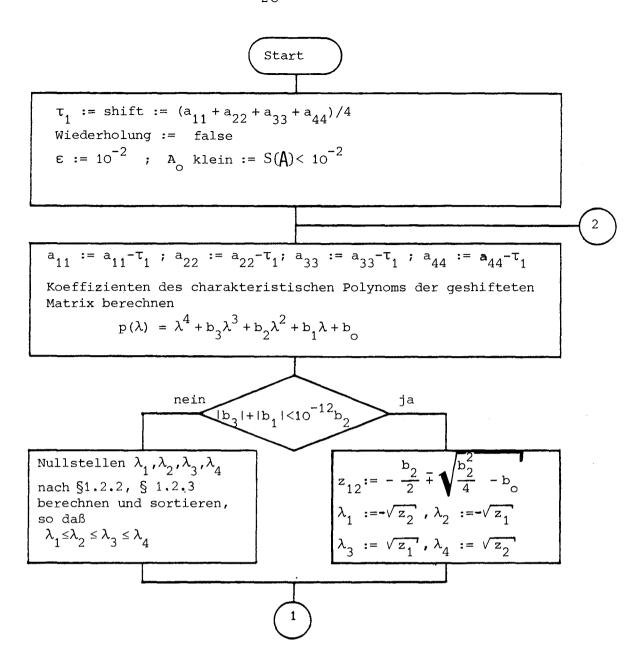
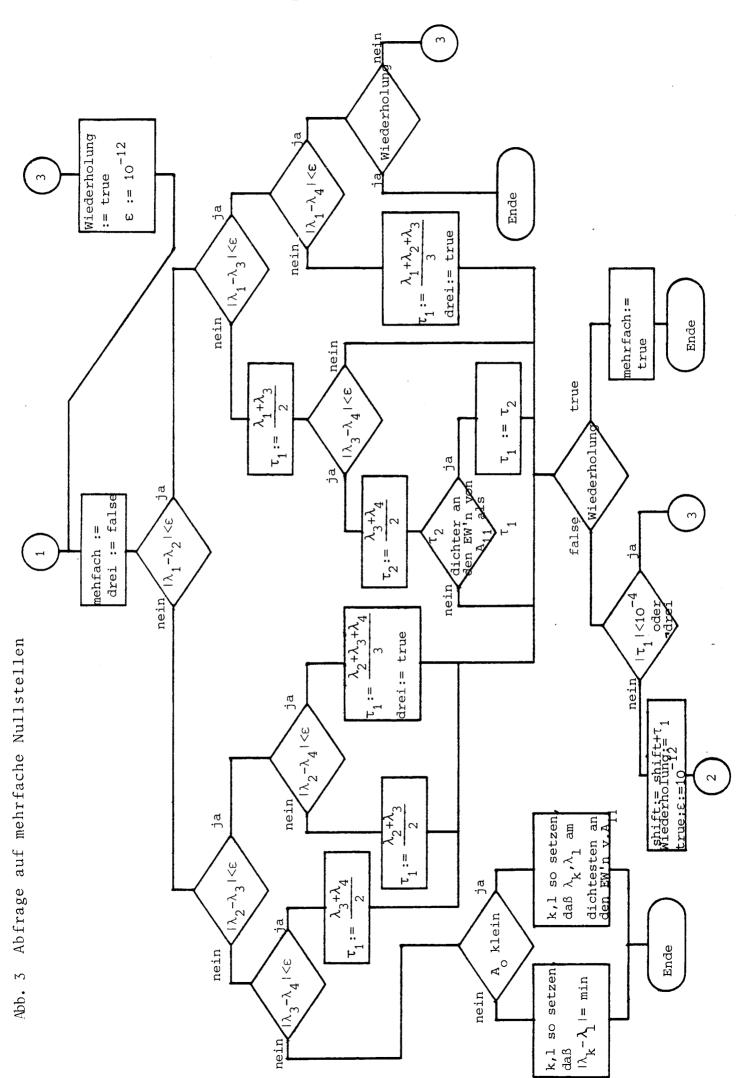


Abb. 2 Berechnen der Nullstellen des charakteristischen Polynoms einer symmetrischen 4×4-Matrix.



2. ELEMENTARE MATRIZEN FÜR JACOBI-ÄHNLICHE VERFAHREN

In diesem Kapitel werden elementare Matrizen behandelt, die bei Verwendung in Jacobi-ähnlichen Verfahren für bestimmte Klassen von Matrizen Konvergenz gegen Blockdiagonalgestalt ermöglichen.

Unter elementaren Matrizen verstehen wir Transformationsmatrizen R, die in der gleichen Form blockgeteilt sind wie A (1.1.2) und sich nur in vier Blöcken, die durch die Pivotindices

$$(2.0.1)$$
 r,s, $1 \le r < s \le p$

bestimmt sind, von der Einheitsmatrix unterscheiden:

$$R = (R_{ij})_{i,j=1(1)p}$$

$$(2.0.2)$$

$$R_{ij} = I\delta_{ij} , 1 \le i, j \le p , \{i,j\} \cap \{r,s\} = \emptyset .$$

Die vier noch freien Blöcke R_{rr}, R_{rs} , R_{sr} und R_{ss} bestimmen die elementare Matrix. Es genügt deshalb, die aus diesen vier Blöcken gebildete Untermatrix

(2.0.3)
$$\widehat{R}_{rs} := \begin{pmatrix} R_{rr} & R_{rs} \\ & & \\ R_{sr} & R_{ss} \end{pmatrix}$$

anzugeben, um eine elementare Matrix zu definieren.

Unter einem Jacobi-ähnlichen Verfahren verstehen wir eine Rechenvorschrift, die eine Folge von elementaren Matrizen

(2.0.4)
$$R_{k} = R(\kappa_{k}; .), \quad k = 1, 2, ..., \\ \kappa_{k} = (r_{k}, s_{k}), \quad 1 \le r_{k} < s_{k} \le p$$

erzeugt, mit deren Hilfe ausgehend von einer Matrix $A_o = A$ eine Folge ähnlicher Matrizen

(2.0.5)
$$A_k := R_k^{-1} A_{k-1} R_k$$
, $k = 1, 2, ...$

gebildet wird.

Man unterscheidet verschiedene Pivotstrategien für die Wahl der Folge $(\kappa_k)_{k\in\mathbb{N}}$. Ist die Folge der Pivotpaare κ_k zyklisch, d.h.

$$(2.0.6) \kappa_{\mathbf{k}+\mathbf{P}} = \kappa_{\mathbf{k}} , \quad \mathbf{k} \in \mathbb{N} ,$$

$$(2.0.7) P = \frac{1}{2} p(p-1) ,$$

so spricht man von allgemeiner zyklischer Pivotstrategie, wenn in je P aufeinander folgenden Folgengliedern κ_k jedes Element der Menge

$$\{(r,s) \mid 1 \le r < s \le p\}$$

vorkommt.

Eine spezielle zyklische Pivotstrategie ist die zeilenzyklische Pivotstrategie, wo die Pivotfolge definiert ist durch:

$$\kappa_{k+1} := (1,2)$$

$$\kappa_{k+1} := \begin{cases} (r_k, s_{k+1}) &, & \text{falls} \quad r_k < p-1 \\ (r_{k+1}, r_{k+2}), & \text{falls} \quad r_k < p-1 \\ (1,2) &, & \text{falls} \quad r_k = p-1 \\ \end{cases}, s_k = p .$$

Alle hier beschriebenen Verfahren benutzen diese Pivotstrategie. Zeichnet man ein Element der Pivotfolge aus, so nennen wir je P aufeinander folgende Folgenglieder, die mit diesem Pivotpaar beginnen, einen Zyklus.

- 2.1 DIE ELEMENTARE JOHNSEN-VESELIĆ-VOEVODIN-MATRIX
- 2.1.1 Definition und normreduzierende Eigenschaften

Sei
$$\widehat{T}_{rs}$$
 von der Form
$$\widehat{T}_{r,s} = \begin{pmatrix} I & X \\ 0 & I \end{pmatrix},$$

$$(2.1.1)$$

$$T = T(X) = T(\kappa; X), \kappa = (r,s),$$

so erhält man die elementare Johnsen-Veselić-Voevodin-Matrix. Eine Matrix dieser Form, jedoch mit I=1 und $X\in \mathbb{R}$, wurde zuerst von Voevodin [39] angegeben und in einem Jacobi-ähnlichen Verfahren zur Normreduzierung benutzt. Johnsen [17] und Veselić [34] fanden gleichzeitig und unabhängig voneinander die Verallgemeinerung (2.1.1) und benutzten sie zur Normreduzierung in Jacobi-ähnlichen Verfahren (s. auch Veselić und Wenzel [38]). Eine ausführliche Diskussion der normreduzierenden Eigenschaften findet man bei Veselić [34], wo auch die elementaren Matrizen

(2.1.2)
$$7 = 7(\kappa; Y) = T^{T}(\kappa; Y^{T})$$

für beliebige Blockteilung eingeführt werden.

Betrachtet man die elementaren Matrizen T (bzw.Z) – im folgenden kurz T – (Z-) Matrizen genannt – für einen festen Pivot κ = (r,s), so erhält man auf der Menge dieser Matrizen eine, für die weiteren Überlegungen wichtige Struktur.

Bemerkung 2.1.1

Die Abbildung $X \mapsto T(X) = T(\kappa; X)$ ist für festes κ ein injektiver Gruppen-Homomorphismus von der abelschen Gruppe der 2×2 -Matrizen bezüglich der Addition in die Gruppe der $n\times n$ -Matrizen bezüglich der Multiplikation. Die elementaren T-Matrizen bilden deshalb bei festem r und s eine vierparametrige, kommutative Untergruppe der $n\times n$ -Matrizen bezüglich der Multiplikation, und es gilt

$$T(X + Y) = T(X) \cdot T(Y)$$
, $T(0) = I$, $T(-X) = T^{-1}(X)$.

Die obige Bemerkung gilt ebenso wie alle weiteren Resultate dieses Abschnitts für die elementaren Z-Matrizen entsprechend. Eine Ähnlichkeitstransformation mit einer elementaren T-(Z-) Matrix nennen wir im Folgenden T-(Z-) Transformation.

Nach einer T-Transformation $A' = T^{-1}AT$ am Pivot $\kappa = (r,s)$ gilt:

$$A'_{rj} = A_{rj} - XA_{sj}, \quad j \neq s,$$

$$A'_{is} = A_{is} + A_{ir}X, \quad i \neq r,$$

$$A'_{rs} = A_{rs} + A_{rr}X - XA_{ss} - XA_{sr}X,$$

$$A'_{ij} = A_{ij} \text{ sonst.}$$

Zur Erörterung der normreduzierenden Eigenschaften der T-Transformation definieren wir

$$(2.1.4) f(X) = || T^{-1}(X) A T(X) ||^2 ,$$

und erhalten nach [34]*):

(2.1.5)
$$f(X) - f(0) = 2 < C_{rs}, X > + \sum_{i \neq r}^{p} \|A_{ir}X\|^{2} + \sum_{j \neq s}^{p} \|XA_{sj}\|^{2} + \|A_{rr}X - XA_{ss}\|^{2} - 2 < A_{rs}, XA_{sr}X > - 2 < XA_{sr}X, A_{rr}X - XA_{ss} > + \|XA_{sr}X\|^{2}.$$

Die folgenden Eigenschaften der Funktion f wurden von Veselić[34] hergeleitet.

*) Wir schreiben
$$\sum_{i=r}^{p}$$
 ... als Abkürzung für $\sum_{i=1}^{p}$ i #r

Satz 2.1.2

Sei $X_0 \neq 0$ fest. Hat die Funktion

$$\alpha \mapsto f(\alpha X_0)$$

kein Minimum in IR, dann ist die Funktion

$$\alpha \mapsto T^{-1}(\alpha X_{o}) A T (\alpha X_{o})$$

auf $\mathbb R$ konstant. Hat die Matrix $\mathbb A$ getrennte Eigenwerte *), dann hat die Funktion $\alpha \longmapsto f(\alpha X_0)$ mindestens ein Minimum für jedes $X_0 \neq 0$, und es gilt

$$\lim_{\|X\|\to\infty} f(X) = +\infty .$$

Satz 2.1.3

Seien A beliebig, C = C(A) gemäß (1.1.2), (1.1.5) und die Pivot-Indices r,s fest. Dann gilt für jedes X

(2.1.6)
$$\operatorname{grad}_{X} || T^{-1}(X) A T(X) ||^{2} = 2[C(T^{-1}(X) A T(X))]_{r,s}$$

oder mit den Bezeichnungen (2.1.3), (2.1.4):

(2.1.7)
$$\operatorname{grad}_{X} f(X) = 2C'_{rs}$$
.

Wenn man eine Matrix X_o findet, die die Funktion f minimiert, so verschwindet an dem entsprechenden Pivot der Kommutator der transformierten Matrix. Das absolute Minimum der Funktion f ist – wenn überhaupt – nicht leicht zu berechnen; deshalb wird man sich mit Näherungen begnügen müssen. Dazu werden vier Vorschläge gemacht, von denen der vierte ein iterativer Prozess zur beliebig genauen Bestimmung eines lokalen Minimums ist. In Anlehnung an Voevodin [39] schlägt Veselić [34] eine Näherung vor, die globale Konvergenz gegen Normalität garantiert, aber keine asymptotisch quadratische Konvergenz ermöglicht. Dazu ist in [34] der folgende Satz bewiesen, der die monotone Abnahme von || A|| garantiert.

*) Vergl. Fußnote S. 3

Satz 2.1.4

Sei $C_{rs} \neq 0$ und $T = T((r,s);X_{o})$. Für

(2.1.8)
$$X_o := \frac{-C_{rs}}{\max{\{\|C_{rs}\|, m_{rs}\}}}$$
,

(2.1.9)
$$m_{rs} = \sum_{i \neq r}^{p} \|A_{ir}\|^{2} + \sum_{j \neq s}^{p} \|A_{sj}\|^{2}$$

$$+ (\|A_{rr}\| + \|A_{ss}\| + \|A_{sr}\|)^{2} + \|A_{sr}\|^{2} + \|A_{rs}\|^{2}$$

gilt die Abschätzung:

Bei der Wahl (2.1.8) von X_{o} findet man eine Näherung für ein Minimum von f, indem man ein festes Stück in entgegengesetzter Richtung des Gradienten geht und eine Abnahme von ||A|| erreicht. Eine stärkere Reduzierung der Norm erhält man, wenn man in Richtung des Gradienten das absolute Minimum von f findet, d.h.

(2.1.11)
$$\min\{p(\alpha) \mid \alpha \in \mathbb{R}\}$$
, $p(\alpha) := f(\alpha C_{rs})$,

wobei $p(\alpha) \ge 0$ ein Polynom vom Grade 4 in α ist. Das absolute Minimum von p kann man numerisch nach dem in Abschnitt 1.2.1 beschriebenen Verfahren berechnen. Mit der daraus gewonnenen Näherung $X = \alpha C_{rs}$ wird in [38] gerechnet, und eine schnellere Normabnahme beobachtet als bei der Wahl von X_{o} aus (2.1.8) als Transformationsparameter. Wir setzen hier

$$(2.1.12) X_1 := \begin{cases} \alpha C_{rs}, & \text{falls } |\alpha| \leq 2 \min \left\{ \frac{1}{\|C_{rs}\|}, \frac{1}{m_{rs}} \right\}; \\ X_0 & \text{sonst} \end{cases}$$

diese Einschränkung ist beweistechnischer Natur, um den Transformationsparameter X_1 nach oben abschätzen zu können.

Nach Konstruktion von X_1 gilt offensichtlich

$$(2.1.13)$$
 $0 \le f(X_1) \le f(X_0)$,

und es gilt für X_1 die Abschätzung (2.1.10) entsprechend. Damit folgt wegen $m_{rs} \le 4 \|\mathbf{A}\|^2$:

$$(2.1.14) \qquad \|A\|^{2} - \|T^{-1}(X_{v})AT(X_{v})\|^{2} \ge \min \left\{ \|C_{rs}\|, \frac{\|C_{rs}\|^{2}}{4\|A\|^{2}} \right\} \qquad , \ v \in \{0,1\} \ .$$

Der Index ν deutet in diesem und dem nächsten Kapitel stets an, daß eine T-Transformation mit dem Parameter X_{ν} durchgeführt wird.

An dieser Stelle wollen wir bemerken, daß sich analog zu dem von Hari [8] für die elementare Matrix von Voevodin [39] bewiesenen, auch für ein Jacobi-ähnliches Verfahren mit der elementaren T-Transformation mit Parameter $X_{o}(\text{oder }X_{1})$ und allgemeiner zyklischer Pivotstrategie die Konvergenz gegen blockdiagonalen Kommutator, d.h.

$$S(C(A_k)) \rightarrow 0 , k \rightarrow \infty$$

beweisen läßt. Wird nach jedem Zyklus eine Eberlein-Jacobi-Transformation (s.§ 2.4) auf jeden Diagonalblock angewendet, so konvergiert die Folge (A_k) gegen Normalität.

Obwohl die Konvergenz bei Verwendung von X_1 als Parameter schneller ist als bei X_0 , wird keine quadratische Konvergenz erreicht, wenn man die T-Transformation mit einer, quadratische Konvergenz ermöglichenden, orthogonalen Ähnlichkeitstransformation kombiniert (vgl. Kap.4, [38],[34]). In den nächsten beiden Abschnitten beschäftigen wir uns deshalb mit Parametern X_{v} , $v \in \{2,3\}$, die quadratische Konvergenz ermöglichen.

2.1.2 Der Parameter X_2 für quadratische Konvergenz

Eine Näherung für die Lösung des Minimierungsproblems von f (s.(2.1.4)), die eine asymptotisch quadratische Konvergenz ermöglicht – aber keine globale Konvergenz garantiert – wurde von Veselić [34] angegeben. Dabei wird nicht die Funktion f (s.(2.1.5)), sondern ein quadratisches Polynom

$$\varphi(X) = f(X) + O(\|X\|^3) ,$$

das man durch Taylor-Entwicklung von f erhält, minimiert, was exakt möglich ist. Es ist

(2.1.15)
$$\varphi(X) = \|A\|^2 + 2 < C_{rs}, X > + < KX, X > ,$$

mit

$$(2.1.16) K = L^{T}L + \mathcal{D} , \quad \mathcal{D} X = XA + BX - HX ,$$

$$LX = A_{rr} X - XA_{ss} ,$$

$$A = \sum_{j \neq s}^{p} A_{sj}A_{sj}^{T} , \quad B = \sum_{i \neq r}^{p} A_{ir}^{T}A_{ir} ,$$

$$HX = A_{sr}^{T} X^{T}A_{rs} + A_{rs} X^{T}A_{sr}^{T} .$$

Eine einfache Rechnung zeigt, daß die Operatoren K, L^TL , \mathcal{H} , A und B. symmetrisch und L^TL , .A und B. mindestens positiv semidefinit sind. Der Gradient $\operatorname{grad}_X \phi(X)$ verschwindet genau in den Lösungen der Gleichung

$$(2.1.17)$$
 $KX_2 = -C_{rs}$

(vergl. [34]). Jede 2×2 -Matrix X, auf der die Funktion ϕ ein Minimum annimmt, erfüllt also die Gleichung (2.1.17). Die folgenden Lemmata zeigen, daß unter gewissen Voraussetzungen über die Matrix A eine eindeutige Lösung der Operatorgleichung (2.1.17) existiert.

Lemma 2.1.5

Sei A eine reelle Blockmatrix mit den Eigenwerten *) λ_i , i = 1(1)n. Es gelte $\delta > 0$ (δ aus (1.1.26)), p > 1,

(2.1.18)
$$S(A) \leq \frac{\delta}{16\sqrt{p-1}}$$
,

(2.1.19)
$$\|C(D)\| \leq \frac{\delta^2}{64}$$
 (vergl.(1.1.22)).

Dann gilt für den Abstand der Eigenwerte $\{\mu_{j}^{(i)} | j \in \{1,2\}\}$ der Diagonalblöcke A_{ii} , i = 1(1)p von A:

$$(2.1.20) \quad \min\{|\mu_{j}^{(i)} - \mu_{1}^{(k)}| | j, 1 \in \{1, 2\}, 1 \le i < k \le p\} > \frac{\delta}{2}.$$

*) Vergl. Fußnote S.3

Beweis: Für n = 2 folgt aus Satz 1.1.16:

$$(2.1.21) \Delta(A_{ii}) \leq \sqrt[4]{\frac{1}{2}} \|C(A_{ii})\|^{1/2} \leq 0.9 \frac{\delta}{8} ,$$

wegen $\|C(A_{ii})\| \le \|C(D)\|$ und Voraussetzung (2.1.19). Das Spektrum von A ist nach Meyer und Veselić [21] enthalten in der Vereinigung

$$G = \bigcup_{i=1}^{p} G_{i}, G_{i} = K_{1}^{(i)} \cup K_{2}^{(i)}, i = 1(1)p,$$

$$(2.1.22) \qquad K_{j}^{(i)} = \left\{ z \in \mathbb{C} | | \mu_{j}^{(i)} - z | \leq \Delta(A_{ii}) + \sum_{k \neq i}^{p} \|A_{ik}\| \right\},$$

$$j \in \{1, 2\}, i = 1(1)p.$$

Wegen $\sum_{k=1}^{p} \|A_{ik}\| \le \sqrt{p-1} S(A)$ ist mit (2.1.21) und (2.1.18):

$$K_{\mathbf{j}}^{(i)} \subset \widetilde{K}_{\mathbf{j}}^{(i)} = \left\{ z \in \mathbb{C} \mid |\mu_{\mathbf{j}}^{(i)} - z| \leq \frac{7\delta}{40} \right\}$$
.

Da der Radius der Kreise $\widetilde{K}_{j}^{(i)}$ kleiner ist als $\delta/4$, enthält jeder Kreis genau einen Eigenwert von A, und die Kreise sind paarweise disjunkt. Sie sind zentriert um einen Eigenwert von A_{ii} , so daß für den Abstand der Eigenwerte der Diagonalblöcke A_{ii} die Behauptung (2.1.20) erfüllt ist.

Lemma 2.1.6

Unter den Voraussetzungen von Lemma 2.1.5 sind die Operatoren $L^{\mathbf{T}}L$ und K positiv definit, mit

und es gilt

$$(2.1.24)$$
 $\|X_2\| \le \frac{64}{3\delta^2} \|C_{rs}\|$.

Beweis: Wegen Lemma 2.1.5 sind die Spektren von A_{rr} und A_{ss} getrennt, so daß Lemma 1.1.24 anwendbar ist, und es folgt:

$$\|A_{rr}X - XA_{ss}\| \ge \frac{\delta}{4} \|X\|$$
.

Daraus ergibt sich

$$< L^{T}LX, X> = \|A_{rr}X - XA_{ss}\|^{2} \ge \frac{\delta^{2}}{16} \|X\|^{2}$$

und

$$< KX , X > \ge < L^{T}LX , X > - < HX , X >$$

$$\ge \frac{\delta^{2}}{16} \|X\|^{2} - \frac{\sqrt{2}\delta^{2}}{256} \|X\|^{2} \ge \frac{3\delta^{2}}{64} \|X\|^{2}$$

sowie

$$\|X_2\| \le \|K^{-1}\| \|C_{rs}\| \le \frac{64}{3\delta^2} \|C_{rs}\|$$
.

Aus der positiven Definitheit von K folgt das

Korollar 2.1.7.

Unter den Voraussetzungen von Lemma 2.1.5 ist

$$X_2 = -K^{-1} C_{rs}$$

das eindeutige Minimum von ϕ .

In dem folgenden Satz 2.1.8 beweisen wir,unter Anwendung der obigen Lemmata, Eigenschaften einer T-Transformation mit dem Parameter X_2 , die zur Konstruktion eines Jacobi-ähnlichen Verfahrens mit asymptotisch quadratischer Konvergenz erforderlich sind (vgl. Veselić[34], Satz 3.7).

Satz 2.1.8

Sei A eine reelle Blockmatrix mit $\delta > 0$, und

$$\begin{array}{c} \varepsilon < \min \{ \varepsilon_0, \varepsilon_1, \varepsilon_3 \} \text{ , } \varepsilon_0 := \frac{\delta}{16\sqrt{p-1}\|A\|} \text{ ,} \\ \\ \varepsilon_1 := \frac{\delta^2}{64\|A\|^2} \text{ , } \varepsilon_3 := \left[4\|A\|^4 \left(\frac{9710}{\delta^6} \|A\|^2 + \frac{74}{\delta^4} \right) \right]^{-\frac{1}{2}} \text{ ,} \end{array}$$

(2.1.26)
$$S(A) \leq \frac{1}{4} |A| \epsilon$$
,

(2.1.27)
$$\|C(D)\| \le \frac{\delta^2}{64}$$

und

(2.1.28)
$$A' = T^{-1}((r,s);X_2) AT((r,s);X_2)$$
, X_2 aus (2.1.17).

Dann gilt:

$$(2.1.29) \|C'_{rs}\| \le \frac{128^2 (72\sqrt{2} + 1)}{54 \delta^4} \|A\|^3 S^3(A)$$

und

$$\|A\|^2 - \|A'\|^2 \ge \frac{\|C_{rs}\|^2}{4\|A\|^2}$$

Beweis: Nach Bemerkung 1.1.22 ist mit Voraussetzung (2.1.26):

(2.1.31)
$$S(C(A)) \le 4|A|S(A) \le |A|^2 \epsilon$$
.

Nach Satz 2.1.3 und (2.1.17) gilt mit (2.1.28):

$$2C'_{rs} = [grad_{x}f(X)]_{X=X_{2}} = [grad_{x}(f(X) - \phi(X))]_{X=X_{2}}$$

$$= 2(-A^{T}_{rr}X_{2}A_{sr}X_{2} + X_{2}A_{sr}X_{2}A^{T}_{ss} + (X_{2}A_{ss} - A_{rr}X_{2})X_{2}^{T}A^{T}_{sr} + A^{T}_{sr}X_{2}^{T}(X_{2}A_{ss} - A_{rr}X_{2}) + A^{T}_{rs}X_{2}^{T}X_{2}A_{rs}X_{2} + X_{2}A_{rs}X_{2}X_{2}^{T}A^{T}_{rs})$$

und damit wegen $\|X_2\| \le \frac{64}{3\delta^2} \|C_{rs}\|$ und den Voraussetzungen des Satzes:

$$\begin{split} \|C_{rs}^{\dagger}\| & \leq 3 \left(\|A_{rr}\| + \|A_{ss}\|\right) \|A_{sr}\| \|X_{2}\|^{2} + 2\|A_{rs}\|^{2} \|X_{2}\|^{3} \\ & \leq 3\sqrt{2} \|A\| S(A) \frac{64^{2}}{9\delta^{4}} \|C_{rs}\|^{2} + 2 S^{2}(A) \frac{64^{3}}{27\delta^{6}} \|C_{rs}\|^{3} \\ & \leq \frac{16\sqrt{2} 64^{2}}{3\delta^{4}} \|A\|^{3} S^{3}(A) + \frac{4^{4}64^{3}}{54\delta^{6}} \|A\|^{3} S^{5}(A) \\ & \leq \frac{128^{2} (72\sqrt{2} + 1)}{54\delta^{4}} \|A\|^{3} S^{3}(A) , \end{split}$$

womit (2.1.29) bewiesen ist.

Nach (2.1.16) ist

$$K = L^{T}L + D ,$$

mit

$$\| \mathcal{D} \| \le 3 \, S^2 (A) \le \frac{3}{16} \| A \|^2 \varepsilon^2 \le \frac{3 \, \delta^2}{16^3}$$
.

Mit (2.1.23) gilt dann $\|v\| \le \frac{\delta^2}{16} \le \|(L^T L)^{-1}\|^{-1}$, und der Satz über die Inverse benachbarter Operatoren ([19], S. 108) ist auf K und $L^T L$ anwendbar. Danach gilt

$$(2.1.33) K^{-1} = (L^{T}L)^{-1} + D' ,$$

mit

$$\| \mathcal{D}' \| \leq \frac{\| (L^{T}L)^{-1} \|^{2}}{1 - \| (L^{T}L)^{-1} \| \| \mathcal{D} \|} \| \mathcal{D} \|$$

$$\leq \frac{3 \cdot 16}{\delta^{4} (1 - \frac{3}{256})} \| A \|^{2} \varepsilon^{2}$$

$$\leq \frac{50 \| A \|^{2}}{\delta^{4}} \varepsilon^{2} .$$

Aus (2.1.5) folgt mit (2.1.16) und (2.1.33):

$$f(0) - f(X_{2}) = \langle C_{rs}, K^{-1}C_{rs} \rangle - \|X_{2}A_{sr}X_{2}\|^{2}$$

$$+ 2\langle X_{2}A_{sr}X_{2}, A_{rr}X_{2} - X_{2}A_{ss} \rangle$$

$$= \|(L^{T})^{-1}C_{rs}\|^{2} + K ,$$

wobei sich K unter Anwendung von $\|X_2\| \le \frac{64}{3\delta^2}$ $\|C_{rs}\|$, (2.1.31) und (2.1.33) abschätzen läßt durch:

$$\begin{split} \|K\| &\leq \sqrt{2} \left(\|\mathcal{D}'\| \|C_{rs}\|^2 + 2 \left(\|A_{rr}\| + \|A_{ss}\| \right) \|A_{sr}\| \|X_2\|^3 \right) + \|A_{sr}\|^2 \|X_2\|^4 \\ &\leq \left(\frac{74}{\delta^2} + \frac{9710}{\delta^6} \|A\|^2 \right) \|A\|^2 \varepsilon^2 \|C_{rs}\|^2 \\ &\leq \frac{\|C_{rs}\|^2}{4 \|A\|^2} \leq \frac{1}{2} \|(L^T)^{-1} C_{rs}\|^2 \quad . \end{split}$$

Die letzte Ungleichung folgt dabei aus

$$\|L^{T}X\| \leq (\|A_{rr}^{T}\| + \|A_{ss}^{T}\|)\|X\| \leq \sqrt{2}\|A\|\|X\| ,$$

mit $X = (L^T)^{-1}C_{rs}$. Zusammen folgt dann die Behauptung (2.1.30).

Wenden wir uns nun der vierten Wahl des normreduzierenden Parameters X_3 einer T-Transformation zu.

2.1.3 Ein verbesserter Parameter X_3 für quadratische Konvergenz

Durch wiederholte Durchführung einer T-Transformation mit dem Parameter X_2 und festen Pivot-Indices (r,s) erreicht man dasjenige lokale Minimum der Funktion f, dessen Matrix X_3 , auf der es angenommen wird, minimale Norm hat. Die zugehörige T-Matrix hat damit die kleinste Abweichung von der Einheitsmatrix unter allen T-Matrizen, die $\|A'\|$ minimieren. Falls mehrere Minima von f existieren, kann allerdings nicht garantiert werden, daß es sich bei dem gefundenen Minimum um das absolute Minimum von f handelt. Die Konvergenz des oben angedeuteten Verfahrens ist, wie wir später sehen werden (s. Satz 2.1.9), lokal quadratisch.

Wegen T(X)T(Y) = T(X+Y) ist die Anwendung der T-Transformation nicht in jedem Iterationsschritt, sondern nur am Schluß einer Iterationsfolge einmal erforderlich, wodurch der Rechenaufwand erheblich reduziert wird. Das setzt allerdings voraus, daß bekannt ist, wie sich der Operator K und der Kommutator C_{rs} unter einer T-Transformation ändern (vergl.(2.1.16)). Unter Beachtung der Transformationsformeln (2.1.3) ergibt sich, daß die Matrizen A und B aus (2.1.16) unter T-Transformationen invariant sind. Setzt man

(2.1.34)
$$A(X) = (A_{ij}(X)) := T^{-1}(X) A T(X),$$

$$(2.1.35) \qquad F: X \mapsto C_{rs}(X) := [C(A(X))]_{r,s},$$

(2.1.36)
$$K(X)H := HA + BH + L^{T}(X)L(X)H - H(X)H$$

$$L(X)H := A_{rr}(X)H - HA_{ss}(X)$$

$$H(X)H := A_{sr}^{T}H^{T}A_{rs}(X) + A_{rs}(X)H^{T}A_{sr}^{T},$$
A und B wie in (2.1.16),

so lautet das Iterationsverfahren:

(2.1.37)
$$\widetilde{X}^{(k)} := \begin{cases} 0 & \text{für } k = 0 \\ \widetilde{X}^{(k-1)} + X_2^{(k-1)} & \text{für } k \ge 1 \end{cases}$$

$$(2.1.38) X_2^{(k)} := -K^{-1}(\tilde{X}^{(k)}) C_{rs}(\tilde{X}^{(k)}), k \ge 0.$$

Wegen T(X)T(Y) = T(X+Y) und $T^{-1}(X) = T(-X)$ gilt

(2.1.39)
$$A(\tilde{X}^{(k+1)}) = T^{-1}(X_2^{(k)}) A(\tilde{X}^{(k)}) T(X_2^{(k)}),$$

und man erkennt an (2.1.36), (2.1.3), daß lediglich die Blöcke $A_{rr}(\widetilde{X}^{(k)})$, $A_{ss}(\widetilde{X}^{(k)})$ und $A_{rs}(\widetilde{X}^{(k)})$ berechnet werden müssen, um den Operator $K(\widetilde{X}^{(k)})$ bestimmen zu können. Der Pivotblock des Kommutators der transformierten Matrix ergibt sich nach (1.1.5) und (2.1.3) zu

$$(2.1.40) C_{rs}(X) = \widetilde{C}_{rs} + \widetilde{B}X + X\widetilde{A} + A_{rr}^{T}(X)A_{rs}(X) - A_{rr}(X)A_{sr}^{T}$$
$$+ A_{sr}^{T} A_{ss}(X) - A_{rs}(X)A_{ss}^{T}(X)$$

mit

$$(2.1.41) \qquad \widetilde{C}_{rs} = \sum_{i \neq r, s}^{p} A_{ir}^{T} A_{is} - A_{ri} A_{si}^{T} ,$$

$$(2.1.41) \qquad \widetilde{A} = \sum_{i \neq r,s}^{\underline{p}} A_{si} A_{si}^{T}, \quad \widetilde{B} = \sum_{i \neq r,s}^{\underline{p}} A_{ir}^{T} A_{ir},$$

wobei \widetilde{C}_{rs} , \widetilde{A} und \widetilde{B} von X unabhängig sind. Wie ein Vergleich mit (1.1.6) bzw.(2.1.16) zeigt, unterscheiden sich \widetilde{C}_{rs} , \widetilde{A} und \widetilde{B} von C_{rs} , A und B nur durch das Fehlen von ein oder zwei Summanden, so daß kein zusätzlicher Rechenaufwand entsteht. Über die von X unabhängigen Terme hinaus werden auch hier nur die Blöcke $A_{rr}(\widetilde{X}^{(k)})$, $A_{ss}(\widetilde{X}^{(k)})$ und $A_{rs}(\widetilde{X}^{(k)})$ benötigt, so daß der Rechenaufwand der Iteration (2.1.37 ... 39) von der Dimension der Matrix unabhängig ist. Ist für ein $k \in \mathbb{N}$ die gewünschte Genauigkeit erreicht, etwa $\|C_{rs}(\widetilde{X}^{(k)})\| < 10^{-16}$, so wird eine T-Transformation mit dem Parameter $\widetilde{X}^{(k)}$ auf der Matrix A durchgeführt.

Eine elementare Rechnung zeigt, daß das oben angeführte Iterationsverfahren ein Newton-Verfahren zur Bestimmung einer Nullstelle von F ist, und unter den Bedingungen

$$S(A) \leq \frac{1}{4} \|A\| \min\{\epsilon_0, \epsilon_2\}, \|C(D)\| \leq \frac{\delta^2}{64},$$

$$\epsilon_2 := \frac{3\delta^2}{512(4+\sqrt{2}) \|A\|^2}$$

die Voraussetzungen des Satzes von Newton-Kantorowitsch in der von Ortega [23] bewiesenen Form erfüllt sind. Damit gilt der

Satz 2.1.9.

Sei A eine reelle Blockmatrix mit $\delta > 0$, die die Voraussetzungen (2.1.42) erfüllt. Dann sind die Newton-Iterierten $\widetilde{X}^{(k)}$ (s.(2.1.37)) wohldefiniert, konvergieren quadratisch gegen die in $\{X \mid \|X\| \le t^*\}$ eindeutige Lösung \widetilde{X} von $C_{rs}(X) = 0$, mit

$$t^* \ge \frac{\delta^2}{64(4+\sqrt{2})\|A\|S(A)} - \frac{32}{\delta^2}\|C_{rs}\| \ge \frac{13}{6}$$
,

und es gilt:

(2.1.43)
$$\|\tilde{X}\|$$
, $\|\tilde{X}^{(k)}\| \le \frac{32}{\delta^2} \|C_{rs}\|$, $k \ge 0$.

Bemerkung 2.1.10

Der Operator $K(\widetilde{X})$ ist positiv definit; deshalb wird das in der Kugel $\|X\| \le t^*$ eindeutige lokale Minimum von f auf der Lösung \widetilde{X} angenommen. Dieses Minimum ist dadurch charakterisiert, daß es, falls mehrere lokale Minima existieren, dasjenige ist, für das $\|X\|$ minimal wird. Weiterhin sei bemerkt, daß $t^* \to \infty$ für $S(A) \to 0$.

Nach diesen Vorbereitungen können zu Satz 2.1.8 äquivalente Aussagen auch für die Wahl von

$$(2.1.44)$$
 $X_3 := \tilde{X}$

als Transformationsparameter gezeigt werden.

<u>Satz 2.1.11</u>

Sei A eine reelle Blockmatrix mit $\delta > 0$, und

$$\begin{split} \varepsilon &< \min \{ \varepsilon_0, \varepsilon_2, \varepsilon_3 \} \quad , \\ S(A) &\leq \frac{1}{4} \| A \| \varepsilon \quad , \| C(D) \| \leq \frac{\delta^2}{64} \quad , \\ A' &= T^{-1} \left((r,s); X_3 \right) \; A \; T \left((r,s); X_3 \right) \quad . \end{split}$$

Dann gilt

$$(2.1.45)$$
 $\|C'_{rs}\| = 0$,

$$\|A\|^2 - \|A'\|^2 \ge \frac{\|C_{rs}\|^2}{4\|A\|^2}$$

Beweis: Wir bemerken zuerst, daß die Voraussetzungen aller bisherigen Lemmata und Sätze in den obigen Voraussetzungen enthalten sind, und damit alle bisher erzielten Resultate angewendet werden können.

Die Aussage (2.1.45) ist eine unmittelbare Folgerung aus Satz 2.1.9. Weiter gilt (s.(2.1.24) und Satz 2.1.9):

$$\|X_2\| \le \frac{64}{3\delta^2} \|C_{rs}\| \le \frac{1}{3} < \frac{13}{6} < t^*$$
,

und es folgt wegen der Eindeutigkeit des Minimums von f in $\|X\| \le t^*$ die Behauptung (2.1.46) aus Satz 2.1.8. Damit ist Satz 2.1.11 bewiesen.

Für den Konvergenzbeweis in § 3.2 werden noch die beiden folgenden Abschätzungen für die Änderung des Pivot-Blocks sowie des außerblock-diagonalen Teils einer Matrix durch eine T-Transformation mit dem Parameter X_2 bzw. X_3 benötigt.

<u>Lemma 2.1.12</u>

Die Voraussetzungen von Satz 2.1.11 seien erfüllt.

Mit
$$A' = T^{-1}((r,s); X_v) A T((r,s); X_v)$$
, $v \in \{2,3\}$ gilt:

$$(2.1.47)$$
 $A'_{rs} = A_{rs} + F$,

$$(2.1.48) A_{rs}^{'} = A_{rs}^{-} + \frac{1}{2} F,$$

mit

(2.1.49)
$$\| \mathbf{F} \| \le \widetilde{K}_{V} S(A)$$
, $v \in \{2,3\}$, $\widetilde{K}_{2} := \frac{128\sqrt{2}}{\sqrt{3}\delta^{2}} \|A\|^{2}$, $\widetilde{K}_{3} := \frac{64\sqrt{6}}{\delta^{2}} \|A\|^{2}$.

Weiter gilt:

(2.1.50)
$$\|A_{rs}^{'-}\| \leq \frac{1}{2}(\sqrt{2} + \widetilde{K}_{v}) \quad S(A).$$

Beweis: Es gilt für $v \in \{2,3\}$ (s.(2.1.3)):

$$A'_{sr} = A_{sr}$$
,
 $A'_{rs} = A_{rs} + A_{rr}X_{v} - X_{v}A_{ss} - X_{v}A_{sr}X_{v} = A_{rs} + F$,

mit

$$\| F \| \leq \sqrt{3} \| A \| \| X_{yy} \| ,$$

wegen $\|X_{v}\| \le 1$ und der Ungleichung von Cauchy-Schwarz.

Daraus folgt für v = 2:

$$\| F \| \le \sqrt{3} \frac{128\sqrt{2}}{3\delta^2} \| A \|^2 \quad S(A) ,$$

wegen $\|X_2\| \le \frac{64}{3\delta^3} \|C_{rs}\|$ und $\|C_{rs}\| \le \frac{1}{\sqrt{2}} S(C(A)) \le 2\sqrt{2} \|A\|S(A)$ nach (2.1.31). Ebenso folgt (2.1.49) für v = 3 aus $\|X_3\| \le \frac{32}{\delta^2} \|C_{rs}\|$. Damit ist (2.1.47) gezeigt. Die Gleichung (2.1.48) folgt aus

$$A_{rs}^{-} = \frac{A_{rs}^{-} - A_{sr}^{-}}{2} = A_{rs}^{-} + \frac{F}{2}$$
.

Schließlich folgt daraus (2.1.50) wegen $\|A_{rs}^-\| \le \frac{1}{2} \sqrt{2} S(A)$. Damit ist Lemma 2.1.12 bewiesen.

Lemma 2.1.13

Gilt zusätzlich zu den Voraussetzungen von Satz 2.1.11:

(2.1.51)
$$S(A) \le \frac{1}{4} \|A\| \varepsilon_{v-1}^{-}, v \in \{2,3\}$$

mit

$$\varepsilon_{\nu-1}^- := \frac{\delta^-}{4 \|A\| (a_{\nu} + 2\tilde{K}_{\nu} + \tilde{K}_{\nu}^2)^{1/2}}$$
, $a_2 := \frac{13}{9}$, $a_3 := \frac{7}{4}$,

so gilt mit A' wie in Lemma 2.1.12:

$$(2.1.52)$$
 $S(A') \le \frac{\delta^{-}}{16}$.

Beweis: Es ist (s.(2.1.3)):

$$S^{2}(A') - S^{2}(A) \leq \sum_{i \neq r, s}^{p} \left(\|A_{ir}\|^{2} \|X_{v}\|^{2} + 2\|A_{ir}\| \|A_{is}\| \|X_{v}\| + \|A_{si}\|^{2} \|X_{v}\|^{2} + 2\|A_{ri}\| \|A_{si}\| \|X_{v}\| \right) + \|A_{ri}\|^{2} - \|A_{rs}\|^{2},$$

$$S^{2}(A') \leq S^{2}(A) (1 + \|X_{v}\| + \|X_{v}\|^{2} + 2\widetilde{K}_{v} + \widetilde{K}_{v}^{2}),$$

$$S(A') \leq S(A) (a_{v} + 2\widetilde{K}_{v} + \widetilde{K}_{v}^{2})^{1/2} \leq \frac{\delta^{-}}{16},$$

wegen $\|X_2\| \le \frac{1}{3}$, $\|X_3\| \le \frac{1}{2}$ und Voraussetzung (2.1.51). Damit ist Lemma 2.1.12 bewiesen.

Hiermit schließen wir die Untersuchung der elementaren Johnsen-Veselić-Voevodin-Matrix ab, und kommen zur Paardekooper-Jacobi-Matrix. Da sowohl bei dem Beweis der asymptotisch quadratischen Konvergenz (s. § 3.2), wie auch in den Verfahren zur praktischen Berechnung (s.Kapitel 4), einer Transformation der Matrix A mit einer Paardekooper-Jacobi-Matrix stets eine T-Transformation zur Normreduzierung vorausgeht, wird, aus Gründen der Konsistenz, die Eingangsmatrix für eine Paardekooper- ebenso wie für eine Householder-Transformation (s.§ 2.2 und § 2.3) mit A', die Ausgangsmatrix mit A'' bezeichnet.

2.2 DIE ELEMENTARE PAARDEKOOPER-JACOBI-MATRIX

Die elementare Paardekooper-Jacobi-Matrix ([26],[10]) dient zum Annullieren eines außerdiagonalen 2×2-Blocks einer schiefsymmetrischen Matrix. Wir verwenden sie, um den Pivotblock A's des schiefsymmetrischen Teils A'von A'zu annullieren (vergl.Kap.3 und 4). Im folgenden beschreiben wir die Annullierung eines 2×2-Blocks A's , $1 \le r < s \le p$. Sei

$$(2.2.1) A''_{rr} = \begin{pmatrix} 0 & \alpha \\ -\alpha & 0 \end{pmatrix} A''_{rs} = \begin{pmatrix} x & v \\ w & y \end{pmatrix} A''_{ss} = \begin{pmatrix} 0 & \beta \\ -\beta & 0 \end{pmatrix} ,$$

und bezeichne $O(1,m;\varphi)$ die Matrix zur Jacobi-Rotation in der (1,m)-Ebene mit den nichttrivialen Elementen $o_{lm} := \sin \varphi =: -o_{ml}$, $o_{ll} := o_{mm} := \cos \varphi$ (s.[16],[29],[14]).

Der Pivotblock A's wird annulliert durch vier nacheinander ausgeführte Jacobi-Rotationen auf Å. Das Produkt aus diesen vier Jacobi-Rotationsmatrizen, von Paardekooper [26] "Jacobi annihilator" genannt, nennen wir elementare Paardekooper-Jacobi-Matrix zum Annullieren des Blocks A's , und bezeichnen sie mit $U(\kappa; \phi_1, \phi_2, \phi_3, \phi_4)$, wobei $\kappa = (r,s)$ der Pivotindex und ϕ_1 bis ϕ_4 die Rotationswinkel sind:

Die Rotationswinkel $\phi_{\textbf{i}}$, i = 1(1)4 werden durch die folgenden Gleichungen bestimmt:

$$(2.2.3) \qquad \tan 2 \, \phi_1 = 2 \, \frac{\alpha w - \beta v}{\alpha^2 - \beta^2 + v^2 - w^2} \, , \quad -\frac{\pi}{4} < \phi_1 \le \frac{\pi}{4} \, ,$$

$$= \left\{ \begin{array}{l} -\frac{v \cos \phi_1 - \beta \sin \phi_1}{\alpha \cos \phi_1 + w \sin \phi_1} \, , \quad \alpha \cos \phi_1 + w \sin \phi_1 \ne 0 \\ -\frac{w \cos \phi_1 - \alpha \sin \phi_1}{\beta \cos \phi_1 + v \sin \phi_1} \, , \quad \beta \cos \phi_1 + v \sin \phi_1 \ne 0 \end{array} \right. ,$$

dabei setzt man $\phi_2=\frac{\pi}{2}$, falls beide Nenner in (2.2.4) verschwinden und mindestens ein Zähler nicht, sonst $\phi_2=0$. In dem zweiten Fall gilt ungefähr $v=w=0=\alpha=\beta$, und man kann auch $\phi_1=0$ wählen. Die Winkel ϕ_1 und ϕ_2 sind so gewählt, daß nach den beiden Rotationen die Außerdiagonalelemente von A_{rs}^{-1} verschwinden. Wir haben also mit

$$A_1 = A'$$
, $A_{k+1} = 0_k^{-1} A_k 0_k$, $k = 1(1)4$, $A'' = A_5$, $A_k = (A_{ij}^{(k)})$

nach der zweiten Rotation:

$$(2.2.5) A_{rr}^{(3)^{-}} = \begin{pmatrix} 0 & \alpha_{3} \\ -\alpha_{3} & 0 \end{pmatrix}, A_{rs}^{(3)^{-}} = \begin{pmatrix} x & 0 \\ 0 & y \end{pmatrix}, A_{ss}^{(3)^{-}} = \begin{pmatrix} 0 & \beta_{3} \\ -\beta_{3} & 0 \end{pmatrix},$$

$$\alpha_{3} = \cos \varphi_{2} (\alpha \cos \varphi_{1} + w \sin \varphi_{1}) - \sin \varphi_{2} (v \cos \varphi_{1} - \beta \sin \varphi_{1}),$$

$$\beta_{3} = \cos \varphi_{2} (\beta \cos \varphi_{1} + v \sin \varphi_{1}) - \sin \varphi_{2} (w \sin \varphi_{1} - \alpha \sin \varphi_{1}).$$

Die Winkel ϕ_3 und ϕ_4 werden so bestimmt, daß $A_{rs}^{"} = 0$ gilt:

$$(2.2.6) \qquad \tan 2\,\varphi_3 = 2\,\frac{\alpha_3 y + \beta_3 x}{\alpha_3^2 - \beta_3^2 + x^2 - y^2} \,, \quad -\frac{\pi}{4} < \varphi_3 \le \frac{\pi}{4} \,,$$

$$(2.2.7) \qquad \tan \varphi_4 = \begin{cases} -\frac{x \cos \varphi_3 + \beta_3 \sin \varphi_3}{\alpha_3 \cos \varphi_3 + y \sin \varphi_3} \,, & \alpha_3 \cos \varphi_3 + y \sin \varphi_3 \ne 0 \\ -\frac{\pi}{2} < \varphi_4 \le \frac{\pi}{2} \,. \end{cases}$$

$$\frac{y \cos \varphi_3 - \alpha_3 \sin \varphi_3}{\beta_3 \cos \varphi_3 - x \sin \varphi_3} \,, \quad \beta_3 \cos \varphi_3 - x \sin \varphi_3 \ne 0$$

Für (2.2.7) gilt das zu (2.2.4) gesagte entsprechend. Paardekooper [26] hat gezeigt, daß

(2.2.8)
$$S^{2}(A''^{-}) = S^{2}(A'^{-}) - 2 \|A_{rs}^{'-}\|^{2}$$

gilt.

Eine Transformation

(2.2.9)
$$A'' = U^{T}A'U , U = U(\kappa; \varphi_{1}, \varphi_{2}, \varphi_{3}, \varphi_{4})$$

nennen wir U-Transformation. Wegen der Orthogonalität von \bigcup gilt mit $\kappa = (r,s)$:

Daraus folgt wegen $A''_{ij} = A'_{ij}$, $\{i,j\} \cap \{r,s\} = \emptyset$:

$$(2.2.11) S2(A'') - S2(A') = ||A''_{rs}||2 + ||A''_{sr}||2 - ||A'_{rs}||2 - ||A'_{sr}||2.$$

Weiter gilt

$$A''_{ri} = U^{T}_{rr}A'_{ri} + U^{T}_{sr}A'_{si}$$

$$A''_{si} = U^{T}_{rs}A'_{ri} + U^{T}_{ss}A'_{si}$$

$$A''_{si} = A'_{rs}U_{rr} + A'_{ss}U_{sr}$$

$$A''_{ir} = A'_{ir}U_{rs} + A'_{is}U_{ss}$$

$$A''_{is} = A'_{ir}U_{rs} + A'_{is}U_{ss}$$
, $i \neq r, s$

und

$$A''_{rr} = U_{rr}^{T} A'_{rr} U_{rr} + U_{sr}^{T} A'_{sr} U_{rr} + U_{rr}^{T} A'_{rs} U_{sr} + U_{sr}^{T} A'_{ss} U_{sr},$$

$$A''_{rs} = U_{rr}^{T} A'_{rr} U_{rs} + U_{sr}^{T} A'_{sr} U_{rs} + U_{rr}^{T} A'_{rs} U_{ss} + U_{sr}^{T} A'_{ss} U_{ss},$$

$$A''_{sr} = U_{rs}^{T} A'_{rr} U_{rr} + U_{ss}^{T} A'_{sr} U_{rr} + U_{rs}^{T} A'_{rs} U_{sr} + U_{ss}^{T} A'_{ss} U_{sr},$$

$$A''_{ss} = U_{rs}^{T} A'_{rr} U_{rs} + U_{ss}^{T} A'_{sr} U_{rs} + U_{rs}^{T} A'_{rs} U_{ss} + U_{ss}^{T} A'_{ss} U_{ss},$$

wobei analog zu (2.0.3) mit $s_i := \sin \phi_i$, $c_j := \cos \phi_j$, j = 1(1)4:

$$(2.2.14) \begin{pmatrix} U_{rr} & U_{rs} \\ U_{sr} & U_{ss} \end{pmatrix} := \widehat{U}_{rs} = \begin{pmatrix} c_{1}c_{3} & -s_{1}s_{4} & c_{1}s_{3} \\ -s_{2}s_{3} & c_{2}c_{4} & c_{2}s_{4} & s_{2}c_{3} \\ -s_{1}c_{3} & -c_{1}s_{4} & c_{1}c_{4} & -s_{1}s_{3} \\ -c_{2}s_{3} & -c_{4}s_{2} & -s_{2}s_{4} & c_{2}c_{3} \end{pmatrix}$$

gesetzt wird. Für die Pivotblöcke des Kommutators von $A: C_{rr}$, $C_{rs} = C_{sr}^{T}$ und C_{ss} gelten ebenso wie für die Blöcke des Kommutators von $\widehat{A}_{r\bar{s}}: \widetilde{C}_{rr}$, $\widetilde{C}_{rs} = \widetilde{C}_{sr}^{T}$ und \widetilde{C}_{ss} die Gleichungen (2.2.13) entsprechend.

Die Rotationswinkel (2.2.3 ... 4), (2.2.6 ... 7) können nach Hari [10] folgendermaßen abgeschätzt werden:

Sei $A^{'}$ eine schiefsymmetrische Matrix, die die Bedingungen

$$(2.2.15) 3\tilde{\delta} := \min\{|v_1 - v_m| | 1 \le 1 < m \le p\} > 0 ,$$

wobei $\pm iv_1$, $v_1 > 0$, l = 1(1)p die Eigenwerte *) von A'- sind und

$$(2.2.16) S(A') \leq \frac{\sqrt{2}}{2} \delta$$

erfüllt. Sei \bigcup ((r,s); ϕ_1 , ϕ_2 , ϕ_3 , ϕ_4) die elementare Paardekooper-Jacobi-Matrix (2.2.2) zum Annullieren von $A_{rs}^{\prime-}$. Dann genügen die Winkel ϕ_1, ϕ_2, ϕ_3 und ϕ_4 den Ungleichungen

(2.2.17)
$$\sum_{j=1}^{4} \sin^{2} \varphi_{j} \leq a^{2} \frac{\|A_{rs}^{j-1}\|^{2}}{\tilde{\delta}^{2}} ,$$

und

$$\begin{cases} \frac{c}{\delta} \sqrt{v^2 + w^2}, & j = 1 \\ \frac{b}{\delta} \sqrt{v^2 + w^2}, & j = 2 \\ \frac{c}{\delta} \sqrt{x^2 + y^2}, & j = 3 \\ \frac{b}{\delta} \sqrt{x^2 + y^2}, & j = 4 \end{cases}$$

Vergl.Fußnote S.3

mit

(2.2.19)
$$a = 1.1336$$
, $b = 1.004$, $c = \frac{10}{19}$.

Um diesen Satz anwenden zu können, müssen wir aus δ (s.(1.1.27)) eine Abschätzung für $\widetilde{\delta}$ gewinnen. Das geschieht in dem folgenden

Lemma 2.2.2. Sei A' eine reelle Blockmatrix mit $\delta > 0$ und

$$S(A') \le \frac{\delta^{-}}{16}$$
, (2.2.20) $\Delta(A') \le \frac{\delta^{-}}{16}$.

Dann gilt

$$(2.2.21) \qquad \tilde{\delta} \geq \frac{5}{16} \delta^{-} \qquad ,$$

und der Abstand der Spektren $\pm i\mu_j$, $\mu_j > 0$ der Diagonalblöcke Ajd des schiefsymmetrischen Teils Ad von A ist nach unten beschränkt durch $\frac{7}{8} \delta^{-}$. Weiter gilt $\mu_{j} \ge \frac{2}{5} \delta^{-}$, j = 1(1)p.

Beweis: Die Matrix A' läßt sich schreiben als Summe ihres symmetrischen und schiefsymmetrischen Teils:

$$A' = A'' + A'' - .$$

Sei N eine normale Matrix mit denselben Eigenwerten λ_i , i = 1(1)n, wie $\[A' \]$, gemäß Satz 1.1.18. Diese läßt sich analog zerlegen in

$$N = N^+ + N^- ,$$

wobei nach Satz 1.1.13 N^+ die Eigenwerte Re λ_{i} und N^- die Eigenwerte $\operatorname{Im} \lambda_{\mathbf{i}}$, i=1(1)n hat. Für j=1(1)p habe der Diagonalblock $A_{\mathbf{j}\mathbf{j}}^{\mathbf{i}}$ die Eigenwerte $\pm i\mu_{\mathbf{j}}$, $\mu_{\mathbf{j}}>0$. Die Matrizen $A_{\mathbf{j}\mathbf{j}}^{\mathbf{i}}$, $N_{\mathbf{j}}^{\mathbf{i}}$ und diag $(A_{\mathbf{j}\mathbf{j}}^{\mathbf{i}})$ sind nach Konstruktion schiefsymmetrisch, so daß der Satz von Wielandt-Hoffmann für schiefsymmetrische Matrizen [10], [15] auf die Paare (A'^- , N^-) und (A'^- , diag (A'_{jj})) anwendbar ist, und es folgt, da A' wegen $\delta^- > 0$ nur konjugiert komplexe Eigenwerte hat, mit Satz 1.1.18 und $\|A^{-}\| \leq \|A\|$:

$$2 \sum_{j=1}^{p} |v_{j} - Im \lambda_{j}|^{2} \le ||A' - N^{-}||^{2} \le ||A' - N||^{2} \le \Delta^{2}(A'),$$

und

$$2 \sum_{j=1}^{p} |v_{j} - \mu_{j}|^{2} \le ||A'^{-} - \operatorname{diag}(A'_{jj})||^{2}$$
$$= S^{2}(A'^{-}) \le S^{2}(A'),$$

für eine geeignete Ordnung der ν_j und λ_j , da die Ordnung der Diagonalblöcke A; durch die Matrix Λ^{-j} vorgegeben ist.

Aus der ersten Ungleichung folgt für i + j:

$$|v_{i} - \operatorname{Im} \lambda_{i}|^{2} + |v_{j} - \operatorname{Im} \lambda_{j}|^{2} \leq \frac{\Delta^{2}(A')}{2} \leq \frac{(\delta^{-})^{2}}{2 \cdot 16^{2}}$$

und damit wegen der Ungleichung von Cauchy-Schwarz:

$$|v_{\mathbf{i}} - v_{\mathbf{j}}| \ge |\operatorname{Im} \lambda_{\mathbf{i}} - \operatorname{Im} \lambda_{\mathbf{j}}| - (|v_{\mathbf{i}} - \operatorname{Im} \lambda_{\mathbf{i}}| + |v_{\mathbf{j}} - \operatorname{Im} \lambda_{\mathbf{j}}|)$$

$$\ge \delta^{-} - \sqrt{2} \frac{\delta^{-}}{\sqrt{2} \cdot 16} = \frac{15}{16} \delta^{-},$$

womit der erste Teil der Behauptung bewiesen ist.

Faßt man beide Ungleichungen zusammen, so erhält man

(2.2.22)
$$\sum_{j=1}^{p} |\mu_{j} - \operatorname{Im} \lambda_{j}|^{2} \leq S^{2}(A') + \Delta^{2}(A') \leq \frac{(\delta^{-})^{2}}{128} ,$$

also für i + j:

$$|\mu_{i} - Im \lambda_{i}| + |\mu_{j} - Im \lambda_{j}| \le \frac{\sqrt{2} \delta^{-}}{\sqrt{128!}} = \frac{\delta^{-}}{8}$$
.

Damit ergibt sich für den Abstand der Spektren der Diagonalblöcke des schiefsymmetrischen Teils von A':

$$|\mu_{i} - \mu_{j}| \ge |\operatorname{Im} \lambda_{i} - \operatorname{Im} \lambda_{j}| - (|\mu_{i} - \operatorname{Im} \lambda_{i}| + |\mu_{j} - \operatorname{Im} \lambda_{j}|)$$

$$\ge \delta^{-} - \frac{\delta^{-}}{8} = \frac{7}{8} \delta^{-}.$$

Aus (2.2.22) folgt $\mu_j \ge \text{Im } \lambda_i - \delta^-/\sqrt{128} \ge \frac{2}{5} \delta^-$, j = 1(1)p. Damit ist Lemma 2.2.2 bewiesen.

Lemma 2.2.3

Sei A eine reelle Blockmatrix mit $\delta > 0$, (r,s) fest und

$$\varepsilon \leq \min\{\varepsilon_0, \varepsilon_2, \varepsilon_3, \varepsilon_{\nu-1}^{-1}\}, \quad \nu \in \{2,3\},$$

$$S(A) \leq \frac{1}{4} \|A\| \varepsilon,$$

$$\|C(D)\| \leq \frac{\delta^2}{64},$$

$$\Delta(A) \leq \frac{\delta^{-1}}{16},$$

$$A' := T((r,s); X_{\nu})^{-1} A T((r,s); X_{\nu})^{*}$$

wobei T die elementare T-Matrix (2.1.1) mit X_{ν} aus (2.1.17) bzw. (2.1.44) ist.

Dann gilt

mit

$$b_{rs}^{(v)} = \frac{a}{2} (\sqrt{2} + \widetilde{K}_{v})$$

und

$$(2.2.25)$$
 $\|U_{rs}\| \leq \frac{a}{7}$, $\|U_{sr}\| \leq \frac{a}{7}$.

^{*)} Dieses ist eine Generalvoraussetzung für diesen Paragraphen, auf die im folgenden verwiesen wird. Die Wahl $\nu \in \{2,3\}$ ist so zu verstehen, daß stets $\nu = 2$ oder $\nu = 3$ zu setzen ist, je nachdem die normreduzierende T-Transformation mit dem Parameter χ_2 oder χ_3 durchgeführt wird.

Beweis: Mit (2.2.23) sind die Voraussetzungen der Lemmata 2.1.12 und 2.1.13 erfüllt, so daß auch Lemma 2.2.2 angewendet werden kann, da wegen der Abnahme der Euklidischen Norm unter einer T-Transformation: $\Delta(A') \leq \Delta(A)$ gilt. Wegen

$$(2.2.26) S(A') \leq \frac{\delta^{-}}{16} \leq \frac{\delta}{5} < \frac{\sqrt{2}}{2} \delta,$$

nach Lemma 2.2.2, sind die Voraussetzungen von Satz 2.2.1 erfüllt, und es folgt:

$$||U_{rs}||^{2} \le s_{1}^{2} + s_{2}^{2} + s_{3}^{2} + s_{4}^{2} \le a^{2} \frac{||A_{rs}^{\prime}||^{2}}{\delta^{2}} ,$$

$$||U_{rs}|| \le \frac{a}{2\delta} (\sqrt{2} + \widetilde{K}_{V}) S(A) ,$$

wegen Lemma 2.1.12. Damit ist (2.2.24) gezeigt, denn für $\|U_{sr}\|$ gilt das oben gesagte analog. Weiter erhält man aus (2.2.27) mit (2.2.26):

$$\|U_{rs}\|^2 \le \frac{a^2}{2\tilde{\kappa}} S^2(A') \le \frac{a^2}{50}$$
,

und daraus folgt (2.2.25), womit Lemma 2.2.3 bewiesen ist.

Die Außerdiagonalblöcke von $\widehat{\mathbb{U}}_{rs}$ sind damit abgeschätzt. Es bleiben die Diagonalblöcke abzuschätzen. Dazu zerlegen wir \mathbb{U}_{rr} und \mathbb{U}_{ss} auf zwei Arten:

$$(2.2.28)$$
 $U_{rr} = U_{rr}^{d} + U_{rr}^{a}$,

mit

$$U_{rr}^{d} = \begin{pmatrix} c_{1}c_{3} & 0 \\ 0 & c_{2}c_{4} \end{pmatrix} , \quad U_{rr}^{a} = \begin{pmatrix} 0 & -s_{1}s_{4} \\ -s_{2}s_{3} & 0 \end{pmatrix}$$

und
$$U_{ss}$$
 analog, $U_{rr} = I + U'_{rr}$, $U'_{rr} = \begin{pmatrix} c_1 c_3 - 1 & -s_1 s_4 \\ -s_2 s_3 & c_2 c_4 - 1 \end{pmatrix}$, U_{ss} analog.

Lemma 2.2.4

Sei A eine reelle Blockmatrix mit $\delta^- > 0$, die die Voraussetzungen (2.2.23) erfüllt. Dann gilt

$$(2.2.30)$$
 $\|U_{rr}^{d}\|_{Sp} \le 1$, $\|U_{rr}^{a}\|_{Sp} \le 0.0106$

$$(2.2.31)$$
 $\|U_{ss}^{d}\|_{sp} \le 1$, $\|U_{ss}^{a}\|_{ss} \le 0.0148$,

$$(2.2.32)$$
 $\|U_{rr}\| \le 1.4249$, $\|U_{ss}\| \le 1.4291$,

$$(2.2.33) \|U_{rr}^{a}\| \leq 0.1322 (\sqrt{2} + \widetilde{K}_{v})^{2} \frac{S^{2}(A)}{\delta^{2}},$$

$$(2.2.34) \| U_{ss}^{a} \| \leq 0.1848 \left(\sqrt{2} + \widetilde{K}_{v} \right)^{2} \frac{S^{2}(A)}{\widetilde{\delta}^{2}}.$$

Die Behauptungen folgen durch einfache Rechnung aus Satz 2.2.1, unter Beachtung von $S(A') \leq \frac{\delta}{5} (s.(2.2.26))$ und $\|A_{rs}^{-}\| \leq \frac{1}{2} (\sqrt{2} + \widetilde{K}_{v}) S(A)$ (s.Lemma 2.1.12).

Nachdem die Darstellung (2.2.28) abgeschätzt ist, kommen wir nun zu einer Abschätzung der Darstellung (2.2.29).

Lemma 2.2.5

Sei A eine reelle Blockmatrix mit $\delta > 0$, die die Voraussetzungen (2.2.23) erfüllt. Dann gilt:

$$(2.2.35)$$
 $\|U'_{rr}\| \le 0.0259$, $\|U'_{ss}\| \le 0.0279$ und

$$(2.2.36) \|U_{rr}^{\prime}\| \leq b_{r}^{(v)} \frac{S^{2}(A)}{\delta^{2}} , \|U_{ss}^{\prime}\| \leq b_{s}^{(v)} \frac{S^{2}(A)}{\delta^{2}} ,$$

mit

$$b_r^{(v)} := 0.1871 \left(\sqrt{2} + \widetilde{K}_v\right)^2 , \quad b_s^{(v)} := 0.2947 \left(\sqrt{2} + \widetilde{K}_v\right)^2 .$$

Beweis: Mit $S(A') \le \frac{\widetilde{\delta}}{5}$ erhält man aus Satz 2.2.1 für $j \in \{1,3\}$:

(2.2.37)
$$|\sin \varphi_{j}| \leq \frac{c}{\delta} \|A_{rs}^{-1}\| \leq \frac{c}{5\sqrt{2}} \leq 0.0745$$
,

und analog für $j \in \{2,4\}$:

$$(2.2.38) |\sin \varphi_i| \le \frac{b}{5\sqrt{2}} \le 0.142 .$$

Daraus folgen mit $\cos \varphi_j = \sqrt{1-\sin^2 \varphi_j}$ die Abschätzungen (2.2.35). Durch Taylorentwicklung von $\sqrt{1-x}$ um O erhält man wegen $|s_j| = |\sin \varphi_j| \le 1$, j = 1(1)4:

$$\cos \varphi_{j} = c_{j} = \sqrt{1 - s_{j}^{2}} \ge 1 - \frac{s_{j}^{2}}{2\sqrt{1 - s_{j}^{2}}}$$

und daraus für $j \in \{1,3\}$ mit (2.2.37):

$$\cos \varphi_{i} \ge 1 - K_{1} s_{i}^{2}$$
 , $K_{1} := 0.5015$.

Entsprechend erhält man für $j \in \{2,4\}$ mit (2.2.38):

$$\cos \varphi_{j} \ge 1 - K_{2} s_{j}^{2}$$
 , $K_{2} := 0.5052$.

Für die Diagonalelemente von U'rr folgt damit:

$$1 \ge c_1 c_3 \ge (1 - K_1 s_1^2) (1 - K_1 s_3^2)$$

$$\ge 1 - K_1 c^2 \frac{\|A_{rs}^{'}\|^2}{\delta^2}$$

$$\ge 1 - 0.0348 (\sqrt{2} + K_{v})^2 \frac{S^2(A)}{\delta^2}$$

dabei wurde zuerst Satz 2.2.1 und dann Lemma 2.1.12 angewendet. Entsprechend ergibt sich:

$$1 \ge c_2 c_4 \ge 1 - 0.1274 \left(\sqrt{2} + \widetilde{K}_{V}\right)^2 \frac{S^2(A)}{\widetilde{S}^2}$$

Damit ist zusammen mit Lemma 2.2.4:

$$\|U_{rr}^{\prime}\|^{2} \leq \|U_{rr}^{a}\|^{2} + (0.0348^{2} + 0.1274^{2}) (\sqrt{2} + \widetilde{K}_{v})^{4} \frac{S^{4}(A)}{\delta^{4}},$$

$$\|U_{rr}^{\prime}\| \leq 0.1871 (\sqrt{2} + \widetilde{K}_{v})^{2} \frac{S^{2}(A)}{\delta^{2}}.$$

Der zweite Teil von (2.2.36) folgt analog, womit das Lemma 2.2.6 bewiesen ist.

Hiermit sind alle Hilfsmittel, die in § 3.2 zum Beweis der quadratischen Konvergenz benötigt werden, bereitgestellt.

2.3 DIE ELEMENTARE VESELIĆ-HOUSEHOLDER-MATRIX

Die elementare Veselić-Householder-Matrix Q zum Annullieren eines Pivotblocks $A_{rs}^{'\pm}$ des symmetrischen Teils $A^{'+}$ bzw. des schiefsymmetrischen Teils $A^{'-}$ von $A^{'}$ ist definiert durch (vergl.(2.0.2), (2.0.3)):

(2.3.1)
$$\widehat{Q}_{rs} = Q_4 Q_3$$

$$Q_4 = (I - 2 w w^T) , w^T w = 1 , w \in \mathbb{R}^4 ,$$

$$Q_3 = (I - 2 v v^T) \oplus (1) , v^T v = 1 , v \in \mathbb{R}^3 ,$$

wobei Q_3 und Q_4 Householder-Matrizen sind. Sind die Eigenwerte von \widehat{A}_{rs}^+ bzw. \widehat{A}_{rs}^- bekannt (sie können nach den Verfahren aus § 1.2 berechnet werden), so kann man die orthogonale Matrix \widehat{Q}_{rs} durch QL-Zerlegung bestimmen (vergl. [37],[1],[41]).

Betrachten wir zunächst den symmetrischen Fall; hier gibt es keine Probleme, da alle Eigenwerte reell sind.

1.) A :=
$$\widehat{A}_{rs}^{'+}$$
 habe getrennte Eigenwerte $\lambda_1 < \lambda_2 < \lambda_3 < \lambda_4$. Wir setzen

$$(2.3.2) \qquad \qquad \text{B} := (\text{A} - \lambda_{\mathbf{k}} \text{I}) (\text{A} - \lambda_{\mathbf{l}} \text{I}) \quad \text{k,l nach Abschnitt 1.2.5} \,,$$

$$\text{B':= B H = (b'_{ij})} \quad \text{, H Permutations matrix,}$$

so daß die letzten beiden Spalten von B' die größten in der euklidischen Vektornorm sind. Berechnet man von B' die QL-Zerlegung B' = QL , so gilt $Q^TBQ = \begin{pmatrix} 0 & 0 \\ 0 & B_2 \end{pmatrix}$ und mit $\widehat{Q}_{rs} = Q$:

$$(2.3.3) \qquad \widehat{A}_{rs}^{"+} := \widehat{Q}_{rs}^{T} \widehat{A}_{rs}^{'+} \widehat{Q}_{rs} = \begin{pmatrix} A_{rr}^{"+} & 0 \\ 0 & A_{ss}^{"+} \end{pmatrix} .$$

2.) A besitze einen doppelten Eigenwert
$$\tau = \lambda_k = \lambda_1$$
 , $\lambda_i \neq \tau$, $j \notin \{k,1\}$. Dann setzen wir

$$(2.3.4)$$
 B = A - τI ,

und verfahren weiter wie unter 1.) .

3.) A besitze einen dreifachen Eigenwert $\tau = \lambda_k = \lambda_1 = \lambda_j$, $\lambda_i \neq \tau$, $i \notin \{k,1,j\}$. Dann setzen wir auch (2.3.4); die Matrix B hat Rang 1, zur QL-Zerlegung reicht eine Householder-Matrix Q_4 , und es gilt

4.) A besitze einen vierfachen Eigenwert. Dann hat Λ_{rs}^+ die Gestalt (2.3.5) mit $\lambda_i = \tau$ und es ist nichts zu tun.

Damit ist der symmetrische Fall abgehandelt, und wir kommen zum schiefsymmetrischen Fall. Hier gibt es nur zwei Möglichkeiten:

1.) A = $\widehat{A}_{rs}^{'}$ habe zwei verschiedene Eigenwertpaare $\lambda_1 = \overline{\lambda}_2 * \lambda_3 = \overline{\lambda}_4$. Wir setzen

$$(2.3.6) B = A^2 + \lambda_1 \bar{\lambda}_1 I = A^2 + |\lambda_1|^2 I = A^2 - z I.$$

B ist reell und symmetrisch, so daß weiter wie im symmetrischen Fall unter 1.) verfahren werden kann.

2.) A habe ein doppeltes konjugiert komplexes Eigenwertpaar $\lambda_1 = \overline{\lambda}_2 = \lambda_3 = \overline{\lambda}_4$. Bildet man hier B wie in (2.3.6), so hat B den Rang O, und B ist die Nullmatrix. Mit dem obigen Verfahren kann also keine Transformationsmatrix bestimmt werden; hier ist ein Ersatzverfahren erforderlich. Dazu führt man nach [37],[1] auf der komplexen Matrix $A - \lambda_1 I$ das Gauß'sche Eliminationsverfahren mit vollständiger Pivotisierung durch, und erhält die reelle Matrix

mit

$$a'_{14} = (a_{14} + a_{23})/2$$
,
 $a'_{24} = (a_{24} + a_{13})/2$,
 $a'_{34} = (a_{34} + a_{12})/2$ und
 $A = (a_{ij})_{i,j=1} (1) 4$.

Das obere (untere) Vorzeichen ist zu setzen, je nachdem $a_{14}a_{23} < 0$ (>0) ist (vergl. Abschnitt 1.2.4). Mit dieser Matrix B verfährt man weiter wie im symmetrischen Fall unter 1.), und erhält durch QL-Zerlegung eine Matrix Q, die ebenfalls die Form (2.3.3) erzeugt; man benötigt also auch hier nur reelle Arithmetik.

Damit ist die elementare Veselić-Householder-Matrix (2.3.1) bestimmt. In [1] wurde damit ein Jacobi-ähnliches Verfahren für schiefsymmetrische und für symmetrische Matrizen konstruiert und die globale und asymptotisch quadratische Konvergenz bei optimaler Pivotstrategie bewiesen. Hier wird auf eine weitergehende theoretische Untersuchung dieser Transformationsmatrix, wie sie in § 2.2 für die Paardekooper-Jacobi-Matrix durchgeführt ist, verzichtet, da es den Rahmen der Arbeit sprengen würde. Es lassen sich aber vermutlich zu Kapitel 3 analoge Resultate mit derselben Technik erzielen (vergl. § 4.2).

In Kapitel 4 wird mit diesen Matrizen ein Jacobi-ähnliches Verfahren für allgemeine reelle Matrizen konstruiert und numerisch getestet.

2.4 DIE ELEMENTARE EBERLEIN-JACOBI-MATRIX

Die elementare Eberlein-Jacobi-Matrix dient hier zur Normreduzierung auf den Diagonalblöcken. Sie ist im Gegensatz zu den anderen elementaren Matrizen, die hier behandelt werden, nur in einem 2×2 Diagonalblock von der Einheitsmatrix verschieden, und besteht aus dem Produkt $0\cdot S$ einer(orthogonalen) Jacobi-Rotation 0 und einer nichtorthogonalen hyperbolischen Transformation S (s.[4],[7], [8],[38]). Die nichttrivialen Blöcke von $0((1,m);\phi_r)$ bzw. $S((1,m);\psi_r)$, 1=2r-1, m=2r sind gegeben durch

$$(2.4.1) O_{rr} = \begin{pmatrix} \cos \varphi_{r} & -\sin \varphi_{r} \\ \sin \varphi_{r} & \cos \varphi_{r} \end{pmatrix},$$

mit

(2.4.2)
$$\tan 2 \varphi_{r} = -\frac{c_{11} - c_{mm}}{2 c_{1m}} *), -\frac{\pi}{4} < \varphi_{r} \le \frac{\pi}{4}$$

und

(2.4.3)
$$S_{rr} = \begin{pmatrix} ch \psi_r & sh \psi_r \\ sh \psi_r & ch \psi_r \end{pmatrix},$$

mit

(2.4.4) th
$$\psi_r = \frac{-c_{1m}^r}{G + 2(D^2 + E^2)}$$
,

$$G = \sum_{i=1,m}^{n} (a_{i1}^{2} + a_{im}^{2} + a_{ii}^{2} + a_{mi}^{2}) ,$$

$$D = a'_{11} - a'_{mm}$$
, $E = a'_{1m} - a'_{m1}$.

Dabei ist

(2.4.5)
$$A' = O^T A O$$
, $C(A') = (c'_{ij})$,

$$(2.4.6)$$
 A" = S⁻¹A'S

gesetzt.

^{*)} c_{11}, c_{1m}, c_{mm} aus (1.1.5).

Eberlein [4] hat für die Transformation (2.4.5) mit (2.4.1) und (2.4.2) gezeigt:

$$(2.4.7)$$
 $c_{11}^{\prime} = c_{mm}^{\prime}$,

(2.4.8)
$$c'_{1m} = sgn(c_{1m}) \sqrt{c_{1m}^2 + \frac{1}{4}(c_{11} - c_{mm})^2}$$
.

Für die Transformation (2.4.6) mit (2.4.3) und(2.4.4) hat Eberlein [4] gezeigt:

$$(2.4.9)$$
 | th ψ | $\leq \frac{1}{2}$,

(2.4.10)
$$\|A'\|^2 - \|A''\|^2 \ge \frac{1}{3} \frac{c_{1m}^{'2}}{\|A\|^2} .$$

Veselic [33] hat für die letztere Transformation gezeigt, daß für alle $1 \le i, j \le n$:

(2.4.11)
$$|a'_{ij} - a_{ij}| \le \sqrt{\frac{3}{2} \max\{|c_{k1}| | 1 \le k < 1 \le n\}}$$

gilt, und daraus folgt

$$|a_{ij}^{\prime} - a_{ij}^{\prime}| \leq \sqrt{\frac{3}{2} \max\{|c_{k1}^{\prime}| | 1 \leq k < 1 \leq n\}}.$$

Diese Ergebnisse werden in Kapitel 3 benötigt. Damit ist die Diskussion der elementaren Matrizen abgeschlossen, und wir kommen zu den Konvergenzbeweisen für ein Jacobi-ähnliches Verfahren.

3. KONVERGENZBEWEISE

In diesem Kapitel werden zwei Konvergenzsätze für ein Jacobiähnliches, normreduzierendes Verfahren formuliert, das mit Johnsen-Veselić-Voevodin-Matrizen zur Normreduzierung und. Paardekooper-Jacobi-Matrizen zur Blockdiagonalisierung des schiefsymmetrischen Teils der Matrix arbeitet. Die Diagonalblöcke werden jeweils mit Eberlein-Jacobi-Matrizen zur Normreduzierung transformiert.

Wir haben zwei Klassen von normreduzierenden Parametern: X_{ν} , $\nu \in \{0,1\}$, die stets eine Normreduzierung gewährleisten, und X_{ν} , $\nu \in \{2,3\}$, für die lediglich asymptotisch eine Normreduzierung und Beschränktheit von $\|X_{\nu}\|$ nachweisbar ist, die aber quadratische Konvergenz ermöglichen. Um beide Vorteile miteinander zu verbinden, wählen wir den Parameter X der T-Transformation (vergl.[34] und [38]) – mit $\nu \in \{2,3\}$ je nach gewünschter Variante des Verfahrens – zu: $X = X_{\nu}$, falls X_{ν} berechnet werden kann und $f(X_{\nu}) < f(X_{1})$ ist, anderenfalls $X = X_{1}$.

Die Konvergenzsätze werden der Übersichtlichkeit halber nur für beide Klassen von Parametern getrennt formuliert, und nicht für das oben angeführte kombinierte Verfahren. Der Beweis der globalen Konvergenz ist für eine Variante des oben beschriebenen Verfahrens analog möglich, wenn man den Transformationsparameter geeignet einschränkt, etwa in der Form

$$X := \left\{ \begin{array}{ll} X_{\mathcal{V}} & \text{, falls } f(X_{\mathcal{V}}) < f(X_{1}) & \text{, } \|X_{\mathcal{V}}\| \leq 2\|X_{1}\|, \nu \in \{2,3\} \\ X_{1} & \text{sonst} \end{array} \right.$$

Daß die asymptotisch quadratische Konvergenz auch für dieses kombinierte Verfahren erhalten bleibt, wird am Schluß von Abschnitt 3.2.4 angedeutet.

3.1 ZUR GLOBALEN KONVERGENZ EINES JACOBI-ÄHNLICHEN NORMREDUZIERENDEN VERFAHRENS BEI OPTIMALER PIVOTWAHL

Analog zu dem Hauptsatz von Hari [7]*), wo an Stelle der hier verwendeten T-Transformationen die Normreduzierung auf der Außerblockdiagonalen mit hyperbolischen Transformationen (s. § 2.4) durchgeführt wird, läßt sich der folgende Satz beweisen. Wir wollen hier auf die Ausführung des Beweises aus Platzgründen verzichten, da der Beweis direkt aus [7] übertragen werden kann, wenn ein technisches Lemma gezeigt ist, daß nämlich nach einer T-Transformation mit Pivot (r,s) und X_{N} , $v \in \{0,1\}$ als Parameter:

$$\|A_{ij}^{'-} - A_{ij}^{-}\| \le 2\sqrt{\|C_{rs}\|}, 1 \le i, j \le p$$

gilt; mit den in § 2.1 bereitgestellten Hilfsmitteln ist das leicht möglich.

Satz 3.1.1 (Erster Hauptsatz)

Sei A eine beliebige reelle Blockmatrix der Dimension n = 2p; sei

$$A_o := A$$
 , $A_k := R_k^{-1} A_{k-1} R_k$, $k \ge 1$,

wobei R_k entweder eine Paardekooper-Jacobi-Matrix $U(\kappa_k;\ldots)$, eine Johnsen-Veselić-Voevodin-Matrix $T(\kappa_k;X_{_{\!\mathcal{V}}})$, $\nu\in\{0,1\}$ mit $\kappa_k=(r_k,s_k)$, $1\leq r_k < s_k \leq p$, oder eine Eberlein-Jacobi-Matrix 0_kS_k mit Pivotindices $(2r_k-1,2r_k)$, $1\leq r_k\leq p$ ist, abhängig davon, welche der folgenden Größen

$$\begin{array}{lll} a_{k-1} &:= \max\{\|A_{ij}^{(k-1)}\|^2 \mid 1 \leq i < j \leq p\} &, \\ c_{k-1} &:= \max\{\|C_{ij}^{(k-1)}\| & |1 \leq i < j \leq p\} &, \\ \\ d_{k-1} &:= \max\left\{\sqrt{\left(c_{ii+1}^{(k-1)}\right)^2 + \frac{1}{4}\left(c_{ii}^{(k-1)} - c_{i+1}^{(k-1)}\right)^2} \mid i = 1 \ (2) \ n-1 \right\} \end{array}$$

am größten ist. Das Pivot-Paar κ_k bzw. $(2r_k-1,2r_k)$ wird so gewählt, daß dort das jeweilige Maximum angenommen wird.

^{*)} Der dort geführte Beweis ist eine Übertragung eines Beweises von Veselić [33] zur globalen Konvergenz des Verfahrens von Eberlein [6].

Dann gilt mit $A_k = (A_{ij}^{(k)})$:

(I)
$$C(A_k) \rightarrow 0$$
, $k \rightarrow \infty$.

(II) Die Folge der schiefsymmetrischen Teile A_k^- von A_k konvergiert gegen eine feste Matrix

$$M = \bigoplus_{i=1}^{p} \begin{pmatrix} 0 & a_{2i-1 \ 2i}^{\circ-} \\ -a_{2i-1 \ 2i}^{\circ-} & 0 \end{pmatrix}$$

- (III) $A_{ij}^{(k)} \rightarrow 0, \quad k \rightarrow \infty \quad \text{für jedes Paar (i,j) , i # j ,}$ $\text{für das } a_{2i-1 \ 2i}^{0-} \quad \# a_{2j-1 \ 2j}^{0-} \quad \text{gilt.}$
- (IV) Hat A höchstens einen (doppelten) reellen Eigenwert, und sind die Imaginärteile der (konjugiert) komplexen Eigenwerte von A getrennt, so konvergiert (A_k) gegen die Murnaghan-Form von A.

- ZUR ASYMPTOTISCH QUADRATISCHEN KONVERGENZ EINES JACOBI-ÄHNLICHEN NORMREDUZIERENDEN VERFAHRENS BEI ZEILENZYKLISCHER PIVOTSTRATEGIE
- 3.2.1 Beschreibung des Verfahrens

Wir gehen aus von einer reellen Blockmatrix A der Dimension n=2p mit $\delta^->0$. A^N bezeichne die Matrix A nach der Durchführung des N-ten Zyklus *), und A_k^N die Matrix A nach der k-ten Transformation des (N+1)-ten Zyklus. Wir setzen für den ersten Teil eines Zyklus:

$$A^{O} = A , A_{O}^{N} = A^{N} , P = \frac{p(p-1)}{2} ,$$

$$A_{k}^{N} := U_{k}^{T} T_{k}^{-1} A_{k-1}^{N} T_{k} U_{k} , k = 1(1)P ,$$

$$T_{k} = T(\kappa_{k}; X_{v}), v \in \{2,3\} ,$$

$$U_{k} = U(\kappa_{k}; \varphi_{1}, \varphi_{2}, \varphi_{3}, \varphi_{4}) ,$$

wobei κ_k in zeilenzyklischer Pivotstrategie (2.09), T_k nach (2.1.1), (2.1.17) bezw. (2.1.44) und U_k nach (2.2.2 ... 7) bestimmt sind; X_{ν} , $\nu \in \{2,3\}$, sowie ϕ_1 , ϕ_2 , ϕ_3 und ϕ_4 hängen natürlich von k und N ab, was in (3.2.1) aus Gründen der Übersichtlichkeit nicht angedeutet ist. Die Wahlmöglichkeit $\nu \in \{2,3\}$ ist so zu verstehen, daß für die Durchführung eines Verfahrens stets $\nu = 2$ oder stets $\nu = 3$ gewählt wird; es ergibt sich für jedes ν ein anderes Verfahren.

Für den zweiten Teil eines Zyklus setzen wir

$$A_k^N := (0_k S_k)^{-1} A_{k-1}^N 0_k S_k , k = P+1(1)P+p ,$$

$$A^{N+1} := A_{P+p}^N ,$$

wobei

$$0_k = 0((2(k-P)-1, 2(k-P)); \varphi) \text{ nach } (2.4.1), (2.4.2)$$

und

^{*)} Unter einem Zyklus verstehen wir die durch (3.2.1) und (3.2.2) definierte Folge von P+p Transformationen, P = (p(p-1))/2.

$$S_k = S((2(k-P)-1, 2(k-P)); \psi) \text{ nach } (2.4.3), (2.4.4)$$

bestimmt werden.

Der obere Index N wird aus Gründen der Übersichtlichkeit weggelassen, wenn nur Transformationen eines festen Zyklus betrachtet werden. Bei den vorbereitenden Lemmata (Abschnitte 3.2.2 und 3.2.4) wird auch der Index k weggelassen, und statt dessen A' bzw. A''für die Matrix nach der ersten bzw. zweiten Transformation geschrieben (vergl.Kap. 2).

Wir sind an den Eigenwerten der Matrix A interessiert, so daß wir die quadratische Konvergenz der Näherungen für die Eigenwerte gegen die Eigenwerte zu zeigen haben. Der Abstand der Eigenwerte der Diagonalblöcke A_{ii} von den entsprechenden Eigenwerten von A kann abgeschätzt werden durch die Radien $r_{j}^{(i)}$ der für Blockmatrizen verallgemeinerten Gerschgorin-Kreise (vergl. (2.1.22),[21]), und für diese Radien gilt

$$r_{j}^{(i)} \le 0.9 \| C(D) \|^{1/2} + \sqrt{p-1} S(A)$$

Es genügt also, die quadratische Konvergenz von $\mathbb{C}(D)$ und S(A) zu zeigen. Da dieser Beweis sehr umfangreich ist, geben wir hier zunächst eine Beweisskizze mit Landauschen Symbolen (vergl.[34]). Sei A eine Matrix, deren Eigenwerte getrennte Imaginärteile haben (d.h. $\delta^->0$). Wir setzen für ein $\epsilon_N>0$, ϵ_N hinreichend klein, voraus:

$$S(A^N) = O(\epsilon_N)$$
 , $||C(D^N)|| = O(\epsilon_N)$,

und haben zu zeigen:

$$S(A^{N+1}) = O(\varepsilon_N^2)$$
 , $||C(D^{N+1})|| = O(\varepsilon_N^2)$.

Zunächst zeigen wir in den vorbereitenden Lemmata unter Anwendung der in den Abschnitten 2.1.2 und 2.1.3, sowie § 2.2 bereitgestellten Hilfsmittel, daß nach einer kompletten Transformation im ersten Teil eines Zyklus an einem festen Pivotpaar (r,s): $S(A'') = O(\epsilon_N)$ gilt. Wegen

$$\widehat{T}_{rs} = \begin{pmatrix} I & O(\varepsilon_{N}) \\ O & I \end{pmatrix} \quad \text{and} \quad \widehat{U}_{rs} = \begin{pmatrix} I + O(\varepsilon_{N}^{2}) & O(\varepsilon_{N}) \\ O(\varepsilon_{N}) & I + O(\varepsilon_{N}^{2}) \end{pmatrix},$$

da die Imaginärteile der Eigenwerte von A getrennt sind, gilt: A_{rr}'' , $A_{rr}' = A_{rr} + O(\epsilon_N^2)$; A_{ss}'' , $A_{ss}' = A_{ss} + O(\epsilon_N^2)$, und damit:

$$\|\,C\,(A_{\mathtt{j}\,\mathtt{j}}^{\prime\prime})\,\|\,\,\,,\,\,\|\,C\,(A_{\mathtt{j}\,\mathtt{j}}^{\,\prime})\,\|\,\,=\,\,\|\,C\,(A_{\mathtt{j}\,\mathtt{j}}^{\,\prime})\,\|\,\,+\,\,O\,(\epsilon_{N}^{2})\,\,\,,\,\,\,j\!\in\!\{\,r\,,s\,\}\,\,\,.$$

Daraus folgt $\|C(\underline{D}'')\| = O(\epsilon_N)$.

Da nach einer T-Transformation $\|C'_{rs}\| = O(\epsilon_N^3)$, und nach einer U-Transformation für den Außerdiagonalblock \widetilde{C}''_{rs} von $C(\widehat{A}''_{rs})$: $\|\widetilde{C}''_{rs}\| = \|\widetilde{C}'_{rs}\| + O(\|\widetilde{C}'_{rr}\| + \|\widetilde{C}'_{ss}\|) O(\epsilon_N)$, sowie $\|\widetilde{C}'_{jj}\| = C(A_{jj}) + O(\epsilon_N^2)$ und $\|\widetilde{C}'_{rs}\| = \|C'_{rs}\| + O(\epsilon_N^2)$ gilt, erhält man $\|\widetilde{C}''_{rs}\| = O(\epsilon_N^2)$. Weil $A''_{rs} = O$ ist, und die Imaginärteile der Eigenwerte von A getrennt sind, folgt daraus:

$$\|A_{rs}^{"}\| = \|A_{sr}^{"}\| = O(\epsilon_{N}^{2})$$
.

Aus $S(A'') = O(\epsilon_N)$ und $||C(D'')|| = O(\epsilon_N)$ erhält man für den ersten Teil eines Zyklus:

$$S(A_k) = O(\epsilon_N)$$
 , $||C(D_k)|| = O(\epsilon_N)$, $k = O(1)P$.

Unter Anwendung dieser Aussagen und einer Abschätzung des Normzuwachses der Nichtpivotblöcke* wird dann, einer Idee von Ruhe [28] folgend, im zweiten Hauptsatz

$$S(A_p) = O(\epsilon_N^2)$$

gezeigt.

Im zweiten Teil eines Zyklus erhält man nach der ersten (orthogonalen) Transformation S(A') = S(A) und mit

$$C(A'_{rr}) = \begin{pmatrix} \widetilde{c}'_{11} & \widetilde{c}'_{12} \\ \widetilde{c}'_{12} & \widetilde{c}'_{22} \end{pmatrix} , \text{ we gen } \widetilde{c}'_{11} = -\widetilde{c}'_{22} , c'_{11} = c'_{mm} ,$$

$$1 = 2r-1, m = 2r$$

^{*)} Vergl. Lemma 3.2.8 und Lemma 3.2.9 .

Da die Imaginärteile der Eigenwerte von A getrennt sind, folgt weiter $a'_{11} - a'_{mm} = O(\epsilon_N)$, $a'_{1m} - a'_{ml} = O(1)$, und damit S(A'') = S(A) $(1 + O(\epsilon_N))$, wegen th $\psi = -(a'_{mm} - a'_{11})/(2(a'_{m1} - a'_{1m})) + O(\epsilon_N^2) = O(\epsilon_N)$. Die nach dem ersten Teil eines Zyklus erreichte Ordnung $S(A_p) = O(\epsilon_N^2)$ wird im zweiten Teil also nicht zerstört, d.h.

$$S(A^{N+1}) = O(\epsilon_N^2)$$
.

Wegen obiger Darstellung von th ψ gilt nach der zweiten (hyperbolischen) Transformation: $|\widetilde{c}_{12}''| = 0(\epsilon_N^2) \text{ und wegen}$ $|\widetilde{c}_{11}'| = |\widetilde{c}_{22}'| = 0(\epsilon_N^4) \text{ auch } |\widetilde{c}_{11}''| = |\widetilde{c}_{22}''| = 0(\epsilon_N^4) \text{ . Weil die Diagonalblöcke, die nicht Pivotblock sind, sich nicht ändern, gilt nach dem zweiten Teil eines Zyklus:$

$$\|C(D^{N+1})\| = O(\epsilon_N^2)$$
.

Damit sind die Vorbetrachtungen abgeschlossen, und wir beginnen mit dem vorbereitenden Lemmata.

3.2.2 Vorbereitende Lemmata

Nach den Vorbemerkungen des ersten Abschnitts werden nun die für den Beweis des zweiten Hauptsatzes benötigten Lemmata bewiesen.

Lemma 3.2.1

Sei A eine beliebige reelle Blockmatrix mit $\delta > 0$, und (r,s) fest, sowie

$$\varepsilon \leq \min\{\varepsilon_{0}, \varepsilon_{2}, \varepsilon_{3}, \varepsilon_{v-1}^{-}\}, v \in \{2,3\},$$

$$S(A) \leq \frac{1}{4} \|A\| \varepsilon$$

$$\|C(D)\| \leq \frac{\delta^{2}}{64}$$

$$\Delta(A) \leq \frac{\delta^{-}}{16}$$

$$A' = T((r,s); X_{v})^{-1} A T((r,s); X_{v})$$

$$A'' = U((r,s); \varphi_{1}, \varphi_{2}, \varphi_{3}, \varphi_{4})^{T} A'U((r,s); \varphi_{1}, \varphi_{2}, \varphi_{3}, \varphi_{4})$$

wobei T die Johnsen-Veselić-Voevodin-Matrix ((2.1.1),(2.1.17) bzw. (2.1.44)) und U die Paardekooper-Jacobi-Matrix (2.2.2 ... 2.2.7) sind. Dann gilt

$$(3.2.4)$$
 $S^{2}(A'') \leq H_{y}S^{2}(A)$,

mit

$$\begin{split} & H_{_{\mathcal{V}}} := a_{_{\mathcal{V}}} + \widetilde{k}_{_{\mathcal{V}}}^{\; 2} + \widetilde{1}_{_{\mathcal{V}}}^{\; 2} \quad , \quad \nu \in \{2,3\} \\ & \widetilde{k}_{_{\mathcal{V}}} := g_{_{\mathcal{V}}} \; + \; 2.0364 \quad \widetilde{K}_{_{\mathcal{V}}} + \; 2.0627 \quad , \\ & \widetilde{1}_{_{\mathcal{V}}} := g_{_{\mathcal{V}}} \; + \; 0.0263 \quad \widetilde{K}_{_{\mathcal{V}}} + \; 2.0627 \quad , \\ & g_{_{\mathcal{V}}} := 6.4676 \quad b_{_{\mathbf{TS}}}^{\; (\nu)} \; \frac{\|\mathbf{A}\|}{\delta^{-}} \quad , \\ & a_{_{\mathcal{V}}} \; \text{aus Lemma 2.1.13.} \end{split}$$

Beweis: Es ist nach den Formeln der T-Transformation (s.(2.1.3)):

$$S^{2}(A') - S^{2}(A) \leq \sum_{j \neq r, s}^{p} \left(\|A_{jr}\|^{2} \|X_{v}\|^{2} + 2\|A_{jr}\|\|A_{js}\|\|X_{v}\| + \|A_{sj}\|^{2} \|X_{v}\|^{2} + 2\|A_{sj}\|\|A_{ri}\|\|X_{v}\| \right) + \|A_{rs}'\|^{2} - \|A_{rs}\|^{2},$$

also

$$S^{2}(A') \leq S^{2}(A) (1+\|X_{y}\| + \|X_{y}\|^{2}) + \|A_{rs}^{\dagger}\|^{2} - \|A_{rs}\|^{2}$$
.

Nach einer folgenden U-Transformation gilt dann:

$$(3.2.5) S2(A'') = S2(A') + ||A''rs||2 + ||A''rs||2 - ||A'rs||2 - ||A'rs||2 \leq avS2(A) + ||A''rs||2 + ||A''rs||2 - ||Ars||2 - ||Asr||2 .$$

Es bleiben $\|A''_{rs}\|$ und $\|A''_{sr}\|$ abzuschätzen, was nach den Formeln der U-Transformation (s.(2.2.13)) und den Lemmata 2.1.12, 2.2.3 und 2.2.4 wie folgt geschieht:

$$\|A_{rs}^{"}\| \leq \left(\|U_{rr}\| \|A_{rr}^{'}\| + \|U_{ss}\| \|A_{ss}^{'}\| \right) b_{rs}^{(v)} \frac{S(A)}{\delta}$$

$$+ \|U_{rs}\| \|U_{sr}\| S(A) + \|U_{rr}\| \|U_{ss}\| \left(\|A_{rs}\| + \widetilde{K}_{v}S(A) \right)$$

$$\leq \left(\max\{ \|U_{rr}\|, \|U_{ss}\| \} \sqrt{2} \frac{\|A\|}{\delta} b_{rs}^{(v)} + \|U_{rs}\| \|U_{sr}\| \right)$$

$$+ \|U_{rr}\| \|U_{ss}\| \left(1 + \widetilde{K}_{v} \right) \right) S(A)$$

$$\leq \left[1.4291 \sqrt{2} b_{rs}^{(v)} \frac{\|A\|}{\delta} + 1.4249 \cdot 1.4291 \left(1 + \widetilde{K}_{v} \right) + \frac{a^{2}}{49} \right] S(A)$$

$$\leq \widetilde{K}_{v} S(A) ,$$

mit $\|A'\| \le \|A\|$ wegen Satz 2.1.8 bzw. Satz 2.1.11 und Lemma 2.2.2.

Analog ergibt sich für ||A'' ||:

$$\|A_{sr}^{"}\| \leq \left[\max\{\|U_{rr}\|,\|U_{ss}\|\} \sqrt{2} b_{rs}^{(v)} \frac{\|A\|}{\delta} + \|U_{rr}\|\|U_{ss}\| + \|U_{rs}\|\|U_{sr}\| (1 + \widetilde{K}_{v}) \right] S(A) \leq \widetilde{1}_{v} S(A) .$$

Zusammen mit (3.2.5) folgt daraus die Behauptung (3.2.4), womit Lemma 3.2.1 bewiesen ist.

Aus diesem Lemma erhält man ein Korollar, das zum Beweis von Lemma 3.2.7 benötigt wird.

Korollar 3.2.2

Gilt zusätzlich zu den Voraussetzungen von Lemma 3.2.1:

$$(3.2.6) \qquad \varepsilon \leq \varepsilon_{\nu+1}^{-} := \frac{\delta^{-}}{4\|\mathbf{A}\|\sqrt{\mathbf{H}_{\nu}}} , \quad \nu \in \{2,3\} ,$$

so gilt

$$(3.2.7)$$
 $S(A'') \leq \frac{\delta^{-1}}{16}$.

Nachdem der maximale Zuwachs von S(A) unter einer Transformation im ersten Teileines Zyklus abgeschätzt ist, kommen wir zur Abschätzung der Änderung der Blockdiagonalen unter einer solchen Transformation.

Lemma 3.2.3

Sei A eine reelle Blockmatrix mit $\delta > 0$, die (3.2.3) erfüllt. Dann gilt für ein festes Pivotpaar (r,s):

$$(3.2.8) A''_{rr} = A_{rr} + R_{rr},$$

$$(3.2.9)$$
 $A''_{ss} = A_{ss} + R_{ss}$,

mit

Beweis: Analog zu Lemma 2.1.12 erhält man

Daraus folgt mit den Formeln für die T-Transformation:

$$\|A_{rj}' - A_{rj}\| \leq \frac{\widetilde{K}_{v}}{\sqrt{3} \|A\|} S^{2}(A) , j \neq s ,$$

$$(3.2.12)$$

$$\|A_{is}' - A_{is}\| \leq \frac{\widetilde{K}_{v}}{\sqrt{3} \|A\|} S^{2}(A) , i \neq r .$$

Für eine U-Transformation gilt mit $U_{rr} = I + U'_{rr}$:

$$(3.2.13)$$
 $A''_{rr} = A'_{rr} + R'_{rr}$,

mit

$$\leq \|A\| \left((2 + \|U_{rr}\|) \|U_{rr}\| + \|U_{sr}\|^2 \right) + \|U_{rr}\| \|U_{sr}\| (\sqrt{2} + \widetilde{K}_{v}) S(A)$$

$$\leq \left((2.0259 \ b_{r}^{(v)} + b_{rs}^{(v)})^2 \right) \frac{\|A\|}{\widetilde{\delta}^2} + 1.4249 \ b_{rs}^{(v)} \frac{\sqrt{2} + \widetilde{K}_{v}}{\widetilde{\delta}^2} v \right) S^2(A)$$

$$\leq \left(0.7004 \ \frac{\|A\|}{\widetilde{\delta}} + 0.8077 \right) \frac{(\sqrt{2} + \widetilde{K}_{v})^2}{\widetilde{\delta}^2} S^2(A) ,$$

wobei die Lemmata 2.1.12, 2.2.3, 2.2.4 und 2.2.5 benutzt werden. Zusammen mit (3.2.12) folgt daraus

$$A''_{rr} = A'_{rr} + R'_{rr} = A_{rr} + R_{rr}$$

mit

$$\|\mathbf{R}_{rr}\| \le \frac{d_{v}}{\|\mathbf{A}\|} S^{2}(\mathbf{A})$$

Die etwas schlechteren Konstanten in (3.2.10) ergeben sich aus der analog durchzuführenden Abschätzung von R'_{ss} . Damit ist das Lemma 3.2.3 bewiesen.

Lemma 3.2.4

Sei A eine reelle Blockmatrix mit $\delta > 0$, die (3.2.3) erfüllt. Dann gilt für den Kommutator der Diagonalblöcke A_{jj} , $j \in \{r,s\}$ und festes Pivotpaar $\{r,s\}$:

$$(3.2.14)$$
 $\|C(A''_{jj})\| \le \|C(A_{jj})\| + e_{v} S^{2}(A)$,

(3.2.15)
$$\|C(A_{jj}')\| \le \|C(A_{jj})\| + c_{v} S^{2}(A)$$
,

mit

$$e_{v} = 2 \left(2 \|A\| + \frac{\delta^{2} d_{v}}{4096 \|A\|}\right) \frac{d_{v}}{\|A\|}, \quad v \in \{2,3\},$$

$$c_{2} = \frac{244}{\delta^{2}} \|A\|^{2}, \quad c_{3} = \frac{367}{\delta^{2}} \|A\|^{2}.$$

Beweis: Mit den Bezeichnungen von Lemma 3.2.3 ist für $j \in \{r,s\}$:

wobei in die letzte Ungleichung (3.2.3) eingeht und ϵ durch ϵ_0 abgeschätzt wird. Damit ist (3.2.14) gezeigt. Analog zu (3.2.16) erhält man mit (3.2.11), $j \in \{r,s\}$ und $\nu = 2$:

$$||C(A_{jj})|| \le ||C(A_{jj})|| + 2 \left(2||A|| + \frac{128\sqrt{2}}{3\delta^2} ||A|| S^2(A)\right) \frac{128\sqrt{2}}{3\delta^2} ||A|| S^2(A)$$

$$\le ||C(A_{jj})|| + 2 \left(2||A|| + \frac{\sqrt{2}}{92} ||A||\right) \frac{128\sqrt{2}}{3\delta^2} ||A|| S^2(A)$$

$$\le ||C(A_{jj})|| + c_2 S^2(A) .$$

Für v = 3 ergibt sich (3.2.15) analog, womit Lemma 3.2.4 bewiesen ist.

Nach diesen Vorbereitungen kann $\|C(D'')\|$ abgeschätzt werden.

Lemma 3.2.5

Sei A eine reelle Blockmatrix mit $\delta > 0$, die (3.2.3) erfüllt. Dann gilt mit A = D + E (s.(1.1.22)) für ein festes Pivotpaar (r,s):

Beweis: Es ist mit Lemma 3.2.4 und Voraussetzung (3.2.3):

$$\|C(D'')\|^{2} = \sum_{i=r,s}^{p} \|C(A''_{ii})\|^{2} + \|C(A''_{rr})\|^{2} + \|C(A''_{ss})\|^{2}$$

$$\leq \sum_{i \neq r,s}^{p} \|C(A_{ii})\|^{2} + (\|C(A_{rr})\| + e_{v}S^{2}(A))^{2}$$

$$+ (\|C(A_{ss})\| + e_{v}S^{2}(A))^{2}$$

$$\leq \sum_{i=1}^{p} \|C(A_{ii})\|^{2} + 2(\sqrt{2}\|C(D)\|e_{v}S^{2}(A) + e_{v}^{2}S^{4}(A))$$

$$\leq \|C(D)\|^{2} + 2(\sqrt{2}\frac{\delta^{2}}{64} + e_{v}\frac{\delta^{4}}{256^{2}\|A\|^{2}}) e_{v}S^{2}(A)$$

$$= \|C(D)\|^{2} + h_{v}\|A\|^{2}S^{2}(A) ,$$

womit Lemma 3.2.5 bewiesen ist.

Mit diesem Lemma ist auch der Zuwachs des Kommutators der Blockdiagonalen unter einer Transformation im ersten Teil eines Zyklus nach oben abgeschätzt. Es folgt die Abschätzung des Kommutators am Pivotpaar (r,s) nach einer solchen Transformation, womit dann auch die Pivotblöcke A''_{rs} und A''_{sr} abgeschätzt werden können.

Lemma 3.2.6

Sei A eine reelle Blockmatrix mit $\delta > 0$, die (3.2.3) erfüllt. Dann gilt für ein festes Pivotpaar (r,s) mit

$$(3.2.19) C(\widehat{A}_{rs}) =: \begin{pmatrix} \widetilde{C}_{rr} & \widetilde{C}_{rs} \\ \widetilde{C}_{rs}^{T} & \widetilde{C}_{ss} \end{pmatrix} :$$

mit

$$g_{v}$$
 aus Lemma 3.2.1 , $v \in \{2,3\}$,
$$f_{2} := f + 0.5708 \ b_{rs}^{(2)} (c_{2} + (1 + \widetilde{K}_{2})^{2}) ,$$

$$f_{3} := 2.0626 + 0.5708 \ b_{rs}^{(3)} (c_{3} + (1 + \widetilde{K}_{3})^{2}) ,$$

$$f := 2.0626 \ (\frac{5}{6} + \frac{c_{2}}{2}) .$$

Beweis: Nach einer Paardekooper-Transformation gilt für \tilde{C}''_{rs} unter Anwendung der Lemmata 2.2.3 und 2.2.4:

$$\begin{split} \|\widetilde{C}_{rs}^{"}\| & \leq \|U_{rr}\| (\|U_{ss}\|\|\widetilde{C}_{rs}^{'}\| + \|U_{rs}\|\|\widetilde{C}_{rr}^{'}\|) + \\ & + \|U_{ss}\|\|U_{sr}\|\|\widetilde{C}_{ss}^{'}\| + \|U_{rs}\|\|U_{sr}\|\|\widetilde{C}_{rs}^{'}\| \\ & \leq 1.4249 (1.4291\|\widetilde{C}_{rs}^{'}\| + b_{rs}^{(v)} \frac{S(A)}{\delta} \|\widetilde{C}_{rr}^{'}\|) + \\ & + 1.4291 b_{rs}^{(v)} \frac{S(A)}{\delta} \|\widetilde{C}_{ss}^{'}\| + \frac{a^{2}}{49} \|\widetilde{C}_{rs}^{'}\| \\ & \leq 2.0626 \|\widetilde{C}_{rs}^{'}\| + (1.4249\|\widetilde{C}_{rr}^{'}\| + 1.4291\|\widetilde{C}_{ss}^{'}\|)b_{rs}^{(v)} \frac{S(A)}{\delta} . \end{split}$$

Es bleibt abzuschätzen, wie groß die Elemente von $C(\widehat{A}_{rs})$ nach einer T-Transformation sein können. Dazu schreiben wir:

$$\widetilde{C}'_{rr} = C(A'_{rr}) + A'_{sr}^{T}A'_{sr} - A'_{rs} A'_{rs}^{T},$$

$$\widetilde{C}'_{ss} = C(A'_{ss}) + A'_{rs}^{T}A'_{rs} - A'_{sr} A'_{sr}^{T},$$

$$\widetilde{C}'_{rs} = A'_{rr}^{T}A'_{rs} - A'_{rr}A'_{sr}^{T} + A'_{sr}^{T}A'_{ss} - A'_{rs}A'_{ss}^{T}.$$

Damit folgt nach Lemma 2.1.12 und den Formeln der T-Transformation:

wobei bei der letzten Ungleichung Lemma 3.2.4 eingeht. Auf die gleiche Weise ergibt sich auch

$$\|\widetilde{C}_{ss}^{\dagger}\| \le \|C(A_{ss})\| + (c_{v} + (1 + \widetilde{K}_{v})^{2}) S^{2}(A)$$

aus (3.2.22). Für \widetilde{C}'_{rs} folgt aus (3.2.22) wie oben:

$$\widetilde{C}'_{rs} = C'_{rs} - \sum_{i \neq r,s}^{p} \left(A'_{ir} A'_{is} - A'_{ri} A'_{si} \right) ,$$

$$||\tilde{C}'_{rs}|| \leq ||C'_{rs}|| + \sum_{i \neq r, s}^{p} (||A_{ir}|| (||A_{is}|| + ||X|||A_{ir}||) + ||A_{ri}|| + ||X|||A_{si}||) ||A_{si}||) + ||A_{ri}|| + ||X|||A_{si}||) ||A_{si}||)$$

$$\leq ||C'_{rs}|| + (\frac{1}{2} + ||X||) S^{2}(A) .$$

Daraus folgt für v=2 , d.h. $X=X_2$, wegen $\|X_2\| \le \frac{1}{3}$ und Satz 2.1.8, sowie Voraussetzung (3.2.3):

$$\|\widetilde{C}_{rs}'\| \leq \left(\frac{128^{2}(72\sqrt{2}+1)}{54\delta^{4}}\|A\|^{3}S(A) + \frac{1}{2} + \frac{1}{3}\right) S^{2}(A)$$

$$\leq \left(\frac{122}{\delta^{2}}\|A\|^{2} + \frac{5}{6}\right) S^{2}(A)$$

$$= \frac{f}{2.0626} S^{2}(A) .$$

Zusammen erhält man mit (3.2.21) für v = 2:

$$\begin{split} \|\widetilde{C}_{rs}^{"}\| & \leq f \, S^{2} \, (A) \, + \, \left(\, (1.4249 \| \, C \, (A_{rr}) \| \, + \, 1.4291 \| \, C \, (A_{ss}) \|) \, + \\ & + \, 2.854 \, (c_{2} \, + \, (1 \, + \, \widetilde{K_{2}})^{2}) \, S^{2} \, (A) \right) b_{rs}^{(2)} \, \frac{S(A)}{\delta} \\ & \leq \left[f \, + \, 0.5708 \, \left(c_{2} \, + \, (1 \, + \, \widetilde{K_{2}})^{2} \, b_{rs}^{(2)} \, \right) \right] \, S^{2} \, (A) \, + \\ & + \, 2.0211 \, \frac{b_{rs}^{(2)}}{\delta} \, \| \, C \, (D) \, \| \, S \, (A) \\ & = \, f_{2} \, S^{2} \, (A) \, + \, \frac{g_{2}}{\|A\|} \, \, S(A) \, \| \, C \, (D) \, \| \, \right] \, , \end{split}$$

wobei in die letzte Ungleichung $\delta \geq \frac{5}{16} \delta^- \geq 5 S(A)$ eingeht.

Entsprechend erhält man für v=3, d.h. $X=X_3$, aus (3.2.24) wegen $\|X_3\| \le \frac{1}{2}$ und $C'_{rs}=0$:

$$\|\widetilde{C}_{rs}^{\prime}\| \le \|C_{rs}^{\prime}\| + (\frac{1}{2} + \|X_3\|) S^2(A) = S^2(A)$$
,

und weiter mit (3.2.21):

$$\|\widetilde{C}_{rs}^{"}\| \leq f_3 S^2(A) + \frac{g_3}{\|A\|} \|C(B)\|S(A) .$$

Damit ist Lemma 3.2.6 bewiesen.

Lemma 3.2.7

Sei A eine reelle Blockmatrix mit δ > 0 , die (3.2.3) erfüllt, und es gelte zusätzlich $\varepsilon < \varepsilon_{\nu+1}$ (s.(3.2.6)). Dann gilt für ein festes Pivotpaar (r,s):

Beweis: Wir zerlegen $\widehat{A}_{rs}^{"} = \frac{1}{2}(\widehat{A}_{rs}^{"} + \widehat{A}_{rs}^{"}) + \frac{1}{2}(\widehat{A}_{rs}^{"} - \widehat{A}_{rs}^{"}) = \widehat{A}_{rs}^{"} + \widehat{A}_{rs}^{"}$; damit ist

$$C(\widehat{A}_{rs}^{"}) = 2(\widehat{A}_{rs}^{"+}\widehat{A}_{rs}^{"-} - \widehat{A}_{rs}^{"-}\widehat{A}_{rs}^{"+})$$
,

und wegen $A_{rs}^{"-} = -A_{sr}^{"-T} = 0$ infolge der U-Transformation ergibt sich daraus die Gleichung (s.(2.0.3),(3.2.19)):

$$(3.2.26) \frac{1}{2} \tilde{C}_{rs}^{"} = A_{rr}^{"-T} A_{rs}^{"+} - A_{rs}^{"+} A_{ss}^{"-T} .$$

Nach Korollar 3.2.2 ist $S(A'') \leq \frac{\delta^-}{16}$, und wegen der Invarianz der Euklidischen Norm unter einer U-Transformation und der Normabnahme unter einer T-Transformation folgt

$$\triangle(A'') \le \triangle(A) \le \frac{\delta^{-1}}{16}$$

nach Voraussetzung (3.2.3). Damit sind die Voraussetzungen von Lemma 2.2.2 für A'' erfüllt, und es gilt für den Abstand der Spektren $\pm i \mu_j$ der Diagonalblöcke A''- des schiefsymmetrischen Teils A''- von A'' mit $\mu_j > 0$, j = 1(1)p:

$$\min\{|\mu_{i} - \mu_{j}| | i \neq j\} \ge \frac{7}{8} \delta^{-}.$$

Damit ist Lemma 1.1.24 auf die Gleichung (3.2.26) anwendbar, und es folgt wegen $\Delta(A_{rr}^{"-T}) = \Delta(A_{ss}^{"-T}) = 0$:

$$\|A_{rs}^{"+}\| \leq \frac{\|\widetilde{C}_{rs}^{"}\|}{2|\mu_{r}-\mu_{s}|} \leq \frac{4}{7\delta^{-}} \|\widetilde{C}_{rs}^{"}\|,$$

und daraus folgt mit Lemma 3.2.6 die Behauptung.

Nachdem die Pivot-B1öcke abgeschätzt sind, benötigen wir noch zwei technische Lemmata über die maximale Änderung der Außerblockdiagonale von A und eine Abschätzung von S(A) und C(D) über den ersten Teil eines Zyklus (3.2.1), bevor der zweite Hauptsatz bewiesen werden kann.

Lemma 3.2.8

Sei A eine reelle Blockmatrix mit δ > 0 , die (3.2.3) erfüllt. Dann gilt für $\{i,j\}\cap\{r,s\} = \emptyset$, $v\in\{2,3\}$, $\{r,s\}$ fest:

$$\|A_{rj}^{"}\|^{2} + \|A_{sj}^{"}\|^{2} \leq a_{v}(\|A_{rj}\|^{2} + \|A_{sj}\|^{2})$$

$$\|A_{ir}^{"}\|^{2} + \|A_{is}^{"}\|^{2} \leq a_{v}(\|A_{ir}\|^{2} + \|A_{is}\|^{2}),$$

mit a_{v} aus Lemma 2.1.13.

Beweis: Dem Beweis von Lemma 3.2.1 entnimmt man:

$$\|A_{rj}^{'}\|^{2} \le \|A_{rj}^{2}\|^{2} + (\|A_{rj}^{2}\|^{2} + \|A_{sj}^{2}\|^{2})\|X_{v}\| + \|A_{sj}^{2}\|^{2}\|X_{v}\|^{2}$$
, $j \neq s$,

und damit ist wegen $\|A'_{sj}\| = \|A_{sj}\|$, $j \neq s$:

$$\|A_{rj}^{'}\|^{2} + \|A_{sj}^{'}\|^{2} \leq (1 + \|X_{v}\| + \|X_{v}\|^{2}) (\|A_{rj}\|^{2} + \|A_{sj}\|^{2})$$

$$\leq a_{v} (\|A_{rj}\|^{2} + \|A_{sj}\|^{2}) .$$

Wegen der Orthogonalität der U-Transformation ist für $j \notin \{r,s\}$:

$$\|A_{rj}^{"}\|^2 + \|A_{sj}^{"}\|^2 = \|A_{rj}^{"}\|^2 + \|A_{sj}^{"}\|^2$$

und damit folgt der erste Teil der Behauptung (3.2.27). Analog erhält man den zweiten Teil der Behauptung, womit das Lemma bewiesen ist.

Lemma 3.2.9

Sei A_k eine reelle Blockmatrix mit $\delta > 0$, die (3.2.3) erfüllt. Dann gilt mit $v \in \{2,3\}$, (r,s) fest:

mit

L := 1.0916,
$$m_{v}^{(k)} := \left(0.5835 \frac{\tilde{K}_{v}}{\|A\|} + \frac{16 b_{rs}^{(v)}}{5 \delta^{-}}\right) S(A_{k})$$

Beweis: Nach den Transformationsformeln (2.1.3) und (3.2.11) gilt für $v \in \{2,3\}$:

$$\|A_{rj}^{(k)'}\| \leq \|A_{rj}^{(k)}\| + \frac{\widetilde{K}_{v}}{\sqrt{3}\|A\|} S(A_{k}) \|A_{sj}^{(k)}\|, j \neq s$$

$$\|A_{is}^{(k)'}\| \leq \|A_{is}^{(k)}\| + \frac{1}{2} \|A_{ir}^{(k)}\|, \text{ we gen } \|X_{v}\| \leq \frac{1}{2}, i \neq r.$$

Weiter gilt nach den Transformationsformeln (2.2.12), (3.2.29) und Voraussetzung (3.2.3):

$$\|A_{rj}^{(k+1)}\| \leq \|(U_{rr}^{dT} + U_{rr}^{aT}) A_{rj}^{(k)'}\| + \|U_{sr}^{T} A_{sj}^{(k)'}\|, j \notin \{r,s\}$$

$$\leq \|U_{rr}^{d}\|_{sp} \left(\|A_{rj}^{(k)}\| + \frac{\tilde{K}_{v}}{\sqrt{3}\|A\|} S(A_{k})\|A_{sj}^{(k)}\| \right) +$$

$$+ \|U_{rr}^{a}\| \left(\|A_{rj}^{(k)}\| + \frac{\tilde{K}_{v}}{\sqrt{3}\|A\|} S(A_{k})\|A_{sj}^{(k)}\| \right) +$$

$$+ \|U_{sr}\|\|A_{sj}^{(k)}\|$$

$$= (\|U_{rr}^{d}\|_{sp} + \|U_{rr}^{a}\|)\|A_{rj}^{(k)}\| +$$

$$+ \left(\| \mathbf{U}_{rr}^{d} \|_{sp} + \| \mathbf{U}_{rr}^{a} \|_{1} + \frac{\tilde{K}_{v}}{\sqrt{3} \| A \|} + \frac{b_{rs}^{(v)}}{\tilde{\delta}} \right) S(A_{k}) \| A_{sj}^{(k)} \|_{2}$$

$$\leq L \| A_{rj}^{(k)} \| + m_{v}^{(k)} \| A_{sj}^{(k)} \|_{2} ,$$

wobei zunächst Lemma 2.2.3 und dann Lemma 2.2.4 zur Abschätzung angewandt wird. Entsprechend erhält man den zweiten Teil der Behauptung, womit das Lemma 3.2.9 bewiesen ist.

Es wird nun eine Aussage darüber erforderlich, wie klein S(A) und $\|C(D)\|$ sein müssen, damit die Bedingungen (3.2.3) für alle iterierten Matrizen A_k während des ersten Teils eines Zyklus garantiert werden können. Dazu dient das folgende

Lemma 3.2.10.

Sei A eine reelle Blockmatrix mit $\delta > 0$, und es gelte:

$$\widetilde{\varepsilon} \leq \min\{\varepsilon_{0}, \varepsilon_{2}, \varepsilon_{3}, \varepsilon_{0}^{-}, \varepsilon_{\nu-1}^{-}, \varepsilon_{\nu+1}^{-}, \varepsilon_{\nu+3}^{-}\} ,$$

$$\varepsilon_{0}^{-} := \frac{(\delta^{-})^{2}}{64 \|A\|^{2}} , \quad \varepsilon_{\nu+3}^{-} := \frac{(\delta^{-})^{2}}{16 \|A\|^{2}} \left(\frac{(P^{2}-1)(H_{\nu}-1)}{P^{2} h_{\nu}}\right)^{1/2}, \nu \in \{2,3\},$$

$$(3.2.30)$$

$$S(A) \leq \left(\sqrt{H_{\nu}}\right)^{-P} \frac{1}{4} \|A\|^{2} \tilde{\varepsilon} ,$$

$$\|C(D)\| \leq \frac{1}{P} \|A\|^{2} \tilde{\varepsilon} ,$$

$$\Delta(A) \leq \frac{\delta^{-}}{16} .$$

Dann gilt:

$$S(A_{k}) \leq \frac{1}{4} \|A\| \tilde{\varepsilon} ,$$

$$\|C(D_{k})\| \leq \frac{(\delta^{-})^{2}}{64} ,$$

$$\Delta(A_{k}) \leq \frac{\delta^{-}}{16} , \qquad k = O(1)P .$$

Beweis: Der Beweis wird induktiv über k geführt. Für k = 0 sind die Aussagen (3.2.31) nach Voraussetzung (3.2.30) wegen $H_{y} \ge 1$ und

 $P \ge 1$ mit $A_o = A$ erfüllt. Es seien nun für alle $1 \le k < P$ die Aussagen

$$S(A_1) \le \frac{1}{4} \|A\| \tilde{\epsilon}$$
 , $\|C(D_1)\| \le \frac{(\delta^-)^2}{64}$, $\Delta(A_1) \le \frac{\delta^-}{16}$

erfüllt. Dann ist Satz 2.1.8 bzw. Satz 2.1.11 auf A_k anwendbar, und es gilt: $\|A_{k+1}\| \le \|A_k\|$, und damit

$$\Delta(A_{k+1}) \leq \Delta(A_k) \leq \Delta(A_0) \leq \frac{\delta^{-1}}{16}$$
.

Weiter können die Lemmata 3.2.1 und 3.2.5 auf $A_{\mathbf{k}}$ angewendet werden, und es gilt:

$$S(A_{k+1}) \le \sqrt{H_{v}} S(A_{k}) \le (\sqrt{H_{v}})^{k+1} S(A_{o})$$

$$\le \frac{1}{4} \|A\| \tilde{\epsilon}$$

nach Voraussetzung (3.2.30).

Weiter gilt nach Lemma 3.2.5 mit Lemma 3.2.1:

also $\|C(D_{k+1})\| \le \frac{(\delta^-)^2}{64}$. Damit ist der Induktionsschluß vollzogen und Lemma 3.2.10 bewiesen.

Aus diesem Lemma ergibt sich das

Korollar 3.2.11.

Die Voraussetzungen von Lemma 3.2.10 seien für eine reelle Blockmatrix A und die Matrix nach dem N-ten Zyklus A^N erfüllt, und es gelte

$$S\left(A^{N}\right) \leq \frac{1}{4} \|A\| \epsilon_{N} ,$$

$$(3.2.32)$$

$$\|C\left(D^{N}\right)\| \leq \|A\|^{2} \epsilon_{N} .$$

Dann gilt für $v \in \{2,3\}$:

$$(3.2.33) \qquad \max \left\{ S\left(A_{k}^{N}\right) \mid 0 \le k \le P \right\} \le \frac{1}{4} \|A\| \left(\sqrt{H_{v}}\right)^{P} \varepsilon_{N}$$

und

$$(3.2.34) \qquad \max\{\|C(\mathbb{D}_k^N)\| | 0 \le k \le P\} \le w_{\mathcal{V}} \|A\|^2 \varepsilon_{N},$$

mit

$$w_{v} := \left(1 + \frac{h_{v}(H_{v}^{P} - 1)}{16(H_{v} - 1)}\right)^{1/2}$$

Nach diesen Vorbereitungen kann jetzt im nächsten Abschnitt der zweite Hauptsatz, daß nach dem ersten Teil eines Zyklus S(A) quadratisch klein ist, bewiesen werden.

3.2.3 Der zweite Hauptsatz

In diesem Abschnitt wird gezeigt, daß nach dem ersten Teil eines Zyklus (s.(3.2.1)) die Außerblockdiagonale unter gewissen Voraussetzungen quadratisch klein ist.

Satz 3.2.12 (zweiter Hauptsatz)

Sei A eine reelle Blockmatrix mit $\delta > 0$, die so weit blockdiagonalisiert und normalisiert ist, daß sie (3.2.30) erfüllt.

Sei A^N die Matrix nach Durchführung des N-ten Zyklus gemäß (3.2.1) und (3.2.2), und erfülle A^N die Bedingungen (3.2.30) und

$$S(A^N) \le \frac{1}{4} \|A\| \epsilon_N$$
 , $\|C(D^N)\| \le \|A\|^2 \epsilon_N$.

Dann existiert eine Konstante $K^{(v)}(\|A\|, \delta, \delta^-, p)$, die nur von der Matrix A abhängt, so daß

(3.2.35)
$$S(A_p^N) \leq K^{(v)}(\|A\|,\delta,\delta^-,p)\|A\|\epsilon_N^2$$
, $v \in \{2,3\}$

gilt.

Beweis: Der Beweis folgt einer Idee von Ruhe [28]. Da wir hier nur einen festen Zyklus betrachten, lassen wir den oberen Index N weg und setzen abkürzend mit $A_k = (A_{ij}^{(k)})$:

$$(3.2.36) S_{ij}^{(k)} := \|A_{ij}^{(k)}\|^2 + \|A_{ji}^{(k)}\|^2$$

und

$$(3.2.37) \qquad \varepsilon^2 := \max \left\{ \frac{S(A_k)^2}{\|A\|^2}, \frac{S(A_k)\|C(D_k)\|}{\|A\|^3} \mid 0 \le k \le P \right\}$$

Zuerst beweisen wir durch Induktion über r die

Zwischenbehauptung:

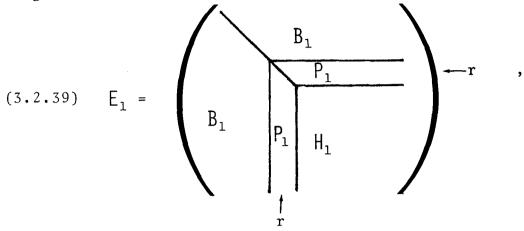
Während des N-ten Iterationszyklus (3.2.1) existiert für jedes r = O(1)p-1 eine Konstante $K_r^{(v)}(\|A\|,\delta,\delta^-,p) = K_r^{(v)}$, so daß für k = k(r) mit $\kappa_{k+1} = (r+1,r+2)^*$:

(3.2.38)
$$\sum_{i=1}^{r} \sum_{j=i+1}^{p} S_{ij}^{(k)} \leq K_{r}^{(v)} \|A\|^{2} \epsilon^{4}$$

gilt.

*) κ_{k+1} aus (2.0.4), d.h. man betrachtet die Matrix, nachdem der Pivotindex die r-te Zeile durchlaufen hat.

Beweis der Zwischenbehauptung: Wir teilen für $k-p+r \le 1 \le k$ den nichtblockdiagonalen Teil E_1 der Matrix A_1 gemäß der folgenden Skizze auf:



d.h. es werden Matrizen B_1 , P_1 und H_1 definiert durch

$$B_{ij}^{(1)} := \begin{cases} A_{ij}^{(1)} & \text{für } 1 \le i < r \text{ oder } 1 \le j < r \text{, } i \neq j \\ \\ 0 & \text{sonst ,} \end{cases}$$

$$P_{ij}^{(1)} := \begin{cases} A_{ij}^{(1)} & \text{für } i = r, i < j \le p \text{ oder } j = r, j < i \le p \\ \\ 0 & \text{sonst }, \end{cases}$$

$$H_{ij}^{(1)} := \begin{cases} A_{ij}^{(1)} & \text{für } r < i \le p \text{, } r < j \le p \text{, } i \ne j \\ \\ 0 & \text{sonst ,} \end{cases}$$

$$E_1 = B_1 + P_1 + H_1$$
.

Dabei besteht für 1=k-p+r, d.h. $\kappa_{1+1}=(r,r+1)$, β_1 aus den Blöcken von ξ_1 , die bereits als Pivot verwandt worden sind,

 P_1 aus den Teilen der Blockzeile und Blockspalte, die im folgenden als Pivot gewählt werden (Pivot-Zeile) und H_1 aus dem Rest.

Mit diesen Bezeichnungen ist die Behauptung (3.2.38) äquivalent mit:

(3.2.40)
$$\|B_{k} + P_{k}\|^{2} \le K_{r}^{(v)} \|A\|^{2} \varepsilon^{4}$$
.

Nun kann der Induktionsbeweis wie folgt geführt werden.

Induktions-Anfang: Für r = 0 ist $\kappa_{k+1} = (r+1, r+2) = (1,2) = \kappa_1$, also k = 0 und

$$\sum_{i=1}^{r} \sum_{j=i+1}^{p} S_{ij}^{(k)} = 0 ,$$

da es sich um die leere Summe handelt, so daß für r=0 die Behauptung (3.2.38) mit $K_r^{(v)}:=0$ erfüllt ist.

<u>Induktions-Voraussetzung:</u> Sei nun die Konstante $K_{r-1}^{(v)}$, für die (3.2.38) erfüllt ist, bereits berechnet, d.h. es gelte

$$||B_{k-p+r}||^2 \le K_{r-1}^{(v)} ||A||^2 \varepsilon^4$$

<u>Zu zeigen ist:</u> Es existiert eine Konstante $K_r^{(v)}(\|A\|, \delta, \delta^-, p, K_{r-1}^{(v)})$, die nur von A und $K_{r-1}^{(v)}$ abhängt, so daß

gilt.

Dazu stellen wir zunächst fest, daß

$$\|H_1\|^2 \le \|E_1\|^2 = S^2(A_1) \le \|A\|^2 \varepsilon^2$$

für $0 \le 1 \le P$ ist.

Weiterhin ist der maximale Zuwachs der Norm der Blöcke A_{ij} von B abzuschätzen; diese bleiben entweder unberührt, wenn $\{i,j\} \cap \{r,s\} = \emptyset$ ist, oder der Zuwachs kann nach Lemma 3.2.8 abgeschätzt werden. Da das Pivot-Paar (r,s) die Folge

$$(r,r+1),(r,r+2),\ldots,(r,p)$$

durchläuft, ergibt sich aus Lemma 3.2.8 und der Induktionsvoraussetzung (3.2.41):

$$||B_{k}||^{2} = \sum_{i=1}^{r-1} \sum_{j=i+1}^{p} S_{ij}^{(k)}$$

$$\leq \sum_{i=1}^{r-1} \sum_{j=i+1}^{p} a_{v}^{p-r} S_{ij}^{(k-p+r)}$$

$$= a_{v}^{p-r} ||B_{k-p+r}||^{2}$$

$$\leq a_{v}^{p-r} K_{r-1}^{(v)} ||A||^{2} \epsilon^{4} .$$

Es bleibt $\|P_k\|^2$ abzuschätzen. Dazu verfolgen wir zunächst, wie sich ein als Pivot benutzter Block $A_{rj}^{(k+j-p)}$ verändert, wenn anschließend die weiteren Blöcke dieser r-ten Zeile nacheinander als Pivot verwendet werden. Diese Änderung kann nach Lemma 3.2.9 folgendermaßen abgeschätzt werden:

$$\|A_{rj}^{(k)}\| \le L^{p-j} \|A_{rj}^{(k+j-p)}\| + \sum_{i=j+1}^{p} L^{p-i} m_{v}^{(k+i-p-1)} \|A_{ij}^{(k+i-p-1)}\| .$$

Durch Quadrieren erhält man unter Anwendung der Ungleichung von Cauchy-Schwarz:

$$\|A_{rj}^{(k)}\|^{2} \leq 2 \left[L^{2(p-j)} \|A_{rj}^{(k+j-p)}\|^{2} + \left(\sum_{i=j+1}^{p} L^{p-i} \|A_{ij}^{(k+i-p-1)}\| A_{ij}^{(k+i-p-1)} \| \right)^{2} \right]$$

$$\leq 2 L^{2(p-j)} \left[\|A_{rj}^{(k+j-p)}\|^{2} + \left(m_{v}^{(k+i-p-1)} \right)^{2} \cdot \sum_{i=j+1}^{p} \|A_{ij}^{(k+i-p-1)}\|^{2} \right] ,$$

wobei benutzt wird, daß wegen $p \ge i > j$ und L > 1 gilt:

$$L^{2(p-i)} \le L^{2(p-j)}$$
.

Analog ergibt sich aus Lemma 3.2.9:

$$(3.2.46) \|A_{jr}^{(k)}\|^{2} \le 2L^{2(p-j)} \left[\|A_{jr}^{(k+j-p)}\|^{2} + \sum_{i=j+1}^{p} (m_{v}^{(k+i-p-1)})^{2} \cdot \sum_{i=j+1}^{p} \|A_{ji}^{(k+i-p-1)}\|^{2} \right],$$

und man erhält durch Addition von (3.2.45) und (3.2.46) unter Beachtung von (3.2.36), sowie Summation über die Pivots der r-ten Zeile, wobei man wegen L > 1 und p \geq j > r die Terme L^{2(p-j)} nach oben durch L^{2(p-r)} abschätzen und vor die Summe ziehen kann:

$$\sum_{j=r+1}^{p} S_{rj}^{(k)} \le 2L^{2(p-r)} \left[\sum_{j=r+1}^{p} S_{rj}^{(k+j-p)} + \frac{\sum_{j=r+1}^{p} \left(\sum_{i=j+1}^{p} (m_{v}^{(k+i-p-1)})^{2} \sum_{i=j+1}^{p} S_{ij}^{(k+i-p-1)} \right) \right]$$

$$\le 2L^{2(p-r)} \left[\sum_{j=r+1}^{p} S_{rj}^{(k+j-p)} + \frac{\sum_{j=r+1}^{p} (m_{v}^{(k+j-p)})^{2}}{\sum_{j=r+1}^{p} S_{rj}^{(k+j-p)}} + \frac{\sum_{j=r+1}^{p} (m_{v}^{(k+j-p)})^{2}}{\sum_{j=r+1}^{p} S_{rj}^{(k+j-p)}}}$$

$$+ \sum_{i=r+2}^{p} (m_{v}^{(k+i-p-1)})^{2} \sum_{i=r+2}^{p} \sum_{j=r+1}^{i-1} S_{ij}^{(k+i-p-1)}$$

Der Term vor der Doppelsumme kann nach Lemma 3.2.9 weiter abgeschätzt werden durch:

$$\sum_{i=r+2}^{p} (m_{v}^{(k+i-p-1)})^{2} \leq (p-r-1) \left(0.5835 \frac{\widetilde{K}_{v}}{\|A\|} + \frac{16b_{rs}^{(v)}}{5 \delta^{-}}\right)^{2} \|A\|^{2} \epsilon^{2}.$$

Für die erste Summe erhält man nach Lemma 3.2.7 unter Beachtung von (3.2.36) und (3.2.37) die Abschätzung:

$$S_{rj}^{(k+j-p)} = \|A_{rj}^{(k+j-p)}\|^{2} + \|A_{jr}^{(k+j-p)}\|^{2}$$

$$\leq \frac{32}{49(\delta^{-})^{2}} (f_{v} + g_{v})^{2} \|A\|^{4} \epsilon^{4} ,$$

und für die zweite Summe gilt, da die dort vorkommenden Blöcke in H liegen (s.(3.2.43)):

$$\sum_{j=r+1}^{i-1} S_{ij}^{(k+i-p-1)} \le S^{2}(A_{k+i-p-1}) \le ||A||^{2} \epsilon^{2}.$$

Zusammen folgt damit:

$$\|P_{k}\|^{2} = \sum_{j=r+1}^{p} S_{rj}^{(k)}$$

$$\leq 2 L^{2(p-r)} \left[(p-r) \frac{32}{49(\delta^{-})^{2}} (f_{v} + g_{v})^{2} \|A\|^{4} \epsilon^{4} + (p-r-1) \|A\|^{2} \epsilon^{2} (p-r-1) \left(0.5835 \frac{\tilde{K}_{v}}{\|A\|} + \frac{16b_{rs}^{(v)}}{58}\right)^{2} \|A\|^{2} \epsilon^{2} \right]$$

$$=: Q_{r}^{(v)} \|A\|^{2} \epsilon^{4} .$$

Durch Addition der Abschätzungen für $\|B_k\|^2$ (s.(3.2.44)) und $\|P_k\|^2$ ergibt sich:

$$\begin{split} \| \, B_k \, + \, P_k \|^{\, 2} \, & \leq \, 2 \, (a_{\nu}^{p-r} \, K_{r-1}^{(\nu)} \, + \, Q_r^{(\nu)}) \, \| \, A \|^{\, 2} \, \, \epsilon^{\, 4} \\ & = : \, K_r^{\, (\nu)} \, \| \, A \|^{\, 2} \, \, \epsilon^{\, 4} \quad , \end{split}$$

und $K_r^{(v)}$ hängt ebenso wie - nach Induktionsvoraussetzung - $K_{r-1}^{(v)}$ nur von $\|A\|$, δ , δ und p ab. Damit ist die Zwischenbehauptung (3.2.38) bewiesen.

Fassen wir noch einmal zusammen: es ist gezeigt, daß für alle r=0(1)p-1 eine von ϵ unabhängige Konstante $K_r^{(\nu)}$ existiert, die (3.2.38) erfüllt. Damit gilt nach Durchführung des ersten Teils eines Zyklus, d.h. $\kappa_{k+1}=(p,p+1)$, k=P:

$$\|E_{p}\|^{2} = \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} S_{ij}^{(k)} \le K_{p-1}^{(v)} \|A\|^{2} \varepsilon^{4}$$
.

Nach Korollar 3.2.11 ist unter Beachtung von (3.2.37):

$$\varepsilon^2 \leq \max \left\{ \frac{1}{16} H_{v}^{P}, \frac{1}{4} (\sqrt{H_{v}})^{P} W_{v} \right\} \varepsilon_{N}^2$$

und damit

$$S(A_p^N) = \|E_p^N\| \le K^{(v)}(\|A\|, \delta, \delta^-, p) \|A\| \epsilon_N^2$$

mit

$$K^{(\vee)} = K^{(\vee)} \left(\| A \| , \delta , \delta^-, p \right) := \sqrt{K_{P-1}^{(\vee)}} \max \left\{ \frac{1}{16} H_{V}^{P}, \frac{1}{4} \left(\sqrt{H_{V}} \right)^{P} W_{V} \right\}$$

und diese Konstante $K^{(v)}$ hängt nur von der Matrix A ab. Damit ist der zweite Hauptsatz bewiesen.

Bemerkung 3.2.13

Die Konstanten $Q_r^{(v)}$, r=1(1)p-1 aus Satz 3.2.12 sind nach oben beschränkt durch $Q_1^{(v)}$. Da die Konstanten

$$K_{O}^{(v)} = 0$$
 , $K_{r}^{(v)} = 2(a_{v}^{p-r} K_{r-1}^{(v)} + Q_{r}^{(v)})$, $r = 1(1)p-1$

endlich rekursiv definiert sind, können sie durch eine endliche Summe ausgedrückt werden, und daraus kann man mit obigem und der Formel für die geometrische Summe eine obere Abschätzung von $K^{(v)}$ gewinnen.

Aus Satz 3.2.12 ergibt sich unmittelbar das

Korollar 3.2.14.

Sei A eine reelle Blockmatrix mit $\delta > 0$, die (3.2.30) erfüllt. Weiter gelte

$$\Delta(A^{N}) \leq \frac{\delta^{-}}{16} ,$$

$$S(A^N) \le \frac{1}{4} \|A\| \epsilon_N$$
 , $\|C(D^N)\| \le \|A\|^2 \epsilon_N$,

mit

$$\varepsilon_{N} < \min \left\{ \left(\sqrt{H_{\nu}} \right)^{-P} \tilde{\varepsilon}, \frac{\tilde{\epsilon}}{P}, \frac{1}{4K^{(\nu)}} \right\} * \right\}.$$

Dann gilt

$$S(A_{P}^{N}) \leq \frac{1}{4} ||A|| \epsilon_{N}$$
,

d.h. S(A) ist nach dem ersten Teil eines Zyklus nicht größer als vorher.

*)
$$\tilde{\epsilon}$$
 aus (3.2.30).

3.2.4 Normreduzierung der Blockdiagonalen

Zur Normreduzierung der Blockdiagonalen im zweiten Teil eines Zyklus werden Eberlein-Jacobi-Matrizen verwendet (vergl. § 2.4 und (3.2.2)). Für den Beweis des dritten Hauptsatzes, daß nach dem zweiten Teil eines Zyklus auch ||C([])|| quadratisch klein ist, werden einige Aussagen über die Eberlein-Jacobi-Transformationen benötigt, die in den folgenden Lemmata bereitgestellt werden. Zunächst führen wir noch eine Bezeichnung ein:

(3.2.48)
$$C(A_{rr}^{(k)}) = \begin{pmatrix} \tilde{c}_{11} & \tilde{c}_{12} \\ & & \\ \tilde{c}_{12} & \tilde{c}_{22} \end{pmatrix}$$
 , $k = P + r - 1$, $r = 1(1)p$,

und bemerken, daß im zweiten Teil eines Zyklus $A_{rr}^{(k')} = A_{rr}^{(k'+1)}$ gilt, für $k' \neq P+r-1$; damit gilt für jeden Pivotblock, bevorer transformiert wird, die Abschätzung:

$$(3.2.49)$$
 $\|C(A_{rr}^{(k)})\| \le \|C(D_p)\|$, $k = P + r - 1$, $r = 1(1)p$.

Des weiteren gilt:

$$\widetilde{C}_{11} = -\widetilde{C}_{22} = (a_{1m}^{(k)} + a_{m1}^{(k)})(a_{m1}^{(k)} - a_{1m}^{(k)}) ,$$

$$(3.2.50) \quad \widetilde{C}_{12} = (a_{mm}^{(k)} - a_{11}^{(k)})(a_{m1}^{(k)} - a_{1m}^{(k)}) ,$$

$$1 = 2 r - 1 , m = 2r , k = P + r - 1 .$$

In diesem Paragraphen ist stets der hier notierte Zusammenhang von 1, m, k und r zu sehen; wir lassen deshalb bei den beiden folgenden Lemmata den Index k aus Gründen der Übersichtlichkeit weg.

<u>Lemma 3.2.15</u>

Sei A eine reelle Blockmatrix mit $\delta > 0$, die die Bedingungen

(3.2.51)
$$S(A) \leq \frac{\delta^{-}}{16}$$
 , $\Delta(A) \leq \frac{\delta^{-}}{16}$

erfüllt. Dann gilt nach einer Jacobi-Rotation $A' = 0^T A 0$ (s.(2.4.5)) mit (2.4.1),(2.4.2) und Pivotpaar (1,m):

$$(3.2.52)$$
 $S(A') = S(A)$,

$$(3.2.53)$$
 $|\tilde{c}_{11}^{\prime}| \leq S^{2}(A)$,

$$|a_{11} - a_{mm}| \leq \frac{5 \|C(A_{rr})\|}{4 \delta^{-}}.$$

Beweis: Für eine Jacobi-Rotation gilt:

$$a_{i1}^{'2} + a_{im}^{'2} = a_{i1}^{2} + a_{im}^{2}$$
,
 $a_{ii}^{'2} + a_{mi}^{'2} = a_{ii}^{2} + a_{mi}^{2}$, $i \notin \{1,m\}$,

und damit $S^2(A') = S^2(A)$, womit (3.2.52) gezeigt ist.

Mit
$$C(A'_{rr}) = \begin{pmatrix} \widetilde{c}'_{11} & \widetilde{c}'_{12} \\ \widetilde{c}'_{12} & \widetilde{c}'_{22} \end{pmatrix}$$
 gilt:
 $|c'_{11} - \widetilde{c}'_{11}| \le \sum_{i \neq 1, m}^{n} (a'_{i1} - a'_{1i}) \le S^{2}(A') = S^{2}(A)$,

und analog

$$|c_{mm}^{\dagger} - \widetilde{c}_{22}^{\dagger}| \leq S^2(A)$$
.

Daraus folgt mit (3.2.50) und $c_{11}' = c_{mm}'$ (s.(2.4.7)):

$$2|\widetilde{c}_{11}'| = |\widetilde{c}_{11}' - \widetilde{c}_{22}'| \le |\widetilde{c}_{11}' - c_{11}'| + |c_{mm}' - \widetilde{c}_{22}'| + |c_{11}' - c_{mm}'|,$$

$$\le 2S^{2}(A),$$

und das ist (3.2.53).

Um die Abschätzung (3.2.54) zu erhalten, betrachtet man die Änderung der Elemente von A_{rr} unter einer Jacobi-Rotation; dazu setzen wir abkürzend $c=\cos\phi_r$, $s=\sin\phi_r$ und definieren:

$$e_1 := a_{1m} + a_{m1}$$
 , $e_2 := a_{11} - a_{mm}$.

Aus (3.2.50) folgt mit (3.2.48) und $|a_{1m} - a_{m1}| \ge \frac{4}{5} \delta^-$ wegen Lemma 2.2.2:

$$|e_{j}| \le \frac{5 \|C(A_{rr})\|}{4 \sqrt{2} \delta^{-}}$$
, $j \in \{1,2\}$.

Weiter gilt:

$$a'_{11} = c^2 a_{11} + cs(a_{m1} + a_{1m}) + s^2 a_{mm}$$
,
 $a'_{mm} = s^2 a_{11} - cs(a_{1m} + a_{m1}) + c^2 a_{mm}$,
 $a'_{11} - a'_{mm} = (a_{11} - a_{mm})(c^2 - s^2) + 2(a_{1m} + a_{m1})cs$
 $= e_2 \cos 2\phi_r + e_1 \sin 2\phi_r$.

Daraus folgt für $(a_{11}' - a_{mm}')^2$ unter Anwendung der Cauchy-Schwarzschen Ungleichung:

$$(a_{11}' - a_{mm}')^2 \le 2 \left(\frac{5 \| C(A_{rr})\|}{4 \sqrt{2} \delta^{-1}}\right)^2$$

und daraus folgt (3.2.54). Damit ist Lemma 3.2.15 bewiesen.

Lemma 3.2.16

Sei A eine reelle Blockmatrix mit $\delta > 0$, die (3.2.51) und

$$(3.2.55)$$
 $\|C(A_{rr})\| \leq \frac{(\delta^{-})^2}{64}$

erfüllt. Dann gilt nach einer hyperbolischen Transformation (2.4.6) mit (2.4.4) und Pivotpaar (1,m):

$$(3.2.56)$$
 $S(A'') \le 1.031 S(A)$.

Beweis: Es gilt mit $ab \le \frac{1}{2}(a^2 + b^2)$:

$$c_{1m}' = \widetilde{c}_{12}' + \Sigma$$
,

$$|\Sigma| \le \sum_{i=1,m}^{n} (|a_{i1}' a_{im}'| + |a_{1i}' a_{mi}'|) \le \frac{1}{2} S^{2}(A)$$
.

Weiter gilt nach (2.4.4) wegen $G \ge 0$ und $D^2 \ge 0$ mit (3.2.50) und $|a'_{lm} - a'_{ml}| \ge \frac{4}{5} \delta^-$:

$$| \tanh \psi | \leq \left| \frac{(a'_{mm} - a'_{11})}{2(a'_{1m} - a'_{m1})} + \frac{\Sigma}{2(a'_{1m} - a'_{m1})^2} \right|$$

$$\leq \frac{25 \| C(A_{rr}) \|}{32 (\delta^-)^2} + \frac{25 S^2(A)}{64 (\delta^-)^2}$$

$$\leq 0.014 ,$$

nach den Voraussetzungen (3.2.55) und (3.2.51). Daraus ergibt sich:

$$(3.2.58) |sh\psi| = \frac{|th\psi|}{\sqrt{1-th^{2}_{\psi}}} \le 1.008 |th\psi| \le 0.0142.$$

Für ch $\psi = \sqrt{1-\text{th}^2\psi}$ erhält man durch Entwicklung in eine Taylor-reihe:

(3.2.59)
$$\text{ch } \psi = 1 + \eta$$
, mit
$$|\eta| \le \frac{\text{th}^2 \psi}{2 (1 - \text{th}^2 \psi)} \le 0.508 \text{ th}^2 \psi$$
 ≤ 0.0001 .

Setzt man A' = D' + E' (s.(1.1.22)), so ist, da S und S^{-1} block-diagonal sind (vergl.(2.4.6)):

$$A'' = D'' + E''$$
, $E'' = S^{-1} E' S$,

und damit

(3.2.60)
$$S(A'') = ||E''|| \le ||S^{-1}||_{Sp} ||S||_{Sp} S(A')$$
.

Da S und S^{-1} symmetrisch sind, ist $\|S\|_{Sp} = spr S$; S und S^{-1} haben dasselbe charakteristische Polynom

$$\det(S - \lambda I) = (\lambda^2 - 2\lambda \cosh \psi + 1)(\lambda - 1)^{n-2}$$

mit den Wurzeln

$$\lambda_{1,2} = ch \psi + sh \psi$$
, $\lambda_{i} = 1$, $i = 2(1)n$.

Daraus folgt wegen $|\lambda_{1,2}| \le 1.015$ nach (3.2.58), (3.2.59): $\|S\|_{Sp} = \|S^{-1}\|_{Sp} \le 1.015$ und mit (3.2.60) die Behauptung (3.2.56). Damit ist Lemma 3.2.16 bewiesen.

1.

Aus diesem Lemma ergibt sich zusammen mit Korollar 3.2.14, Lemma 3.2.10 und Satz 3.2.12 das folgende Korollar 3.2.17, welches besagt, daß die im ersten Teil eines Zyklus erreichte Ordnung von S(A) im zweiten Teil erhalten bleibt.

Korollar 3.2.17

Sei A eine reelle Blockmatrix mit δ > 0, die (3.2.30) erfüllt. Weiter gelte

$$\begin{split} &\Delta\left(A^{N}\right) \leq \frac{\delta^{-}}{16} \ , \\ &S\left(A^{N}\right) \leq \frac{1}{4}\|A\|\epsilon_{_{N}} \ , \quad \|C\left(D_{_{N}}\right)\| \leq \|A\|^{2}\epsilon_{_{N}} \ , \end{split}$$

mit

$$\varepsilon_{N} < \min\left\{ (\sqrt{H_{\nu}})^{-P} \tilde{\varepsilon}, \frac{\tilde{\varepsilon}}{P}, \frac{1}{4\kappa^{(\nu)} + 0.51^{P}} \right\}$$
 *)

Dann gilt

$$(3.2.61) S(A_{P+r}^{N}) \leq \frac{1}{4} \|A\| \tilde{\epsilon} ,$$

*)
$$\tilde{\epsilon}$$
 aus (3.2.30).

$$(3.2.62)$$
 $S(A_{p+r}^{N}) \le 1.031^{p} S(A_{p}^{N})$, $r = O(1)p$,

(3.2.63)
$$S(A^{N+1}) \le 1.031^p K^{(v)} |A| \varepsilon_N^2 < \frac{1}{4} |A| \varepsilon_N$$

Beweis: Nach Lemma 3.2.10 ist $\|C(D_p)\| \le (\delta^-)^2/64$ und damit (s.(3.2.49)) $\|C(A_{rr})\| \le (\delta^-)^2/64$. Für p=1 sind die Aussagen trivial; für $p \ge 2$ folgt (3.2.61) mit Korollar 3.2.14 und Lemma 3.2.16 induktiv über r, da wegen $H_{\mathcal{V}}^{p(p-1)/2} \ge (\sqrt{3})^p \ge 1.031^p$ die Voraussetzungen von Lemma 3.2.16 für A_{p+r-1} jeweils erfüllt sind. Damit folgen (3.2.62) und (3.2.63) unmittelbar aus Lemma 3.2.16 und Satz 3.2.12.

Damit ist S(A) nach einem kompletten Zyklus abgeschätzt, und es bleibt zu zeigen, daß durch die zweite (hyperbolische) Transformation im zweiten Teil eines Zyklus für \tilde{c}_{11} und \tilde{c}_{22} die nach der ersten (orthogonalen) Transformation erreichte Ordnung erhalten bleibt und gleichzeitig \tilde{c}_{12} quadratisch klein wird.

Lemma 3.2.18

Unter den Voraussetzungen von Korollar 3.2.17 gilt nach einer hyperbolischen Transformation (2.4.6) mit (2.4.4) und Pivotpaar (1,m):

$$|\widetilde{c}_{11}^{"}| = |\widetilde{c}_{22}^{"}| \le 1.77 \cdot 1.063^{p} \frac{\|A\|}{\delta^{-}} S^{2}(A_{p}),$$

$$|\widetilde{c}_{12}''| \leq \frac{0.115 \|A\|^2}{(\delta^-)^4} \|C(D_p)\|^2 +$$

$$+ (0.8 \cdot 10^{-4} \frac{\|A\|}{\delta^-} + 0.91) \frac{\|A\|}{\delta^-} 1.063^p S^2(A_p) .$$

Beweis: Nach einer hyperbolischen Transformation $A_{k+1}^N = A'' = S^{-1} A' S$, mit $A' = 0^T A_k^N O$ (s.(2.4.3) bis (2.4.6)) und Pivotpaar (1,m) gilt mit ch = ch ψ_r , sh = sh ψ_r und th = th ψ_r :

$$a_{11}^{"} = a_{11}^{"} ch^{2} + (a_{1m}^{"} - a_{m1}^{"}) ch sh - a_{mm}^{"} sh^{2} ,$$

$$a_{mm}^{"} = a_{mm}^{"} ch^{2} + (a_{m1}^{"} - a_{1m}^{"}) ch sh - a_{11}^{"} sh^{2} ,$$

$$a_{1m}^{"} = a_{1m}^{"} ch^{2} + (a_{11}^{"} - a_{mm}^{"}) ch sh - a_{m1}^{"} sh^{2} ,$$

$$a_{m1}^{"} = a_{m1}^{"} ch^{2} + (a_{mm}^{"} - a_{11}^{"}) ch sh - a_{1m}^{"} sh^{2} .$$

Daraus ergibt sich wegen (3.2.58) mit ch = 1+ η nach (3.2.59):

$$\tilde{c}_{12}^{"} = (a_{mm}^{"} - a_{11}^{"})(1+2 \text{ sh}^{2}) + 2 \text{ ch sh}(a_{m1}^{"} - a_{1m}^{"})(a_{m1}^{"} - a_{1m}^{"})$$

$$= T_{1}^{"} T_{2}^{"},$$

mit

$$|T_{2}| = |a''_{m1} - a''_{1m}| \le \sqrt{2} |A''| \le \sqrt{2} |A''|,$$

$$|T_{1}| = \left| (a'_{mm} - a'_{11}) + 2 \frac{th}{\sqrt{1-th^{2}}} (a'_{m1} - a'_{1m}) + 2 \left((a'_{mm} - a'_{11}) sh^{2} + (a'_{m1} - a'_{1m}) \eta sh \right) \right|.$$

Wendet man auf $T_1(2.2.4)$, (3.2.54), (3.2.57), (3.2.58), (3.2.59), sowie $(|x| < 1 \Rightarrow \frac{1}{\sqrt{1-|x|}} - 1 \le \frac{|x|}{2(1-|x|)})$ und $|a'_{lm} - a'_{ml}| \ge \frac{4}{5} \delta^-$ an, so erhält man mit G, D, E nach (2.4.4) und Σ nach (3.2.57):

$$|T_{1}| \leq \left| (a_{mm}^{\dagger} - a_{11}^{\dagger}) + \frac{a_{mm}^{\dagger} - a_{11}^{\dagger}}{a_{1m}^{\dagger} - a_{m1}^{\dagger}} \frac{a_{m1}^{\dagger} - a_{1m}^{\dagger}}{\sqrt{1 - th^{2}}} \right| + \left| \frac{\sum (a_{m1}^{\dagger} - a_{1m}^{\dagger})}{E^{2} \sqrt{1 - th^{2}}} \right| + 2 \frac{(G + 2D^{2}) |D|}{4 E^{2} \sqrt{1 - th^{2}}} + 2 \left(|a_{mm}^{\dagger} - a_{11}^{\dagger}| \cdot 1.008^{2} th^{2} + |a_{m1}^{\dagger} - a_{1m}^{\dagger}| \cdot 0.508 th^{2} \cdot 0.0142 \right)$$

$$\leq |a_{mm}^{\dagger} - a_{11}^{\dagger}| \left| 1 - \frac{1}{\sqrt{1 - th^{2}}} \right| + 1.0001 \left(\frac{|\Sigma|}{|a_{1m}^{\dagger} - a_{m1}^{\dagger}|} + \frac{(G + 2D^{2}) |D|}{2 E^{2}} \right) + 2 \left(1.017 |a_{mm}^{\dagger} - a_{11}^{\dagger}| + 0.0072 |a_{m1}^{\dagger} - a_{1m}^{\dagger}| \right) th^{2}$$

$$\begin{split} |T_{1}| &\leq 1.0001 \; \left(\frac{5}{8 \, \delta^{-}} \; S^{2}(A') \; + \; \frac{125}{8192 \, \delta^{-}} \; S^{2}(A') \; + \; \frac{3125}{65 \, 536 \, (\delta^{-})^{3}} \; \| \, C(D_{p}) \|^{2} \right) \\ &+ \left(\frac{5 \, \delta^{-}}{4 \cdot 64} \cdot 0.508 + 0.04\delta^{-} \; + \; 0.021 \; \| \, A \| \, \right) \; \mathrm{th}^{2} \\ &\leq \left(0.626 + 0.016 \right) \; \frac{S^{2}(A_{k})}{\delta^{-}} \; + \; 0.024 \; \frac{\| \, A \|}{(\delta^{-})^{4}} \; \| \, C(D_{p}) \|^{2} \\ &+ \; 0.046 \; \| \, A \| \; \frac{25^{2}}{32^{2} \, (\delta^{-})^{4}} \; \left(\| \, C(D_{p}) \| \; + \; \frac{1}{2} \; S^{2}(A_{k}) \; \right)^{2} \\ &\leq 0.081 \; \frac{\| \, A \|}{(\delta^{-})^{4}} \, \| \, C(D_{p}) \|^{2} \; + \; \left(0.56 \cdot 10^{-4} \; \frac{\| \, A \|}{\delta^{-}} \; + \; 0.642 \; \right) \; \frac{S^{2}(A_{k})}{\delta^{-}} \; , \end{split}$$

wobei bei den letzten Ungleichungen $\delta^- \le \frac{1}{2} \|A\|$ für dim $A \ge 4$, (3.2.57), (3.2.51) und die Ungleichung von Cauchy-Schwarz eingehen. Damit gilt nach Korollar 3.2.17:

$$|\tilde{c}_{12}''| \le 0.115 \frac{\|\tilde{A}\|^2}{(\delta^-)^4} \|C(\tilde{D}_p)\|^2 +$$

$$+ \left(0.8 \cdot 10^{-4} \frac{\|\tilde{A}\|}{\delta^-} + 0.91\right) \frac{\|\tilde{A}\|}{\delta^-} 1.063^p S^2(\tilde{A}_p),$$

und das ist (3.2.65). Weiter gilt nach (3.2.66):

$$|\widetilde{c}_{11}''| = |\widetilde{c}_{22}''| \le (|ch^2 - sh^2| |a_{1m}' + a_{m1}'|) \sqrt{2} |A|$$

$$= \sqrt{2} |A| |a_{1m}' + a_{m1}'|.$$

Nach (3.2.50) und $|a'_{lm} - a'_{ml}| \ge \frac{4}{5} \delta^-$ (s.Lemma 2.2.2), sowie Korollar 3.2.17 gilt:

$$S^{2}(A_{k}) \geq |\tilde{c}'_{11}| \geq |a'_{1m} + a'_{m1}| \frac{4}{5} \delta^{-}$$
,
 $|a'_{1m} + a'_{m1}| \leq \frac{5}{4} \cdot 1.063^{p} \frac{S^{2}(A_{p})}{s^{-}}$,

und damit

$$|\widetilde{c}_{11}^{"}| = |\widetilde{c}_{22}^{"}| \le 1.77 \cdot 1.063^{p} \frac{\|A\|}{\delta} S^{2}(A_{p})$$

Damit ist Lemma 3.2.18 bewiesen.

Nach diesen Vorbereitungen können wir den dritten Hauptsatz, daß nach dem zweiten Teil eines Zyklus auch der Kommutator der Block-diagonalen quadratisch klein ist, beweisen.

Satz 3.2.19 (Dritter Hauptsatz)

Unter den Voraussetzungen von Korollar 3.2.17 gilt:

mit

$$C^{(v)} := p \left[\left(0.24 + 0.71 \cdot 10^{-5} \frac{\|A\|}{\delta^{-}} \right) \cdot 1.063^{p} K^{(v)} + 0.17w_{v}^{2} \frac{\|A\|^{4}}{(\delta^{-})^{4}} \right].$$

Beweis: Aus Lemma 3.2.18 folgt mit $\sqrt{a^2 + b^2} \le |a| + |b|$ und (3.2.48):

$$(3.2.68) \quad \|C(A_{rr}^{(k+1)}\| = \sqrt{2(\tilde{c}_{11}^{"2} + \tilde{c}_{12}^{"2})}$$

$$\leq \sqrt{2} \left[\left(1.77 + 0.91 + 0.8 \cdot 10^{-4} \frac{\|A\|}{\delta^{-}} \right) \frac{\|A\|}{\delta^{-}} 1.063^{p} S^{2}(A_{p}^{N}) + 0.115 \frac{\|A\|^{2}}{(\delta^{-})^{4}} \|C(D_{p}^{N})\|^{2} \right], \quad r = 1(1)p .$$

Nach Lemma 3.2.10 ist

$$S(A_P^N) \leq \frac{\delta^{-1}}{16}$$
,

und nach Satz 3.2.12 gilt:

$$S(A_{P}^{N}) \leq K^{(v)} \|A\| \epsilon_{N}^{2}$$
,

also

$$S^{2}(A_{P}^{N}) \leq \frac{\delta^{-}}{16} K^{(v)} \|A\| \epsilon_{N}^{2}$$
.

Für $\|C(D_p^N)\|$ folgt aus Korollar 3.2.11:

$$\|C(\mathbb{D}_{P}^{N})\|^{2} \leq w_{N}^{2} \|A\|^{4} \varepsilon_{N}^{2}$$
,

und mit (3.2.68) gilt:

$$\| (A_{rr}^{(P+p)}) \| \leq \sqrt{2} \left[\left(2.68 + 0.8 \cdot 10^{-4} \frac{\|A\|}{\delta^{-}} \right) \frac{1}{16} \cdot 1.063^{p} K^{(v)} + 0.115 w_{v}^{2} \frac{\|A\|^{4}}{(\delta^{-})^{4}} \right] \|A\|^{2} \varepsilon_{N}^{2}$$

$$\leq \frac{C^{(v)}}{p} \|A\|^{2} \varepsilon_{N}^{2} , \quad r = 1(1)p .$$

Damit ist

$$\|C(D^{N+1})\| \le \sum_{r=1}^{p} \|C(A_{rr}^{(p+p)})\|$$
 $\le C^{(v)} \|A\|^2 \varepsilon_N^2$,

und der Satz 3.2.19 ist bewiesen.

Aus Satz 3.2.19 ergibt sich unmittelbar das

Korollar 3.2.20.

Gilt zusätzlich zu den Voraussetzungen von Satz 3.2.19:

$$(3.2.69) \qquad \varepsilon_{N} < \frac{1}{C^{(v)}} ,$$

so ist

$$(3.2.70) \qquad ||C(D^{N+1})|| < ||A||^2 \epsilon_{N} .$$

Aus Satz 3.2.19 mit Korollar 3.2.20 ergibt sich zusammen mit Korollar 3.2.17 der vierte Hauptsatz, der die Ergebnisse dieses Paragraphen zusammenfaßt.

Satz 3.2.21 (vierter Hauptsatz)

Sei A eine reelle Blockmatrix mit δ > 0, und es gelte

$$\widetilde{\varepsilon} := \min \left\{ (\sqrt{H_{v}})^{-P} \ \widetilde{\varepsilon} \ , \frac{\widetilde{\varepsilon}}{P} \ , \frac{1}{4K^{(v)}1.031^{P}} \ , \frac{1}{C^{(v)}} \right\} ,$$

$$(3.2.71) S(A) \le \frac{1}{4} \|A\| \widetilde{\varepsilon} \ ,$$

$$\|C(D)\| \le \|A\|^{2} \ \widetilde{\varepsilon} \ ,$$

$$\Delta(A) \le \frac{\delta^{-}}{16} \ .$$

Dann kann das durch (3.2.1) und (3.2.2) definierte Verfahren beliebig fortgesetzt werden, und die Matrixfolge (A^N) konvergiert quadratisch gegen die Murnaghan-Form von A.

Beweis: Gilt für ein $N \ge 0$, $\varepsilon_N \le \frac{\approx}{\epsilon}$:

$$S\left(A^{N}\right) \ \leq \ \frac{1}{4} \|A\| \, \epsilon_{_{N}} \ , \ \|C\left(\textstyle{\textstyle{\textstyle \bigcap}}^{N}\right) \| \ \leq \ \|A\|^{2} \ \epsilon_{_{N}} \quad ,$$

so erhält man aus Korollar 3.2.17 und Satz 3.2.19 mit

$$\varepsilon_{N+1} := \max \left\{ 4 \cdot 1.031^{p} K^{(v)}, C^{(v)} \right\} \varepsilon_{N}^{2} :$$

$$(i)$$
 $\epsilon_{N+1} < \epsilon_N$

(ii)
$$S(A^{N+1}) \le \frac{1}{4} ||A|| \varepsilon_{N+1}$$
,

(iii)
$$\|C(D^{N+1})\| \le \|A\|^2 \varepsilon_{N+1}$$
.

Damit sind die Bedingungen (3.2.71) für alle A^N erfüllt, da $\Delta(A^N)$ monoton fallend ist. Außerdem ist mit der Nullfolge (ϵ_N) eine quadratisch konvergente Majorante von $4 \text{ S}(A^N) / \|A\|$ und $\|C(D^N)\| / \|A\|^2$ gefunden, so daß für das durch (3.2.1) und (3.2.2) definierte Verfahren die quadratische Konvergenz gegen Blockdiagonalgestalt und Normalität nachgewiesen ist. Da alle verwendeten Transformationsmatrizen, bis auf die Jacobi-Rotation bei der Eberlein-Jacobi-Matrix gegen die Einheitsmatrix konvergieren, und die Form

$$\begin{pmatrix} a & b \\ -b & a \end{pmatrix}$$

invariant unter Jacobi-Rotationen ist, folgt, analog zu dem Beweis von Satz 3.1.1, die Konvergenz der Matrix-Folge (A^N) gegen die Murnaghan-Form von A. Damit ist der Satz 3.2.21 bewiesen.

Für das am Anfang von § 3.2 definierte Jacobi-ähnliche Verfahren ist damit die asymptotisch quadratische Konvergenz bewiesen. Wählt man in (3.2.1) den normreduzierenden Parameter X_{ν} , $\nu \in \{1,2\}$, je nachdem welcher die Euklidische Norm der Matrix A besser reduziert *), so hat man für das zyklische Verfahren zwar keinen globalen Konvergenzbeweis, kann aber nach den praktischen Erfahrungen mit Jacobi-

ähnlichen Verfahren globale Konvergenz erwarten *). Für fast blockdiagonales A kann man nicht garantieren, daß stets X_2 gewählt wird, obwohl das in der experimentellen Praxis immer beobachtet wurde. Es existiert aber nach Bemerkung 2.1.10 in der Kugel $\overline{K} = \{X \mid \|X\| \le t^*\}$ genau ein Minimum \widetilde{X} von f, und es gilt $t^* \to \infty$ und $\|\widetilde{X}\| \to 0$ für $S(A) \to 0$; und Veselić [34] hat bemerkt, daß bei fast blockdiagonalem A und $f(X_0) < f(X_2)$ die Parameter X_0 und X_2 im wesentlichen identisch sind. Die gleiche Argumentation trifft auch auf X_1 statt X_2 zu, so daß die in diesem Paragraphen über das mit X_2 arbeitende Verfahren gemachten Aussagen auch für das o.a. kombinierte Verfahren qualitativ gültig sind.

Wählt man X_{ν} , $\nu \in \{1,3\}$, je nachdem welcher die Norm besser reduziert, so gilt das oben zur globalen Konvergenz gesagte entsprechend. Wegen Bemerkung 2.1.10 ist dagegen unter den Voraussetzungen (2.1.42): $f(X_1) < f(X_3)$ unmöglich, weil $\|X_1\| \le 2$ und $t^* \ge \frac{13}{6}$ ist, so daß asymptotisch stets X_3 gewählt wird. Damit gelten die in diesem Paragraphen über das mit X_3 arbeitende Verfahren gemachten Aussagen auch für das kombinierte Verfahren.

Hiermit sind die theoretischen Überlegungen abgeschlossen, und wir kommen im nächsten Kapitel zur Beschreibung der numerischen Verfahren und zur Diskussion der Ergebnisse.

4. JACOBI-ÄHNLICHE VERFAHREN FÜR DIE PRAKTISCHE BERECHNUNG DER EIGENWERTE UND EIGENVEKTOREN

In diesem Kapitel beschäftigen wir uns mit Jacobi-ähnlichen Verfahren mit zeilenzyklischer Pivotstrategie (vergl.Kapitel 2) für nichtnormale Matrizen. Im ersten Paragraphen wird eine praktisch erfolgreich anwendbare Variante des in Kapitel 3 theoretisch untersuchten Verfahrens angegeben. Das im zweiten Paragraphen beschriebene Verfahren entsteht aus dem ersten, indem man die Paardekooper-Jacobi-Transformation durch die Veselic-Householder-Transformation ersetzt, und wahlweise den symmetrischen oder schiefsymmetrischen Teil annulliert, je nachdem welcher größer ist. Beiden Verfahren liegt derselbe globale Ablaufplan zugrunde, der hier in ALGOL-ähnlicher Notation angegeben sei *):

Der Teil < Normreduzierung auf A_{rr} > ist in beiden Verfahren identisch und durch (3.2.2) und § 2.4 beschrieben (vergl. auch [4] und [38]).

```
*) Es ist |A_{ij}| := \max\{|a_{kl}| | k, l \in \{2i-1,2i\}\} . Für die Größe von \epsilon vergl. Fußnote S. 100 .
```

Auch der Teil <Normreduzierung am Pivot A_{rs} oder A_{sr}> unterscheidet sich in beiden Verfahren nicht; hier gibt es aber für jedes Verfahren zwei Varianten, je nachdem der normreduzierende Parameter bei Variante 1 nach 2.1.1 und 2.1.2 (vergl. Abb.4) oder bei Variante 2 nach 2.1.1 und 2.1.3 (vergl.Abb. 5) gewählt wird.

Im Unterschied zu Kapitel 3 (3.2.1) und [38] wird hier die Normreduzierung mit der elementaren T-(Z-)Transformation vorgenommen, je nachdem $\|A_{rs}\|(\|A_{sr}\|)$ größer ist. Das erscheint sinnvoll, da A_{sr} (A_{rs}) unter einer T-(Z-)Transformation invariant ist, und wegen der normreduzierenden Eigenschaft zu erwarten ist, daß auch $\|A_{rs}\|(\|A_{sr}\|)$ reduziert wird. Praktisch führt diese Maßnahme zu einer z.T. erheblichen Verbesserung der Konvergenz im globalen Bereich, während die Konvergenz im asymptotischen Bereich erwartungsgemäß weitgehend unberührt bleibt. Die Konvergenzbeweise aus Kap.3 gelten auch für die wahlweise Verwendung der T-bzw. Z-Transformation entsprechend, da, wie zu Beginn von § 2.1 bereits gesagt, alle für die T-Transformation gemachten Aussagen auch für die Z-Transformation gelten.

Bei dem praktischen Rechenverfahren wird auf die Beschränkung von X_1 (vergl. 2.1.12) verzichtet, da sie lediglich beweistechnischer Natur ist und für das hier verwendete zyklische Verfahren ohnehin kein Beweis der globalen Konvergenz existiert. Stattdessen wird in (1.2.4) das Minimum α gewählt, für das $|\alpha|p(\alpha)$ minimal wird. Das bleibt, nach dem am Ende von Kapitel 3 gesagten, auf die asymptotische Konvergenzordnung ohne Einfluß. Der in Variante 2 berechnete normreduzierende Parameter X_t stimmt für eine fast normale und hinreichend blockdiagonaldominante Matrix A, bis auf den Abbruchfehler der Iteration mit der Lösung X_3 überein, so daß auch für diese Variante die Aussagen von § 3.2 qualitativ zutreffen.

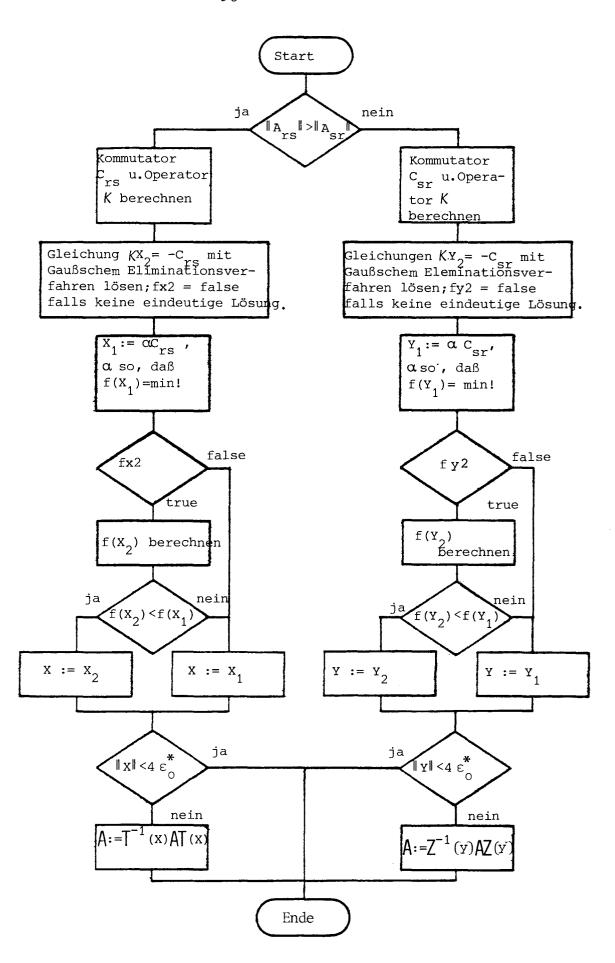


Abb. 4 : Normreduzierung am Pivot A_{rs} oder A_{sr} , Variante 1

^{*)} Vergl. Fußnote S. 100 .

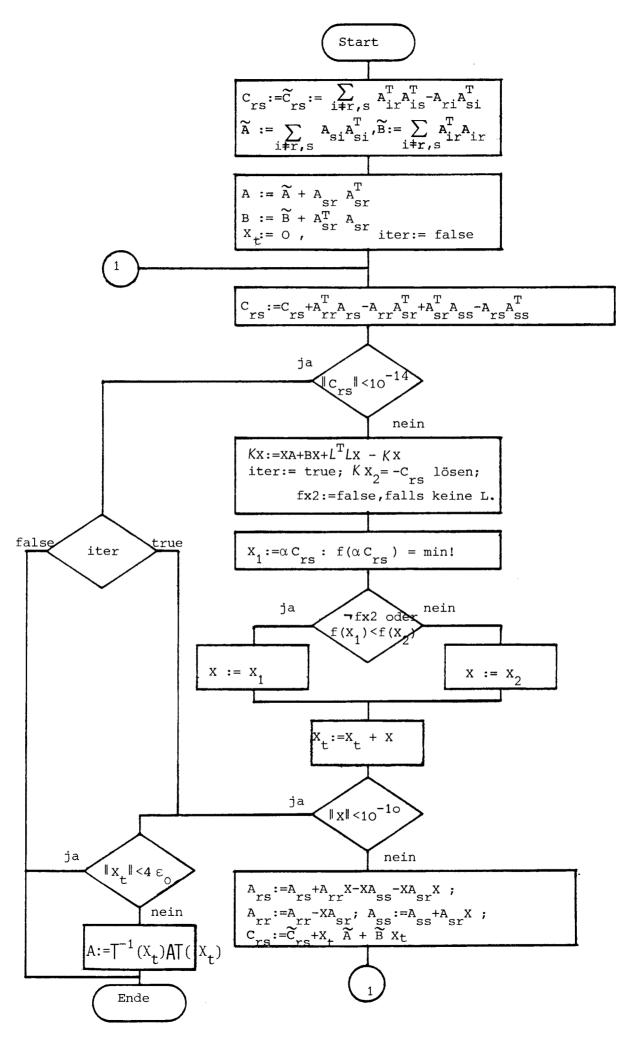


Abb.5: Variante 2 der Normreduzierung am Pivot A_{rs} (am Pivot A_{sr}) analog (vergl. Abb.4)).

Die beiden Varianten des Algorithmus zur < Normreduzierung am Pivot A_{rs} oder A_{sr} > sind in Abb.4 und Abb.5 in Form von Programmablaufplänen dargestellt. Weitere Details entnimmt der interessierte Leser dem im Anhang abgedruckten Programm für das Verfahren II (vergl. § 4.2) mit optimaler Normreduzierung (Variante 2). Dieses Programm wurde ausgewählt, weil daraus alle anderen abgeleitet werden können; die Variante 1 der Normreduzierung, indem nur ein Iterationsschritt ausgeführt wird, und das Verfahren I (vergl. § 4.1) in beiden Varianten, indem jeweils die Veselić-Householder-Transformation durch das in [26] veröffentlichte Programm, angewendet auf den schiefsymmetrischen Teil von A, ersetzt wird.

Bei den Programmen wird davon ausgegangen, daß die Matrix normiert ist, z.B. in der Form:

$$\sum_{i,j=1}^{n} |a_{ij}| = n^2$$
.

Alle Verfahren liefern für die Klasse von Matrizen, für die sie konzipiert sind, als Ergebnis eine blockdiagonaldominante Matrix \widetilde{A} , deren Außerdiagonalblöcke in der Norm kleiner als eine feste Schranke ϵ *) sind, sowie wahlweise die Transformationsmatrix R, die die Startmatrix A durch eine Ähnlichkeit in die Endmatrix \widetilde{A} transformiert, und/oder ihre Inverse R^{-1} .

Die Näherungen für die Eigenwerte von A können durch Lösung von quadratischen Gleichungen aus den Diagonalblöcken von \widetilde{A} gewonnen werden; daraus erhält man durch einfache Rechnung Näherungen für Eigenvektoren von \widetilde{A} , und mit $R(R^{-1})$ Näherungen für Rechts-(Links-)Eigenvektoren von A.

^{*)} Wir wählen $\varepsilon = \sqrt{\varepsilon_0}/100$, wobei ε_0 die kleinste positive Zahl ist, für die die Maschinenoperation $1+\varepsilon_0>1$ ist; bei der hier verwendeten DVA ist $\varepsilon_0=2^{-52}\approx 0.22\cdot 10^{-15}$.

Der Fehler in den berechneten Eigenwerten, der von Programmen zur Eigenwertberechnung einer Matrix A gewöhnlich gemacht wird, liegt nach Osborne [24] mindestens in der Größenordnung ϵ_0 A, wobei ϵ_0 die Maschinengenauigkeit ist (vergl.Fußnote S.100). Er schlägt deshalb vor, die Euklidische Norm der Matrix zunächst durch Ähnlichkeitstransformationen mit Diagonalmatrizen zu reduzieren. Darauf aufbauend haben Parlett und Reinsch [27] einen Algorithmus entwickelt, der Rundungsfehler bei diesen Transformationen vermeidet. Sie minimieren die Norm der Matrix nicht exakt, sondern gleichen die Betragssummennorm $\|\cdot\|_1$ der entsprechenden Zeilen- und Spalten-(Vektoren) bis auf einen Faktor β an , der als Maschinenbasis gewählt wird:

$$\frac{1}{\beta} \le \frac{\|a^{i}\|_{1}}{\|a_{i}\|_{1}} \le \beta$$
 , $i = 1(1)n$,

wobei a_i die i-te Spalte und a^i die i-te Zeile der Matrix A bezeichnet. Dieser Prozess wird Balancierung der Matrix genannt, und wir haben bei allen Verfahren die Möglichkeit vorgesehen, ihn vor Eintritt in die Blockdiagonalisierung durchzuführen. Die Diagonalmatrizen können so gewählt werden, daß ihre Elemente Potenzen von β sind, so daß keine Rundungsfehler auftreten, da bei allen Operationen nur der Exponent verändert wird und die Mantisse unverändert bleibt.

Wir wählen $\beta=2$, obwohl die Maschinenbasis 16 ist, weil sich die Verwendung von $\beta=16$ als zu grob und damit wenig wirksam erwiesen hat, während die Verwendung von $\beta=2$ zu guten Ergebnissen führt. Dabei sind allerdings neben den Manipulationen des Exponenten auch Verschiebungen der Mantisse durchzuführen, was aber höchstens zu dem Verlust von 3 Bit der Mantisse und einer etwas größeren Rechenzeit führt, wobei letztere aber immer noch vernachlässigbar klein gegenüber der Rechenzeit für die Blockdiagonalisierung ist. Bei schlecht balancierten Matrizen steht diesem kleinen Mehraufwand eine z.T. erhebliche Verringerung der erforderlichen Zyklen gegenüber (vergl. Beispiele 1b),2b),3b),4b),5b),6b),7 und 8).

Es ist also empfehlenswert, stets zunächst eine Balancierung der Matrix vorzunehmen, es sei denn; es ist bekannt, daß die Matrix balanciert ist (vergl. Beispiele 1a),2a),3a),4a),5a) und 6a)).

4.1 EIN VERFAHREN FÜR REELLE MATRIZEN MIT KOMPLEXEN EIGENWERTEN, DEREN IMAGINÄRTEILE GETRENNT SIND (VERFAHREN 1)

Dieses Verfahren ist aus dem in [38] beschriebenen Verfahren weiterentwickelt. Die Vertauschung der Reihenfolge der Blockdiagonal- und Außerblockdiagonaltransformationen wurde mit Rücksicht auf den Konvergenzbeweis in Kapitel 3 vorgenommen. Die wesentliche Verbesserung liegt in der Wahl der T- oder Z-Transformation, je nachdem $\|A_{rs}\|$ oder $\|A_{sr}\|$ größer ist (vergl. Abb.4), und bei Variante 2 zusätzlich in der optimalen Normreduzierung (vergl. Abb.5).

In dem Teil < Annullierung von $A_{rs}^{'\pm} > wird mit der Paardekooper-Jacobi-Transformation der Pivotblock <math>A_{rs}^{'-}$ des schiefsymmetrischen Teils $A^{'-}$ von $A^{'}$ annulliert (vergl.[26], § 2.2 und Kapitel 3). Über das in § 3.2 bewiesene hinaus, hat sich in der experimentellen Praxis die asymptotisch quadratische Konvergenz auch bei dem Auftreten von zwei reellen, oder einem doppelten reellen, nicht defektiven Eigenwert gezeigt. Bei dem Auftreten eines vierfachen reellen, nicht defektiven Eigenwertes bleibt die asymptotische Konvergenz quadratisch, solange nicht aufgrund von Rundungsfehlern im Laufe des Verfahrens daraus getrennte Eigenwerte werden, die von dem Verfahren als solche erkannt werden. Letzteres wird bei größeren Matrizen (Dimension 20) und großer Abweichung von der Normalität, d.h. $\Delta(A) \geq 1000$, mit

(4.1.1)
$$\Delta_{\mathbf{r}}(\mathbf{A}) := \frac{\Delta(\mathbf{A})}{\left(\sum_{i=1}^{n} |\lambda_{i}|^{2}\right)^{1/2}}$$

beoabachtet. In diesem Fall tritt, ebenso wie bei Matrizen mit mehr als zwei verschiedenen reellen Eigenwerten, oder konjugiert komplexen Eigenwertpaaren mit verschiedenen Real- und gleichen Imaginärteilen, noch Konvergenz gegen Normalität auf, ohne daß eine Blockdiagonalgestalt erzeugt wird (vergl.[38]). Das Verfahren bricht abnormal ab, weil entweder die Matrix normal und der schiefsymmetrische Teil blockdiagonal im Rahmen der Rechengenauigkeit ist (vgl.Beispiel 9 in § 4.3), also keine Transformation mehr ausgeführt wird, oder der obige Zustand nach 50 Zyklen noch nicht erreicht ist (vergl. Beispiel 8 in § 4.3).

Bei dem Auftreten eines doppelten komplexen, nicht defektiven Eigenwertpaares ist die Konvergenz auch noch asymptotisch quadratisch, während bei gehäuftem Auftreten mehrfacher komplexer Eigenwerte keine quadratische Konvergenz mehr zu erkennen ist. Allerdings wird die Konvergenz bei der Größenordnung von S(A), wo bei getrennten Imaginärteilen die quadratische Konvergenz einsetzt, deutlich schneller. Dasselbe Verhalten zeigt sich auch bei Matrizen mit einem doppelten reellen, defektiven Eigenwert; hier kommt es allerdings häufig zu einigen quadratisch konvergenten Schritten, was sich aber meist nicht bis zum Ende des Verfahrens fortsetzt (vergl. Diskussion der Beispiele 1 und 2 in § 4.3).

Zusammenfassend läßt sich sagen:

Das Verfahren konvergiert asymptotisch quadratisch auf Matrizen mit konjugiert komplexen Eigenwertpaaren, deren Imaginärteile getrennt sind, wobei einzelne doppelt, aber nicht defektiv sein dürfen, und die zusätzlich höchstens einen doppelten reellen, nicht defektiven Eigenwert oder zwei verschiedene reelle Eigenwerte haben.

Es konvergiert noch gegen Blockdiagonalgestalt, wenn die mehrfachen, nicht defektiven Eigenwerte gehäuft auftreten, die Imaginärteile der verschiedenen Eigenwerte getrennt sind, und wenn ein doppelter reeller, defektiver Eigenwert vorliegt. Im letzteren Fall wird die Blockdiagonalgestalt ohne katastophales Anwachsen der Kondition der Transformationsmatrix erreicht.

Es konvergiert sehr langsam unter katastrophalem Anwachsen der Kondition der Transformationsmatrix gegen Blockdiagonalgestalt, wenn mehrfache komplexe, defektive oder reelle, defektive Eigenwerte mit Vielfachheit größer als 2 auftreten.

Das Verfahren konvergiert nicht gegen Blockdiagonalgestalt, wenn komplexe Eigenwerte mit gleichen Imaginär- und verschiedenen Realteilen oder mehr als zwei verschiedene reelle Eigenwerte auftreten (vergl. auch [38], sowie die Beispiele 8 und 9 in § 4.3).

4.2 EIN VERFAHREN FÜR ALLGEMEINE REELLE MATRIZEN (VERFAHREN II)

Dieses Verfahren ist von dem Verfahren I aus § 4.1 abgeleitet. Die Paardekooper-Jacobi-Transformation zum Annullieren von A'rs wird dabei ersetzt durch die Veselië-Householder-Transformation $A'' = Q^T A' Q$ zum Annullieren von A'_{rs} oder A'_{rs} . Ist die Bedingung

wobei $\mu_{i}^{(k)}$, $i \in \{1,2\}$ die Eigenwerte von $A_{kk}^{'}$, $1 \le k \le p$ sind, erfüllt, so wird der Pivotblock $A_{rs}^{'+}$ des symmetrischen Teils $A^{'+}$ von $A^{'}$, anderenfalls der Pivotblock $A_{rs}^{'-}$ des schiefsymmetrischen Teils $A^{'-}$ von $A^{'}$ annulliert. Die Bedingung (4.2.1) ist analog zu der entsprechenden Bedingung im komplexen Verfahren von Eberlein [5] gewählt.

Der Beweis der globalen Konvergenz von Verfahren II ist sicher schwerer zu erbringen, als der für das komplexe Verfahren von Eberlein; und auch das Problem ist noch offen. Weil auch der Beweis asymptotisch quadratischer Konvergenz technisch aufwendiger ist, als der Beweis in § 3.2 belassen wir es hier bei der folgenden Motivation.

Wegen $S^2(Q^TA'^{\pm}Q) = S^2(A'^{\pm}) - 2\|A_{rs}^{\prime\pm}\|$ ist es günstiger, den Teil zu annullieren, welcher größer ist, in der Erwartung, daß der andere Teil dabei nicht zu stark anwächst. Daher rührt der jeweils erste Summand in der Bedingung (4.2.1), der einen wesentlichen Beitrag leistet, solange noch keine Blockdiagonaldominanz vorliegt.

Hat die Matrix A' getrennte Eigenwerte, und ist sie bereits normal und hat Blockdiagonalgestalt bis auf ein $O(\epsilon)$, mit ϵ hinreichend klein, so ist der zweite Summand bestimmend, und es wird die Transformation zum Annullieren des symmetrischen (schiefsymmetrischen) Teils ausgewählt, je nachdem die Realteile (Imaginärteile) der Eigenwerte der entsprechenden Diagonalblöcke besser getrennt sind.

Damit ist aber nach [35],[1]:

$$(4.2.2) \qquad \widehat{Q}_{rs} = \begin{pmatrix} I + O(\epsilon^2) & O(\epsilon) \\ O(\epsilon) & I + O(\epsilon^2) \end{pmatrix} \cdot \begin{pmatrix} U_1 & O \\ O & U_2 \end{pmatrix} ,$$

wobei U_1 und U_2 orthogonale 2×2 -Matrizen sind, und man kann analog zu § 3.2 zeigen, daß die Größenordnung der Abweichung von der Normalität und von S(A') unter einer Q-Transformation erhalten bleibt; man beachte, daß unter einer Ähnlichkeitstransformation mit einer blockdiagonalen, orthogonalen Matrix die Norm eines jedes Blockes von A' und C(A') invariant ist.

Weiter gilt nach der kompletten Transformation mit $A_{rs}^{"+}=0$ ($A_{rs}^{"-}=0$), wegen $\|\widetilde{C}_{rs}^{"}\|=0(\epsilon^2)$ und der Getrenntheit der Realteile (Imaginärteile) der Eigenwerte der entsprechenden Diagonalblöcke auch: $A_{rs}^{"-}=0(\epsilon^2)(A_{rs}^{"+}=0(\epsilon^2))$. Damit wird dieses Verfahren für alle Matrizen mit getrennten Eigenwerten asymptotisch quadratische Konvergenz liefern.

Um asymptotisch die Form (4.2.2) zu erhalten, muß man die "richtigen" Eigenwerte von $\widehat{A}_{rs}^{'+}$ $(\widehat{A}_{rs}^{'-})$ zum shift *) auswählen, d.h. diejenigen, die den Eigenwerten des symmetrischen (schiefsymmetrischen) Teils $A_{rr}^{'+}(A_{rr}^{'-})$ des Diagonalblocks $A_{rr}^{'}$ am nächsten sind (vergl.Abb.6 und Abb.3). Die globale Konvergenz kann durch eine ungeschickte Auswahl der zum shift verwendeten Eigenwerte verzögert oder sogar verhindert werden. Wir wählen bei der Transformation zum Annullieren des symmetrischen Teils die am dichtesten benachbarten Eigenwerte von $\widehat{A}_{rs}^{'+}$ zum shift, damit mehrfache Eigenwerte des symmetrischen Teils $A^{'+}$ von $A^{'-}$ zusammengeführt werden, was bei Matrizen mit konjugiert komplexen Eigenwerten λ = a ± ib wichtig ist, da der symmetrische Teil der assoziierten normalen Matrix den doppelten reellen Eigenwert a hat. Eine unnötige "Vertauschung" der Diagonalblöcke verhindert man durch eine, gegebenenfalls durchzuführende Permutation der Transformationsmatrix \widehat{Q}_{rs} . Es gilt mit

$$(4.2.3) \qquad \widehat{Q}_{rs}^{T} \widehat{A}_{rs}^{'\pm} \widehat{Q}_{rs} = \begin{pmatrix} A_{rr}^{"\pm} & 0 \\ 0 & A_{ss}^{"\pm} \end{pmatrix}$$

und

$$(4.2.4) P := \begin{pmatrix} 0 & I \\ & \\ I & 0 \end{pmatrix} , \quad \widehat{\widehat{Q}}_{rs} := \widehat{Q}_{rs} P$$

die Gleichung:

$$(4.2.5) \qquad \widehat{\widehat{Q}}_{rs}^{T} \widehat{A}_{rs}^{'\pm} \widehat{\widehat{Q}}_{rs} = \begin{pmatrix} A_{ss}^{''\pm} & O \\ O & A_{rr}^{''\pm} \end{pmatrix}$$

Die Transformationsmatrix $\widehat{\mathbb{Q}}_{rs}$ bewirkt also genau die andere Verteilung der Diagonalblöcke wie $\widehat{\mathbb{Q}}_{rs}$, weshalb wir diejenige orthogonale Transformation (4.2.3) oder (4.2.5) wählen, deren Transformationsmatrix die kleinere Außerblockdiagonale hat (vergl. Abb.7), d.h.

$$\widehat{Q}_{rs} \quad \text{falls} \quad S(\widehat{Q}_{rs}) \leq S(\widehat{\overline{Q}}_{rs}) \quad ,$$

$$\widehat{\overline{Q}}_{rs} \quad \text{sonst.}$$

Bei der praktischen Berechnung wird nicht mit der Matrix $\widehat{\mathbb{Q}}_{rs}$ bzw. $\widehat{\mathbb{Q}}_{rs}$ gearbeitet, sondern mit den Householder-Vektoren wund v (vergl.(2.3.1)), da so weniger Operationen benötigt werden. Für weitere Details verweisen wir den interessierten Leser auf das im Anhang abgedruckte Programm.

Da die Transformationsmatrix zum Annullieren von $A_{rs}^{'-}$ wegen der einfacheren Formeln für die Eigenwerte von $A_{rs}^{'-}$ schneller zu berechnen ist, als dieselbe für A_{rs}^{+} , wird diese Transformationsart stärker gewichtet ($\Lambda < 1$). Experimentell hat sich der Faktor $\Lambda = 10^{-2}$ gut bewährt; die Transformation zum Annullieren des symmetrischen Teils wird nur dann verwendet, wenn sie zur Blockdiagonalisierung der Matrix A unentbehrlich ist, d.h. in solchen Fällen, in denen das Verfahren I keine Blockdiagonalgestalt liefert **).

^{*)} Vergl. § 1.2, § 2.3 und das im Anhang abgedruckte Programm.

^{**)} Vergl. § 4.1

Zur Umschaltung der Strategie bei der Auswahl der zum Shift verwendeten Eigenwerte, wie sie nach den obigen Ausführungen erforderlich ist, wird für das Verfahren II der Ablaufplan zu Beginn von Kapitel 4 nach der Zeile "while ..." erweitert um die Zeile:

$$A_{o}$$
 klein := $S(A) < 10^{-2}$;.

Der globale Ablauf des Teils < Annulliere A_{rs}^{\pm} > ist in dem Programm-ablaufplan in Abb. 6 u. 7 dargestellt; weitere Details entnimmt der interessierte Leser dem im Anhang abgedruckten Programm.

Die experimentellen Beobachtungen zur Konvergenz des Verfahrens II decken sich in allen Fällen, in denen das Verfahren I nicht versagt, weitgehend mit den in § 4.1 gemachten Feststellungen. Die benötigte CPU-Zeit ist bei gleicher Anzahl von Zyklen und Transformationen annähernd gleich; allerdings werden für etliche Beispiele weniger Zyklen und Transformationen benötigt. Auch bei dem gehäuften Auftreten mehrfacher komplexer und eines vielfachen reellen Eigenwertes ist die Konvergenz im Gegensatz zu Verfahren I asymptotisch quadratisch, mit Ausnahme bei großer Abweichung von der Normalität ($\Delta_r(A) \ge 1000$), wo keine quadratische Konvergenz erkannt werden kann; in diesen Fällen bricht das Verfahren I allerdings abnormal nach 50 Zyklen ab (vergl. § 4.1). Das kann aber nicht als Mangel empfunden werden, denn bei solchen Matrizen liefert kein Eigenwert-Verfahren zufriedenstellende Ergebnisse.

Auch bei dem Auftreten von mehr als zwei verschiedenen reellen oder verschiedenen mehrfachen reellen, sowie konjugiert komplexen Eigenwertpaaren mit gleichen Imaginär- und verschiedenen Realteilen ist die Konvergenz des Verfahrens II asymptotisch quadratisch.

Zusammenfassend kann festgestellt werden, daß die Konvergenz bei Verfahren II stets asymptotisch quadratisch ist, solange kein defektiver Eigenwert vorliegt. Bei dem Auftreten eines doppelten reellen, defektiven Eigenwertes ist die Konvergenz asymptotisch schnell (fast quadratisch), ohne daß die Kondition der Transformationsmatrix katastrophal anwächst. Liegt ein defektiver reeller Eigenwert mit Vielfachheit größer als zwei, oder ein defektiver komplexer Eigenwert vor, so tritt erwartungsgemäß auch hier langsame Konvergenz gegen Blockdiagonalgestalt bei gleichzeitiger Divergenz der Kondition der Transformationsmatrix auf.

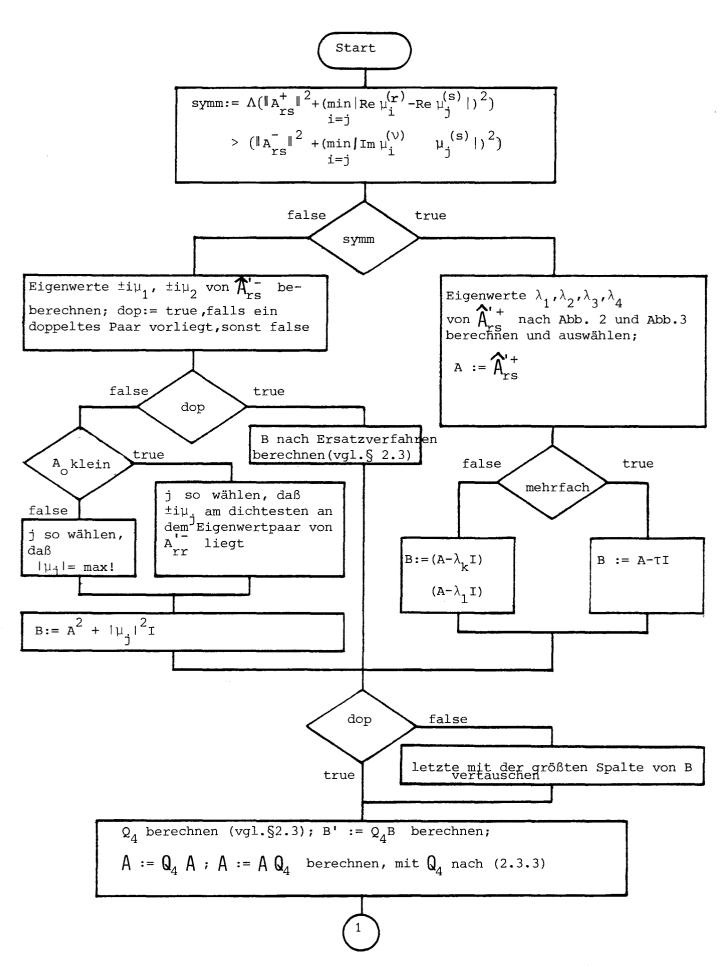


Abb.6: Veselić-Householder-Transformation zum Annullieren von A_{rs}^{\pm} (1.Teil)

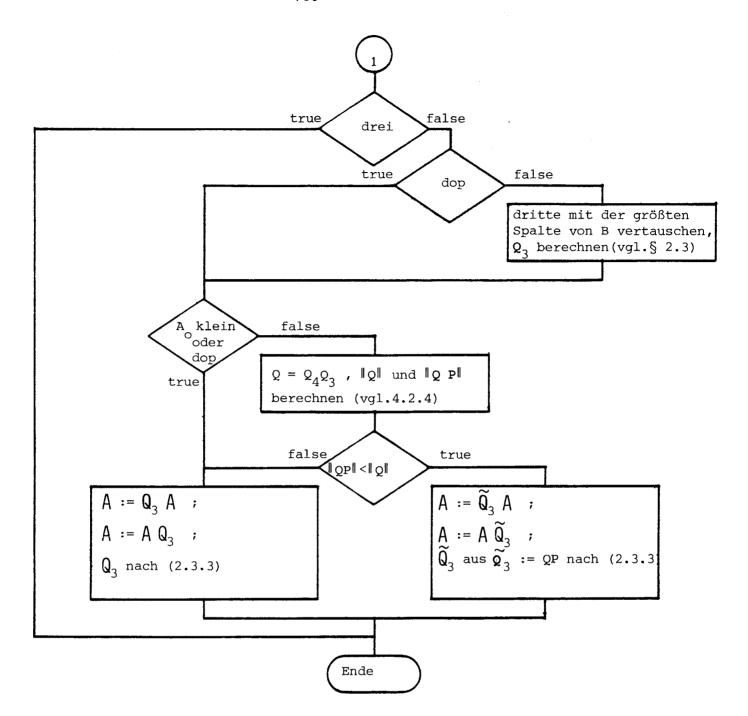


Abb.7 : Veselić-Householder-Transformation zum Annullieren von A_{rs}^{\pm} (2.Teil).

4.3 NUMERISCHE RESULTATE

Um das Konvergenzverhalten der beiden Verfahren aus § 4.1 und § 4.2 zu untersuchen, werden jeweils 4 Varianten, nämlich mit einfacher (optimaler) Normreduzierung, sowie mit und ohne Balancierung der Ausgangsmatrix auf 130 Testmatrizen angewendet. Die Aussagen von § 4.1 und § 4.2 beruhen auf der Auswertung der Ergebnisse von generierten Testmatrizen mit vorgegebenen Eigenwerten. Ausgehend von der Jordanschen Normalenform J bildet man

$$(4.3.1) \qquad A = Z(Y)^{-1} T(X)^{-1} J T(X) Z(Y) ,$$

wobei T(X) und Z(Y) analog zu (2.1.1) und (2.1.2) mit dim $X=\dim Y=(\dim A)/2$ definiert sind. Die Elemente von X bzw. Y werden mit einem Zufallszahlengenerator bestimmt, und liegen gleichverteilt wahlweise im Intervall [0,0.1],[0,1] oder [0,10]. Um Fehler in den Eingangsdaten zu vermeiden, wurden die Operationen in (4.3.1) durch ein PL/I-Programm mit "extended precision", d.h. $\varepsilon_0=2^{-108}$ *) durchgeführt.

Die hier ausführlicher vorgestellten Beispiele sollen einen stärkeren Bezug zur Praxis haben. Wir wählen zunächst

$$(4.3.2) G_1 = \begin{pmatrix} O & L^T \\ -L & -\tau I \end{pmatrix} G_2 = \begin{pmatrix} O & I \\ -T & -\tau I \end{pmatrix}$$

 $\tau > 0$, mit

$$T = (p+1)^{2} \begin{pmatrix} 2 & -1 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ & \ddots & \ddots & \ddots & & \\ 0 & \ddots & \ddots & \ddots & \ddots & \\ 0 & \ddots & \ddots & 0 & -1 & 2 & -1 \\ 0 & \ddots & \ddots & 0 & -1 & 2 \end{pmatrix}, \text{ dim } T = p$$

und

$$LL^{T} = T$$
, L obere Dreiecksmatrix (verg1. [38])

^{*)} Vergl. Fußnote Seite 100.

Die Matrizen (4.3.2) erhält man durch Linearisierung des quadratischen Eigenwertproblems, das in der Theorie der gedämpften linearen Schwingungen auftritt (vergl. Veselić [37]). Die Eigenwerte von $\mbox{\bf G}_1$ und $\mbox{\bf G}_2$ sind gegeben durch die Gleichungen

(4.3.3)
$$\lambda_{12}^{(k)}(p) = \frac{-\tau \pm \sqrt{\tau^2 - 4t_k(p)}}{2}, \quad k = 1(1)p,$$

wobei $t_{\nu}(p)$ die Eigenwerte von T sind:

$$t_k(p) = (2(p+1))^2 \sin^2 \frac{k\pi}{2(p+1)}$$
, $k = 1(1)p$.

Zum Test wählen wir die Matrizen:

1) a)
$$A = G_1$$
, b) $A = G_2$, jeweils $p = 10$, $\tau = 2\sqrt{t_1(p)}$,

2) a)
$$A = G_1$$
, b) $A = G_2$, jeweils $p = 50$, $\tau = 2\sqrt{t_1(p)}$,

3) a)
$$A = G_1$$
, b) $A = G_2$, jeweils $p = 10$, $\tau = 1$

4) a)
$$A = G_1$$
, b) $A = G_2$, jeweils $p = 50$, $\tau = 1$

5) a)
$$A = G_1$$
, b) $A = G_2$, jeweils $p = 10$, $\tau = 30$

6) a)
$$A = G_1$$
, b) $A = G_2$, jeweils p = 50, τ = 100.

Die Dämpfung τ ist dabei so gewählt, daß in den Beispielen 1 und 2 der betragskleinste Eigenwert doppelt reell, defektiv ist (aperiodische Dämpfung der kleinsten Eigenfrequenz), in den Beispielen 3 und 4 nur komplexe Eigenwerte (schwache Dämpfung) und in den Beispielen 5 und 6 sowohl komplexe wie einfache reelle Eigenwerte auftreten (starke Dämpfung).

Sei

$$(4.3.4) \qquad W_n = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ & & \ddots & \ddots & & \\ 0 & 0 & \dots & 0 & 0 & 1 \\ \varepsilon & 0 & \dots & 0 & 0 & 0 \end{pmatrix} , \dim W_n = n, \varepsilon > 0 ;$$

die Eigenwerte von W_n sind alle komplexen n-ten Wurzeln von ε (vergl. [38]).

Wir wählen $\varepsilon = 0.01$ und

7)
$$A = W_{19} \oplus (0) ,$$

8)
$$A = W_{20}$$
.

Damit sind die Imaginärteile der Eigenwerte in Beispiel 7) getrennt, während in Beispiel 8) 4 Paare konjugiert komplexer Eigenwerte mit gleichen Imaginär- und verschiedenen Realteilen auftreten. Wir testen noch ein Beispiel von Eberlein [6] (vergl. auch [38]):

9)
$$A = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \oplus (0)$$
,

mit den einfachen Eigenwerten $\lambda = -1$, $\pm i$, $(1/\sqrt{2}) \cdot (1 \pm i)$, $(1/\sqrt{2}) \cdot (-1 \pm i)$ und 0 .

Zum Schluß noch ein Beispiel aus der Praxis, das bei der Lösung eines Schwingungsproblems auftritt: Beispiel 10. Die Matrix A hat die Dimension 90 und ausschließlich konjugiert komplexe Eigenwerte, deren Imaginärteile getrennt sind und zwischen 8 und 756 liegen; die Realteile sind negativ und liegen zwischen -1.352 und -0.8·10⁻⁵. Das durch die Matrix repräsentierte dynamische System ist eine mit Federn und Dämpfern gelagerte Maschinenbau-Struktur. Die Matrix A ist dünn besetzt und besitzt eine Bandstruktur mit "Löchern".

Um neben diesen konkreten Beispielen statistisches Material über die Verfahren zu erhalten, wurden alle Varianten noch auf je 10 Zufallsmatrizen *) der Dimension 10, 20 und 40, sowie 5 Zufallsmatrizen *) der Dimension 80 angewendet.

^{*)} Zufallsmatrizen sind hier Matrizen, deren Elemente von einem Zufallszahlengenerator erzeugt werden, der Zahlen zwischen -1 und 1 gleichverteilt generiert.

Die interessantesten Daten sind im folgenden tabellarisch aufgeführt – bei den Zufallsmatrizen *) jeweils die Mittelwerte der normal beendeten Läufe; dabei bedeuten:

"Zyklen" Anzahl der benötigten Zyklen, um ein Abbruchkriterium zu erfüllen **); ein abnormaler Abbruch wird angezeigt. Falls asymptotisch quadratische Konvergenz vorliegt wird in Klammer der
Zyklus angegeben, nach dem die quadratische Konvergenz einsetzt.

"Transform" Anzahl der durchgeführten normreduzierenden, sowie orthogonalen Transformationen; durch Komma getrennt. Bei Verfahren I sind die einzelnen Jacobi-Rotationen gezählt; das sind bis zu 4 je Block.

"CPU-Zeit" Vom Verfahren benötigte CPU-Zeit, ohne Ein-/
Ausgabe der Matrix, aber mit Berechnung und
Ausgabe der Verfahrensstatistik wie im Programm
im Anhang.

"Kondition" Kondition der Transformationsmatrix R, definiert als

$$(4.3.5) K(R) = ||R^{-1}||_1 ||R||_1,$$

mit

$$\|R\|_1 = \max \left\{ \sum_{i=1}^n |r_{ij}| | 1 \le j \le n \right\}$$
.

Wenn die Ausgangsmatrix balanciert wird, wird die Kondition der Gesamttransformationsmatrix B, wobei die Transformationsmatrix B die Balancierung der Matrix bewirkt, in Klammern angegeben.

^{*)} Vergl. Fußnote auf Seite 112.

^{**)} Vergl. Algorithmus Seite 96 .

"S(A)" Euklidische Norm des außerblockdiagonalen Teils der Matrix nach Abschluß des Verfahrens.

"C(A)" Euklidische Norm des Kommutators von A und

"S(C(A))" Euklidische Norm des außerblockdiagonalen Teils des Kommutators von A nach Abschluß des Verfahrens.

Die Tabellen beginnen auf Seite 116.

Die Rechenzeit je Zyklus (Transformation) ist bei beiden Verfahren etwa gleich; die Variante 2 benötigt jeweils mehr Rechenzeit, was sich bei größerer Dimension erwartungsgemäß weniger bemerkbar macht, liefert aber nur bei starker Abweichung von der Normalität und sonst in Ausnahmefällen eine z.T. erhebliche Reduzierung der erforderlichen Zyklen, was dann aber, besonders bei großen Matrizen, zu einer wesentlichen Verringerung der Rechenzeit führt.

In den Beispielen 1 und 2 liegt ein doppelter reeller, defektiver Eigenwert vor; trotzdem benötigen alle Varianten - wenn überhaupt - nur wenig mehr Zyklen als für die Beispiele 3 und 4 (schwache Dämpfung) und z.T. weniger als für die Beispiele 5 und 6 (starke Dämpfung), wo asymptotisch quadratische Konvergenz auftritt. Eine genaue Analyse der Verfahrens-Statistik zeigt, daß trotz des reellen, defektiven Eigenwertes bei den Beispielen 1 und 2 asymptotisch quadratische (2a) oder"fast quadratische" Konvergenz vorliegt. Für Beispiel 1a) ist die Konvergenz bei allen Varianten vom dritten bis zum sechsten Zyklus quadratisch und wird dann wieder linear; für Beispiel 1b) tritt dieser Effekt mit Balancierung vom dritten bis fünften, ohne Balancierung vom siebten bis neunten Zyklus auf. Für Beispiel 2b) kann bei keiner Variante quadratische Konvergenz erkannt werden, obwohl auch hier die Konvergenz in den letzten fünf bis sechs Zyklen deutlich schneller wird.

Für die Beispiele 3 und 4 gehen alle wesentlichen Daten aus den Tabellen hervor. Bei Beispiel 5(6), wo 10(32) verschiedene reelle Eigenwerte und 5(34) konjugiert komplexe Eigenwertpaare vorliegen, versagt erwartungsgemäß das Verfahren I. Bei Beispiel 6b) ohne Balancierung bricht das Verfahren I nach 37 bzw. 34 Zyklen definiert abnormal ab, weil die zur Verfügung gestellte CPU-Zeit verbraucht ist *); es wird nicht neu aufgesetzt, weil auch nach 50 Zyklen kein

^{*)} Vergl. Programm im Anhang.

wesentlich anderes Ergebnis zu erwarten ist. Aus weiteren Beispielen, die hier nicht protokolliert sind, geht hervor, daß die Abweichung der Matrix A von der Normalität $(\Delta(A))$ bei dem Abbruch des Verfahrens I nach 50 Zyklen noch umso größer ist, je größer die Dämpfung τ gewählt wird. Das Verfahren II benötigt mit zunehmender Dämpfung τ mehr Zyklen um die Matrix zu blockdiagonalisieren; die Konvergenz ist in den letzten drei bis vier Zyklen quadratisch.

Für die Beispiele 1) bis 6) vom Typ ${\sf G}_1$ steigt die Kondition der Transformationsmatrix bei allen Varianten schrittweise auf den tabellierten Wert an. Für die entsprechenden Beispiele vom Typ ${\sf G}_2$ ohne Balancierung steigt die Kondition der Transformationsmatrix bei Variante 1 beider Verfahren im ersten – z.T. auch noch im zweiten – Zyklus sprunghaft auf das 7- bis 65-fache des protokollierten Endwertes an, um dann "exponentiell" auf den Endwert abzufallen. Bei Variante 2 erreicht der Faktor sogar Werte von 34 bis 6600; sonst ist das Verhalten ähnlich wie bei Variante 1. Durch Balancierung der Eingangsmatrix mit der Basis 16 verringert sich der Faktor auf Werte von 1.2 bis 3, während sich bei der Balancierung mit der Basis 2 ein Verhalten wie bei den entsprechenden Matrizen vom Typ ${\sf G}_1$ zeigt *) . Die diesbezüglichen Bemerkungen zu Beginn von Kapitel 4 beruhen auf diesen Beobachtungen.

Für Matrizen der Form (4.3.4) bringt die Balancierung der Matrix A für ε ≤ 0.1 eine, mit kleiner werdendem ε zunehmende, Verringerung der erforderlichen Zyklen; die Variante 2 bringt bei nicht balancierter Matrix eine geringe Verbesserung gegenüber Variante 1. Für Beispiel 7 liefert das Verfahren II bei nicht balancierter Matrix A etwas bessere Ergebnisse als Verfahren I. Für Beispiel 8 – und andere Matrizen dieser Form – liefert das Verfahren I erwartungsgemäß keine Blockdiagonalgestalt, und die nach 50 Zyklen erzeugte Endmatrix hat noch eine große Abweichung von der Normalität. Dagegen zeigt das Verfahren II ein ähnliches Verhalten wie für Beispiel 7.

^{*)} Vergl. Kap. 4, S. 101.

	Varian	t e 1	Varia	nte 2
	ohne	mit	ohne	mit
Beispiel 1a)	Balancieru	ng	Balancierung	
Zyklen	8		8	
Transform. normred., ortho.	259,1075		259,1067	
CPU-Zeit	8.91		9.84	
Kondition	142	dto.	145	dto.
S(A)	0.38 ₁₀ -12		0.36 ₁₀ -12	
" C(A) ", S(C(A))	0.01,03 ₁₀ -11		0.01,03 ₁₀ -11	
Deionic 1 1h)				
Beispiel 1b)	11	8	10	0
Zyklen Transform.	11	8	10	8
normred., ortho.	474 , 1852	291,1116	426 , 1680	277,1118
CPU-Zeit	13.88	9.20	16.97	10.33
Kondition	629	201 (378)	320	201 (366)
S(A)	0.22 ₁₀ -11	0.93 ₁₀ -11	0.12 ₁₀ -9	0.74 ₁₀ -11
"C(A)",S(C(A))	0.0003,0.2 ₁₀ -11	0.0004,08 ₁₀ -11	0.0004,0.1 ₁₀ -9	0.0004,0.6 ₁₀ -11
Beispiel 2a)				<u>.</u>
Zyklen	9(6)		9(6)	
Transform. normred.,ortho.	8429,33977		8431,34068	
CPU-Zeit	905.66	dto.	926•50	dto.
Kondition	380		384	
S(A)	0.20 ₁₀ -11		0.20 ₁₀ -11	
"C (Д)", S(C(Д))	0.02,0.3 ₁₀ -10		0.02,0.3 ₁₀ -10	
		·-··		· · · · · · · · · · · · · · · · ·
Beispiel 2b)	4.5	10	4.5	4.4
Zyk1en	15	12	15	11
Transform. normred.,ortho.	17629,69148	9663,37782	17279,68275	9091,36356
CPU-Zeit	1700,14	1071.37	1827.23	1030.12
Kondition	4865	2085 (4349)	4891	2084 (4520)
S(A)	0.50 ₁₀ -10	0.64 ₁₀ -11	0.17 ₁₀ -11	0.1210-9
" C(A) ", S(C(A))	0.2 ₁₀ -4,0.1 ₁₀ -10	0.2 ₁₀ -4,0.1 ₁₀ -11	0.2 ₁₀ -4,0.3 ₁₀ -12	0.4 ₁₀ -4,0.3 ₁₀ -10
<u> </u>				

	Variante	1	Variante	2
	ohne	mit	ohne	mit
Beispiel 1a)	Balancierung		Balancierung	2
Zyklen	7		7	
Transform.				
normred.,ortho.	281,276	_	281,275	
CPU-Zeit	8.53	dto.	10.15	dto.
Kondition	108		108	
S(A)	0.25 ₁₀ -10		0.24 ₁₀ -10	
$\mathbb{C}(A)$, $\mathbb{S}(\mathbb{C}(A))$	0.03,0.2 ₁₀ -9		0.02,0.2 ₁₀ -9	
Beispiel 1b)				
Zyklen	11	8	11	8
Transform. normred.,ortho.	457,442	315 , 302	467 , 458	298,295
CPU-Zeit	13.5	9.63	19.17	11.06
Kondition	415	210(387)	403	197 (383)
S(A)	0.28 ₁₀ -12	0.44 ₁₀ -10	0.49 ₁₀ -11	0.13 ₁₀ -10
<pre>#C(A)#,S(C(A))</pre>	0.0002,0.3 ₁₀ -12	0.0004,0.4 ₁₀ -10	0.0001,0,4 ₁₀ -11	0.0004,0.1 ₁₀ -10
			}	
Beispiel 2a)				
Zyklen	9 (6)		9 (6)	
Transform.				
normred., ortho.	8832,9263	_	8820,9270	1.
CPU-Zeit	945.66	dto.	979.35	dto.
Kondition	414		409	
S(A)	0.58 ₁₀ -12		0.56 ₁₀ -12	
"С(Д)",S(С(Д))	0.02,0.3 ₁₀ -10		0.02,0.3 ₁₀ -10	
Beispiel 2b)				
Zyk1en	15	12	15	12
Transform.	17101 16600	0710 0111	45056 46056	0445 0007
normred., ortho.	17184,16688	9542,9111		9143,8983
CPU-Zeit	1639.20	1035.85	1802.20	1040.49
Kondition	5529	2004 (4168)	4771	2216 (4433)
S(A)	0.60 ₁₀ -12	0.55 ₁₀ -11	10	0.91 ₁₀ -11
$\mathbb{C}(A)$, $S(C(A))$	0.1 ₁₀ -4,0.1 ₁₀ -12	0.2 ₁₀ -4,0.1 ₁₀ -11	0.4 ₁₀ -4,0.4 ₁₀ -10	0.2 ₁₀ -4,0.2 ₁₀ -11

	Variante	: 1	Variante	2
	ohne	mit	ohne	mit
Beispiel 3a)	Balancierung		Balancierung	
Zyklen	6(2)		6(2)	
Transform. normred.,ortho.	238,932		237,930	
CPU-Zeit	7.24	_	8.11	dto.
Kondition	16.7	dto.	15.8	
S(A)	0.37 ₁₀ -14		0.38 ₁₀ -14	
 С(Д) ,S(С(Д))	0.1 ₁₀ -11,0.1 ₁₀ -12		0.2 ₁₀ -11,0.1 ₁₀ -12	
Beispiel 3b)				
Zyklen	10 (5)	6 (2)	10 (5)	6 (2)
Transform. normred., ortho.	423,1648	264,1007	424,1676	258 , 1006
CPU-Zeit	12.50	7.69	17.05	9.16
Kondition	248.9	29.2(209)	299.8	29.9(201)
S(A)	0.89 ₁₀ -15	0.56 ₁₀ -14	0.45 ₁₀ -14	0.71 ₁₀ -14
"C(A)",S(C(A))	0.7 ₁₀ -14,0.2 ₁₀ -14	0.2 ₁₀ -13,0.2 ₁₀	-13 0.1 ₁₀ -13,0.1 ₁₀ -13	0.2 ₁₀ -13,0.2 ₁₀ -13
Beispiel 4a)				
Zyklen	8 (5)		8 (5)	
Transform.				
normred., ortho.	8319,33475	,	8318,33530	
CPU-Zeit	868.94	dto.	888.3	dto.
Kondition	72		74	
S(A)	0.94 ₁₀ -13		0.11 ₁₀ -12	
"C(A)",S(C(A))	0.6 ₁₀ -10,0.2 ₁₀ -10		0.6 ₁₀ -10, 0.2 ₁₀ -10	
Beispiel 4b)				
Zyklen	15 (11)	10 (7)	15 (12)	10 (7)
Transform. normred.,ortho.	17417,68305	9703 , 37837	17006,67416	9483,37685
CPU-Zeit	1685.37	1007.12	1809.34	1031.39
Kondition	4952	506 (4586)	4860	384 (4616)
S(A)	0.57 ₁₀ -14	0.48 ₁₀ -14	0.72 ₁₀ -13	0.88 ₁₀ -13
"C(♠)",S(C(♠))	0.3 ₁₀ -13,0.1 ₁₀ -13	0.2 ₁₀ -13,0.9 ₁₀	-14	10
(n) - • - (- (n))	10 13,0110 13	10 10,000	0.110-12,0.110-12	0.9 ₁₀ -13,0.9 ₁₀ -13

	Variant	e 1	Variante	2
	ohne	mit	ohne	mit
Beispiel 3a)	Balancierung		Balancierung	
Zyklen	6(3)		6 (3)	
Transform. normred.,ortho.	254,253		254,253	
CPU-Zeit	7 . 69	dto.	8.91	dto.
Kondition	16.98		14.77	
S(A)	0.41 ₁₀ -10		0.41 ₁₀ -10	
 С(Д) ,S(С(Д))	0.2 ₁₀ -9,0.2 ₁₀ -9		0.210-9,0.210-9	
Beispiel 3b)				
Zyklen	10 (6)	6 (3)	9 (6)	6 (3)
Transform. normred.,ortho.	441,427	270,259	405,395	269,258
CPU-Zeit	12.77	7.78	16.37	9.56
Kondition	277	30.4 (188)	329	30.8(182)
S(A)	0.23 ₁₀ -10	0.13 ₁₀ -11	0.13 ₁₀ -9	0.13 ₁₀ -11
C(A), ,S(C(A))	0.6 ₁₀ -11,0.6 ₁₀ -11	0.2 ₁₀ -11,0.2 ₁₀ -11	0.210-9,0.210-9	0.1 ₁₀ -11,0.1 ₁₀ -11
Beispiel 4a)				
Zyklen	9 (6)		9 (6)	
Transform. normred.,ortho.	8742 , 9228	dto.	8746 , 9225	dto.
CPU-Zeit	958.96		976.69	
Kondition	80		79	
S(A)	0.16 ₁₀ -12		0.16 ₁₀ -12	
"C(A)",S(C(A))			-	
"C(A)",3(C(A))	0.8 ₁₀ -10,0.3 ₁₀ -10		0.8 ₁₀ -10,0.3 ₁₀ -10	
Beispiel 4b)				
Zyklen	15 (10)	10 (5)	15 (10)	10 (6)
Transform.	16776 16240	0571 0175	17104 16907	0560 0407
normred., ortho.	16736 , 16249	9571 , 9175 971 . 2	17104,16807 1797.15	9568,9403 1017.76
CPU-Zeit Kondition	1617 . 84 4321	432(4554)	4715	397 (4224)
S(A)	0.23 ₁₀ -13	0.28 ₁₀ -13	0.78 ₁₀ -13	0.67 ₁₀ -13
	10	10	10	10
"C(A)",S(C(A))	0.4 ₁₀ -13,0.1 ₁₀ -13	0.2 ₁₀ -13,0.9 ₁₀ -14	0.9 ₁₀ -13,0.9 ₁₀ -13	0.8 ₁₀ -13,0.8 ₁₀ -13
	L			L

	Varian	te 1	Variante	2
	ohne	mit	ohne	mit
Beispiel 5a)	Balancieru	ng	Balancierung	
Zyklen	50 abnormal		5o abnormal	
Transform. normred.,ortho.	1051,4533		1054,4493	
CPU-Zeit	46.07	dto.	48.51	dto.
Kondition	47		45	
S(A)	keine Blockd	iagonalgestalt	keine Blockdiag	gonalgestalt
С(Д) ,S(С(Д))	0.210-3,0.210-3		0.1 ₁₀ -4,0.1 ₁₀ -4	
Beispiel 5b)				
Zyklen	5o abnormal	50 abnormal	50 abnormal	50 abnormal
Transform. normred.,ortho.	1224,4815	1058,3997	1173,4687	1038,4124
CPU-Zeit	47.57	44.31	50	43.96
Kondition	460	46 (409	476	58 (484)
S(A)	keine Blockdiagonalgestalt		keine Blockdiagonalgestalt	
$\mathbb{C}(A)$, $\mathbb{S}(\mathbb{C}(A))$	0.110-3,0.110-3	0.910-6,0.910-6	0.210-4,0.210-4	٠ -
Beispiel 6a)				
Zyklen	50 abnormal		50 abnormal	
Transform. normred., ortho.	22045,122036		22578 , 122630	
CPU-Zeit	3775.22	dto.	3788.93	dto.
Kondition	216		210	
S(A)	keine Block	liagonalgestalt	keine Blockdiagonalgestalt	
 С(Д) ,S(С(Д))	1.5 , 1.5		1.7 , 1.7	
Beispiel 6b)				
Zyklen	37 Zeitüberschr.	50 abnormal	34 Zeitübersch	r. 50 abnormal
Transform.	20.100 11077	04==1 == 1==		04445 00505
normred., ortho.	28409 , 112384	21754,87602	27573 , 113660	21445,90393
CPU-Zeit	3210.5	3218.74	3213.25	3125.93
Kondition	8249	488 (9627)	7773	415 (8913)
S(A)		kdiagonalgestalt	keine Blockdiag	_
"C(♠)",S(C(♠))	0.02, 0.02	0.9 ₁₀ -3,0.9 ₁₀ -3	0.01 , 0.01	0.001,0.001

	Variante	e 1	Variante	2
	ohne	mit	}	mit
Beispiel 5a)	Balancierung		Balancierung	
Zyklen	7 (4)		8 (4)	
Transform. normred.,ortho.	292,283		328,320	
CPU-Zeit	8.8	dto.	12.87	dto.
Kondition	41		43	
S(A)	0.65 ₁₀ -14		0.74	
"C(A)",S(C(A))	0.3 ₁₀ -12,0.2 ₁₀ -13		0.3 ₁₀ -12,0.8 ₁₀ -13	
Beispiel 5b)				
Zyklen	14 (12)	8(5)	11 (9)	8 (5)
Transform. normred.,ortho.	607,579	340,318	490,480	356,346
CPU-Zeit	17.99	10.12	19.97	13.36
Kondition	359	45 (372)	386	49 (397)
S(A)	0.13 ₁₀ -9	0.74 ₁₀ -14	0.73 ₁₀ -11	0.78 ₁₀ -10
" С(Д) ", S(С(Д))	0.4 ₁₀ -10,0.4 ₁₀ -10	0.7 ₁₀ -14,0.1 ₁₀ -14	0.2 ₁₀ -10,0.2 ₁₀ -10	0.2 ₁₀ -9,0.2 ₁₀ -9
Beispiel 6a)				
Zyklen	13 (9)		12 (9)	
Transform. normred.,ortho.	13711 , 13753		12504, 12424	
CPU-Zeit	1425.66	dto.	1341.68	dto.
Kondition	262		212	
S(A)	0.15 ₁₀ -12		0.16 ₁₀ -12	
С(Д) ,S(С(Д))	0.6 ₁₀ -10,0.2 ₁₀ -10		0.6 ₁₀ -10,0.2 ₁₀ -10	
Beispiel 6b)				
Zyklen	22 (19)	12 (9)	23 (20)	14 (11)
Transform.				
normred.,ortho.	20970,20014	11573,10976	22344,22101	13542,13381
CPU-Zeit	2158.38	1179.01	2443.78	1462.37
Kondition	7371	472 (9356)	7379	347 (9042)
S(A)	0.54 ₁₀ -13	0.11 ₁₀ -10	0.18 ₁₀ -12	0.33 ₁₀ -11
"C(A)",S(C(A))	0.5 ₁₀ -13,0.2 ₁₀ -13	0.2 ₁₀ -12,0.2 ₁₀ -12	0.1 ₁₀ -12,0.1 ₁₀ -12	0.5 ₁₀ -12,0.5 ₁₀ -12

	Variant	e 1	Variante	2
	ohne mit		ohne	mit
Beispiel 7	Balancier	ıng	Balancierung	
Zyklen	16 (13)	12 (10)	15 (13)	12 (10)
Transform. normred.,ortho.	670,2708	513,2060	657,2637	501,2031
CPU-Zeit	20.23	15.48	25.92	19.72
Kondition	401	168 (358)	471	158 (353)
S(A)	0.16 ₁₀ -13	0.20 ₁₀ -13	0.21 ₁₀ -10	0.23 ₁₀ -13
" C(A) ", S(C(A))	0.2 ₁₀ -11,0.5 ₁₀ -12	0.1 ₁₀ -11,0.5 ₁₀ -12	0.210-9,0.710-9	0.9 ₁₀ -12,0.6 ₁₀ -12
Beispiel 8				
Zyklen	50 abnormal	50 abnormal	50 abnormal	50 abnormal
Transform. normred.,ortho.	2076,8361	2035,8851	2081,8411	2025,8175
CPU-Zeit	63.12	63.14	75.33	72.9
Kondition	347	172 (347)	371	167 (374)
S(A)	keine Blockdiagon	algestalt	keine Blockdiagonalgestalt	
∥C(A)∥,S(C(A))	4.44 , 4.44	2.77 , 2.77	3.82 , 3.82	2.69 , 2.69
Beispiel 9				
Zyklen	16 abnormal	14 abnormal	28 abnormal	14 abnormal
Transform.	10 00110111101	, , , delle i mar		. ,
normred.,ortho.	70,278	64,257	87,375	65,263
CPU-Zeit	2.28	2.02	3.75	2.50
Kondition	10.87	11.8(9.9)	10.4	11.1(9.6)
S(A)	0.01	0.02	2.49	0.005
"С(Д) ", S(С(Д))	0.6 ₁₀ -13,0.1 ₁₀ -13	0.5 ₁₀ -13,0.4 ₁₀ -13	0.7 ₁₀ -13,0.6 ₁₀ -1	3 0.3 ₁₀ -13,0.3 ₁₀ -13
Beispiel 10				
Zyklen	11 (7)		10 (7)	
Transform.				
normred.,ortho.	4538,18212		4646,18447	
CPU-Zeit	528.9		510.8	
Kondition	25	dto.	23	dto.
S(A)	0.61 ₁₀ -13	ļ	0.24 ₁₀ -12	
<pre>#C(A)#,S(C(A))</pre>	0.1,0-9,0.1,0-10		0.7,0-10,0.3,0-1	0
	10 10	1		

	Varian	te 1	Variant	e 2
	ohne	mit	ohne	mit
Beispiel 7	Balancier	ung	Balancieru	ng
Zyklen	14 (12)	12 (10)	14 (12)	12 (9)
Transform. normred.,ortho.	614,625	522 , 533	603,612	499,510
CPU-Zeit	18.6	15.59	25.1	20.15
Kondition	395	167 (392)	388	179 (396)
S(A)	0.75 ₁₀ -12	0.95 ₁₀ -11	0.24 ₁₀ -13	0.34 ₁₀ -13
∥С(Д)∥,S(С(Д))	0.2 ₁₀ -10,0.2 ₁₀ -10	0.2 ₁₀ -9,0.2 ₁₀ -9	0.9 ₁₀ -12,0.7 ₁₀ -12	0.2 ₁₀ -11,0.7 ₁₀ -12
Beispiel 8				
Zyklen	15 (13)	12 (9)	15 (12)	12 (9)
Transform. normred.,ortho.	659, 670	513, 526	631, 640	490,501
CPU-Zeit	16.7	15.5	26.37	19.54
Kondition	397	208 (448)	381	218 (450)
S(A)	0.41 ₁₀ -13	0.71 ₁₀ -13	0.38 ₁₀ -14	0.20 ₁₀ -13
<pre>#C(A)#,S(C(A))</pre>	0.2 ₁₀ -11,0.1 ₁₀ -11	0.1 ₁₀ -11,0.1 ₁₀ -11	0.5 ₁₀ -12,0.1 ₁₀ -12	0.1 ₁₀ -11,0.7 ₁₀ -12
Beispiel 9				
Zyklen	5 (2)	6 (2)	5 (2)	6 (2)
Transform.	30,30	35,35	30,30	35 , 35
CPU-Zeit	0.94	1.06	1.22	1.44
Kondition	9.05	14.4(11.7)	10.9	12.8(10.8)
S (A)	0.1910-9	0.24 ₁₀ -15	0.81 ₁₀ -11	0.15 ₁₀ -14
∥C(A)∥,S(C(A))	0.110-8,0.110-8	0.4 ₁₀ -13,0.2 ₁₀ -14	0.4 ₁₀ -10,0.4 ₁₀ -10	0.1 ₁₀ -13,0.1 ₁₀ -13
Beispiel 10				
Zyklen	10 (6)		9 (6)	
Transform.	(0)			
normred.,ortho.	7757,8548		7821,8021	
CPU-Zeit	807.75	dto.	774.01	dto.
Kondition	26		25	
S(A)	0.2010-12		0.18 ₁₀ -12	
∥C(A)∥,S(C(A))	0.110-9,0.610-10		0.9 ₁₀ -10,0.4 ₁	o ⁻¹⁰
			L	

Zufallsmatrizen	Variante	e 1	Variant	e 2
Mittelwerte	ohne	mit	ohne	mit
Dimension 10	Balancier	ıng	Balancieru	ng
Zyklen	8	8	7	7
Transform. normred.,ortho.	75 , 300	74 , 297	72, 288	72, 288
CPU-Zeit	1.85	1.85	2.55	2.54
Kondition	45.3	44.8(45.4)	44.0	43.9(44.2)
s (A)	0.33 ₁₀ -10	0.33 ₁₀ -10	0.86 ₁₀ -11	0.37 ₁₀ -12
Dimension 20				
Zyklen	11	11	11	11
Transform. normred.,ortho.	470,1881	470 , 1881	470,1880	470 , 1880
CPU-Zeit	13.09	13.18	17.47	17.53
Konditon	105	105 (105)	108	108 (108)
S(A)	0.67 ₁₀ -10	0.67 ₁₀ -10	0.28 ₁₀ -10	0.28 ₁₀ -10

Von den 10 Zufallsmatrizen der Dimension 10 werden 3 Matrizen von allen Varianten des Verfahrens I nicht blockdiagonalisiert, weil sie 4 reelle Eigenwerte haben; die o.a. Mittelwerte beziehen sich nur auf die restlichen 7 Matrizen.

Bei den 10 Zufallsmatrizen der Dimension 20 versagt das Verfahren I mit allen Varianten in 6 Fällen, bei denen mehr als zwei reelle Eigenwerte vorliegen; in zwei Fällen treten 6 und in 4 Fällen 4 reelle Eigenwerte auf. Für die restlichen 4 Fälle , in denen 2 reelle Eigenwerte vorliegen, sind die Mittelwerte oben angeführt.

Für die 10 Zufallsmatrizen der Dimension 40 und 5 Zufallsmatrizen der Dimension 80 sind keine Mittelwerte der Varianten von Verfahren 1 aufgeführt, weil diese Verfahren in keinem Fall normal terminieren. Bei den 10 Matrizen der Dimension 40 treten in je 2 Fällen 4 bzw. 8 und in den restlichen 6 Fällen 6 reelle Eigenwerte auf, während bei den 5 Matrizen der Dimension 80 in 2 Fällen 6 und in je einem Fall 8, 10 bzw. 12 reelle Eigenwerte vorliegen.

Die Varianten von Verfahren II liefern in allen Fällen eine "fast" normale und blockdiagonale Endmatrix, aus der die Eigenwerte abgelesen werden können.

Zufallsmatrizen	llsmatrizen Variante 1		Variant	e 2
Mittelwerte	ohne	mit	ohne	mit
Dimension 10	Balancierun	g	Balancier	ıng
Zyklen	7	7	7	7
Transform. normred., ortho.	72, 72	71, 71	69, 69	69, 68
CPU-Zeit	1,95	1.93	2.67	2.66
Kondition	43.3	42.1(42.5)	42.97	42.2(42.8)
S(A)	0.60 ₁₀ -13	0.65 ₁₀ -13	0.6310-10	0.44 ₁₀ -10
Dimension 20				
Zyklen	10	10	9	9
Transform. normred.,ortho.	413,411	413,411	401 400	401 400
CPU-Zeit	12.26	12.34	15.76	15.8
Kondition	100	100(100)	101	101 (101)
S(A)	0.54 ₁₀ -10	0.54 ₁₀ -10	0.32 ₁₀ -10	0.32 ₁₀ -10
Dimension 40				
Zyklen	12	12	11	11
Transform. normred.,ortho.	2167,2148	2167,2148	2050,2043	2050,2043
CPU-Zeit	99.14	98.18	111.97	110.39
Kondition	281	281 (281)	274	274 (274)
S(A)	0.44 ₁₀ -10	0.44 ₁₀ -10	0.2610-10	0.2610-10
Dimension 80				
Zyklen	14	14	14	14
Transform. normred.,ortho.	10235,10183	10234,10183	10178,10146	10178, 10146
CPU-Zeit	832.58	827.82	904.58	901.99
Kondition	905	905 (905)	921	921 (921)
S(A)	0.71 ₁₀ -10	0.7110-10	0.3810-12	0.3810-12

Für Beispiel 9 zeigt das Verfahren I Konvergenz gegen Normalität, ohne Blockdiagonalgestalt zu erzeugen; die starke Schwankung der Zyklen-Zahl bei den verschiedenen Varianten deutet an, daß es weitgehend vom Zufall abhängt, wann der Zustand erreicht ist, bei dem keine Transformation mehr durchgeführt wird. Dagegen benötigt das Verfahren II noch 2 Zyklen weniger als das komplexe Verfahren von Eberlein (vergl.[38]).

Für Beispiel 9 benötigt das Verfahren II jeweils einen Zyklus weniger als Verfahren I; trotzdem aber eine erheblich größere Zahl von Transformationen und mehr Rechenzeit. Eine genaue Analyse der Verfahrens-Statistik zeigt hier, daß bei Verfahren I vom zweiten bis zum sechsten Zyklus einschließlich, scwohl orthogonal wie normreduzierend, konstant 540 von 990 möglichen Transformationen durchgeführt werden - das sind ca. 54%; nach dem sechsten Zyklus werden es langsam weniger. Demgegenüber werden bei Verfahren II vom zweiten Zyklus an die volle Zahl von 990 Transformationen durchgeführt. Dieses bei Verfahren I beobachtete überraschende Phänomen liegt in der speziellen Struktur der schwach besetzten Matrix begründet, die offenbar von den verwendeten Transformationen (fast) nicht gestört wird, so daß viele Pivotblöcke so klein bleiben, daß dort keine Transformation erforderlich ist.

Soweit für die Beispiele die exakten Eigenwerte bekannt sind, können die berechneten Eigenwerte auf ihre Genauigkeit überprüft werden. Dabei zeigt sich, daß in beiden Verfahren die Genauigkeit der berechneten Eigenwerte im Vergleich zu anderen normreduzierenden Jacobiähnlichen Verfahren als sehr zufriedenstellend zu bezeichnen ist; der relative Fehler liegt in der Größenordnung von $K(R) \epsilon_0^{*}$, so daß die berechneten Eigenwerte bei gut konditionierten Problemen auf 14 Dezimalstellen exakt sind. Tritt ein doppelter reeller, defektiver Eigenwert auf, so ist dieser erwartungsgemäß nur auf etwa 7 Dezimalstellen exakt, während für die anderen Eigenwerte das oben gesagte zutrifft.

^{*)} Vergl. (4.3.5) und Fußnote S.100.

ANHANG

A) Konstanten- und Schrankenverzeichnis (in Klammern ist angegeben, wo die Konstanten definiert sind; es gilt stets $\nu \in \{2,3\}$)

$$a = 1.1336$$
, $b = 1.1004$, $c = \frac{10}{19}$ (Satz 2.2.1)

$$a_2 = \frac{13}{9}$$
, $a_3 = \frac{7}{4} = 1.75$ (Lemma 2.1.13)

$$b_{r}^{(v)} = 0.1871(\sqrt{2} + \tilde{K}_{v})^{2},$$

$$b_{s}^{(v)} = 0.2947(\sqrt{2} + \tilde{K}_{v})^{2},$$
(Lemma 2.2.5)

$$b_{rs}^{(v)} = \frac{a}{2} \left(\sqrt{2} + \widetilde{K}_{v}\right)$$
 (Lemma 2.2.3)

$$c_2 = \frac{244}{6^2} \|A\|^2$$
 , $c_3 = \frac{367}{8^2} \|A\|^2$ (Lemma 3.2.4)

$$C^{(v)} = p \left[\left(0.24 + 0.71 \cdot 10^{-5} \frac{\|A\|}{\delta^{-}} \right) \cdot 1.063^{p} K^{(v)} + 0.12 \cdot w_{v}^{2} \frac{\|A\|^{4}}{(\delta^{-})^{4}} \right]$$
(Lemma 3.2.19)

$$\delta = \min\{|\lambda_{i} - \lambda_{j}| | 1 \le i < j \le n \}$$

$$\delta^{-} = \min\{|\operatorname{Im} \lambda_{i} - \operatorname{Im} \lambda_{j}| | 1 \le i < j \le n \}$$
(Def. 1.1.29)

$$\widetilde{\delta} = \frac{1}{3} \min\{|\nu_i - \nu_j| | 1 \le i < j \le p , \nu_j \ge 0 , \pm i \nu_j \text{ EWe von } A^-\}$$
(Satz 2.2.1)

$$d_{v} = \frac{\tilde{K}_{v}}{\sqrt{3}} + \left(2.941 \frac{\|A\|}{\delta^{-}} + 0.8101\right) \frac{16(\sqrt{2} + \tilde{K}_{v})^{2}}{5 \delta^{-}} \|A\| \quad \text{(Lemma 3.2.3)}$$

$$e_{v} = 2\left(2\|\mathbf{A}\| + \frac{\delta^{2} d_{v}}{4096\|\mathbf{A}\|}\right) \frac{d_{v}}{\|\mathbf{A}\|}$$
 (Lemma 3.2.4)

$$\varepsilon_{0} = \frac{\delta}{16\sqrt{p-1}} \|A\| , \quad \varepsilon_{1} = \frac{\delta^{2}}{64\|A\|^{2}}$$

$$\varepsilon_{3} = \left[4\|A\|^{4} \left[\frac{9710}{\delta^{6}} \|A\|^{2} + \frac{74}{\delta^{4}}\right]\right]^{-1/2}$$
(Satz 2.1.8)

$$\varepsilon_2 = \frac{3 \delta^2}{512(4 + \sqrt{2}) \|\Delta\|^2} < \varepsilon_1$$
 (Satz 2.1.9)

$$\varepsilon_{0}^{-} = \frac{\left(\delta^{-}\right)^{2}}{64 \left\|\Delta\right\|^{2}}$$
 (Lemma 3.2.10)

$$\varepsilon_{v-1}^{-} = \frac{\delta^{-}}{4 \|A\| (a_{v} + 2\tilde{K}_{v} + \tilde{K}_{v}^{2})^{1/2}} < \frac{\delta^{-}}{4 \|A\|}$$
 (Lemma 2.1.13)

$$\varepsilon_{\nu+1}^{-} = \frac{\delta^{-}}{4 \|\mathbf{A}\| \sqrt{H_{\nu}}} < \varepsilon_{\nu-1}^{-}$$
 (Korollar 3.2.2)

$$\varepsilon_{\nu+3}^{-} = \frac{(\delta^{-})^{2}}{16 \|A\|^{2}} \left(\frac{(P^{2}-1)(H_{\nu}-1)}{P^{2}h_{\nu}} \right)^{1/2}$$
 (Lemma 3.2.10)

$$f = 2.0626 \left[\frac{5}{6} + \frac{c_2}{2} \right]$$

$$f_2 = f + 0.5708 b_{rs}^{(2)} (c_2 + (1 + \tilde{K}_2)^2)$$

$$f_3 = 2.0626 + 0.5708 b_{rs}^{(3)} (c_3 + (1 + \tilde{K}_3)^2)$$
(Lemma 3.2.6)

$$g_{v} = 6.4676 b_{rs}^{(v)} \frac{\|A\|}{6}$$
 (Lemma 3.2.1)

$$h_{v} = \frac{2e_{v}\delta^{2}}{64\|A\|^{2}} \left(\sqrt{2} + \frac{e_{v}\delta^{2}}{1024\|A\|^{2}} \right)$$
 (Lemma 3.2.5)

$$H_{v} = a_{v} + (g_{v} + 2.0364 \tilde{K}_{v} + 2.0627)^{2} + (g_{v} + 0.0263 \tilde{K}_{v} + 2.0627)^{2}$$
(Lemma 3.2.1)

$$\widetilde{K}_{2} = \frac{128\sqrt{2}}{\sqrt{3}} \|A\|^{2}$$
, $\widetilde{K}_{3} = \frac{\sqrt{6} 64}{8^{2}} \|A\|^{2}$ (Lemma 2.1.12)

$$K_{r}^{(v)} = 2(a_{v}^{p-r} K_{r-1}^{(v)} + Q_{r}^{(v)}), K_{o}^{(v)} = 0$$

$$K^{(v)} = \sqrt{K_{p-1}^{(v)}} \max \left\{ \frac{1}{16} H_{v}^{p}, \frac{1}{4} (\sqrt{H_{v}})^{p} w_{v} \right\}$$
(Satz 3.2.12)

$$L = 1.0916$$

$$m_{V}^{(k)} = \left(0.5835 \frac{\tilde{K}_{V}}{\|A\|} + \frac{16 b_{rs}^{(V)}}{5 \delta^{-}}\right) S(A_{k})$$

$$P = \frac{p(p-1)}{2}$$
(Lemma 3.2.9)

$$Q_{r}^{(v)} = 2 L^{2(p-r)} \left[(p-r) \frac{32}{49(\delta^{-})^{2}} (f_{v} + g_{v})^{2} + (p-r-1)^{2} \left(0.5835 \frac{\tilde{K}_{v}}{\|A\|} + \frac{16 b_{rs}^{(v)}}{5 \delta^{-}} \right)^{2} \right] \|A\|^{2}$$
(Satz 3.2.12)

$$w_{\nu} = 1 + \frac{h_{\nu}(H_{\nu}^{P} - 1)}{16(H_{\nu} - 1)}$$
 (Korollar 3.2.11)

- B) ALGOL 60-Programm für Verfahren II, Variante 2
 - 1) Beschreibung der Ein-/Ausgabe-Variablen:
 - N Dimension der Matrix A (Eingabe)
 - A Matrix A, von der die Eigenwerte berechnet werden sollen, enthält nach Rücksprung die transformierte Matrix A (Ein-/Ausgabe)
 - L Anzahl der benötigten Iterationszyklen (Ausgabe)

MNORM Anzahl der normreduzierenden Transformationen (Ausgabe)

MORTH Anzahl der orthogonalen Transformationen (Ausgabe)

NAO $S(\widetilde{A})$ nach Rücksprung (Ausgabe)

LINKS Schalter zur Steuerung, ob die (inverse)
RECHTS Transformationsmatrix akkumuliert werden soll(en)(Eingabe)

- T akkumulierte Transformationsmatrix (Ausgabe)
- TI Inverse der akkumulierten Transformationsmatrix (Ausgabe)
- Z benötigte CPU-Zeit in (1/100) Sek. (Ausgabe)

FMARKE Sprungziel im Fehlerfall (Eingabe)

- F Fehlercode (Ausgabe)
- LIM zur Verfügung gestellte CPU-Zeit in (1/100) Sek. (Eingabe)

Mögliche Fehlercodes sind:

- 1 N ist ungerade,
- 2 50 Zyklen durchgeführt, ohne daß Blockdiagonalgestalt von A erreicht ist,
- Während des letzten Zyklus wurde keine Transformation durchgeführt,

- die zur Verfügung gestellte CPU-Zeit ist verbraucht.
- 2) Externe Prozeduren:
- GAUSS 2 (N,A,X,FM) berechnet die Lösung eines linearen Gleichungssymstems der Dimension N mit vollständiger Pivotisierung, wobei die rechte Seite als (N+1)-te Spalte in A gespeichert ist. Rücksprung über FM, falls das System singulär ist.
- KOMMUTATOR (P,A,SCA, NCA,NCD) berechnet von der Matrix A der Dimension 2P die Werte SCA = S(C(A)), NCA = ||C(A)|| und NCD = ||C(D)||.
- KOND (N,T,TI) berechnet aus der Transformationsmatrix T der Dimension N und ihrer Inversen TI die Kondition von T in der Spaltensummennorm.
- MINIMUM (X,B1,B2,B3,B4,B0,PX) berechnet die Minimalstelle X des Polynoms $P(X)=B4 \cdot X^4 + B3 \cdot X^3 + B2 \cdot X^2 + B1 \cdot X + B0$ nach § 1.2 mit |X|P(X) = min! und den Wert PX = P(X)-B0.
- NORM (P,A,NA,NAO) berechnet von der Matrix A der Dimension 2P die Euklidische Norm NA = ||A|| und NAO = S(A).
- RMIN (A,B) berechnet das Minimum von A und B.

Die Prozeduren PUTDATA, PUTLIST und PUTRDAT dienen der formatierten Ausgabe.

```
*PROCEDURE* EIGEN7(N) MATRIX:(A) ITERATIONEN:(L)
                                                                           000
 0
                         TRANSFORMATIONEN: (MNORM, MORTH) S: (NAO)
                                                                           000
 1
                                                                           000
                         EIGENVEKTOREN: (LINKS, RECHTS)
 1
                                                                           000
                         TRANSFORMATIONSMATRIZEN: (TI. T)
 1
                         VERBRAUCHTE RECHENZEIT: (Z)
                                                                           000
 1
                         FEHLERRUECKSPRUNG: (FMARKE)
                                                                           000
 1
                                                                           000
                         FEHLERCODF:(F)
 1
                                                                           000
                         ZFITLIMIT: (LIM);
 1
                                                                           000
 1
                                                                           001
 1
    1
                                                                          *001
 1
    ×
                                                                          *001
            UNTERPROGRAMM EIGENWERT 7 ZUM BERECHNEN DER EIGENWERTE
 1
            BELIEBIGER REELLER MATRIZEN.
                                                                          *001
 1
    ¥
                                                                          *001
 1
    ×
                                                                          *001
 1
    ×
            VERSION 5 VOM 08. 02. 1982
                                                                          *001
 1
                                                                          *001
    ¥
 1
                                                                          *001
            VERFASSER: HANS JOACHIM WENZEL, FB. MATHEMATIK
 1
    ×
                                                                          *002
 1
            LETZTE BEARBEITUNG : 11. 11. 1982
                                                                          *002
 1
                                                                          *002
            VERSION 5.6
 1
                                                                          *002
 1
    1
                                                                           002
 1
                                                                           002
 1
                                                                           002
    .VALUE! N;
 1
                                                                           002
 2
    "INTEGER" No Lo Zo Fo LIMo MNORMo MORTHS
    "REAL" NAO;
                                                                           002
 3
    *BOOLEAN* LINKS, RECHTS;
                                                                           003
                                                                           003
 5
    *LABEL* FMARKE;
                                                                           003
    *ARRAY* A, TI, T;
 6
                                                                           003
 7
    *BEGIN*
                                                                           003
 8
                                                                           003
       *PROCEDURE* GAUSS2; *CODE*;
8
                                                                           003
       *PROCEDURE* KOMMUTATOR; *CODE*;
10
                                                                           003
       *REAL **PROCEDURE* KOND;
                                *CODE *;
12
                            *CODE*;
                                                                           003
       *PROCEDURE* MINIMUM;
14
                                                                           003
       *PROCEDURE* NORM#
                          *CODE*;
16
       *PROCEDURE * PUTDATA; *CODE *;
                                                                           004
18
                                                                           004
       *PROCEDURE* PUTLIST; *CODE*;
20
                                                                           004
                             .CODE.:
       *PROCEDURE* PUTRDAT;
22
                                                                           004
24
       *REAL **PROCEDURE * RMIN;
                                *CODE *3
                                                                           004
26
       "INTEGER"P, P1, R, S, R1, MM, Z1, Z2, MMM, MD, MDIAG, I, J,
                                                                           004
26
                                                                           004
               NSY, NSCHIE, NT4, NT3, NV4, NV3, NP;
27
                                                                           004
       *REAL * EP, EPS, NA, K, ASUM, SSUM, SCA, NCA, NCD, LAMBDA;
27
                                                                           004
       *BOOLEAN* ITSTEHT, SYMM, SCHIEF, AOKLEIN;
28
                                                                           004
29
       *ARRAY* M(/1:4, 1:5/), X(/1:4/), D(/1:N/);
                                                                           005
30
                                                                           005
30
                                                                           005
30
                                                                           005
30
                                                                           005
       *PROCEDURE * EBERLEIN;
                              *BEGIN*
30
```

```
005
32
   32
                                                         ×nn5
32
          AUF DEN DIAGONALBLOECKEN R = S = 1 (1) P WIRD
                                                         *005
32
          JE EINE EBERLEIN - TRANSFORMATION ZUR NORMALISIERUNG
                                                         *005
32
                                                         *006
         DIESER BLOECKE DURCHGEFUEHRT
32
                                                         *006
32
  32
                                                          006
32
                                                          006
32
       "INTEGER" II, K, M, I, SIG, M1, K1;
       "REAL" AII, AIK, AKI, G, H, NDA, AIH, AHI, AKM, AMK, AKK,
                                                          006
33
          AMM, AIKO, AIMO, H2, CX, SX, COT2X, COTX, C2X, S2X, TH, CH, SHOO6
34
          S SC C D E AMIQ AKIQ CKKO CMM CMKO C15
                                                          006
34
                                                          006
34
       *BOOLEAN* TRANS#
                                                          006
35
       *FOR* II:=1 *STEP* 1 *UNTIL* P *DO**BEGIN*
                                                          007
35
                                                          007
37
          H == II×2#
                                                          007
38
          K := M-1;
          TRANS := "FALSE";
                                                          007
39
                                                          007
40
  40
                                                         *007
40
         BERECHNEN DES ROTATIONSPARAMETERS X
                                                         *007
  ×
40
                                                         *007
         UND DES SCHERUNGSPARAMETERS Y
40
                                                         *007
40
40
  800
40
            AKM := A(/K_2 M/) 3 \qquad AMK := A(/M_2 K/) 3
                                                          800
40
                                                          008
            AKK := A(/K_B K/); AMM := A(/M_B M/);
42
            G := H := H2 := AIKO := AINO := AMIO := AKIO := O;
                                                          800
44
                                                          800
            M1 := M - 1;
45
                                                          008
            K1 := K-1;
46
            *FOR* I:=1 *STEP* 1 *UNTIL* K1,
                                                          800
47
                   K+1 'STEP' 1 'UNTIL' M1,
                                                          008
48
                   M+1 *STEP* 1 *UNTIL* N *DO**BEGIN*
                                                          008
48
              AIK := A(/I + K/); \quad AKI := A(/K + I/);
                                                          009
49
                                                          009
              51
                                                          009
              AIKO := AIK**2 + AIKO;
53
                                                          009
54
              AIMO := AIM**2 + AIMOS
                                                          009
55
              AMIQ := AMI**2 + AMIQ;
56
              AKIQ := AKI**2 + AKIQ;
                                                          009
                                                          009
              H := H + AKI*AMI - AIK*AIM;
57
           *END*E DER SCHLEIFE UEBER I
                                                          009
58
                                                          009
59
           G := AKIO + AIKO + AIMO + AMIO;
            H2 := AKIO + AIMO - AIKO - AMIO;
                                                          009
60
                                                          010
            D := AKK - AMM;
61
                                                          010
62
            C := AKM + AMKs
                                                          010
63
            E := AKM - AMKS
                                                          010
64
   64
                                                         *010
64
         BERECHNEN DES KOMMUTATORS C(II, II) VON A
                                                         *010
64
                                                         *010
64
   64
                                                          010
64
```

```
CKK := AMK**2 - AKM**2 + AIKO - AKIO3
                                                             011
64
             CMM := AKM**2 - AMK**2 + AIMO - AMIQ3
                                                             011
65
             CMK := AKK*AMK + AKM*AMM - AKK*AKM - AMK*AMM + HJ
                                                             011
66
                                                             011
             C1 := CKK - CMM;
67
                                                             011
68
   68
                                                             *011
68
                                                            *011
          BERECHNEN DER ROTATIONSMATRIX R
68
                                                            *011
68
    68
                                                             012
68
             *IF * ABS(C1) <= EP
                                                             012
68
             *THEN ** BEGIN ** COMMENT * R IST EINHEITSMATRIX ;
                                                             012
69
                                                             012
               CX := 1; SX := 0
70
                                                             012
             *END *E DES THEN - ZWEIGES
72
                                                             012
             *ELSE**BEGIN*
73
                                                             012
75
                TRANS := "TRUE";
                COT2X := 2*CMK/C1;
                                                             012
76
                SIG := "IF" COT2XCO "THEN" -1 "ELSE" 1;
                                                             012
77
               COTX := COT2X + SIG * SORT(1+COT2X**2);
                                                             012
78
                SX := SIG / SQRT(1 + COTX**2);
                                                             013
79
                                                             013
               CX := SX \times COTX
80
                                                             013
             *END *E DES ELSE - ZWEIGES;
81
                                                             013
82
    82
                                                            *013
82
                                                            #013
82
    ×
          BERECHNEN DER SCHERUNGSMATRIX S
                                                            *013
82
    82
                                                             013
82
                                                             014
             C2X := CX**2 - SX**2*
82
                                                             014
             S2X := SX*CX*2;
83
                                                             014
             D := D \times C \times X + C \times S \times X
84
                                                             014
             H == H*C2X*2 - H2*S2X;
85
             TH := (E*D - H/2) / (G + (E**2 + D**2)*2);
                                                             014
86
                                                             014
87
             *IF * ABS(TH) <= EP
             "THEN "BEGIN " COMMENT" S IST EINHEITSMATRIX;
                                                             014
88
                CH := 1; SH := 0
                                                             014
89
             *END *E DES THEN - ZWEIGES
                                                             014
91
                                                             014
             'ELSE''BEGIN'
92
                                                             015
                TRANS := *TRUE*;
94
                CH := 1 / SQRT(1 - TH**2);
                                                             015
95
                  SH := CH XTH
                                                             015
96
             *END *E DES ELSE - ZWEIGES;
                                                             015
97
                                                             015
98
                                                             015
             *IF* TRANS "THEN**BEGIN*
98
                                                             015
100
    100
                                                            *015
100
           BERECHNEN DER TRANSFORMATIONSMATRIX T
                                                             *015
100
                                                             *016
100
    100
                                                             016
100
                                                             016
               MD := MD + 1;
100
                                                             016
                S := SH*SX; SC := SX*CH;
101
```

AST ALGOL COMPILER OF DELFT RELEASE OF 1/ 2/1977

```
AMK := -CX*SH; AMM := CX*CH;
                                                                                                                                                                                                                         016
103
                                                        AKK == AMM + SI
105
                                                                                                                                                                                                                         016
106
                                                        AMM := AMM - SI
                                                                                                                                                                                                                         016
                                                        AKM == AMK - SC#
107
                                                                                                                                                                                                                         016
108
                                                         AMK := AMK + SC;
                                                                                                                                                                                                                         016
109
                                                                                                                                                                                                                         017
109
              «СОММЕNТ» яккии и киминичи и тиминичи и тим
109
                                                                                                                                                                                                                      *017
                                     MULTIPLIKATION MIT T**-1 VON LINKS
109
                                                                                                                                                                                                                       *017
109
                                                                                                                                                                                                                       *017
109
              017
109
109
                                                        *FOR* I:=1 *STEP* 1 *UNTIL* N *DO**BEGIN*
                                                                                                                                                                                                                         017
                                                                 AKI := A(/K_B I/); AMI := A(/M_B I/);
111
                                                                                                                                                                                                                         017
                                                                 A(/K, I/) := AMM*AKI - AKM*AMI;
113
                                                                                                                                                                                                                         017
                                                                 A(/Ms I/) := AKK*AMI - AMK*AKIS
114
                                                                                                                                                                                                                         018
115
                                                                 *IF* LINKS *THEN**BEGIN*
                                                                                                                                                                                                                         018
117
                                                                                                                                                                                                                         018
117
              117
                                                                                                                                                                                                                      *018
117
                                     BERECHNEN DER LINKSEIGENVEKTOREN
                                                                                                                                                                                                                      *018
117
                                                                                                                                                                                                                      *018
117
              117
                                                                                                                                                                                                                         018
                                                                         AKI := TI(/K, I/); AMI := TI(/M, I/);
117
                                                                                                                                                                                                                         018
119
                                                                         TI(/K = I/) == AMM*AKI - AKM*AMI;
                                                                                                                                                                                                                         019
120
                                                                         TI(/M. I/) := AKK*AMI - AMK*AKI
                                                                                                                                                                                                                         019
                                                                 "END"
                                                                                                                                                                                                                         019
121
                                                        *END*E DER MULTIPLIKATION VON LINKS;
122
                                                                                                                                                                                                                         019
123
                                                                                                                                                                                                                         019
123
              «COMMENT» яччнининин кинининининининин кининин кининин кининин кинин ки
123
                                                                                                                                                                                                                      *019
                                       MULTIPLIKATION HIT T VON RECHTS
123
                                                                                                                                                                                                                      ×019
123
                                                                                                                                                                                                                      *019
123
              123
                                                                                                                                                                                                                         020
123
                                                       *FOR* I:=1 *STEP* 1 *UNTIL* N *DO**BEGIN*
                                                                                                                                                                                                                         020
                                                                 AIK := A(/I, K/); AIM := A(/I, M/);
125
                                                                                                                                                                                                                         020
                                                                 A(/I, K/) := AKK*AIK + AMK*AIM;
127
                                                                                                                                                                                                                         020
                                                                 A(/I, M/) := AKM×AIK + AMM*AIM;
128
                                                                                                                                                                                                                         020
129
                                                                                                                                                                                                                         020
129
                                                                 *IF* RECHTS *THEN**BEGIN*
                                                                                                                                                                                                                         020
131
                                                                                                                                                                                                                         020
131
              131
                                                                                                                                                                                                                      *020
131
                                       BERECHNEN DER RECHTSEIGENVEKTOREN
                                                                                                                                                                                                                      *021
131
                                                                                                                                                                                                                      *021
131
              131
                                                                                                                                                                                                                         021
131
                                                                        AIK := T(/I \rightarrow K/); AIM := T(/I \rightarrow M/);
                                                                                                                                                                                                                         021
133
                                                                         I(/I_s K/) := AKK*AIK + AMK*AIMs
                                                                                                                                                                                                                         021
                                                                         T(/I, H/) := AKM*AIK + AMM*AIM
134
                                                                                                                                                                                                                         021
135
                                                                 "END"
                                                                                                                                                                                                                         021
136
                                                       PEND E DER MULTIPLIKATION VON RECHTS
                                                                                                                                                                                                                         021
                                    *END*E EINER TRANSFORMATION
137
                                                                                                                                                                                                                         021
```

INCRONRO+SOURCE L

```
022
          *END*E DER SCHLEIFE UEBER II
138
                                                                        022
       *END*E DER PROZEDUR EBERLEIN;
139
                                                                        022
140
                                                                        022
140
                                                                        022
140
                                                                        022
140
                                                                        022
140
       *PROCEDURE* BLOCKTRANS7; *BEGIN*
                                                                        022
142
    142
                                                                       *022
142
             FUER DIE INDIZES R = 1 (1) P-1 s = R+1 (1) P
                                                                       *023
142
    ×
             WIRD EINE BLOCKTRANSFORMATION ZUM MINIMIEREN DER
                                                                      * 023
142
                                                                       *023
             EUKLIDISCHEN NORM DER MATRIX A VORGENOMMEN ...
142
            ANSCHLIESSEND EINE ORTHOGONALE TRANSFORMATION ZUM
                                                                       *023
142
                                                                       *023
            DIAGONALISIEREN DES SCHIFSYMMETRISCHEN BEZN.
142
                                                                       *023
142
            SYMMETRISCHEN TEILS DER MATRIX A .
                                                                       *023
142
                                                                       *023
            VERSION 5 VOM 08. 02. 1982
142
                                                                       *023
142
                                                                       *023
            LETZTE BEARBEITUNG : 11. 11. 1982
142
                                                                       *024
142
             VERSION 5.6
                                                                       *024
142
    142
                                                                        024
142
                                                                        024
          PARRAY! EW(/1:4/);
142
                                                                        024
143
          *INTEGER* R2, R21, R22, S2, S21, S22, K, J, S1, L, J1, I, SIG,
                                                                        024
143
                   U. V. W. SP. NITE
                                                                        024
144
                                                                        024
144
          *REAL* C11, C12, C21, C22, B11, B12, B22, ALPHA,
                                                                        024
144
                 C11S, C12S, C21S, C22S, B11S, B12S, B22S, A11S, A12S, A22S, 025
145
                XT11, XT12, XT21, XT22, AR11, AR22, AS11, AS22,
                                                                        025
145
                x11, x12, x21, x22, AR21, AR12, AS12, AS21, ARS11, ARS12,
                                                                        025
145
                                                                        025
                ARS21, ARS22, ASR11, ASR12, ASR21, ASR22, C120, D120,
145
                C210, D210, C1D1, C1D2, C2D1, C2D2, H, MJJ, H1, H2, H3, H4,025
145
                H5, H6, H7, H8, NCRS, F2, F1, F1A, Y11, Y12, Y21, Y22,
                                                                        025
145
                                                                        025
                N1, N2, SP1, SP2, SP3, SP4, SP5,
145
                                                                        025
145
                KA, WUR, TAU1, TAU2, PLU, MAL, QUA, C.
                KO41, KO42, KO43, KO44, KO31, KO32, KO33,
                                                                        025
145
                                                                        025
                A120, A130, A140, A230, A240, A340,
145
                                                                        026
                Z1, Z2, Z3, Z4, Z5, Z6, PE, QU, DISKR, X0, X1, X2,
145
                                                                        026
                P3, p, RG, A11, A123 A13, A14, A22, A23, A24,
145
               A33, A34, A44, A21, A31, A32, A41, A42, A43,
                                                                        026
145
                AF11, AF12, AF13, AF14, AF22, AF23, AF24,
                                                                        026
145
                                                                        026
                AF33, AF34, AF44, TAU, SHIFT, EPSI, Y1, Y2,
145
                                                                        026
                VE31, VE32, VE33, VEK31, VEK32, VEK33,
145
                VE41, VE42, VE43, VE44, VEK41, VEK42, VEK43, VEK44,
                                                                        026
145
                D1, D2, D3, SPUR, U1, U2, U3, U12, U13, U23,
                                                                        026
145
                NQ1, NQ2, NQ3, NQ4, AR1I, ARI, L1, L2,
                                                                        026
145
                NO1, NO2, NO3, NO4,
                                                                        026
145
                IMLR, RELR1, RELR2, IMLS, RELS1, RELS2, DRE, DIM;
                                                                        027
145
                                                                        027
145
         *BOOLEAN* FX2, DREI, DOP, VZW, MEHRFACH, WIEDERHOLUNG, ITER;
                                                                        027
145
                                                                        027
146
                                                                        027
         *FOR* R:=1 *STEP* 1 *UNTIL* P1 *DO**BEGIN*
146
```

```
027
148
            R1 := R - 1;
                                                                    027
148
149
            R2 := R * 2;
                                                                    027
150
            R21 := R2 - 1;
                                                                    027
151
            R22 := R2 - 2#
                                                                    027
152
                                                                    028
            *FOR* S:=R+1 *STEP* 1 *UNTIL* P *DO**BEGIN*
                                                                    028
152
                                                                    028
154
               S2 := S * 2;
                                                                    028
154
               S21 := S2 - 1;
                                                                    028
155
                                                                    028
               S22 := S2 - 2;
156
                                                                    028
157
157
              AS11 := A(/2*S-1) 2*S-1/);
                                                                    850
              AS12 := A(/2*S-1) 2*S/);
                                                                    028
158
              AS21 := A(/2*S, 2*S-1/);
                                                                    028
159
              AS22 := A(/2*S) 2*S/);
                                                                    029
160
                                                                    029
161
                                                                    029
161
              AR11 := A(/2*R-1, 2*R-1/);
                                                                    029
              AR12 := A(/2*R-1) 2*R/)
162
                                                                    029
              AR21 := A(/2*R) 2*R-1/);
163
                                                                    029
              AR22 := A(/2*R*2*R/)
164
                                                                    029
165
                                                                    029
              ASR11 := A(/2×S-1, R21/);
165
              ASR12 := A(/2*S-1, R2 /);
                                                                    029
166
              ASR21 := A(/2*S , R21/);
                                                                    029
167
                                                                    030
              ASR22 := A(/2*S + R2 /);
168
                                                                    030
169
                                                                    030
169
              ARS11 := A(/R21, 2*S-1/);
              ARS12 := A(/R21 + 2*S /) +
                                                                    030
170
                                                                    030
              ARS21 := A(/R2 + 2*S-1/)
171
              ARS22 := A(/R2 + 2*S /);
                                                                    030
172
                                                                    030
173
              C11S := C12S := C21S := C22S := A11S := A12S := A22S :=
173
                                                                    030
174
                                                                    030
              B11S := B12S := B22S := O;
                                                                    030
174
                                                                    031
174
                       ARS11**2 + ARS12**2 + ARS21**2 + ARS22**2
                                                                    031
174
                    > ASR11**2 + ASR12**2 + ASR21**2 + ASR22**2
                                                                    031
175
                                                                    031
175
               *THEN**BEGIN*
                                                                    031
176
    • COMMENT • **********************
176
                                                                   *031
176
                                                                   *031
            HARSII > HASRII
176
176
                                                                   *031
    176
    176
                                                                   *032
176
176
            BERECHNEN DES KOMMUTATORS CRS*, A* UND B*
                                                                   *032
                                                                   *032
176
    176
                                                                    032
176
              *FOR* K:=1 *STEP* 1 *UNTIL* R22*
                                                                    032
176
                       R2+1 'STEP' 1 'UNTIL' S22
                                                                    032
177
                       S2+1 *STEP* 1 *UNTIL* N *DO**BEGIN*
                                                                    032
177
                                                                    032
178
```

```
A11S := A(/S21*K/)**2 + A11S*
                                                                       033
178
                  A12S := A(/S21, K/) * A(/S2, K/) + A12S;
                                                                       033
179
                  A22S := A(/S2 \cdot K/) * * 2 + A22S 
                                                                       033
180
                                                                       033
181
181
                  B11S := A(/K, R21/)**2 + B11S;
                                                                       033
                  B12S := A(/K, R2/) * A(/K, R21/) + B12S;
                                                                       033
182
                  B22S := A(/K, R2/)**2 + B22S;
                                                                       033
183
                                                                       033
184
                  C11S := A(/K_2, R21/)*A(/K_2, S21/)
                                                                       033
184
                                                                       033
                        -A(/R21, K/)*A(/S21, K/) + C11S_3
185
                  C12S := A(/K_B R21/) * A(/K_B S2/)
                                                                       034
185
                        - A(/R21 + K/) \times A(/S2 + K/) + C12S1
                                                                       034
186
186
                  C21S := A(/K, R2/) * A(/K, S21/)
                                                                       034
                        -A(/R2, K/) * A(/S21, K/) + C21S3
                                                                       034
187
                  C22S := A(/K, R2/) * A(/K, S2/)
                                                                       034
187
                        - A(/R2 + K/) \times A(/S2 + K/) + C22S
                                                                       034
188
              *END *E DER BERECHNUNG VON CRS*, A* UND B*;
                                                                       034
188
                                                                       034
189
               C11 := C11S;
                                                                       034
189
                                                                       034
               C12 := C12S;
190
                                                                       035
191
               C21 := C21S;
                                                                       035
192
               C22 := C22S;
193
                                                                       035
    193
                                                                      *035
193
            BERECHNEN VON A UND B
                                                                      *035
193
                                                                      ×035
193
193
    035
193
               A11 := ASR11**2;
                                                                       035
193
               A12 := ASR11*ASR21 + ASR12*ASR22 + A12S;
194
                                                                       036
               A22 := ASR22**2;
                                                                       036
195
196
               H := ASR21**2;
                                                                       036
               B11 := A11 + H + B115;
                                                                       036
197
               B22 := ASR12**2;
                                                                       036
198
199
               A11 := A11 + B22 + A11S;
                                                                       036
              B22 := B22 + A22 + B22S;
                                                                       036
200
               A22 := A22 + H + A225;
                                                                       036
201
               B12 := ASR12*ASR11 + ASR22*ASR21 + B12S;
                                                                       036
202
                                                                       036
203
               XT11 := XT12 := XT21 := XT22 := F1A := 0;
                                                                       037
203
204
               ITER := "FALSE";
                                                                       037
               NIT := 0;
                                                                       037
205
                                                                       037
206
    ITERX:
                                                                       037
206
               NIT := NIT + 1;
               C120 := AR12 ** 2;
                                                                       037
208
               C210 := AR21 ** 2;
                                                                       037
209
               D120 := AS12 ** 23
                                                                       037
210
211
               D210 := AS21 ** 2;
                                                                       037
               C1D1 := AR11 - AS11;
                                                                       037
212
                                                                       038
213
               C1D2 := AR11 - AS22#
                                                                       038
               C2D1 := AR22 - AS11;
214
                                                                       038
               C2D2 := AR22 - AS223
215
                                                                       380
216
    216
```

```
*038
216
    ×
216
             BERECHNEN DES KOMMUTATORS CRS
                                                                       *038
    ¥
216
                                                                       *038
    216
216
                                                                        038
               C11 := (ARS11 - ASR11) \times (AR11 - AS11)
                                                                        039
216
                                                                        039
                    + ARS21 * AR21 - ASR12 * AR12
217
                    + ASR21 × AS21 - ARS12 × AS12
                                                + C11;
                                                                        039
217
217
               C12 := (ARS12 - ASR21) \times (AR11 - AS22)
                                                                        039
                    + ARS22 * AR21 - ASR22 * AR12
                                                                        039
218
                    + ASR11 * AS12 - ARS11 * AS21
                                                + 6123
                                                                        039
218
              C21 := (ARS21 - ASR12) \times (AR22 - AS11)
                                                                        039
218
                    + ARS11 * AR12 - ASR11 * AR21
                                                                        039
219
                    + ASR22 * AS21 - AR822 * AS12 + C213
                                                                        039
219
              C22 := (ARS22 - ASR22) * (AR22 - AS22)
219
                                                                        039
                    + ARS12 * AR12 - ASR21 * AR21
                                                                        040
220
                    + ASR12 * AS12 - ARS21 * AS21 + C22;
                                                                        040
220
                                                                        040
220
         NCRS := SQRT(C11**2 + C12**2 + C21**2 + C22**2);
                                                                        040
220
                                                                        040
221
              *IF* NCRS < *-14 *THEN**BEGIN*
                                                                        040
221
                   *IF * ITER *THEN**GOTO* TRANSX
                                                                        040
223
                                *FLSE**GOTO* KEITRA
                                                                        040
225
               "END"E NCRS < "-14;
                                                                        040
227
                ITER := "TRUE";
                                                                        040
228
                                                                        041
229
    - COMMENT - **********************************
229
                                                                       *041
229
                                                                       *041
229
             BESTIMMEN DER NAEHERUNG X2 FUER EIN MINIMUM
                                                                       *041
             DER FUNKTION F(X)
229
                                                                       *041
229
             BERECHNEN DER MATRIX FUER DEN GAUSS - ALGORITHMUS
                                                                       *041
229
             ZUR LOESUNG DES GLEICHUNGSSYSTEMS K X = -CRS
229
                                                                       * 041
229
                                                                       *041
    229
                                                                        042
229
               M(/1,1/):=-2*ASR11*ARS11+ C1D1**2 + C210 + D120 + A11 + B11;042
229
               N(/2,1/) := M(/1,2/):=-ASR11*ARS12 - ASR21*ARS11 - C1D2*AS12042
230
231
                         - C1D1*AS21 + A12;
                                                                        042
              H(/3_31/) := H(/1_33/) := -ASR12 \times ARS11 - ASR11 \times ARS21
                                                                        042
231
                         + C2D1 * AR21 + C1D1 * AR12 + B12;
                                                                        042
232
              M(/4_11/) := M(/1_4/) := -ASR21 \times ARS21 - ASR12 \times ARS12
                                                                        042
232
                            AR21 * AS21 - AR12 * AS12;
                                                                        042
233
               M(/2,2/):=-2*ASR21*ARS12+ C1D2**2 + C210 + D210 + A22 + B11;042
233
               M(/3,2/) := M(/2,3/) := -ASR11 \times ARS22 - ASR22 \times ARS11
                                                                        042
234
                            AR21 * AS12 - AR12 * AS213
                                                                        043
235
         H(/4,2/) := H(/2,4/) := -ASR22 \times ARS12 - ASR21 \times ARS22
                                                                        043
235
                         + C2D2 * AR21 + C1D2 * AR12 + B12;
236
                                                                        043
              M(/3,3/) := -2 * ASR12 * ARS21 + C2D1 ** 2 + C120
                                                                        043
236
                         + 0120 + A11 + B223
                                                                        043
237
              M(/4_3/) := M(/3_4/) := -ASR12 \times ARS22 - ASR22 \times ARS21
                                                                        043
237
                         - C2D2 * AS12 - C2D1 * AS21 + A12;
                                                                        043
238
              M(/4,4/) := -2 * ASR22 * ARS22 + C2D2 ** 2 + C120
                                                                        043
238
                        + D210 + A22 + B22;
                                                                        043
239
              H(/1,5/) := -C11;
                                                                        043
239
```

```
M(/2,5/) := -0.12
                                                       044
240
            M(/3,5/) := -C21;
                                                       044
241
                                                       044
            M(/4.5/) := -C22;
242
                                                       1144
243
   243
                                                      * 044
243
          LOESEN DES LINEAREN GLEICHUNGSSYSTEMS K X = -CRS
                                                      *044
243
                                                      * 044
         MIT DEM GAUSS - ALGORITHMUS
243
243
   ****************
243
                                                       045
243
            FX2 := "TRUE";
                                                       045
243
            GAUSS2(4, M, X, SINGULAER);
                                                       045
244
                                                       045
245
   245
                                                      ×045
245
                                                      *045
245
         BESTIMMUNG DES PARAMETERS X2
245
   245
                                                       045
245
                                                       046
245
            X11 := X(/1/);
                                                       046
            X12 := X(/2/)
246
                                                       046
            X21 := X(/3/);
247
                                                       046
            X22 := X(/4/);
248
                                                       046
249
                X11 **2 + X12 **2 + X21 **2 + X22 **2 > 100 *THEN*046
249
250
   SINGULAER:
               FX2 := "FALSE";
                                                       046
                                                       046
252
   252
                                                      *046
252
          BESTIMMEN DER NACHERUNG X1 FUER EIN MINIMUM
                                                      *047
252
                                                      *047
252
          DER FUNKTION F(X)
                                                      * 047
252
   252
   252
                                                      *047
252
                                                      *047
252
          BESTIMMEN DES PARAMETERS X1
                                                      *047
252
         BERECHNEN VON F1 = F(X1)
                                                      *047
252
252
   252
                                                       048
252
              H1 := ASR11 * C11 + ASR12 * C213
                                                       048
252
                                                       048
              H2 := ASR21 * C11 + ASR22 * C21;
253
                                                       048
              H3 := ASR11 \times C12 + ASR12 \times C223
254
              H4 := ASR21 * C12 + ASR22 * C22;
                                                       048
255
              SP2 := (ARS11 * C11 + ARS21 * C21) * H1
                                                       048
256
                   (ARS11 * C12
                            + ARS21 × C22) × H2
                                                       940
257
                             + ARS22 * C21)
                                         * H3
                                                       048
                   (ARS12 * C11
257
                  (ARS12 × C12 + ARS22 × C22)
                                         * H43
                                                       048
257
                                                       049
             H5 := H1 × C11 + H2 × C12;
257
                                                       049
             H6 := H1 * C21
                         + H2 \times C22;
258
             H7 := H3 \times C11 + H4 \times C12;
                                                       049
259
                                                       049
              H8 := H3 \times C21 + H4 \times C22;
260
              H1 := C1D1 * C11 + AR12 * C21 - AS21 * C12;
                                                      049
261
```

```
H2 := AR21 * C11
                                          C2D1 * C21 - AS21 * C223
                                                                                049
262
                                          AR12 * C22
                                                       - AS12 * C113
                    H3 := C1D2 * C12 +
263
                                                                                049
                                       + C2D2 * C22 - AS12 * C21;
                    H4 := AR21 * C12
264
                                                                                049
                    SP3 == H5 × H1 + H6 × H2 + H7 × H3 + H8 × H43
                                                                                049
265
                                       + C12 * C12 + C21 * C21 + C22*C22$049
266
                    SP1 := C11 * C11
                    SP4 := (A11 * C11 + A12 * C12) * C11
267
                                                                                050
                            (A12 \times C11 + A22 \times C12)
                                                           C12
                                                                                050
268
                                                                                050
268
                            (A11 × C21
                                       + A12 * C22)
                                                           C21
                                           A22 * C22)
                                                          C22;
                                                                                050
                            (A12 × C21
                                       +
268
                    SP5 == (B11 * C11
                                       +
                                            B12 * C21)
                                                                                05 C
268
                                                          C11
269
                            (B11 \times C12)
                                            B12 * C22)
                                                          C12
                                                                                050
                                        + B22 × C21) ×
269
                            (B12 \times C11)
                                                          C21
                                                                                050
                            (B12 \times C12 + B22 \times C22) \times C22
                                                                                05C
269
                   + (B12 * C12 * D22 * C22 * ....
N1 := H1 **2 + H2 **2 + H3 **2 + H4 **2;
                                                                                05 C
269
                  N2 := H5 **2 + H6 **2 + H7 **2 + H8 **2;
MINIMUM(ALPHA; 2*SP1; SP4+SP5+N1-2*SP2; -2*SP3;
270
                                                                                050
271
                                                                                051
272
                                                                                051
                            N2 , NA , F1);
272
                                                                                051
                   Y11 := C11 * ALPHA;
                                                                                051
272
                   Y12 := C12 * ALPHA;
                                                                                051
273
                                                                                051
274
                   Y21 := C21 * ALPHA;
275
                                                                                051
                   Y22 := C22 * ALPHA;
276
                                                                                051
                 •IF• FX2 •THEN••BEGIN•
                                                                                051
276
                                                                                051
278
375
     278
                                                                               ×052
              BERECHNEN VON F2 = F(X2)
                                                                               *052
278
                                                                               *052
278
278
     052
278
278
                    H1 := ASR11 * X11
                                        + ASR12 * X21;
                                                                                052
                                        + ASR22 * X21;
279
                    H2 == ASR21 * X11
                                                                                052
                                           ASR12 * X223
                    H3 := ASR11 * X12
                                        +
                                                                                052
280
                    H4 == ASR21 * X12 +
                                          ASR22 * X225
                                                                                052
281
                    SP2 := (ARS11 \times X11 + ARS21 \times X21)
                                                                                053
282
                                                           * H1
                            (ARS11 * X12 + ARS21 * X22) * H2
                                                                                053
283
                            (ARS12 \times X11 + ARS22 \times X21) \times H3
                                                                                053
283
                            (ARS12 * X12 +
                                            ARS22 \times X22) \times H41
                                                                                053
283
                                     + H2 * X12;
283
                    H5 := H1 * X11
                                                                                053
                    H6 := H1 \times X21
                                       H2 * X22;
                                                                                053
284
                   H7 := H3 \times X11 + H4 \times X12;
                                                                                053
285
                    H8 := H3 \times X21 + H4 \times X221
                                                                                053
286
                   H1 := C1D1 \times X11 + AR12 \times X21 - AS21 \times X12;
                                                                                053
287
                                                       - AS21 * X22;
                    H2 := AR21 \times X11 + C2D1 \times X21
                                                                                053
288
                    H3 := C1D2 * X12
                                       + AR12 * X22
                                                       - AS12 \times X11;
                                                                                054
289
                    H4 := AR21 * X12
                                       + C2D2 * X22 - AS12 * X213
290
                                                                                054
                    SP3 := H5 × H1 + H6 × H2 + H7 × H3 + H8 × H4; 054
SP1 := C11 × X11 + C12 × X12 + C21 × X21 + C22×X22;054
291
                    SP1 := C11 * X11
292
293
                    SP4 := (A11 \times X11 +
                                           A12 \times X12
                                                           X11
                                                                                054
                                                        ¥
294
                            (A12 \times X11 + A22 \times X12)
                                                        ¥
                                                           X12
                                                                                054
294
                            (A11 \times X21)
                                       + A12 \times X22)
                                                           X21
                                                                                054
294
                            (A12 \times X21)
                                       + A22 × X22)
                                                           X223
                                                                                054
                    SP5 := (B11 * X11
                                       + B12 * X21)
                                                                                054
294
                                                        ¥
                                                           X 1 1
                                       + B12 × X22)
295
                            (B11 \times X12)
                                                        ×
                                                           X12
                                                                                054
```

```
055
295
                        (B12 \times X11 + B22 \times X21) \times X21
                       (B12 \times X12 + B22 \times X22) \times X22
                                                                       055
295
                  N1 := H1 **2 + H2 **2 + H3 **2 + H4 **2;

N2 := H5 **2 + H6 **2 + H7 **2 + H8 **2;

F2 := (SP1 - SP2 - SP3) * 2 + SP4 + SP5 + N1 + N2
                                                                      055
295
                                                                       055
296
                                                                       055
297
                                                                      055
298
                                                                       055
               *FND*F DER BERECHNUNG VON F1 UND F2;
298
                                                                       055
299
   299
                                                                     *055
299
299
            VERGLEICH, WELCHE NAEHERUNG DIE FUNKTION F(X)
                                                                      ×056
            BESSER MINIMIERT : X1 ODER X2
299
                                                                      *056
                                                                      ×056
299
    299
299
              TITE FX2 ! F1<F2 THEN BEGIND
                                                                       056
299
                  X11 := Y11;
                                                                       056
301
                  X12 := Y12;
X21 := Y21;
                                                                       056
302
                                                                       056
303
                                                                       056
                  X22 := Y22
304
              PENDPE DES SETZENS DER BESSEREN NAEHERUNG;
                                                                       057
305
                                                                       057
306
                                                                       057
              X111 := X111 + X111
306
                                                                       057
              XT12 := XT12 + X12#
307
               XT21 := XT21 + X21;
                                                                       057
308
                                                                       057
309
               XT22 := XT22 + X22#
                                                                       057
310
                    ABS(X11) + ABS(X12) + ABS(X21) + ABS(X22) < 1-10
                                                                       057
310
                     ! NIT > 29 ! (ABS(F1A - F1) < ^-8 & F2 > 0)
                                                                       057
311
                                                                       057
              *THEN* GOTO* TRANSX;
311
                                                                       058
               F1A := F1;
312
                                                                       058
313
    313
                                                                      *058
313
            TRANSFORMATION DER PIVOT - BLÖCKE
                                                                      ×058
313
                                                                      *N58
313
    313
                                                                       058
313
                                                                       058
               H1 := X11 * ASR11 + X12 * ASR21;
313
               H2 := X11 * ASR12 + X12 * ASR223
                                                                       058
314
                                                                       059
               H3 := X21 * ASR11 + X22 * ASR21;
315
              H4 := X21 * ASR12 + X22 * ASR22;
                                                                       059
316
              AR11 := AR11 - H1;
                                                                       059
317
                                                                       059
              AR12 := AR12 -
                               H2 3
318
                                                                       059
              AR21 := AR21
                            - H3s
319
                                                                       059
              AR22 := AR22 - H45
320
                                                                       059
              ARS11 := (AR11 - AS11) \times X11 - AS21 \times X12
321
                      + AR12 * X21 + ARS113
                                                                       059
322
              ARS12 := (AR11 - AS22) \times X12 - AS12 \times X11
                                                                       059
322
                      + AR12 * X22 + ARS123
                                                                       059
323
              ARS21 := (AR22 - AS11) * X21 - AS21 * X22
                                                                       060
323
                                                                      060
                     + AR21 * X11 + ARS21;
324
              ARS22 := (AR22 - AS22) \times X22 - AS12 \times X21
                                                                      060
324
                                                                       060
                     + AR21 * X12 + ARS22;
325
              AS11 := AS11 + (ASR11 * X11 + ASR12 * X21);
                                                                       060
325
```

```
326
            AS12 := AS12 +
                        (ASR11 \times X12 + ASR12 \times X22);
                                                         060
            AS21 := AS21 + (ASR21 * X11 + ASR22 * X21);
327
                                                         060
328
            AS22 := AS22
                        (ASR21 \times X12 + ASR22 \times X22);
                                                         060
329
                                                         061
329
   329
                                                        *061
329
          MODIFIKATION DES KOMMUTATORS CRS
                                                        *061
329
                                                        *061
329
   329
                                                         061
329
            C11 := B12S * XT21 + A12S * XT12
                                                         061
                + (B115 + A115) * XT11 + C115;
330
                                                         061
            C12 := B12S * XT22 + A12S * XT11
330
                                                         061
                + (B11S + A22S) * XT12 + C12S;
331
                                                         061
            C21 := B12S * XT11 + A12S * XT22
331
                                                         061
332
                + (B22S + A11S) \times XT21 + C21S
                                                         062
            C22 := B125 * XT12 + A125 * XT21
332
                                                         062
333
                + (B22S + A22S) * XT22 + C22SJ
                                                         062
333
                                                         062
333
            *GOTO* ITERX;
                                                         062
334
                                                         062
334
   334
                                                        *062
334
          FALLS DIE TRANSFORMATIONSMATRIX IDENTISCH MIT DER
                                                        *062
334
          EINHEITSMATRIX IST, WIRD KEINE TRANSFORMATION DURCHGEFUEHRT
                                                        *062
334
                                                        *063
334
   334
                                                         063
334
   TRANSX:
            "IF" ABS(XT11) + ABS(XT12) + ABS(XT21) + ABS(XT22) < FP
                                                         063
336
              *THEN * *GOTO * KEITRA;
                                                         063
337
                                                         063
            ITSTEHT := "FALSE";
337
                                                         063
338
            MM := MM + 13
                                                         063
339
                                                         063
   339
339
                                                        *064
339
          TRANSFORMATION DER 2×R-1 - TEN UND 2×R - TEN
                                                        *064
339
          ZEILE DER MATRIX A
                                                        *064
339
                                                        *064
339
   339
                                                         064
339
            *FOR* I:=1 *STEP* 1 *UNTIL* N *DO**BEGIN*
                                                         064
              A(/R21, I/) := A(/R21, I/) - A(/S21, I/) \times X111
341
                                                         064
                                   - A(/S2, I/) * XT12;
342
                                                         064
342
              A(/R2, I/) := A(/R2, I/) - A(/S21, I/) \times XT21
                                                         064
343
                                   A(/S2, I/) \times XT22;
                                                         065
               *IF * LINKS *THEN**BEGIN*
                                                         065
343
345
                                                         065
345
   345
                                                        *065
345
         BERECHNEN DER LINKSEIGENVEKTOREN
                                                        ¥065
345
                                                        *065
   345
345
                                                         065
345
                TI(/R21, I/) := TI(/R21, I/) - TI(/S2, I/) \times XT12
                                                         065
```

```
346
                                                                                               - TI(/S21_{2} I/) \times XT11_{2}
                                                                                                                                             066
                                          TI(/R2, I/) := TI(/R2, I/) - TI(/S21, I/) \times XT21
346
                                                                                                                                             066
347
                                                                                            - TI(/S2, I/) \times XT22
                                                                                                                                             066
                                     .FND.
347
                                                                                                                                             066
348
                               *END *E DER ZEILENTRANSFORMATIONEN;
                                                                                                                                             066
349
                                                                                                                                             066
349
         349
                                                                                                                                           *066
                          TRANSFORMATION DER 2*S-1 - TEN UND 2*S - TEN
349
         ×
                                                                                                                                           *066
349
         ¥
                          SPALTE DER MATRIX A
                                                                                                                                           *066
349
                                                                                                                                           *067
349
         349
                                                                                                                                             067
                              *FOR* I:=1 *STEP* 1 *UNTIL* N *DO**BEGIN*
349
                                                                                                                                             067
351
                                    A(/I) S21/) := A(/I) R21/) × XT11 +
                                                                                                                                             067
352
                                                                A(/I) R2/) * XT21 + A(/I) S21/);
                                                                                                                                             067
352
                                     A(/I_{2} S2/) := A(/I_{2} R21/) \times XT12 +
                                                                                                                                             067
353
                                                              A(/I) R2/) * XT22 + A(/I) $2/);
                                                                                                                                             067
353
                                     ·IF ·
                                                RECHTS
                                                              *THEN**BEGIN*
                                                                                                                                             067
355
                                                                                                                                             067
355
         355
                                                                                                                                           * 068
355
                         BERECHNEN DER RECHTSEIGENVEKTOREN
                                                                                                                                           * 068
355
                                                                                                                                           *068
355
         355
                                                                                                                                             068
355
                                          T(/I_3 S21/) := T(/I_3 S21/) + T(/I_3 R21/) \times XT11
                                                                                                                                             8 60
356
                                                                                           + T(/1, R2/) * XT21;
                                                                                                                                             068
356
                                          T(/I), S2/) := T(/I), S2/) + T(/I), R21/) * XT12
                                                                                                                                             860
357
                                                                                        + T(/I) R2/) * XT22
                                                                                                                                             068
357
                                    *END *
                                                                                                                                             069
358
                             *END*E DER SPALTENTRANSFORMATION
                                                                                                                                             069
359
                               069
360
                                                                                                                                             069
360
                               *ELSE**BEGIN*
                                                                                                                                             069
362
                                                                                                                                             069
         362
362
                                                                                                                                           *069
                         362
                                                                                                                                           ×069
362
                                                                                                                                           *069
362
         икий нижиминий и жими кими кими и жими и жим
362
                                                                                                                                             070
362
                                                                                                                                             070
362
         362
                                                                                                                                           *07C
362
                         Dieser Fall wird analog zu dem Fall | | IARSII > | LIASRII
                                                                                                                                           *07C
362
                         abgehandelt, wobei R und S zu vertauschen sind.
                                                                                                                                           *07C
                         Aus Platzgründen wird dieser Teil nicht mit abgedruckt.
362
         ¥
                                                                                                                                           *07C
362
362
         362
                                                                                                                                             071
362
                                                                                                                                             071
                             "END"E DES FALLS
                                                                362
                                                                                                                                            071
                                                                                                                                             071
363
                                                                                                                                             071
363
         KEITRA:
```

```
071
363
363
     *071
364
              TRANSFORMATION ZUM ANNULLIEREN DES ANTISYMMETRISCHEN (ARS-)
364
                                                                             *071
              BEZW. SYMMETRISCHEN (ARS+) TEILS VON A AM PIVOT (RJS) >
                                                                             *071
364
     ¥
              MITTELS EINER 4x4 - HOUSEHOLDERMATRIX
                                                                             *072
364
                                                                             *072
364
     364
                                                                              072
364
                                                                              072
            DREI := DOP := 'FALSE';
364
                                                                              072
365
                                                                              072
            AF13 := A(/R21_32*S-1/) - A(/2*S-1_3R21/)
365
                                                                              072
            AF14 := A(/R21_{3}2*S/) - A(/2*S_{3}R21/);
366
            AF23 := A(/R2,2*S-1/) - A(/2*S-1,R2/);
                                                                              072
367
                                   - A(/2*S)R2/);
                                                                              072
            AF24 := A(/R2)2*S/)
368
                                                                              073
369
                                                                              073
            A13 := A31 := A(/R21,2*S-1/) + A(/2*S-1,R21/);
369
                                                                              073
            A14 := A41 := A(/R21_2*S/) + A(/2*S_2R21/);
370
                                                                              073
371
            A23 := A32 := A(/R2_12*S-1/) + A(/2*S-1_1R2/)_{3}
372
            A24 := A42 := A(/R2)2*S/) + A(/2*S)R2/);
                                                                              073
                                                                              073
            A11 := A(/2 \times R - 1) \times 2 \times R - 1/)
373
                                                                              073
            A22 := A(/2*R_22*R/) ;
374
                                                                              073
375
            A33 := A(/2 \times S - 1) \times (S - 1/)
                                                                              073
            A44 := A(/2*S,2*S/) ;
376
                                                                              073
            A12 := A(/2*R-1)2*R/)
377
                                                                              074
            A21 := A(/2*R_32*R-1/);
378
            A34 := A(/2*S-1*2*S/);
                                                                              074
379
                                                                              074
            A43 := A(/2*S,2*S-1/);
380
                                                                              074
381
            IMLR := ((A11 - A22) / 2) **2 + A12 * A213
                                                                              074
381
            RELR1 := (A11 + A22) / 2;
                                                                              074
382
                               *THEN**BEGIN*
                                                                              074
            • IF • IMLR >= 0
383
                                                                              074
               IMLR := SQRT(IMLR);
385
               RELR2 := RELR1 - IMLR;
                                                                              074
386
               RELR1 := RELR1 + IMLR;
                                                                              074
387
                                                                              075
               IMLR := 0
388
                                                                              075
            *END*
389
                                                                              075
390
            'ELSE'
               IMLR := SQRT(-IMLR);
                                                                              075
391
                                                                              075
392
            IMLS := ((A33 - A44) / 2) \times 2 + A34 \times A435
                                                                              075
392
            RELS1 := (A33 + A44) / 21
                                                                              075
393
                                                                              075
            *IF* IMLS >= 0
                              "THEN" BEGIN
394
                                                                              075
               IMLS := SORT(IMLS);
396
                                                                              075
               RELS2 := RELS1 - IMLS;
397
               RELS1 := RELS1 + IMLS;
                                                                              076
398
                                                                              076
               IMLS := 0
399
                                                                              076
400
            • END •
                                                                              076
            'ELSE'
401
                                                                              076
               IMLS := SQRT(-IMLS);
402
                                                                              076
403
            • IF • IMLR = 0
                                                                              076
                               *THEN* BEGIN*
403
               'IF' IMLS = 0
                                                                              076
405
                                  *THEN**BEGIN*
                  ORE := RELR1 - RELS1;
                                                                              076
407
                  DIM := IMLR - IMLS
                                                                              076
408
```

```
077
               "END"
409
               'ELSE''BEGIN'
                                                                              077
410
                  DRE := RMIN(ABS(RELR1-RELS1), ABS(RELR1-RELS2));
                                                                              077
412
                  DIM := IMLR
                                                                              077
413
                                                                              077
               'END'
414
                                                                              077
            · END ·
415
            *ELSE**IF* INLS "= 0 *THEN**BEGIN*
                                                                              077
416
                                                                              077
               DRE := RMIN(ABS(RELR1-RELS1), ABS(RELR2-RELS1));
419
                                                                              077
               DTM := IMLS
420
                                                                              077
            *END*
421
                                                                              078
            *ELSE**BEGIN*
422
               DRE := RMIN( RMIN( ABS(RELR1-RELS1), ABS(RELR1-RELS2) ),
                                                                              078
424
                            RMIN( ABS(RELR2-RELS1), ABS(RELR2-RELS2) ) );
                                                                              078
425
               DIM := 0
                                                                              078
425
                                                                              078
            *END*E DER ABSTANDSBERECHNUNG:
426
                                                                              078
427
                                                                              078
            SYMM := ( (A13 **2 + A14 **2 + A23 **2 + A24 **2)
427
                                                                              078
                      + DRE**2 ) * LAMBDA
428
                                                                              078
                  > (AF13 **2 + AF14 **2 + AF23 **2 + AF24 **2)
428
                                                                              078
                      + DIM **2;
428
                                                                              079
428
                                                                              079
            • 1F •
                   SYMM
                          *THEN * BEGIN'
428
                                                                              079
430
               NSY := NSY + 1;
                                                                              079
430
                                                                              079
               A11 := A11 * 2;
431
                                                                              079
               A22 := A22 × 2;
432
                                                                              079
433
               A33 := A33 * 2;
                                                                              079
434
               A44 := A44 * 25
                                                                              079
               A12 := A21 := A12 + A21;
435
                                                                              079
436
              A34 := A43 := A34 + A43;
                                                                              080
437
              WUR := SQRT(((A11-A22) /2) **2 + A12 **2);
                                                                              080
437
               L2 := (A11 + A22) / 23
                                                                               080
438
               L1 := L2 - WUR;
                                                                              380
439
                                                                               080
               L2 := L2 + WUR
440
                                                                              380
441
                                                                              080
441
            *END*
           *ELSE **BEGIN*
                                                                              080
442
                                                                              080
444
                                                                              380
               NSCHIE := NSCHIE + 1;
444
               AF12 := A(/2*R-1)2*R/) - A(/2*R)2*R-1/);
                                                                              081
445
               AF34 := A(/2*S-1)2*S/) - A(/2*S) 2*S-1/);
                                                                              081
446
                                                                              081
               A13 := AF13;
447
                                                                              081
               A14 := AF143
448
                                                                              081
449
               A23 := AF23;
               A24 := AF24
                                                                               081
450
                                                                              081
451
            *END*;
                                                                               081
452
                  ABS(A13) + ABS(A14) + ABS(A23) + ABS(A24) > 1-14
                                                                               081
            · IF ·
452
                                                                               081
            'THEN' BEGIN'
453
                                                                              082
454
                   SYMM
                         *THEN * BEGIN *
                                                                              082
454
            · IF ·
                                                                              082
456
     456
                                                                             *082
456
     ×
```

```
* 082
                 SPURSHIFT
456
    ×
                                                                    *082
456
    ×
    456
                                                                     082
456
               SHIFT := TAU1 := (A11 + A22 + A33 + A44) / 43
                                                                     082
456
                                                                     083
457
               WIEDERHOLUNG := MEHRFACH := "FALSE";
                                                                     083
458
               EPSI := 1-2;
                                                                     083
459
    459
                                                                    *083
459
459
            BERECHNEN DER HILFSVARIABLEN
                                                                    *083
                                                                    *083
459
    459
                                                                     880
459
                                                                     083
459
              A120 := A12**2;
                                                                     084
460
               A130 := A13**2;
               A140 := A14**2;
                                                                     084
461
                                                                     084
              A230 := A23**2;
462
              A240 := A24**23
                                                                     084
463
                                                                     084
              A340 := A34**2#
464
                                                                     084
              Z1 := A24 \times A34;
465
                                                                     084
466
              72 := A12 * A13;
              H := 2 \times (A14 \times (A12 \times A24 + A13 \times A34) + A23 \times (Z2 + Z1));
                                                                    084
467
              TAU := A120 + A130 + A140 + A230 + A240 + A3403
                                                                     084
468
                                                                     084
469
                                                                     085
469
   POLYNOM:
              A11 := A11 - TAU1;
               A22 := A22 - TAU1;
                                                                     085
471
                                                                     085
               A33 := A33 - TAU1;
472
              A44 := A44 - TAU1;
                                                                     085
473
                                                                     085
474
              L1 := L1 - TAU1;
              L2 := L2 - TAU1;
                                                                     085
475
                                                                     085
476
              Z3 := A33 * A44;
                                                                     085
476
               74 := A22 × A44 - A240;
                                                                     085
477
                                                                     085
               Z5 := A33 + A443
478
                                                                     086
               76 := 75 + A22;
479
                                                                     086
480
    480
                                                                    *086
480
           KOEFFIZIENTEN DES CHARAKTERISTISCHEN POLYNOMS
                                                                    *086
480
                                                                    *086
480
    480
                                                                     086
480
               K041 := A11 \times (A33 \times Z4 - A22 \times A340 - A44 \times A230)
                                                                     086
480
                                                                     086
                    + (A340 - Z3) \times A120 - Z4 \times A130
481
                    + (A230 - A22*A33) * A140
                                                                     087
481
                                                                    087
                    + 2 \times (A11 \times A23 \times Z1 + Z2 \times (A23 \times A44 - Z1)
481
                          + A14 \times (A12 \times (A24 \times A33 - A23 \times A34)
                                                                    087
481
                                 + A13 * (A22*A34 - A23*A24)));
                                                                    087
481
             K042 := (A22 + A44) * (A130 - A11*A33) - (A11 + A33) * Z4
                                                                    087
481
                    + (A11 + A22) \times A340 + (A11 + A44) \times A230
                                                                     087
482
                    + (A22 + A33) \times A140 + Z5 \times A120 - H_{3}
                                                                     087
482
             K043 := A11*Z6 + A22*Z5 + Z3 - TAU;
                                                                     087
482
              KO44 := *IF* WIEDERHOLUNG
                                                                     087
483
                       "THEN" -A11 - Z6 "ELSE" 0;
                                                                     087
484
```

```
880
484
          *IF * ABS(K042) + ABS(K043) < 1-13 * ABS(K044)
                                                            880
484
          *THEN**BEGIN*
                                                            880
485
                                                            880
486
486
             SHIFT := SHIFT - TAU1;
                                                            880
                                                            380
487
            A11 := A11 + TAU1;
                                                            088
487
            A22 := A22 + TAU1;
                                                            880
488
                                                            380
            A33 := A33 + TAU1;
489
                                                            880
            A44 := A44 + TAU1;
490
            L1 := L1 + TAU1;
                                                            089
491
            L2 := L2 + TAU1;
         089
492
                                                            089
493
                                                            089
494
                                                            089
495
                                                            089
          ^{\bullet}IF ^{\bullet} ABS(K044) + ABS(K042) ( ^{-1}-12 * K043
495
                                                            089
496
           'THEN 'BEGIN'
                                                            089
497
   • COMMENT • *****************
497
                                                           ×089
497
          LÖSEN DER BIQUADRATISCHEN GLEICHUNG
                                                           *090
497
                                                           *09C
497
    ************************************
497
                                                            090
497
                                                            090
            Z3 := - K043/2
497
                                                            090
            Z4 := Z3 **2;
498
499
                                                            090
            DISKR := Z4 - K0413
500
            DISKR := "IF" DISKR< "-14*(Z4+K041) "THEN"
                                                            090
                                           *ELSE*
                                                 SORT(DISKR):
                                                            090
501
                                                            090
            Y1 := Z3 - DISKR:
501
                                                            091
            Y2 := Z3 + DISKR;
502
            091
503
            Y2 := SQRT(Y2);
EW(/1/) := -Y2;
                                                            091
504
                                                            091
505
            EW(/2/) := -Y1;
                                                            091
506
                                                            091
            EW(/3/) := Y1;
507
                                                            091
508
            EW(/4/) := Y2;
          PENDPE DES LOSENS DER BIQUADRATISCHEN GLEICHUNG
                                                            091
509
                                                            091
510
          *ELSE**BEGIN*
512
   512
512
                                                           ×092
          KOEFFIZIENTEN DER "FERRARI - RESOLVENTE"
                                                            *092
512
                                                           *092
512
    512
                                                            092
512
             Z3 := 4 * K043 - K044 **2;
                                                            092
512
             K031 := K041 * Z3 - K042 **2;
                                                            092
513
             K032 := K044 * K042 - 4 * K041;
                                                            092
514
515
             K033 := -K043;
                                                            092
                                                            093
516
    516
                                                           *093
516
          BERECHNEN EINER NULLSTELLE DER "FERRARI - RESOLVENTE"
                                                           *093
516
    ¥
                                                            * 093
516
    #
```

```
516
     516
                                                                            093
516
                PE := K033 **2 - 3 * K032;
                                                                            093
                QU := KO\bar{3}3 \times (9 \times KO\bar{3}2 - 2 \times KO\bar{3}3 \times 2) - 27 \times KO\bar{3}1;
517
                                                                            093
518
                DISKR := QU **2 - 4 * PE **3;
                                                                            6P0
                                                                            094
519
                *IF* PE < *-13 *THEN* XO := -K033/3
                                                                            094
519
                *ELSE**IF* PE < *-9 & DISKR > 0
                                                                            094
521
                    *THEN**BEGIN*
                                                                            094
523
                          XD := 2 \times SQRT(PE);
                                                                            094
524
                          X1 := (X0 - K033) / 3;
                                                                            094
525
                                                                            094
526
                          X2 := (-X0 - K033) / 3;
527
                          XD := ^{\bullet}IF^{\bullet}
                                                                             094
528
                                ABS( ((X1 + K033) \times X1 + K032) \times X1 + K031 ) 094
                              ABS((X2 + K033) \times X2 + K032) \times X2 + K031) 094
528
                               "THEN"
                                                                            095
                                      X1 "ELSE" X2
528
528
                   'END'
                                                                            095
                                                                            095
529
                 'ELSE''BEGIN'
                                                                            095
                    •IF • DISKR > 0
531
                                                                            095
532
                    *THEN*
                                                                            095
532
                       XO := 2 \times SORT(PE) \times SIGN(QU)
                    'ELSE' BEGIN'
                                                                            095
533
535
                      XD := 2 \times SQRT(PE) \times SIGN(OU);
                                                                            095
536
                      P3 := 3 * PE;
                                                                            095
                       *FOR* I:=0,I+1 *WHILE* I<10 & ABS(D)>*-11*ABS(X0) *DO*095
537
                                                                            096
538
                       *BEGIN*
539
                          RO := XO **2;
                                                                            096
                          D := ((RQ - P3) * X0 - QU) / (3 * (RQ - PE));
                                                                            096
540
                         xo := xo - b
                                                                            096
541
542
                       PEND'E DES NEWTON - VERFAHRENS
                                                                            096
                    *END *E DER IF - ANNEISUNG:
                                                                            096
543
                                                                            096
544
                   x0 := (x0 - k033) / 3i
                                                                            096
544
                   x_0 := x_0 - (((x_0 + k_{033}) \times x_0 + k_{032}) \times x_0 + k_{031}) /
                                                                            096
545
                             ((3 \times X0 + 2 \times K033) \times X0 + K032)
                                                                            096
546
546
                 *END *E DES FALLS, BEI DEM KEINE DREIFACHE NULLSTELLE
                                                                            097
547
                       VORLIEGT:
                                                                            097
                                                                            097
547
547
     *097
547
547
             BERECHNEN DER NULLSTELLEN DES CHARAKTERISTISCHEN
                                                                           *097
547
             POLYNOMS
                                                                           *097
                                                                           *D97
547
     197
547
547
                                                                            097
                                                                            390
547
                Z5 := X0 **2;
                Z6 == 4 × XO;
                                                                            390
548
                24 := 4 \times K041;
                                                                            390
549
                QU := Z5 - Z4;
550
                                                                            890
                                                                            098
                PE := Z6 - Z3;
551
552
                'IF' QU<= -14 "THEN" QU := 0;
                                                                            098
                098
554
556
                *IF* QU > -14 ! PE > -14 *THEN *BEGIN*
                                                                            098
                                                                            390
                   *IF * PE*ABS(25+24) > QU*ABS(26+23)
558
559
                      *THEN* BEGIN*
                                                                            098
```

```
PE := SQRT(PE);
                                                                                 099
560
                           QU := (K044 \times X0 - 2 \times K042) / PE
                                                                                 099
561
                                                                                 099
                        "END"
562
                                                                                 099
                        *FLSE**BEGIN*
563
                                                                                 099
                           QU := SQRT(QU);
565
                                                                                 099
                           PE := (K044 \times X0 - 2 \times K042) / QU
566
                                                                                 099
                        · END ·
567
                • END • E DER BERECHNUNG VON P UND Q;
                                                                                 099
568
                                                                                 099
569
                                                                                 099
                 Z4 := (KO44 + PE) / 4;
569
                 Z5 := Z4 **2;
                                                                                 100
570
                                                                                 100
571
                 23 := (x_0 + u_0) / 2;
                                                                                 100
                 DISKR := Z5 - Z3;
572
                 100
573
                                                          *ELSE* SORT(DISKR);
                                                                                 100
574
                                                                                 100
574
                                                                                 100
574
                 EV(/1/) := DISKR - 24;
                 EW(/2/) := -DISKR - Z4;
                                                                                 100
575
                                                                                 100
576
                 24 := (K044 - PE) / 4;
                                                                                 100
576
                                                                                 101
                 Z5 == Z4 **2;
577
                 Z3 := (X0 - Q0) / 23
                                                                                 101
578
                                                                                 101
                 DISKR := Z5 - Z3;
579
                 DISKR = "IF" DISKR ( -13*(25+23)
                                                          'THEN' D
                                                                                 101
580
                                                           *ELSE* SORT(DISKR);
                                                                                 101
581
                                                                                 101
581
                                                                                 101
581
                 EW(/3/) := DISKR - Z4;
                 EW(/4/) := -DISKR - Z4;
                                                                                 101
582
                                                                                 101
583
                                                                                 101
                 *FOR* U:=1 *STEP* 1 *UNTIL* 3 *DO**BEGIN*
583
                                                                                 102
                     ¥ := U;
585
                                                                                 102
                     *FOR* Y:=U+1 *STEP* 1 *UNTIL* 4 *DO*
586
                                  ER(\A\) < ER(\A\)
                                                                                 102
                        FIFE
587
                                  W := V;
                                                                                 102
                        "THEN"
588
                              พ ^=บ
                                                                                 102
                     • 1F •
                                       *THEN**BEGIN*
589
                                                                                 102
                        TAU1 := EW(/W/);
591
                        EW(/W/) := EW(/U/);
                                                                                 102
592
                        EW(/U/) := TAU1
                                                                                 102
593
                     *END*E DER VERTAUSCHUNG
                                                                                 102
594
                                                                                 102
                  *FND*E DES SORTIERENS
595
                                                                                 103
              *END *E DER BERECHNUNG DER EIGENWERTE;
596
                                                                                 103
597
     VERGLEICH: DREI := "FALSE";
                                                                                 103
597
                        ABS(EN(/1/) - EN(/2/)) < EPSI "THEN"BEGIN"
                                                                                 103
                  ·IF ·
599
                           ABS(EW(/1/) - EW(/3/)) < EPSI "THEN"BEGIN"
                                                                                 103
                     ·IF ·
601
                             ABS(EV(/1/) - EV(/4/)) < EPSI *THEN**BEGIN*
                                                                                 103
603
                           IF .
                                                 *THEN**GOTO* ENDE
                                                                                 103
                                  WIEDERHOLUNG
605
                           *ELSE**BEGIN*
                                                                                 103
607
                              WIEDERHOLUNG := "TRUE";
                                                                                 103
609
     NEUVER:
                              EPSI := 1-12;
                                                                                 103
611
                                                                                 104
                              *GOTO* VERGLEICH
612
                                                                                 104
                           *END*E ABSCHLUSS
613
                                                                                 104
                       PENDPE VIERFACH
614
                                                                                 104
                        *ELSE**BEGIN*
615
                           TAU1 := (EW(/1/) + EW(/2/) + EW(/3/)) / 3;
                                                                                 104
617
```

```
DREI := "TRUE"
                                                                                 104
618
                        *END*E DREIFACH
                                                                                 104
619
                     *END*E DREIFACH ODER VIERFACH
                                                                                 104
620
                                                                                 104
621
                     *ELSE **BEGIN*
                        TAU1 := (EW(/1/) + EW(/2/)) / 2;
                                                                                 104
623
                               ABS(EN(/3/) - EN(/4/)) < EPSI *THEN**BEGIN*
                                                                                 105
624
                           TAU2 := (EN(/3/) + EN(/4/)) / 23
                                                                                 105
626
                                  ABS((L1 + L2) / 2 - TAU2)
                                                                                 105
627
                                ABS((L1 + L2) / 2 - TAU1)
                                                                                 105
628
                                    TAU1 := TAU2
                                                                                 105
                           *THEN*
628
                        *END *E ZWEI DOPPELTE EIGENWERTE
                                                                                 105
629
                                                                                 105
                     *END*E DOPPELT ODER MEHRFACH
630
                 *END*E GENAU EIN DOPPELTER EIGENWERT
                                                                                 105
631
                                                                                 105
632
                  'ELSE'
633
                     • IF •
                            ABS(EW(/2/) - EW(/3/)) < EPSI "THEN"BEGIN"
                                                                                 105
                               ABS(EW(/2/) - EW(/4/)) < EPSI "THEN" BEGIN"
                                                                                 106
635
                           TAU1 := (EN(/2/) + EN(/3/) + EN(/4/)) / 3;
                                                                                 106
637
                           DREI := "TRUE"
                                                                                 106
638
                        *END *E DREIFACH
                                                                                 106
639
                                 TAU1 := (EN(/2/) + EN(/3/)) / 2
                                                                                 106
640
                        *ELSE*
642
                     *END*E ZWEIFACH ODER DREIFACH
                                                                                 106
                     · ELSE •
                                                                                 106
643
                               ABS(EV(/3/) - EW(/4/)) < EPSI *THEN*
                                                                                 106
644
                           TAU1 := (EN(/3/) + EN(/4/)) / 2
                                                                                 106
645
                        *ELSE**BEGIN*
                                                                                 106
646
                                             "THEN "BEGIN"
                           *IF *
                                  ADKLEIN
                                                                                 107
648
                                                                                 107
                              K := L := 1;
650
                              *FOR* U:=2 *STEP* 1 *UNTIL* 4 *DO**BEGIN*
                                                                                 107
651
                                  *IF * ABS(L1 - EW(/U/)) < ABS(L1 - EW(/K/))
                                                                                 107
653
654
                                  "THEN " K := U;
                                                                                 107
655
                                  *IF * ABS(L2 - EW(/U/)) < ABS(L2 - EW(/L/))
                                                                                 107
                                  *THEN* L := U
                                                                                 107
656
                              *END *E DER EIGENWERTE NAHE DENEN VON A RRS
                                                                                 107
657
                              *IF * K = L *THEN**BEGIN*
                                                                                 107
658
                                  *IF * K = 1 *THEN*
                                                                                 107
                                                     L := 2
660
                                  *ELSE**IF * K = 4 *THEN* K := 3
                                                                                 108
662
                                        *ELSE**IF* ABS(L1 - EW(/K-1/))
                                                                                 108
665
                                                   < ABS(L2 - EW(/K+1/))
                                                                                 108
667
                                              "THEN" K := K - 1
                                                                                 108
667
                                              "ELSE" L := L + 1
                                                                                 108
668
                              *END*E DES FALLS K = L
                                                                                 108
670
                           *END*E EIGENWERTE NACHE BEI DENEN VON ARR
                                                                                 108
671
                           *ELSE ** BEGIN*
                                                                                 108
672
674
                              D1 := ABS(EW(/1/) - EW(/2/));
                                                                                 108
                              D2 := ABS(EV(/2/) - EV(/3/));
675
                                                                                 108
                              D3 := ABS(EV(/3/) - EV(/4/));
                                                                                 109
676
                                     ·IF ·
                                                                                 109
677
                                        D1 < D3
                                  · IF ·
                                                   *THEN**BEGIN*
                                                                                 109
679
                                     K := 1; L := 2
                                                                                 109
681
                                  *END*
                                                                                 109
683
                                  *ELSE * BEGIN*
                                                                                 109
684
                                                                                 109
686
                                     K := 3;
                                             L == 4
                                  *END*
                                                                                 109
688
                              "END"
                                                                                 109
689
                              'ELSE''IF'
                                            D2 < D3 "THEN" BEGIN"
                                                                                 109
690
```

```
110
693
                                   K = 23
                                             L := 3
                                                                              110
695
                                *END*
696
                                *ELSE * *BEGIN*
                                                                              110
                                                                              110
698
                                   K := 3; L := 4
                                                                              110
700
                                . END .
701
                          *END*E DER AM DICHTESTEN BENACHBARTEN EIGENWERTE;
                                                                              110
                                                                              110
702
                          •GOTO• TRANS
                       *END *E GETRENNTE EIGENWERTE;
                                                                             110
703
                                                                              110
704
                *IF* WIEDERHOLUNG *THEN*
704
                                             MEHRFACH == "TRUE"
                                                                              110
                                                                              111
706
                'ELSE'
                    *IF * ABS(TAU1) < 1-4 ! DREI
                                                   *THEN * GOTO * NEUVER
                                                                              111
707
                                                                              111
                    *ELSE**BEGIN*
709
                                                                              111
711
                       WIEDERHOLUNG := "TRUE";
712
                       EPSI := !-12;
                                                                              111
                       SHIFT := SHIFT + TAU1;
713
                                                                              111
                       *GOTO* POLYNOM
                                                                              111
714
                    • END • E START ZUR WIEDERHOLTEN POLYNOMBERECHNUNG;
                                                                              111
715
                                                                              111
716
     TRANS:
                 *IF * MEHREACH
                                 *THEN* *BEGIN*
                                                                              111
716
                    A11 := A11 - TAU1;
                                                                              112
719
                    A22 := A22 - TAU1;
                                                                              112
720
                                                                              112
                    A33 := A33 - TAU1;
721
                                                                              112
                    A44 := A44 - TAU1;
722
                                                                              112
                 *END*
723
                 *ELSE * BEGIN*
                                                                              112
724
                                                                              112
726
     726
                                                                            ×112
726
     ¥
                 SHIFT A := (A - EW(/K/)*I) * (A - EW(/L/)*I)
                                                                             ×112
726
     ¥
726
     726
                                                                              113
726
                    PLU := -EW(/K/) - EW(/L/);
                                                                              113
726
                    MAL := EV(/K/) \times EV(/L/);
                                                                              113
727
                    AF11 := A11**2 + A12**2 + A13**2 + A14**2 + PLU*A11+ MALJ113
728
                    AF22 := A12**2 + A22**2 + A23**2 + A24**2 + PLU*A22+ MAL$113
729
                    AF33 := A13**2 + A23**2 + A33**2 + A34**2 + PLU*A33+ MALJ113
730
                    AF44 := A14××2 + A24××2 + A34××2 + A44××2 + PLU×A44+ MAL1113
731
                    AF12 := (A11 + A22 + PLU) \times A12 + A13 \times A23 + A14 \times A24; 113
732
                    AF13 := (A11 + A33 + PLU) * A13 + A12 * A23 + A14 * A34; 114
733
                   AF14 := (A11 + A44 + PLU) × A14 + A12 × A24 + A13 × A34; 114
734
                   AF23 := (A22 + A33 + PLU) * A23 + A12 * A13 + A24 * A34; 114
735
736
                    AF24 := (A22 + A44 + PLU) \times A24 + A12 \times A14 + A23 \times A343 114
                    AF34 := (A33 + A44 + PLU) × A34 + A13 × A14 + A23 × A24; 114
737
                                                                              114
738
                                                                              114
                    A11 := AF11;
738
                                                                              114
739
                    A22 := AF22#
                   A33 := AF33;
                                                                              114
740
                                                                              114
                   A44 := AF44;
741
                                                                              115
                   A12 := A21 := AF12;
742
                                                                              115
                   A13 := A31 := AF13;
743
                   A14 := A41 := AF14;
                                                                              115
744
                                                                              115
                    A23 := A32 := AF23;
745
                                                                              115
746
                    A24 := A42 := AF24;
```

```
A34 := A43 := AF34
                                                                   115
747
                                                                   115
               *END *E DES ERSTEN SHIFTS
748
749
                                                                   115
           • END • E DES SYMMETRISCHEN FALLS
                                                                   115
749
750
           *ELSE ** BEGIN*
                                                                   115
                                                                   116
752
              PLU := (AF12**2 + AF13**2 + AF14**2 + AF23**2 + AF24**2 +
                                                                   116
752
753
                                                                   116
                      AF34××2)/23
              MAL := (PLU**2 - (AF12 * AF34
753
                                                                   116
                   116
754
                                                                   116
754
              *IF* MAL> -- 14 *THEN* *BEGIN*
                                                                   116
754
                                                                   116
756
   756
756
                                                                 ×116
756
               DIE BEIDEN PAARE KONJUGIERT KOMPLEXER EIGENWERTE
                                                                  ×117
    ¥
756
                                                                  *117
               SIND VERSCHIEDEN.
    ×
               BERECHNEN DER MATRIX AA = AF \times \times 2 + Z(1) \times I
756
                                                                  ×117
                                                                  *117
756
               (VERGL. 5 2.3)
                                                                  ×117
756
    756
                                                                   117
756
                                                                   117
                 MAL == SQRT(MAL);
756
                                                                   117
757
                 *IF * ADKLEIN *THEN* *BEGIN*
                    L1 := PLU - MAL;
                                                                   117
759
                    L2 := PLU + MAL;
                                                                   118
760
                    MAL := AF12 **25
                                                                   118
761
762
                    PLU := 'IF' ABS(L1 - MAL) ( ABS(L2 - MAL)
                                                                   118
                                                                   118
                          "THEN" L1
763
                                                                   118
763
                          *ELSE* L2
                • END• E EIGENVERTE DICHT BEI DEM VON A RR
                                                                   118
763
                                                                   118
764
                 *ELSE* PLU := PLU - MAL;
                                                                   118
764
                                                                   118
766
                                                                  118
                A11 := -(AF12**2 + AF13**2 + AF14**2 - PLU);
766
                A22 := -(AF12**2 + AF23**2 + AF24**2 - PLU);
                                                                  119
767
                A33 := -(AF13**2 + AF23**2 + AF34**2 - PLU);
                                                                  119
768
                A44 := -(AF14**2 + AF24**2 + AF34**2 - PLU);
                                                                  119
769
                 A21 := A12 := -(AF13 * AF23 + AF14 * AF24);
                                                                  119
770
                 A31 := A13 := AF12 * AF23 - AF14 * AF34;
                                                                   119
771
                 A41 := A14 := AF12 * AF24 + AF13 * AF34;
772
                                                                   119
                 A32 := A23 := -(AF12 * AF13 + AF24 * AF34);
                                                                   119
773
                 A42 := A24 := AF23 × AF34 - AF12 × AF14#
                                                                   119
774
                 A43 := A34 := -(AF13 * AF14 + AF23 * AF24)
                                                                   119
775
776
               "END"E DES FALLS MAL > 0
                                                                   119
               "ELSE" 'BEGIN'
                                                                   120
777
                                                                   120
779
    779
                                                                  *12C
779
               DIE BEIDEN PAARE KONJUGIERT KOMPLEXER EIGENWERTE
                                                                  ×120
779
                                                                  *120
779
               SIND GLEICH.
               AA WIRD NACH DEM ERSATZVERFAHREN BERECHNET
                                                                  *12C
779
                                                                  *12C
779
               (VERGL. 5 2.3)
                                                                  *120
779
    779
```

```
779
                                                                         121
779
                   DOP := "TRUE";
                                                                         121
                   A11 := A12 := A13 := A14 :=
                                                                         121
780
                   A21 := A22 :=
                                                                         121
781
                  A31 := A32 :=
                                     A34 :=
                                                                         121
781
                                          := 0;
                                                                        121
781
                  A41 := A42
                  MAL := "IF" MAL>O "THEN" SORT(MAL) "ELSE" O;
                                                                        121
781
                  PLU := SQRT(PLU - MAL);
                                                                        121
782
                                                                        121
783
                  *IF * AF12 × AF34 < O *THEN* *BEGIN*
                                                                         121
783
                     VZW := "TRUE";
                                                                         122
785
                     AF34 := (AF34 - AF12) / 2;
                                                                         122
786
                     AF24 := (AF24 + AF13) / 2;
                                                                         122
787
                     AF14 := (AF14 - AF23) / 2
                                                                        122
788
                  • END•E DES FALLS VERSCHIEDENER VORZEICHEN
                                                                        122
789
                                                                        122
790
                  *ELSE**BEGIN*
792
                     VZW := "FALSE";
                                                                         122
                     AF34 := (AF34 + AF12) / 2
                                                                         122
793
                     AF24 := (AF24 - AF13) / 2;
                                                                         122
794
                     AF14 := (AF14 + AF23) / 2
                                                                         122
795
                  *END*E DES FALLS GLEICHER VORZEICHEN;
                                                                         123
796
                                                                         123
797
                                                                         123
797
                  A23 := AF24 * AF34;
                                                                         123
798
                  A24 := AF14 * PLU;
                  A33 := AF14**2 + AF34**2;
                                                                         123
799
                  A43 := AF14 * AF24;
                                                                         123
800
                                                                         123
801
                  A44 := -AF34 * PLU;
                                                                         123
802
                  *IF * VZW *THEN**BEGIN*
                                                                         123
802
                                                                         123
                     A24 := -A24;
804
                     A43 := -A43
                                                                         124
805
                   •END•E DES VORZEICHENVECHSELS
                                                                         124
806
                                                                         124
                *END *E DES FALLS MAL = 0
807
                                                                         124
808
             • END • E DES SCHIEFSYMMETRISCHEN FALLS;
                                                                         124
808
809
                                                                         124
809
             •IF • DOP
                        *THEN**BEGIN*
                                                                         124
                                                                         124
811
    811
                                                                       *124
811
                VERTAUSCHUNG DER LETZTEN MIT DER GROESSTEN SPALTE
                                                                        *125
811
                                                                        ×125
811
    811
                                                                         125
811
                                                                         125
811
                NV4 == NV4 + 1;
                NO1 := A11**2 + A21**2 + A31**2 + A41**2;
                                                                         125
812
               NO2 := A12**2 + A22**2 + A32**2 + A42**2;
                                                                         125
813
               NO3 := A13**2 + A23**2 + A33**2 + A43**2;
                                                                         125
814
               NO4 := A14**2 + A24**2 + A34**2 + A44**2;
                                                                         125
815
               SP := "IF" NO1>NO2 & NO1>NO3 & NO1>NO4 "THEN" 1
                                                                        125
816
                          *ELSE* *IF* NO2>NO3 & NO2>NO4 *THEN* 2
                                                                        126
817
                                 *ELSE* *IF* NO3>NO4 *THEN* 3 *ELSE* 4;
                                                                         126
817
                                                                         126
817
              "IF" SP=1 "THEN" "BEGIN"
                                                                         126
817
                                                                         126
                   KA := N01;
819
```

COMMENT BERECHNEN VON VEK4 = P\$

```
132
885
            VEK41 := (VE41 * A11 + VE42 * A21
                                                                     132
885
                     + VE43 * A31 + VE44 * A41)/KA;
                                                                     132
886
            VEK42 := (VE41 * A12 + VE42 * A22
886
                                                                     132
                     + VE43 * A32 + VE44 * A42)/KA1
                                                                     132
887
            VEK43 := (VE41 * A13 + VE42 * A23
                                                                     132
887
                     + VE43 \times A33 + VE44 \times A43)/KA1
                                                                     132
888
            VEK44 := (VE41 * A14 + VE42 * A24
                                                                     132
888
                     + VE43 × A34 + VE44 × A44)/KA$
                                                                     132
889
889
                                                                     132
                 BERECHNEN VON A := Q \times A = A - U \times PT
                                                                     133
889
    *COMMENT *
                                                                     133
889
            A11 := A11 - VE41 * VEK413
                                                                     133
889
                                                                     133
890
            A12 := A12 - VE41 * VEK423
            A13 := A13 - VE41 * VEK43;
                                                                     133
891
            A21 := A21 - VE42 * VEK41;
                                                                     133
892
            A22 := A22 - VE42 * VEK423
                                                                     133
893
            A23 := A23 - VE42 * VEK43;
                                                                     133
894
                                                                     133
            A31 := A31 - VE43 * VEK41;
895
896
            A32 := A32 - VE43 * VEK42;
                                                                     133
897
            A33 := A33 - VE43 * VEK43;
                                                                     134
898
                                                                     134
898
                 BERECHNEN VON VEK4 = U(SCHLANGE);
                                                                     134
    *COMMENT*
                                                                     134
898
                                                                     134
898
            VEK41 := VE41/KA;
899
            VEK42 := VE42/KA;
                                                                     134
                                                                     134
900
            VEK43 := VE43/KA;
                                                                     134
901
            VEK44 := VE44/KA3
                                                                     134
902
902
    ×135
902
    ¥
                   ZEILENTRANSFORMATIONEN MIT Q4
                                                                    ×135
902
                                                                    ×135
902
    902
                                                                     135
902
               *FOR*U:=1*STEP*1*UNTIL*N*DO**BEGIN*
                                                                     135
902
                  OUA := VE41*A(/R21,U/) + VE42*A(/R2,U/)
                                                                     135
904
                      + VE43*A(/S21*U/) + VE44*A(/S2*U/);
                                                                     135
905
                  A(/R21,U/) := A(/R21,U/) - VEK41*QUA;
                                                                     135
905
                  A(/R2,U/) := A(/R2,U/) - VEK42*QUA;
                                                                     135
906
                                                                     136
907
                  A(/S21_3U/) := A(/S21_3U/) - VEK43*QUA;
                  A(/S2,U/) := A(/S2,U/) - VEK44*QUA;
                                                                     136
908
                                                                     136
                  *IF * LINKS *THEN**BEGIN*
909
                                                                     136
911
    911
                                                                    *136
911
    ×
                                                                    *136
911
               BERECHNEN DER LINKSEIGENVEKTÜREN
                                                                    ×136
911
    911
911
                                                                     136
                    QUA := VE41*TI(/R21;U/) + VE42*TI(/R2;U/)
                                                                     137
911
                         + VE43*TI(/S21,U/) + VE44*TI(/S2,U/);
                                                                     137
912
                    TI(/R21_JU/) := TI(/R21_JU/) - VEK41*QUA;
                                                                     137
912
                    TI(/R2.u/) := TI(/R2.u/) - VEK42*QUA;
                                                                     137
913
                                                                     137
                    TI(/S21,U/) := TI(/S21,U/) - YEK43*QUA;
914
```

```
915
                                        TI(/S2,U/) := TI(/S2,U/) - VEK44*QUA
                                                                                                                                        137
916
                                   *END *
                                                                                                                                        137
917
                              *END*E DER ZEILENTRANSFORMATION;
                                                                                                                                       137
918
                                                                                                                                       137
918
         918
                                                                                                                                     *138
918
                                    SPALTENTRANSFORMATION MIT 04
                                                                                                                                      *138
918
                                                                                                                                      ×138
918
         918
                                                                                                                                        138
918
                            *FOR*U:=1*STEP*1*UNTIL*N*DO**BEGIN*
                                                                                                                                       138
920
                                   OUA := VE41*A(/U,R21/) + VE42*A(/U,R2/)
                                                                                                                                       138
921
                                            + VE43*A(/U)S21/) + VE44* A(/U)S2/);
                                                                                                                                       138
                                  A(/U_1R21/) := A(/U_1R21/) - VEK41*QUA;
921
                                                                                                                                       138
922
                                  A(/U_{\bullet}R2/) := A(/U_{\bullet}R2/) - VEK42*QUA;
                                                                                                                                       138
                                   A(/U_{3}S21/) := A(/U_{3}S21/) - VEK43*QUA;
923
                                                                                                                                       139
924
                                   A(/U,S2/) := A(/U,S2/) - VEK44×QUA;
                                                                                                                                       139
925
                                   *IF * RECHTS *THEN**BEGIN*
                                                                                                                                       139
927
927
         927
                                                                                                                                     ×139
                             BERECHNEN DER RECHTSEIGENVEKTOREN
927
                                                                                                                                      ×139
927
                                                                                                                                      ×139
927
         25. Гайыны кайыны кайы
927
                                                                                                                                        139
927
                                        QUA := VE41*T(/U_1R21/) + VE42*T(/U_1R2/)
                                                                                                                                       140
                                                 + VE43*T(/U,S21/) + VE44* T(/U,S2/);
928
                                                                                                                                       140
928
                                        T(/U_JR21/) := T(/U_JR21/) - VEK41*QUAJ
                                                                                                                                       140
929
                                        T(/U_{\bullet}R2/) := T(/U_{\bullet}R2/) - VEK42*GUA;
                                                                                                                                       140
                                        T(/U,S21/) = T(/U,S21/) - VEK43*QUA;
930
                                                                                                                                       140
931
                                        T(/U_3S2/) := T(/U_3S2/) - VEK44*QUA
                                                                                                                                       140
932
                                   'END'
                                                                                                                                       140
                            PENDPE DER SPALTENTRANSFORMATION:
933
                                                                                                                                       140
934
                                                                                                                                       140
                             "IF" DREI "THEN" BEGIN"
934
                                                                                                                                       140
936
                                   "IF" DOP "THEN" BEGIN"
                                                                                                                                       141
938
                                                                                                                                       141
938
         938
                                                                                                                                     ×141
                                      VERTAUSCHUNG DER DRITTEN MIT DER GROESSTEN SPALTE,
938
                                                                                                                                     ×141
938
                                     NUR, WENN KEIN DOPPELTES PAAR KOMPLEXER EIGENWERTE
                                                                                                                                     ×141
                                      VORLIEGT (DOP = "FALSE" !) .
938
                                                                                                                                      ×141
                                      BEZW. KEIN DREIFACHER EIGENWERT VORLIEGT
938
                                                                                                                                     ×141
938
                                      (DREI = "FALSE" !) .
                                                                                                                                     ×141
938
                                                                                                                                     ×141
938
         938
                                                                                                                                       142
938
                                        NV3 := NV3 + 1;
                                                                                                                                       142
939
                                        NO1 := A11**2 + A21**2 + A31**2;
                                                                                                                                       142
940
                                        N02 := A12**2 + A22**2 + A32**2;
                                                                                                                                       142
941
                                        NO3 := A13**2 + A23**2 + A33**2;
                                                                                                                                       142
942
                                                                                                                                       142
                                       SP := "IF" N01>N02 & N01>N03 "THEN" 1
942
                                                                                                                                       142
                                                            *ELSE**IF * NO2>NO3 *THEN* 2 *ELSE* 3;
943
                                                                                                                                       142
943
                                                                                                                                       142
```

```
*IF * SP=1 *THEN* *BEGIN*
                                                                 143
943
945
                      KA := N01;
                                                                 143
                      TAU1 := A11; A11 := A13; A13 := TAU1;
946
                                                                 143
                      TAU1 := A21; A21 := A23; A23 := TAU1;
949
                                                                 143
                      TAU1 := A31; A31 := A33; A33 := TAU1
952
                                                                 143
                                                                 143
955
                   ·END ·
956
                   *ELSE*
                                                                 143
                   "IF " SP=2 "THEN" "BEGIN"
957
                                                                 143
                                                                 143
959
                      KA := NO2;
                      TAU1 := A12; A12 := A13; A13 := TAU1;
                                                                 143
960
                     TAUT == A12, A12 == A23, A23 := TAU1;
                                                                 144
963
                     TAU1 := A32; A32 := A33; A33 := TAU1
966
                                                                 144
                  ·END ·
                                                                 144
969
970
                   *ELSE*
                                                                 144
                     KA == NO3
                                                                 144
971
                "END'E DES FALLS DOP = "FLASE"
                                                                 144
972
973
                 'ELSE'
                                                                 144
                   KA := A13**2 + A23**2 + A33**2;
974
                                                                 144
975
                                                                 144
975
    975
                                                                ×145
                  TRANSFORMATION MIT Q3
                                                                *145
975
                                                                * 145
975
    975
                                                                 145
975
                *IF * KA < 1-32 *THEN * GOTO * ENDE;
                                                                 145
975
                                                                 145
977
                                                                 145
977
                C := SORT(KA);
978
                KA := KA + C \times ABS(A33)
                                                                 145
                                                                 145
979
                NT3 := NT3 + 13
                                                                 146
                VE31 := A13;
980
                                                                 146
                VE32 := A23;
981
                VE33 := A33 + SIGN(A33) * C;
                                                                 146
982
                *1F * A33=0 *THEN * VE33 := VE33 + C;
                                                                 146
983
                                                                 146
985
                VEK31 := VE31/KA;
985
                                                                 146
                VEK32 := VE32/KA;
                                                                 146
986
987
                VEK33 := VE33/KA;
                                                                 146
988
                                                                 146
                •IF • TAOKLEIN & TOOP • THEN • BEGIN •
                                                                 146
988
                                                                 147
990
    990
990
                                                                ×147
990
              BERECHNEN VON Q = (I - 2 W*WT) * (I - 2 W*VT)
                                                                *147
                                                                *147
990
    990
990
                                                                 147
                   SPUR := VEK41*VE31 + VEK42*VE32 + VEK43*VE33;
                                                                 147
990
                   U1 := VEK41 - SPUR * VEK311
                                                                 147
991
                   U2 := VEK42 - SPUR * VEK32;
992
                                                                 147
                   U3 := VEK43 - SPUR * VEK33;
                                                                 148
993
                                                                 148
                   U12 := VE31 * VEK32;
994
995
                  U13 := VE31 * VEK33;
                                                                 148
                                                                 148
996
                   U23 := VE32 * VEK33;
                                                                 148
997
```

```
NQ1 := (1 - VE41*U1 - VE31*VEK31) **2
997
                                                             148
                      + (1 - VE42*U2 - VE32*VEK32) **2
                                                             148
998
                      + (VE41*U2 + U12) **2 + (VE42*U1 + U12) **23
998
                                                             148
                  NO2 := (1 - VE43*U3 - VE33*VEK33) **2
                                                             148
998
                      + (1 - VE44*VEK44) **2
999
                                                             148
                      + (VE43*VEK44) **2 + (VE44*U3) **2;
                                                             149
999
                  NO3 := (VE41*U3 + U13) **2 + (VE41*VEK44) **2
                                                             149
999
                      + (YE42*U3 + U23) **2 + (YE42*VEK44) **23
                                                             149
1000
                  NQ4 := (VE43*U1 + U13) **2 + (VE43*U2 + U23) **2
                                                             149
1000
                                                             149
1001
                      + (¥E44*U1) **2 + (YE44*U2) **23
                                                             145
1001
                  *IF * NO1 + NO2 < NO3 + NO4 *THEN**BEGIN*
                                                             149
1001
                                                             145
1003
    1003
1003
1003
                   TRANSFORMATION DER DER GROBEN MATRIX A
                                                            *150
1003
                   MIT DER HOUSEHOLDER - MATRIX Q * P
                                                            ×15[
1003
                                                            *15(
    1003
1003
    *15(
1003
                         ZEILENTRANSFORMATION
                                                            ×150
1003
                                                            *15[
1003
    1003
                                                             150
1003
                                                             151
1003
                     NP := NP + 1;
                     *FOR*U:=1*STEP*1*UNTIL*N*DO**BEGIN*
                                                             151
1004
                                                             151
                       QUA := VE31*A(/R21,U/) + VE32*A(/R2,U/)
1006
                                                             151
1007
                           + VE33*A(/S21;U/);
1007
                       AR11 := A(/R21, U/);
                                                             151
                                                             151
                       ARI := A(/R2, U/);
1008
                       A(/R21_3U/) := A(/S21_3U/) - VEK33*QUA;
                                                             151
1009
                       A(/R2,U/) := A(/S2,U/);
                                                             151
1010
                       A(/S21.U/) := AR1I - VEK31*QUA3
                                                             151
1011
                       A(/S2,U/) := ARI - VEK32*QUA;
                                                             151
1012
                                                            152
                       *IF * LINKS *THEN**BEGIN*
1013
                                                             152
1015
    1015
1015
                                                            ×152
                      BERECHNEN DER LINKSEIGENVEKTOREN
                                                            ×152
1015
                                                            ×152
1015
    1015
                                                             152
1015
                                                             152
1015
                         OUA := VE31*TI(/R21;U/) + VE32*TI(/R2;U/)
                             + VE33*TI(/S21;U/);
                                                             152
1016
                         AR1I := TI(/R21, U/);
                                                             153
1016
                         ARI := TI(/R2, U/);
                                                             153
1017
                         TI(/R21,U/) := TI(/S21,U/) - VEK33*QUA;
                                                             153
1018
                         TI(/R2,U/) := TI(/S2,U/);
                                                             153
1019
                         TI(/S21.JU/) := AR1I - VEK31*QUA;
                                                             153
1020
                                                             153
                         TI(/S2,U/) := ARI - VEK32*QUA
1021
                                                             153
                       "END"
1022
                     PEND E DER ZEILENTRANSFURMATION;
                                                             153
1023
                                                             153
1024
    1024
```

```
×154
1024
                            SPALTENTRANSFORMATION
                                                             ×154
1024
                                                             ×154
1024
    ¥
    1024
1024
                     *FOR *U:=1*STEP*1*UNTIL*N*DO**BEGIN*
                                                              154
1024
                                                             154
                       QUA := VE31*A(/UJR21/) + VE32*A(/UJR2/)
1026
                                                             154
                           + VE33*A(/U,S21/);
1027
                                                              154
                       AR11 == A(/U, R21/);
1027
                                                              154
                       ARI := A(/U, R2/);
1028
                                                              155
1029
                       A(/U)R21/) := A(/U)S21/) - VEK33*QUAS
                       A(/U_2R2/) := A(/U_2S2/)
                                                              155
1030
                       A(/U)S21/) := AR1I - VEK31*QUAJ
                                                              155
1031
                       A(/U,S2/) := ARI - VEK32*QUA;
                                                              155
1032
                       *IF* RECHTS *THEN**BEGIN*
                                                              155
1033
                                                              155
1035
    1035
                                                             *155
1035
                      BERECHNEN DER RECHTSEIGENVEKTOREN
                                                             ×155
1035
    ×
                                                             *155
1035
    1035
                                                              156
1035
                          QUA := VE31*T(/U,R21/) + VE32*T(/U,R2/)
                                                              156
1035
                                                              156
                              + VE33*T(/U,S21/);
1036
                                                              156
                          AR1I := T(/U, R21/);
1036
                                                              156
                          ARI := T(/U, R2/);
1037
                          T(/U,R21/) := T(/U,S21/) - VEK33*QUA;
                                                              156
1038
                                                              156
                          T(/U_{3}R2/) := T(/U_{3}S2/);
1039
                                                              156
                          T(/U_sS21/) := AR1I - VEK31*QUA;
1040
                          T(/U,S2/) := ARI - VEK32*QUA
                                                              156
1041
                                                              157
                       "END"
1042
                                                              157
                     *END *E DER SPLATENTRANSFORMATION
1043
                   *END*E DER TRANSFORMATION MIT Q * P
                                                              157
1044
                                                              157
1045
                                                              157
                   *ELSE**GOTO* TRANSQ
1045
                *END*E DES FALLS "ADKLEIN & " DOP
                                                              157
1047
                                                              157
                *FLSE**BEGIN*
1048
                                                              157
1050
    • COMMENT • ************************
1050
                                                             ×157
1050
                                                             *158
               TRANSFORMATION DER GROBEN MATRIX A
1050
                                         Q
                                                             ×158
              MIT DER HOUSEHOLDER - MATRIX
1050
                                                             *158
1050
    1050
    1050
                                                             *158
1050
                      ZEILENTRANSFORMATION
                                                             *158
1050
1050
    1050
                                                              158
1050
                                                              159
1050
    TRANSO:
                   *FOR*U:=1*STEP*1*UNTIL*N*DO**BEGIN*
                                                              159
1050
                                                              159
                     QUA := YE31 × A (/R21 • U/) + YE32 × A (/R2 • U/)
1053
                                                              159
1054
                         + VE33*A(/S21,U/);
                     A(/R21,U/) := A(/R21,U/) - VEK31*QUA;
                                                              159
1054
```

```
A(/R2*U/) := A(/R2*U/) - VEK32*QUA*
                                                                 155
1055
                      A(/S21,U/) := A(/S21,U/) - VEK33*QUA;
                                                                 159
1056
                      *IF * LINKS *THEN ** BEGIN *
1057
                                                                 159
1059
                                                                 155
     1059
                                                                ×16(
1059
                     BERECHNEN DER LINKSEIGENVEKTOREN
                                                                *16(
1059
     ¥
                                                                *16E
1059
1059
     1059
                                                                 160
                         QUA := VE31*TI(/R21,U/) + VE32*TI(/R2,U/)
                                                                 160
1059
                                                                 160
                             + VE33*TI(/S21,U/);
1060
                         TI(/R21,U/) := TI(/R21,U/) - VEK31*QUA;
                                                                 160
1060
                         TI(/R2_3U/) := TI(/R2_3U/) - VEK32*QUA3
                                                                 160
1061
1062
                         TI(/S21,U/) := TI(/S21,U/) - VEK33*QUA
                                                                 16[
                      *END*
                                                                 161
1063
                    *END*E DER ZEILENTRANSFORMATIONS
                                                                 161
1064
1065
                                                                 161
     1065
1065
                                                                *161
                           SPALTENTRANSFORMATION
                                                                ×161
1065
                                                                *161
1065
     1065
1065
                                                                 161
                   *FOR*U:=1*STEP*1*UNTIL*N*DO**BEGIN*
                                                                 161
1065
                                                                 162
                      QUA := VE31 \times A(/U_1R21/) + VE32 \times A(/U_1R2/)
1067
                                                                 162
                          + VE33*A(/U,S21/);
1068
                      A(/U,R21/) := A(/U,R21/) - VEK31*QUA;
                                                                 162
1068
                      A(/U_R2/) := A(/U_R2/) - VEK32*QUA;
1069
                                                                 162
                      A(/U,S21/) := A(/U,S21/) - VEK33*QUA;
                                                                 162
1070
                      *IF * RECHTS *THEN **BEGIN*
                                                                 162
1071
                                                                 162
1073
1073
     *162
1073
                      BERECHNEN DER RECHTSEIGENVEKTOREN
                                                                *162
1073
     ×
                                                                ×163
1073
     16.
1073
                                                                 163
1073
                         QUA := VE31*T(/U,R21/) + VE32*T(/U,R2/)
                                                                 163
1073
                             + VE33*T(/UJS21/);
                                                                 163
1074
                                                                 163
                         T(/U_JR21/) := T(/U_JR21/) - VEK31*QUA;
1074
                         T(/UsR2/) := T(/UsR2/) - VEK32*QUA;
                                                                 163
1075
                         T(/U,S21/) := T(/U,S21/) - VEK33*QUA
                                                                 163
1076
                                                                 163
1077
                      · END ·
                    "END'E DER SPLATENTRANSFORMATION
                                                                 163
1078
                 *END*E DER TRANSFORMATION MIT Q
                                                                 164
1079
               "END'E DES FALLS DREI = "FALSE"
                                                                 164
1080
            *END *E EINER TRANSFORMATION;
                                                                 164
1081
                                                                 164
1082
     ENDE:
                                                                 164
            *END*E DER SCHLEIFE UEBER S
1082
          *END*E DER SCHLEIFE UEBER R
                                                                 164
1084
                                                                 164
1085
       *END*E DER PROZEDUR BLOCKTRANS7;
                                                                 164
1086
                                                                 164
1086
       F := 0;
                                                                 164
1086
```

```
*IF * N *MOD * 2 = O *THEN *BEGIN *
                                                                      165
1087
          165
1089
        *END*E DER IF - ANWEISUNG;
                                                                      165
1092
                                                                       165
1093
        SETTIM;
                                                                      165
1094
        *IF * LINKS *THEN * BEGIN*
                                                                      165
1094
                                                                      165
1096
     1096
                                                                      *165
1096
             INVERSE TRANSFORMATIONSMATRIX ALS EINHEITSMATRIX SETZEN
1096
                                                                     *165
                                                                      *166
1096
     1096
                                                                      166
1096
                                                                      166
1096
          *FOR* I:=1 *STEP* 1 *UNTIL* N *DO**BEGIN*
             TI(/I, I/) := 1;
1098
                                                                      166
             *FOR* J:= I+1 *STEP* 1 *UNTIL* N *DO*
                                                                      166
1099
                                                                      166
                TI(/I, J/) := TI(/J, I/) := 0
1100
                                                                      166
          *END *
1101
                                                                      166
        *END*;
1102
                                                                      166
1103
        PIFF
             RECHTS *THEN**BEGIN*
                                                                      167
1105
     « COMMENT» ***************************
1105
                                                                      × 167
1105
             TRANSFORMATONSMATRIX ALS EINHEITSMATRIX SETZEN
                                                                      ×167
1105
                                                                      ×167
1105
     1105
                                                                      167
1105
          *FOR* I:=1 *STEP* 1 *UNTIL* N *DO**BEGIN*
                                                                      167
1105
                                                                      167
1107
             T(/I, I/) := 1;
             •FOR• J:= I+1 *STEP* 1 *UNTIL* N *DO*
                                                                      167
1108
                T(/I, J/) := T(/J, I/) := 0
                                                                      168
1109
           · END ·
                                                                      168
1110
                                                                      168
1111
        "END";
                                                                      168
1112
                                                                      168
        LAMBDA := 1-2;
1112
                                                                      168
        P := N/2;
1113
                                                                      168
        P1 := P - 1;
1114
                                                                      168
        EP := EPSILON * 4;
1115
        EPS := SQRT(EP) * 1-2;
                                                                      168
1116
                                                                      168
        MNORM := MORTH := MDIAG := 0;
1117
                                                                      165
        NORM(P, A, NA, NAO);
1118
        PUTRDAT(1, 0, 7, "('||A|| = ")', NA, 0);
1119
                                                                      169
        PUTRDAT(3, 0, 18, *(*fiail = *)*, NA, 0);
                                                                      169
1120
        PUTRDAT(1: 0: 2: *(*:
                                                                      169
                              S(A) = 1)1 \cdot NAO \cdot 1);
1121
                                                                      169
1122
        PUTRDAT(3, 0, 2, *(*)
                              S(A) = 1)1, NAO, 3);
                                                                      169
1123
        *FOR* L:=1 *STEP* 1 *UNTIL* 50 *DO**BEGIN*
                                                                      169
1123
          ITSTEHT := !TRUE";
                                                                      169
1125
          NSY := NSCHIE := NT4 := NT3 := NV4 := NV3 := 0;
                                                                      169
1126
                                                                    169
          MM := MMM := MD := O;
1127
                                                                      170
1128
          AUKLEIN := NAO < 1-2;
                                                                      170
1129
                                                                      170
          Z1 := ASKTIM#
1129
                                                                      170
          BLOCKTRANS;
1130
                                                                      170
1131
          EBERLEIN:
```

```
170
1132
            Z2 := ASKTIMs
            NORM(P) A, NAD NAD);
                                                                                  170
1133
1134
            AFIX(1, 2, 0, L);
                                                                                  170
                                                                                  170
1135
            AFIX(3, 12, 0, L);
            MNORM := MNORM + MM:
                                                                                  170
1136
                                                                                  171
            MORTH := MORTH + MMM;
1137
                                                                                  171
1138
            MDIAG := MDIAG + MD;
                                                                                  171
1139
            PUTDATA(3, 0, 3, *(*BLOCKNORMREDUZIERUNG :
                                                                                  171
                                                           *) ** MM);
1139
            PUTLIST(3, 0, 2, *(*) DIAGONAL : *)*, MD,
                                                                                  171
1140
                              • ( • TRANSFORMATIONEN DURCHGEFUEHRT)
                                                                     *) *);
                                                                                  171
1141
            PUTDATA(3, 1, 18, '('ORTHOGONAL : ')', MMM);
1141
                                                                                  171
            PUTLIST(3, 0, 1, *(*TRANSFORMATIONEN DURCHGEFUEHRT)
                                                                                  171
1142
                                                                                  171
               , Z2-Z1, *( * ZENTISEKUNDEN GEBRAUCHT*) *);
1143
                                                                                  171
1143
            PUTRDAT(1, 0, 2, "("||A|| = ")", NA, 0);
                                                                                  172
1143
            PUTRDAT(3, 1, 18, *(*!!A!! = *)*, NA, 0);
                                                                                  172
1144
            PUTRDAT(1, 0, 2, *(*,
                                      S(A) = 1)1, NAO, O);
                                                                                  172
1145
                                                                                  172
            PUTRDAT(3, 0, 2, 1(1)
                                      S(A) = 1)1, NAO, O);
1146
                                                                                  172
1147
            "IF" RECHTS & LINKS "THEN" BEGIN"
                                                                                  172
1147
               K := KOND(N \cdot T \cdot TI)
                                                                                  172
1149
               PUTRDAT(1, 1, 10, "("KOND(T) = ")", K, 0);
                                                                                  172
1150
                                                                                  172
               PUTRDAT(3, 0, 2, 0, 0, KOND(T) = 0, K, 0)
1151
                                                                                  172
1152
            "END";
                                                                                  173
1153
                                                                                  173
1153
                                                                                  173
1153
            SSUM := ASUM := 0;
                                                                                  173
            *FOR* R:=N-3 *STEP* -2 *UNTIL* 1 *DO*
1154
1155
            *FOR* S:=R+2 *STEP* 1 *UNTIL* N *DO**BEGIN*
                                                                                  173
               ASUM := ASUM + (A(/R_JS/) - A(/S_JR/)) **2
                                                                                  173
1157
                                                                                  173
                             + (A(/R+1)S/) - A(/S)R+1/)) **2;
1158
                                                                                  173
               SSUM := SSUM + (A(/R_sS/) + A(/S_sR/)) **2
1158
                             + (A(/R+1,S/) + A(/S,R+1/)) **2
                                                                                  173
1159
                                                                                  173
1159
            *END *E DER NORMBERECHNUNG;
                                                                                  174
1160
                                                                                  174
            ASUM := SQRT(ASUM);
1160
            SSUM := SQRT(SSUM);
                                                                                  174
1161
                                                                                  174
1162
            PUTRDATA(1, 1, 10, ((S(A+) = 1)), SSUM, 0);
                                                                                  174
1162
            PUTRDATA(3, 1, 18, *(*S(A+) = *)*, SSUM, 0);
1163
                                                                                  174
            PUTRDATA(1, 0, 5, ((S(A-) = 1)), ASUM, 0);
                                                                                  174
1164
            PUTRDATA(3, 0, 5, (S(A-) = P), ASUN, 0);
                                                                                  174
1165
                                                                                  174
1166
            PUTDATA(3, 1, 18, *(*NSCHIE = *)*, NSCHIE);
                                                                                  174
1166
            PUTDATA(3, 0, 4, ('NSY = ')', NSY);
                                                                                  175
1167
            PUTDATA(3, 0, 4, *(*NV4 = *)*, NV4);
                                                                                  175
1168
            PUTDATA(3, U, 4, "("NT4 = ")", NT4);
                                                                                  175
1169
            PUTDATA(3, 0, 4, *(*NV3 = *)*, NV3);
                                                                                  175
1170
                                                                                  175
1171
            PUTDATA(3, 0, 4, ('NT3 = )', NT3);
                                                                                  175
1172
                                                                                  175
            KOMMUTATOR(P. A. SCA. NCA. NCD);
1172
                                                                                  175
1173
                                                                                  175
1173
            PUTRDAT(1, 1, 10, '('S(C(A)) = ')', SCA, 0);
                                                                                  175
1174
            PUTRDAT(3, 1, 18, *(*S(C(A)) = *)*, SCA, 0);
```

```
1175
           PUTRDAT(1, 0, 2, 1(1,
                                    llc(A)ll = ")" NCA D);
                                                                            176
           PUTRDAT(3, 0, 2, "(",
                                    IIC(A)II = ")" NEAD O)
                                                                            176
1176
           PUTRDAT(1, 1, 10, *(*!!C(D)!! = *)*, NCD, 0);
                                                                            176
1177
                                    llc(D)ll = ")" NCD = 0);
                                                                            176
1178
           PUTRDAT(3, 0, 2, *(')
                                                                            176
1179
                                                                            176
1179
           LINE(1, 1);
                                                                            176
           LINE(3, 2);
1180
                                                                            176
1181
                                                                            176
           *IF * ITSTEHT *THEN * *BEGIN *
1181
                                                                            176
              Z := ASKTIM;
1183
                                                                            177
              F := 3;
1184
              *GOTO* FMARKE
                                                                            177
1185
                                                                            177
           *END *E DER IF-ANWEISUNG;
1186
                                                                            177
1187
      1187
                                                                           ×177
1187
                                                                           ×177
1187
              BERECHNEN DES KONVERGENZKRITERIUMS
     ×
                                                                           *177
1187
      1187
                                                                            177
1187
           *FOR* R:=1 *STEP* 1 *UNTIL* P *DO**BEGIN*
                                                                            178
1187
                                                                            178
              R1 := R - 15
1189
              *FOR* S:=1 *STEP* 1 *UNTIL* R1,
                                                                            178
1190
                       R+1 *STEP* 1 *UNTIL* P *DO*
                                                                            178
1191
                       ABS(A(/2*R-1) 2*S-1/)) > EPS
                                                                            178
                 * I F *
1191
1192
                                                                            178
                       ABS(A(/2*R-1, 2*S /)) > EPS
                                  , 2×S-1/)) > EPS
                                                                            178
1192
                       ABS(A(/2*R
                                                    *THEN**GOTO* WEITER;
                       ABS(A(/2*R \rightarrow 2*S /)) > EPS
                                                                            178
1192
                                                                            178
           *END *E DES KONVERGENZTESTS;
1193
                                                                            178
1194
           "GOTO" ENDE;
                                                                            179
1195
     WEITER:
                                                                            179
1195
           ·IF ·
                2×Z2 - Z1 > LIM "THEN" BEGIN"
                                                                            179
              Z := Z2;
1198
              F := 6;
                                                                            179
1199
              GOTO FMARKE
                                                                            179
1200
                                                                            179
            • END • E DER ZEITUEBERSCHREITUNG
1201
                                                                            179
1202
        *END*E DER ITERATIONEN;
                                                                            179
1203
        L := L - 1;
                                                                            179
1204
        F := 2;
                                                                            179
1205
        Z := ASKTIM;
                                                                            180
1206
        *GOTO* FMARKE*
                                                                            180
     ENDE: Z := ASKTIM;
1207
        BLANK(3, 2); LINE(3, 2); BLANK(3, 10);
                                                                            180
1209
                                                                            180
1212
        OUTSTRING(3, *(*GESAMT:*)*);
        PUTDATA(3, 0, 3, '('BLOCKNORMREDUZIERUNG :
                                                       ·) · . MNORM);
                                                                            180
1213
        PUTLIST(3, 0, 2, '(', DIAGONAL: ')', MDIAG,
                                                                            180
1214
                         *( * TRANSFORMATIONEN DURCHGEFUEHRT .*) *);
                                                                            180
1215
        PUTLIST(3, 1, 20, (ORTHOGONAL:
                                                                            180
1215
                                               ·)' MORTH
                         *(* TRANSFORMATIONEN DURCHGEFUEHRT.*)*)
                                                                            180
1216
                                                                            180
      *END *E DER PROZEDUR EIGEN7;
1216
```

PROZEDUR 'EIGENWERT 7, VERSION 5.6" NORMAL BEENDET

BEISPIEL 34, MATRIX 'B' DER DIMENSION 100, MIT TAU =

DIE MATRIX WURDE NORMIERT 100 DIMENSION DER MATRIX : 90 ZENTISEKUNDEN ZUM BALANCIEREN GEBRAUCHTE RECHENZEIT : BASIS

ES WURDEN 10 ZYKLEN DURCHGEFUEHRT, MIT 9568 NORMREDUZEIRENDEN UND 9403 ORTHOGONALEN TRANSFORMATIONEN

GESAMT 101866 ZENTISEKUNDEN VERBRAUCHTE RECHENZEIT : 101776 ZENTISEKUNDEN

396 • 9385917704082 KONDITION DER TRANSFORMATIONSMATRIX 42 23 +8 07 46 65 47 17 2 KONDITION DER TRANSFORMATIONSMATRIX (INKLUSIV BALANCE):

+.669227408093--13 NORM DES AUSSERBLOCKDIAGONALEN TEILS DER MATRIX:

EIGENVERTE DER MATRIX

REAL TEIL IMAGINAERTEIL

71.00390204340165	-71.60390204340165
499999999999537	49999999999997

3 • 101045615577706	-3.101045615577706
499999999999321	-•49999999999321

42.62902300472312	-42.62902300472312
982666666666666	9826666666666669*

36.84325735623693	-36.84325735623693
92166666666666669*-	9826666666666669*-

68,71513657172002	-68.71513657172002
6076666666666667*-	6076666666666667*-

6.259273814635493	-6.259273814635493
-449999999999999999	-*4999999999999

18.72257829355884

S(A) =

IIAII = 18.72308154935928

BLOCKNORMREDUZIERUNG: 915 s DIAGONAL: 50 T GRTHOGONAL: 731 TRANSFORM ATIO NEN DURCHGEF UEHRT. 11 A 1

423 TRANSFORMATIONEN DURCHGEFUEHRIS

9403 TRANSFORMATIONEN DURCHGEFUEHRT.

9568

BLOCKNORMREDUZIERUNG :

GESAMT:

ORTHOGONAL

DIAGONAL :

```
.0014451729059942
                                                                                                               .0071742282598550
                                                                                                                                                                                                                                                                                                                                                                                                                                    • 0000100196440658
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               ||C(D)|| = +.281946183944*-13
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        110(0)11 = +.315201929290*-11
                                           KOND(T) = 348.8782891546430
                                                                                                                                                                                                                                                                                                                                                               KOND(T) = 400.3373709353124
                                                                                                                                                                                                     KOND(T) = 387.5734405725318
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               396.9385917704082
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        395 6218142077677
                                                                                                                                                                                                                                                                                                                                                                                                             NT3 = 1110
                                                                                                                                                                                                                                                NI3 = 1225

• IIC(0)|| =
                                                                                         DIAGONAL : 50 TRANSFORMATIONEN DURCHGEFUEHRIS
                                                                                                                                                                                                                                                                                                                   BLOCKNORMREDUZIERUNG: 1108 . DIAGONAL: 45 TRANSFORMATIONEN DURCHGEFUEHRT. ORTHOGONAL: 1110 TRANSFORMATIONEN DURCHGEFUEHRT. 10362 ZENTISEKUNDEN GEBRAUCHT
 50 TRANSFORMATIONEN DURCHGEFUEHRIS
                                                                                                                                                                                                                                                                                                                                                                                                                                                                             BLOCKNORMREDUZIERUNG: 177 , DIAGONAL: 21 TRANSFORMATIONEN DURCHGEFUEHRT, ORTHOGONAL: 194 TRANSFORMATIONEN DURCHGEFUEHRT, 3162 ZENTISEKUNDEN GEBRAUCHT
                       1225 TRANSFORMATIONEN DURCHGEFUEHRT, 11176 ZENTISEKUNDEN GEBRAUCHT
                                                                                                                                                        BLOCKNORMREDUZIERUNG: 1225 , DIAGONAL: 50 TRANSFORMATIONEN DURCHGEFUEHRT ORTHOGONAL: 1225 TRANSFORMATIONEN DURCHGEFUEHRT, 11132 ZENTISEKUNDEN GEBRAUCHT
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      KOND(T) =
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             KOND(T) =
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   BLOCKNORMREDUZIERUNG: 18 , DIAGONAL: 7 TRANSFORMATIONEN DURCHGEFUEHRI, ORTHOGONAL: 18 TRANSFORMATIONEN DURCHGEFUEHRI, 1796 ZENTISEKUNDEN GEBRAUCHT
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    N13 = 194
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      11C(A)11 = +.352212068702*-07
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           3 \text{ NT4} = 18 \text{ NV3} = 18 \text{ NT3} = 18

| 1|C(A)| | = +844841014507*-13 
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  •
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        •
                                                                                                                                                                                                                                                                                                                                                              s(A) = .0007661847010530 s
S(A-) = .0000394050013975
VV4 = 1110 NT4 = 1110 NV3 = 1110
                                                                                     NSCHIE = 1225 NSY = 0 NV4 = 1225 NT4 = 1225 NV3 = 1225 S(C(A)) = .0179601764117472 J HC(A)11 = .0193400488072182
                                                                                                                                                                                                                                                 KV3 = 1225
                                                                                                                                                                                                                             .0245737449980804
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    NA3 = 194
                                           S(A) = .0582838257494650
                                                                                                                                                                                         s(A) = .0158016522357789
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     N14 = 194
  DIAGONAL :
                                                                 S(A-) =
                                                                                                                                                                                                                                                                                                                                                                                                             NV4 = 1110
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  NV4 = 194
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    S(C(A)) = +.352212067291*-07
                                              •0785831120391547
                                                                                                                                                                                                                                                                                                                                                                                     *0010828320452037
                                                                                                                                                                                                                                                                                                                                                                                                                                S(C(A)) = .0002732919097695
  1225
                                                                                                                                                                                                                                                                      S(C(A)) = .0025924555866016
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             S(C(A)) = +.7964061081831-13
                                                                                                                                                                                                                          S(A+) = *0220609407183735
                                          ||A|| = 14.0021688872960
|S(A+) = .0010828320452037
                                                                                                                                                                                                    IIAII = 14.00218905692034
                                                                                                                                                                                                                                                                                                                                                                                                              NSY = 0
                                                                                                                                                                                                                                                 NSCHIE = 1225 NSY = 0
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  NSCHIE = 1225 NSY = 0
SL OCKNORMREDUZIERUNG :
                                                                                                                                                                                                                                                                                                                                                                                                             NSCHIE = 1225
                                                                   S(A+) =
                                                                                                                                                                                                                                                                                                                       ထ
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 ş
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      10
      Ó
                                                                                                                                                           /
```

Literaturverzeichnis

- [1] Caldewey, G.: Ein Jacobi-ähnliches Verfahren zur Eigenwertbestimmung normaler reeller Matrizen;
 Diplomarbeit, Universität Dortmund, 1980.
- [2] Dörrie, H.: Kubische und biquadratische Gleichungen; München 1948.
- [3] Dollinger, E.: Ein linear konvergentes zyklisches jacobiähnliches Verfahren für beliebige reelle Matrizen; Numer. Math. 38, 245-253 (1981).
- [4] Eberlein, P.J.: A Jacobi-like method for the automatic computation of eigenvalues and eigenvektors of an arbitrary matrix; SIAM J. 10, 74-88 (1962).
- [5] Eberlein, P.J.: Solution to the complex eigenproblem by a norm reducing Jacobi-type method; Numer. Math. 14, 232-245 (1970).
- [6] Eberlein, P.J., Boothroyd, J.: Solution to the eigenproblem by a norm-reducing Jacobi-type method;
 Numer.Math. 11, 1-12 (1968).
- [7] Hari, V.: A Jacobi-like eigenvalue algorithm for general real matrices; Glasnik Matematički 11, (31), 367-378 (1976).
- [8] Hari,V.: On convergence of certain cyclic Jacobi-like norm-reducing processes for real matrices; unveröffentlichtes Manuskript.
- [9] Hari, V.: On the global convergence of the Eberlein method for real matrices; Numer. Math. 39, 361-369 (1982).
- [10] Hari, V.: On the quadratic convergence of the Paardekooper method I; Glasnik Matematički 17 (37), 183-195 (1982).
- [11] Hari, V.: On the quadratic convergence of the Paardekooper method II; erscheint in: Glasnik Matematički.
- [12] Hari,V.: On the quadratic convergence of the Paardekooper method III; erscheint in Glasnik Matematički.
- [13] Henrici,P.: Bounds for iterates, inverses, spectral variation and fields of values of non normal matrices; Numer.Math. $\underline{4}$, 24-41 (1962).
- [14] Henrici,P.: On the speed of convergence of cyclic and quasicyclic Jacobi methods for computing eigenvalues of Hermitian matrices; SIAM J. 6, 144-162 (1958).

- [15] Hoffmann, A.J., Wielandt, H.W.: The variation of the spectrum of a normal matrix; Duke Math. J. 20, 37-39 (1953).
- [16] Jacobi, C.G.J.: Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen; J.Reine Angew. Math. 30, 51-94 (1846).
- [17] Johnsen, Th.L.: A numerical method for eigenreduction of non-symmetric real matrices; Computer & Structures 8, 399-402 (1978).
- [18] Kress, R., de Vries, H.L., Weymann, R.: On nonnormal matrices; Lin. Alg. Appl. 8, 109-120 (1974).
- [19] Ljusternik, L.A., Sobolev, W.I.: Elemente der Funktional-analysis; Frankfurt/M. 1979.
- [20] Meyer,D., Veselić,K.: Abschätzung des Konditionsmaßes einer Matrix S, die eine Matrix blockdiagonalisiert; unveröffentlichtes Manuskript.
- [21] Meyer, D., Veselić, K.: On some new inclusion theorems for the eigenvalues of partitioned matrices; Numer. Math. 34, 431-437 (1980).
- [22] Mirsky,L.: On the minimization of matrix norms; Am.Math.Monthly 65, 106-107 (1958).
- [23] Ortega, J.M.: The Newton-Kantorovich Theorem; Amer.Math. Monthly 75, 658-660 (1968).
- [24] Osborne, E.E.: On pre-conditioning of matrices; J.Assoc. Comput. Mach. 7, 338-345 (1960).
- [25] Paardekooper, M.H.C.: An eigenvalue algorithm based on norm-reducing transformation; Thesis, Techn. Hoogesch. Einhoven 1969.
- [26] Paardekooper, M.H.C.: An eigenvalue algorithm for skew-symmetric matrices; Numer. Math. 17, 189-202 (1971).
- [27] Parlett, B.N., Reinsch, C.: Balancing a matrix for calculation of eigenvalues and eigenvectors; Numer. Math. 13, 293-304 (1969).
- [28] Ruhe, A.: On the quadratic convergence of a generalization of the Jacobi-method to arbitrary matrices; BIT $\underline{8}$, 210-231 (1968).
- [29] Rutishauser, H.: The Jacobi method für real symmetric matrices; Numer. Math. $\underline{9}$, 1-10 (1966).
- [30] Schönhage, A.: Zur Konvergenz des Jacobi-Verfahrens; Numer. Math. 3, 374-380 (1961).

- [31] Schönhage, A.: Zur quadratischen Konvergenz des Jacobi-Verfahrens; Numer. Math. 6, 410-412(1964).
- [32] Schur, I.: Über die charakteristischen Wurzeln einer linearen Substitution mit einer Anwendung auf die Theorie der Integralgleichungen; Math. Ann. 66, 488-501 (1909).
- [33] Veselić, K.: A convergent Jacobi-method for solving the eigenproblem of arbitrary real matrices; Numer.Math. 25, 179-184 (1976).
- [34] Veselić, K.: On a class of Jacobi-like procedures for diagonalising arbitrary real matrices; Numer. Math. 33, 157-172 (1979).
- [35] Veselić, K.: On a new class of elementary matrices; Numer. Math. 33, 173-180 (1979).
- [36] Veselić, K.: On optimal linearisation of the quadratic eigenvalue problem; Lin.Mult.Alg., 8, 253-258 (1980).
- [37] Veselić, K.: Reduktion einer beliebigen reellen Matrix der Ordnung 4 auf Quasidreiecksform; unveröffentlichtes Manuskript.
- [38] Veselić, K., Wenzel, H.J.: A quadratically convergent Jacobi-like method for real matrices with complex eigenvalues; Numer. Math. 33, 425-435 (1979).
- [39] Voevodin, V.V.: Čislennye methody lineinoj algebra; Nauka, Moskva 1966.
- [40] Wilkinson, J.H.: Note on the quadratic convergence of the cyclic Jacobi process; Numer. Math. 4, 296-300 (1962).
- [41] Wilkinson, J.H.: The algebraic eigenvalue problem; Oxford 1965.

Lebenslauf

05.06.1948	Geburt in Dortmund Vater: Wilhelm Wenzel, verstorben 19.01.1956 Mutter: Edith Wenzel, geb. Redecker Beruf des Vaters: Bergmann Staatsangehörigkeit: deutsch Konfession: evangelisch
1954 - 1959	Volksschule in Dortmund-Barop
1959 - 1965	Realschule in Dortmund
1965 - 1968	Physiklaborantenlehre am Max-Planck- Institut für Arbeitsphysiologie in Dortmund
Juli 1970	Abitur
WS 1970/71 - SS 1977	Studium der Mathematik, Informatik, Physik an der Universität Dortmund
Juni 1977	Hauptdiplom in Mathematik (Universität Dortmund)
01.09.1977	Verwalter der Stelle eines Wissenschaft- lichen Assistenten an der Universität Dortmund
07.11.1978	Ernennung zum Wissenschaftlichen Assistenten daselbst
19.12.1979	Versetzung an die Fernuniversität - Gesamthochschule- Hagen
01.04.1983	Wissenschaftlicher Mitarbeiter am Rechenzentrum der Fernuniversität -Gesamthochschule- Hagen
13.05.1983	Einreichung der Dissertation
05.07.1983	mündliche Doktorprüfung

Hagen, den 05.07.1983 Hans Joachim Wenzel