Kapitel 5

Toeplitz-Matrizen

5.1 Der Algorithmus von Trench

Im letzten Abschnitt haben wir effiziente Algorithmen zur zyklischen Faltung definiert. Wir hatten gezeigt, daß die Faltung zweier Vektoren der Anwendung einer Matrix A entspricht mit $A = P^t AP$ und $P = (e_2, e_3, \ldots, e_n, e_1)$. Hierbei bewirkt die Rechtsmultiplikation mit P eine zyklische Verschiebung der Spalten nach links, eine Linksmultiplikation mit P^t eine zyklische Verschiebung der Zeilen nach oben.

Wir definieren nun die Toeplitz-Matrizen als Erweiterung des Matrizenraums der zyklischen Faltung.

Definition 5.1.1 *Eine (n, n)–Matrix der Form*

$$A = \begin{pmatrix} a_0 & a_1 & \cdots & a_{n-2} & a_{n-1} \\ a_{-1} & \ddots & \ddots & & a_{n-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{-(n-2)} & & \ddots & \ddots & a_1 \\ a_{-(n-1)} & a_{-(n-2)} & \cdots & a_{-1} & a_0 \end{pmatrix}$$

heißt Toeplitz–Matrix. A ist symmetrische Faltung genau dann, wenn a periodisch ist mit Periode n.

Bemerkung: Für alle Toeplitz–Matrizen A gilt $\mathrm{rang}(A - P^tAP) \leq 2$, denn

$$A - P^tAP = \begin{pmatrix} 0 & \cdots & 0 & a_{n-1} - a_{-1} \\ \vdots & & \vdots & \vdots \\ 0 & & 0 & a_1 - a_{n+1} \\ a_{-(n-1)} - a_1 & \cdots & a_{-1} - a_{n-1} & 0 \end{pmatrix}.$$

Der Rang ist also zwar nicht mehr unbedingt 0, aber klein.

Satz 5.1.2 *Für eine (n,n) Toeplitz–Matrix A und einen Vektor x kann Ax durch eine zyklische Faltung der Länge $M \geq 2n-1$ berechnet werden.*

Beweis: Wir erweitern A sinnvoll so, daß sich eine zyklische Faltung ergibt. Sei A definiert durch den Vektor a_k, $k = -n-1 \ldots n-1$. Definiere den Vektor α_k aus $\mathbb{C}^M$, $k = -M-1 \ldots M-1$, durch

$$\alpha_k := \begin{cases} a_k & -n-1 \leq k \leq n-1 \\ a_{k-M} & k > M-n \\ a_{k+M} & k < n-M \\ 0 & \text{sonst.} \end{cases} .$$

Der so definierte Vektor ist periodisch mit der Periode M. α definiert eine (M,M)–Toeplitz–Matrix A', die eine zyklische Faltung ist, und deren linke obere (n,n)–Untermatrix mit A übereinstimmt. Wir wenden A' auf den Vektor

$$x' = \begin{pmatrix} x \\ 0 \end{pmatrix} \in \mathbb{C}^M$$

an. Da $A_{i,k} = A'_{i,k}$ für $i = 1 \ldots n$, $k = 1 \ldots n$ gilt

$$A'x' = \begin{pmatrix} Ax \\ y \end{pmatrix}$$

und wir haben Ax berechnet. Dieses Vorgehen heißt aus naheliegenden Gründen "zero-padding".

Beispiel: Wir betrachten ein Beispiel mit $n = 3$ und $M = 6$. Dann gilt

$$A = \begin{pmatrix} a_0 & a_1 & a_2 \\ a_{-1} & a_0 & a_1 \\ a_{-2} & a_{-1} & a_0 \end{pmatrix}$$

und

$$A' = \begin{pmatrix} a_0 & a_1 & a_2 & 0 & a_{-2} & a_{-1} \\ a_{-1} & a_0 & a_1 & a_2 & 0 & a_{-2} \\ a_{-2} & a_{-1} & a_0 & a_1 & a_2 & 0 \\ 0 & a_{-2} & a_{-1} & a_0 & a_1 & a_2 \\ a_2 & 0 & a_{-2} & a_{-1} & a_0 & a_1 \\ a_1 & a_2 & 0 & a_{-2} & a_{-1} & a_0 \end{pmatrix}.$$

Bemerkung: Wir haben insbesondere gezeigt, daß sich jede zyklische Faltung der Länge n auch als zyklische Faltung der Länge M mit $M > n$ berechnen läßt. Daher beschränken sich viele Bibliotheken auf die Betrachtung von Faltungen z.B. der Länge 2^p.

Bemerkung: Fouriertransformationen von der Ordnung p^n mit $p \neq 2$ prim lassen sich im wesentlichen als Faltungen schreiben. Der Satz liefert uns damit auch Verfahren zur schnellen Durchführung von Fouriertransformationen, wenn nur wenige Faltungsalgorithmen implementiert sind.

Bemerkung: Toeplitz–Matrizen tauchen zum Beispiel in der Computer–Tomographie auf. Es muß das Gleichungssystem $Ax = b$ mit einer Toeplitz–Matrix A gelöst werden. Wir untersuchen daher jetzt den Aufwand zur Lösung dieses Gleichungssystems.

Satz 5.1.3 *Sei A eine (n, n) Toeplitz–Matrix. Alle Hauptminoren von A seien ungleich null (z.B. A positiv definit). Dann kann das Gleichungssystem $Ax = b$ mit $\frac{13}{4}n^2 + O(n)$ Operationen gelöst werden.*

Beweis: Wir definieren eine Hierarchie von Matrizen A_k für $k = -(n-1)\ldots(n-1)$, $A_0 := A$. Für $k > 0$ sind k Nebendiagonalen oberhalb der

Hauptdiagonalen 0, für $k < 0$ sind k Nebendiagonalen unterhalb der Hauptdiagonalen 0. Es gilt also für $k > 0$

$$A_k = \begin{pmatrix} * & 0 & \cdots & 0 & * & \cdots & * \\ \vdots & \ddots & \ddots & & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & & \ddots & * \\ \vdots & & & \ddots & \ddots & & 0 \\ \vdots & & & & \ddots & \ddots & \vdots \\ \vdots & & & & & \ddots & 0 \\ * & \cdots & \cdots & \cdots & \cdots & \cdots & * \end{pmatrix}.$$

A_{-k} hat die transponierte Gestalt, aber nicht notwendig mit denselben Einträgen für die Elemente mit $*$.

Wir beschreiben den Algorithmus für $n = 4$, indem wir elementare Zeilenumformungen von A angeben, die A_k erzeugen. Wir starten mit der Matrix $A = A_0$ und konstruieren A_{-1} und A_1 gleichzeitig.

$$A_0 = \begin{pmatrix} a_0^0 & a_1^0 & a_2^0 & a_3^0 \\ a_{-1}^0 & a_0^0 & a_1^0 & a_2^0 \\ a_{-2}^0 & a_{-1}^0 & a_0^0 & a_1^0 \\ a_{-3}^0 & a_{-2}^0 & a_{-1}^0 & a_0^0 \end{pmatrix}, \; A_0 = \begin{pmatrix} a_0^0 & a_1^0 & a_2^0 & a_3^0 \\ a_{-1}^0 & a_0^0 & a_1^0 & a_2^0 \\ a_{-2}^0 & a_{-1}^0 & a_0^0 & a_1^0 \\ a_{-3}^0 & a_{-2}^0 & a_{-1}^0 & a_0^0 \end{pmatrix}.$$

Aus der linken Matrix konstruieren wir A_{-1}. Die oberste Zeile hat bereits die richtige Form. Für $k > 1$ ziehen wir von Zeile k das $\frac{a_{-1}^0}{a_0^0}$–fache der $k-1$. Zeile der rechten Matrix ab. In der rechten Matrix ziehen wir für $k < n$ von Zeile k das $\frac{a_1^0}{a_0^0}$–fache der $k+1$. Zeile der linken Matrix ab. Dadurch erhalten wir folgende Matrizen:

$$A_{-1} = \begin{pmatrix} a_0^0 & a_1^0 & a_2^0 & a_3^0 \\ 0 & a_0^{-1} & a_1^{-1} & a_2^{-1} \\ a_{-2}^{-1} & 0 & a_0^{-1} & a_1^{-1} \\ a_{-3}^{-1} & a_{-2}^{-1} & 0 & a_0^{-1} \end{pmatrix}, \; A_1 = \begin{pmatrix} a_0^1 & 0 & a_2^1 & a_3^1 \\ a_{-1}^1 & a_0^1 & 0 & a_2^1 \\ a_{-2}^1 & a_{-1}^1 & a_0^1 & 0 \\ a_{-3}^0 & a_{-2}^0 & a_{-1}^0 & a_0^0 \end{pmatrix}.$$

Die Operation ist durchführbar, denn a_0^0 ist der erste Hauptminor der Matrix A und damit nicht Null. Zur Durchführung dieses Schritts müssen wir einmal dividieren und links und rechts $2n-2$ Elemente neu berechnen. Da dieselbe Rechnung auch auf der rechten Seite (b) durchgeführt werden muß, benötigen wir insgesamt $2(3n-3)$ Rechenoperationen und eine Division.

Im nächsten Schritt bringen wir die nächste Diagonale auf Null. Man sieht schnell, daß dies nicht durch elementare Zeilenumformungen innerhalb der Matrizen möglich ist. Da wir aber wissen, daß gleichzeitig $A_{-1}x = b_{-1}$ und $A_1 x = b_1$, können wir auch Zeilen der rechten Matrix auf die linke addieren und umgekehrt. Für $k \leq 2$ hat A_{-1} bereits die richtige Form. Für $k > 2$ addieren wir zu Zeile k der linken Matrix das $\frac{a_{-2}^{-1}}{a_0^1}$–fache der $(k-2)$. Zeile der rechten Matrix. Für $k < n+1-2$ addieren wir zur Zeile k der rechten Matrix das $\frac{a_2^1}{a_0^{-1}}$–fache der $(k+2)$. Zeile der linken Matrix. Wir erhalten

$$A_{-2} = \begin{pmatrix} a_0^0 & a_1^0 & a_2^0 & a_3^0 \\ 0 & a_0^{-1} & a_1^{-1} & a_2^{-1} \\ 0 & 0 & a_0^{-2} & a_1^{-2} \\ a_{-3}^{-2} & 0 & 0 & a_0^{-2} \end{pmatrix},\ A_2 = \begin{pmatrix} a_0^2 & 0 & 0 & a_3^2 \\ a_{-1}^2 & a_0^2 & 0 & 0 \\ a_{-2}^1 & a_{-1}^1 & a_0^1 & 0 \\ a_{-3}^0 & a_{-2}^0 & a_{-1}^0 & a_0^0 \end{pmatrix}.$$

Die Divisionen sind wieder ausführbar. Die obere linke $(2,2)$-Teilmatrix von A_1 ist durch elementare Zeilenumformungen aus der oberen linken $(2,2)$-Teilmatrix von A hervorgegangen (genauer Beweis in den Aufgaben). Diese ist invertierbar, denn der zweite Hauptminor ist nicht Null, also ist a_0^2 nicht Null. Der Rechenaufwand für diesen Schritt ist eine Division und $2(3n-6)$ Operationen.

Wir beschreiben nun Schritt j. Seien also A_{j-1} und $A_{-(j-1)}$ bereits berechnet. Addiere für $j+1 \leq k \leq n$ auf die k. Zeile der linken Matrix das $\frac{a_{-j}^{-j+1}}{a_0^{j-1}}$–fache der $(k-j)$. Zeile der rechten Matrix. Addiere für $1 \leq k \leq n-j$ das $\frac{a_j^{j-1}}{a_0^{-(j-1)}}$–fache der $(k+j)$. Zeile der linken Matrix auf die k. Zeile der rechten Matrix.

Der Aufwand zur Umformung einer Matrix und seiner rechten Seite beträgt damit $3(n-j)$ für Schritt j. Insgesamt erhalten wir einen Aufwand von

$$\sum_{j=1}^{n-1} 2(3n-j) = 3n^2 + O(n).$$

Wir müssen das Gleichungssystem nun noch lösen durch Rückwärts- und Vorwärtseinsetzen. Für die ersten $\frac{n}{2}$ Variablen benutzen wir $A_{-(n-1)}$ und Vorwärtseinsetzen, für den Rest A_{n-1} und Rückwärtseinsetzen, macht noch einmal $2\frac{1}{2}\left(\frac{n}{2}\right)^2$ Operationen, insgesamt also $13\frac{n^2}{4}$ Rechenoperationen.

5.2 Der Algorithmus von Morf

Wir wollen den Begriff der Toeplitz–Matrix noch einmal verallgemeinern. Wir hatten oben bereits gesehen, daß für den Rang d der Matrix $A - P^tAP$ einer Toeplitz–Matrix A immer $d \leq 2$ gilt. Wir definieren den $+$– und $-$–Verschiebungsrang.

Definition 5.2.1 *Sei M die (n,n)–Matrix $(e_2, e_3, \ldots, e_n, 0)$. Für eine (n,n)–Matrix A nennen wir*

$$\begin{aligned} \alpha_+(A) &:= \mathrm{rk(A - MAM^t)} \\ \alpha_-(A) &:= \mathrm{rk(A - M^tAM)} \end{aligned}$$

den $+$– bzw. $-$–Verschiebungsrang von A.

Sei $A = (a_{ik})$, $i = 1, \ldots, n$, $k = 1, \ldots, n$. Dann ist

$$MAM^t = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & a_{1,1} & \cdots & a_{1,n-1} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n-1,1} & \cdots & a_{n-1,n-1} \end{pmatrix}$$

und

$$M^t AM = \begin{pmatrix} a_{2,2} & \cdots & a_{2,n} & 0 \\ \vdots & & \vdots & \vdots \\ a_{n,2} & \cdots & a_{n,n} & 0 \\ 0 & \cdots & 0 & 0 \end{pmatrix}$$

Beispiel: Toeplitz–Matrizen haben den Veriebungsrang $\alpha_+(A) = \alpha_-(A) \leq 2$.

Satz 5.2.2 *Für invertierbare Matrizen gilt:*

$$\alpha_+(A) = \alpha_-(A^{-1}).$$

Beweis: Wir zeigen zunächst den Hilfssatz:

Für beliebige (n, n)–Matrizen A und B gilt:

$$\mathrm{rk(I - AB) = rk(I - BA)}$$

Beweis des Hilfssatzes: Sei $(x_1, \ldots, x_m)$ eine Basis des Kerns von $(I - AB)$, und $y_i := Bx_i$. Die y_i sind linear unabhängig, denn sonst wären die $Ay_i = ABx_i = x_i$ linear abhängig. Außerdem gilt

$$(I - BA)y_i = (I - BA)Bx_i = B(I - AB)x_i = 0$$

und die y_i sind im Kern von $(I - BA)$ enthalten. Die Dimension des Kerns von $(I - BA)$ ist also mindestens so groß wie die Dimension des Kerns von $(I - AB)$ und es gilt

$$\mathrm{rk(I - BA) \leq rk(I - AB)}.$$

Durch Vertauschen von A und B erhält man den Hilfssatz.

Beweis des Satzes:

$$\begin{aligned}
\alpha_+(A) &= \mathrm{rk}(\mathrm{I}-\mathrm{MAM}^{\mathrm{t}}) \\
&= \mathrm{rk}((\mathrm{I}-\mathrm{MAM}^{\mathrm{t}}\mathrm{A}^{-1})\mathrm{A}) \\
&= \mathrm{rk}(\mathrm{I}-\mathrm{MAM}^{\mathrm{t}}\mathrm{A}^{-1}) \\
&= \mathrm{rk}(\mathrm{I}-\mathrm{M}^{\mathrm{t}}\mathrm{A}^{-1}\mathrm{MA}) \qquad \text{nach dem Hilfssatz} \\
&= \mathrm{rk}(\mathrm{A}^{-1}-\mathrm{M}^{\mathrm{t}}\mathrm{A}^{-1}\mathrm{MAA}^{-1}) \\
&= \mathrm{rk}(\mathrm{A}^{-1}-\mathrm{M}^{\mathrm{t}}\mathrm{A}^{-1}\mathrm{M}) \\
&= \alpha_-(A^{-1}).
\end{aligned}$$

Für unsere weiteren Betrachtungen ist fundamental, daß sich Matrizen mit kleinem Verschiebungsrang als Summe einfacher Matrizen darstellen lassen. Wir definieren zunächst die unteren und oberen Dreiecks–Toeplitz– Matrizen durch

Definition 5.2.3 *Sei $x \in \mathbb{C}^n$, $x = (x_i)$, $i = 0, \ldots, n-1$. Dann definieren wir die zugehörige untere (n, n) Dreiecks–Toeplitz–Matrix durch*

$$L(x)_{i,i-l} := \begin{cases} x_l, & l \geq 0 \\ 0, & \text{sonst} \end{cases}, \text{ also } L(x) = \begin{pmatrix} x_0 & & 0 \\ \vdots & \ddots & \\ x_{n-1} & \cdots & x_0 \end{pmatrix}$$

und die zugehörige obere (n, n) Dreiecks–Toeplitz–Matrix durch

$$U(x) := L(x)^t.$$

Die Bezeichnungen U und L stehen natürlich für Upper und Lower.

Satz 5.2.4 *Seien x^j, $y^j \in \mathbb{C}^n$, $j = 1, \ldots, \alpha$, $\alpha > 0$. Dann ist äquivalent:*

1. $A - MAM^t = \sum_{j=1}^{\alpha} x^j (y^j)^t$
2. $A = \sum_{j=1}^{\alpha} L(x^j) U(y^j)$

Beweis: Wir zeigen zunächst $2 \Rightarrow 1$. Sei $A = (a_{i,j}), i = 0, \ldots, n-1,\ j = 0, \ldots, n-1$. Dann gilt

$$\begin{aligned} a_{k,l} &= \sum_{j=1}^{\alpha} \sum_{i} L(x^j)_{k,i} U(y^j)_{i,l} \\ &= \sum_{j=1}^{\alpha} \sum_{i=0}^{n-1} x^j_{k-i} y^j_{l-i} \end{aligned}$$

mit der Definition $x_i = y_i = 0$ für $i < 0$. Wir definieren weiter $a_{k,l} = 0$ für $k < 1$ oder $l < 1$. Dann gilt

$$\begin{aligned} (A - MAM^t)_{k,l} &= a_{k,l} - a_{k-1,l-1} \\ &= \sum_{j=1}^{\alpha} \sum_{i=0}^{n-1} x^j_{k-i} y^j_{l-i} - x^j_{k-1-i} y^j_{l-1-i} \\ &= \sum_{j=1}^{\alpha} x^j_k y^j_l \\ &= \sum_{j=1}^{\alpha} (x^j (y^j)^t)_{k,l} \end{aligned}$$

und $A - MAM^t = \sum_{j=1}^{\alpha} x^j (y^j)^t$.

Die umgekehrte Richtung folgt sofort (Übungen).

Satz 5.2.5 *Zu jeder (n, n) Matrix A gibt es Vektoren x^j, y^j, $j = 1, \ldots, \alpha_+(A)$, und Vektoren $\tilde{x}^j$, $\tilde{y}^j$, $j = 1, \ldots, \alpha_-(A)$, so daß*

$$A = \sum_{j=1}^{\alpha_+(A)} L(x^j) U(y^j) \ \textit{und}\ A = \sum_{j=1}^{\alpha_-(A)} U(\tilde{y}^j) L(\tilde{x}^j).$$

Beweis: Für die Matrix $B := A - MAM^t$ vom Rang $\alpha_+(A)$ gibt es Vektoren $x_0, \ldots, x_{k-1}$ und $y_0, \ldots, y_{k-1}$ mit

$$B = \sum_{j=1}^{\alpha} x^j (y^j)^t.$$

Nach Satz (??) folgt die Behauptung für $A - MAM^t$.

Satz 5.2.6 *Seien A, B (n,n)–Matrizen mit festem von n unabhängigem Verschiebungsrang $\alpha = \alpha_+(A)$. Dann kann man die LU-Zerlegung von AB mit $n \log n$ Operationen berechnen.*

Beweis:

1. Die LU-Zerlegung von A kann nach dem Beweis von Satz ?? in $O(n)$ berechnet werden. (Anmerkung: Mit hoher Wahrscheinlichkeit? nachfragen!) Das Bestimmen der l.u. Vektoren kann $O(n^2)$ Operationen kosten.

2. Das Produkt von unteren oder oberen Toeplitz–Dreiecksmatrizen ist $O(n \log n)$. **Beweis der Zwischenbehauptung:**

$$\begin{aligned}(L(x)L(y))_{kl} &= \sum_{i=l}^{k} x_{k-i}y_{i-l}, \ k \geq l \\ &= \sum_{j=0}^{k-l} x_j y_{k-l-j} =: z_i\end{aligned}$$

   Die z_i können durch schnelle Faltungen berechnet werden in Zeit $O(n \log n)$.

3. Sei U eine obere, L eine untere Dreiecks–Toeplitz–Matrix. Dann kann man in $O(n)$ untere bzw. obere Dreiecks–Toeplitz–Matrizen L_i, U_i berechnen mit

$$UL = \sum_{i=1}^{3} L_i U_i$$

   Beweis: Erweitere U und L wie im Beweis zu Satz ?? zu zyklischen Faltungen der Länge $2n$. Die Matrizen $\tilde{U}$ und $\tilde{L}$ haben dann die Form

$$\tilde{U} = \begin{pmatrix} \tilde{U} & \tilde{L}_1 L \\ \tilde{L}_2 & \tilde{U}_3 \end{pmatrix}, \tilde{L} = \begin{pmatrix} L & \tilde{U}_1 \\ \tilde{U}_2 & \tilde{L}_3 \end{pmatrix}, \tilde{U}\tilde{L} = \begin{pmatrix} UL + \tilde{L}_1\tilde{U}_1 & * \\ * & * \end{pmatrix}.$$

   Das Produkt dieser zyklischen Faltung ist wieder eine zyklische Faltung. Also ist es mit $O(n \log n)$ Operationen berechenbar. Außerdem ist die Matrix $LU - U_1 L_1$ Toeplitz, also $LU - U_1 L_1 = \sum_{U_i L_i}$, und die Zerlegung ist in $O(n)$ berechenbar.

Wir setzen diese Ergebnisse nun zusammen. Gesucht sei das Produkt von A und B. Nach i) gilt

$$A = \sum_{i=1}^{\alpha} L_i U_i, \; B = \sum_{i=1}^{\alpha} \tilde{L}_i \tilde{U}_i.$$

Damit gilt

$$AB = \sum_{i,j=1}^{\alpha} L_i U_i \tilde{L}_j U_j.$$

Nach Punkt iii) finden wir Matrizen mit

$$U_i \tilde{L}_j = \sum_{k=1}^{3} L_k^{i,j} U_k^{i,j}$$

und

$$AB = \sum_{i,j,k=1}^{3} L_i L_k^{i,j} U_k^{i,j} \tilde{U}_j = \sum_{j=1}^{3\alpha^2} \hat{L}_l \hat{U}_l.$$

Nach einem Satz ist die Multiplikation von Matrizen ungefähr so aufwendig wie die Inversion. Wir erwarten also, daß es einen schnellen Algorithmus auch zur Inversion von Matrizen mit kleinem Verschiebungsrang gibt.

Sei nun

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

mit $(\frac{n}{2}, \frac{n}{2})$-Matrizen $A_{i,k}$. In Lemma 2.3 zum Strassen– Algorithmus wurde diese Matrix bereits invertiert. Zur Blockinversion sind einige Produkte von Matrizen $A_{i,k}$ und die Inversen der Matrizen $A_{1,1}$ und $A_{2,2} - A_{2,1} A_{1,1}^{-1} A_{1,2}$ zu berechnen. Wenn wir zeigen können, daß alle diese Matrizen kleinen Verschiebungsrang haben, können wir einen schnellen Algorithmus definieren.

Satz 5.2.7 *Sei A (n,n)–Matrix mit $n = 2m$ wie oben aufgeteilt. Dann gilt:*

- $\alpha_+(A_{1,1}) \leq \alpha_+(A)$.
- $\alpha_-(A_{2,2}) \leq \alpha_-(A)$.
- $\alpha_+(A_{2,2}) \leq \alpha_+(A) + 2$.
- $\alpha_+(A_{1,2}) \leq \alpha_+(A) + 1$.
- $\alpha_+(A_{2,1}) \leq \alpha_+(A) + 1$.
- $\alpha_+(A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}) \leq \alpha_+(A)$.

Beweis: Sei $A = \sum L_i U_i$. Spalte L_i und U_i auf wie A in Matrizen

$$L_j = \begin{pmatrix} L_j^{1,1} & 0 \\ L_j^{2,1} & L_j^{2,2} \end{pmatrix}, \; U_j = \begin{pmatrix} U_j^{1,1} & U_j^{1,2} \\ 0 & U_j^{2,2} \end{pmatrix}, \; L_j U_j = \begin{pmatrix} L_j^{1,1}U_j^{1,1} & L_j^{1,1}U_j^{1,2} \\ L_j^{2,1}U_j^{1,1} & L_j^{2,1}U_j^{1,2} + L_j^{2,2}U_j^{2,2} \end{pmatrix}$$

mit Dreiecks–Toeplitz–Matrizen $L_j^{1,1}$, $U_j^{1,1}$, $L_j^{2,2}$ und $U_j^{2,2}$ sowie mit Toeplitz–Matrizen $L_j^{2,1}$ und $U_j^{1,2}$. Es gilt $A = \sum_{j=1}^{\alpha_+(A)} L_j U_j$. Aus dieser Darstellung folgt Punkt 1. Aus der entsprechenden UL- Darstellung folgt Punkt 2 und hieraus Punkt 3. Die Toeplitz–Matrizen kann man als Summe von Dreiecks–Toeplitz–Matrizen schreiben. Hieraus folgen 4 und 5. Die Matrix in Punkt 6 ist untere Teilmatrix von A^{-1}, also gilt

$$\alpha_+(A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}) = \alpha_-(C_{2,2}) \leq \alpha_-(A^{-1}) \leq \alpha_+(A).$$

Satz 5.2.8 *Die Inversion einer Matrix mit festem, von n unabhängigem Verschiebungsrang α kann mit $O(n(\log n)^2)$ Operationen durchgeführt werden.*

Beweis: Seien $T(n)$ die benötigten Operationen zur Inversion einer (n,n)–Matrix. Dann gilt $T(n) = 2T(n/2) + O(n \log n)$. Nach Satz 1.1 der Vorlesung gilt damit auch $T(n) = O(n(\log n)^2)$.