

## Aufgabe 1

Es gilt

$$P \cdot A = \begin{pmatrix} -2 & 6 & 3 & 10 \\ 0 & -4 & 10 & \frac{15}{2} \\ 2 & -6 & 7 & -\frac{11}{2} \\ -2 & 10 & -12 & 0 \end{pmatrix}$$

und

$$Pb = \begin{pmatrix} 59 \\ 52 \\ -11 \\ -18 \end{pmatrix}$$

Nun können wir ab jetzt die Zerlegung ohne weitere Pivotisierung durchführen. Im ersten Schritt erhalten wir

$$L_1 \cdot PA = \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ 1 & 0 & 1 & \\ -1 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -2 & 6 & 3 & 10 \\ 0 & -4 & 10 & \frac{15}{2} \\ 2 & -6 & 7 & -\frac{11}{2} \\ -2 & 10 & -12 & 0 \end{pmatrix} = \begin{pmatrix} -2 & 6 & 3 & 10 \\ 0 & -4 & 10 & \frac{15}{2} \\ 0 & 0 & 10 & \frac{9}{2} \\ 0 & 4 & -15 & -10 \end{pmatrix}$$

Schritt 2 liefert dann

$$L_2 \cdot L_1 PA = \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ 0 & 0 & 1 & \\ 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -2 & 6 & 3 & 10 \\ 0 & -4 & 10 & \frac{15}{2} \\ 0 & 0 & 10 & \frac{9}{2} \\ 0 & 4 & -15 & -10 \end{pmatrix} = \begin{pmatrix} -2 & 6 & 3 & 10 \\ 0 & -4 & 10 & \frac{15}{2} \\ 0 & 0 & 10 & \frac{9}{2} \\ 0 & 0 & -5 & -\frac{5}{2} \end{pmatrix}$$

Schritt 3 ergibt

$$L_3 \cdot L_2 L_1 PA = \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ 0 & 0 & 1 & \\ 0 & 0 & \frac{1}{2} & 1 \end{pmatrix} \cdot \begin{pmatrix} -2 & 6 & 3 & 10 \\ 0 & -4 & 10 & \frac{15}{2} \\ 0 & 0 & 10 & \frac{9}{2} \\ 0 & 0 & -5 & -\frac{5}{2} \end{pmatrix} = \begin{pmatrix} -2 & 6 & 3 & 10 \\ 0 & -4 & 10 & \frac{15}{2} \\ 0 & 0 & 10 & \frac{9}{2} \\ 0 & 0 & 0 & -\frac{1}{4} \end{pmatrix}$$

Wir erhalten

$$L = L_3^{-1} L_2^{-1} L_1^{-1} = \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ -1 & 0 & 1 & \\ 1 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ 0 & 0 & 1 & \\ 0 & -1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ 0 & 0 & 1 & \\ 0 & 0 & -\frac{1}{2} & 1 \end{pmatrix} = \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ -1 & 0 & 1 & \\ 1 & -1 & -\frac{1}{2} & 1 \end{pmatrix}$$

$$U = \begin{pmatrix} -2 & 6 & 3 & 10 \\ 0 & -4 & 10 & \frac{15}{2} \\ 0 & 0 & 10 & \frac{9}{2} \\ 0 & 0 & 0 & -\frac{1}{4} \end{pmatrix}$$

Es gilt nun  $\det PA = \det L \det U = 1 \cdot -20 = -20$ , also  $-1 \cdot \det A = -20 \implies \det A = 20$ . Für  $Ax = b \iff PAx = Pb$  lösen wir zunächst  $Ly = Px$ . Durch Einsetzen erhalten wir sofort

$$y = \begin{pmatrix} 59 \\ 52 \\ 48 \\ -1 \end{pmatrix}$$

Nun lösen wir noch  $Ux = y$ . Daraus ergibt sich mit rückwärts Einsetzen

$$x = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$$

Zur Berechnung von  $A^{-1}$  lösen wir  $(LU)x_i = Pe_i$ ,  $i = 1, \dots, 4$ . Dann sind  $x_i$  die Spalten von  $A^{-1}$ . Als erstes berechnen wir  $Ly_i = Pe_i$ , also  $Ly_1 = e_2$ ,  $Ly_2 = e_1$ ,  $Ly_3 = e_3$ ,  $Ly_4 = e_4$ . Durch Einsetzen erhalten wir

$$y_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, y_2 = \begin{pmatrix} 1 & 0 & 1 & -\frac{1}{2} \end{pmatrix}, y_3 = \begin{pmatrix} 0 & 0 & 1 & \frac{1}{2} \end{pmatrix}, y_4 = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}$$

und Lösen nun noch  $Ux_i = y_i$ . Daraus ergibt sich

$$x_1 = \begin{pmatrix} -\frac{541}{20} \\ -\frac{13}{4} \\ \frac{9}{5} \\ -4 \end{pmatrix}, x_2 = \begin{pmatrix} \frac{271}{20} \\ \frac{7}{4} \\ -\frac{4}{5} \\ 2 \end{pmatrix}, x_3 = \begin{pmatrix} -\frac{49}{4} \\ -\frac{5}{4} \\ 1 \\ -2 \end{pmatrix}, x_4 = \begin{pmatrix} -\frac{263}{10} \\ -3 \\ \frac{9}{5} \\ -4 \end{pmatrix}$$

und sofort

$$A^{-1} = \begin{pmatrix} -\frac{541}{20} & \frac{271}{20} & -\frac{49}{4} & -\frac{263}{10} \\ -\frac{13}{4} & \frac{7}{4} & -\frac{5}{4} & -3 \\ \frac{9}{5} & -\frac{4}{5} & 1 & \frac{9}{5} \\ -4 & 2 & -2 & -4 \end{pmatrix}$$

Für die Konditionszahl gilt nach Vorlesung

$$\text{cond}_\infty = \|A\|_\infty \cdot \|A^{-1}\|_\infty = 24 \cdot 1583 \cdot \frac{1}{20} = 1899.6$$

## Aufgabe 2

(a) Es gilt

$$(T \cdot v_k)_i = c \cdot \nu^{i-1} \sin\left((i-1)\frac{k\pi}{n+1}\right) + a \cdot \nu^i \sin\left(i\frac{k\pi}{n+1}\right) + b \cdot \nu^{i+1} \sin\left((i+1)\frac{k\pi}{n+1}\right)$$

für  $1 < i < n$ . Für  $i = 1$  wird aber der erste Term 0, da  $\sin(0) = 0$  und für  $i = n$  wird der letzte Term 0, da  $\sin(k\pi) = 0$ . Also gilt dieser Ausdruck für alle  $1 \leq i \leq n$ . Umformen ergibt nun

$$\begin{aligned} &= c \cdot \underbrace{\sqrt{\frac{b^2}{c^2}}}_{=1} \nu^2 \cdot \nu^{i-1} \sin\left((i-1)\frac{k\pi}{n+1}\right) + a \nu^i \sin\left(i\frac{k\pi}{n+1}\right) + b \nu^{i+1} \sin\left((i+1)\frac{k\pi}{n+1}\right) \\ &= a \nu^i \sin\left(i\frac{k\pi}{n+1}\right) + b \nu^{i+1} \cdot \left(\sin\left((i-1)\frac{k\pi}{n+1}\right) + \sin\left((i+1)\frac{k\pi}{n+1}\right)\right) \end{aligned}$$

Anwenden der Formel vom Übungsblatt

$$\begin{aligned}
 &= a\nu^i \sin\left(i \frac{k\pi}{n+1}\right) + b\nu^{i+1} \cdot 2 \cos\left(\frac{k\pi}{n+1}\right) \sin\left(i \frac{k\pi}{n+1}\right) \\
 &= \left(a + 2b\nu \cos\left(\frac{k\pi}{n+1}\right)\right) \cdot \sin\left(i \frac{k\pi}{n+1}\right) \\
 &= \left(a + 2b\nu \cos\left(\frac{k\pi}{n+1}\right)\right) \cdot (v_k)_i
 \end{aligned}$$

Also ist  $a + 2b\nu \cos\left(\frac{k\pi}{n+1}\right)$  ein Eigenwert zum Eigenvektor  $v_k$ .

- (b) Für diese Werte von  $a, b, c$  gilt  $\lambda_k = 2 - 2 \cos\left(\frac{k}{n+1}\pi\right)$ . Da der Cosinus im Intervall  $(0, \pi)$  streng monoton fallend ist, erhalten wir  $\lambda_{\max} = \lambda_n = 2 - 2 \cos\left(\frac{n}{n+1}\pi\right)$  und  $\lambda_{\min} = \lambda_1 = 2 - 2 \cos\left(\frac{1}{n+1}\pi\right)$ . Es gilt demnach  $\lim_{n \rightarrow \infty} \lambda_{\max} = 4$  und  $\lim_{n \rightarrow \infty} \lambda_{\min} = 0$ . Es folgt  $\text{cond}_2(T) = \frac{\lambda_{\max}}{\lambda_{\min}} = \frac{2 - 2 \cos\left(\frac{n}{n+1}\pi\right)}{2 - 2 \cos\left(\frac{1}{n+1}\pi\right)}$  und  $\lim_{n \rightarrow \infty} \text{cond}_2(T) = \infty$ .

### Aufgabe 3

- (a) Diese Aussage folgt fast sofort aus der zweiten Formulierung der Gauß-Elimination im Skript (Algorithmus 7.8).  $\frac{1}{\alpha^{(k)}}\sigma^{(k)}$  besteht nach dieser Definition einfach aus den unteren  $n - k - 1$  Einträgen von  $l^{(k)}$ ,  $\omega^{(k)}$  aus den unteren  $n - k - 1$  Einträgen von  $u^{(k)}$ . Beim Gaußverfahren würde nun  $l^{(k)} \cdot (u^{(k)})^T$  auf die Matrix  $A^{(k)}$  addiert,  $\frac{1}{\alpha^{(k)}}\sigma^{(k)} \cdot (\omega^{(k)})^T$  entspricht genau der unteren rechten  $(n - k - 1) \times (n - k - 1)$ -Untermatrix von  $l^{(k)} \cdot (u^{(k)})^T$ . Nun wird  $\frac{1}{\alpha^{(k)}}\sigma^{(k)} \cdot (\omega^{(k)})^T$  auf die unteren rechte  $(n - k - 1) \times (n - k - 1)$ -Untermatrix von  $A^{(k)}$  addiert. Insgesamt erhalten wir also durch  $C^{(k)} - \frac{1}{\alpha^{(k)}}\sigma^{(k)}(\omega^{(k)})^T$  genau die  $(n - k - 1) \times (n - k - 1)$ -Untermatrix von  $A^{(k+1)}$ , die wir gemäß unserer Blockzerlegung  $B^{(k+1)}$  nennen.
- (b) Dieser Algorithmus geht die Matrix  $A$  zeilenweise durch, wobei  $i$  stets die aktuelle Zeile angibt. Die erste Zeile findet sich exakt so in der Matrix  $U$  wieder, sodass der Algorithmus bei  $i = 2$  anfangen kann, außerdem dient uns die erste Zeile als Induktionsanfang. Wir nehmen als Induktionsvoraussetzung an, dass der Algorithmus alle Zeilen über Zeile  $i$  bereits korrekt zerlegt hat, d.h. in der Zeile  $\nu$  gilt  $\forall \eta < \nu \ a_{\nu,\eta} = l_{\nu}^{(\eta)}$  und  $\forall \eta \geq \nu$  gilt  $a_{i,\eta} = u_{\eta}^{(i)}$ . Nun führen wir noch eine Induktion über die Spalten durch. Als Induktionsanfang benutzen wir, dass in der dritten Programmzeile für  $j = 2$  der Eintrag  $a_{i,1} = a_{i,1}/a_{1,1}$  gesetzt wird. Das ist genau die Definition von  $l_i^{(1)}$ . Also ist unsere Induktionsvoraussetzung, dass für die ersten  $\nu - 1$  Einträge bereits gilt  $a_{\eta} = l_i^{(\eta)}$ . Die erste for-Schleife geht nun von  $j = 2$  bis  $j = i$ . Für den Eintrag  $a_{ij}$  werden nun durch die Schleife in Programmzeile 4 die Schritte 1 bis  $j - 1$  durchgeführt, sodass nach dem Durchführen der Schleife der Eintrag  $a_{i,\nu}^{(\nu-1)}$  dasteht. Im  $\eta$ -ten Schritt wird nämlich  $a_{i,j}^{(\eta)} = a_{i,j}^{\eta-1} - a_{i,\eta} \cdot a_{\eta,j} = a_{i,j}^{\eta-1} - l_i^{(\eta)} \cdot u_j^{(\eta)}$ , was genau dem im Skript beschriebenen Update entspricht. Für  $\nu \neq i$  wird nun  $j$  auf  $\nu + 1$  gesetzt und in Programmzeile 3 dann  $a_{i,\nu+1-1}^{(\nu-1)} = a_{i,\nu}^{(\nu-1)} = \frac{a_{i,\nu}^{(\nu-1)}}{a_{\nu,\nu}^{(\nu-1)}} = \frac{\tilde{a}_{i,\nu}^{(\nu)}}{u_{\nu}^{(\nu)}} = \frac{\tilde{a}_{i,\nu}^{(j)}}{a_{\nu,\nu}^{(j)}} = l_i^{(j)}$ , da sich die ersten  $\nu$  Einträge der  $\nu$ -te Spalte nach Schritt  $\nu$  nicht mehr ändern (siehe Teilaufgabe a). Ab  $\nu = i$  wird in der zweiten for-Schleife einfach immer  $a_{i,j} = a_{i,j}^{(\nu-1)} = \tilde{a}_{i,j}^{(\nu)} = \tilde{a}_{i,j}^{(i)} = u_j^{(i)}$  gesetzt. (Auch hier benutzen wir, dass sich die ersten  $\nu$  Einträge der  $\nu$ -ten Zeile nach Schritt  $\nu$  nicht mehr ändern.) Damit ist aber schon der Induktionsschluss für die Zeile und damit auch für die Spalten gezeigt.

## Aufgabe 4

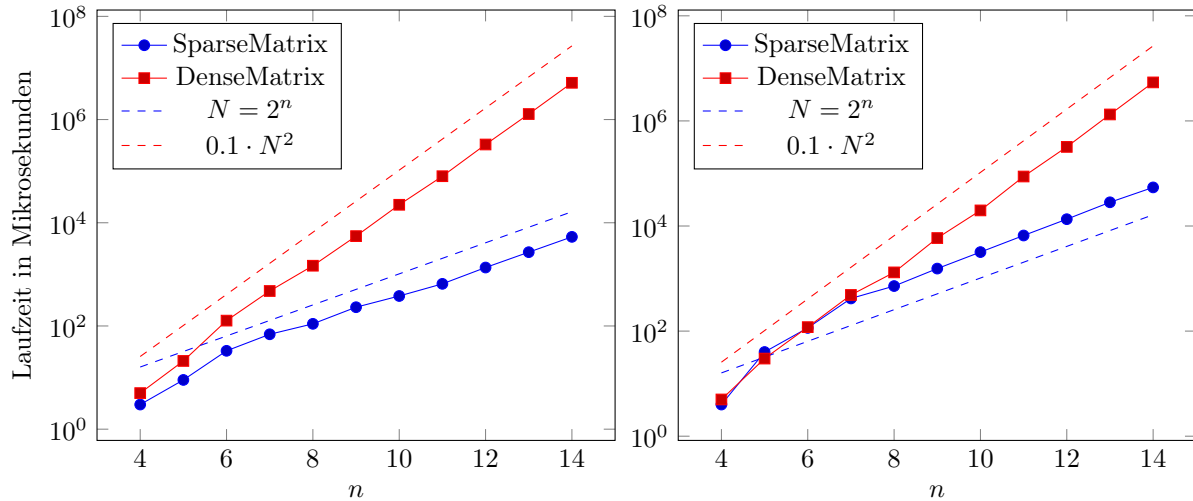


Abbildung 1: Laufzeit der Multiplikation einer Matrix der Dimension  $N \times N$  mit  $N = 2^n$  mit einem Vektor.

In rot ist die Laufzeit bei Implementierung mit DenseMatrix dargestellt. Durch den Vergleich mit der rot gestrichelten Linie erkennt man, dass die Laufzeit quadratisch in der Größe wächst. In blau dargestellt ist die Laufzeit bei Implementierung mit SparseMatrix. Hier sieht man am Vergleich mit einer blau gestrichelten Linie, dass die Laufzeit linear in  $N$  wächst. Auf der linken Seite ist die Bandbreite 10, auf der rechten Seite 100. Man erkennt, dass rechts zu Beginn also DenseMatrix ebenfalls quadratisch wächst, da die Bandbreite noch größer ist als  $N$ . Ab  $n = 7 \Rightarrow N = 128$  spielt die Bandbreite eine Rolle und die Laufzeit wird linear. Das entspricht also exakt den Erwartungen für die Laufzeiten bei konstanter Bandbreite.