Kapitel 1
Angewandte lineare Algebra

1.1 Vier spezielle Matrizen

Eine $m \times n$-Matrix hat m Zeilen, n Spalten und nm Elemente. Wir bearbeiten diese Zeilen und Spalten, um lineare Gleichungssysteme $Ax = b$ und Eigenwertprobleme $Ax = \lambda x$ zu lösen. Aus den Eingaben A und b erhalten wir (mithilfe solcher Software wie MATLAB) die Ausgaben x und λ. Ein schneller und stabiler Algorithmus ist dabei außerordentlich wichtig. Genau das finden Sie in diesem Buch.

Matrizen werden benutzt, um Informationen zu speichern. In der angewandten Mathematik betrachten wir sie hingegen oft unter einem anderen Gesichtspunkt: Eine Matrix ist ein „Operator". ***Der Operator A wirkt auf Vektoren x und erzeugt andere Vektoren Ax.*** Die Komponenten von x haben eine Bedeutung – sie stehen für Auslenkungen, Drücke, Spannungen, Preise oder Konzentrationen. Auch der Operator A hat eine Bedeutung – in diesem Kapitel bildet er Differenzen. Also sind Ax Druckdifferenzen, Spannungsabfälle oder Preisdifferenzen.

Bevor wir das Problem dem Rechner überlassen – und auch später, wenn wir `A\b` oder `eig(A)` interpretieren – sind wir an der Bedeutung genauso interessiert wie an den Zahlen.

Am Anfang dieses Buches stehen **vier spezielle Matrizenfamilien**. Sie sind einfach, nützlich und wirklich fundamental. Wir sehen uns zuerst die Eigenschaften dieser speziellen Matrizen K_n, C_n, T_n und B_n an. (Einige Eigenschaften sind offensichtlich, andere sind versteckt.) Es ist ein Vergnügen, lineare Algebra mit wirklich bedeutsamen Matrizen zu treiben. Es folgen die Matrizen K_2, K_3, K_4 aus der ersten Familie, auf deren Diagonalen die Elemente -1, 2 und -1 stehen:

$$K_2 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \qquad K_3 = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \qquad K_4 = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}.$$

Was macht die Matrizen K_2, K_3, K_4 und schließlich die $n \times n$-Matrix K_n aus? Ich werde sechs Antworten in der Reihenfolge liefern, in der sie meine Studenten ge-

G. Strang et al., *Wissenschaftliches Rechnen*, Springer-Lehrbuch Masterclass,
DOI 10.1007/978-3-540-78495-1_1, © Springer-Verlag Berlin Heidelberg 2010

geben haben. Fangen wir mit vier Eigenschaften der Matrizen K_n an, die Sie sofort einsehen können.

1. Die Matrizen sind **symmetrisch**. Das Element aus Zeile i, Spalte j steht auch in Spalte j, Zeile i. Daher gilt auf gegenüberliegenden Seiten der Hauptdiagonale (der ersten oberen und unteren Nebendiagonale) $K_{ij} = K_{ji}$. Die Symmetrie kann man auch ausdrücken, indem man die gesamte Matrix transponiert: $K = K^{\mathrm{T}}$.
2. Die Matrizen K_n sind **dünn besetzt**. Für große n ist der überwiegende Teil der Elemente null. K_{1000} hat eine Million Elemente, aber nur $1000+999+999$ davon sind von null verschieden.
3. Die von null verschiedenen Elemente liegen auf einem „Band" um die Hauptdiagonale. Daher ist jede Matrix K_n eine *Bandmatrix*. Es gibt nur eine obere und eine untere Nebendiagonale, also sind diese Matrizen **tridiagonal**.

Weil K eine Tridiagonalmatrix ist, kann $Ku = f$ schnell gelöst werden. Wenn der gesuchte Vektor u tausend Komponenten besitzt, können wir diese in einigen tausend Schritten bestimmen (was mit dem Computer einen kleinen Bruchteil einer Sekunde dauert). Bei einer voll besetzten Matrix der Ordnung $n = 1000$ würde die Lösung von $Ku = f$ hunderte Millionen Schritte brauchen. Natürlich müssen wir an erster Stelle fragen, ob die linearen Gleichungen überhaupt eine Lösung haben. Zu dieser Frage kommen wir gleich.

4. Die Matrizen haben **konstante Diagonalen**. Diese Eigenschaft verlangt geradezu nach Fourier-Tranformation. Sie ist ein Hinweis darauf, dass sich eine Größe nicht ändert, wenn wir uns im Raum oder in der Zeit bewegen. Das Problem ist translations- oder zeitinvariant. Die Koeffizienten sind konstant. Die tridiagonale Matrix ist durch die drei Zahlen $-1, 2, -1$ vollständig bestimmt. Diese Matrizen heißen eigentlich „Matrizen der zweiten Differenzen". Meine Studenten benutzen diesen Begriff aber nie.

Die ganze Welt der Fourier-Transformation ist an Matrizen mit konstanten Diagonalen geknüpft. Bei der Signalverarbeitung dient die Matrix $D = K/4$ als „Hochpassfilter". Mit Du wird die sich schnell veränderte (hochfrequente) Komponente eines Vektors u selektiert. Der Ausdruck liefert eine *Faltung* mit $\frac{1}{4}(-1, 2, -1)$. Wir benutzen diese Begriffe schon hier, um die Aufmerksamkeit auf das Kapitel über Fourier-Transformation (Kapitel 4) zu lenken.

Mathematiker bezeichnen die Matrix K als ***Toeplitz-Matrix***, und MATLAB benutzt diese Bezeichnung ebenso:

Der Befehl $K = \mathsf{toeplitz}([2 \quad -1 \quad \mathsf{zeros}(1,2)])$ **konstruiert** K_4 **aus Zeile 1.**

Die Fourier-Transformation lässt sich sogar noch besser anwenden, wenn wir in K_n zwei kleine Änderungen vornehmen. Wir tragen in die obere rechte und die untere linke Ecke das Element -1 ein. Damit sind zwei (zirkulierende) Diagonalen komplett. Jeder Zeilenvektor von C_4 ist nun relativ zum darüberliegenden Zeilenvektor um ein Element nach rechts verschoben. Die Matrix C_4 heißt auch „*periodische Matrix*" oder „*zyklische Faltung*" oder **zirkulante Matrix**:

Zirkulante Matrix $C_4 = \begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix} = \mathsf{toeplitz}([2 \quad -1 \quad 0 \quad -1]).$

Diese Matrix ist *singulär*. Sie ist *nicht invertierbar*. Ihre Determinante ist null. Anstatt diese Determinante zu berechnen, bestimmt man besser einen von null verschiedenen Vektor u, der $C_4u = 0$ löst. (**Hätte C_4 eine Inverse, wäre die einzige Lösung zu $C_4u = 0$ der Nullvektor.** Wir könnten mit C_4^{-1} multiplizieren, um festzustellen, dass $u = 0$ gilt.) Bei dieser Matrix ist es der Spaltenvektor aus lauter Einsen (also $u = (1,1,1,1)$), der die Gleichung $C_4u = 0$ löst.

Die Summe der Spalten der Matrix C ist die Nullspalte. Der Vektor $u = \mathsf{ones}(4,1)$ liegt im *Nullraum* von C_4. Der Nullraum enthält alle Lösungen zu $Cu = 0$.

Wenn die Summe der Elemente in jeder Zeile einer Matrix null ist, ist die Matrix zweifellos singulär. Grund dafür ist wieder der Spaltenvektor aus lauter Einsen. Bei der Matrixmultiplikation Cu werden alle Spaltenvektoren addiert. Die Summe ist jeweils null. Der konstante Vektor $u = (1,1,1,1)$ bzw. $u = (c,c,c,c)$ aus dem Nullraum verhält sich wie die Konstante C bei der Integration einer Funktion. In der Analysis lässt sich die „Integrationskonstante“ nicht aus der Ableitung bestimmen. In der Algebra kann man die Konstante in $u = (c,c,c,c)$ nicht aus der Gleichung $Cu = 0$ bestimmen.

5. Alle Matrizen $K = K_n$ sind **invertierbar**. Im Gegensatz zu den Matrizen C_n sind sie nicht singulär. Es existiert eine Quadratmatrix K^{-1}, sodass $K^{-1}K = I$ gilt. Die Matrix I ist die *Einheitsmatrix*. Hat eine Quadratmatrix eine linksseitige Inverse, dann gilt auch $KK^{-1} = I$. Wenn die Matrix K symmetrisch ist, ist diese „*inverse Matrix*“ ebenfalls symmetrisch. *Allerdings ist die Matrix K^{-1} nicht dünn besetzt.*

Die Invertierbarkeit einer Matrix lässt sich nicht ohne weiteres überblicken. Theoretisch könnte man die Determinante berechnen. Nur wenn $\det K = 0$ ist, existiert keine Inverse, weil die Berechnung von K^{-1} eine Division durch $\det K$ beinhaltet. In der Praxis berechnet man die Determinante jedoch fast nie. Das ist eine umständliche Art, $u = K^{-1}f$ zu bestimmen.

Tatsächlich machen wir mit den Eliminationsschritten weiter, die $Ku = f$ lösen. Diese Schritte vereinfachen die Matrix so, dass sie triangular wird. Die von null verschiedenen Pivotelemente auf der Hauptdiagonalen der tridiagonalen Matrix zeigen, dass die ursprüngliche Matrix K invertierbar ist. (Wichtiger Hinweis: Wir wollen und brauchen K^{-1} nicht, um $u = K^{-1}f$ zu bestimmen. Die Inverse wäre eine voll besetzte Matrix mit positiven Elementen. Wir berechnen nur den Lösungsvektor u.)

6. Die symmetrischen Matrizen K_n sind **positiv definit**. Dieser Begriff ist Ihnen vielleicht neu. Kapitel 1 soll Ihnen unter anderem erklären, was diese wichtige Eigenschaft bedeutet (K_4 besitzt sie, C_4 hingegen nicht). Lassen Sie mich zunächst positive Definitheit und Invertierbarkeit anhand der Begriffe „Pivotelemente“ und „Eigenwerte“ einander gegenüberstellen. Diese Begriffe werden Ihnen bald vertraut sein. *Sehen Sie sich den Anhang an, der die Elemente der linearen Algebra zusammenfasst.*

Pivotelemente: Eine invertierbare Matrix hat *n von null verschiedene* Pivotelemente. Eine *positiv definite*, symmetrische Matrix hat *n* ***positive*** **Pivotelemente**.

Eigenwerte: Eine invertierbare Matrix hat *n von null verschiedene* Eigenwerte. Eine *positiv definite*, symmetrische Matrix hat *n* ***positive*** **Eigenwerte**.

Anhand der Pivotelemente und Eigenwerte lässt sich die positive Definitheit testen. Die Matrix C_4 ist nicht positiv definit, weil sie singulär ist. Tatsächlich hat die Matrix C_4 drei positive Pivotelemente und Eigenwerte, sodass sie den Test „fast" besteht. Aber ihr vierter Eigenwert ist null (die Matrix ist singulär). Da keiner der Eigenwerte negativ ist ($\lambda \geq 0$), bezeichnet man C_4 als **positiv** *semi***definit**.

Wenn wir in Abschnitt 1.3 $Ku = f$ durch Elimination lösen, stehen die Pivotelemente in der Hauptdiagonalen. Die Eigenwerte kommen in $Kx = \lambda x$ vor. Man kann die positive Definitheit auch anhand der Determinante testen (aber nicht einfach $\det K > 0$). Die exakte Definition einer symmetrischen, positiv definiten Matrix (die etwas mit positiver Energie zu tun hat) geben wir in Abschnitt 1.6.

Von K_n zu T_n

Neben den beiden Familien K_n und C_n gibt es zwei weitere Familien, die Sie kennen müssen. Matrizen aus diesen Familien sind symmetrisch und tridiagonal, wie die aus der Familie K_n. Bei Matrizen aus der Familie T_n ist aber das Element an der Position $(1,1)$ nicht 2, sondern 1:

$$\mathbf{T_n(1,1) = 1} \quad T_2 = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} \quad \text{und} \quad T_3 = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}. \tag{1.1}$$

Diese erste Zeile (T steht für *englisch* top) hat mit einer neuen Randbedingung zu tun, deren Bedeutung wir bald verstehen werden. Im Moment benutzen wir die Matrix T_3 als perfektes Demonstrationsbeispiel für die Elimination. Durch Zeilenoperationen machen wir die Elemente unter der Hauptdiagonalen zu null. Die Pivotelemente kennzeichnen wir mit einem Kreis, wenn wir sie bestimmt haben. In **zwei Eliminationsschritten** machen wir aus der Matrix T die **obere Dreiecksmatrix** U.

Schritt 1. Addieren Sie Zeile 1 und Zeile 2, sodass alle Elemente unter dem ersten Pivotelement null sind.

Schritt 2. Addieren Sie die neue Zeile 2 und Zeile 3, sodass U entsteht.

$$T = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \xrightarrow[\text{Schritt 1}]{} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 2 \end{bmatrix} \xrightarrow[\text{Schritt 2}]{} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} = U.$$

Alle Pivotelemente der Matrix T sind 1. Wir können unseren Test für die Invertierbarkeit anwenden (alle Pivotelemente müssen ungleich null sein). Die Matrix T_3 besteht sogar den Test auf positive Definitheit (drei *positive* Pivotelemente). In der Tat ist jede Matrix T_n aus dieser Familie positiv definit, weil immer alle Pivotelemente 1 sind.

Die Matrix U besitzt eine Inverse (die automatisch eine obere Dreiecksmatrix ist). Außergewöhnlich an dieser speziellen Inversen U^{-1} ist, dass *alle Elemente oberhalb der Hauptdiagonalen* 1 *sind*:

$$U^{-1} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \texttt{triu(ones(3))}. \tag{1.2}$$

Daraus lernen wir, dass die Inverse einer 3×3-„Differenzenmatrix" eine 3×3-„**Summenmatrix**" ist. Diese hübsche Inverse der Matrix U wird uns in Aufgabe 1.1.2 auf Seite 10 auf die Inverse der Matrix T führen. **Das Produkt $U^{-1}U$ ist die Einheitsmatrix I.** Die Matrix U bildet Differenzen, ihre Inverse U^{-1} bildet Summen. Werden zunächst Differenzen und dann Summen gebildet, erhält man den ursprünglichen Vektor (u_1, u_2, u_3) zurück:

Differenzen aus U
$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} u_1 - u_2 \\ u_2 - u_3 \\ u_3 - 0 \end{bmatrix}$$

Summen aus U^{-1}
$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 - u_2 \\ u_2 - u_3 \\ u_3 - 0 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}.$$

Von T_n zu B_n

Bei der vierten Familie B_n ist auch das letzte Element nicht 2, sondern 1. Die neue Randbedingung gilt für beide Enden (B steht für *englisch* both). Die Matrizen B_n sind symmetrisch und tridiagonal. Wie sie gleich sehen werden, sind sie *nicht invertierbar*. Die Matrizen B_n sind positiv semidefinit, aber *nicht positiv definit*:

$$\mathbf{B_n(n,n) = 1} \quad B_2 = \begin{bmatrix} \mathbf{1} & -1 \\ -1 & \mathbf{1} \end{bmatrix} \quad \text{und} \quad B_3 = \begin{bmatrix} \mathbf{1} & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & \mathbf{1} \end{bmatrix}. \tag{1.3}$$

Wieder offenbart die Elimination die Eigenschaften der Matrix. Die ersten $n-1$ Pivotelemente sind 1, weil sich hier gegenüber der Matrix T_n nichts geändert hat. Weil aber das letzte Element der Matrix B nun 1 ist, ändert sich das letzte Element der Matrix U:

$$B = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & \mathbf{1} \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & \mathbf{1} \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & \mathbf{0} \end{bmatrix} = U. \tag{1.4}$$

Es gibt nur zwei Pivotelemente. (Ein Pivotelement muss von null verschieden sein.) Die letzte Matrix U ist selbstverständlich nicht invertierbar. Ihre Determinante ist null, weil die dritte Zeile nur Nullen enthält. Der konstante Vektor $(1,1,1)$ liegt im **Nullraum** der Matrix U und daher auch im Nullraum der Matrix B:

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{und ebenso} \quad \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Sinn und Zweck der Elimination war, ein lineares System wie $Bu = 0$ zu vereinfachen, *ohne die Lösungen zu verändern*. Dass die Matrix B nicht invertierbar ist, hätten wir in diesem Fall auch daran sehen können, dass die Summe der Elemente in jeder Zeile null ist. Dann ist die Summe der Spalten die Nullspalte. Genau diese erhalten wir, wenn wir die Matrix B mit dem Vektor $(1,1,1)$ multiplizieren.

Lassen Sie mich diesen Abschnitt in vier Stichpunkten zusammenfassen (alle behandelten Matrizen sind symmetrisch):

Die Matrizen K_n und T_n sind invertierbar (und sogar) positiv definit.

Die Matrizen C_n und B_n sind singulär (und sogar) positiv semidefinit.

Die Nullräume der Matrizen C_n und B_n enthalten die konstanten Vektoren $u = (c, c, \ldots, c)$. Ihre Spalten sind abhängig.

Die Nullräume der Matrizen K_n und T_n enthalten nur den Nullvektor $u = (0, 0, \ldots, 0)$. Ihre Spalten sind unabhängig.

Matrizen in MATLAB

Lineare Algebra wird gern mit MATLAB ausgeführt. Der Leser kann sich aber auch für eine andere Software entscheiden. (Octave ist sehr ähnlich und frei verfügbar. Mathematica und Maple eignen sich für symbolische Berechnungen, LAPACK liefert in netlib kostenlos ausgezeichneten Code, und es gibt viele weitere Pakete für lineare Algebra.) Wir werden in der komfortablen Sprache von MATLAB Matrizen konstruieren und mit ihnen arbeiten.

Im ersten Schritt wollen wir die Matrizen K_n konstruieren. Im Fall $n = 3$ können wir die 3×3-Matrix zeilenweise in eckigen Klammern eingeben. Dabei werden die Zeilen durch ein Semikolon voneinander getrennt:

$$K = [\ 2 \quad -1 \quad 0\ ; \quad -1 \quad 2 \quad -1\ ; \quad 0 \quad -1 \quad 2\ ]\,.$$

Bei großen Matrizen ist diese Vorgehensweise zu zeitaufwändig. Wir können die Matrix K_8 auch mit den Befehlen „eye“ und „ones“ erzeugen:

$$\mathsf{eye}(8) = 8 \times 8 \text{ Einheitsmatrix} \qquad \mathsf{ones}(7,1) = \text{ Spaltenvektor aus sieben Einsen}\,.$$

Die Hauptdiagonalelemente erzeugen wir mit $2{*}\mathsf{eye}(8)$. Das Symbol $*$ steht für die Multiplikation. Die Diagonalelemente über der Hauptdiagonalen der Matrix K_8 werden durch den Vektor $-\mathsf{ones}(7,1)$ auf der ersten oberen Nebendiagonalen der Matrix E erzeugt:

Elemente der oberen Nebendiagonale $\quad E = -\mathsf{diag}(\mathsf{ones}(7,1),1)\,.$

Die Elemente *unter* der Hauptdiagonalen der Matrix K_8 liegen auf der Nebendiagonale mit der Nummer -1 (der ersten unteren Nebendiagonale). Zu ihrer Darstellung können wir im letzten Argument von E aus 1 eine -1 machen. Wir können E aber auch einfach transponieren. In MATLAB ist E' das Symbol für E^{T}. Die Matrix K lässt sich also aus drei Diagonalmatrizen erzeugen:

Tridiagonale Matrix $\boldsymbol{K_8}$ $\quad K = 2 * \mathsf{eye}(8) + E + E'$.

Anmerkung Die nullte Diagonale (Hauptdiagonale) ist per Defaulteinstellung gemeint, wenn das zweite Argument fehlt. Also ist $\mathsf{eye}(8) = \mathsf{diag}(\mathsf{ones}(8{,}1))$. Dann ist $\mathsf{diag}(\mathsf{eye}(8)) = \mathsf{ones}(8,1)$.

Die konstanten Diagonalelemente machen K zu einer Toeplitz-Matrix. Der Befehl toeplitz erzeugt die Matrix K, wenn in jeder Diagonalen nur dasselbe Element 2, -1 oder 0 steht. Benutzen Sie den Befehl zeros für einen Nullvektor, mit dem Sie die sechs Nullen in der ersten Zeile der Matrix K_8 erzeugen:

Symmetrische Toeplitz-Matrix

$\mathrm{row1} = [2\ \ -1\ \ \mathsf{zeros}(1,6)];\ K = \mathsf{toeplitz}(\mathrm{row1})$.

Bei einer unsymmetrischen Matrix mit konstanten Diagonalelementen brauchen Sie den Befehl toeplitz(col1, row1). Mit $\mathrm{col1} = [1\ -1\ 0\ 0]$ und $\mathrm{row1} = [1\ 0\ 0]$ erzeugt dieser Befehl eine 4×3-Matrix der Rückwärtsdifferenzen, die nur auf zwei Diagonalen von null verschiedene Elemente hat, nämlich die Elemente 1 auf der einen und die Elemente -1 auf der anderen Diagonalen.

Um aus der Matrix K die Matrizen T, B und C zu erzeugen, müssen nur einzelne Elemente geändert werden, wie in den letzten drei Zeilen des M-Files mit dem Namen KTBC.m angegeben. Die Eingabe ist die Größe n, erzeugt werden vier Matrizen. Das Semikolon unterdrückt die Anzeige der Matrizen K,T,B und C:

```
function [K,T,B,C] = KTBC(n)
% Erzeuge die vier speziellen Matrizen unter der Annahme n>1
K = toeplitz ([2 −1 zeros(1,n−2)]);
T = K; T(1,1) = 1;
B = K; B(1,1) = 1; B(n,n) = 1;
C = K; C(1,n) = −1; C(n,1) = −1;
```

Würden wir ihre Determinanten bestimmen wollen (das sollten wir nicht!), dann erzeugt im Fall $n = 8$ der Befehl

$$\begin{bmatrix} \det(K) & \det(T) & \det(B) & \det(C) \end{bmatrix} \quad \text{die Ausgabe} \quad \begin{bmatrix} 9 & 1 & 0 & 0 \end{bmatrix}.$$

Übrigens: MATLAB kann die Matrix K_n bei $n = 10\,000$ nicht als voll besetzte Matrix speichern. **Die $\mathbf{10^8}$ Elemente belegen etwa 800 Megabyte Speicher, wenn wir die Matrix K nicht als dünn besetzt erkennen.** Der Code sparseKTBC.m, der auf den Internetseiten zu diesem Buch zu finden ist, vermeidet das Speichern (und Bearbeiten) aller Nullen. Seine ersten beiden Argumente sind eine der Matrizen

K, T, B oder C und die Größe n. Als drittes Argument steht 1 für eine dünn besetzte und 0 für eine voll besetzte Matrix (der Defaultwert ist 0).

Zur Eingabe von sparse in MATLAB gehören die Positionen aller von null verschiedener Elemente. Aus den Vektoren i, j, s, deren Komponenten alle von null verschiedenen Elemente s mit ihren dazugehörigen Positionen i, j sind, erzeugt der Befehl $A =$ sparse(i, j, s, m, n) eine dünn besetzte $m \times n$-Matrix. Bei der Elimination mit lu(A) können weitere von null verschiedene Elemente entstehen (sogenannte fill-ins), deren Positionen die Software richtig bestimmt. Bei der normalen „vollen" Variante werden die Nullen wie alle anderen Zahlen behandelt.

Am günstigsten ist es, zunächst eine Liste mit den Tripeln i, j, s zu erstellen und anschließend sparse aufzurufen. Die Eintragungen mit $A(i, j) = s$ oder $A(i, j) = A(i, j) + s$ sind aufwändiger. Darauf kommen wir in Abschnitt 3.6 zurück.

Der auf den Internetseiten angegebene Code sparseKTBC.m greift auf den Befehl spdiags zurück, um die drei Diagonalen zu füllen. Hier ist die toeplitz-Variante, die die Matrix K_8 erzeugt. Alle Objekte werden dünn besetzt behandelt, weil der erste Vektor als dünn besetzt definiert wurde:

```
vsp = sparse([2 −1 zeros(1, 6)]) % Sehen Sie sich jede Ausgabe an.
Ksp= toeplitz(vsp) % sparse liefert die Positionen der Einträge ungleich 0.
bsp = Ksp(:, 2)      % colon behält alle Zeilen von Spalte 2,
                     % also ist bsp = Spalte 2 von Ksp.
usp = Ksp\bsp        % Die Nullen in Ksp und bsp werden nicht verarbeitet,
                     % Lösung: usp(2) = 1.
uuu= full(usp)       % Kehrt vom dünn besetzten Format zum voll besetzten
                     % Format uuu = [0 1 0 0 0 0 0 0] zurück.
```

Anmerkung Auch die Open-Source-Sprache Python ist sehr attraktiv und zweckmäßig.

In den nächsten Abschnitten werden wir alle vier Matrizen in die folgenden Grundzusammenhänge der linearen Algebra stellen:

(1.2) Die Matrizen K, T, B, C der finiten Differenzen gehören zu Randbedingungen.
(1.3) Elimination erzeugt Pivotelemente in D und triangulare Faktoren in LDL^{T}.
(1.4) Punktlasten führen auf die inversen Matrizen K^{-1} und T^{-1}.
(1.5) Die Eigenwerte und Eigenvektoren der Matrizen K, T, B, C enthalten Sinus- und Kosinusfunktionen.

Behandeln werden wir K\f in Abschnitt 1.2, lu(K) in Abschnitt 1.3, inv(K) in Abschnitt 1.4, eig(K) in Abschnitt 1.5 und chol(K) in Abschnitt 1.6.

Ich hoffe sehr, dass Sie diese speziellen Matrizen kennenlernen und mögen werden.

Anschauungsbeispiele

1.1 A Die Matrixgleichungen $Bu = f$ und $Cu = f$ können selbst dann lösbar sein, wenn die Matrizen B und C singulär sind.

Zeigen Sie, dass jeder Vektor $f = Bu$ die Eigenschaft $f_1 + f_2 + \cdots + f_n = 0$ besitzt. Physikalische Bedeutung: **Die äußeren Kräfte heben sich auf**. In der linearen Algebra bedeutet es: Die Matrixgleichung $Bu = f$ ist lösbar, wenn f orthogonal zum Spaltenvektor $e = (1, 1, 1, 1, \dots) = \mathtt{ones}(n, 1)$ ist.

Lösung Bu ist ein Vektor aus Differenzen der Komponenten von u. Die Summe dieser Differenzen ist stets null:

$$f = Bu = \begin{bmatrix} 1 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} u_1 - u_2 \\ -u_1 + 2u_2 - u_3 \\ -u_2 + 2u_3 - u_4 \\ -u_3 + u_4 \end{bmatrix}.$$

Alle Terme in $(u_1 - u_2) + (-u_1 + 2u_2 - u_3) + (-u_2 + 2u_3 - u_4) + (-u_3 + u_4) = 0$ heben sich gegenseitig auf. Das Skalarprodukt von f mit dem Vektor $e = (1, 1, 1, 1)$ ist die Summe $f^{\mathrm{T}} e = f_1 + f_2 + f_3 + f_4 = 0$:

Skalarprodukt $f \cdot e = f^{\mathrm{T}} e = f_1 e_1 + f_2 e_2 + f_3 e_3 + f_4 e_4$ (in MATLAB $\mathtt{f}' * \mathtt{e}$).

Eine zweite Erklärung für $f^{\mathrm{T}} e = 0$ geht von der Tatsache aus, dass $Be = 0$ ist. Der Vektor e ist im Nullraum der Matrix B. Transponieren von $f = Bu$ liefert $f^{\mathrm{T}} = u^{\mathrm{T}} B^{\mathrm{T}}$, **weil die Transponierte eines Produktes das Produkt der Transponierten der einzelnen Faktoren des Produktes in umgekehrter Reihenfolge ist.** Die Matrix B ist symmetrisch, sodass $B^{\mathrm{T}} = B$ gilt. Dann ist

$$f^{\mathrm{T}} e = u^{\mathrm{T}} B^{\mathrm{T}} e \quad \text{gleichbedeutend mit} \quad u^{\mathrm{T}} Be = u^{\mathrm{T}} 0 = 0.$$

Fazit: Die Matrixgleichung $Bu = f$ ist nur lösbar, wenn f orthogonal zu e ist. (*Gleiches gilt für $Cu = f$*. Auch in diesem Fall heben sich die Differenzen auf.) **Die äußeren Kräfte heben sich auf,** wenn die Summe der Komponenten des Vektors f null ist. Der Befehl `B\f` liefert `Inf`, weil die Matrix B quadratisch und singulär ist. Doch die „Pseudoinverse" `u = pinv(B) * f` existiert und wird ausgegeben. (Alternativ können Sie der Matrix B und dem Vektor f eine Zeile aus Nullen hinzufügen, ehe Sie den Befehl `B\f` eingeben, um aus der quadratischen Matrix eine Rechteckmatrix zu machen.)

1.1 B Bei der Matrix H zu „fest-freien" Randbedingungen wird aus dem *letzten Element der Matrix K* eine 1. Stellen Sie einen Zusammenhang zwischen der Matrix H und der Matrix T (zu „frei-festen" Randbedingungen, *erstes Matrixelement* = 1) her, indem Sie die umgekehrte Einheitsmatrix J benutzen:

$$H = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \quad \begin{array}{l} \text{entsteht durch } JTJ \text{ mit der} \\ \textit{umgekehrten Einheitsmatrix } J \end{array} \quad J = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Kapitel 2 zeigt, wie sich die Matrix T aus einer *Turmstruktur* (freie Randbedingungen oben) ergibt. Die Matrix H ist mit einer *hängenden* Struktur (freie Randbedingungen unten) verknüpft. Zwei MATLAB-Befehle dafür sind:

$$H = \mathsf{toeplitz}([2 \quad -1 \quad 0]);\ H(3,3) = 1 \ \textbf{oder}\ J = \mathsf{fliplr}(\mathsf{eye}(3));\ H = J * T * J\,.$$

Lösung Das Produkt JT vertauscht die Zeilen der Matrix T. Anschließend vertauscht JTJ die Spalten, sodass die Matrix H entsteht:

$$T = \begin{bmatrix} \mathbf{1} & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \qquad \begin{matrix} JT = \\ \text{(Zeilen)} \end{matrix} \begin{bmatrix} 0 & -1 & 2 \\ -1 & 2 & -1 \\ \mathbf{1} & -1 & 0 \end{bmatrix} \qquad \begin{matrix} (JT)J = H\,. \\ (\textit{Spalten ebenso}) \end{matrix}$$

Wir hätten mit dem Produkt TJ auch zuerst die Spalten vertauschen können. Anschließend hätte $J(TJ)$ auf dieselbe Matrix H geführt wie $(JT)J$. **In $(AB)C = A(BC)$ kommt es auf die Klammern nicht an.**

Jede Permutationsmatrix, wie etwa die Matrix J, besitzt die Zeilen der Einheitsmatrix in einer gewissen Reihenfolge. Es gibt sechs 3×3-Permutationsmatrizen, weil es sechs Permutationen der Zahlen $1,2,3$ gibt. *Die Inverse jeder Permutationsmatrix ist ihre Transponierte.* Die Permutationsmatrix J ist symmetrisch, sodass $J = J^{\mathrm{T}} = J^{-1}$ ist, wie Sie leicht überprüfen können:

$$H = JTJ, \quad \text{sodass} \quad H^{-1} = J^{-1}T^{-1}J^{-1} \quad \text{und} \quad H^{-1} = JT^{-1}J. \tag{1.5}$$

Verwendet man in MATLAB back = 3:−1:1, erfolgt die Umordnung auf JTJ mit dem Befehl H = T(back, back).

Aufgaben zu Abschnitt 1.1

Die Aufgaben 1.1.1–1.1.4 befassen sich mit der inversen Matrix T^{-1}. Die Aufgaben 1.1.5–1.1.8 behandeln die inverse Matrix K^{-1}.

1.1.1 Die Inversen der Matrizen T_3 und T_4 (mit dem Element $T_{11} = 1$) sind

$$T_3^{-1} = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{und} \quad T_4^{-1} = \begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Raten Sie die Matrix T_5^{-1} und multiplizieren Sie sie mit der Matrix T_5. Schreiben Sie zuerst eine einfache Gleichung für die Elemente der Matrix T_n^{-1} unterhalb der Hauptdiagonalen $(i \geq j)$ auf und anschließend eine für die Elemente oberhalb der Hauptdiagonalen $(i \leq j)$.

1.1.2 Berechnen Sie aus den Matrizen U und U^{-1} aus Gleichung (1.2) auf Seite 5 die Matrix T_3^{-1} in den drei folgenden Schritten:

1. Prüfen Sie die Gültigkeit der Gleichung $T_3 = U^{\mathrm{T}}U$. Auf der Hauptdiagonalen der Matrix U stehen die Elemente 1, auf der oberen Nebendiagonalen die Elemente -1. Die Transponierte U^{T} ist eine untere Dreiecksmatrix.

2. Prüfen Sie die Gültigkeit der Gleichung $UU^{-1} = I$, wenn die Elemente der Inversen U^{-1} auf der Hauptdiagonalen und oberhalb davon 1 sind.
3. Invertieren Sie $U^{\mathrm{T}}U$, um die Inverse $T_3^{-1} = (U^{-1})(U^{-1})^{\mathrm{T}}$ zu bestimmen. *Beim Invertieren wird die Reihenfolge vertauscht!*

1.1.3 Die Differenzenmatrix $U = U_5$ ist in MATLAB eye(5)−diag(ones(4,1),1). Konstruieren Sie die Summenmatrix S aus triu(ones(5)). (Dieser Befehl erhält den oberen Dreiecksteil der 5×5-Matrix aus lauter Einsen.) Führen Sie die Multiplikation $U * S$ aus, um sich davon zu überzeugen, dass $S = U^{-1}$ gilt.

1.1.4 Für alle n ist die Matrix $S_n = {U_n}^{-1}$ eine obere Dreiecksmatrix, die in der Hauptdiagonalen und oberhalb nur die Elemente 1 hat. Überprüfen Sie im Fall $n = 4$, dass SS^{T} die Matrix ${T_4}^{-1}$ aus Aufgabe 1.1.1 erzeugt. Warum ist die Matrix SS^{T} symmetrisch?

1.1.5 Die Inversen der Matrizen K_3 und K_4 (bitte invertieren Sie auch die Matrix K_2) enthalten die Brüche $\dfrac{1}{\det} = \dfrac{1}{4}, \dfrac{1}{5}$:

$$K_3^{-1} = \frac{1}{4}\begin{bmatrix} 3 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 3 \end{bmatrix} \quad \text{und} \quad K_4^{-1} = \frac{1}{5}\begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 6 & 4 & 2 \\ 2 & 4 & 6 & 3 \\ 1 & 2 & 3 & 4 \end{bmatrix}.$$

Erraten Sie zunächst die Determinante der Matrix $K = K_5$. Berechnen Sie anschließend $\mathsf{det}(K)$, $\mathsf{inv}(K)$ und $\mathsf{det}(K)*\,\mathsf{inv}(K)$ – Sie dürfen dazu jede Software benutzen.

1.1.6 (anspruchsvoll) Stellen Sie eine Gleichung für das Element i, j der Matrix K_4^{-1} *unter der Hauptdiagonalen* $(i \geq j)$ auf. Die Elemente wachsen auf jeder Zeile und jeder Spalte linear. (In Abschnitt 1.4 kommen wir auf diese wichtigen Inversen zurück.) Die folgende Aufgabe 1.1.7 wird im Anschauungsbeispiel aus Abschnitt 1.4 ausgebaut.

1.1.7 Multipliziert man einen Spaltenvektor u mit einem Zeilenvektor v^{T}, entsteht die Matrix uv^{T} vom Rang 1. Alle Spalten sind Vielfache des Vektors u, alle Zeilen sind Vielfache des Vektors v^{T}. ${T_4}^{-1} - {K_4}^{-1}$ ist vom Rang 1:

$$T_4^{-1} - K_4^{-1} = \frac{1}{5}\begin{bmatrix} 16 & 12 & 8 & 4 \\ 12 & 9 & 6 & 3 \\ 8 & 6 & 4 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix} = \frac{1}{5}\begin{bmatrix} 4 \\ 3 \\ 2 \\ 1 \end{bmatrix}\begin{bmatrix} 4 & 3 & 2 & 1 \end{bmatrix}.$$

Schreiben Sie die Matrix $K_3 - T_3$ in dieser besonderen Form als uv^{T}. Überlegen Sie sich eine ähnliche Gleichung für die Matrix $T_3^{-1} - K_3^{-1}$.

1.1.8 (a) Erraten Sie mit dem Ergebnis aus Aufgabe 1.1.7 das Element i, j der Matrix $T_5^{-1} - K_5^{-1}$ unter der Hauptdiagonalen.

(b) Subtrahieren Sie das Ergebnis aus (a) von der Lösung aus Aufgabe 1.1.1 (der Gleichung für die Elemente der Matrix T_5^{-1} mit $i \geq j$). Das führt auf eine nichttriviale Gleichung für die Elemente der Matrix K_5^{-1}.

1.1.9 Folgen Sie dem Anschauungsbeispiel **1.1 A**, wobei Sie die Matrix B durch die Matrix C ersetzen. Zeigen Sie, dass der Vektor $e = (1,1,1,1)$ orthogonal zu jeder Spalte der Matrix C_4 ist. Lösen Sie $Cu = f = (1,-1,1,-1)$ mit der singulären Matrix C durch den Befehl $\mathsf{u} = \mathsf{pinv(C)} * \mathsf{f}$. Testen Sie den Befehl $\mathsf{u} = \mathsf{C}\backslash\mathsf{e}$ und $\mathsf{C}\backslash\mathsf{f}$ vor und nach dem Hinzufügen einer fünften Gleichung $0 = 0$.

1.1.10 In der Matrix H zu „hängenden Randbedingungen" aus dem Anschauungsbeispiel **1.1 B** wird das letzte Element der Matrix K_3 auf $H_{33} = 1$ geändert. Bestimmen Sie die inverse Matrix mit $H^{-1} = JT^{-1}J$. Bestimmen Sie die inverse Matrix auch mit $H = UU^{\mathrm{T}}$ (obere mal untere Dreiecksmatrix) und $H^{-1} = (U^{-1})S^{\mathrm{T}}U^{-1}$.

1.1.11 Die Matrix U sei eine obere Dreiecksmatrix und die Matrix J die umgekehrte Einheitsmatrix aus dem Anschauungsbeispiel **1.1 B**. Dann ist die Matrix JU eine „Südostmatrix". Welche geographische Lage haben die Matrizen UJ und JUJ? Machen Sie ein Experiment: Das Produkt einer Südostmatrix und einer Nordwestmatrix ist _____ ?

1.1.12 Wenden Sie das Eliminationsverfahren auf die 4×4-Matrix C_4 an, um eine obere Dreiecksmatrix U zu erzeugen (oder probieren Sie den MATLAB-Befehl $[L,U] = \mathsf{lu}(C)$). Zwei Bemerkungen: Das letzte Element der Matrix U ist _____ , weil die Matrix C singulär ist. Die letzte Spalte der Matrix U hat neue, von null verschiedene Elemente. Erklären Sie, woher diese „fill-ins" kommen.

1.1.13 Kann man die zirkulante Matrix C_4 (die nur auf drei Diagonalen von null verschiedene Elemente hat) in LU faktorisieren? Die Matrizen L und U sollen dabei zirkulant sein und nur auf zwei Diagonalen von null verschiedene Elemente haben. (Die Matrizen haben dann keine echte Dreiecksgestalt.)

1.1.14 Reduzieren Sie die Diagonalelemente $2, 2, 2$ der Matrix K_3 durch schrittweises Umformen so lange, bis eine singuläre Matrix M entsteht. Dazu müssen die Diagonalelemente _____ sein. Prüfen Sie dabei die Determinante und bestimmen Sie einen von null verschiedenen Vektor, der $Mu = 0$ löst.

Die Aufgaben 1.1.15–1.1.21 befassen sich mit wesentlichen Eigenschaften der Matrixmultiplikation.

1.1.15 Wie viele Multiplikationen sind notwendig, um Ax, A^2 und AB zu berechnen?

$$A_{n \times n} x_{n \times 1} \qquad A_{n \times n} A_{n \times n} \qquad A_{m \times n} B_{n \times p} = (AB)_{m \times p}$$

1.1.16 Sie können die Multiplikation Ax zeilenweise (wie üblich) oder **spaltenweise** (bedeutsamer) ausführen. Probieren Sie beide Varianten:

Zeilenweise $$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} \text{Skalarprodukt aus Zeile 1} \\ \text{Skalarprodukt aus Zeile 2} \end{bmatrix}$$

Spaltenweise $$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = 1 \begin{bmatrix} 2 \\ 4 \end{bmatrix} + 2 \begin{bmatrix} 3 \\ 5 \end{bmatrix} = \begin{bmatrix} \textbf{Kombination} \\ \textbf{der Spalten} \end{bmatrix}.$$

1.1.17 Das Produkt Ax ist eine **Linearkombination der Spalten der Matrix** A. Die Gleichungen $Ax = b$ haben genau dann einen Lösungsvektor x, wenn der Vektor b eine _____ der Spalten ist.
Geben Sie ein Beispiel für den Fall an, dass der Vektor b *nicht im Spaltenraum* der Matrix A liegt. Es gibt keine Lösung zu $Ax = b$, weil der Vektor b keine Kombination der Spalten der Matrix A ist.

1.1.18 Berechnen Sie das Produkt $C = AB$, indem Sie die Matrix A mit jeder Spalte der Matrix B multiplizieren:

$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 8 & * \\ 14 & * \end{bmatrix}.$$

Folglich ist A $*$ B(:,j) = C(:,j).

1.1.19 Sie können das Produkt AB auch berechnen, indem Sie jede Zeile der Matrix A mit der Matrix B multiplizieren:

$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 2 * \text{Zeile } 1 + 3 * \text{Zeile } 2 \\ 4 * \text{Zeile } 1 + 5 * \text{Zeile } 2 \end{bmatrix} = \begin{bmatrix} 8 & 16 \\ * & * \end{bmatrix}.$$

Warum ist eine Lösung zu $Bx = 0$ gleichzeitig auch eine Lösung zu $(AB)x = 0$? Wie kommen wir von

$$Bx = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{auf} \quad ABx = \begin{bmatrix} 8 & 16 \\ * & * \end{bmatrix} \begin{bmatrix} -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}?$$

1.1.20 Die folgenden vier Möglichkeiten, das Produkt AB zu bestimmen, ergeben Zahlen, Spalten, Zeilen und **Matrizen**:

1	***(Zeilen der Matrix A)* mal *(Spalten der Matrix B)***	C(i,j) = A(i,:) $*$ B(:,j)
2	***Matrix A* mal *(Spalten der Matrix B)***	C(:,j) = A $*$ B(:,j)
3	***(Zeilen der Matrix A)* mal *Matrix B***	C(i,:) = A(i,:) $*$ B
4	***(Spalten der Matrix A)* mal *(Zeilen der Matrix B)***	for k = 1:n, C = C + A(:,k) $*$ B(k,:); end

Beenden Sie diese 8 Multiplikationen von **Spalten mal Zeilen**. Wie viele Multiplikationen sind es im Fall $n \times n$?

$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix} \begin{bmatrix} 1 & 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 5 \end{bmatrix} \begin{bmatrix} 2 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 4 & 8 \end{bmatrix} + \begin{bmatrix} * & * \\ * & * \end{bmatrix} = \begin{bmatrix} 8 & * \\ * & * \end{bmatrix}.$$

1.1.21 Welche dieser Gleichungen gilt für alle $n \times n$-Matrizen A und B?

$$AB = BA \qquad (AB)A = A(BA) \qquad (AB)B = B(BA) \qquad (AB)^2 = A^2B^2\,.$$

1.1.22 Verwenden Sie $n = 1000$; $e = \mathsf{ones}(n,1)$; $K = \mathsf{spdiags}([-e, 2*e, -e], -1:1, n, n)$; in MATLAB, um die dünn besetzte Matrix K_{1000} einzugeben. Lösen Sie die Gleichung $Ku = e$ durch den Befehl u = K\e. Stellen Sie die Lösung mit dem Befehl plot(u) grafisch dar.

1.1.23 Erzeugen Sie Vektoren u, v, w mit vier Komponenten und geben Sie den Befehl $\mathsf{spdiags}([u, v, w], -1:1, 4, 4)$ ein. Welche Komponenten der Vektoren u und w bleiben in den ersten oberen und unteren Nebendiagonalen der Matrix A unberücksichtigt?

1.1.24 Erzeugen Sie die dünn besetzte Einheitsmatrix $I = \mathsf{sparse}(i, j, s, 100, 100)$, indem Sie Vektoren i, j, s aus Positionen i, j mit den von null verschiedenen Elementen s erzeugen. (Sie können dazu eine for-Schleife benutzen.) In diesem Fall ist $\mathsf{speye}(100)$ schneller. Bedenken Sie aber, dass der Befehl $\mathsf{sparse}(\mathsf{eye}(10000))$ zu einer Katastrophe führen würde, denn es gibt keinen Platz, um $\mathsf{eye}(10000)$ zu speichern, bevor der Befehl sparse ausgeführt wird.

1.1.25 Die einzige Lösung zu $Ku = 0$ oder $Tu = 0$ ist der Vektor $u = 0$, sodass die Matrizen K und T invertierbar sind. Zum Beweis nehmen wir an, dass u_i die größte Komponente des Vektors u ist. Aus $-u_{i-1} + 2u_i - u_{i+1}$ gleich null folgt $u_{i-1} = u_i = u_{i+1}$. Aus den nächsten Gleichungen ergibt sich anschließend $u_j = u_i$. Wenn wir schließlich bei den Randbedingungen ankommen, ist $-u_{n-1} + 2u_n$ nur dann null, wenn $u = 0$ gilt. Warum versagt dieses Argument der „Diagonaldominanz" bei den Matrizen B und C?

1.1.26 Für welche Vektoren v ist $\mathsf{toeplitz}(v)$ eine zirkulante Matrix?

1.1.27 (bedeutsam) Zeigen Sie, dass sich die 3×3-Matrix K aus $A_0^{\mathrm{T}} A_0$ ergibt:

$$A_0 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad \text{ist eine „}\textbf{Differenzenmatrix}\text{"}.$$

Welche Spalte der Matrix A_0 würden Sie streichen, um die Matrix A_1 mit $T = A_1^{\mathrm{T}} A_1$ zu erzeugen? Welche Spalte würden Sie anschließend streichen, um die Matrix A_2 mit $B = A_2^{\mathrm{T}} A_2$ zu erzeugen? Die Differenzenmatrizen A_0, A_1, A_2 gehören zu 0, 1, 2-Randbedingungen. Das gilt auch für die „Matrizen der zweiten Differenzen" K, T und B.

1.2 Differenzen, Ableitungen und Randbedingungen

Dieser wichtige Abschnitt stellt eine Verbindung zwischen Differenzengleichungen und Differentialgleichungen her. Eine typische Zeile in unseren Matrizen hat die Elemente $-1, 2, -1$. Wir wollen verstehen, wie aus diesen Zahlen eine **zweite Differenz** (oder genauer eine zweite Differenz mit negativem Vorzeichen) wird. Die zweite Differenz ist eine natürliche Näherung für die **zweite Ableitung**. Die Matrizen K_n, C_n, T_n und B_n kommen alle in der Näherung zu folgender Gleichung vor:

$$-\frac{d^2u}{dx^2} = f(x) \quad \text{mit Randbedingungen bei } x = 0 \text{ und } x = 1\,. \tag{1.6}$$

Beachten Sie, dass die Variable nicht t, sondern x ist. Das Problem ist kein Anfangswertproblem, sondern ein Randwertproblem. Es gibt Randbedingungen bei $x = 0$ und $x = 1$ und keine Anfangsbedingungen bei $t = 0$. Die Bedingungen spiegeln sich