Skript zur Vorlesung

Numerische Methoden

für die Fachrichtungen Elektro- und Informationstechnik, Meteorologie, Geodäsie und Geoinformatik

Prof. Dr. Wolfgang Reichel
Institut für Analysis, Fakultät für Mathematik, KIT

Unter Verwendung von Skripten von
Prof. Dr. Wilhelm Niethammer
Institut für angewandte und numerische Mathematik, Fakultät für Mathematik, KIT

Gehalten von Prof. Dr. Michael Plum
Institut für Analysis, Fakultät für Mathematik, KIT

Sommersemester 2016

Inhaltsverzeichnis

Vorwort

Dieses Skript entstand anläßlich der Vorlesung *Numerische Mathematik für die Fachrichtung Elektroingenieurwesen* im Sommersemester 2009 und wurde zum Sommersemester 2010 in seine erste Form gebracht. Als Vorlage dienten einerseits die Notizen von Prof. Dr. Michael Plum und andererseits die in vielen Semestern erprobten und detailliert ausgearbeiteten Numerik-Skripten von Prof. Dr. Wilhelm Niethammer. Beiden Kollegen danke ich sehr für die Zurverfügungstellung ihres Materials. Ebenso danke ich herzlich Frau Dipl.-Math. Susanne Pohlig, Frau Dipl.-Math. Dorothee Frey und Herrn Dipl.-Math.techn. Rainer Mandel für die Arbeit, das Skript zu tippen, die Bilder zu erstellen und das Ergebnis Korrektur zu lesen.

Beginnend mit dem Sommersemester 2011 wurde das Skript neu angepasst. Insbesondere fielen aus der numerischen linearen Algebra (Kapitel 1 und Kapitel 2) die QR-Zerlegung und das QR-Verfahren heraus. Anstatt dessen beschränkte ich mich bei den Eigenwertverfahren in Kapitel 2 auf die sogenannte Potenzmethode (bzw. Vektoriteration oder von-Mises-Iteration). Innerhalb des Skriptes finden sich die Verfahren zwar weiterhin — sie sind aber nicht Gegenstand der aktuellen Konzeption der Vorlesung und sind als sogenannte *Ergänzungen* kenntlich gemacht.

Durch diese Verkürzung wird in der Vorlesung Platz geschaffen für eine ausführlicher Behandlung der finite-Differenzen- und der finite-Elementeverfahren zur Lösung partieller Differentialgleichungen.

Karlsruhe, im Sommersemester 2014

Wolfgang Reichel

Kapitel 1

Direkte Verfahren zur Auflösung linearer Gleichungssysteme

1.1 Einleitung

In diesem Paragraphen befassen wir uns mit *direkten Methoden zur Auflösung linearer Gleichungssysteme*. Als direkt (im Gegensatz zu iterativ) bezeichnet man diejenigen Verfahren, welche es im Prinzip gestatten, die Lösung eines linearen Gleichungssystems in endlich vielen Schritten zu berechnen. Auf lineare Gleichungssyteme stößt man in der Praxis an vielen Stellen. Wir befassen uns hier nur mit einigen besonders typischen Situationen.

Elektrische Netzwerke lassen sich mit Hilfe der *Kirchhoffschen Regeln* beschreiben. (Ähnliche Zusammenhänge findet man auch in der Statik, etwa bei der Behandlung von Fachwerken.) Wir betrachten ein Netzwerk mit Spannungsquellen und Widerständen des folgenden, in Abbildung 4.1 dargestellten Typs.

Es gelten folgende Regeln, die das Netzwerk vollständig beschreiben:

- *Knotenregel*: An jedem Knotenpunkt ist die Summe der zufließenden Ströme gleich der abfließenden Ströme, d.h. es gilt (unter Berücksichtigung der Stromrichtung):
$$\sum I_k = 0.$$
- *Maschenregel*: In jedem beliebig herausgegriffenen, in sich geschlossenen Stromkreis („Masche“) ist die Summe der Spannungsabfälle in den einzelnen Zweigen gleich der Summe der vorhandenen elektromotrischen Kräfte:
$$\sum I_k R_k = \sum U_k.$$

Bei gegebenen Widerständen R_k und elektromotorischen Kräften U_k erhält man offensichtlich ein lineares Gleichungssystem für die Ströme I_k, wenn man alle Knoten und Maschen betrachtet. Möglicherweise ergeben sich dabei mehr Gleichungen als Unbekannte; die *redundanten* Gleichungen sind dann zu streichen.
Wir diskutieren exemplarisch das in Abbildung 4.1 dargestellte Netzwerk. Die Kirchhoffschen Regeln ergeben je vier Knoten- und Maschengleichungen für die unbekannten

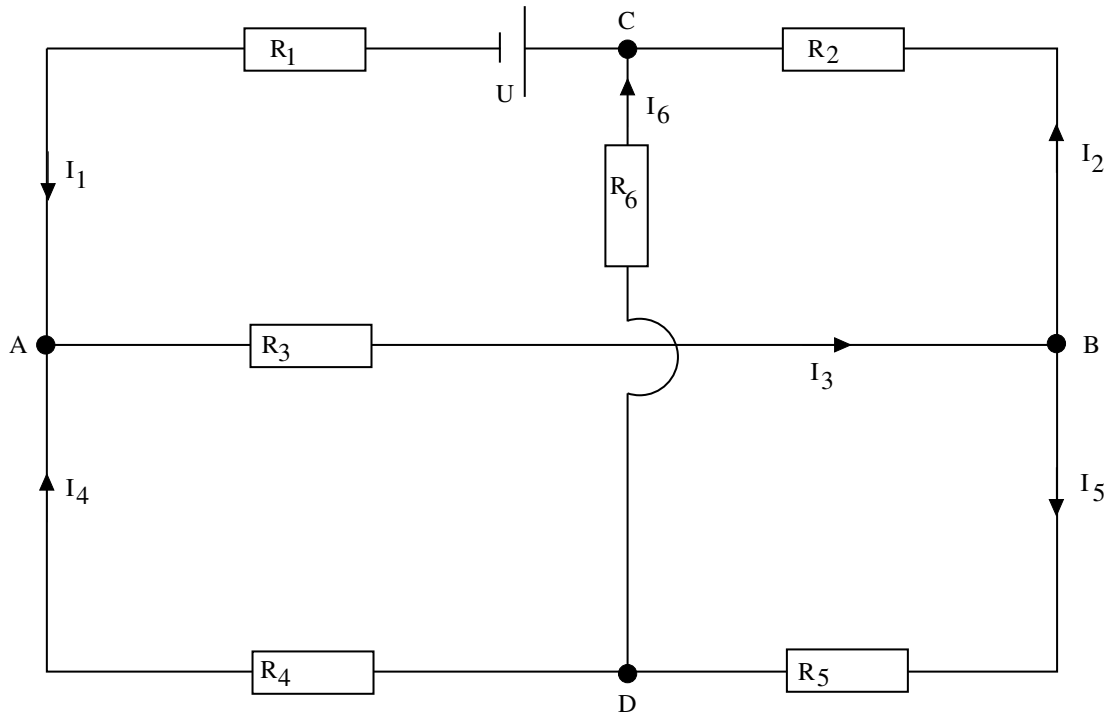


Abbildung 1.1: Schaltbild eines elektrischen Netzwerkes

Ströme $I_1, ..., I_6$. Es wird sich zeigen, dass von diesen acht Gleichungen zwei redundant sind. Im einzelnen gilt:

$$
\begin{array}{lrrrrrrcl}
\text{Knoten A :} & I_1 & & -I_3 & +I_4 & & & = & 0 \\
\text{Knoten B :} & & -I_2 & +I_3 & & -I_5 & & = & 0 \\
\text{Knoten C :} & -I_1 & +I_2 & & & & +I_6 & = & 0 \\
\text{Knoten D :} & & & & -I_4 & +I_5 & -I_6 & = & 0 \\
 & & & & & & & & \\
\text{Maschen ADC:} & R_1 I_1 & & & -R_4 I_4 & & +R_6 I_6 & = & \text{U} \\
\text{Maschen BCD:} & & R_2 I_2 & & & -R_5 I_5 & -R_6 I_6 & = & 0 \\
\text{Maschen ABD:} & & & R_3 I_3 & +R_4 I_4 & +R_5 I_5 & & = & 0 \\
\text{Maschen ABC:} & R_1 I_1 & +R_2 I_2 & +R_3 I_3 & & & & & \text{U}
\end{array}
$$

Wir stellen das System in Matrix-Vektor-Schreibweise dar:

$$
\begin{bmatrix}
1 & 0 & -1 & 1 & 0 & 0 \\
0 & -1 & 1 & 0 & -1 & 0 \\
-1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & -1 & 1 & -1 \\
R_1 & 0 & 0 & -R_4 & 0 & R_6 \\
0 & R_2 & 0 & 0 & -R_5 & -R_6 \\
0 & 0 & R_3 & R_4 & R_5 & 0 \\
R_1 & R_2 & R_3 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ U \\ 0 \\ 0 \\ U \end{bmatrix}
$$

Man erkennt, dass die Gleichungen z.T. voneinander linear abhängig sind, und zwar die vierte von den drei ersten und die letzte von der fünften, sechsten und siebten. Diese linear abhängigen (redundanten) Gleichungen können wir streichen und erhalten

so das System:

$$\begin{bmatrix} 1 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 1 \\ R_1 & 0 & 0 & -R_4 & 0 & R_6 \\ 0 & R_2 & 0 & 0 & -R_5 & -R_6 \\ 0 & 0 & R_3 & R_4 & R_5 & 0 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ U \\ 0 \\ 0 \end{bmatrix}$$

Wir notieren uns typische Eigenschaften dieses Gleichungssystems:

- Die Zeilen der Koeffizienten, welche von den Knotengleichungen herstammen, enthalten nur die Werte -1, 0, 1. Dies rührt von der *topologischen Struktur* des Netztes her; man erkennt, welche Knoten direkt miteinander verbunden sind. Die Knotengleichungen sind außerdem homogen.
- Die Maschengleichungen sind imhomogen, wobei die Inhomogenitäten von den elektromotorischen Kräften herstammen.
- Das System hat eine eindeutig bestimmte Lösung zu jeder rechten Seite. Man kann diesen Sachverhalt physikalisch „beweisen“: Wenn keine elektromotorische Kraft U anliegt, fließen keine Ströme I_k; also hat das homogene System nur die triviale Lösung und jedes zugehörige System ist folglich auf genau eine Weise lösbar.

1.2 Das Eliminationsverfahren von Gauß und die LR-Zerlegung

In der linearen Algebra ist Ihnen das Prinzip des *Gaußschen Eliminationsverfahrens* möglicherweise schon begegnet. Wir verwenden dieses Verfahren zur Lösung eines gegebenen linearen Gleichungssystems. Wir gehen aus von einem System

$$Ax = b;$$

dabei sei A eine gegebene reguläre Matrix mit m Zeilen und m Spalten, x der gesuchte Lösungsvektor und b ein gegebener Vektor, also $A \in \mathbb{K}^{m\times m}$, $x \in \mathbb{K}^m$, $b \in \mathbb{K}^m$, wobei $\mathbb{K}$ den Körper der reellen oder der komplexen Zahlen bezeichne.

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \begin{bmatrix} \\ x \\ \\ \end{bmatrix} = \begin{bmatrix} \\ b \\ \\ \end{bmatrix}.$$

Das Ziel des Eliminationsverfahrens von Gauß ist es, dieses Ausgangssystem so umzuformen, dass die Komponenten des Lösungsvektors x sich leicht berechnen lassen. Man versucht zu diesem Zweck, aus dem gegebenen System ein äquivalentes System

aufzubauen, dessen Koeffizientenmatrix $\tilde{A}$ *Dreiecksgestalt* hat:

$$Ax = b \qquad \Leftrightarrow \qquad \tilde{A}x = \tilde{b}$$

$$\underbrace{\begin{bmatrix} * & * & * & \dots & \dots & * \\ 0 & * & * & \dots & \dots & * \\ \vdots & \ddots & \ddots & & & \vdots \\ \vdots & & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & * \\ 0 & \dots & \dots & \dots & 0 & * \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} * \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ * \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} * \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ * \end{bmatrix}}_{\tilde{b}}$$

(* bezeichnet Elemente, die möglicherweise von Null verschieden sind.)

Diese Reduktion auf Dreiecksgestalt kann man dadurch erreichen, dass man *Gleichungen vertauscht und Vielfache von Gleichungen zu anderen Gleichungen addiert.*

Aufgabe 1.1.
Man bringe das Gleichungssystem

$$\begin{bmatrix} 0 & 5 & 2 \\ 4 & 2 & -1 \\ 6 & -5 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \\ 36 \end{bmatrix}$$

auf Dreiecksgestalt und ermittle dann die Lösung.

Die Vertauschung und die Linearkombination von Gleichungen können wir durch *Multiplikation mit speziellen Matrizen* erreichen. Zu diesem Zweck führen wir folgende Bezeichnungen ein.

Definition 1.2 (Permutationsmatrix)**.**
Eine Matrix P, die in jeder Zeile und jeder Spalte genau eine Eins und sonst nur Nullen enthält heißt Permutationsmatrix*.*

Wir listen einige Permutationsmatrizen auf:

m=1: [1]

m=2: $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

m=3: $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

Da die Spalten einer Permutationsmatrix durch Permutation der Spalten der Einheitsmatrix entstehen, existieren genau $m!$ m-reihige Permutationsmatrizen.
Man macht sich nun leicht klar, dass

- Multiplikation einer Matrix von *links* mit einer Permutationsmatrix eine Permutation der *Zeilen* bewirkt,
- Multiplikation einer Matrix von *rechts* mit einer Permutationsmatrix eine Permutation der *Spalten* bewirkt.

Definition 1.3 (elementare untere Dreiecksmatrix)**.**
Eine Matrix des Typs

$$\Lambda_i = \begin{bmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & 1 & & \\ & & \lambda_{i+1,i} & \ddots & \\ & & \vdots & & \ddots \\ 0 & & \lambda_{m,i} & 0 & 1 \end{bmatrix}_{m\times m}$$

die sich von der Einheitsmatrix nur in der i-ten Spalte unterhalb der Diagonalen unterscheidet, bezeichnen wir als elementare untere Dreiecksmatrix.

Elementare untere Dreiecksmatrizen lassen sich auf einfache Weise invertieren. Dies sollen Sie in der folgenden Aufgabe zeigen.

Aufgabe 1.4 (Die Inverse einer elementaren unteren Dreiecksmatrix)**.**
Man zeige: Für

$$\Lambda_i = \begin{bmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & 1 & & \\ & & \lambda_{i+1,i} & \ddots & \\ & & \vdots & & \ddots \\ 0 & & \lambda_{m,i} & 0 & 1 \end{bmatrix}_{m\times m}$$

gilt

$$\Lambda_i^{-1} = \begin{bmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & 1 & & \\ & & -\lambda_{i+1,i} & \ddots & \\ & & \vdots & & \ddots \\ 0 & & -\lambda_{m,i} & 0 & 1 \end{bmatrix}_{m\times m}.$$

Wir kehren nun zur Ausgangssituation zurück und betrachten das System

$$Ax = b,$$

das in Komponentenschreibweise die folgende Gestalt hat:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & \dots & a_{2m} \\ \vdots & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ a_{m1} & a_{m2} & \dots & \dots & a_{mm} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ \vdots \\ \vdots \\ b_m \end{bmatrix}$$

Dieses System formen wir in einem *rekursiven* Prozess um. Wir setzten dabei zunächst

$$A^{(1)} := A, \qquad b^{(1)} := b,$$

wobei A invertierbar sei. Dann existiert in der ersten Spalte von $A^{(1)}$ ein von Null verschiedenes Element $a_{k1}^{(1)}$, $1 \le k \le m$. Durch Vertauschung der ersten mit der k-ten Zeile erreichen wir, dass in der Position $(1,1)$ ein von Null verschiedenes Element steht. Diesen Vertauschungsprozess leistet die Linksmultiplikation mit der Permutationsmatrix

$$P_1 = \begin{bmatrix} 0 & \dots & \dots & 0 & 1 & & & & & \\ \vdots & 1 & & 0 & 0 & & & & & \\ \vdots & & \ddots & & \vdots & & \mathcal{O} & & & \\ 0 & 0 & & 1 & \vdots & & & & & \\ 1 & 0 & \dots & \dots & 0 & \dots & \dots & \dots & \dots & \leftarrow k. \\ & & & & \vdots & 1 & & & 0 & \\ & & \mathcal{O} & & \vdots & & \ddots & & & \\ & & & & \vdots & & & \ddots & & \\ & & & & \vdots & 0 & & & 1 & \end{bmatrix}$$

$$\uparrow$$
$$k$$

(Ist $k = 1$, so hat man $P_1 = E$ =Einheitsmatrix zu wählen.) ($\mathcal{O}$ steht hier und im Folgenden für die Nullmatrix.)

Für

$$\tilde{A}^{(1)} := P_1 A^{(1)}, \qquad \tilde{b}^{(1)} := P_1 b^{(1)}$$

gilt also

$$\tilde{a}_{11}^{(1)} \neq 0.$$

Jetzt adieren wir zur zweiten Zeile das $-\frac{\tilde{a}_{21}^{(1)}}{\tilde{a}_{11}^{(1)}}$-fache der ersten,..., allgemein zur k-ten Zeile das $-\frac{\tilde{a}_{k1}^{(1)}}{\tilde{a}_{11}^{(1)}}$-fache der ersten Zeile, $k = 2, ..., m$. Dann stehen in der ersten Spalte

unterhalb der Diagonalen nur Nullen:

$$\begin{bmatrix} \tilde{a}_{11}^{(1)} & \tilde{a}_{12}^{(1)} & \dots & \dots & \tilde{a}_{1m}^{(1)} \\ 0 & * & \dots & \dots & * \\ \vdots & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ 0 & * & \dots & \dots & * \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} \tilde{b}_1^{(1)} \\ * \\ \vdots \\ \vdots \\ * \end{bmatrix}.$$

Diese $(m-1)$-malige Linearkombination lässt sich aber gerade als Linksmultiplikation mit der elementaren unteren Dreicksmatrix

$$L_1 := \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ -l_{21} & 1 & & & 0 \\ \vdots & & \ddots & & \\ \vdots & & & \ddots & \\ -l_{m1} & 0 & & & 1 \end{bmatrix}$$

interpretieren, wobei

$$l_{k1} := \frac{\tilde{a}_{k1}^{(1)}}{\tilde{a}_{11}^{(1)}}, \qquad k = 2, ..., m$$

gesetzt wurde.
Aus

$$A^{(1)}x = b^{(1)}$$

erhalten wir also

$$L_1 P_1 A^{(1)} x = L_1 P_1 b^{(1)}$$

Wir setzen abkürzend

$$A^{(2)} := L_1 P_1 A^{(1)}, \qquad b^{(2)} := L_1 P_1 b^{(1)}.$$

Dann hat $A^{(2)}$ die gewünschten Nullen in den Positionen $(k, 1)$, $k = 2, ..., m$. Dabei ist das ursprüngliche Gleichungssystem $Ax = b$ äquivalent zu modifizierten System $A^{(2)}x = b^{(2)}$.

Auf $A^{(2)}$ wendent man nun einen entsprechenden *Eliminationsprozess* an, der die erste Spalte festlässt und die zweite Spalte unterhalb der Diagonalen annulliert. Rekursive Fortführung liefert in $m-1$ Schritten ein äquivalentes Gleichungssystem in *Dreiecks-*

form:

$$\underbrace{\begin{bmatrix} * & * & * & \dots & \dots & * \\ 0 & * & * & \dots & \dots & * \\ 0 & 0 & * & \dots & \dots & * \\ \vdots & & \ddots & \ddots & & \vdots \\ \vdots & 0 & & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & * \end{bmatrix}}_{A^{(m)}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_m \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} * \\ * \\ * \\ \vdots \\ \vdots \\ * \end{bmatrix}}_{b^{(m)}}.$$

Hieraus kann die Lösung x, wie am Ende dieses Paragraphen beschrieben, durch „Rücksubstitution“ berechnet werden.

Bemerkung 1.5 (Spaltenpivotisierung)**.**
Man beachte, dass die Regularität von A nicht ausreicht, in jedem Schritt das Nichtverschwinden des Diagonalelements zu sichern.

$$\textit{triviales Beispiel:} \qquad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

*Man muss also eventuell Gleichungen vertauschen, um ein nichtverschwindendes Diagonalelement (*Pivot*-Element, von engl.* pivot*: Türangel, Drehpunkt) zu erhalten. Es erweist sich als besonders günstg, wenn man möglichst betragsgroße Pivot-Elemente wählt, da dann bei der Division zur Bildung der $l_{\nu\mu}$ möglichst betragsgroße Nenner entstehen. Es empfiehlt sich deshalb in jedem Schritt eine* Spalten-Pivot-Suche *durchzuführen, die darin besteht, dass man in der Spalte, die gerade dem Eliminationsprozess unterworfen wird, ein betragsgrößtes Element unterhalb der Diagonalen (einschließlich des Diagonalelements) als Pivot-Element wählt.*

Wir fassen unser Vorgehen algorithmisch.

Algorithmus 1.6 (Gauß-Eliminationsverfahren mit Spaltenpivotisierung)**.**
Es sei

$$A^{(1)} = (a_{ik}^{(1)}) := A = (a_{ik})_{i,k=1,\dots,m},$$

A invertierbar,

$$b^{(1)} = (b_i^{(1)}) := b = (b_i)_{i=1,\dots m}.$$

Für $\nu = 1, .., m-1$ bilde man $A^{(\nu+1)} = (a_{ik}^{(\nu+1)})_{i,k=1,\dots m}$ auf folgende Weise aus $A^{(\nu)}$: Zunächst bestimmt man ein Pivot-Element $a_{\mu\nu}^{(\nu)}$, $\mu \in \{\nu, ..., m\}$, in der ν-ten Spalte gemäß

$$|a_{\mu\nu}^{(\nu)}| = \max_{\nu \le i \le m} |a_{i\nu}^{(\nu)}|.$$

Durch Vertauschung der ν-ten mit der μ-ten Gleichung des Systems

$$A^{(\nu)} x = b^{(\nu)}$$

erhält man das System

$$\tilde{A}^{(\nu)}x = \tilde{b}^{(\nu)}$$

wobei

$$\tilde{a}_{\nu\nu}^{(\nu)} \neq 0$$

ist.
Anschließend werden die Elemente der ν-ten Spalte unterhalb der Diagonalen mit Hilfe der folgenden Transformationsformel eliminiert:
Für $i = \nu+1, \nu+2, ..., m$ setze:

$$\begin{aligned} l_{i\nu} &:= \frac{\tilde{a}_{i\nu}^{(\nu)}}{\tilde{a}_{\nu\nu}^{(\nu)}}, \\ a_{i\nu}^{(\nu+1)} &:= 0, \\ a_{ik}^{(\nu+1)} &:= \tilde{a}_{ik}^{(\nu)} - l_{i\nu}\tilde{a}_{\nu k}^{(\nu)}, \qquad k = \nu+1, ..., m, \\ b_i^{(\nu+1)} &:= \tilde{b}_i^{(\nu)} - l_{i\nu}\tilde{b}_\nu^{(\nu)}, \end{aligned}$$

Sonst setze man

$$a_{ik}^{(\nu+1)} := \tilde{a}_{ik}^{(\nu)}, \qquad b_i^{(\nu+1)} := \tilde{b}_i^{(\nu)}$$

Die Vertauschung leistet eine Linksmultiplikation mit der Permutationsmatrix

$$P_\nu = \begin{bmatrix} 1 & & & & & & & & 0 & \\ & \ddots & & & & & & & & \\ & & 1 & & & & & & & \\ & & & 0 & \dots & \dots & \dots & 1 & & & \leftarrow \nu \\ & & & \vdots & 1 & & & \vdots & & & \\ & & & \vdots & & \ddots & & \vdots & & & \\ & & & \vdots & & & 1 & \vdots & & & \\ & & & 1 & \dots & \dots & \dots & 0 & & & \leftarrow \mu \\ & & & & & & & & 1 & & \\ & & & & & & & & & \ddots & \\ 0 & & & & & & & & & & 1 \end{bmatrix}$$

$$\uparrow \qquad\qquad \uparrow$$
$$\nu \qquad\qquad \mu$$

$$\tilde{A}^{(\nu)} := P_\nu A^{(\nu)}, \qquad \tilde{b}_i^{(\nu)} := Pb^{(\nu)}.$$

Der Übergang von $\tilde{A}^{(\nu)}$ nach $A^{(\nu+1)}$ lässt sich durch Linksmultiplikation mit der elementaren unteren Dreiecksmatrix

$$L_\nu := \begin{bmatrix} 1 & & & & & 0 \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{\nu+1,\nu} & \ddots & & \\ & & \vdots & & \ddots & \\ 0 & & -l_{m,\nu} & & 0 & 1 \end{bmatrix}$$

deuten, wobei

$$l_{i\nu} := \frac{\tilde{a}_{i\nu}^{(\nu)}}{\tilde{a}_{\nu\nu}^{(\nu)}}, \qquad i = \nu+1, ..., m,$$

gesetzt wurde.
Die Matrix

$$A^{(m)} = L_{m-1}P_{m-1}L_{m-2}P_{m-2}\cdots L_1P_1A$$

ist eine obere Dreiecksmatrix:

$$\underbrace{\begin{bmatrix} * & * & * & \dots & \dots & * \\ 0 & * & * & \dots & \dots & * \\ 0 & \ddots & * & \dots & \dots & * \\ \vdots & & \ddots & \ddots & & \vdots \\ \vdots & 0 & & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & * \end{bmatrix}}_{A^{(m)}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_m \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} * \\ * \\ * \\ \vdots \\ \vdots \\ * \end{bmatrix}}_{b^{(m)}}.$$

Resultat 1.7 (LR-Zerlegung)**.**
Das Gaußsche Eliminationsverfahren liefert die sogenannte „LR-Zerlegung“

$$PA = LR$$

wobie P eine Permutationsmatrix, L ein untere und R eine obere Dreiecksmatrix ist, die sich wie folgt ergeben:

$$\begin{aligned} A^{(m)} &= L_{m-1}P_{m-1}L_{m-2}P_{m-1}\cdots L_1P_1A \\ &= L_{m-1}(P_{m-1}L_{m-2}P_{m-1})(P_{m-1}P_{m-2}L_{m-3}P_{m-2}P_{m-1})\cdots \\ &\quad \cdots(P_{m-1}P_{m-2}\cdots P_2L_1P_2\cdots P_{m-2}P_{m-1})\cdot \underbrace{(P_{m-1}P_{m-2}\cdots P_2P_1)}_{=:P}A \\ &= L_{m-1}\tilde{L}_{m-2}\cdots\tilde{L}_1PA \end{aligned}$$

mit elementaren Dreiecksmatrizen L_{m-1}, $\tilde{L}_{m-2}, \dots, \tilde{L}_1$. D.h.

$$PA = \underbrace{\tilde{L}_1^{-1}\cdots\tilde{L}_{m-2}^{-1}L_{m-1}^{-1}}_{=:L}\underbrace{A^{(m)}}_{:=R}$$

Aufgabe 1.8. *Man löse das lineare Gleichungssystem*

$$\begin{bmatrix} 2 & 3 & -1 & 0 \\ -6 & -5 & 0 & 2 \\ 2 & -5 & 6 & -6 \\ 4 & 6 & 2 & -3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 20 \\ -45 \\ -3 \\ 58 \end{bmatrix}$$

und bestimme explizit die Permutationsmatrix P_ν sowie die elementaren unteren Dreiecksmatrizen L_ν,$\nu = 1, ..., m-1$, die bei der Gauß-Elimination mit spaltenweiser Pivot-Suche auftreten.

Bemerkung 1.9 (diagonale Pivot-Wahl)**.**
*Falls keine Vertauschungen von Gleichungen notwendig sind (*diagonale Pivot-Wahl*), hat man die Darstellung*

$$A^{(m)} = L_{m-1} \cdots L_1 A$$

oder

$$A = L_1^{-1} \cdots L_{m-1}^{-1} A^{(m)}.$$

In Aufgabe 1.4 hatten wir gezeigt, dass $L_\nu^{-1}, \nu = 1, .., m-1$, ebenfalls eine elemtare untere Dreiecksmatrix ist, die sich von L_ν nur im Vorzeichen in der ν-ten Spalte unterhalb der Diagonalen unterscheidet. Man kann leicht zeigen, dass das Produkt von elemtaren unteren Dreiecksmatrizen eine untere Dreiecksmatrix ist, die in der Diagonalen nur Einsen enthält. Wir folgern dies aus

Aufgabe 1.10.
Die untere Dreiecksmatrizen mit normierter Diagonale

$$\Delta := \begin{bmatrix} 1 & & & 0 \\ * & \ddots & & \\ \vdots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots \\ * & \dots & \dots & * \quad 1 \end{bmatrix}_{m \times m}$$

bilden bezüglich der Multiplikation eine Gruppe.

Die Matrix

$$A = L_1^{-1} \cdots L_{m-1}^{-1} A^{(m)}$$

ist somit zerlegt in das Produkt einer *unteren Dreiecksmatrix mit normierter Diagonale*

$$\begin{aligned} L := L_1^{-1} \cdots L_{m-1}^{-1} &= L_1^{-1} + L_2^{-1} + ... + L_{m-1}^{-1} - (m-2)E \\ &= -(L_1 + L_2 + ... + L_{m-1}) + mE \end{aligned}$$

und einer *oberen Dreiecksmatrix*

$$R := A^{(m)}, \qquad \text{also:}$$

$$\underbrace{\begin{bmatrix} * & \dots & \dots & * \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ * & \dots & \dots & * \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & & & 0 \\ * & \ddots & & \\ \vdots & \ddots & \ddots & \\ * & \dots & * & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} * & \dots & \dots & * \\ & \ddots & & \vdots \\ & & \ddots & \vdots \\ 0 & & & * \end{bmatrix}}_{R}$$

Resultat 1.11 (LR-Zerlegung bei diagonaler Pivot-Wahl)**.**
Ist diagonale Pivot-Wahl möglich, so liefert das Gaußsche Eliminationsverfahren die „LR-Zerlegung“

$$A = LR$$

der Matrix A. Hierbei ergibt sich L aus den im Verlauf des Verfahrens verwendeten elementaren unteren Dreiecksmatrizen gemäß

$$L = mE - (L_1 + L_2 + ... + L_{m-1}).$$

Wir haben gesehen, dass Algorithmus 1.6 ein reguläres Gleichungssystem

$$Ax = b$$

in ein äquivalentes Gleichungssystem

$$A^{(m)}x = b^{(m)}$$

mit oberer Dreiecksmatrix $A^{(m)}$ transformiert. Ein solches System löst man durch *Rücksubstitution*:

Algorithmus 1.12 (Rücksubstitution)**.**
Es gelte

$$A^{(m)}x = b^{(m)}$$

mit regulärer oberer Dreiecksmatrix $A^{(m)} = (a_{ik}^{(m)})_{m\times m}, b^{(m)} = (b_i^{(m)})$*. Dann erhält man die Komponenten* $x_i, i = 1, .., m$*, von x durch Rücksubstitution:*
Iteration: *für* $i = m, m-1, ..., 1$ *berechne*

$$x_i = \frac{1}{a_{ii}^{(m)}} \left\{ b_i^{(m)} - \sum_{k=i+1}^{m} a_{ik}^{(m)} x_k \right\}.$$

Zusammenfassung:

In diesem Abschnitt wurde Ihnen das von der Schule her bekannte Eliminationsverfahren in systematischer, für die Programmierung geeigneter Form vorgestellt. Hierbei wurde die Notwendigkeit der Spaltenpivotisierung betont. Damit lässt sich ein Gleichungssytem mit regulärer Matrix in ein äquivalentes System umformen, das durch Rücksubstitution einfach gelöst werden kann.

Bemerkung 1.13 (gestaffeltes Gleichungssystem - Informationselement)**.**
Wenn man ein lineares Gleichungssystem auflösen will, wird man in der Regel direkt das Eliminationsverfahren aus Abschnitt 1.2 anwenden und nicht explizit die LR-Zerlegung aufstellen.

Sind dagegen mehrere Gleichungssysteme hintereinander mit derselben Koeffizientenmatrix A zu lösen, so empfiehlt es sich, die LR-Zerlegung von A zu ermitteln und dann jeweils das gestaffelte System

$$\begin{aligned} Ly &= b, \\ Rx &= y \end{aligned}$$

zu lösen. Dabei bestimmt man y durch das sogenannte Vorwärtseinsetzen *(erst y_1 berechnen, dann y_2 usw.) und danach x durch* Rücksubstitution *(erst x_m berechnen, dann x_{m-1} usw.).*
Die LR-Zerlegung spielt ebenfalls eine wichtige Rolle bei der Berechnung von Eigenwerten nach dem LR-Verfahren. darauf werden wir später noch eingehen.

Vorteile bietet die LR-Zerlegung auch bei Matrizen speziellen Typs, inbesondere bei Bandmatrizen.

Bezeichnung 1.14 (Band-Matrix der Bandbreite d - Tridiagonalmatrix)**.**
Eine Matrix des Typs

$$A = \begin{bmatrix} * & * & \dots & * & & 0 \\ * & \ddots & \ddots & & \ddots & \\ \vdots & \ddots & \ddots & \ddots & & * \\ * & & \ddots & \ddots & \ddots & \vdots \\ & \ddots & & \ddots & \ddots & * \\ 0 & & * & \dots & * & * \end{bmatrix},$$

wobei $a_{ik} = 0$ für $|i-k| \geq d$ gilt, bezeichnet man als Bandmatrix der Bandbreite d. *Spezielle Bandmatrizen sind die* Diagonalmatrizen *mit $d = 1$ und die* Tridiagonalmatrizen *mit $d = 2$.*

Bei der *LR*-Zerlegung ohne Pivotsuche bleibt die Bandstruktur erhalten. Dies zeigt man in:

Aufgabe 1.15.
Für eine Bandmatrix A der Bandbreite d haben die zugehörigen Matrizen L und R die Bandbreite d.

Zusammenfassung:
In diesem Abschnitt wurde festgestellt, dass bei diagonaler Pivot-Wahl die Bandbreite einer Matrix bei der *LR*-Zerlegung erhalten bleibt.

1.3 Die Cholesky-Zerlegung

Beim Gaußschen Eliminationsverfahren muss man i.A. Zeilen- (bzw. Spalten-) vertauschungen vornehmen, um das Nichtverschwinden des Diagonalelementes zu sichern. *Diagonale Pivot-Wahl* ist nur bei speziellen Matrizen möglich. Wir zeigen dies in diesem Abschnitt für *positiv definite, Hermitesche* Matrizen. Dazu erinnern wir uns an folgende Definition aus der Lineare Algebra.

Definition 1.16 (Hermitesche und positiv definite Matrizen)**.**
Eine Matrix $A = (a_{ik})_{m\times m}$ heißt

- Hermitesch*, falls* $A^H = A$
- positiv definit*, falls* $x^H Ax > 0$ *für alle* $0 \neq x \in \mathbb{C}^m$

Dabei bezeichnet $x \mapsto x^H$ bzw. $A \mapsto A^H$ den Übergang zu konjugiert-transponierten Vektoren bzw. Matrizen.

Satz 1.17.
Die Diagonalelemente einer positiv definiten Matrix sind sämtlich positiv.

Beweis:
Es sei $e_i = (0, ..., 0, 1, 0, ..., 0)^t$, $i = 1, .., m$, der i-te Einheitsvektor. Dann gilt wegen Definition 1.16

$$0 < e_i^t A e_i = a_{ii}.$$

□

Eine weitere wichtige Eigeschaft betrifft die Eigenwerte. Dies ist Inhalt der folgenden Aufgabe.

Aufgabe 1.18.
Die Eigenwerte einer positiv definiten Matrix A sind positiv.

Bei einer positiv definiten Matrix A können wir versuchen, den Gauß-Algorithmus *symmetrisch* durchzuführen. Dies ergibt dann das so genannte *Cholesky-Verfahren.*
Wir gehen aus von

$$A^{(1)} := A$$

und beachten, dass $a_{11} > 0$ und $A^{(1)}$ Hermitesch ist. Wir eliminieren nun sowohl die Nichtdiagonalglieder in der ersten Zeile als auch in der ersten Spalte, indem wir

$$A^{(2)} := L_1 A^{(1)} L_1^H$$

mit

$$L_1 := \begin{bmatrix} 1 & 0 & \dots & \dots & \dots & 0 \\ -\frac{a_{21}^{(1)}}{a_{11}^{(1)}} & 1 & \ddots & & & \vdots \\ \vdots & 0 & \ddots & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & & \ddots & 1 & 0 \\ -\frac{a_{m1}^{(1)}}{a_{11}^{(1)}} & 0 & \dots & \dots & 0 & 1 \end{bmatrix}$$

bilden. $A^{(2)}$ hat dann die Form

$$A^{(2)} = \begin{bmatrix} a_{11}^{(1)} & 0 & \dots & \dots & 0 \\ 0 & & \dots & \dots & \dots \\ \vdots & \vdots & & & \\ \vdots & \vdots & & B^{(2)} & \\ 0 & \vdots & & & \end{bmatrix}, \qquad a_{11}^{(1)} = a_{11}.$$

Wegen

$$(A^{(2)})^H = L_1 (A^{(1)})^H L_1^H = L_1 A^{(1)} L_1^H = A^{(2)}$$

ist $A^{(2)}$ und damit auch $B^{(2)}$ Hermitesch. $A^{(2)}$ ist sogar positiv definit: Denn für jeden Vektor $0 \neq x \in \mathbb{C}^m$ gilt wegen der Regularität von L_1^H

$$y := L_1^H x \neq 0;$$

folglich ergibt sich

$$\begin{aligned} x^H A^{(2)} x &= x^H L_1 A^{(1)} L_1^H x \\ &= (L_1^H x)^H A^{(1)} (L_1^H x) \\ &= y^H A^{(1)} y > 0 \end{aligned}$$

wegen der positiven Definitheit von $A^{(1)}$. Mit $A^{(2)}$ ist aber offensichtlich auch $B^{(2)}$ positiv definit. Wegen Satz 1.17 gilt dann

$$a_{22}^{(2)} > 0,$$

und die symmetrische Dreieckzerlegung lässt sich in der beschriebenen Weise auf $B^{(2)}$ statt $A^{(1)}$ anwenden. Nach $m-1$ Schritten erhält man die Zerlegung

$$L_{m-1} \cdots L_1 A L_1^H \cdots L_{m-1}^H = \underbrace{\begin{bmatrix} a_{11}^{(1)} & & & & 0 \\ & a_{22}^{(2)} & & & \\ & & \ddots & & \\ & & & \ddots & \\ 0 & & & & a_{mm}^{(m)} \end{bmatrix}}_{=:\mathrm{diag}(a_{11}^{(1)},\dots,a_{mm}^{(m)})}.$$

Wir setzten

$$D := \mathrm{diag}(a_{11}^{(1)}, .., a_{mm}^{(m)})$$

und

$$\sqrt{D} := \mathrm{diag}(\sqrt{a_{11}^{(1)}}, ..., \sqrt{a_{mm}^{(m)}}).$$

Man beachte, dass $\sqrt{D}$ wegen $a_{ii}^{(i)} > 0$, $i = 1, ..., m$, wohldefiniert ist.
Dann gilt

$$D = \sqrt{D}\sqrt{D}$$

Insgesamt haben wir die Darstellung

$$A = L_1^{-1}...L_{m-1}^{-1}\sqrt{D}\sqrt{D}(L_{m-1}^H)^{-1}...(L_1^H)^{-1}$$

gefunden. Wir setzen

$$\begin{aligned} L &:= L_1^{-1}...L_{m-1}^{-1}\sqrt{D}, \\ L^H &= \sqrt{D}(L_{m-1}^{-1})^{-1}...(L_1^H)^{-1} \end{aligned}$$

Es resultiert dann die Cholesky-Zerlegung

$$A = LL^H;$$

dabei ist L eine untere Dreiecksmatrix. Man beachte, dass die Diagonalelemente von L jetzt (im Gegensatz zur LR-Zerlegung beim Gaußschen Eliminationsverfahren) nicht mehr notwendig den Wert Eins haben.

Wir fassen zusammen:

Satz 1.19 (Cholesky-Zerlegung einer positiv definiten, Hermiteschen Matrix)**.**
Eine positiv definite, Hermitesche Matrix A lässt sich zerlegen in das Produkt einer unteren Dreiecksmatrix L mit ihrer konjugiert-transponierten:

$$A = LL^H.$$

Wir verwenden zur expliziten Bestimmung der Elemente l_{ik}, $i = 1, ..., m$, $k = 1, ..., i$, eine geeignete Reihenfolge. Hier geht man am einfachsten *zeilenweise* vor.
Aus

$$A = LL^H$$

erhält man

$$a_{ik} = \sum_{\mu=1}^{m} l_{i\mu}\overline{l_{k\mu}}, \qquad i, k = 1, ..., m,$$

woraus wegen der Dreiecksgestalt von L die Darstellung

$$a_{ik} = \sum_{\mu=1}^{min(i,k)} l_{i\mu}\overline{l_{k\mu}}, \qquad i, k = 1, .., m,$$

folgt. Insgesamt ergibt sich so:

Algorithmus 1.20 (Cholesky-Verfahren)**.**
Für $i = 1, ..., m$:

1. *Für $k = 1, ..., i-1$ berechne*

$$l_{ik} := \frac{1}{l_{kk}}\left\{a_{ik} - \sum_{\mu=1}^{k-1} l_{i\mu}\bar{l}_{k\mu}\right\}.$$

2. *Berechne*

$$l_{ii} := \sqrt{a_{ii} - \sum_{\mu=1}^{i-1} |l_{i\mu}|^2}.$$

Der Rechenaufwand beim Cholesky-Verfahren ist wesentlich niedriger als beim Gauß-Algorithmus. Wir lösen in diesem Zusammenhang folgende Aufgabe.

Aufgabe 1.21.
Man bestimme den Rechenaufwand, der bei der Cholesky-Zerlegung einer positiv definiten $(m \times m)$-Matrix erforderlich ist

Zusammenfassung:
In diesem Abschnitt haben Sie eine symmetrische Variante des Gaußschen Eliminationsverfahrens für positiv definite Matrizen kennengelernt. Dieses sogenannte Cholesky-Verfahren ist insbesondere auch numerisch weniger aufwendig.

1.4 Die QR-Zerlegung (Ergänzung)

Der Gauß-Algorithmus liefert mit Hilfe elementarer Dreiecksmatrizen die LR-Zerlegung einer regulären Matrix; dabei wird ein gegebenes System $Ax = b$ auf (obere) Dreiecksgestalt gebracht. Wir werden bei der Fehleranalyse des Gauß-Algorithmus feststellen, dass die LR-Zerlegung u.U. ein ungünstiges Fehlerverhalten zeigen kann. In gewissen Fällen kann es daher angebracht sein, die Reduktion auf Dreiecksgestalt mit Hilfe unitärer (oder orthogonaler) Matrizen durchzuführen. Zu diesem Zweck erinnern wir uns an das *Orthogonalisierungsverfahren von E. Schmidt*, das Ihnen in der Lineare Algebra vorgestellt wurde.

Aufgabe 1.22 (Orthogonalisierung nach E. Schmidt)**.**
Es seien $a_1, .., a_m$ m linear unabhängige Vektoren aus $\mathbb{C}^m$ (oder aus $\mathbb{R}^m$). Man konstruiere dazu rekursiv eine Orthonormalbasis $q_1, ..., q_m$, d.h. für $q_1, ..., q_m$ gelte

$$\begin{aligned} (q_\nu, q_\mu) &= \delta_{\nu\mu}, \qquad \nu, \mu = 1, ..., m \qquad \textit{und} \\ a_\nu &\in \operatorname{span}(q_1, ..., q_\nu), \qquad \nu = 1, ..., m, \end{aligned}$$

wenn $(.,.)$ das kanonische Skalarprodukt in $\mathbb{C}^m$ (oder in $\mathbb{R}^m$) bezeichnet.

Die in dieser Aufgabe hergeleiteten Formeln, welche die Vektoren a_ν mit den q_ν, $\nu = 1, ..., m$, verbinden, sind vom Typ

$$\begin{aligned} a_1 &= r_{11} q_1, \\ a_2 &= r_{12} q_1 + r_{22} q_2, \\ &\vdots \\ a_m &= r_{1m} q_1 + r_{2m} q_2 + ... + r_{mm} q_m \end{aligned}$$

(Beachten Sie, dass die sich hier ergebende Dreiecksform vom rekursiven Aufbau des Schmidtschen Orthogonalisierungsverfahren herrührt.) Wenn wir jetzt die Vektoren $a_1, ..., a_m$ und $q_1, ..., q_m$ jeweils als Spaltenvektoren einer Matrix A und einer Matrix Q auffassen, können wir die Orthogonalisierungsformeln in der Form

$$(a_1, ..., a_m) = (q_1, ..., q_m) \begin{bmatrix} r_{11} & \dots & r_{1m} \\ & \ddots & \vdots \\ 0 & & r_{mm} \end{bmatrix}$$

schreiben. Komponentenweise mit

$$\begin{aligned} a_\nu &=: (a_{1\nu}, a_{2\nu}, ..., a_{m\nu})^t, \qquad \nu = 1, ..., m \\ q_\nu &=: (q_{1\nu}, q_{2\nu}, ..., q_{m\nu})^t, \qquad \nu = 1, ..., m \end{aligned}$$

gilt dann

$$A := (a_{\mu\nu})_{\substack{\mu=1,...,m \\ \nu=1,...,m}} = \underbrace{\begin{bmatrix} q_{11} & \dots & q_{1m} \\ \vdots & & \vdots \\ q_{m1} & \dots & q_{mm} \end{bmatrix}}_{:=Q} \underbrace{\begin{bmatrix} r_{11} & \dots & r_{1m} \\ & \ddots & \vdots \\ 0 & & r_{mm} \end{bmatrix}}_{:=R}$$

Resultat 1.23 (QR-Zerlegung von A)**.**
Das Orthogonalisierungsverfahren nach E. Schmidt liefert die Darstellung

$$A = QR \qquad (\text{„}QR\text{-Zerlegung“});$$

dabei ist Q eine unitäre Matrix, d.h.

$$Q^H Q = E, \qquad Q^{-1} = Q^H,$$

und R eine obere Dreiecksmatrix.

Mit Hilfe der QR-Zerlegung ist die Auflösung eines linearen Gleichungssystems

$$Ax = b$$

auf die Auflösung eines „gestaffelten“ Systems

$$Rx = Q^{-1}b = Q^H b$$

zurückgeführt, die sich rekursiv durch Rücksubstitution wie bei

$$\begin{aligned} Ly &= b, \\ Rx &= y \end{aligned}$$

ergibt.

Bemerkung 1.24.
In der Praxis stößt man beim Orthogonalisierungsverfahren nach E. Schmidt auf Schwierigkeiten, falls die Ausgangsvektoren a_i „fast“ linear abhängig sind. Hier können schon geringe Rundungsfehler eine starke Störung der Orthogonalität der berechneten Vektoren $q_1, ..., q_m$ verursachen.

Günstiger lässt sich die QR-Zerlegung mit Hilfe von geeigneter Spiegelung, den sogenannten Householder-Transformationen*, gewinnen.*

Hierzu betrachten wir einen Vektor $w \in \mathbb{C}^m$ der euklidischen Länge 1, d.h. es gelte $w^H w = 1$. Ist E die Einheitsmatrix der Dimension m, so ist die $(m \times m)$-Matrix

$$H_w := E - 2ww^H$$

wegen

$$(E - 2ww^H)^H = E^H - 2(ww^H)^H = E - 2ww^H$$

Hermitesch.

Definition 1.25 (elementare Hermitesche Matrix - Householder-Transformation)**.**
Eine Matrix des Typs

$$H_w := E - 2ww^H, \qquad w \in \mathbb{C}^m, \qquad w^H w = 1, \qquad E = m \times m\text{-Einheitsmatrix}$$

heißt elementare Hermitesche Matrix*; die durch H_w vermittelte Abbildung des Raumes $\mathbb{C}^m$ in sich bezeichnet man als* Householder-Transformation*.*

Wir stellen einige wichtige Eigenschaften elementarer Hermitescher Matrizen zusammen:

Satz 1.26.
Es sei H_w eine elementare Hermitesche $(m \times m)$-Matrix.
Dann ist H_w

1. *unitär: $H_w^{-1} = H_w^H$,*
2. *involutorisch: $(H_w)^2 = E$, und*
3. *die Abbildung $x \mapsto H_w x$ entspricht einer Spiegelung von x an der zu w orthogonalen Hyperebene L.*

Beweis:

1. Wegen
$$H_w^H H_w = H_w H_w = (E - 2ww^H)(E - 2ww^H) = E - 4ww^H + 4ww^H ww^H$$
folgt unter Berücksichtigung von $w^H w = 1$ die Beziehung $H_w^H \cdot H_w = E$; also gilt:
$$H_w^{-1} = H_w^H.$$

2. ist wegen (1) und wegen $H_w^H = H_w$ klar.

3. Einen Vektor $x \in \mathbb{C}^m$ projizieren wir auf die eindeutig bestimmte Hyperebene L, welche orthogonal zu w ist, vgl. Abbildung 1.2.
Dann lässt sich x in eindeutiger Weise in der Form
$$x = tw + v \qquad \text{mit} \qquad t \in \mathbb{C}, v \in L$$
darstellen. Spiegelung an L entspricht dem Übergang von
$$x \mapsto \tilde{x} := -tw + v.$$
Wegen
$$w^H x = tw^H w + w^H v = t$$
folgt
$$\tilde{x} = -w^H xw + v$$
und daraus mit
$$v = x - tw = x - w^H xw$$

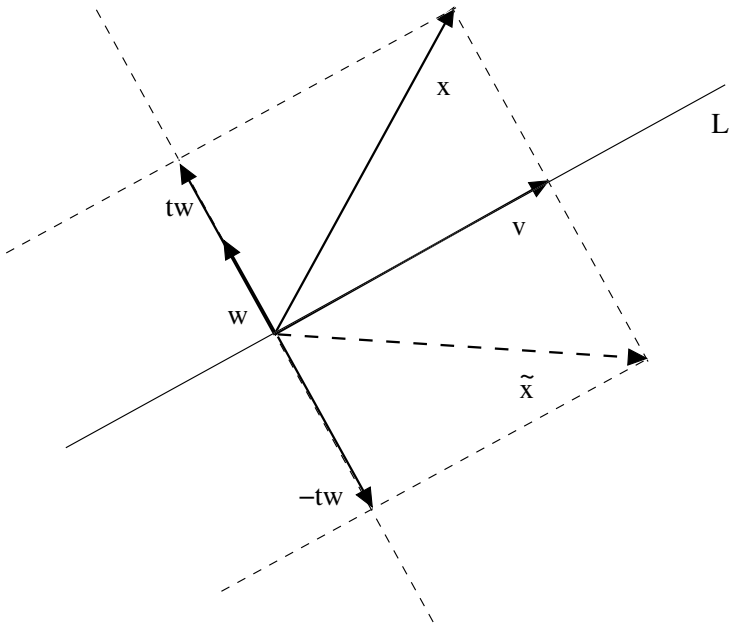


Abbildung 1.2: Spiegelung von x an der zu w orthogonalen Hyperebene L

die Darstellung

$$\begin{aligned}\tilde{x} &= -w^H x w + x - w^H x w \\ &= x - 2w^H x w \\ &= x - 2ww^H x \qquad \text{(beachte: } w^H x \in \mathbb{C} \text{ !)} \\ &= (E - 2ww^H)x \\ &= H_w x.\end{aligned}$$

□

Mit Hilfe von Householder-Transformationen werden wir ebenfalls die QR-Zerlegung konstruieren. Dazu benötigen wir den algorithmischen Zusammenhang, der die Spiegelung eines gegebenen Vektors x in ein skalares Vielfaches eines weiteren gegebenen Vektors y beschreibt; y wird ein kanonischer Einheitsvektor e_k, $1 \leq k \leq m$, sein. Wir machen uns zunächst anschaulich klar, dass es zwei Spiegelungen gibt, die das gewünschte leisten, vgl. Abbildung 1.3.

Nach diesen anschaulichen Überlegungen stellen wir nun den entsprechenden algorithmischen Zusammenhang auf.

Gegeben sei der Vektor $x \neq 0$. Wir suchen elementare Matrizen

$$H_w = E - 2ww^H, \qquad w^H w = 1,$$

welche

$$H_w x = \alpha e_k, \qquad k \in 1, ..., m \text{ fest}, \qquad 0 \neq \alpha \in \mathbb{C},$$

liefern; dabei bezeichnet

$$e_k = (0, ..., 0, 1, 0, ..., 0)^t$$

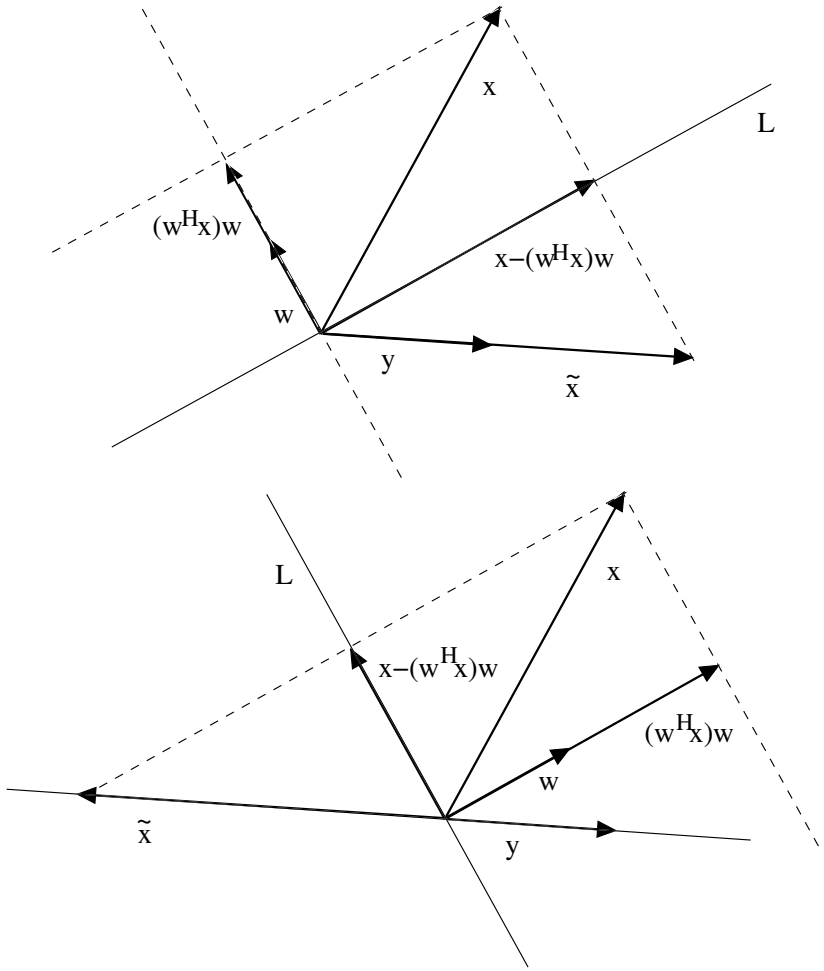


Abbildung 1.3: Zwei mögliche Spiegelungen

den k-ten Einheitsvektor.

Zunächst beachten wir, dass eine Spiegelung die euklidische Länge eines Vektors invariant lässt; also gilt

$$|\alpha| = \|x\|_e := \sqrt{x^H x} = \sqrt{\sum_{\nu=1}^{m} |x_\nu|^2} \neq 0,$$

wenn x die Komponenten $x_1, ..., x_m$ hat. Wir verwenden die Darstellung

$$x_k = |x_k| \cdot e^{i\varphi_k} \qquad (\ \varphi_k := 0 \text{ für } x_k = 0)$$

von x_k in Polarkoordinaten.
Wegen $x \neq 0$ können wir nun $w \in \mathbb{C}^m$ mit $w^H w = 1$ und $\alpha \in \mathbb{C}$ derart bestimmen, dass

$$\alpha e_k = H_w x$$

gilt. Denn aus

$$\alpha e_k = (E - 2ww^H)x = x - 2ww^H x$$

folgt

$$x - \alpha e_k = 2ww^H x.$$

Hieraus erhält man durch Multiplikation mit x^H von links

$$x^H x - \alpha x^H e_k = 2x^H w w^H x$$

oder

$$x^H x - \alpha \overline{x}_k = 2|w^H x|^2.$$

Diese Gleichung zeigt unter Beachtung der Tatsache, dass $x^H x$ reell ist, dass auch $\alpha \cdot \overline{x}_k$ reell sein muss. Ist $x_k \neq 0$, so kann α wegen

$$x_k = |x_k| \cdot e^{i\varphi_k} \qquad \text{und} \qquad \alpha \neq 0$$

nur die Darstellungen

$$\alpha_1 = |\alpha| \cdot e^{i\varphi_k} \qquad \text{oder} \qquad \alpha_2 = -|\alpha| \cdot e^{i\varphi_k}$$

besitzen. Beide Vorzeichen sind möglich; dies steht im Einklang mit Abbildung 1.3, wo wir die beiden möglichen Spiegelungen dargestellt haben. Ist $x_k = 0$, so kann das Argument von α zunächst beliebig gewählt werden; im Einklang mit der Festlegung $\varphi_k = 0$ beschränken wir uns auch hier auf die Fälle $\alpha_1 = |\alpha|$ oder $\alpha_2 = -|\alpha|$. Den Vektor w erhalten wir aus der Beziehung

$$x - \alpha_{1,2} e_k = 2ww^H x = 2(w^H x)w,$$

d.h. w is ein Vielfaches von $x - \alpha_{1,2} e_k$. Welches ist die bessere Wahl: α_1 oder α_2? Wegen

$$\|x - \alpha_{1,2} e_k\|_e^2 = |x_1|^2 + ... + |x_{k-1}|^2 + |x_k - \alpha_{1,2}|^2 + |x_{k+1}|^2 + ... + |x_m|^2$$

gilt

$$x - \alpha_2 e_k \neq 0 \qquad \text{und} \qquad \|x - \alpha_2 e_k\|_e > \|x - \alpha_1 e_k\|_e.$$

Man setzt deshalb

$$w := \frac{x - \alpha_2 e_k}{\|x - \alpha_2 e_k\|_e},$$

um den Rundungsfehler der Komponente w_k von w möglichst klein zu halten. Diese Wahl von w leistet das Gewünschte, denn $\alpha_2 = -\|x\|_e e^{i\varphi_k}$ und daher

$$\begin{aligned} H_w x &= x - 2w(w^H x) \\ &= x - 2\frac{\|x\|_e^2 - \overline{\alpha}_2 x_k}{\|x\|_e^2 - \alpha_2 \overline{x}_k - \overline{\alpha}_2 x_k + |\alpha_2|^2}(x - \alpha_2 e_k) \\ &= x - 2\frac{\|x\|_e^2 - \overline{\alpha}_2 x_k}{\|x\|_e^2 - \overline{\alpha}_2 x_k - \overline{\alpha}_2 x_k + |\alpha_2|^2}(x - \alpha_2 e_k) \\ &= \alpha_2 e_k. \end{aligned}$$

Wir stellen die Transformationsformeln für eine Householder-Spiegelung nochmals übersichtlich zusammen.

Algorithmus 1.27 (Spiegelung durch Householder-Transformation)**.**
Es sei $x = (x_1, ..., x_m)^t \neq 0$ *ein gegebener Vektor und* $k \in 1, ..., m$.
Ist $x_k \neq 0$*, so setze man*

$$x_k =: |x_k| e^{i\varphi}.$$

Ist $x_k = 0$, so setze man

$$\varphi := 0.$$

Man berechne

$$\begin{aligned} |\alpha| &:= \|x\|_e = \sqrt{\sum_{\nu=1}^{m} |x_\nu|^2}, \\ \alpha &:= -|\alpha| e^{i\varphi}, \\ \beta &:= \|x - \alpha e_k\|_e := [|x_1|^2 + ... + |x_{k-1}|^2 + (|x_k| + |\alpha|)^2 + |x_{k+1}|^2 + ... + |x_m|^2]^{1/2}, \\ w &:= \frac{1}{\beta}(x - \alpha e_k). \end{aligned}$$

Dann gilt

$$w^H w = 1,$$

und die durch die elementare Hermitesche Matrix

$$H_w := E - 2ww^H$$

gegebene Householder-Transformation spiegelt den Vektor x auf den Vektor $\alpha e_k \neq 0$,

$$H_w := x \mapsto \alpha e_k.$$

(Für reelles $x_k \neq 0$ berücksichtige man, dass $e^{i\varphi} = \text{sign}(x_k)$ gilt; für $x \in \mathbb{R}^m$ verläuft die Rechnung also im Reellen).

Diesen Algorithmus verwenden wir nun dazu, die QR-Zerlegung

$$A = QR$$

einer gegebenen regulären Matrix A zu erzeugen. Wir gehen rekursiv vor und spiegeln im ersten Schritt den ersten Spaltenvektor a_1 von $A = (a_1, ..., a_m)$ auf ein skalares Vielfaches von e_1. Dies erreichen wir durch geeignete Wahl von $w_1 \in \mathbb{C}^m$ entsprechend dem Algorithmus 1.27. Dann hat

$$A^{(1)} := H_{w_1} A = (E - 2w_1 w_1^H)A, \qquad w_1^H w_1 = 1,$$

die Gestalt

$$A^{(1)} = \begin{bmatrix} * & a_{12}^{(1)} & * & \dots & \dots & * \\ 0 & a_{22}^{(1)} & * & \dots & \dots & * \\ 0 & \vdots & \vdots & & & \vdots \\ \vdots & \vdots & \vdots & & & \vdots \\ \vdots & \vdots & \vdots & & & \vdots \\ 0 & a_{n2}^{(1)} & * & \dots & \dots & * \end{bmatrix}.$$

Im zweiten Schritt wird $w_2 \in \mathbb{C}^m$ so konstruiert, dass H_{w_2} in der zweiten Spalte unterhalb der Diagonalen Nullen erzeugt und die erste Spalte nicht verändert:

$$w_2 = \begin{pmatrix} 0 \\ \tilde{w}_2 \end{pmatrix}; \qquad \tilde{w}_2 \in \mathbb{C}^{m-1}$$

so, dass die Householder-Transformation $H_{\tilde{w}_2}$ in $\mathbb{C}^{m-1}$ liefert:

$$H_{\tilde{w}_2} \begin{pmatrix} a_{22}^{(1)} \\ \vdots \\ a_{n2}^{(1)} \end{pmatrix} = \beta \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{C}^{m-1}.$$

Dann

$$H_{w_2} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad H_{w_2} \begin{pmatrix} a_{12}^{(1)} \\ \vdots \\ \vdots \\ a_{k2}^{(1)} \end{pmatrix} = \begin{pmatrix} a_{12}^{(1)} \\ \beta \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

also

$$H_{w_2} A^{(1)} = \left(\begin{array}{cc|ccc} * & * & * & \cdots & * \\ 0 & * & \vdots & & \vdots \\ \vdots & 0 & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & * & \dots & * \end{array}\right) =: A^{(2)}$$

Im dritten Schritt gehen wir wie folgt vor:

$$w_3 = \begin{pmatrix} 0 \\ 0 \\ \tilde{w}_3 \end{pmatrix}, \qquad \tilde{w}_3 \in \mathbb{C}^{m-2},$$

wobei $\tilde{w}_3$ in $\mathbb{C}^{m-2}$ so gewählt wird, dass $H\tilde{w}_3$ den Vektor $\begin{pmatrix} a_{33}^{(2)} \\ \vdots \\ a_{n3}^{(2)} \end{pmatrix} \in \mathbb{C}^{m-2}$ auf $\gamma \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{C}^{m-2}$ abbildet, etc.. In dieser Weise fortfahrend erhält man in $m-1$ Schritten die Darstellung

$$A^{(m-1)} = H_{w_{m-1}} ... H_{w_1} A =: R = \begin{bmatrix} * & \dots & \dots & * \\ & \ddots & & \vdots \\ & & \ddots & \vdots \\ 0 & & & * \end{bmatrix}.$$

Dabei hat R obere Dreicksform und nichtverschwindende Diagonalelemente.

Aufgabe 1.28. *Man formuliere einen Algorithmus, der mit Hilfe von Householder-Transformationen die QR-Zerlegung einer reellen regulären Matrix liefert.*

Man beachte, dass die QR-Zerlegung für jede reguläre $(m \times m)$-Matrix existiert, während die LR-Zerlegung selbst bei regulären Maritzen i.A. nur unter Verwendung von Zeilen- (oder Spalten-) permutationen (Pivot-Suche) möglich ist.

Resultat 1.29 (QR-Zerlegung durch Householder-Transformation)**.**
Eine reguläre $(m \times m)$-Matrix A lässt sich mit Hilfe von $m-1$ Householder-Transformationen in der Form

$$A = QR \qquad mit \qquad Q^H Q = E, \qquad R = \text{ obere Dreiecksmatrix,}$$

darstellen. Das Gleichungssystem $Ax = b$ ist nun leicht auflösbar: $QRx = b \Leftrightarrow Rx = Q^H b$ da Q unitär (Auflösen durch Rückwärtssubstitution).

Zusatz: Es sei bemerkt, dass man auch *QR-Zerlegungen singulärer quadratischer Matrizen* bestimmen kann, wenn Spaltenpermutationen zugelassen werden. In diesem Fall wird zu Beginn des k-ten Schrittes in der oben (für reguläre Matrizen) beschriebenen QR-Zerlegung die Matrix

$$A^{(k)} = \begin{bmatrix} * & \cdots & \cdots & \cdots & * \\ & \ddots & & & \vdots \\ & & * & \cdots & * \\ 0 & & & B^{(k)} & \end{bmatrix}$$

von rechts mit einer $(m \times m)$-Permutationsmatrix $P^{(k)}$ multipliziert,

$$P^{(k)} = \begin{bmatrix} 1 & & & 0 \\ & \ddots & & \\ & & 1 & \\ 0 & & & P_1^{(k)} \end{bmatrix}$$

wobei die $((m-k+1) \times (m-k+1))$-Permutationsmatrix $P_1^{(k)}$ so gewählt wird, dass die Matrix $B^{(k)} \cdot P_1^{(k)}$ in der ersten Spalte eine nichtverschwindende Komponente besitzt. Auf $A^{(k)}P^{(k)}$ kann nun die im Algorithmus 1.27 beschriebene Householder-Transformation angewendet werden. Das Verfahren bricht nach genau $r := Rang(A)$ Schritten ab (wenn A singulär ist) und liefert

$$A^{(r)} = H_{w_r} \cdots H_{w_1} A P^{(1)} \cdots P^{(r)} = \begin{bmatrix} * & \cdots & \cdots & \cdots & \cdots & * \\ & \ddots & & & & \vdots \\ & & * & \cdots & \cdots & * \\ 0 & & & & \mathcal{O} & \end{bmatrix}$$

bzw.

$$AP^{(1)} \cdots P^{(r)} = H_{w_1} \cdots H_{w_r} A^{(r)} =: QR.$$

Kapitel 2

Eigenwertprobleme

2.1 Begriffe und Definitionen

Gegeben sei eine Matrix $A \in \mathbb{K}^{m\times m}$ ($\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$).

Definition 2.1 (Eigenwert, Eigenvektor)**.**
Ein Wert $\lambda \in \mathbb{C}$ heißt Eigenwert *von A, falls ein Vektor $u \in \mathbb{C}^m \setminus \{0\}$ existiert mit*

$$Au = \lambda u.$$

Jeder derartige Vektor $u \neq 0$ heißt ein zu λ gehöriger Eigenvektor*.*

Die Eigenwerte einer Matrix können berechnet werden als Nullstellen des charakteristischen Polynoms

$$p_A(\lambda) := \det(A - \lambda E)$$

der Matrix A. Dieses Vorgehen ist numerisch instabil und daher nicht zu empfehlen. Ziel der Untersuchungen dieses Kapitels ist es, numerisch praktikable Verfahren zu finden, mit denen man (unter geeigneten Voraussetzungen an die Matrix A) Näherungen an die Eigenwerte und Eigenvektoren von A berechnen kann.

Wir werden uns im Folgenden auf die Klasse der *diagonalisierbaren* Matrizen beschränken. Dabei heißt eine $m \times m$ Matrix A *diagonalisierbar*, falls es m linear unabhängige Eigenvektoren $u^1, \ldots, u^m$ der Matrix A gibt. Die Vektoren $u^1, \ldots, u^m$ bilden eine Basis des $\mathbb{C}^m$ aus Eigenvektoren. Der zu u^i gehörige Eigenwert sei mit λ_i bezeichnet, $i = 1, \ldots, m$. Setze nun

$$T := \left[\begin{array}{c|c|c|c} & & & \\ u^1 & u^2 & \ldots & u^m \\ & & & \end{array}\right].$$

Die $m \times m$-Matrix T ist regulär, da ihre Spalten $u^1, \ldots, u^m$ linear unabhängig sind. Wir berechnen

$$\begin{aligned} AT &= \left[\begin{array}{c|c|c|c} & & & \\ Au^1 & Au^2 & \ldots & Au^m \\ & & & \end{array}\right] = \left[\begin{array}{c|c|c|c} & & & \\ \lambda_1 u^1 & \lambda_2 u^2 & \ldots & \lambda_m u^m \\ & & & \end{array}\right] \\ &= \left[\begin{array}{c|c|c|c} & & & \\ u^1 & u^2 & \ldots & u^m \\ & & & \end{array}\right] \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix} = T \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix}, \end{aligned}$$

und erhalten somit für eine diagonalisierbare Matrix A die Darstellung

$$T^{-1}AT = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix} = \operatorname{diag}(\lambda_1, \ldots, \lambda_m)$$

bzw.

$$A = T \operatorname{diag}(\lambda_1, \ldots, \lambda_m) T^{-1}.$$

Unter jeder der folgenden vier Bedingungen ist die Matrix A diagonalisierbar:

(1) A besitzt m verschiedene Eigenwerte.

(2) A ist *normal*, d.h. $AA^H = A^H A$. In diesem Fall kann man die Eigenvektoren sogar so wählen, daß $u_1 \ldots, u_m$ eine Orthonormalbasis ist. Damit ist T eine unitäre Matrix, denn

$$T^H T = \left[\begin{array}{c} \overline{u^{1H}} \\ \hline \overline{u^{2H}} \\ \hline \vdots \\ \hline \overline{u^{mH}} \end{array}\right] \left[\begin{array}{c|c|c|c} & & & \\ u^1 & u^2 & \ldots & u^m \\ & & & \end{array}\right] = E, \quad \text{d.h.} \quad T^{-1} = T^H.$$

(3) A ist Hermitesch. Denn dann ist wegen $A^H = A$ die Matrix A insbesondere normal. In diesem Fall sind die Eigenwerte $\lambda_1, \ldots, \lambda_m$ alle reell.

(4) A ist unitär. Denn dann ist wegen $A^H A = E = AA^H$ die Matrix A insbesondere normal. In diesem Fall gilt für die Eigenwerte: $|\lambda_1| = \ldots = |\lambda_m| = 1$.

2.2 Die von-Mises Iteration

Ziel diese Abschnittes ist es, ein iteratives Verfahren zur Bestimmung eines Eigenwertes einer diagonalisierbaren Matrix zu formulieren. Es sei also $A \in \mathbb{C}^{m \times m}$ eine diagonalisierbare $m \times m$-Matrix. Wir ordnen die Eigenwerte gemäß der Größe ihres Betrages

$$|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_m|.$$

Im Folgenden betrachten wir den Spezialfall

$$|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_m|$$

und beschreiben ein Verfahren zur Bestimmung des betragsgrößten Eigenwertes λ_1. Sei also $u^1, \ldots, u^m$ mit $Au^i = \lambda_i u^i$, $i = 1, \ldots, m$ die Basis aus Eigenvektoren von A. Ein beliebiger Vektor $x^0 \in \mathbb{C}^m$ lässt sich in der Basis entwickeln:

$$x^0 = \rho_1 u^1 + \rho_2 u^2 + \ldots + \rho_m u^m \text{ mit } \rho_1, \ldots, \rho_m \in \mathbb{C}.$$

Wir nehmen an, dass $\rho_1 \neq 0$ gilt.

Definiere

$$x^k := Ax^{k-1} \text{ für } k = 1, 2, \ldots.$$

Dann gilt

$$\begin{aligned} x^k = Ax^{k-1} = A(Ax^{k-2}) = \ldots = \underbrace{A \cdot A \cdot \ldots \cdot A}_{=:A^k} x^0 \\ = A^k x^0. \end{aligned}$$

Es folgt

$$\begin{aligned} x^k = A^k x^0 &= A^k(\rho_1 u^1 + \rho_2 u^2 + \ldots + \rho_m u^m) \\ &= A^{k-1}(\rho_1 A u^1 + \rho_2 A u^2 + \ldots + \rho_m A u^m) \\ &= A^{k-1}(\rho_1 \lambda_1 u^1 + \rho_2 \lambda_2 u^2 + \ldots + \rho_m \lambda_m u^m) \\ &= \ldots \\ &= \rho_1 \lambda_1^k u^1 + \rho_2 \lambda_2^k u^2 + \ldots + \rho_m \lambda_m^k u^m. \end{aligned}$$

Folglich gilt:

$$\begin{aligned} \frac{x^k}{\lambda_1^k} &= \rho_1 u^1 + \rho_2 \underbrace{\left(\frac{\lambda_2}{\lambda_1}\right)^k}_{\to 0} u^2 + \ldots + \rho_m \underbrace{\left(\frac{\lambda_m}{\lambda_1}\right)^k}_{\to 0} u^m \\ &\to \rho_1 u^1 \text{ für } k \to \infty. \end{aligned}$$

Problem: λ_1 ist gesucht, d.h. im Allgemeinen nicht bekannt! Abhilfe wird durch die folgende Modifikation der obigen Iteration geschaffen.

Algorithmus 2.2 (Von-Mises Iteration)**.**
Sei A eine diagonalisierbare $m \times m$-Matrix; wähle Startvektor $x^0 \in \mathbb{C}^m$ wie zuvor.
Iteration: *Für $k = 1, 2, ...$ bilde die Vektoren*

$$\begin{aligned} z^k &:= Ax^{k-1} \\ x^k &:= \frac{z^k}{z^k_{i_k}}, \text{ wobei der Index } i_k \text{ so gewählt ist, dass gilt: } |z^k_{i_k}| = \max_{i=1}^{m} |z^k_i| \end{aligned}$$

Resultat 2.3 (Konvergenz der von-Mises Iteration)**.**
Für die von-Mises Iteration gilt:

- $z^{k+1}_{i_k} \to \lambda_1$ *für* $k \to \infty$
- x^k *nähert sich einem Vielfachen des Eigenvektors* u^1*. Genauer:*

$$x^k - \frac{u^1}{u^1_{i_k}} \to 0 \text{ für } k \to \infty.$$

Beweis: Wir skizzieren die Beweisschritte.

1. Es gilt folgende Darstellung für den k-ten iterierten Vektoren

$$\begin{aligned} x^k = \frac{z^k}{z^k_{i_k}} &= \frac{Ax^{k-1}}{(Ax^{k-1})_{i_k}} = \frac{Az^{k-1} \cdot \frac{1}{z^{k-1}_{i_{k-1}}}}{(Az^{k-1})_{i_k} \cdot \frac{1}{z^{k-1}_{i_{k-1}}}} \\ &= \frac{A^2x^{k-2}}{(A^2x^{k-2})_{i_k}} \\ &= \dots \\ &= \frac{A^kx^0}{(A^kx^0)_{i_k}} \end{aligned}$$

2. Nun betrachten wir $z^{k+1}_{i_k}$ und erhalten

$$\begin{aligned} z^{k+1}_{i_k} = (Ax^k)_{i_k} &= \frac{(A^{k+1}x^0)_{i_k}}{(A^kx^0)_{i_k}} \\ &= \frac{(\rho_1\lambda_1^{k+1}u^1 + \rho_2\lambda_2^{k+1}u^2 + \dots + \rho_m\lambda_m^{k+1}u^m)_{i_k}}{(\rho_1\lambda_1^ku^1 + \rho_2\lambda_2^ku^2 + \dots + \rho_m\lambda_m^ku^m)_{i_k}} \\ &= \lambda_1 \frac{\left(\rho_1u^1 + \rho_2\left(\frac{\lambda_2}{\lambda_1}\right)^{k+1}u^2 + \dots + \rho_m\left(\frac{\lambda_m}{\lambda_1}\right)^{k+1}u^m\right)_{i_k}}{\left(\rho_1u^1 + \rho_2\left(\frac{\lambda_2}{\lambda_1}\right)^{k}u^2 + \dots + \rho_m\left(\frac{\lambda_m}{\lambda_1}\right)^{k}u^m\right)_{i_k}} \\ &\to \lambda_1 \text{ für } k \to \infty, \end{aligned}$$

da die Terme $\left(\frac{\lambda_2}{\lambda_1}\right)^k, \dots, \left(\frac{\lambda_m}{\lambda_1}\right)^k$ für $k \to \infty$ gegen Null streben. Damit ist die erste Aussage des Resultates bewiesen.

3. Die zweite Aussage folgt auf ähnliche Art und Weise:

$$\begin{aligned} x^k - \frac{u^1}{u^1_{i_k}} &= \frac{A^kx^0}{(A^kx^0)_{i_k}} - \frac{u^1}{u^1_{i_k}} \\ &= \frac{\rho_1\lambda_1^ku^1 + \rho_2\lambda_2^ku^2 + \dots + \rho_m\lambda_m^ku^m}{(\rho_1\lambda_1^ku^1 + \rho_2\lambda_2^ku^2 + \dots + \rho_m\lambda_m^ku^m)_{i_k}} - \frac{u^1}{u^1_{i_k}} \\ &= \frac{\rho_1u^1 + \rho_2\left(\frac{\lambda_2}{\lambda_1}\right)^ku^2 + \dots + \rho_m\left(\frac{\lambda_m}{\lambda_1}\right)^ku^m}{\left(\rho_1u^1 + \rho_2\left(\frac{\lambda_2}{\lambda_1}\right)^ku^2 + \dots + \rho_m\left(\frac{\lambda_m}{\lambda_1}\right)^ku^m\right)_{i_k}} - \frac{u^1}{u^1_{i_k}} \\ &\to 0 \text{ für } k \to \infty, \end{aligned}$$

da wiederum die Terme $\left(\frac{\lambda_2}{\lambda_1}\right)^k, \dots, \left(\frac{\lambda_m}{\lambda_1}\right)^k$ für $k \to \infty$ gegen Null streben. Damit ist die Konvergenz der von-Mises Iteration geklärt. □

Bemerkung: Die Bedingung $\rho_1 \neq 0$ bei der Wahl des Startvektors x^0 bereitet keine Probleme, denn aufgrund von Rundungsfehlern ist dies spätestens ab dem zweiten Iterationsschritt gewährleistet.

2.3 Die inverse Iteration von Wielandt

Die folgende Variante der von-Mises Iteration liefert Approximationen des betragskleinsten Eigenwertes einer diagonalisierbaren und invertierbaren Matrix A. Wiederum seien $\lambda_1, \dots \lambda_m$ die Eigenwerte von A mit zugehöriger Basis aus Eigenvektoren u^i mit

$$Au^i = \lambda_i u^i \text{ für } i = 1, \dots, m,$$

wobei gelten möge:

$$|\lambda_1| \geq |\lambda_2| \geq \dots > |\lambda_m|.$$

Da A als invertierbar vorausgesetzt wurde, gilt

$$u^i = \lambda_i A^{-1} u^i \quad \text{bzw.} \quad A^{-1}u^i = \frac{1}{\lambda_i} u^i \text{ für } i = 1, \dots, m.$$

Das bedeutet, dass die Matrix A^{-1} die Eigenwerte $\frac{1}{\lambda_1}, \dots, \frac{1}{\lambda_m}$ mit zugehörigen Eigenvektoren $u^1, \dots, u^m$ hat. Außerdem gilt

$$\frac{1}{|\lambda_m|} > \frac{1}{|\lambda_{m-1}|} \geq \dots \geq \frac{1}{|\lambda_1|}.$$

Nun wenden wir die von-Mises Iteration auf die Matrix A^{-1} an, deren betragsgrößter Eigenwert gerade $1/\lambda_m$ ist. Das entstehende Verfahren heißt *inverse Iteration.*

Algorithmus 2.4 (Inverse Iteration)**.**
Sei A eine diagonalisierbare, reguläre $m \times m$-Matrix; wähle Startvektor $y^0 \in \mathbb{C}^m$.
Iteration: *Für $k = 1, 2, \dots$ bilde die Vektoren,*

$$w^k := A^{-1} y^{k-1}$$

$$y^k := \frac{w^k}{w^k_{i_k}}, \text{ wobei der Index } i_k \text{ so gewählt ist, dass gilt: } |w^k_{i_k}| = \max_{i=1}^{m} |w^k_i|$$

Resultat 2.5 (Konvergenz der inversen Iteration)**.**
Für die inverse Iteration gilt:

- $w^{k+1}_{i_k} \to \frac{1}{\lambda_m}$ *für* $k \to \infty$
- y^k *nähert sich einem Vielfachen des Eigenvektors u^m. Genauer:*

$$y^k - \frac{u^m}{u^m_{i_k}} \to 0 \text{ für } k \to \infty.$$

In der Praxis wird A^{-1} nicht berechnet. Anstelle dessen wird der Algorithmus der inversen Iteration wie folgt formuliert:

Algorithmus 2.6 (Inverse Iteration – Variante)**.**
Sei A eine diagonalisierbare, reguläre $m \times m$-Matrix; wähle Startvektor $y^0 \in \mathbb{C}^m$.
Iteration: *Für $k = 1, 2, \dots$ bestimme die folgenden Vektoren:*

löse das lineare Gleichungssystem $Aw^k = y^{k-1}$

$$y^k := \frac{w^k}{w^k_{i_k}}, \text{ wobei der Index } i_k \text{ so gewählt ist, dass gilt: } |w^k_{i_k}| = \max_{i=1}^{m} |w^k_i|$$

Für die Lösung des linearen Gleichungssystems kann man z.B. die LR-Zerlegung der Matrix A bestimmen, vgl. Kapitel 1.

Zusammenfassung:

Die $m \times m$-Matrix A sei diagonalisierbar. Die von-Mises Iteration liefert Approximationen an den betragsgrößten Eigenwert von A (und den zugehörigen Eigenvektor). Die inverse Iteration liefert (sofern A invertierbar ist) Approximationen an den Kehrwert des betragskleinsten Eigenwertes von A (und den zugehörigen Eigenvektor).

Praktischer Nutzen der inversen Iteration:

Angenommen, wir kennen bereits eine brauchbare Approximation $\tilde{\lambda}$ an einen Eigenwert λ_j, d.h.

$$|\tilde{\lambda} - \lambda_j| < |\tilde{\lambda} - \lambda_l| \text{ für alle } l \neq j.$$

Die bedeutet, dass $\lambda_j - \tilde{\lambda}$ der betragskleinste Eigenwert der Matrix $A - \tilde{\lambda}E$ ist. Nun können wir die inverse Iteration anwenden.

Wähle Startvektor $y^0 \in \mathbb{C}^m$.
Iteration: Für $k = 1, 2, ...$ bestimme die folgenden Vektoren:

löse das lineare Gleichungssystem $(A - \tilde{\lambda}E)w^k = y^{k-1}$

$$y^k := \frac{w^k}{w^k_{i_k}}, \text{ wobei der Index } i_k \text{ so gewählt ist, dass gilt: } |w^k_{i_k}| = \max_{i=1}^{m} |w^k_i|$$

Diese Iteration liefert das Ergebnis:

$$w^{k+1}_{i_k} \to \frac{1}{\lambda_j - \tilde{\lambda}} \text{ für } k \to \infty$$

bzw.

$$\tilde{\lambda} + \frac{1}{w^{k+1}_{i_k}} \to \lambda_j \text{ für } k \to \infty,$$

d.h. wir erhalten ein iteratives Verfahren, das verbesserte Approxmationen an λ_j liefert.

Bemerkung: Es liegt nahe, in jedem Iterationsschritt $\tilde{\lambda}$ durch die gerade neu gewonnene bessere Approximation an λ_j zu ersetzen. Dies führt zu folgendem Algorithmus:

Wähle Startvektor $y^0 \in \mathbb{C}^m$. Setze $\mu_0 := \tilde{\lambda} =: \mu_1$.
Iteration: Für $k = 1, 2, ...$ bestimme die folgenden Vektoren:

löse das lineare Gleichungssystem $(A - \mu_{k-1}E)w^k = y^{k-1}$

$$y^k := \frac{w^k}{w^k_{i_k}}, \text{ wobei der Index } i_k \text{ so gewählt ist, dass gilt: } |w^k_{i_k}| = \max_{i=1}^{m} |w^k_i|$$

$$\mu_k := \mu_{k-1} + \frac{1}{w^k_{i_{k-1}}} \text{ für } k \geq 2.$$

2.4 Das LR-Verfahren und das QR-Verfahren für Eigenwertprobleme (Ergänzung)

Wir bereits zuvor erwähnt, setzen wir voraus, daß die Matrix A diagonalisierbar ist. Von Rutishauser wurde 1958 das *LR-Verfahren* vorgeschlagen. Wie der Name schon andeutet, wird dabei an wesentlicher Stelle die LR-Zerlegung aus Abschnitt 1.2 benützt.

Wir beginnen mit einer Motivation für das weitere Vorgehen: Für die Matrix $A =: A_1$ existiere die LR-Zerlegung (ohne Zeilenvertauschungen)

$$A_1 = L_1 R_1,$$

wobei

$$L_1 = \begin{bmatrix} 1 & & & & 0 \\ * & \ddots & & & \\ \vdots & \ddots & \ddots & & \\ \vdots & & \ddots & \ddots & \\ * & \dots & \dots & * & 1 \end{bmatrix},$$

$$R_1 = \begin{bmatrix} * & \dots & \dots & \dots & * \\ & \ddots & & & \vdots \\ & & \ddots & & \vdots \\ & & & \ddots & \vdots \\ 0 & & & & * \end{bmatrix}$$

untere bzw. obere Dreiecksmatrizen sind. Nun wollen wir überlegen, wie die LR-Zerlegung der k-ten Potzenz A^k von A aussieht. Für die k-te Potenz A^k folgt zuerst

$$A^k = A_1^k = \underbrace{L_1 R_1 L_1 R_1 \cdots L_1 R_1 L_1 R_1}_{k-\text{mal der Faktor } L_1 R_1}.$$

Wir setzen

$$A_2 := R_1 L_1$$

und erhalten

$$A_1^k = L_1 \underbrace{A_2 \cdots A_2}_{k-1 \text{ Faktoren}} R_1 = L_1 A_2^{k-1} R_1.$$

Man erkennt, wie man durch Wiederholung dieses Prozesses eine Faktorisierung von A^k erhalten kann.

Resultat 2.7.
Setze

$$A_1 := A.$$

Wir nehmen an, daß die LR-Zerlegungen (ohne Zeilenvertauschungen)

$$A_\nu = L_\nu R_\nu, \qquad \nu = 1, ..., k,$$

für A_1 und für die Matrizen

$$A_\nu := R_{\nu-1} L_{\nu-1}, \qquad \nu = 2, ..., k$$

existieren. Dann gilt

$$A^k = L_1 L_2 \cdots L_k R_k R_{k-1} \cdots R_1.$$

Setzt man

$$\tilde{L}_k := L_1 L_2 \cdots L_k \text{ und } \tilde{R}_k := R_k R_{k-1} \cdots R_1,$$

so ist $\tilde{L}^k$ eine untere Dreiecksmatrix mit normierter Diagonale und $\tilde{R}_k$ eine obere Dreiecksmatrix. Damit haben wir die LR-Zerlegung

$$A^k := \tilde{L}_k \tilde{R}_k$$

von A^k bestimmt.

Wir betrachten nun eine diagonalisierbare $m \times m$-Matrix A, bei der λ_1 der dominante Eigenwert ist, d.h.:

$$|\lambda_1| > |\lambda_\nu|, \quad \nu = 2, ..., m.$$

Die zugehörige Basis aus Eigenvektoren sei $\{u_1, ..., u_m\}$. Wir starten nun ein Iterationsverfahren mit

$$z_0 := e_1 = (1, 0, ..., 0)^t$$

und setzen

$$z_k := A z_{k-1} = A^k z_0 = A^k e_1.$$

In der Basisdarstellung des Vektors e_1

$$e_1 = \rho_1 u_1 + \rho_2 u_2 + ... + \rho_m u_m$$

gelte die Annahme

$$\rho_1 \neq 0.$$

Dann folgt

$$\begin{aligned} z_k &= A^k(\rho_1 u_1 + \rho_2 u_2 + ... + \rho_m u_m) \\ &= \rho_1 \lambda_1^k u_1 + \rho_2 \lambda_2^k u_2 + ... + \rho_m \lambda_m^k u_m \\ &= \lambda_1^k \left\{ \rho_1 u_1 + \sum_{\mu=2}^{m} \rho_\mu \left[\frac{\lambda_\mu}{\lambda_1} \right]^k u_\mu \right\}, \end{aligned}$$

und daraus

$$\frac{z_k}{\lambda_1^k} \longrightarrow \rho_1 u_1 \qquad \text{für } k \to \infty.$$

Zum Vergleich berechnen wir erneut z_k; diesmal jedoch unter Verwendung der zuvor erörterten LR-Zerlegung von A^k

$$A^k = \tilde{L}_k \tilde{R}_k.$$

Unter Ausnützung der Dreiecksgestalt von $\tilde{L}_k$ und $\tilde{R}_k$ sowie der besonderen Form des kanonischen Einheitsvektors e_1 berechnet man

$$\begin{aligned} z_k = A^k e_1 &= \underbrace{\begin{bmatrix} 1 & & & 0 \\ \tilde{l}_{21}^{(k)} & \ddots & & \\ \vdots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots \\ \tilde{l}_{m1}^{(k)} & \dots & \dots & \tilde{l}_{mm-1}^{(k)} \quad 1 \end{bmatrix}}_{\tilde{L}_k} \underbrace{\begin{bmatrix} \tilde{r}_{11}^{(k)} & \dots & \dots & \dots & \tilde{r}_{1m}^{(k)} \\ & \ddots & & & \vdots \\ & & \ddots & & \vdots \\ & & & \ddots & \vdots \\ 0 & & & & \tilde{r}_{mm}^{(k)} \end{bmatrix}}_{\tilde{R}_k} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \\ &= \tilde{r}_{11}^{(k)} \left(1, \tilde{l}_{21}^{k}, ..., \tilde{l}_{m1}^{k}\right)^t. \end{aligned}$$

Wir erhalten also

$$\frac{z_k}{\lambda_1^k} = \frac{\tilde{r}_{11}^{(k)}}{\lambda_1^k} \underbrace{\begin{bmatrix} 1 \\ \tilde{l}_{21}^{(k)} \\ \vdots \\ \tilde{l}_{m1}^{(k)} \end{bmatrix}}_{=:\tilde{l}_1^{(k)}} \to \rho_1 \underbrace{\begin{bmatrix} \eta_{11} \\ \eta_{21} \\ \vdots \\ \eta_{m1} \end{bmatrix}}_{:=u_1 \neq 0} \qquad \text{für } k \to \infty$$

und insbesondere (bei Betrachtung der ersten Komponente):

$$\frac{\tilde{r}_{11}^{(k)}}{\lambda_1^k} \to \rho_1 \eta_{11} \qquad \text{für } k \to \infty.$$

Setzt man

$$R_k =: \left(r_{ij}^{(k)}\right)_{m \times m},$$

so erhält man aus

$$\tilde{R}_k = R_k \tilde{R}_{k-1}$$

die Beziehung

$$\tilde{r}_{11}^{(k)} = r_{11}^{(k)} \tilde{r}_{11}^{(k-1)}.$$

Unter der Annahme

$$\eta_{11} \neq 0$$

folgt dann aus der Konvergenz

$$\tilde{r}_{11}^{(k)} \lambda_1^{-k} \to \rho_1 \eta_{11} \qquad \text{für } k \to \infty,$$

dass das linke obere Element der oberen Dreiecksmatrizen R_k gegen den dominanten Eigenwert konvergiert:

$$r_{11}^{(k)} = \frac{\tilde{r}_{11}^{(k)}}{\tilde{r}_{11}^{(k-1)}} = \frac{\tilde{r}_{11}^{(k)} \lambda_1^{-k} \lambda_1}{\tilde{r}_{11}^{(k-1)} \lambda_1^{-(k-1)}} \to \frac{\rho_1 \eta_{11} \lambda_1}{\rho_1 \eta_{11}} = \lambda_1 \qquad \text{für } k \to \infty.$$

Weiter konvergiert die erste Spalte der unteren Dreiecksmatrizen $\tilde{L}_k$ wegen

$$\frac{z_k}{\lambda_1^k} = \frac{\tilde{r}_{11}^{(k)}}{\lambda_1^k} \left(1, \tilde{l}_{21}^{(k)}, ..., \tilde{l}_{m1}^{(k)}\right)^t \to \rho_1 \left(\eta_{11}, \eta_{21}, ..., \eta_{m1}\right)^t, \qquad k \to \infty,$$

gegen einen zu λ_1 gehörenden Eigenvektor:

$$\left(1, \tilde{l}_{21}^{(k)}, ..., \tilde{l}_{m1}^{(k)}\right)^t \to \frac{1}{\eta_{11}} (\eta_{11}, ..., \eta_{m1})^t = \frac{u_1}{\eta_{11}}, \qquad k \to \infty.$$

Diese Konvergenzaussagen, insbesondere die Aussage

$$r_{11}^{(k)} \to \lambda_1 \qquad \text{für } k \to \infty,$$

erhielten wir unter einschränkenden Bedingungen an λ_1 (dominanter Eigenwert) und u_1 ($\eta_{11} \neq 0$), sowie unter der Annahme, dass die LR-Zerlegung für jedes A_k existiert.

Damit wird die Vermutung nahegelegt, dass unter weiteren Bedingungen an A (vgl. Resultat 2.9) die Hauptdiagonalelemente von R_k gegen die Eigenwerte von A konvergieren und alle Spalten von $\tilde{L}_k$ - und damit die Matrix $\tilde{L}_k$ selbst - konvergieren. Da die L_k normierte untere Dreiecksmatrizen sind, folgt wegen

$$\tilde{L}_{k+1} = \tilde{L}_k L_k$$

aus der Konvergenz der $\tilde{L}_k$, dass die L_k gegen die Einheitsmatrix konvergieren und damit in

$$A_k = L_k R_k$$

die Elemente unterhalb der Hauptdiagonalen für $k \to \infty$ im Grenzwert verschwinden. Wir formulieren daher:

Algorithmus 2.8 (LR-Verfahren)**.**
Sei A eine reguläre $m \times m$-Matrix; setze $A := A_1$.
Iteration: *Für $k = 1, 2, ...$ bilde die LR-Zerlegung von A_k (falls diese existiert),*

$$A_k = L_k R_k,$$

und berechne

$$A_{k+1} := R_k L_k.$$

Man stoppe die Iteration, falls die Komponenten von A_{k+1} unterhalb der Diagonalen kleiner sind als eine vorgegebene Fehlerschranke.

Es stellt sich natürlich die Frage, ob das LR-Verfahren konvergiert. Wir wenden uns daher nun der *Konvergenzbetrachtung des LR-Verfahrens* zu und formulieren folgendes Resultat. Der Beweis findet sich im Anschluss.

Resultat 2.9 (Konvergenz des LR-Verfahrens)**.**
Für die Matrix $A =: A_1 \in \mathbb{K}^{m \times m}$ seien folgende Voraussetzungen erfüllt:

1. *A sei diagonalisierbar,*

   $$A = T \operatorname{diag}(\lambda_1, ..., \lambda_m) T^{-1},$$

   wobei für T und für T^{-1} die LR-Zerlegungen (ohne Zeilenvertauschungen) existieren mögen.

2. *Die Eigenwerte von A lassen sich betragsmäßig ordnen gemäß*

   $$|\lambda_1| > |\lambda_2| > ... > |\lambda_m| > 0.$$

3. Das LR-Verfahren (ohne Zeilenvertauschungen)

$$A_k =: L_k R_k, \qquad A_{k+1} := R_k L_k, \qquad k = 1, 2, ...,$$

sei durchführbar, d.h. in jedem k-ten Schritt existiere die LR-Zerlegung der Matrix A_k. Dann gilt

$$\lim_{k\to\infty} A_k = \lim_{k\to\infty} R_k = \begin{bmatrix} \lambda_1 & * & \dots & \dots & * \\ & \lambda_2 & \ddots & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & * \\ 0 & & & & \lambda_m \end{bmatrix}$$

und

$$\lim_{k\to\infty} L_k = E.$$

Beweis: Wir führen nun den Beweis der Konvergenz des LR-Verfahrens unter den obigen Annahmen. Aus der Darstellung

$$A = TDT^{-1}$$

mit

$$D = \mathrm{diag}(\lambda_1, ..., \lambda_m)$$

folgt für die Potenzen A^k, $k = 1, 2, ...$

$$A^k = (TDT^{-1})^k = TD^kT^{-1}.$$

Wenn für T und T^{-1} die LR-Zerlegungen

$$\begin{aligned} T &= L_T R_T, \\ T^{-1} &= L_{T^{-1}} R_{T^{-1}} \end{aligned}$$

existieren, so folgt

$$A^k = L_T R_T D^k L_{T^{-1}} R_{T^{-1}}.$$

Mit A ist auch D regulär. In diesem Fall gilt

$$A^k = L_T R_T (D^k L_{T^{-1}} D^{-k}) D^k R_{T^{-1}}, \qquad k = 1, 2, ...$$

In der folgenden Aufgabe zeigen wir, dass in gewissen Fällen die Folge $\left\{D^k L_{T^{-1}} D^{-k}\right\}_{k\in\mathbb{N}}$ gegen die Einheitsmatrix strebt.

Aufgabe 2.10.
Es gelte

$$|\lambda_1| > |\lambda_2| > ... > |\lambda_m| > 0.$$

Dann konvergiert die Folge

$$\left\{D^k L_{T^{-1}} D^{-k}\right\}_{k=1,2,...}, \qquad D := \mathrm{diag}(\lambda_1, ..., \lambda_m),$$

geometrisch gegen die Einheitsmatrix:

$$\begin{aligned} D^k L_{T^{-1}} D^{-k} &= E + F_k, \qquad k = 1, 2, ..., \\ \lim_{k\to\infty} F_k &= 0. \end{aligned}$$

Dieses Resultat verwenden wir nun, um das Verhalten der Matrixpotenzen A^k, $k = 1, 2, ...$, zu untersuchen. Es folgt

$$\begin{aligned} A^k &= L_T R_T (D^k L_{T^{-1}} D^{-k}) D^k R_{T^{-1}} \\ &= L_T R_T (E + F_k) D^k R_{T^{-1}} \\ &= L_T (E + R_T F_k R_T^{-1}) R_T D^k R_{T^{-1}}. \end{aligned}$$

Wir beachten, dass mit

$$\lim_{k\to\infty} F_k = 0$$

auch

$$\lim_{k\to\infty} R_T F_k R_T^{-1} = 0$$

gilt. Für genügend große k existiert somit die LR-Zerlegung der Matrix

$$E + R_T F_k R_T^{-1} =: \hat{L}_k \hat{R}_k.$$

Dabei gilt

$$\lim_{k\to\infty} \hat{L}_k = E = \lim_{k\to\infty} \hat{R}_k,$$

wie man unmittelbar aus

$$\lim_{k\to\infty} R_T F_k R_T^{-1} = 0$$

folgert.

Für hinreichend großes k folgt also für die Matrixpotenzen A^k die Darstellung

$$A^k = L_T \hat{L}_k \hat{R}_k R_T D^k R_{T^{-1}}.$$

In dieser Faktorisierung ist $L_T \hat{L}_k$ eine untere Dreiecksmatrix mit normierter Diagonale und $\hat{R}_k R_T D^k R_{T^{-1}}$ eine obere Dreiecksmatrix. Also folgt aufgrund der Eindeutigkeit der LR-Zerlegung durch Vergleich mit Resultat 2.7:

$$\tilde{L}_k = L_T \hat{L}_k, \qquad \tilde{R}_k = \hat{R}_k R_T D^k R_{T^{-1}},$$

wobei - wie oben gezeigt -

$$\lim_{k\to\infty} \hat{L}_k = \lim_{k\to\infty} \hat{R}_k = E,$$

gilt. Wegen

$$L_k = \tilde{L}_{k-1}^{-1} \tilde{L}_k = \hat{L}_{k-1}^{-1} L_T^{-1} L_T \hat{L}_k = \hat{L}_{k-1}^{-1} \hat{L}_k$$

und

$$\begin{aligned} R_k = \tilde{R}_k \tilde{R}_{k-1}^{-1} &= \hat{R}_k R_T D^k R_{T^{-1}} R_{T^{-1}}^{-1} D^{-(k-1)} R_T^{-1} \hat{R}_{k-1}^{-1} \\ &= \hat{R}_k R_T D R_T^{-1} \hat{R}_{k-1}^{-1} \end{aligned}$$

folgt somit

$$\begin{aligned} \lim_{k\to\infty} L_k &= E \\ \lim_{k\to\infty} R_k &= R_T D R_T^{-1} \end{aligned}$$

und damit

$$\lim_{k\to\infty} A_k = \lim_{k\to\infty} L_k R_k = R_T D R_T^{-1}.$$

Dabei hat die Matrix $R_T D R_T^{-1}$ obere Dreiecksform, und in der Diagonalen stehen die Diagonalelemente von D:

$$R_T D R_T^{-1} = \begin{bmatrix} \lambda_1 & * & \dots & \dots & * \\ & \lambda_2 & \ddots & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & * \\ 0 & & & & \lambda_m \end{bmatrix}.$$

Insgesamt ist damit gezeigt, dass das LR-Verfahren in gewissen Fällen (nämlich unter obigen Annahmen an die Matrix A) iterativ sämtliche Eigenwerte von A liefert. Dies beendet die Konvergenzuntersuchung. □

Bemerkung 2.11.
Ist die Matrix A positiv definit, so kann unter Verwendung der Cholesky-Zerlegung eine symmetrische Variante des LR-Verfahrens durchgeführt werden:

$$\begin{aligned} A &=: A_1, \\ A_k &=: L_k L_k^H, \qquad A_{k+1} := L_k^H L_k, \qquad k = 1, 2, ... \end{aligned}$$

Es ist zu bemerken, dass die Matrizen A_k alle postiv definit sind; folglich existiert in jedem Iterationsschritt die Cholesky-Zerlegung.

Die wesentliche Schwäche des LR-Verfahrens liegt darin, dass die LR-Zerlegung nicht für jede reguläre Matrix notwendig existiert bzw. ihre Berechnung numerisch instabil ist. Wie wir in Abschnitt 1.4 gesehen haben, existiert aber bei regulärer Matrix stets eine QR-Zerlegung, die z.B. nach Algorithmus 1.27 berechnet werden kann. Analog zum LR-Verfahren erhalten wir so das von Francis vorgeschlagene QR-Verfahren:

Algorithmus 2.12 (QR-Verfahren)**.**
Sei $A =: A_1$ eine reguläre $m \times m$-Matrix.
Iteration: *Für $k = 1, 2, ...$ zerlege A_k:*

$$A_k =: Q_k R_k,$$

wo Q_k unitär und R_k obere Dreiecksmatrix ist, und berechne

$$A_{k+1} := R_k Q_k.$$

Breche die Iteration ab, wenn die Komponenten von A_{k+1} unterhalb der Diagonalen dem Betrag nach kleiner sind als eine gegebene Fehlerschranke.

Zur Untersuchung des QR-Verfahrens gehen wir ähnlich wie beim LR-Verfahren vor.

Aufgabe 2.13.
Man zeige, dass mit den in Algorithmus 2.12 gegebenen Matrizen Q_i, R_i eine QR-Zerlegung der Matrixpotenzen A^k aufgebaut werden kann: Man beweise

$$A^k = \tilde{Q}_k \tilde{R}_k, \qquad k = 1, 2, ...,$$

wobei

$$\begin{aligned}\tilde{Q}_k &:= Q_1 Q_2 \cdots Q_k,\\ \tilde{R}_k &:= R_k R_{k-1} \cdots R_1\end{aligned}$$

gesetzt wurde.

Es sei A diagonalisierbar mit dominantem Eigenwert

$$\lambda_1 =: |\lambda_1| e^{i\varphi}.$$

Es gilt wiederum

$$\frac{A^k e_1}{\lambda_1^k} \to \rho_1 u_1, \qquad \frac{\|A^k e_1\|}{|\lambda_1^k|} \to \|\rho_1 u_1\| \text{ für } k \to \infty.$$

Daraus folgt

$$\begin{aligned}e^{-ik\varphi}\frac{A^k e_1}{\|A^k e_1\|} &\longrightarrow \frac{\rho_1 u_1}{\|\rho_1 u_1\|},\\ e^{-i(k-1)\varphi}\frac{A^k e_1}{\|A^{k-1} e_1\|} &\longrightarrow \lambda_1 \frac{\rho_1 u_1}{\|\rho_1 u_1\|} \qquad \text{für } k \to \infty,\end{aligned}$$

falls der kanonische Einheitsvektor e_1 in Bezug auf den zu λ_1 gehörenden Eigenvektor u_1 eine nichtverschwindende Komponente ρ_1 hat. Mit

$$\begin{aligned}R_k &= (r_{ij}^{(k)})_{m\times m},\\ \tilde{Q}_k &= \left[\begin{array}{c|c|c|c} & & & \\ \tilde{q}_1^{(k)} & \tilde{q}_2^{(k)} & \dots & \tilde{q}_m^{(k)} \\ & & & \end{array}\right], \qquad k = 1, 2, ...\end{aligned}$$

folgt

$$A^k e_1 = r_{11}^{(1)} \cdot r_{11}^{(2)} \cdots r_{11}^{(k)} \tilde{q}_1^{(k)}, \qquad \|A^k e_1\| = |r_{11}^{(1)} \cdots r_{11}^{(k)}|,$$

und falls wir

$$\begin{aligned}r_{11}^{(1)} \cdots r_{11}^{(k)} &:= |r_{11}^{(1)} \cdots r_{11}^{(k)}| \cdot e^{i\psi_k} \qquad k = 1, 2, ...,\\ \tilde{q}_1^{(k)} &=: (\tilde{q}_{11}^{(k)}, ..., \tilde{q}_{m1}^{(k)})^t \qquad k = 1, 2, ...,\\ u_1 &=: (\eta_1, ...\eta_m)^t\end{aligned}$$

setzen, ergibt sich für $\nu = 1, ..., m$:

$$\begin{aligned}e^{i(\psi_k - k\varphi)}\tilde{q}_{\nu 1}^{(k)} &\overset{k\to\infty}{\longrightarrow} \frac{\rho_1 \eta_\nu}{\|\rho_1 u_1\|},\\ e^{i(\psi_{k-1} - (k-1)\varphi)} r_{11}^{(k)} \tilde{q}_{\nu 1}^{(k)} &\overset{k\to\infty}{\longrightarrow} \lambda_1 \frac{\rho_1 \eta_\nu}{\|\rho_1 u_1\|}.\end{aligned}$$

Für $\eta_\nu \neq 0$ erhalten wir schließlich unter Verwendung der Phasenfaktoren $e^{i\phi_k}$ mit

$$\phi_k := \varphi + \psi_{k-1} - \psi_k$$

die Konvergenzaussage

$$e^{i\phi_k} r_{11}^{(k)} \longrightarrow \lambda_1 \qquad \text{für } k \to \infty.$$

Die links oben stehenden Komponenten der Matrizen R_k konvergieren demnach nur bis auf einen Phasenfaktor gegen λ_1. (Wählt man hierbei in jedem Schritt die QR-Zerlegung so, dass die Matrizen R_k reelle positive Diagonalelemente haben, so gilt $\psi_k = 0$, $k = 1, 2, ...$, und $r_{11}^{(k)} \to |\lambda_1|$ für $k \to \infty$). Dieses für das QR-Verfahren typische Konvergenzverhalten zeigt sich auch im folgenden Konvergenzsatz:

Satz 2.14 (Konvergenz des QR-Verfahrens)**.**
Für die Matrix $A =: A_1 \in \mathbb{K}^{m\times m}$ seien folgende Voraussetzungen erfüllt:

1. *A sei diagonalisierbar*

$$A = T \operatorname{diag}(\lambda_1, ..., \lambda_m) T^{-1},$$

wobei für T^{-1} die LR-Zerlegung vorausgesetzt wird:

$$T^{-1} = L_{T^{-1}} R_{T^{-1}}.$$

2. *Die Eigenwerte*

$$\lambda_j =: |\lambda_j| e^{i\varphi_j}, \qquad j = 1, ..., m,$$

lassen sich betragsmäßig ordnen gemäß

$$|\lambda_1| > |\lambda_2| > ... > |\lambda_m| > 0.$$

Wählt man in jedem Schritt des QR-Verfahrens

$$A_k =: Q_k R_k, \qquad A_{k+1} := R_k Q_k, \qquad k = 1, 2, ...,$$

die obere Dreiecksmatrix R_k derart, dass in der Diagonalen reelle positive Elemente stehen, so gilt für $k \to \infty$

$$\begin{aligned} Q_k &\longrightarrow \operatorname{diag}(e^{i\varphi_1}, e^{i\varphi_2}, ..., e^{i\varphi_m}), \\ (r_{11}^{(k)}, r_{22}^{(k)}, ..., r_{mm}^{(k)}) &\longrightarrow (|\lambda_1|, |\lambda_2|, ..., |\lambda_m|), \end{aligned}$$

und damit

$$a_{ij}^{(k)} \longrightarrow \begin{cases} 0 & \text{für } i > j, \\ \lambda_i & \text{für } i = j. \end{cases}$$

(Hierbei haben wir $A_k = (a_{ij}^{(k)})_{m\times m}$, $R_k = (r_{ij}^{(k)})_{m\times m}$ gesetzt.)

Beweis: Wir führen den Beweis der Konvergenz des QR-Verfahrens unter obigen Annahmen. In Aufgabe 2.13 haben wir gezeigt, dass die Matrixpotenzen A^k die QR-Zerlegungen

$$\begin{aligned} A^k &= \tilde{Q}_k \tilde{R}_k \qquad \text{mit} \\ \tilde{Q}_k &= Q_1 Q_2 \cdots Q_k \qquad \text{und} \\ \tilde{R}_k &= R_k R_{k-1} \cdots R_1, \qquad k = 1, 2, ..., \end{aligned}$$

besitzen, und diese Zerlegungen sind eindeutig bestimmt, falls für $\tilde{R}_k$ positive Diagonalelemente vorgeschrieben werden. Im folgenden leiten wir eine alternative Darstellung der Matrizen $\tilde{Q}_k$ und $\tilde{R}_k$ ab.

Wir setzen

$$\begin{aligned} D &= \operatorname{diag}(\lambda_1, \lambda_2, ..., \lambda_m), \\ \Delta &= \operatorname{diag}(e^{i\varphi_1}, e^{i\varphi_2}, ..., e^{i\varphi_m}). \end{aligned}$$

Wegen

$$A = TDT^{-1}$$

gilt auch

$$A^k = TD^kT^{-1},$$

und unter Verwendung der QR-Zerlegung von T

$$T = Q_T R_T$$

(R_T besitze positive Diagonalelemente!) und der LR-Zerlegung von T^{-1},

$$T^{-1} = L_{T^{-1}} R_{T^{-1}},$$

ergibt sich

$$A^k = Q_T R_T D^k L_{T^{-1}} R_{T^{-1}}.$$

Für

$$L_k^* := D^k L_{T^{-1}} D^{-k}$$

gilt wegen Aufgabe 2.10

$$L_k^* \to E \qquad \text{für } k \to \infty,$$

da $L_{T^{-1}}$ eine untere Dreiecksmatrix mit normierter Diagonale ist und da die Eigenwerte von A nach Voraussetzung betragsmäßig getrennt liegen. Bildet man die QR-Zerlegung von $R_T L_k^*$ gemäß

$$R_T L_k^* = Q_k^* R_k^*,$$

wobei R_k^* in der Diagonale wiederum positive Elemente besitze, so erhalten wir schließlich

$$A^k = Q_T R_T L_k^* D^k R_{T^{-1}} = Q_T Q_k^* R_k^* D^k R_{T^{-1}}.$$

Hiermit haben wir eine weitere Zerlegung der Matrix A^k in das Produkt einer unitären Matrix $Q_T Q_k^*$ und einer oberen Dreiecksmatrix $R_k^* D^k R_{T^{-1}}$ gefunden. Das Argument in der komplexen Polardarstellung der Diagonalelemente von $R_k^* D^k R_{T^{-1}}$ ist allein durch D^k und die Diagonalelemente $\rho_1, \rho_2, ..., \rho_m$ von $R_{T^{-1}}$ bestimmt, da R_k^* nur positive Diagonalelemente besitzt. Wir setzen

$$\rho_j =: |\rho_j| e^{i\psi_j}, \qquad j = 1, ..., m,$$

und

$$\tilde{\Delta} = \operatorname{diag}(e^{i\psi_1}, e^{i\psi_2}, ..., e^{i\psi_m}).$$

Dann hat

$$\tilde{\Delta}^{-1} \Delta^{-k} R_k^* D^k R_{T^{-1}}$$

in der Diagonalen nur positive Elemente. Da die R_i nach Voraussetzung nur positive Diagonalelemente besitzen, hat auch $\tilde{R}_k$ nur positive Elemente in der Diagonalen. Aufgrund der Eindeutigkeitsaussage für die QR-Zerlegung folgt

$$Q_T Q_k^* \Delta^k \tilde{\Delta} = \tilde{Q}_k = Q_1 Q_2 \cdots Q_k$$

und

$$\tilde{\Delta}^{-1}\Delta^{-k}R_k^*D^kR_{T^{-1}} = \tilde{R}_k = R_kR_{k-1}\cdots R_1.$$

Unter Verwendung dieser Identitäten werden wir die Konvergenzaussagen ableiten können. Zunächst gilt

$$\begin{aligned} Q_k &= \tilde{Q}_{k-1}^{-1}\tilde{Q}_k \\ &= (Q_TQ_{k-1}^*\Delta^{k-1}\tilde{\Delta})^{-1}Q_TQ_k^*\Delta^k\tilde{\Delta} \\ &= \tilde{\Delta}^{-1}\Delta^{-(k-1)}(Q_{k-1}^*)^{-1}Q_k^*\Delta^k\tilde{\Delta}. \end{aligned}$$

Aus

$$R_TL_k^* = Q_k^*R_k^*$$

und

$$L_k^* \to E \qquad \text{für } k \to \infty$$

folgt wegen der Eindeutigkeit der QR-Zerlegung (da sowohl R_T als auch R_k^* in der Diagonale positive Elemente besitzen)

$$Q_k^* \to E, \qquad R_k^* \to R_T \qquad \text{für } k \to \infty.$$

Wegen $Q_k^* \to E$ können wir auf

$$\Delta^{-(k-1)}(Q_{k-1}^*)^{-1}Q_k^*\Delta^k \to \Delta \qquad \text{für } k \to \infty$$

und weiter auf

$$Q_k \to \tilde{\Delta}^{-1}\Delta\tilde{\Delta} = \Delta \qquad \text{für } k \to \infty$$

schließen.
Die Konvergenzaussage für die oberen Dreiecksmatrizen R_k erhält man aus der Darstellung

$$\begin{aligned} R_k &= \tilde{R}_k\tilde{R}_{k-1}^{-1} \\ &= \tilde{\Delta}^{-1}\Delta^{-k}R_k^*D^kR_{T^{-1}}R_{T^{-1}}^{-1}D^{-(k-1)}(R_{k-1}^*)^{-1}\Delta^{k-1}\tilde{\Delta} \\ &= \tilde{\Delta}^{-1}\Delta^{-k}R_k^*D(R_{k-1}^*)^{-1}\Delta^{k-1}\tilde{\Delta} \end{aligned}$$

Wegen

$$R_k^*D(R_{k-1}^*)^{-1} \to R_TDR_T^{-1} = \begin{bmatrix} \lambda_1 & * & \dots & * \\ & \ddots & \ddots & \vdots \\ & & \ddots & * \\ 0 & & & \lambda_m \end{bmatrix}, \qquad k \to \infty,$$

konvergiert die Diagonale von $\Delta^{-k}R_k^*D(R_{k-1}^*)^{-1}\Delta^{k-1}$ gegen die Beträge der Eigenwerte, woraus wegen der Diagonalgestalt der Matrix $\tilde{\Delta}$ die im Satz behauptete Konvergenzaussage für die Matrizen R_k folgt.
Die Konvergenzaussage für die Matrizen A_k erhält man schließlich direkt aus der Darstellung

$$A_k = Q_kR_k, \qquad k = 1, 2, ...,$$

unter Verwendung der schon bewiesenen Konvergenzaussagen für die Matrizen Q_k und R_k. Dies beendet den Beweis der Konvergenz des QR-Verfahrens. □

Wir schließen diesen Abschnitt mit einigen Bemerkungen.

(1) Die für die Konvergenzaussagen 2.9 und 2.14 wesentliche Forderung an die Matrix A ist die Eigenschaft, dass A *diagonalisierbar* ist. Die anderen Forderungen, insbesondere die Forderung

$$|\lambda_1| > |\lambda_2| > ... > |\lambda_m|,$$

können abgeschwächt werden, ohne dass die Konvergenzaussage wesentlich eingeschränkt wird. (Wir verweisen auf Wilkinson: The Algebraic Eigenvalue Problem, Oxford, Clarendon Press, 1965). Es ist allerdings darauf hinzuweisen, dass das LR-Verfahren selbst bei theoretisch gesicherter Konvergenz zusammenbrechen kann, da die Berechnung der LR-Zerlegung in einem Iterationsschritt auf numerische Instabilität führen kann.

(2) Vom Standpunkt der Praxis aus kann man sich bei Hermiteschen Matrizen A immer auf den Spezialfall reeller Matrizen beschränken. Denn, falls A nicht reell ist, so kann man komplexe Arithmetik dadurch vermeiden, dass man A zerlegt in Real- und Imaginärteil,

$$A = A_1 + iA_2$$

und anstelle von A die Matrix

$$\tilde{A} = \begin{bmatrix} A_1 & -A_2 \\ A_2 & A_1 \end{bmatrix}$$

betrachtet. Es ist einfach einzusehen, dass ein Eigenwert λ von A ein zweifacher Eigenwert von $\tilde{A}$ ist. (Verifizieren Sie diese Aussage!)

(3) Der unverhältnismäßig hohe numerische Aufwand von $\mathcal{O}(m^3)$ Rechenoperationen in jedem Schritt der beiden Verfahren kann dadurch vermindert werden, dass man die vorliegende Matrix zuerst mit der in folgenden Abschnitt 2.5 beschriebenen Methoden auf einfachere Gestalt (Tridiagonalform bzw. Hessenberg-Form) bringt. Wesentlich ist, dass diese spezielle Form bei LR- bzw. QR-Verfahren invariant bleibt, so dass der Aufwand pro Iterationsschritt auf $\mathcal{O}(m)$ bzw. $\mathcal{O}(m^2)$ Punktoperationen reduziert wird.

(4) Auch bei Anwendung des LR- bzw. QR-Verfahrens auf eine Matrix $A(=: A_1)$ mit spezieller Bandgestalt haben die in den Verfahren berechneten Matrizen A_k, $k = 1, 2, ...$, dieselbe Bandgestalt, so dass auch hier der Aufwand pro Iterationsschritt vermindert wird. Dies zeigt die folgende Aufgabe:

Aufgabe 2.15.
Man zeige

1. *Ist* $A_1 =: (a_{ij})_{m\times m}$ *eine Bandmatrix mit Bandbreite* d*, d.h. gilt*

$$a_{ij} = 0 \qquad \text{für } |i-j| \geq d,$$

so sind beim LR-Verfahren die A_k*,* $k = 1, 2, ...$*, ebenfalls Bandmatrizen der Bandbreite* d*. Insbesondere bleibt etwa Tridiagonalgestalt (Fall* $d = 2$*) erhalten (*Tridiagonalgestalt *siehe Abschnitt 2.5).*

2. *Ist* A_1 *eine reguläre Hermitesche Bandmatrix mit Bandbreite* d*, so sind es beim QR-Verfahren die* A_k*,* $k = 1, 2, ...$*, ebenfalls.*

3. *Ist* $A_1 =: (a_{ij})_{m\times m}$ *eine reguläre Matrix mit unterem Band der Breite* d, *d.h., gilt*

$$a_{ij} = 0 \qquad \text{für } i - j \geq d,$$

so sind es beim LR-Verfahren die A_k, $k = 1, 2, ...$, *ebenfalls. Insbesondere bleibt die obere Hessenberg-Form (Fall* $d = 2$*) erhalten (*obere Hessenberg-Form *siehe Abschnitt 2.5).*

4. *Ist* A_1 *eine reguläre Matrix mit unterem Band der Breite* d, *so sind es beim QR-Verfahren die* A_k, $k = 1, 2, ...$, *ebenfalls.*

Im Folgenden behandeln wir abschließend *shift-Methoden* in Verbindung mit LR- und QR-Verfahren.

In den Konvergenzsätzen 2.9 und 2.14 hatten wir gesehen, dass in den Matrizen A_k die Elemente unterhalb der Diagonalen gegen Null und die Diagonalelemente gegen die Eigenwerte der Ausgangsmatrix streben. Man darf also erwarten, dass das Element $a_{mm}^{(k)}$ eine von Schritt zu Schritt bessere Näherung für den Eigenwert λ_m wird. Shift-Methoden für LR- und QR-Verfahren werden gemäss folgendem Schema gebildet:

Seien σ_k, $k = 1, 2, ...$, Näherungen für λ_m. Man bilde für $k = 1, 2, ...$ (mit $A =: A_1$)

$$A_k - \sigma_k E =: L_k R_k, \qquad A_{k+1} := R_k L_k + \sigma_k E \qquad \textit{(LR-Verfahren mit shift)}$$

bzw.

$$A_k - \sigma_k E =: Q_k R_k, \qquad A_{k+1} := R_k Q_k + \sigma_k E \qquad \textit{(QR-Verfahren mit shift).}$$

Folgende Strategien haben sich in der Praxis bewährt (wir setzen $A_k = (a_{ij}^{(k)})_{m\times m}$):
Strategie a: $\sigma_k = a_{mm}^{(k)}$.
Strategie b: σ_k wird gewählt als derjenige Eigenwert der 2×2-Matrix

$$\begin{bmatrix} a_{m-1,m-1}^{(k)} & a_{m-1,m}^{(k)} \\ a_{m,m-1}^{(k)} & a_{m,m}^{(k)} \end{bmatrix}$$

der am nächsten bei $a_{mm}^{(k)}$ liegt.
Man kann zeigen, dass die Strategie b, falls sie auf eine (nicht-zerlegbare) Hermitesche Tridiagonalmatrix angewendet wird, mindestens quadratische Konvergenz für

$$\begin{aligned} a_{m,m-1}^{(k)} &\longrightarrow 0, \\ a_{m,m}^{(k)} &\longrightarrow \lambda_m \qquad \text{für } k \to \infty \end{aligned}$$

liefert; in vielen Fällen kann sogar kubische Konvergenz gesichert werden.

Zusammenfassung:

In diesem Abschnitt wurden Ihnen zwei vom Typ ähnliche Iterationsverfahren zur simultanen Berechnung aller Eigenwerte einer Matrix beschrieben. Für beide Verfahren sind unter relativ allgemeinen Voraussetzungen globale Konvergenzaussagen bekannt. In der Praxis werden diese Verfahren üblicherweise nur unter Verwendung von shift-Techniken benützt, um lokal eine höhere Konvergenzordnung zu erhalten.

2.5 Die Reduktionsmethode von Householder auf Hessenberg- bzw. Tridiagonalform (Ergänzung)

Im vorangehenden Abschnitt hatten wir Methoden kennengelernt, die es gestatten, unter gewissen einschränkenen Bedingungen die Eigenwerte einer Matrix A in geschickter Weise zu berechnen. Wir wenden uns nun dem Problem zu, wie man für eine gegebene Matrix A die sogenannte Hessenberg-Form bzw. für eine Hermitesche Matrix die Hermitesche Tridiagonalgestalt erreichen kann. Damit lässt sich der Aufwand der LR- bzw., QR-Verfahrens erheblich reduzieren.

Eine Matrix A hat *obere Hessenberg-Form*, falls sie von folgender Bauart ist:

$$A = \begin{bmatrix} * & \dots & \dots & \dots & * \\ * & \ddots & & & \vdots \\ & \ddots & \ddots & & \vdots \\ & & \ddots & \ddots & * \\ 0 & & & * & * \end{bmatrix}.$$

Eine Matrix A hat *Tridiagonalgestalt*, falls sie von folgender Bauart ist:

$$A = \begin{bmatrix} * & * & & & 0 \\ * & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & * \\ 0 & & & * & * \end{bmatrix}.$$

Bei dem oben formulierten Problem handelt es sich also um Folgendes. Gegeben sei eine Matrix $A \in \mathbb{K}^{m\times m}$ ($\mathbb{K} = \mathbb{R}$ oder $\mathbb{K} = \mathbb{C}$). Da *Ähnlichkeitstransformationen*

$$A \longmapsto T^{-1}AT, \qquad T \text{ regulär},$$

und insbesondere *unitäre Transformationen*

$$A \longmapsto U^H AU, \qquad U \text{ unitär}, \qquad \text{d.h. } U^H = U^{-1},$$

die Eigenwerte von A invariant lassen, kann man versuchen, möglichst einfache Transformationsmatrizen zu finden, die A auf Hermitesche Tridiagonalform oder auf Hessenberg-Form bringen. Es wird sich zeigen, dass man mit einer *endlichen Kette solcher Transformationen* dieses Problem lösen kann:

Sei

$$\begin{aligned} A^{(0)} &:= A, \\ A^{(1)} &:= U_1^H A^{(0)} U_1, \\ &\vdots \\ A^{(k)} &:= U_k^H A^{(k-1)} U_k; \end{aligned}$$

dann hat bei geeigneter Wahl von $U_1, ..., U_k$ die Endmatrix $A^{(k)}$ die gewünschte Hessenberg- oder Hermitesche Tridiagonalform. Die Länge k der Kette hängt von der Zeilenzahl von A und der Bauart der verwendeten Transformationsmatrizen $U_\nu, \nu = 1, ..., k$, ab.

Aufgabe 2.16.
Es sei

$$A = \begin{bmatrix} 2 & 3 & 1 \\ 3 & 4 & 5 \\ 1 & 5 & 6 \end{bmatrix}.$$

1. *Man konstruiere eine Householder-Matrix*

$$H_w := E - 2ww^H, \qquad w^H w = 1,$$

welche den Vektor $(3,1)^t$ *in die Richtung des ersten Einheitsvektors im* $\mathbb{R}^2$ *spiegelt:*

$$H_w : \begin{bmatrix} 3 \\ 1 \end{bmatrix} \longmapsto \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

2. *Man bilde*

$$U := \begin{bmatrix} 1 & 0 & 0 \\ 0 & & \\ 0 & & H_w \end{bmatrix}$$

und zeige, dass

$$A^{(1)} := U^H A U$$

Hermitesche Tridiagonalform hat.

Bei dieser Aufgabe haben wir schon das Prinzip der *Reduktionsmethode von Householder* kennengelernt. Wir betrachten nun eine beliebige m-reihige Matrix A, die wir auf obere Hessenberg-Form transformieren wollen. Es soll gezeigt werden, dass dies mit $m-2$ unitären Transformationen, die aus Householder-Spiegelungen aufgebaut sind, zu erreichen ist.

Wir gehen davon aus, dass durch $i-1$ Schritte ($i \geq 2$) von „links oben her“ schon ein Stück obere Hessenberg-Form erzeugt ist, d.h. die Ausgangsmatrix $A = A^{(0)}$ sei unitär ähnlich einer Matrix $A^{(i-1)}$ des Typs:

$$A^{(i-1)} = U_{i-1}^H \cdots U_1^H A^{(0)} U_1 \cdots U_{i-1} = \begin{bmatrix} & & \vdots & & \\ & B_i & \vdots & D_{m-i} & \\ & & \vdots & & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ & \vdots & \vdots & & \\ \mathcal{O} & \vdots & b_i & \vdots & C_{m-i} \\ & \vdots & \vdots & & \end{bmatrix};$$

$$\begin{array}{rcl} B_i & : & i \times i\text{-Matrix in oberer Hessenbergform;} \\ C_{m-i} & : & (m-i) \times (m-i)\text{-Matrix;} \\ b_i & \in & \mathbb{K}^{m-i} \qquad (\mathbb{K} = \mathbb{R} \text{ oder } \mathbb{K} = \mathbb{C}); \\ D_{m-i} & : & i \times (m-i)\text{-Matrix.} \end{array}$$

Nach Abschnitt 1.4, wo wir die QR-Zerlegung behandelt haben, existiert eine Householdertransformation

$$H_i := E - 2w_i w_i^H, \qquad w_i^H w_i = 1,$$

auf dem Raum $\mathbb{C}^{m-i}$, die den Vektor b_i in ein skalares Vielfaches des kanonischen Einheitsvektors $e_1 \in \mathbb{C}^{m-i}$ spiegelt:

$$H_i : b_i \longmapsto \alpha_i e_i \in \mathbb{C}^{m-i}.$$

Wie man bei gegebenem Vektor b_i den entsprechenden Vektor w_i numerisch am günstigsten bestimmt, haben wir bei der Behandlung der QR-Zerlegung gezeigt. Hier geht es darum, die Transformationsmatrix U_i zu definieren. Wir setzen (entsprechend der Einteilung der Matrix $A^{(i-1)}$):

$$U_i := \begin{bmatrix} 1 & & 0 & \vdots & & & \\ & \ddots & & \vdots & & \mathcal{O} & \\ 0 & & 1 & \vdots & & & \\ \dots & \dots & \dots & & \dots & \dots & \dots \\ & & & \vdots & & & \\ & \mathcal{O} & & \vdots & & H_i & \\ & & & \vdots & & & \end{bmatrix}.$$

Da H_i Hermitesch und unitär ist,

$$H_i = H_i^H = H_i^{-1},$$

gilt dasselbe auch für U_i

$$U_i = U_i^H = U_i^{-1}.$$

Wir betrachten jetzt die unitäre Ähnlichkeitstransformation

$$A^{(i-1)} \longmapsto U_i^H A^{(i-1)} U_i = U_i A^{(i-1)} U_i =: A^{(i)}.$$

Aufgrund der Blockstruktur der Matrizen $A^{(i-1)}$ und U_i ergibt sich

$$
\begin{aligned}
A^{(i)} &= \begin{bmatrix}
1 & & 0 & \vdots & & \\
& \ddots & & \vdots & & \mathcal{O} \\
0 & & 1 & \vdots & & \\
\cdots & \cdots & \cdots & & \cdots & \cdots & \cdots \\
& & & \vdots & & & \\
& \mathcal{O} & & \vdots & & H_i & \\
& & & \vdots & & &
\end{bmatrix} \cdot \begin{bmatrix}
& & & \vdots & & & \\
& & B_i & \vdots & & D_{m-i} & \\
& & & \vdots & & & \\
\cdots & \cdots & \cdots & & \cdots & \cdots & \cdots \\
& & \vdots & \vdots & & & \\
\mathcal{O} & \vdots & b_i & \vdots & & C_{m-i} & \\
& \vdots & & \vdots & & &
\end{bmatrix} \cdot \\
&= \begin{bmatrix}
1 & & 0 & \vdots & & \\
& \ddots & & \vdots & & \mathcal{O} \\
0 & & 1 & \vdots & & \\
\cdots & \cdots & \cdots & & \cdots & \cdots & \cdots \\
& & & \vdots & & & \\
& \mathcal{O} & & \vdots & & H_i & \\
& & & \vdots & & &
\end{bmatrix} \\
&= \begin{bmatrix}
1 & & 0 & \vdots & & \\
& \ddots & & \vdots & & \mathcal{O} \\
0 & & 1 & \vdots & & \\
\cdots & \cdots & \cdots & & \cdots & \cdots & \cdots \\
& & & \vdots & & & \\
& \mathcal{O} & & \vdots & & H_i & \\
& & & \vdots & & &
\end{bmatrix} \cdot \begin{bmatrix}
& & & \vdots & & & \\
& & B_i & \vdots & & D_{m-i}H_i & \\
& & & \vdots & & & \\
\cdots & \cdots & \cdots & & \cdots & \cdots & \cdots \\
& & \vdots & \vdots & & & \\
\mathcal{O} & \vdots & b_i & \vdots & & C_{m-i}H_i & \\
& \vdots & & \vdots & & &
\end{bmatrix} \\
&= \begin{bmatrix}
& & & \vdots & & \\
& B_i & & \vdots & & D_{m-i}H_i \\
& & & \vdots & & \\
\cdots & \cdots & \cdots & & \cdots & \cdots & \cdots \\
& \vdots & \alpha_i & \vdots & & & \\
\mathcal{O} & \vdots & 0 & \vdots & & H_iC_{m-i}H_i & \\
& \vdots & \vdots & \vdots & & & \\
& \vdots & 0 & \vdots & & &
\end{bmatrix} =: \begin{bmatrix}
& & & \vdots & & \\
& B_{i+1} & & \vdots & & D_{m-i-1} \\
& & & \vdots & & \\
\cdots & \cdots & \cdots & & \cdots & \cdots & \cdots \\
& \vdots & & \vdots & & & \\
\mathcal{O} & \vdots & b_{i+1} & \vdots & & C_{m-i-1} & \\
& \vdots & & \vdots & & &
\end{bmatrix},
\end{aligned}
$$

wobei die $(i+1) \times (i+1)$-Matrix B_{i+1} obere Hessenberg-Form besitzt.

Wir erkennen somit, dass jede m-reihige Matrix A durch $m-2$ Ähnlichkeitstransformationen mittels unitärer Transformationsmatrizen auf obere Hessenberg-Form gebracht

werden kann:

$$A \longmapsto \underbrace{U_{m-2}U_{m-1}\cdots U_1}_{=U^H} A \underbrace{U_1U_2\cdots U_{m-2}}_{=:U} = A^{(m-2)} =: \tilde{A},$$

$$U^HU = E, \qquad \tilde{A} = \begin{bmatrix} * & \dots & \dots & \dots & * \\ * & \ddots & & & \vdots \\ & \ddots & \ddots & & \vdots \\ & & \ddots & \ddots & * \\ 0 & & & * & * \end{bmatrix}.$$

Ist A Hermitesch, so liefert der Transformationsprozess wegen

$$\tilde{A}^H = (U^HAU)^H = U^HA^HU = U^HAU = \tilde{A}$$

sogar in $\tilde{A}$ eine Hermitesche Tridiagonalmatrix.

Resultat 2.17 (Reduktion auf obere Hessenberg-Form)**.**
Eine beliebige m-reihige Matrix A lässt sich mit Hilfe von $m-2$ Householder-Transformationen auf eine obere Hessenberg-Form bringen. Ist A Hermitesch, so liefert dieser Transformationsprozess sogar eine Hermitesche Tridiagonalmatrix.

Algorithmus 2.18 (Reduktionsverfahren nach Householder)**.**
Start: *Es sei*

$$A^{(0)} := (a^{(0)}_{\nu\mu})_{\nu,\mu=1,\dots,m} := A = (a_{\nu\mu})_{\nu,\mu=1,\dots,m}.$$

Iteration: *Für $i = 1, ..., m-2$ bilde man*

$$\begin{aligned} A^{(i)} &= (a^{(i)}_{\nu\mu})_{\nu,\mu=1,\dots,m} \qquad \textit{gemäß} \\ A^{(i)} &:= U_iA^{(i-1)}U_i, \end{aligned}$$

wobei

$$U_i := \begin{bmatrix} 1 & & 0 & \vdots & & & \\ & \ddots & & \vdots & & \mathcal{O} & \\ 0 & & 1 & \vdots & & & \\ \dots & \dots & \dots & & \dots & \dots & \dots \\ & & & \vdots & & & \\ & \mathcal{O} & & \vdots & & H_i & \\ & & & \vdots & & & \end{bmatrix}$$

Die $(m-1)$-reihige Matrix H_i ist eine Householder-Spiegelung, die wir mit dem Algorithmus 1.27 durch die Forderung erhalten, dass H_i die i-te Spalte von $A^{(i-1)}$ unterhalb der Diagonalen auf ein skalares Vielfaches des ersten Einheitsvektors von $\mathbb{C}^{m-i}$ spiegelt.

Bemerkung 2.19.
Es ist naheliegend zu fragen, ob mit einer endlichen *Kette solcher Ähnlichkeitstransformationen*

$$A^{(0)} := A \longrightarrow A^{(1)} := T_1^{-1}A^{(0)}T_1 \longrightarrow A^{(2)} := T_2^{-1}A^{(1)}T_2 \qquad ...$$

die volle Jordansche Normalform, d.h. insbesondere bei Hermiteschen Matrizen die Diagonalform erreichbar ist. Wir machen uns klar, dass dies, wenn die Lösung des Eigenwertproblems noch nicht bekannt ist, wovon wir ja ausgehen wollen, nicht möglich ist. Der Grund ist der folgende: Selbst wenn wir von einer Matrix A mit ganzzahligen Elementen $a_{\nu\mu}$, $\nu, \mu = 1, ..., m$ *ausgehen, können die Eigenwerte als Nullstellen eines Polynom m-ten Grades irrational sein und auch nicht durch einen endlichen arithmetischen Ausdruck, der nur die Grundrechenarten und die Quadratwurzelrechnung verwendet, berechnet werden. (In der Sprache der Algebra ausgedrückt: Die Wurzeln eines Polynoms m-ten Grades,* $m > 2$, *liegen sogar bei ganzzahligen Koeffizienten i. A. nicht in einem Erweiterungskörper von* $\mathbb{Q}$, *der durch Adjunktion von Quadratwurzeln entsteht.)Es ist also prinzipiell nicht möglich, mit Reduktionsmethoden des betrachteten Typs die Jordan-Normalform zu erreichen. Deswegen war es auch sinnvoll, dass wir uns mit der Hessenberg- bzw. der Hermiteschen Tridiagonalform begnügten.*

Zusammenfassung:

In diesem Abschnitt wurde Ihnen eine Methode vorgestellt, wie beliebige Matrizen auf obere Hessenberg-Form und insbesondere Hermitesche Matrizen auf Hermitesche Tridiagonalform transformiert werden können. Damit erhalten die Verfahren, die Sie im vorangehenden Abschnitt kennengelernt haben, erst ihre wahre Bedeutung, da Sie nunmehr Algorithmen für die Berechnung der Eigenwerte beliebiger Matrizen in der Hand haben.

Kapitel 3

Lineare Optimierung

Definition 3.1 (lineares Optimierungsproblem)**.**
Gesucht ist ein Extremum - das kann ein Maximum oder ein Minimum sein - der Zielfunktion

$$z(x) = \sum_{k=1}^{n} c_k x_k, \qquad (c_1, \ldots, c_n \in \mathbb{R} \text{ sind gegeben})$$

wobei die m Nebenbedingungen $\sum_{k=1}^{n} a_{ik} x_k \sim b_i$, $(i = 1, ..., m)$ *erfüllt sein müssen. Dabei steht* $\sim$ *für eines der Zeichen* $=$, $\leq$ *oder* $\geq$.

Definition 3.2 (Standardmodell)**.**
Ein Standardmodell ist ein lineares Optimierungsproblem, bei dem das Maximum der Zielfunktion

$$z(x) = \sum_{k=1}^{n} c_k x_k$$

gesucht wird, wobei die m Nebenbedingungen $\sum_{k=1}^{n} a_{ik} x_k \leq b_i$, $(i = 1, ..., m)$ *sowie* $x_k \geq 0$, $(k = 1, ..., n)$ *erfüllt sein müssen.*

Satz 3.3.
Jedes lineare Optimierungsproblem lässt sich in ein Standardmodell überführen, indem man die folgenden Umforungen durchführt:

1. *Eine Nebenbedingung in Gleichungsform wird durch 2 Ungleichungen mit* $\leq$ *und* $\geq$ *ersetzt.*

2. *Eine* $\geq$*-Ungleichung wird mit* (-1) *multipliziert, damit sich das Ungleichheitszeichen umdreht.*

3. *Ein Minimierungsproblem wird zu einem Maximierungsproblem, indem man die Zielfunktion mit* (-1) *multipliziert.*

4. *Variablen* x_k*, die sowohl negativ also auch positiv sein können, werden als Differenz zweier neuer nicht-negativer Variablen geschrieben. Soll etwas* $x_k \geq 0$ *gelten, so schreibt man* $x_k = y_k - \tilde{y}_k$ *mit* $y_k, \tilde{y}_k \geq 0$.

Definition 3.4 (Normalform)**.**
Ein Standardmodell liegt in Normalform vor, falls $b_i \geq 0$ *gitl für* $i = 1, ..., m$.

Definition 3.5 (Zulässiger Bereich)**.**
Die Teilmenge des $\mathbb{R}^n$, in dem alle Nebenbedingungen eines linearen Optimierungsproblems erfüllt sind, heißt zulässiger Bereich.

In der Regel ist der zulässige Bereich ein n-dimensionales Vieleck (Polyeder).

Satz 3.6 (Hauptsatz der linearen Optimierung)**.**
Falls das lineare Optimierungsproblem eine optimale Lösung besitzt, dann gibt es insbesondere eine optimale Lösung in einem Eckpunkt des zulässigen Bereiches.

Folgerung 3.7.
Das bedeutet, dass es genügt nur die zulässigen Ecken als Kandidaten für die optimale Lösung des Optimierungsproblems zu betrachten.

Definition 3.8 (Schlupfvariable)**.**
Durch die Einführung der Schlupfvariablen $y_1, \ldots, y_m$ werden die Ungleichungen

$$\sum_{k=1}^{n} a_{ik} x_k \leq b_i \quad (i = 1, \ldots, m)$$

zu Gleichungen

$$\sum_{k=1}^{n} a_{ik} x_k + y_i = b_i \quad \textit{mit Nebenbedingung } y_i \geq 0 \qquad (i = 1, \ldots, m).$$

Die Variablen $x_1, \ldots, x_n$ heißen Struktur- oder Problemvariablen; die Variablen $y_1, \ldots, y_m$ heißen Schlupfvariablen.

Durch das Einführen von Schlupfvariablen können die Nebenbedingungen eines Linearen Optimierungsproblems $\sum_{k=1}^{n} a_{ik} x_k \leq b_i$ $(i = 1, ..., m)$ als lineares Gleichungssystem geschrieben werden:

$$\begin{array}{ccccccccccc}
a_{11}x_1 & + & a_{21}x_2 & + & \ldots & + & a_{1n}x_n & + & y_1 & = & b_1 \\
a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & + & y_2 & = & b_2 \\
\vdots & & \vdots & & & & \vdots & & \vdots & \vdots & \vdots \\
a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & + & y_m & = & b_m
\end{array} .$$

In Matrixschreibweise erhält man

$$\begin{pmatrix}
a_{11} & a_{21} & \ldots & a_{1n} & 1 & 0 & \ldots & 0 \\
a_{21} & a_{22} & \ldots & a_{2n} & 0 & 1 & \ldots & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \ldots & a_{mn} & 0 & 0 & \ldots & 1
\end{pmatrix}
\begin{pmatrix}
x_1 \\ \vdots \\ x_n \\ y_1 \\ \vdots \\ y_m
\end{pmatrix}
=
\begin{pmatrix}
b_1 \\ b_2 \\ \vdots \\ b_m
\end{pmatrix} \tag{3.1}$$

oder kürzer

$$Ax + y = b, \qquad \text{bzw.} \qquad (A|E) \begin{pmatrix} x \\ y \end{pmatrix} = b.$$

Dieses besitzt $n + m$ Variablen und m Gleichungen. Daher ist $Rg(A|E) = m$, da die letzten m Spalten linear unabhängig sind, und der Lösungsraum ist n dimensional.

Der zulässige Bereich eines Standardmodells in Normalform (vgl. Definition 3.2 und Definition 3.4) besteht aus den Lösungen des Gleichungssystems für die die Bedingungen $x_k \geq 0$, $k = 1, ..., n$ und $y_i \geq 0$, $i = 1, ..., m$ erfüllt sind.

Eine Basislösung erhält man, indem man jeweils n der Variablen $x_1, ..., x_n, y_1, ..., y_m$ auf Null setzt und das Gleichungssystem $(A|E) \begin{pmatrix} x \\ y \end{pmatrix} = b$ löst, sofern die entsprechenden m Spalten von $(A|E)$ linear unabhängig sind.

Definition 3.9 (Basislösung, Basisvariable)**.**
Eine Basislösung erhält man, indem man jeweils n der Variablen $x_1, ..., x_n, y_1, ..., y_m$ auf Null setzt und das Gleichungssystem $(A|E) \begin{pmatrix} x \\ y \end{pmatrix} = b$ löst, wobei die entsprechenden m Spalten von $(A|E)$ linear unabhängig sein müssen. Diejenigen Variablen, die dabei nicht Null gesetzt wurden, heißen Basisvariablen, weil sie in die jeweilige Basislösung eingehen. Die Variablen, die Null gesetzt wurden, heißen Nicht-Basisvariablen.

Algorithmus 3.10.
Naiv-rechnerische Herangehensweise an die Lösung eines Standardmodells.

1. *Stelle das Gleichungssytem $(A|E) \begin{pmatrix} x \\ y \end{pmatrix} = b$ auf.*
2. *Setze jeweils n Variablen gleich Null und löse das Gleichungssystem, sofern die entsprechenden verbleibenden m Spalten linear unabhängig sind. Jede dieser Lösungen ist eine Basislösung.*
3. *Überprüfe die Basislösungen auf Zulässigkeit. Das ist der Fall, wenn $x_k \geq 0$, $k = 1, ..., n$ und $y_i \geq 0$, $i = 1, ..., m$.*
4. *Setze die zulässigen Basislösungen in die Zielfunktion ein.*
5. *Bestimme die Basislösung, an der die Zielfunktion maximal wird.*

Algorithmus 3.10 bedeutet einen sehr hohen Rechenaufwand, wenn die Anzahl n der Strukturvariablen oder die Anzahl der Nebenbedingungen groß wird, weil $\begin{pmatrix} n+m \\ n \end{pmatrix}$ Basislösungen berechnet und auf Zulässigkeit überprüft werden müssen. Der Simplex-Algorithmus umgeht dieses Problem, indem er von vornherein nur zulässige Basislösungen berechnet und hierbei zielstrebig vorgeht, indem er den folgenden Satz beachtet:

Satz 3.11.
Ist eine zulässige Basislösung eines linearen Optimierungsproblems nicht optimal, so gibt es eine benachbarte Basislösung, in der die Zielfunktion einen besseren Wert annimmt.

Der Übergang zu einer benachbarten Basislösung geschieht im Simplex-Algorithmus dadurch, dass jeweils eine Basisvariable gegen eine Nicht-Basisvariable ausgetauscht wird. Dieser Austausch von Variablen geschieht durch Operationen, die schon aus dem Gauß-Verfahren bekannt sind.

Definition 3.12.
Das Ausgangstableau für den Simplex-Algorithmus erhält man aus Gleichung (3.1), indem man als letzte Zeile die Zielfunktionszeile anfügt, wobei die Koeffizienten c_k jeweils mit (-1) multipliziert werden. Da die Zielfunktion nicht von den Schlupfvariablen $y_1, ..., y_m$ abhängt, steht in den entsprechenden Spalten jeweils eine 0. In der ersten Spalte wird vermerkt, welche die aktuellen Basisvariablen sind.

aktuelle Basisvar.	x_1	x_2	$\dots$	x_n	y_1	y_2	$\dots$	y_m	*rechte Seite*
y_1	a_{11}	a_{12}	$\dots$	a_{1n}	1	0	$\dots$	0	b_1
y_2	a_{21}	a_{22}	$\dots$	a_{2n}	0	1	$\dots$	0	b_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
y_m	a_{m1}	a_{m2}	$\dots$	a_{mn}	0	0	$\dots$	1	b_m
	$\underbrace{-c_1}_{=:z_1}$	$\underbrace{-c_2}_{=:z_2}$	$\dots$	$\underbrace{-c_n}_{=:z_n}$	$\underbrace{0}_{=:z_{n+1}}$	$\underbrace{0}_{=:z_{n+2}}$	$\dots$	$\underbrace{0}_{=:z_{n+m}}$	0

Im Ausgangstableau sind die Basisvariablen durch die Schlupfvariablen $y_1, ..., y_m$ gegeben, daher sind alle Strukturvariablen Null und dies entspricht der Basislösung im Koordinatenursprung. Für diese Basislösung gilt $y_1 = b_1, ..., y_m = b_m$, d.h. diese Basislösung ist zulässig, falls $b_1, ..., b_m \geq 0$.

Algorithmus 3.13 (Der Simplex-Algorithmus)**.**
Voraussetzung:
$x_k \geq 0$ und $b_i \geq 0$ für alle $k = 1, \dots, n$ und $i = 1, \dots, m$, d.h. das Optimierungsproblem liegt als Standardmodell in Normalform vor.

1. *Stelle das Ausgangstableau gemäß Def. 3.12 auf.*
2. *Auswahl der Pivotspalte l: Wähle als Pivotspalte diejenige Spalte, die in der Zielfunktionszeile den betragsmäßig größten negativen Wert $z_l < 0$ aufweist.*
3. *Auswahl der Pivotzeile p:*
   (i) *Berechne für jede Zeile j den Wert $\frac{b_j}{a_{jl}}$, sofern $a_{jl} > 0$.*
   (ii) *Falls $a_{jl} \leq 0$ für alle $j = 1, \dots, m$, so existiert keine optimale Lösung, da x_l beliebig groß gewählt werden kann und hierdurch die Zielfunktion beliebig groß wird.*
   (iii) *Falls $a_{jl} > 0$ für mindestens eine Zeile j, so wähle unter diesen Zeilen diejenige aus ($\rightarrow$ Zeile p), für die der Quotient $\frac{b_j}{a_{jl}}$ minimal wird.*
4. *Austausch der Variablen:*
   (i) *Tausche diejenige Basisvariable, in deren Spalte der p-te Einheitsvektor $e_p \in \mathbb{R}^{m+1}$ steht, gegen die Nichtbasisvariable in der l-ten Spalte aus.*
   (ii) *Addiere zu jeder Zeile $j \neq p$ das $-\frac{a_{jl}}{a_{pl}}$-fache der p-ten Zeile.*
   (iii) *Addiere zur Zielfunktionszeile das $\underbrace{-\frac{z_l}{a_{pl}}}_{>0}$-fache der p-ten Zeile.*
   (iv) *Dividiere die p-te Zeile durch a_{pl}.*

Bemerkungen:

- *Für die neue "rechte Seite" b^{neu} gilt die Formel $b_j^{neu} = b_j^{alt} - \frac{a_{jl}}{a_{pl}} \cdot b_p^{alt}$. Somit wird durch 3.(iii) gewährleistet, dass $b_j^{neu} \geq 0$ gilt, denn die Wahl des minimalen Quotienten impliziert:*

$$b_j^{alt} - \frac{a_{jl}}{a_{pl}} \cdot b_p^{alt} = \underbrace{a_{jl}}_{>0} \cdot \Big(\underbrace{\frac{b_j^{alt}}{a_{jl}} - \frac{b_p^{alt}}{a_{pl}}}_{\geq 0} \Big) \geq 0$$

- *Durch die Aktionen 4.(ii)-4.(iv) wird in der l-ten Spalte mittels Zeilenumformung der Einheitsvektor $e_p \in \mathbb{R}^{m+1}$ erzeugt. Man zeigt induktiv, dass der Wert rechts unten im Tableau stets der Wert der Zielfunktion an der aktuellen Basislösung ist. Durch 4.(iii) wird die Zielfunktion in jedem Schritt vergrößert (es sei denn, das Optimalitätskriterium greift).*

5. *Überprüfung des Abbruchkriteriums:*
   Falls alle Werte der letzten Zeile nichtnegativ sind, lies den optimalen Wert der Zielfunktion rechts unten im Tableau ab. Eine optimale (Basis-)Lösung erhält man, indem man die aktuellen Nicht-Basisvariablen auf 0 und die aktuellen Basisvariablen auf die aktuellen $b_1, \ldots, b_m$ setzt (in der durch die aktuellen Basisspalten vorgegebenen Reihenfolge).
   Der Algorithmus ist beendet.
   Falls es in der letzten Zeile negative Werte gibt, so ist das Optimum noch nicht erreicht. Ersetze darum das alte Tableau durch das neue und durchlaufe den obigen Algorithmus ab 2.

Wenn das Standardmodell nicht in Normalform vorliegt, ist die Basislösung 0 nicht zulässig. In diesem Fall muss vor dem Simplex-Algorithmus eine Vorphase durchgeführt werden, in dem eine zulässige Basislösung gesucht wird.

Beispiel 3.14.
Eine Fabrik kann zwei Typen A und B eines Produkts unter folgenden Bedingungen herstellen:

Produkt	*Typ A*	*Typ B*	*maximal möglich*
Stück pro Tag	x_1	x_2	100 *Stück*
Arbeitszeit pro Stück	4	1	160 *Stunden*
Kosten pro Stück	20	10	1100 *EUR*
Gewinn pro Stück	120	40	

Wie müssen x_1 und x_2 gewählt werden, damit der Gewinn maximal wird? Dabei muss offenbar der lineare Ausdruck

$$z(x_1, x_2) := 120x_1 + 40x_2$$

zu einem Maximum gemacht werden unter den linearen Nebenbedingungen

$$\begin{array}{rcrcl} x_1 & + & x_2 & \leq & 100 \\ 4x_1 & + & x_2 & \leq & 160 \\ 20x_1 & + & 10x_2 & \leq & 1100 \end{array} \quad , \quad x_1 \geq 0, \quad x_2 \geq 0.$$

Lösung des Beispiels:

y_1	1	1	1	0	0	100
y_2	4	1	0	1	0	160
y_3	20	10	0	0	1	1100
	-120	-40	0	0	0	0

$120 > 40 \quad \Rightarrow \quad$ erste Spalte wird Pivotspalte

$$\frac{b_1}{a_{11}} = 100, \qquad \frac{b_2}{a_{21}} = 40, \qquad \frac{b_3}{a_{31}} = 55 \qquad : 40 \text{ minimal}$$

$\Rightarrow \quad a_{21} = 4 \quad$ Pivotelement

y_1	0	3/4	1	$-1/4$	0	60
x_1	1	1/4	0	1/4	0	40
y_3	0	5	0	-5	1	300
	0	-10	0	30	0	4800

zweite Spalte wird Pivotspalte (einziger negativer Eintrag unter den aktuellen c_j)

$$\frac{b_1}{a_{12}} = 80, \qquad \frac{b_2}{a_{22}} = 160, \qquad \frac{b_3}{a_{32}} = 60 \qquad : 60 \text{ minimal}$$

$\Rightarrow \quad a_{32} = 5 \quad$ Pivotelement

y_1	0	0	1	1/2	$-3/20$	15
x_1	1	0	0	1/2	$-1/20$	25
x_2	0	1	0	-1	1/5	60
	0	0	0	20	2	5400

alle aktuellen $c_j \geq 0 \quad \Rightarrow \quad$ Optimalität erreicht

Optimum: $x_1 = 25$, $x_2 = 60$
Gewinn: 5400 Euro

Kapitel 4

Fehleranalyse

4.1 Einleitung

In diesem Paragraphen beschäftigen wir uns mit Fehleruntersuchungen. An verschiedenen Stellen dieses Kurses hatten wir schon auf diesen Teil verwiesen und eine detaillierte Analyse der möglichen Fehlerquellen und deren Auswirkungen auf numerische Resultate angekündigt. Um Motivation für unser weiteres Vorgehen zu erhalten, betrachten wir zunächst ein bereits früher untersuchtes Beispiel.

In Kapitel 1 hatten wir das folgende *elektrische Netzwerk* untersucht.

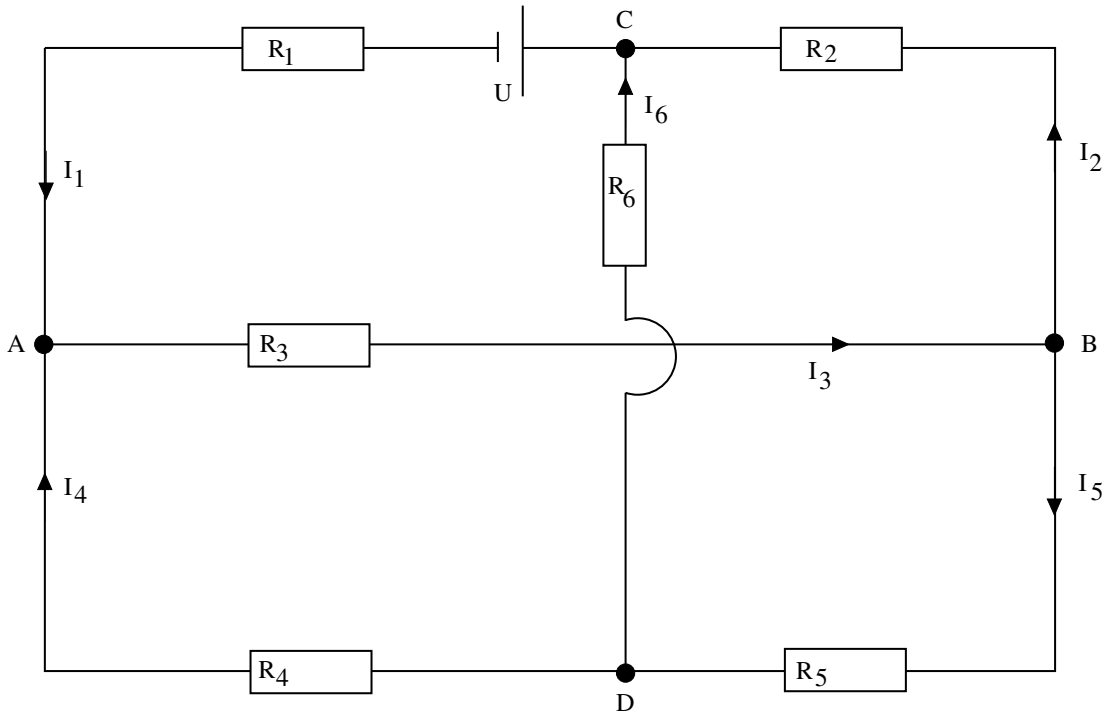


Abbildung 4.1: Schaltbild eines elektrischen Netzwerkes

Die in diesem Netzwerk fließenden Ströme I_ν, $\nu = 1, ..., 6$, lassen sich aus dem linearen

Gleichungssystem

$$\begin{bmatrix} 1 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 1 \\ R_1 & 0 & 0 & -R_4 & 0 & R_6 \\ 0 & R_2 & 0 & 0 & -R_5 & -R_6 \\ 0 & 0 & R_3 & R_4 & R_5 & 0 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ U \\ 0 \\ 0 \end{bmatrix}$$

berechnen.
Vom Standpunkt der Anwendungen her gesehen stellt sich folgendes Problem: Wenn man eine elektromotorische Kraft U misst und dann die Ströme I_ν, $\nu = 1, ..., 6$, aus diesem System berechnet, ist dann garantiert, dass $I_1, ..., I_6$ tatsächlich in der berechneten Stromstärke in dem Netzwerk fließen? Muss man nicht vielmehr damit rechnen, dass die berechneten Stromstärken nur näherungsweise mit den tatsächlich fließenden übereinstimmen? Diese Fragen sind in der Tat wohlbegründet, wie wir uns klar machen wollen.

- Fehler in den physikalischen Gesetzen:
  Mögliche Abweichungen können daher stammen, dass das verwendete physikalische Gesetz den Sachverhalt nicht genau wiedergibt. Denkbar ist, dass das Ohmsche Gesetz $U = R \cdot I$ für die verwendeten Widerstände $R_1, ..., R_6$ nicht exakt erfüllt ist. (Infolge einer geringen Temperaturabhängigkeit besitzen die Widerstände z.B. keine genau lineare Charakteristik.) Einflüsse dieser Art auf die gesuchten Ströme haben mit der numerischen Lösung des Problems offensichtlich nichts zu tun. Hier hat der Anwender zu entscheiden, ob das physikalische Modell, das er benützt, den wahren Sachverhalt genau genug beschreibt.

- Messungenauigkeit:
  Weitere Fehlerquellen resultieren aus Messungenauigkeiten. Die Widerstände $R_1, ..., R_6$ und die elektromotorische Kraft U lassen sich nur mit einer begrenzten Messgenauigkeit bestimmen. Eine Abschätzung (oder gar eine Reduzierung) dieser Messtoleranzen liegt ebenfalls nicht im Arbeitsbereich des Numerikers. In der Numerik kann man aber damit zusammenhängende Fragen klären, wieweit gegebene Toleranzen bei U oder den Widerständen R_ν, $\nu = 1, ..., 6$, die Ströme I_ν beeinflussen können. Zu untersuchen ist also die Abhängigkeit der Resultate von Eingangsdaten (Fehlerfortpflanzung).

- Rundungsfehler:
  Ein eigentlich numerisches Problem stellt sich bei der Auflösung dieses linearen Gleichungssystems, wenn man mit Dezimalzahlen einer festen Stellenlänge arbeitet. Dann ist stets damit zu rechnen, dass Zwischenresultate gerundet werden müssen. Es gilt dann den Einfluss von Rundungsfehlern abzuschätzen und, falls diese sich unangenehm akkumulierern sollten, eventuell eine günstigere Lösungsmethode einzusetzen.

- Schließlich sollte man bei starken Abweichungen der berechneten von den gemessenen Stromstärken immer auch an die Möglichkeit von Rechenfehlern denken.

In folgenden Abschnitt 4.2 untersuchen wir speziell die *Fehlerfortpflanzung bei linearen Gleichungssystemen*. Neben Fehlern der Komponenten der rechten Seite lassen wir auch Störungen der Matrix zu. Ist $N(\cdot)$ eine geeignete Matrixnorm (vgl. Abschnitt 4.2), so zeigt sich, dass beide Fehlertypen um den Faktor

$$\mathrm{cond}_N(A) = N(A)N(A^{-1})$$

verstärkt in das Resultat eingehen können. Dieser Faktor heisst *Konditionszahl* von A zur Matrixnorm N. In diesem Zusammenhang interessieren uns diejenigen Methoden zur Lösung linearer Gleichungssysteme, welche die Konditionszahl verkleinern oder wenigstens nicht vergrößern. Dabei erweist sich das QR-Verfahren bzgl. der Spektralkondition als numerisch günstig.

4.2 Fehlerfortpflanzung bei linearen Gleichungssystemen

Im Folgenden befassen wir uns speziell mit der Fehlerfortpflanzung bei linearen Gleichungssystemen. Gegeben sei also ein lineares Gleichungssystem

$$Ax = b,$$

wobei

$$A \in \mathbb{K}^{m\times m} \qquad (\mathbb{K} = \mathbb{R} \text{ oder } \mathbb{K} = \mathbb{C}), \qquad A \text{ regulär}, \qquad b \in \mathbb{K}^m.$$

Auf $\mathbb{R}^m$ ($\mathbb{C}^m$) sei $\|\cdot\|$ eine *Norm* gegeben und $N(\cdot)$ sei eine mit $\|\cdot\|$ verträgliche *Matrixnorm*, d.h.

$$\|Cy\| \le N(C)\|y\| \qquad \forall C \in \mathbb{K}^{m\times m} \text{ und } \forall y \in \mathbb{K}^m.$$

Meist notiert man $N(C) =: \|C\|$.

Nun sei $b + \triangle b$ eine Störung der rechte Seite des linearen Gleichungssystems und $x + \triangle x$ die Lösung von $A(x + \triangle x) = b + \triangle b$. Also gilt

$$A(\triangle x) = \triangle b \quad \text{ bzw. } \quad \triangle x = A^{-1}(\triangle b).$$

Daraus ergibt sich die folgende Abschätzung für den *absoluten* Fehler

$$\|\triangle x\| \le N(A^{-1}) \cdot \|\triangle b\|,$$

da N und $\|\cdot\|$ ein verträgliches Paar von Vektor- und Matrixnorm bilden.

Um den *relativen* Fehler abschätzen zu können, beachten wir

$$\|b\| = \|Ax\| \le N(A) \cdot \|x\|,$$

also, falls $b \neq 0$ und damit auch $x \neq 0$ gilt,

$$\frac{1}{\|x\|} \le \frac{N(A)}{\|b\|}.$$

Insgesamt folgt

$$\frac{\|\triangle x\|}{\|x\|} \leq N(A)N(A^{-1})\frac{\|\triangle b\|}{\|b\|}.$$

Als Verstärkungsfaktor beim relativen Fehler tritt also gerade die Größe

$$\mathrm{cond}_N(A) := N(A)N(A^{-1})$$

auf.

Bezeichnung 4.1 (Konditionszahl einer Matrix)**.**
Für eine reguläre Matrix A und eine Matrixnorm N bzeichnet man

$$\mathrm{cond}_N(A) := N(A)N(A^{-1})$$

als die zur Matrixnorm N gehörende Konditionszahl *von A.*

Das oben gewonnene Resultat über die Fortpflanzung von Datenfehlern halten wir fest.

Resultat 4.2 (Fehlerfortpflanzung bei Störung von b)**.** *Bezeichnet $\triangle b$ den Vektor der absoluten Datenfehler und $\triangle x$ den Vektor der absoluten Fehler des Resultatvektors, so gelten für ein lineares Gleichungssystem*

$$Ax = b, \qquad A \in \mathbb{K}^{m\times m} \text{ regulär}, \qquad b \in \mathbb{K}^m, \qquad b \neq 0,$$

die Abschätzungen

$$\begin{aligned} \|\triangle x\| &\leq N(A^{-1})\|\triangle b\|, \\ \frac{\|\triangle x\|}{\|x\|} &\leq \mathrm{cond}_N(A)\cdot\frac{\|\triangle b\|}{\|b\|}. \end{aligned}$$

Bemerkung 4.3.
Die Konditionszahl $\mathrm{cond}_N(A)$ einer regulären Matrix A spielt eine wichtige Rolle bei der Fehlerfortpflanzung in der numerischen linearen Algebra. Ein gewisser Nachteil ist allerdings, dass sie nur bei Kenntnis der inversen Matrix zugänglich ist.

Man kann leicht eine untere Schranke für $\mathrm{cond}_N(A)$ angeben, falls N submultiplikativ ist, d.h. falls gilt $N(AB) \leq N(A)N(B)$. In diesem Fall stellt man fest

$$N(E) = N(AA^{-1}) \leq N(A)N(A^{-1}) = \mathrm{cond}_N(A).$$

Aus $0 < N(E) \leq N(E)N(E)$ folgt $N(E) \geq 1$; also kann man die Abschätzung vergröbern zu

$$1 \leq \mathrm{cond}_N(A).$$

Die Konditionszahl $\mathrm{cond}_N(A)$ einer regulären Matrix A hängt nach Definition von der gewählten Matrixnorm N ab. Man kann nun die Frage stellen, ob sich bei gegebener Norm N die Konditionszahl durch den Übergang von A zu einer geeigneten Matrix TA verkleinern lässt. Dazu löse man zunächst folgende Aufgabe.

Aufgabe 4.4.
Durch die Wahl der Matrixnorm

$$\|C\|_2 := \sup_{y\neq 0} \frac{\|Cy\|_2}{\|y\|_2} = \sqrt{\lambda_{max}(C^H C)}, \qquad C \in \mathbb{K}^{m\times m}$$

ensteht als zugehörige Kondition die sogenannte Spektralkondition

$$\mathrm{cond}_2(A) := \|A\|_2 \cdot \|A^{-1}\|_2.$$

Für die so definierte Spektralkondition folgt: ist A eine reguläre Matrix und U eine unitäre Matrix so gilt

$$\mathrm{cond}_2(A) = \mathrm{cond}_2(UA) = \mathrm{cond}_2(AU).$$

Die Spektralkondition bleibt also beim Übergang von A zu UA oder AU invariant, wenn U unitär ist. Dies ist für numerische Zwecke günstig. So haben z.B. beim QR-Verfahren die Matrizen $A_k = Q_k R_k$ und R_k dieselbe Spektralkondition und, da die Matrizen A_k unitär ähnlich sind, haben die Matrizen A_k und R_k für alle k diesselbe Konditionszahl wie die Ausgangsmatrix A.

Wir kommen zurück zur Frage, ob man die Konditionszahl einer Matrix A durch Übergang zur Matrix TA verkleinern kann. Dazu definieren wir die Matrixnorm

$$\|C\|_\infty := \max_{y\neq 0} \frac{\|Cy\|_\infty}{\|y\|_\infty} = \max_i \sum_{k=1}^{m} |c_{ik}|$$

und die zugehörige Kondition

$$\mathrm{cond}_\infty(A) := \|A\|_\infty \cdot \|A^{-1}\|_\infty.$$

Man beachte, dass die Matrixnorm $\|\cdot\|_\infty$ zur Zeilensummennorm gehört. Motivation für unser weiteres Vorgehen liefert die folgende

Aufgabe 4.5.

1. *Für*

$$A := \begin{bmatrix} 1 & 1 & 1 \\ 1 & 10 & 100 \\ 1 & 100 & 10000 \end{bmatrix}$$

   berechne man $\mathrm{cond}_\infty(A)$.

2. *Man bestimme eine Diagonalmatrix $D := \mathrm{diag}(d_{11}, d_{22}, d_{33})$ so, dass die Betragssummen aller Zeilen von DA den Wert Eins haben, und berechne* $\mathrm{cond}_\infty(DA)$.

Das Resultat dieser Aufgabe zeigt, dass man die auf der Zeilensummennorm beruhende Konditionszahl $\mathrm{cond}_\infty(\cdot)$ eventuell dadurch erheblich verkleinern kann, dass man die Zeilen von A mit geeigneten Faktoren multipliziert. Dies entspricht einer Änderung des Maßstabs, in dem die Zeilenelemente gemessen werden. Man spricht deshalb auch von einer *Skalierung*. Teil 2 der obigen Aufgabe 4.5 lässt also folgendes vermuten.

Satz 4.6 (Günstige Skalierung von Matrizen)**.**
Gilt für eine reguläre Matrix $\tilde{A} \in \mathbb{K}^{m\times m}$ die Beziehungen

$$\sum_{k=1}^{m} |\tilde{a}_{ik}| = 1, \qquad i = 1, ..., m,$$

so folgt für jede reguläre Diagonalmatrix D die Ungleichung

$$\mathrm{cond}_\infty(D\tilde{A}) \geq \mathrm{cond}_\infty(\tilde{A}).$$

Daraus folgt für eine beliebige invertierbare Matrix $A \in \mathbb{R}^{m\times m}$ und eine invertierbare Diagonalmatrix $D \in \mathbb{R}^{m\times m}$:

$$\mathrm{cond}_\infty(DA) \geq \mathrm{cond}_\infty(D^*A)$$

mit

$$D^* = \begin{pmatrix} d_1^* & & & 0 \\ & d_2^* & & \\ & & \ddots & \\ 0 & & & d_m^* \end{pmatrix}, \qquad d_i^* = \frac{1}{\sum_{j=1}^m |a_{ij}|}.$$

Dies folgt, weil $\tilde{A} := D^*A$ gerade die im ersten Teil des Satzes verlangte Eigenschaft besitzt.

Beweis:
Wir setzen

$$\tilde{A}^{-1} =: (\alpha_{ik})_{i,k=1,...,m}, \qquad D := \mathrm{diag}(d_{11}, ..., d_{mm}).$$

Dann folgt

$$\begin{aligned} \|\tilde{A}^{-1}D^{-1}\|_\infty &= \max_{i=1,...,m} \sum_{k=1}^{m} |\alpha_{ik}| \cdot \frac{1}{|d_{kk}|} \\ &\geq \left(\max_{i=1,...,m} \sum_{k=1}^{m} |\alpha_{ik}| \right) \cdot \min_{k=1,...,m} \frac{1}{|d_{kk}|} \\ &= \|\tilde{A}^{-1}\|_\infty \cdot \min_{k=1,...,m} \frac{1}{|d_{kk}|} \end{aligned}$$

sowie

$$\|D\tilde{A}\|_\infty = \max_{i=1,..,m} |d_{ii}| \sum_{k=1}^{m} |\tilde{a}_{ik}| = \max_{i=1,...,m} |d_{ii}|.$$

Man erhält so

$$\begin{aligned} \mathrm{cond}_\infty(D\tilde{A}) &= \|D\tilde{A}\|_\infty \cdot \|\tilde{A}^{-1}D^{-1}\|_\infty \\ &\geq \|\tilde{A}^{-1}\|_\infty \cdot \min_{k=1,...,m} \frac{1}{|d_{kk}|} \cdot \max_{i=1,...,m} |d_{ii}|, \end{aligned}$$

woraus wegen

$$\min_{k=1,...,m} \frac{1}{|d_{kk}|} \cdot \max_{i=1,...,m} |d_{ii}| \geq 1 = \|\tilde{A}\|_\infty$$

die Behauptung

$$\mathrm{cond}_\infty(D\tilde{A}) \geq \mathrm{cond}_\infty(\tilde{A})$$

folgt. □

Damit haben wir gezeigt, dass sich bei einer regulären Matrix $\tilde{A}$, deren sämtliche Zeilenvektoren in der l_1-Norm die Länge Eins besitzen, die Kondition $\mathrm{cond}_\infty(\tilde{A})$ durch Skalierung nicht mehr verbessern lässt. Es empfiehlt sich also anstelle des Gleichungssystems

$$Ax = b \qquad \text{das System} \qquad D^*Ax = D^*b$$

zu lösen, wobei die Diagonalmatrix D^* so wie im obigen Satz 4.6 gewählt ist.

Neben Fehlern in den Daten b können auch „Störungen“ bei den Matrixelementen auftreten. Wir wollen jetzt den Fall betrachten, dass ein System

$$(A + \triangle A)\tilde{x} = b$$

gegeben ist. Dabei bezeichnet $\triangle A$ die Fehlermatrix. Gesucht ist eine Abschätzung für den absoluten Fehler

$$\triangle x := \tilde{x} - x,$$

wenn x die Lösung des „ungestörten“ Systems $Ax = b$ bezeichnet.

Wegen

$$\tilde{x} = (A + \triangle A)^{-1}b \qquad (\text{falls } A + \triangle A \text{ regulär}),$$

liegt es nahe, zunächst die „Störempfindlichkeit“ der Inversen zu bestimmen. Dazu dient folgende Aufgabe.

Aufgabe 4.7.
Es sei E die $m \times m$-Einheitsmatrix und $S \in \mathbb{K}^{m\times m}$ mit $N(S) < 1$ für eine submultiplikative Matrixnorm N. Man zeige:

1. *Die Matrix $E + S$ ist regulär.*
2. *Es gilt: $N((E + S)^{-1}) \leq \dfrac{N(E)}{1 - N(S)}$.*

Aus diesem Resultat erhalten wir für eine reguläre Matrix A im Fall

$$N(A^{-1}\triangle A) < 1,$$

dass die Matrix $E + A^{-1}\triangle A$ und damit auch

$$A + \triangle A = A(E + A^{-1}\triangle A)$$

regulär ist und dass die Abschätzung

$$N((A + \triangle A)^{-1}) = N((E + A^{-1}\triangle A)^{-1}A^{-1}) \leq \frac{N(E)}{1 - N(A^{-1}\triangle A)}N(A^{-1})$$

gilt. Falls auch

$$N(A^{-1})N(\triangle A) < 1$$

gilt, folgt hieraus

$$N((A+\triangle A)^{-1}) \leq \frac{N(E)}{1-N(A^{-1})N(\triangle A)} \cdot N(A^{-1}).$$

Auf diesem Weg erhalten wir schließlich

$$\begin{aligned} \frac{N((A+\triangle A)^{-1})}{N(A^{-1})} &\leq \frac{N(E)}{1-N(A^{-1})N(\triangle A)} \\ &= \frac{N(E)}{1-\mathrm{cond}_N(A)\cdot\dfrac{N(\triangle A)}{N(A)}}. \end{aligned}$$

Resultat 4.8 (Störempfindlichkeit der Matrixinversion)**.**
Für die reguläre Matrix A und eine zugehörige „Störmatrix“ $\triangle A$ gelte

$$N(A^{-1})N(\triangle A) < 1.$$

Dann existiert $(A+\triangle A)^{-1}$ und es folgt

$$\frac{N((A+\triangle A)^{-1})}{N(A^{-1})} \leq \frac{N(E)}{1-\mathrm{cond}_N(A)\cdot\dfrac{N(\triangle A)}{N(A)}}.$$

Jetzt können wir uns dem oben gestellten Problem zuwenden, eine Abschätzung für den Fehler $\triangle x$ zu konstruieren, wenn die reguläre Matrix A entsprechend

$$(A+\triangle A)(x+\triangle x) = b$$

mit Fehlern $\triangle A$ behaftet ist. Wegen $Ax = b$ folgt

$$(A+\triangle A)\triangle x = -\triangle A \cdot x.$$

Falls

$$N(A^{-1})N(\triangle A) < 1$$

gilt, existiert $(A+\triangle A)^{-1}$. Dann erhalten wir

$$\triangle x = -(A+\triangle A)^{-1}\triangle A \cdot x$$

und hieraus

$$\|\triangle x\| \leq N((A+\triangle A)^{-1})N(\triangle A)\|x\|, \qquad x \neq 0.$$

Mit Resultat 4.8 folgt für den relativen Fehler

$$\frac{\|\triangle x\|}{\|x\|} \leq N(E)\cdot\frac{\mathrm{cond}_N(A)}{1-\mathrm{cond}_N(A)\cdot\dfrac{N(\triangle A)}{N(A)}}\cdot\frac{N(\triangle A)}{N(A)}.$$

Resultat 4.9 (Abschätzung des relativen Fehlers von x)**.**
Gilt

$$N(A^{-1})N(\triangle A) < 1,$$

so lässt sich der relative Fehler der Lösung des gestörten linearen Gleichungssystems

$$(A + \triangle A)(x + \triangle x) = b, \qquad A \text{ regulär}, \qquad b \neq 0,$$

abschätzen in der Form

$$\frac{\|\triangle x\|}{\|x\|} \leq N(E) \cdot \frac{\text{cond}_N(A)}{1 - \text{cond}_N(A) \cdot \dfrac{N(\triangle A)}{N(A)}} \cdot \frac{N(\triangle A)}{N(A)}.$$

Den allgemeinen Fall, dass sowohl die Matrix A als auch die rechte Seite eines linearen Gleichungssystems mit Fehlern behaftet sind, behandeln wir in der folgenden Aufgabe.

Aufgabe 4.10.
Die Matrix A sei regulär und es gelte:

$$N(A^{-1})N(\triangle A) < 1.$$

Dann lässt sich der relative Fehler der Lösung des linearen Gleichungssystems

$$(A + \triangle A)(x + \triangle x) = b + \triangle b$$

abschätzen in der Form

$$\frac{\|\triangle x\|}{\|x\|} \leq N(E) \frac{\text{cond}_N(A)}{1 - \text{cond}_N(A) \cdot \dfrac{N(\triangle A)}{N(A)}} \left(\frac{\|\triangle b\|}{\|b\|} + \frac{N(\triangle A)}{N(A)} \right), \qquad x \neq 0 \neq b.$$

Zusammenfassung:

In diesem Abschnitt haben Sie Abschätzungen für relative und absolute Fehler kennengelernt. Die Abschätzung des relativen Fehlers von x in Abhängigkeit von der rechten Seite des Gleichungssystems $Ax = b$ führte auf die Konditionszahlen von Matrizen. An diesen Konditionszahlen erkannte man, dass Multiplikationen mit unitären Matrizen und bestimmte Skalierungen numerisch günstig sind.

Kapitel 5

Das Newton-Verfahren bei nichtlinearen Gleichungssystemen

Wir betrachten die Aufgabe, für eine gegebene Abbildung $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ ein $x^* \in D$ zu finden mit

$$F(x^*) = 0.$$

Sind $F_i : D \to \mathbb{R}$ $(i = 1, ..., n)$ die Komponentenabbildungen von F, und ist $x = (x_1, ..., x_n)^t$, so suchen wir die Lösung des i. A. nichtlinearen Gleichungssystems

$$\begin{aligned} F_1(x_1, ..., x_n) &= 0 \\ &\vdots \\ F_n(x_1, ..., x_n) &= 0. \end{aligned}$$

Wir betrachten zuerst den einfachen Fall $n = 1$. Falls F eine differenzierbare Funktion ist und $F' \neq 0$ gilt, so kann man mit dem *Newton-Verfahren* näherungsweise eine Lösung x^* von $F(x^*) = 0$ bestimmen: Dabei ist x^0 ein Startwert (ein ungefähre

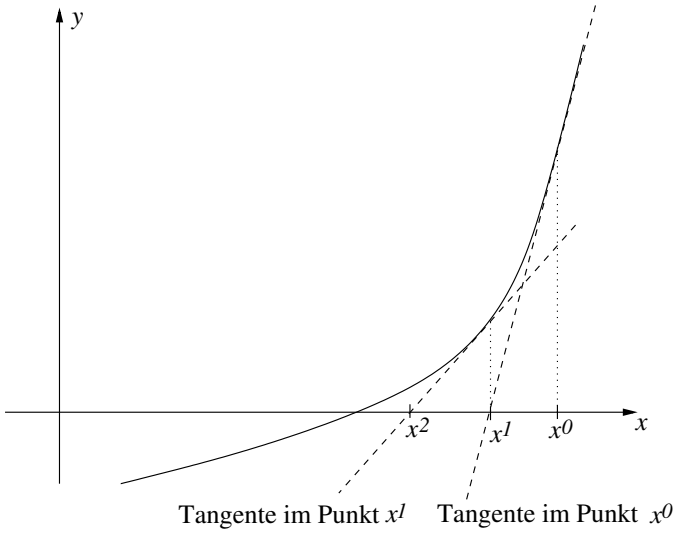


Abbildung 5.1: Das ein-dimensionale Newton-Verfahren

Schätzung) für die gesuchte Nullstelle x^*. Bestimmt man sukzessive x^1, x^2 usw., (vgl.

Abbildung 5.1) so erhält man das Newton-Verfahren

$$x^{k+1} = x^k - \frac{F(x^k)}{F'(x^k)}, \quad k = 0, 1, 2, \ldots.$$

Zu Gunsten einer einheitliche Schreibweise, die auch für den n-dimensionalen Fall geeignet ist, schreiben wir die Iterationsindizes als obere Indizes; eine Verwechslung mit Exponenten kann in diesem Abschnitt kaum vorkommen. Das Newton-Verfahren hat sehr günstige Eigenschaften, wie uns der nachfolgende Satz zeigt.

Satz 5.1. *Im Fall $n = 1$ des eindimensionalen Newton-Verfahrens*

$$x^{k+1} = x^k - \frac{F(x^k)}{F'(x^k)}, \quad k = 0, 1, 2, \ldots$$

liegt lokale quadratische Konvergenz vor, falls F in einer Umgebung der Nullstelle x^* zweifach stetig differenzierbar ist und dort $|F'(x)| \geq \alpha > 0$ für ein gewisses $\alpha > 0$ gilt. Genauer:
Sei $F(x^*) = 0$ und $F : [x^* - \delta_1, x^* + \delta_2] \longrightarrow \mathbb{R}$ sei zweifach stetig differenzierbar. Sind α, β so gewählt, dass $|F'(x)| \geq \alpha > 0$ und $|F''(x)| \leq \beta$ für alle $x \in [x^* - \delta_1, x^* + \delta_2]$ so gilt:

$$|x^* - x^{k+1}| \leq \frac{\beta}{2\alpha}|x^* - x^k|^2.$$

Falls zusätzlich $|x^* - x_0| < \frac{2\alpha}{\beta}$ gilt (d.h falls die Anfangsnäherung "gut genug" ist), so konvergiert x^k gegen x^* für $k \to \infty$.

Beweis:
Nach dem Satz von Taylor gilt:

$$0 = F(x^*) = F(x^k) + F'(x^k)(x^* - x^k) + \frac{1}{2}F''(\zeta)(x^* - x^k)^2 \tag{5.1}$$

mit einem Wert ζ, der zwischen x^k und x^* liegt. Nach der Vorschrift des Newton-Verfahrens gilt

$$0 = F(x^k) + F'(x^k)(x^{k+1} - x^k),$$

so dass man $F(x^k)$ in (5.1) ersetzen kann. Folglich ergibt sich

$$0 = F'(x^k)(x^* - x^{k+1}) + \frac{1}{2}F''(\zeta)(x^* - x^k)^2$$

bzw.

$$\begin{aligned} |x^* - x^{k+1}| =& |\frac{1}{F'(x^k)} \cdot \frac{1}{2}F''(\zeta)(x^* - x^k)^2| \\ \leq& \frac{\beta}{2\alpha}|x^* - x^k|^2. \end{aligned}$$

□

Nachfolgend wollen wir skizzieren, wie das Newton-Verfahren im n-dimensionalen Fall funktioniert. Unter $\|\cdot\|$ wollen wir eine beliebige Vektornorm auf $\mathbb{R}^n$ verstehen; tritt eine Matrixnorm $N(\cdot)$ auf, so wollen wir stets voraussetzen, dass diese Matrixnorm $N(\cdot)$ mit der Vektornorm $\|\cdot\|$ verträglich ist, d.h. dass gilt $\|Ax\| \leq N(A)\|x\|$ für $x \in \mathbb{R}^n$ und $A \in \mathbb{R}^{n\times n}$. Zunächst wollen wir noch an einige Begriffe der Analysis erinnern.

Definition 5.2 (Differenzierbarkeit)**.**
Die Abbildung $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ heißt differenzierbar (Fréchet-differenzierbar) im Punkt $\bar{x} \in int(D)$, wenn eine lineare Abbildung $A : \mathbb{R}^n \to \mathbb{R}^n$ existiert, so dass

$$\lim_{h \to 0} \frac{\|F(\bar{x}+h) - F(\bar{x}) - A \cdot h\|}{\|h\|} = 0$$

ist (dabei bedeutet $h \to 0$, dass alle Nullfolgen $\{h_m\} \subset \mathbb{R}^n$ zugelassen sind mit $\{\bar{x} + h_m\} \subset D$). Wir nennen A die Ableitung von F im Punkt $\bar{x}$

$$A = F'(\bar{x}).$$

Folgerung 5.3.
Ist F differenzierbar bei $\bar{x}$, so existieren alle partiellen Ableitungen $\partial F_i / \partial x_j$ $(i, j = 1, ..., n)$ an der Stelle $\bar{x}$. Die lineare Abbildung $F'(\bar{x}) : \mathbb{R}^n \to \mathbb{R}^n$ wird repräsentiert durch die $n \times n$-Matrix der partiellen Ableitungen (Jacobi-Matrix)

$$F'(\bar{x}) = \left[\frac{\partial F_i}{\partial x_j}(\bar{x}) \right]_{1 \le i,j \le n}.$$

Ist F differenzierbar für alle $x \in D$, so definiert

$$F' : \begin{cases} D & \to & \mathbb{R}^{n \times n}, \\ x & \mapsto & F'(x) \end{cases}$$

eine Abbildung von D in den Raum $\mathbb{R}^{n \times n}$ der quadratischen $n \times n$-Matrizen. Ist diese Abbildung stetig im Punkt $\bar{x} \in D$, so heißt F stetig differenzierbar in $\bar{x}$.

Unter der Annahme, dass $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ eine differenzierbare Abbildung ist und $F(x) = 0$ eine Lösung $x^* \in D$ besitzt, können wir das n-dimensionale Newton-Verfahren

$$x^0 \in D, \qquad x^{k+1} = x^k - F'(x^k)^{-1} F(x^k), \qquad k = 0, 1, ...,$$

bzw.

$$F'(x^k) y^k = -F(x^k), \qquad x^{k+1} := x^k + y^k$$

mit einem Startwert $x^0 \in D$ aufstellen, falls F in jedem Punkt von D (oder zumindest in einer Umgebung von x^*) eine invertierbare Jacobi-Matrix $F'(x)^{-1}$ besitzt.

Satz 5.4 (Lokale Konvergenz des Newton-Verfahrens (ohne Beweis))**.**
Die Abbildung $F : D \subset \mathbb{R}^n \longrightarrow \mathbb{R}^n$ besitze eine Nullstelle x^* mit $F(x^*) = 0$ und sei stetig differenzierbar bei x^*. Ferner sei $F'(x^*)$ invertierbar. In einer Umgebung $U(x^*)$ von x^* gelte ausserdem

$$N(F'(x) - F'(x^*)) \le \beta \|x - x^*\| \qquad \text{für } x \in U(x^*) \tag{5.2}$$

was z.B. erfüllt ist, wenn F zweimal stetig differenzierbar ist. Dann konvergiert die Newton-Iteration

$$x^{k+1} = x^k - F'(x^k)^{-1} F(x^k), \qquad k = 0, 1, ... \tag{5.3}$$

falls x^0 hinreichend nahe bei x^* liegt. Insbesondere existiert ein $C > 0$ und ein $K \in \mathbb{N}$ mit

$$\|x^{k+1} - x^*\| \le C \|x^k - x^*\|^2 \qquad \text{für } k \ge K \qquad \text{(quadratische Konvergenz).} \tag{5.4}$$

Bemerkung 5.5.
Der Fehler im k-ten Schritt ist $x^* - x^k$. Aufgrund der quadratischen Konvergenz verdoppelt sich die Anzahl korrekter Dezimalstellen mit jedem Schritt.

Kapitel 6

Quadratur

6.1 Einleitung

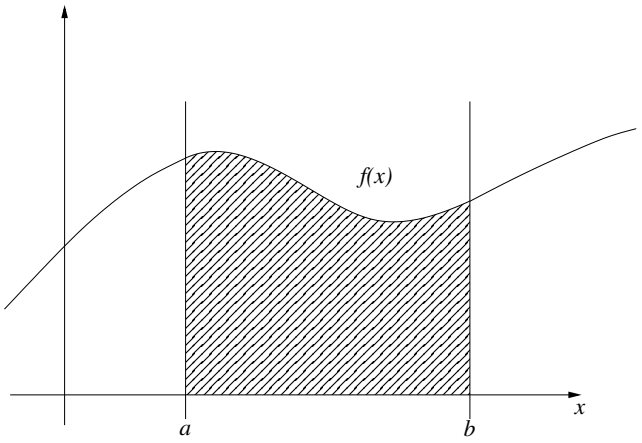
Der Problemkreis, der in diesem Kapitel im Mittelpunkt steht, hat eine lange Geschichte. Das Problem, von Kurven umschlossene Flächen zu berechnen, hat die Mathematiker schon seit dem Altertum beschäftigt und im 17. Jahrhundert zur Entwicklung der Integralrechnung geführt. Durch die Entwicklung fundierter Integralbegriffe wurde dann in der zweiten Hälfte des 19. Jahrhunderts das Flächenbestimmungsproblem vom theoretischen Standpunkt weitgehend abgeschlossen. Durch den *Hauptsatz der Differential- und Integralrechnung* lassen sich jeder im Intervall $[a, b]$ stetigen Funktion f Stammfunktionen F zuordnen mit

$$F' = f;$$

dabei gilt

$$I := \int_a^b f(x)dx = F(b) - F(a).$$

Vom Standpunkt der Praxis aus ist aber damit das Flächenbestimmungsproblem noch



keinesfalls befriedigend gelöst.

Dies hat mehrere Gründe:

1. Die Konstruktion des Integrals einer Funktion, wie sie beim Riemann-Integral durchgeführt wird, beinhaltet einen komplizierten Grenzprozess. Diesen Grenzprozess muss man in der Praxis durch einen finiten Näherungsprozess ersetzen. Darauf werden wir im nächsten Abschnitt genauer eingehen.

2. Für gewisse „einfache“ Funktionen kennt man die Stammfunktionen in geschlossener Form in dem Sinne, dass sie auf „einfache“ Weise aus „einfachen“ Funktionen aufbauen, z. B.

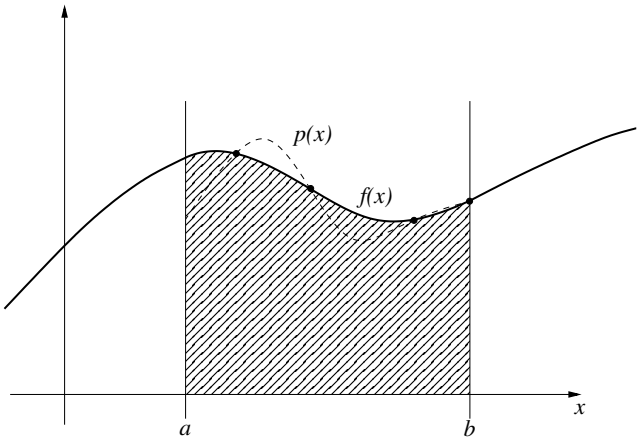
$$\begin{aligned}\int^x t^m dt &= \frac{1}{m+1}x^{m+1} + \text{const}, \qquad m \neq -1,\\ \int^x \frac{dt}{t} &= \log x + \text{const}, \qquad x > 0,\\ \int^x e^t dt &= e^x + \text{const}\,.\end{aligned}$$

   Es wurde aber schon von Liouville um 1835 gezeigt, dass z.B. für Funktionen des Typs

$$x \longmapsto \frac{1}{\sqrt{1+x^4}}, \qquad x \longmapsto \frac{e^x}{x}, \qquad x \longmapsto e^{-x^2}$$

   eine „geschlossene“ Darstellung der Stammfunktionen durch „einfache“ Funktionen unmöglich ist. Mit Hilfe von Stammfunktionen in geschlossener Form kommt man also in der Praxis nicht weit genug.

3. Kennt man schließlich den Integranden nur in einer endlichen Zahl von Punkten (z.B. an Messpunkten), so ist offensichtlich, dass man den Integranden f zunächst in geeigneter Weise (z.B. durch Interpolation) approximieren muss, um dann anschließend ersatzweise diese approximierende Funktion p zu integrieren.



Das Problem, Integrale numerisch auszuwerten, stellt sich in den Anwendungen häufig. Ein Grund dafür ist, dass sich die mechanische Arbeit als Wegintegral der Kraft ergibt:

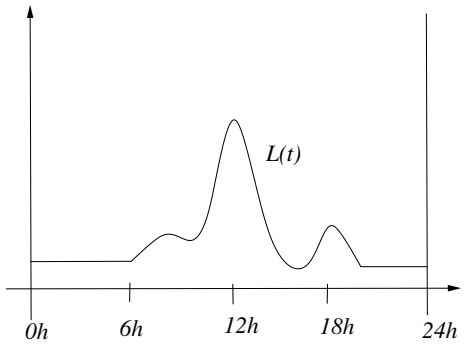
$$A = \int_{x_0}^{x_1} K(x)dx;$$

hier wird angenommen, dass die Kraft parallel zur Bewegungsrichtung wirkt. Ein anderer wichtiger Grund ist, dass die Gesamtenergiemenge das Zeitintegral der Leistung

ist:

$$E = \int_{t_0}^{t_1} L(t)dt.$$

Wir denken uns die in einer Wohnung währen eines Tages aufgenommene elektrische Leistung in einem Leistungsdiagramm aufgetragen.

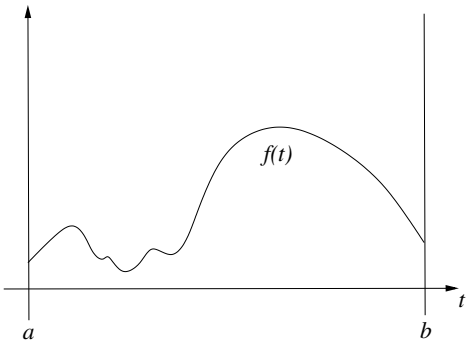


Dann beträgt die insgesamt zu bezahlende elektrische Energie

$$E = \int_{t_0}^{t_1} L(t)dt.$$

Tatsächlich wird aber dieses Leistungsdiagramm gar nicht aufgenommen. Vielmehr „integriert“ der Stromzähler die Leistung auf. (Der Stromzähler stellt also ein Analoggerät zur numerischen Integration der elektrischen Leistung dar.)

Ein ähnliches Problem stellt sich, wenn man den Durchschnittswert einer sich kontinuierlich ändernden Größe bestimmen will (z.B. mittlere Tages- oder Jahrestemperatur, mittlere Niederschlagsmenge in einem Monat).



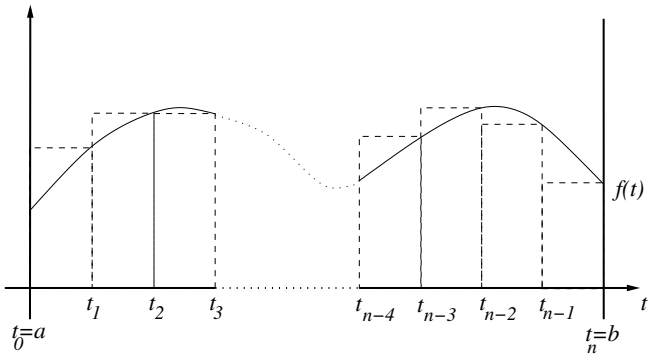
Dann ist der Mittelwert bekanntlich

$$m := \frac{\int_a^b f(t)dt}{b-a}.$$

In der Praxis ersetzt man die Integration durch n äquidistant verteilte Messungen und nimmt als genäherten Mittelwert

$$\overline{m} := h\sum_{i=1}^{n} f(t_i), \quad t_i = a + ih, \qquad i = 0, ..., n,$$

$$h = \frac{b-a}{n}.$$

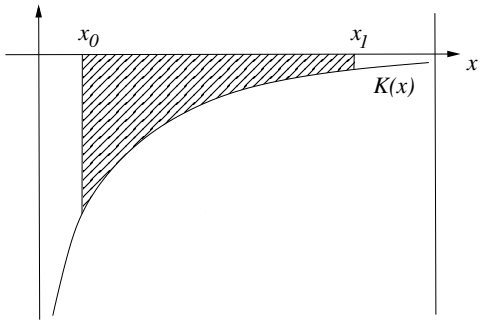


Der Integrand f wird also durch einen Treppenzug ersetzt und das Integral dieses Treppenzuges als Näherung für $\int_a^b f(t)dt$ verwendet. Dieses hier angewendete Prinzip zur numerischen Integration ist der Ausgangspunkt für die meisten Quadraturformeln. Man bezeichnet deshalb eine große Klasse von Quadraturformeln auch als *Mittelwertformeln.*

Als Beispiel zur mechanischen Arbeit betrachten wir den Flug einer senkrecht startenden Rakete. Wenn wir zunächst nur die Gravitation berücksichtigen, so gilt

$$K(x) = -\frac{mga^2}{x^2},$$

wenn wir mit

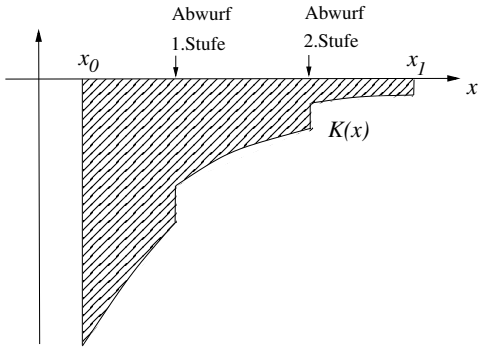


$$\begin{aligned} m &: \text{Raketenmasse,} \\ g &: \text{Erdbeschleunigung,} \\ a &: \text{Erdradius} \end{aligned}$$

bezeichnen. Also folgt mit $x_0 = a$ die Beziehung

$$A = -\int_a^{x_1} mga^2 \frac{dx}{x^2} = mga^2 \left[\frac{1}{x_1} - \frac{1}{a}\right].$$

Tatsächlich ist das Kraftgesetz wesentlich komplizierter, da sowohl die Masse variabel ist (Massenverlust durch Ausstoß der Verbrennungsgase, Mehrstufenrakete) und andererseits neben der Graviatation auch andere Kräfte zu berücksichtigen sind, die vom Luftwiderstand herrühren.



Am einfachsten lässt sich eine Funktion dieses komplizierten Typs mit Hilfe einer Qudraturformel näherungsweise integrieren.

Im Abschnitt 6.2 untersuchen wir exemplarisch die *Trapezregel* und die *Rechteckregel*. Beide Quadraturformeln sind geometrisch naheliegend, wenn man an den Zusammenhang des Integrals mit dem Flächenbegriff denkt. Für beide Formeln leiten wir Fehlerdarstellungen vom Cauchyschen und vom Peanoschen Typ her. Schließlich zeigen wir, dass man durch Zusammensetzen der einfachen Trapez- und der einfachen Rechteckregel zu Formeln höherer Genauigkeit gelangen kann. Für diese iterierten Formeln leiten wir ebenfalls Fehlerdarstellungen her.

Im Abschnitt 6.3 behandeln wir *interpolatorische Quadraturformeln*, die man am einfachsten durch Integration aus der Lagrange-Darstellung von Interpolationspolynomen erhält.

Im Abschnitt 6.4 behandeln wir die *Newton-Côtes-Formeln*, die interpolatorisch zu äquidistant verteilten Stützstellen definiert werden. Insbesondere untersuchen wir eingehend die *Simpsonsche Regel (Keplersche Fassregel)*, welche historisch gesehen die vielleicht wichtigste Quadraturformel überhaupt ist. Die Newton-Côtes-Formeln werden uns bei der numerischen Behandlung von Differentialgleichungen wieder begegnen.

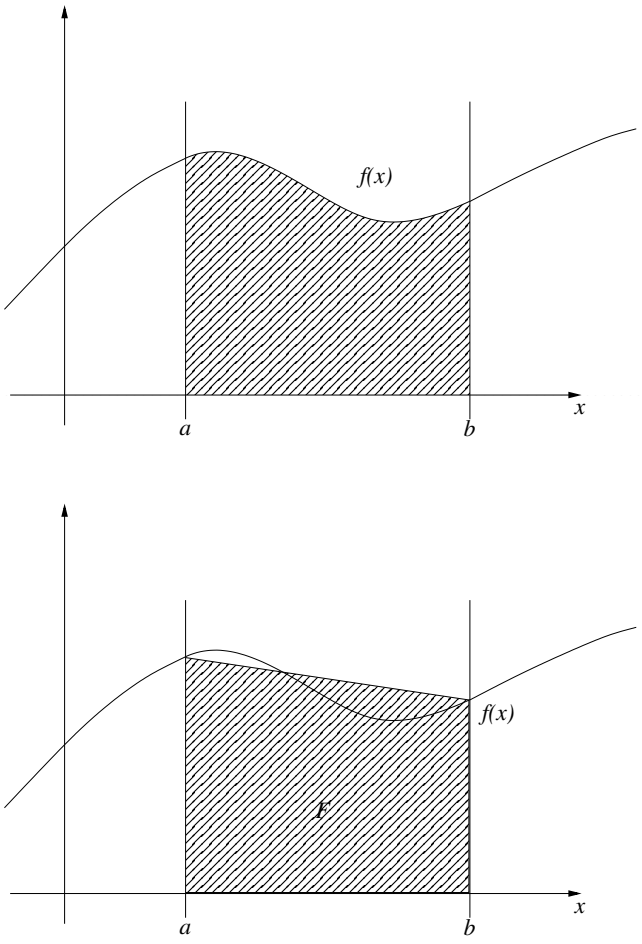
6.2 Die Trapezregel und die Rechteckregel

Um ein Integral

$$I := \int_a^b f(x)dx$$

näherungsweise zu berechnen, erinnern wir uns an die ursprüngliche Problemstellung aus dem 17.Jahrhundert, als man die Integralrechnung zu entwickeln begann. Damals sah man in erster Linie das Problem, Flächen zu berechnen, die vom Graphen der Funktion f, der x-Achse und zwei zur y-Achse parallelen Geraden berandet werden.

Die Anschauung legt es nahe, eine Näherung für I dadurch zu berechnen, dass man den Graphen von f durch die Sekante durch die Punkte $(a, f(a))$ und $(b, f(b))$ ersetzt. (Diese Idee spielt auch bei der Entwicklung des Riemannschen Integralbegriffs eine entscheidende Rolle.)



Die zu berechnende Fläche wird also durch ein Trapez mit der Fläche

$$F = (b-a)\frac{f(a)+f(b)}{2}$$

ersetzt. Wir erhalten so die *zweipunktige Trapezregel*

$$\int_a^b f(x)dx = (b-a)\frac{f(a)+f(b)}{2} + R(f),$$

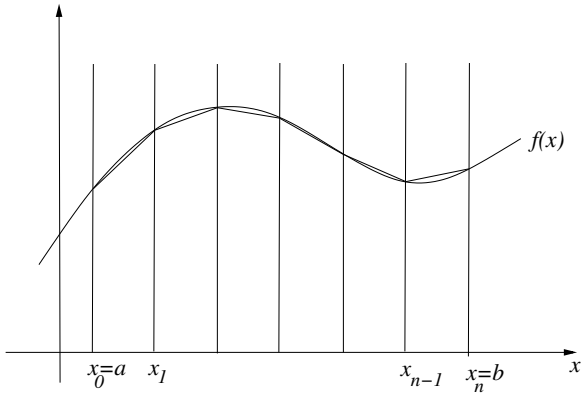
wobei $R(f)$ den auftretenden Fehler bezeichnet, der offensichtlich von der zu integrierenden Funktion f und der Intervalllänge $(b-a)$ abhängt.

Satz 6.1 (Fehler der Trapezregel)**.**
Ist f zweimal stetig differenzierbar auf $[a,b]$, so gilt die Fehlerdarstellung

$$R(f) = -\frac{(b-a)^3}{12}f''(\eta)$$

mit einem $\eta \in (a,b)$.

Die „einfache“ Trapezregel, wie wir sie bisher behandelt haben, wird meistens noch zu ungenaue Resultate liefern. Man teilt deshalb das Integrationsintervall in Teilintervalle gleicher Länge auf und wendet auf jedes die Trapezregel an.



Setzt man

$$\begin{aligned} h &:= \frac{b-a}{n}, \\ x_i &:= a+ih, \qquad i=0,...,n, \end{aligned}$$

so erhält man die *iterierte* oder *zusammengesetzte Trapezregel*

$$\begin{aligned} \int_a^b f(x)dx &= \frac{h}{2}\{f(a)+2f(a+h)+...+2f(b-h)+f(b)\}+R_h(f) \\ &= h\cdot{\sum_{i=0}^{n}}'' f(x_i)+R_h(f), \end{aligned}$$

wenn wir mit $\sum''$ eine Summation bezeichnen, bei der der erste und letzte Summand den Faktor $\frac{1}{2}$ erhalten.

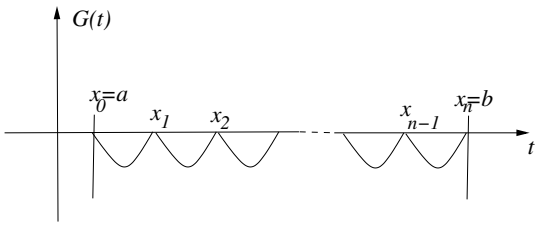
Aufgabe 6.2.
Man zeige:

1. *Für die iterierte Trapezregel gilt die Fehlerdarstellung vom Peano-Typ*

$$R_h(f)=\int_a^b f''(t)\cdot G(t)dt, \qquad f\in C^2[a,b],$$

wobei

$$G(t)=\begin{cases} \dfrac{(x_1-t)(x_0-t)}{2}, & t\in[x_0,x_1,] \\ G(t-ih), & t\in[x_i,x_{i+1}], \quad i=1,...,n-1. \end{cases}$$



2. Es gilt außerdem die Fehlerdarstellung vom Cauchyschen Typ

$$R_h(f) = -\frac{(b-a)}{12} \cdot h^2 \cdot f''(\eta), \qquad \eta \in [a,b].$$

Eine *Variante* der Trapezregel erhält man in der *Rechteckregel (Mittelpunktsregel)*

$$\int_a^b f(x)dx = (b-a) \cdot f(a + \frac{b-a}{2}) + \hat{R}(f).$$

Die zusammengesetzte Rechteck-Regel lautet dann: $h = \frac{b-a}{n}$, $x_i = a + ih$

$$\int_a^b f(x)dx = h \sum_{i=0}^{n-1} f(x_i + \frac{h}{2}).$$

6.3 Interpolatorische Quadraturformeln

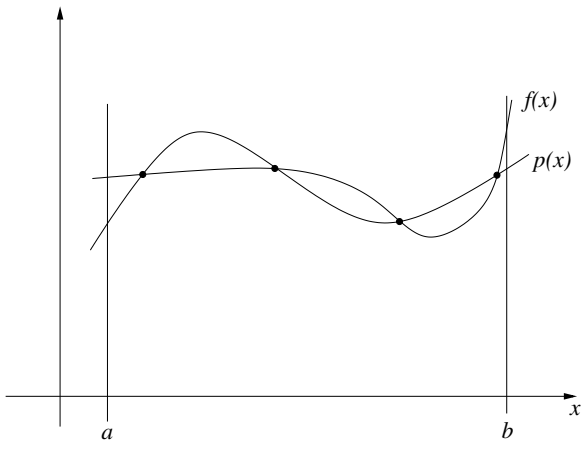
Bei der Konstruktion der Rechteckregel bzw. der Trapezregel hatten wir mit Polynomen nullten bzw. ersten Grades interpoliert. Dieses Konstruktionsprinzip lässt sich unter Verwendung von Polynomen höheren Grades weiterentwickeln.

Zur Berechnung von

$$I := \int_a^b f(x)dx$$

approximieren wir den Integranden f durch ein Interpolationspolynom p_n, wobei wir $n+1$ Stützstellen x_i, $i = 0, ..., n$, aus dem Intervall $[a,b]$ wählen. Es gelte

$$a \le x_0 < x_1 < ... < x_{n-1} < x_n \le b.$$



Das Interpolationspolynom p_n stellen wir in der Lagrange-Form

$$\begin{aligned} p_n(x) &= \sum_{\nu=0}^{n} f(x_\nu) l_\nu(x), \\ l_\nu(x) &= \prod_{\substack{\mu=0 \\ \mu \neq \nu}}^{n} \frac{x - x_\mu}{x_\nu - x_\mu}, \qquad \nu = 0, ..., n, \end{aligned}$$

dar. Offenbar gilt

$$l_\nu(x_\nu) = 1 \text{ und } l_\nu(x_j) = 0 \text{ für } \nu \neq j,$$

und daher erfüllt $p_n(x)$ auch die Interpolationsbedingung $p_n(x_j) = f(x_j)$ für $j = 0, \ldots, n$. Daher folgt

$$\begin{aligned} \int_a^b f(x)dx &= \int_a^b \sum_{\nu=0}^{n} f(x_\nu) l_\nu(x)dx + R_n(f) \\ &= \sum_{\nu=0}^{n} f(x_\nu) \int_a^b l_\nu(x)dx + R_n(f). \end{aligned}$$

Die *Gewichte*

$$a_\nu := \int_a^b l_\nu(x)dx$$

hängen nur von der Lage der Stützstellen x_ν, $\nu = 0, .., n$, nicht aber von f ab.

Bezeichnung 6.3 (Interpolationsquadraturformel)**.**
Zu einem festen Satz von Stützstellen

$$a \leq x_0 < x_1 < ... < x_{n-1} < x_n \leq b.$$

bezeichnet man die Formel

$$\begin{aligned} \int_a^b f(x) &= \sum_{\nu=0}^{n} a_\nu f(x_\nu) + R_n(f), \\ a_\nu &:= \int_a^b l_\nu(x)dx, \qquad \nu = 0, ..., n, \\ l_\nu(x) &= \prod_{\substack{\mu=0 \\ \mu\neq\nu}}^{n} \frac{x - x_\mu}{x_\nu - x_\mu}, \qquad \nu = 0, ..., n, \end{aligned}$$

als die zu diesen Stützstellen gehörende Interpolationsquadraturformel*.*

6.4 Newton-Côtes-Formel

Wählt man bei der Interpolationsformel 6.3 eine äquidistante Stützstellenverteilung, so erhält man die Newton-Côtes-Formeln.

Bezeichnung 6.4 (Newton-Côtes-Formel)**.**
Das Intervall $[a, b]$ werde durch äquidistante Stützstellen gemäß

$$x_i := a + ih, \qquad i = 0, ..., n, \qquad h := \frac{b-a}{n}, \qquad n \in \mathbb{N},$$

in n Teilintervalle zerlegt; dann bezeichnet man die Formel

$$\int_a^b f(x)dx = (b-a) \sum_{i=0}^{n} a_i^{(n)} f(x_i) + R_n(f)$$

mit den Gewichten

$$a_i^{(n)} := \frac{1}{b-a}\int_a^b l_i^{(n)}(x)dx = \frac{1}{b-a}\int_a^b \prod_{\substack{\mu=0\\ \mu\neq i}}^{n} \frac{x-x_\mu}{x_i-x_\mu}dx, \qquad i=0,...,n,$$

als Newton-Côtes-Formel.

Substituiert man mit $x = a + hs$ die Variable x durch s, erhält man für $i = 0, ..., n$

$$l_i^{(n)} = \prod_{\substack{\mu=0\\ \mu\neq i}}^{n} \frac{a+hs-a-\mu h}{a+ih-a-\mu h} = \prod_{\substack{\mu=0\\ \mu\neq i}}^{n} \frac{s-\mu}{i-\mu}$$

und

$$\begin{aligned} a_i^{(n)} &:= \frac{1}{b-a}\int_a^b l_i^{(n)}(x)dx = \frac{h}{b-a}\int_0^n \prod_{\substack{\mu=0\\ \mu\neq i}}^{n} \frac{s-\mu}{i-\mu}ds \\ &= \frac{(-1)^{n-i}}{n\cdot i!(n-i)!}\int_0^n s(s-1)\cdots(s-i+1)(s-i-1)\cdots(s-n)ds. \end{aligned}$$

Wir betrachten die Formel für gerades $n = 2m$, $m \in \mathbb{N}$, näher. Laut Konstruktion ist die Formel exakt für jedes Polynom $p_n \in \Pi_n$ (Π_n sei die Menge der Polynome vom Grad höchstens n), also $R_n(p_n) = 0$. Ist $p_{n+1} \in \Pi_{n+1}$ mit der Zerlegung

$$p_{n+1}(x) = \alpha\left(x - \frac{a+b}{2}\right)^{n+1} + p_n(x),$$

dann gilt

$$\begin{aligned} R_n(p_{n+1}) &= \int_a^b p_{n+1}(x)dx - (b-a)\sum_{i=0}^{n} a_i^{(n)} p_{n+1}(x_i) \\ &= \alpha\int_a^b \left(x-\frac{a+b}{2}\right)^{n+1}dx - (b-a)\sum_{i=0}^{n} a_i^{(n)}\cdot\alpha\cdot\left(x_i-\frac{a+b}{2}\right)^{n+1} + R_n(p_n). \end{aligned}$$

Da $a_i^{(n)} = a_{n-i}^{(n)}$, wie man für $i = 0, ..., n-1$ aus obiger Darstellung leicht verifiziert, und aufgrund der Punktsymmetrie von $x \longmapsto (x - \frac{a+b}{2})^{n+1}$ um $\frac{a+b}{2}$ für $n = 2m$, findet man

$$\begin{aligned} \alpha\int_a^b \left(x-\frac{a+b}{2}\right)^{n+1}dx &= 0, \\ \alpha(b-a)\sum_{i=0}^{n} a_i^{(n)}\left(x_i-\frac{a+b}{2}\right)^{n+1} &= 0; \end{aligned}$$

wegen $R_n(p_n) = 0$ folgt dann $R_n(p_{n+1}) = 0$.

Satz 6.5 (Exaktheit der Newton-Côtes-Formeln)**.**
Für $n \in \mathbb{N}$, n gerade, sind die Newton-Côtes-Formeln exakt für Polynome $p_{n+1} \in \Pi_{n+1}$ vom Grade $n+1$, also gilt $R_n(p_{n+1}) = 0$.

Für gerades n ist der Grad der Exaktheit also um 1 höher, als nach der Konstruktion zu erwarten war. Dies ist in der Symmetrie der Stützstellenverteilung um die Stelle $\frac{a+b}{2}$ begründet.

Für $n = 1$ erhält man wegen

$$\begin{aligned} a_0^{(1)} &= -\int_0^1 (s-1)ds = \frac{1}{2}, \\ a_1^{(1)} &= \int_0^1 s ds = \frac{1}{2} \end{aligned}$$

gerade die Trapezregel.

Für $n = 2$ ergibt sich

$$\begin{aligned} a_0^{(2)} &= \frac{1}{4}\int_0^2 (s-1)(s-2)ds = \frac{1}{4}\int_0^2 (s^2-3s+2)ds = \frac{1}{6}, \\ a_1^{(2)} &= -\frac{1}{2}\int_0^2 s(s-2)ds = -\frac{1}{2}\int_0^2 (s^2-2s)ds = \frac{4}{6}, \\ a_2^{(2)} &= \frac{1}{4}\int_0^2 s(s-1)ds = \frac{1}{4}\int_0^2 (s^2-s)ds = \frac{1}{6}. \end{aligned}$$

Die resultierende Formel

$$\int_a^b f(x)dx = \frac{b-a}{6}\left(f(a) + 4f(\frac{a+b}{2}) + f(b)\right) + R_2(f)$$

bezeichnet man als *Simpsonsche Regel* oder *Keplersche Fassregel*.

Wir erwähnen (ohne Beweis), dass der Fehler $R_2(f)$ der Simpsonsche Regel die folgende Cauchy-Darstellung hat:

$$R_2(f) = -\frac{1}{90}h^5 \cdot f^{(4)}(\eta) \qquad \text{für ein } \eta \in [a,b].$$

Für wachsendes n ergeben sich weitere Newton-Côtes-Formeln. Wir geben (mit $h = \frac{b-a}{n}$) ohne weiteren Beweis an:

Satz 6.6 (Newton-Côtes-Formeln und zugehörige Fehlerdarstellungen vom Cauchy–Typ)**.**

n	$a_0^{(n)}$	$a_1^{(n)}$	$a_2^{(n)}$	$a_3^{(n)}$	$a_4^{(n)}$	$a_5^{(n)}$	$a_6^{(n)}$	*Bezeichnung*	$R_n(f)$
1	$\frac{1}{2}$	$\frac{1}{2}$						*Trapezregel*	$-\frac{1}{12}h^3 f^{(2)}(\eta)$
2	$\frac{1}{6}$	$\frac{4}{6}$	$\frac{1}{6}$					*Simpsonregel*	$-\frac{1}{90}h^5 f^{(4)}(\eta)$
3	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$				*3/8-Regel*	$-\frac{3}{80}h^5 f^{(4)}(\eta)$
4	$\frac{7}{90}$	$\frac{32}{90}$	$\frac{12}{90}$	$\frac{32}{90}$	$\frac{7}{90}$			*Milne-Regel*	$-\frac{8}{945}h^7 f^{(6)}(\eta)$
5	$\frac{19}{288}$	$\frac{75}{288}$	$\frac{50}{288}$	$\frac{50}{288}$	$\frac{75}{288}$	$\frac{19}{288}$		*6-Punkt-Regel*	$-\frac{275}{12096}h^7 f^{(6)}(\eta)$
6	$\frac{41}{840}$	$\frac{216}{840}$	$\frac{27}{840}$	$\frac{272}{840}$	$\frac{27}{840}$	$\frac{216}{840}$	$\frac{41}{840}$	*Weddle-Regel*	$-\frac{9}{1400}h^9 f^{(8)}(\eta)$

Hierbei ist η jeweils ein von f und n abhängiger Zwischenwert. Die angegebene Ableitung von f soll jeweils existieren und stetig sein.

Für $n = 8$ und $n \geq 10$ treten negative Gewichte auf, wodurch sich große Betragssummen der Gewichte ergeben. Des Weiteren ist bei der numerischen Auswertung einer Quadraturformel mit Vorzeichenwechsel in den Gewichten eine starke Fehlerfortpflanzung wegen Auslöscheffekten zu erwarten. Man wendet daher auch die oben angegebenen Newton-Côtes-Formeln vorteilhaft in iterierter (zusammengesetzter) Form an, indem man also zunächst das Ausgangsintervall teilt und die Newton-Côtes-Formel auf die Teilintervalle ansetzt.
Gut erkennt man aus Satz 6.6 auch, dass man günstigerweise Formeln mit geradem n wählt, da man schon wegen Satz 6.5 die Exaktheit der nächsthöheren Formel mit $n+1$ erreicht.

Kapitel 7

Numerische Lösung von Anfangswertproblemen - Einschrittverfahren

7.1 Einleitung

In diesem und den noch folgenden Paragraphen stehen *Diskretisierungsverfahren bei Differentialgleichungen* im Vordergrund. Da sehr viele Probleme aus den Naturwissenschaften und der Technik auf Differentialgleichungen führen, ist die Entwicklung von effizienten Methoden zur Auflösung solcher Gleichungen eine Hauptaufgabe der Numerischen Mathematik. Zu diesem Problemkreis gibt es eine immense Zahl von Publikationen, die sowohl sehr allgemeine und für große Klassen von Differentialgleichungen anwendbare Methoden als auch sehr typabhängige Verfahren, die nur für sehr spezielle Differentialgleichungen angewendet werden können, behandeln. In diesem Kurs können wir dieses ausgedehnte Teilgebiet keinesfalls auch nur annähernd erschöpfend vorstellen.

Im folgenden gehen wir auf einige *Beispiele* aus den Anwendungsgebieten der Mathematik ein, die auf Differentialgleichungen führen. Ein *Wachstums-* oder *Zerfallsprozess* wird durch die Differentialgleichung

$$y' = \lambda y, \quad \lambda \in \mathbb{R}$$

beschrieben, welche gerade besagt, dass zu jedem Zeitpunkt t der Zuwachs (oder die Abnahme) $y'(t)$ proportional zur vorhandenen Substanz $y(t)$ ist. Hier ist bei Angabe einer Anfangssituation im Zeitpunkt t_0

$$y(t_0) = m_0$$

der Verlauf der Lösungsfunktion y vollständig bestimmt. Man erkennt leicht, dass

$$y(t) = m_0 e^{\lambda(t-t_0)}$$

die Lösung unseres Problems ist.

In den Anwendungen wird dieses einfache Wachstumsgesetz nur unter sehr idealisierten Annahmen gelten. In der Biologie wird man oft annehmen müssen, dass die Größe λ

selbst zeitabhängig ist (z.B. jahreszeitlich bedingt). Man hat dann

$$\begin{aligned} y'(t) &= \lambda(t) \cdot y \\ y(t_0) &= m_0 \end{aligned}$$

Hier ist nun

$$y(t) = m_0 e^{\Lambda(t)}$$

die Lösung, wenn Λ mit

$$\Lambda(t) = \int_{t_0}^{t} \lambda(\tau)\, d\tau$$

eine Stammfunktion zu λ bezeichnet. In diesem Fall führen also die früher behandelten Quadraturverfahren zu einer numerischen Lösung.

Während man das bisher behandelte Problem mit nur einer Substanz (Population) relativ leicht übersehen kann, wird es schon schwieriger, wenn mehrere Substanzen (verschiedene Arten) vorliegen, wobei gewisse Substanzen (Arten) auf Kosten der anderen entstehen können. Diese Problemstellung liegt z.B. bei chemischen Reaktionen, bei Zerfallsprozessen in der Atomphysik oder bei gemischten Populationen in der Biologie vor. Bezeichnet man die Menge der i-ten Substanz im Zeitpunkt t mit $y_i(t)$, $i = 1, \ldots, l$, so wird der Zerfallsprozess (die chemische Reaktion, die Zusammensetzung der Population) in seinem zeitlichen Verlauf durch ein System linearer Differentialgleichungen mit konstanten Koefffizienten

$$\begin{aligned} y_1' &= \sum_{k=1}^{l} \lambda_{1k} y_k \\ &\vdots \qquad\qquad\qquad\qquad , \lambda_{ik} \in \mathbb{R},\ i,k = 1, \ldots, l \\ y_l' &= \sum_{k=1}^{l} \lambda_{lk} y_k \end{aligned}$$

beschrieben, wobei außerdem l Anfangsbedingungen

$$y_k(t_0) = v_k, \quad k = 1, \ldots, l\,,$$

erfüllt werden müssen. Während man den Fall zeitunabhängiger Parameter λ_{ik}, $i,k = 1, \ldots, l$, völlig beherrscht, indem man das gegebene System von Differentialgleichungen mit Hilfe einer Hauptvektorbasis der Matrix (λ_{ik}) "entkoppelt", bereitet der Fall, dass die Parameter zeitabhängig sind, erhebliche Schwierigkeiten. Bei der numerischen Lösung ist insbesondere der Fall, dass die Matrix $(\lambda_{ik}(t))$ für gewisse Werte von t Eigenwerte sehr unterschiedlichen Betrags besitzt (*steife Differentialgleichung*), mit erheblichen Komplikationen verbunden.

Die *Kinetik* führt ebenfalls bei vielen Problemen auf Differentialgleichungen. Der einfachste Typ ist die Bewegungsgleichung eines Schwingungsvorgangs

$$\begin{aligned} my'' + Ry' + ky &= f(t)\,, \\ y(t_0) &= y_0\,, \\ y'(t_0) &= y_0^{(1)}\,. \end{aligned}$$

(Hier bedeutet m die Masse, R den Reibungskoeffizienten, k die Rückstellkraft, f die schwingungserzeugende Kraft.) Während man diesen Typ ebenfalls völlig beherrscht, bereitet die Berechnung von Bahnkurven mehrerer Massen, die gegenseitig Gravitationskräfte aufeinander ausüben, erhebliche theoretische und numerische Schwierigkeiten.

Eine Problemstellung ähnlichen Typs liefert die Raumfahrt. Wir betrachten "nur" den Spezialfall einer periodischen Satellitenbahn im Erde-Mond-System: Unter idealisierten Annahmen (*restringiertes Dreikörperproblem*) wird die Bahnkurve in der Bewegungsebene durch ein System von zwei Differentialgleichungen zweiter Ordnung beschrieben. Bezeichnet

$$\mu = \frac{m}{M} = \frac{1}{82,45}$$

das Verhältnis der Mondmasse m zur Erdmasse M, so gilt

$$\begin{aligned} y'' &= y + 2z' - (1-\mu)\frac{y+\mu}{((y+\mu)^2+z^2)^{3/2}} - \mu\frac{y-(1-\mu)}{((y-1+\mu)^2+z^2)^{3/2}} \\ z'' &= z - 2y' - (1-\mu)\frac{z}{((y+\mu)^2+z^2)^{3/2}} - \mu\frac{z}{((y-1+\mu)^2+z^2)^{3/2}} \,. \end{aligned}$$

Mit den Anfangsbedingungen

$$\begin{aligned} y(0) &= 1,2\,, & y'(0) &= 0 \\ z(0) &= 0\,, & z'(0) &= -1,049357\ldots \end{aligned}$$

hat die Bahnkurve $(y(t), z(t))$ den in Abbildung 7.1 dargestellten Verlauf (vgl. E. Fehlberg, zur numerischen Integration von Differentialgleichungen durch Potenzreihenansätze, dargestellt an Hand physikalischer Beispiele, ZAKM 44, 83-88 (1964)). Dabei wird angenommen, dass das Koordinatensystem sich mit der Erde "mitbewegt"; die Erde liegt also stets im Ursprung des Koordinatensystems.

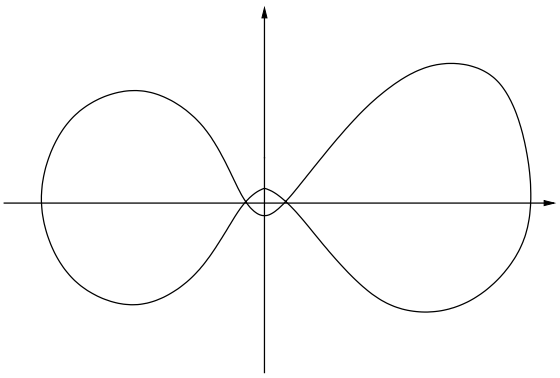


Abbildung 7.1: Abbildung der Bahnkurve des Erde-Mond-Systems

Bei diesen und ähnlich gelagerten Beispielen ist die Auswahl des richtigen Verfahrens von eminenter Bedeutung. Dieses muss sowohl in der Schrittweite als auch in der Ordnung dem Problem angepasst werden. Besonders wichtig ist die Wahl der richtigen Schrittweite. Diese darf relativ groß in dem Bereich gewählt werden, wo die Bahn nur wenig gekrümmt ist, während sie in Erdnähe sehr viel kleiner sein muss. Eine konstante Schrittweite müsste sich am Verhalten der Bahn in Erdnähe orientieren und würde dadurch viel zu klein ausfallen. Der resultierende Rechenaufwand wäre ein Mehrfaches

dessen, was bei variabler Schrittweite ausreicht. Um so erstaunlicher sind aber dann angesichts dieser Problematik die Erfolge der Raumfahrtszentren, in denen derzeit wohl Tausende von Satellitenbahnen rechnerisch verfolgt werden.

Im Abschnitt 7.2 stellen wir einige *Resultate aus der Lösungstheorie für Anfangswertprobleme bei gewöhnlichen Differentialgleichungen* zusammen. Wir definieren zunächst die Problemstellung und erklären, was man unter einer Lösung eines Anfangswertproblems zu verstehen hat. Dann erwähnen wir den Existenz- und Eindeutigkeitssatz von Picard-Lindelöf für Differentialgleichungen, deren rechte Seite einer Lipschitz-Bedingung genügt. Schließlich weisen wir darauf hin, dass jede Lösung, deren Existenz durch den Satz von Picard-Lindelöf zunächst nur in einer kleinen Umgebung des Anfangspunktes gesichert ist, "von Rand zu Rand" des betrachteten Gebietes fortgesetzt werden kann.

Im Abschnitt 7.3 behandeln wir exemplarisch das *Eulersche Polygonzugverfahren*, das als Grundtyp aller Einschrittverfahren angesehen werden kann. Wir motivieren dieses Lösungsverfahren geometrisch mit Hilfe des Richtungsfeldes und analytisch mit Hilfe numerischer Differentiation und numerischer Integration sowie mit Hilfe von Taylorentwicklung. Diese Ansätze liegen auch einer Reihe anderer im folgenden zu behandelnden Methoden zugrunde. Für spezielle Beispiele untersuchen wir die Konvergenz des Polygonzugverfahrens bei kleiner werdender Schrittweite. Schließlich weisen wir noch darauf hin, dass Systeme von Differentialgleichungen erster Ordnung komponentenweise behandelt und Differentialgleichungen höherer Ordnung auf Systeme erster Ordnung zurückgeführt werden können.

Im Abschnitt 7.4 stehen *Definition und grundlegende Eigenschaften von Einschrittverfahren* im Mittelpunkt. Wir definieren zunächst, was wir unter einer Zuwachsfunktion verstehen wollen und erhalten damit die Rekursion eines allgemeinen Einschrittverfahrens. Anschließend führen wir den Begriff des lokalen Diskretisierungsfehlers ein, der es uns dann ermöglicht, den Konsistenzbegriff und die Konsistenzordnung eines Einschrittverfahrens zu definieren.

Im Abschnitt 7.5 lernen wir die *Methode des Taylorabgleichs* kennen. Diese Vorgehensweise ermöglicht es im Prinzip, für genügend oft differenzierbare Funktionen f Einschrittverfahren vorgeschriebener Konsistenzordnung zu konstruieren. Da aber in die Zuwachsfunktion partielle Ableitungen von f entsprechend hoher Ordnung eingehen, hat diese Methode nur einen beschränkten Anwendungsbereich, der entscheidend davon abhängt, ob die erforderlichen partiellen Ableitungen in "einfacher" Weise berechnet werden können.

Im Abschnitt 7.6 behandeln wir exemplarisch die *Runge-Kutta-Verfahren der Konsistenzordnung 2*. Das Prinzip der Runge-Kutta-Verfahren besteht darin, das Auftreten von Ableitungen von f durch einen Mittelungsprozess von Funktionswerten von f an geeignet gewählten Stützstellen zu vermeiden. Wir zeigen durch Taylorentwicklung, dass eine einparametrige Schar von Runge-Kutta-Verfahren der Konsistenzordnung 2 existiert.

Im Abschnitt 7.7 gehen wir auf *allgemeine Runge-Kutta-Verfahren* ein. Wir definieren zunächst, was wir unter einem R-stufigen Runge-Kutta-Verfahren verstehen wollen. Anschließend sondern wir einige praktisch wichtige Beispiele, insbesondere das "klassische" Runge-Kutta-Verfahren, aus. Wie mühsam die Konstruktion von höherstufigen Runge-Kutta-Verfahren mit entsprechend hoher Konsistenzordnung ist, zeigen wir im Fall der

dreistufigen Runge-Kutta-Verfahren der Konsistenzordnung 3. Durch Taylor-Abgleich erhalten wir nach langwieriger Rechnung eine zweiparametrige Schar solcher Verfahren. Zum Schluss gehen wir auf die erreichbare Ordnung bei höherstufigen Verfahren ein und zitieren in diesem Zusammenhang ein tiefliegendes Resultat von Butcher.

Im Abschnitt 7.8 befassen wir uns mit der *Konvergenz von Einschrittverfahren.* Zum Schluss dieses Abschnitts gehen wir auf eine Fehlerabschätzung ein und zeigen, dass die Konsistenzordnung des Einschrittverfahrens die Ordnung des akkumulierten Fehlers bestimmt.

Im Abschnitt 7.9 befassen wir uns mit der *Praxis der Einschrittverfahren.* Dabei steht die Wahl der Schrittweite im Vordergrund.

7.2 Einführung in die Lösungstheorie von Anfangswertproblemen

Anfangswertproblem

Die Differentialgleichung

$$y' = f(x, y)$$

zusammen mit dem Anfangswert

$$y(x_0) = y_0$$

nennt man ein Anfangswertproblem, falls f in einem Gebiet $G \subset \mathbb{R}^2$ stetig ist und (x_0, y_0) in G liegt. In vielen Fällen wird G als Streifen $(\alpha, \beta) \times \mathbb{R}$ vorausgesetzt.

Lösung eines Anfangswertproblems

Eine reellwertige Funktion y heißt Lösung des Anfangswertproblems, falls y auf einem Intervall (a, b) mit $x_0 \in (a, b)$ definiert und stetig differenzierbar ist und ferner gilt:

(1) $(x, y(x)) \in G$ für $x \in (a, b)$

(2) $y'(x) = f(x, y(x))$ für $x \in (a, b)$

(3) $y(x_0) = y_0$.

Lipschitz-stetige Funktionen

Die rechte Seite f der Differentialgleichung heißt Lipschitz-stetig auf G, falls eine Konstante (Lipschitz-Konstante) L existiert derart, dass für alle Punkte $(x, y), (x, \tilde{y}) \in G$ die Bedingung (Lipschitz-Bedingung)

$$|f(x, y) - f(x, \tilde{y})| \leq L|y - \tilde{y}|$$

erfüllt ist.

Integralgleichung vom Volterra-Typ

Das Auffinden einer Lösung des Anfangswertproblems ist äquivalent zum Auffinden einer Lösung der Integralgleichung

$$y(x) = y_0 + \int_{x_0}^{x} f(t, y(t))\, dt.$$

Diese Gleichung heißt Integralgleichung vom Volterraschen Typ.

Lösbarkeit von Anfangswertproblemen

Nach dem Satz von Peano besitzt das Anfangswertproblem (bei stetiger rechter Seite) stets wenigstens eine Lösung. Ist f Lipschitz-stetig auf G, so ist die Lösung nach dem Satz von Picard-Lindelöf lokal eindeutig bestimmt; es existiert in diesem Fall eine eindeutige maximale Lösung, die für wachsendes und fallendes Argument gegen den Rand des Gebiets G läuft.

Picard-Iteration

Bei Lipschitz-stetiger rechter Seite kann die Lösung des Anfangswertproblems lokal durch die sogenannte Picard-Iteration

$$\begin{aligned} y_0(x) &= y_0\,, \\ y_\nu(x) &= y_0 + \int_{x_0}^{x} f(t, y_{\nu-1}(t))\, dt\,, \quad \nu \geq 1\,, \end{aligned}$$

bestimmt werden; die Folge $(y_\nu)_{\nu \geq 0}$ konvergiert lokal gleichmäßig gegen die Lösung.

7.3 Das Eulersche Polygonzugverfahren

Richtungsfeld, Isoklinen

Einen Überblick über die Lösungsgesamtheit einer Differentialgleichung

$$y' = f(x, y)$$

(mit in einem Gebiet $G \subset \mathbb{R}^2$ stetiger rechter Seite) kann man sich verschaffen, indem man das Richtungsfeld zeichnet. Hierzu markiert man in jedem Punkt $(x, y) \in G$ die Steigung

$$p := f(x, y)\,.$$

Die durch

$$f(x, y) = \text{const}$$

definierten Kurven nennt man Kurven gleicher Neigung oder Isolklinen.

Eulersches Polygonzugverfahren

Diese Verfahren erhält man dadurch, dass man stückweise in der durch das Richtungsfeld vorgeschriebenen Richtung weiterläuft: Man gibt sich eine Schrittweite $h > 0$ vor, setzt

$$\begin{aligned} x_\nu &:= x_0 + \nu \cdot h\,, \qquad \nu \geq 0\,, \\ y_0 &:= y(x_0), \end{aligned}$$

und iteriert gemäß

$$y_\nu := y_{\nu-1} + h \cdot f(x_{\nu-1}, y_{\nu-1})\,, \qquad \nu \geq 1\,,$$

solange die rechte Seite definiert ist.

Eulersches Polygonzugverfahren für ein System erster Ordnung

Liegt ein System erster Ordnung

$$\begin{aligned} y_1' &= f_1(x, y_1, \ldots, y_m)\,, \\ y_2' &= f_2(x, y_1, \ldots, y_m)\,, \\ &\vdots \\ y_m' &= f_m(x, y_1, \ldots, y_m) \end{aligned}$$

vor, so iteriert man komponentenweise. Dies ergibt die Rekursion

$$\begin{aligned} y_1^{[\nu]} &= y_1^{[\nu-1]} + h \cdot f_1(x_{\nu-1}, y_1^{[\nu-1]}, \ldots, y_m^{[\nu-1]})\,, \\ &\vdots \\ y_m^{[\nu]} &= y_m^{[\nu-1]} + h \cdot f_m(x_{\nu-1}, y_1^{[\nu-1]}, \ldots, y_m^{[\nu-1]})\,. \end{aligned}$$

Eulersches Polygonzugverfahren für eine Differentialgleichung m-ter Ordnung

Eine Differentialgleichung m-ter Ordnung

$$y^{(m)} = f(x, y, y', \ldots, y^{(m-1)})$$

wandelt man um in ein System von m Differentialgleichungen erster Ordnung

$$\begin{aligned} y_0' &= y_1\,, \\ y_1' &= y_2\,, \\ &\vdots \\ y_{m-2}' &= y_{m-1}\,, \\ y_{m-1}' &= f(x, y_0, \ldots, y_{m-1}) \end{aligned}$$

und wendet darauf das Eulersche Polygonzugverfahren an.

7.4 Einschrittverfahren - Definition und grundlegende Eigenschaften

Allgemeines Einschrittverfahren

Einen Algorithmus

$$\begin{aligned} y_0 &= y(x_0)\,, \\ x_\nu &= x_{\nu-1} + \nu \cdot h\,, \\ y_\nu &= y_{\nu-1} + h \cdot \Phi(x_{\nu-1}, y_{\nu-1}, h)\,, \end{aligned}$$

mit einer auf $G \times [0, \alpha]$ definierten, reellwertigen Funktion Φ heißt allgemeines Einschrittverfahren für das auf dem Gebiet $G \subset \mathbb{R}^2$ definierte Anfangswertproblem

$$y' = f(x, y), \quad y(x_0) = y_0.$$

Man nennt Φ die Zuwachs-, Inkrement- oder Verfahrensfunktion.

Lokaler Diskretisierungsfehler

Ist $z : (a, b) \to \mathbb{R}$ die Lösung des Anfangswertproblems

$$z'(t) = f(t, z(t)), \quad z(x) = y$$

mit einer auf G (bezüglich der zweiten Variablen) Lipschitz-stetigen Funktion f, so heißt die Funktion

$$\Delta(x, y, h) := \begin{cases} \frac{z(x+h)-y}{h}, & \text{für } h \neq 0 \\ f(x, y), & \text{für } h = 0 \end{cases}$$

exakter relativer Zuwachs der Lösung z. Ferner bezeichnet man mit

$$\theta(x, y, h) = \Delta(x, y, h) - \Phi(x, y, h)$$

den *lokalen Diskretisierungsfehler* des durch Φ definierten Einschrittverfahrens an der Stelle $(x, y) \in G$ bei Schrittweite h.

Konsistenzordnung

Ein Einschrittverfahren hat die Konsistenzordnung p, falls für den lokalen Diskretisierungsfehler gilt

$$\theta(x, y, h) = O(h^p) \quad \text{für } h \to 0$$

für alle $(x, y) \in G$ und alle f, deren partielle Ableitungen bis zur p-ten Ordnung in G existieren und dort stetig sind.

Konsistenz

Einschrittverfahren der Konsistenzordnung $p \geq 1$ nennt man konsistent. Bei stetigen Zuwachsfunktionen impliziert die Konsistenz die Forderung

$$\Phi(x, y, 0) = f(x, y)\,.$$

Konsistenz des Eulerschen Polygonzugverfahrens

Das Eulersche Polygonzugverfahren hat die Konsistenzordung 1.

7.5 Die Methode des Taylorabgleichs

Verfahren der Konsistenzordnung p durch Taylorabgleich

Wählt man, falls f p-mal stetig differenzierbar ist, als Verfahrensfunktion Φ das Taylor-Polynom $(p-1)$-ten Grades der exakten Zuwachsfunktion Δ, entwickelt nach Potenzen von h, so erhält man dadurch ein Einschrittverfahren der Konsistenzordnung p. Insbesondere erhält man durch

$$\Phi(x,y,h) = \underbrace{f(x,y)}_{y'(x)} \tag{1}$$

ein Verfahren erster Ordnung (Eulersches Polygonzugverfahren). Durch

$$\Phi(x,y,h) = \underbrace{f(x,y)}_{y'(x)} + \frac{h}{2}\underbrace{(f_x(x,y) + f_y(x,y)f(x,y))}_{y''(x)} \tag{2}$$

erhählt man ein Verfahren zweiter Ordnung. Und durch

$$\begin{aligned}\Phi(x,y,h) &= \underbrace{f(x,y)}_{y'(x)} + \frac{h}{2}\underbrace{(f_x(x,y) + f_y(x,y)f(x,y))}_{y''(x)} \\ &+ \frac{h^2}{6}\underbrace{(f_{xx}(x,y) + 2f_{xy}(x,y)f(x,y) + f_{yy}(x,y)f(x,y)^2 + f_y(x,y)[f_x(x,y) + f_y(x,y)f(x,y)])}_{y'''(x)}\end{aligned} \tag{3}$$

erhält man ein Verfahren dritter Ordnung.

Die Brauchbarkeit dieser Verfahren in der Praxis hängt wesentlich davon ab, wie kompliziert die auftretenden partiellen Ableitungen sind.

7.6 Runge-Kutta-Verfahren der Konsistenzordnung 2

Allgemeines Runge-Kutta-Verfahren der Konsistenzordnung 2

Bei der Aufstellung von Runge-Kutta-Verfahren geht man ähnlich vor wie bei der Methodes des Taylorabgleichs. Für $\beta \neq 0$ wird durch die Verfahrensfunktion

$$\Phi(x,y,h) = (1-\beta)f(x,y) + \beta f\big(x + \frac{h}{2\beta}, y + \frac{h}{2\beta}f(x,y)\big)$$

ein Runge-Kutta-Verfahren definiert. Man stellt fest, dass dieses Verfahren für jeden Wert von $\beta \neq 0$ die Konsistenzordnung 2 besitzt.

Spezialfälle

Für $\beta = \frac{1}{2}$ erhält man das Vefahren von Heun mit

$$\Phi(x, y, h) = \frac{1}{2}\Big(f(x, y) + f\big(x + h, y + hf(x, y)\big)\Big)$$

und für $\beta = 1$ das Halbschrittverfahren mit

$$\Phi(x, y, h) = f\Big(x + \frac{1}{2}h, y + \frac{1}{2}hf(x, y)\Big)$$

Ist f von y unabhängig, so liefern diese beiden Verfahren die Sehnentrapezregel bzw. die Mittelpunktregel.

7.7 Allgemeine Runge-Kutta-Verfahren

Mehrstufige Runge-Kutta-Verfahren (R-stufige Verfahren)

Mit

$$\begin{aligned} k_1 &= f(x, y)\,, \\ k_r &= f(x + ha_r, y + h \cdot \sum_{s=1}^{r-1} b_{rs} k_s)\,, \quad r = 2, \ldots, R, \end{aligned}$$

wobei

$$a_r = \sum_{s=1}^{r-1} b_{rs}\,, \quad r = 2, \ldots, R,$$

wird durch die Verfahrensfunktion

$$\Phi(x, y, h) = \sum_{r=1}^{R} c_r k_r$$

ein R-stufiges Runge-Kutta-Verfahren definiert. Für konsistente Verfahren gilt notwendig

$$\sum_{r=1}^{R} c_r = 1. \qquad \text{(Quadratur!)}$$

Man beschreibt solche Verfahren durch das Schema

$$\begin{array}{c|ccccccc} 0 & & & & & & & \\ a_2 & b_{21} & & & & & & \\ a_3 & b_{31} & b_{32} & & & & & \\ a_4 & b_{41} & b_{42} & b_{43} & & & & \\ \vdots & \vdots & \vdots & \vdots & & & & \\ a_R & b_{R1} & b_{R2} & b_{R3} & \ldots & b_{RR-1} & & \\ \hline & c_1 & c_2 & c_3 & \ldots & c_{R-1} & c_R \end{array}$$

Beispiele:

(1) 1-stufiges Runge-Kutta-Verfahren

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array}$$

Eulersches Polygonzugverfahren

(2) 2-stufige Runge-Kutta-Verfahren

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2\beta} & \frac{1}{2\beta} & \\ \hline & 1-\beta & \beta \end{array}$$

mit $\beta \neq 0$: allgemeines 2-stufiges Runge-Kutta-Verfahren

speziell:

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array}$$

modifiziertes Polygonzugverfahren (Halbschrittverfahren)

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Verfahren von Heun

(3) 3-stufige Runge-Kutta-Verfahren

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{3} & \frac{1}{3} & & \\ \frac{2}{3} & 0 & \frac{2}{3} & \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array}$$

Verfahren dritter Ordnung von Heun

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 1 & -1 & 2 & \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

Verfahren dritter Ordnung von Kutta

(4) 4-stufige Runge-Kutta-Verfahren

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

"Klassisches" Runge-Kutta-Verfahren

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{3} & \frac{1}{3} & & & \\ \frac{2}{3} & -\frac{1}{3} & 1 & & \\ 1 & 1 & -1 & 1 & \\ \hline & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{array}$$

$\frac{3}{8}$*-Regel*

3-stufige Runge-Kutta-Verfahren der Konsistenzordnung 3

Jede Lösung des algebraischen Gleichungssystems

$$\begin{aligned} c_1 + c_2 + c_3 &= 1 \\ c_2 a_2 + c_3 a_3 &= \frac{1}{2} \\ c_2 a_2^2 + c_3 a_3^2 &= \frac{1}{3} \\ c_3 b_{32} a_2 &= \frac{1}{6} \end{aligned}$$

definiert ein 3-stufiges Runge-Kutta-Verfahren

$$\begin{array}{c|ccc} 0 & & & \\ a_2 & a_2 & & \\ a_3 & a_3 - b_{32} & b_{32} & \\ \hline & c_1 & c_2 & c_3 \end{array}$$

der Konsistenzordnung 3.

Dieses Gleichungssystem besitzt eine zweiparametrige Lösungsschar. Unter der Zusatzvoraussetzung $a_2 = a_3$ ergibt sich eine einparametrige Schar von Runge-Kutta-Verfahren:

$$\begin{array}{c|ccc} 0 & & & \\ \frac{2}{3} & \frac{2}{3} & & \\ \frac{2}{3} & \frac{2}{3} - \frac{1}{4\gamma} & \frac{1}{4\gamma} & \\ \hline & \frac{1}{4} & \frac{3}{4} - \gamma & \gamma \end{array}$$

4-stufige Runge-Kutta-Verfahren der Konsistenzordnung 4

4-stufige Runge-Kutta-Verfahren

$$\begin{array}{c|cccc} 0 & & & & \\ a_2 & b_{21} & & & \\ a_3 & b_{31} & b_{32} & & \\ a_4 & b_{41} & b_{42} & b_{43} & \\ \hline & c_1 & c_2 & c_3 & c_4 \end{array}$$

der Konsistenzordnung 4 werden beschrieben durch das algebraische System

$$\begin{aligned} & a_2 = b_{21} \\ & a_3 = b_{31} + b_{32} \\ & a_4 = b_{41} + b_{42} + b_{43} \\ & c_1 + c_2 + c_3 + c_4 = 1 \\ & a_2 c_2 + a_3 c_3 + a_4 c_4 = \frac{1}{2} \\ & a_2^2 c_2 + a_3^2 c_3 + a_4^2 c_4 = \frac{1}{3} \\ & a_2^3 c_2 + a_3^3 c_3 + a_4^3 c_4 = \frac{1}{4} \\ & a_3 b_{43} c_4 + a_2 b_{42} c_4 + a_2 b_{32} c_3 = \frac{1}{6} \\ & a_3 a_4 b_{43} c_4 + a_2 a_4 b_{42} c_4 + a_2 a_3 b_{32} c_3 = \frac{1}{8} \\ & a_3^2 b_{43} c_4 + a_2^2 b_{42} c_4 + a_2^2 b_{32} c_3 = \frac{1}{12} \\ & a_2 b_{32} b_{43} c_4 = \frac{1}{24}\,. \end{aligned}$$

Erreichbare Konsistenzordnung

Bezeichnet man mit $p^*(R)$ die erreichbare Konsistenzordnung eines R-stufigen Runge-Kutta-Verfahrens, so gilt

$$p^*(R) \begin{cases} = R & \text{für } R = 1, 2, 3, 4\,, \\ = R - 1 & \text{für } R = 5, 6, 7\,, \\ = R - 2 & \text{für } R = 8, 9\,, \\ \leq R - 2 & \text{für } R \geq 10\,. \end{cases}$$

7.8 Konvergenz von Einschrittverfahren und eine Fehlerabschätzung

Konvergente Einschrittverfahren

Ein Einschrittverfahren heißt konvergent in einem Rechteck $R := [x_0, b] \times [c, d]$, wenn für alle Anfangswertprobleme (f sei auf R bezüglich y Lipschitz stetig)

$$y' = f(x, y), \quad y(x_0) = y_0 \,,$$

deren exakte Lösung ganz in R verläuft, stets gilt

$$\lim_{\substack{h \to 0 \\ x_0 + n \cdot h = x}} y_n = y(x) \quad \text{gleichmäßig für } x \in [x_0, b].$$

Konvergenz und Konsistenz

Die Zuwachsfunktion Φ mit

$$\Phi : [x_0, b] \times [c, d] \times [0, h_0] \to \mathbb{R}, \quad h_0 > 0$$

sei stetig differenzierbar und genüge einer Lipschitzbedingung bezüglich der zweiten Variablen y der Form

$$|\Phi(x, y, h) - \Phi(x, \tilde{y}, h)| \leq L|y - \tilde{y}| \,.$$

Dann sind die folgenden beiden Aussagen äquivalent:

(1) Das durch Φ definierte Verfahren ist konsistent.

(2) Das durch Φ definierte Verfahren ist konvergent im Rechteck $R := [x_0, b] \times [c, d]$.

Fehlerabschätzung für den Verfahrensfehler

Die Zuwachsfunktion Φ sei stetig in $S := [x_0, b] \times [c, d] \times [0, h_0]$ und genüge einer Lipschitzbedingung der Form

$$|\Phi(x, y, h) - \Phi(x, \tilde{y}, h)| \leq L|y - \tilde{y}| \,, \qquad (x, y, h), (x, \tilde{y}, h) \in S \,.$$

Für den lokalen (relativen) Diskretisierungsfehler gelte die Abschätzung

$$|\theta(x_\nu, y_\nu, h)| \leq Dh^r \,,$$

d.h. das Verfahren habe die Konsistenzordnung r. Dann gilt für $0 < h \leq h_0$ die a-priori-Fehlerabschätzung

$$|y(x_\nu) - y_\nu| \leq e^{L(x_\nu - x_0)}|y(x_0) - y_0| + \frac{e^{L(x_\nu - x_0)} - 1}{L} Dh^r \,, \quad \nu = 1, \ldots, n \,.$$

7.9 Einschrittverfahren in der Praxis

Schrittweitensteuerung

Praktisch wird die Schrittweite h meistens während der Rechnung verändert, d.h. man hat Schrittweiten h_ν ($\nu = 1, 2, \ldots$) statt einer von vornherein fest gewählten Schrittweite h. Zur aktuellen Wahl der Schrittweite h_ν werden Schrittweitensteuerungs-Algorithmen benutzt.

Kapitel 8

Finite Differenzen Verfahren zur Lösung von Randwertproblemen

8.1 Diskretisierung bei Randwertproblemen für gewöhnliche DGlen

In diesem Abschnitt erläutern wir exemplarisch an einer besonders einfachen Situation, wie man mit Hilfe von Differenzenapproximationen der Ableitungen numerische Lösungen von Randwertaufgaben gewinnen kann.

Wir gehen aus von einem *Randwertproblem für eine gewöhnliche Differentialgleichung zweiter Ordnung* des folgenden, beispielhaften Typs

$$\begin{aligned} y'' - y &= 0 \text{ auf } (0,1) \\ y(0) &= 1 \\ y(1) &= \frac{1}{2}(e + \frac{1}{e}) = \cosh(1). \end{aligned}$$

Gesucht ist eine zweimal stetig differenzierbare Funktion

$$y : [0,1] \to \mathbb{R},$$

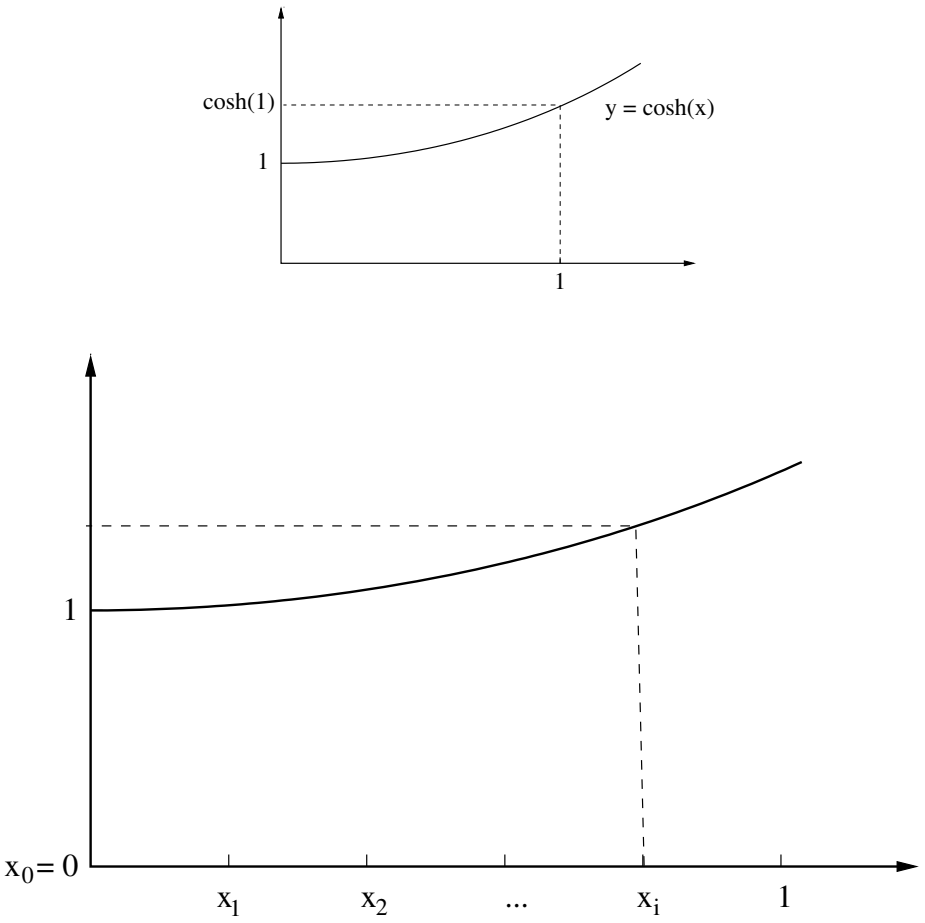
die der Differentialgleichung und den Randbedingungen genügt. Man kann die Lösung dieses Problems leicht erraten; es gilt

$$y(x) = \cosh(x)$$

Aufgabe 8.1. *Bestimmen Sie konstruktiv die Lösung des betrachteten Randwertproblems, indem Sie zunächst die Lösungsgesamtheit der Differentialgleichung bestimmen und anschließend daraus diejenige(n) Lösung(en) aussondern, welche außerdem den Randbedingungen genügen.*

Wir wollen nun ein numerisches Verfahren entwickeln, um Näherungslösungen der gestellten Randwertaufgabe zu gewinnen. Zu diesem Zweck *diskretisieren* wir das Problem und unterteilen das Einheitsintervall $[0,1]$ äquidistant in $N+1$ Teilintervalle gemäß

$$\begin{aligned} x_i &= ih, \qquad i = 0, ..., N+1, \\ h &:= \frac{1}{N+1}. \end{aligned}$$



Gesucht sind die Werte

$$y(x_i), \quad i = 0, ..., N+1;$$

dabei sind uns allerdings wegen der gegebenen Randbedingungen die beiden Werte

$$y(x_0) = 1,$$
$$y(x_{N+1}) = \cosh(1)$$

bereits bekannt. Wir approximieren nun die Ableitungswerte

$$y''(x_i), \quad i = 1, ..., N,$$

mit Hilfe der sogenannten *zweiten Differenzen* gemäß

$$y''(x_i) = \frac{y(x_{i-1}) - 2y(x_i) + y(x_{i+1})}{h^2} + r_i(h).$$

Dabei bezeichnet $r_i(h)$ den Fehler zur Schrittweite h. Es gilt wegen des Taylorschen Satzes

$$r_i(h) = O(h^2) \quad \text{für } h \to 0,$$

denn aus

$$y(x_i - h) = y(x_i) - hy'(x_i) + \frac{h^2}{2}y''(x_i) - \frac{h^3}{6}y'''(x_i) + O(h^4)$$
$$y(x_i + h) = y(x_i) + hy'(x_i) + \frac{h^2}{2}y''(x_i) + \frac{h^3}{6}y'''(x_i) + O(h^4)$$

folgt

$$\frac{y(x_{i-1}) - 2y(x_i) + y(x_{i+1})}{h^2} = y''(x_i) + O(h^2).$$

Wir setzen die obige Entwicklung von $y''(x_i)$ in die Differentialgleichung ein und erhalten an den Stellen x_i, $i = 1, ..., N$, die folgenden Gleichungen

$$\frac{y(x_{i-1}) - 2y(x_i) + y(x_{i+1})}{h^2} - y(x_i) + r_i(h) = 0, \qquad i = 1, ..., N,$$

wobei zusätzlich

$$\begin{aligned} y(x_0) &= 1, \\ y(x_{N+1}) &= \cosh(1) \end{aligned}$$

gilt. Wenn wir nun die Fehlergrößen $r_i(h)$ vernachlässigen, so wird dieses Gleichungssystem im Allgemeinen nicht mehr von den Werten $y(x_i)$ erfüllt. Mit Näherungswerten

$$y_i \approx y(x_i), \quad i = 1, ..., N,$$

und

$$\begin{aligned} y_0 &= y(x_0) \\ y_{N+1} &= y(x_{N+1}) \end{aligned}$$

erhalten wir so das lineare Gleichungssystem

$$\begin{aligned} \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} - y_i = 0, \qquad & i = 1, ..., N \\ y_0 = 1, & \\ y_{N+1} = \cosh(1). & \end{aligned}$$

Wir setzen die bekannten Randwerte in die erste und die N-te Gleichung ein und formen gleichzeitig etwas um. Dies liefert

$$\begin{aligned} (2 + h^2)y_1 - y_2 &= 1, \\ -y_{i-1} + (2 + h^2)y_i - y_{i+1} &= 0, \qquad i = 2, ..., N - 1 \\ -y_{N-1} + (2 + h^2)y_N &= \cosh(1). \end{aligned}$$

Nach Multiplikation jeder Gleichung mit

$$\tau = \frac{1}{2 + h^2}$$

hat das lineare Gleichungssystem die folgende Gestalt:

$$\begin{pmatrix} 1 & -\tau & 0 & 0 & ... & 0 \\ -\tau & 1 & -\tau & 0 & ... & 0 \\ 0 & -\tau & 1 & -\tau & & 0 \\ \vdots & & \ddots & & & \vdots \\ 0 & & ... & -\tau & 1 & -\tau \\ 0 & & ... & & -\tau & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \\ \vdots \\ \\ y_N \end{pmatrix} = \begin{pmatrix} \tau \\ 0 \\ \\ \vdots \\ \\ \tau \cosh(1) \end{pmatrix}$$

Resultat 8.2. *Diskretisiert man das Randwertproblem*

$$\begin{aligned} y'' - y &= 0, \\ y(0) &= 1, \\ y(1) &= \cosh(1) \end{aligned}$$

mit Hilfe von zweiten Differenzen, so erhält man das lineare Gleichungssystem

$$\begin{pmatrix} 1 & -\tau & 0 & 0 & ... & 0 \\ -\tau & 1 & -\tau & 0 & ... & 0 \\ \vdots & & \ddots & & & \vdots \\ 0 & & ... & -\tau & 1 & -\tau \\ 0 & & ... & & -\tau & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ \\ y_N \end{pmatrix} = \begin{pmatrix} \tau \\ 0 \\ \vdots \\ \\ \tau \cosh(1) \end{pmatrix}$$

wobei

$$\begin{aligned} \tau &:= \frac{1}{2+h^2}, \\ h &:= \frac{1}{N+1}, \end{aligned}$$

und y_i, $i = 1, ..., N$ Näherungswerte für $y(ih)$ sind.

Dieses lineare Gleichungssystem hat einige leicht zu verifizierende Eigenschaften, die wir in folgendem Resultat festhalten.

Resultat 8.3. *Die Matrix A des obigen linearen Gleichungssystems hat folgende wichtige Eigenschaften:*

1. *A is reell symmetrisch.*
2. *A hat Tridiagonalgestalt.*
3. *A ist strikt diagonaldominant.*
4. *Die Eigenwerte von A liegen im Intervall $[1 - 2\tau, 1 + 2\tau] \subset (0, 2)$, wie man mit Hilfe des Gerschgorinschen Kreissatzes erkennt. Insbesondere ist A positiv definit.*

Da A die hier aufgelisteten Eigenschaften besitzt, sind offensichtlich einige der uns bekannten Lösungsmethoden für lineare Gleichungssysteme anwendbar wie z.B. die Lösung mit Hilfe der Cholesky-Zerlegung.

8.2 Diskretisierung bei Randwertproblemen für partiellen Differentialgleichungen

Exemplarisch behandeln wir in diesem Zusammenhang das sogenannte "Modellproblem", das, wie schon sein Name besagt, in der Literatur häufig herangezogen wird, um die Wirksamkeit eines Diskretisierungsverfahrens zu demonstrieren oder um verschiedene Näherungsmethoden zu vergleichen.

Wir betrachten das sogenannte *Dirichletproblem* für ein Quadrat. Darunter versteht man die folgende **Problemstellung:**

Gesucht wird eine zweimal stetig differenzierbare Funktion

$$u = u(x, y),$$

die im Innern des Quadrates

$$Q := \{(x, y) : 0 \leq x, y \leq 1\}$$

der Laplace-Gleichung

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

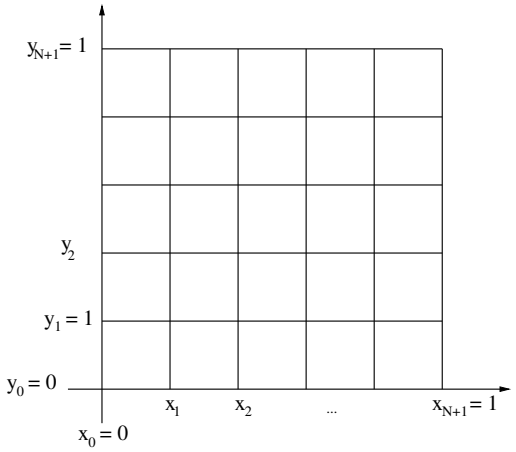
genügt und auf dem Rand ∂Q von Q vorgeschriebene Werte

$$u(x, y) = f(x, y), \quad (x, y) \in \partial Q,$$

annimmt. Beispiele für Lösungen der Laplace-Gleichung sind z.B. elektrische Potentiale von Ladungsverteilungen, falls man die Potentiale außerhalb des Bereichs betrachtet, in dem die Ladungen konzentriert sind.

Ein Näherungsverfahren zur Lösung des Modellproblems entwickeln wir wiederum mit Hilfe von Diskretisierungen. Dazu überdecken wir Q mit einem Gitter der Maschenweite

$$h = \frac{1}{N+1}.$$



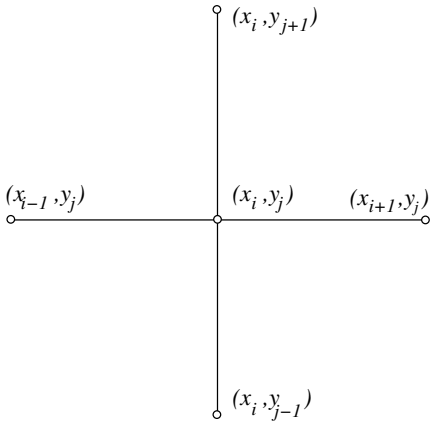
Gesucht sind die Funktionswerte

$$\tilde{u}_{ij} := u(x_i, y_j), \quad i, j = 1, ..., N.$$

Näherungen dafür erhalten wir als Lösungen eines linearen Gleichungssystems, das wir mit Hilfe von Diskretisierung gewinnen. Wir ersetzen dabei die partiellen Ableitungen

$$\frac{\partial^2 u}{\partial x^2}\Big|_{(x_i, y_j)}, \quad \frac{\partial^2 u}{\partial y^2}\Big|_{(x_i, y_j)}$$

durch zweite Differenzen. Dazu betrachten wir einen "Stern" im obigen Gitter:



Mit Hilfe von zweiten Differenzen folgt analog zum eindimensionalen Fall

$$\frac{\partial^2 u}{\partial x^2}\Big|_{(x_i,y_j)} = \frac{\tilde{u}_{i-1,j} - 2\tilde{u}_{ij} + \tilde{u}_{i+1,j}}{h^2} + O(h^2)$$

und

$$\frac{\partial^2 u}{\partial y^2}\Big|_{(x_i,y_j)} = \frac{\tilde{u}_{i,j-1} - 2\tilde{u}_{ij} + \tilde{u}_{i,j+1}}{h^2} + O(h^2)$$

für $h \to 0$. Vernachlässigen wir nun die $O(h^2)$-Terme, so erhalten wir das lineare Gleichungssystem

$$u_{i-1,j} + u_{i,j-1} - 4u_{ij} + u_{i,j+1} + u_{i+1,j} = 0, \qquad i,j = 1,...,N,$$

wobei zusätzlich die Randbedingungen

$$u_{ij} = f(x_i, y_j) =: f_{ij} \quad \text{für } i \text{ oder } j \in \{0, N+1\}$$

zu erfüllen sind. Setzt man diese bekannten Randwerte in das obige lineare Gleichungssystem ein, so erhält man ein lineares Gleichungssystem von N^2 Gleichungen in den N^2 Unbekannten u_{ij} mit $i,j = 1,...,N$. Da in jeder Gleichung höchstens 5 Unbekannte miteinander durch nichtverschwindende Koeffizienten verbunden sind, enthält die Koeffizientenmatrix des betrachteten linearen Gleichungssystems sehr viele Nullen. Es liegt eine sogenannte "sparse matrix" vor.

Resultat 8.4. *Durch Diskretisieren des Modellproblems*

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} &= 0 \qquad \textit{für } (x,y) \in Q \\ u(x,y) &= f(x,y) \quad \textit{für } (x,y) \in \partial Q \end{aligned}$$

mit Hilfe von zweiten dividierten Differenzen erhält man das lineare Gleichungssystem

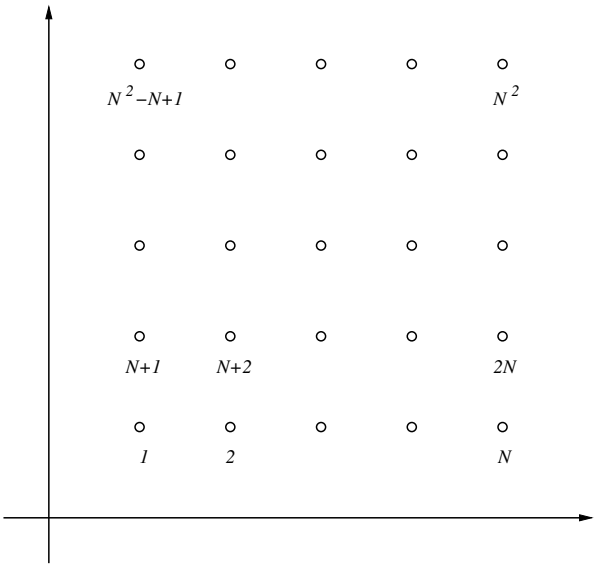
$$\begin{aligned} u_{i-1,j} + u_{i,j-1} - 4u_{ij} + u_{i,j+1} + u_{i+1,j} &= 0, \qquad i,j = 1,...,N, \\ u_{ij} = f_{ij} \qquad &\textit{falls } i \in \{0, N+1\} \textit{ oder } j \in \{0, N+1\}. \end{aligned}$$

Während bei dem oben betrachteten eindimensionalen Randwertproblem die Indizierung der Unbekannten y_i, $i = 1,...,N$, in natürlicher Weise vorgegeben ist, kann man

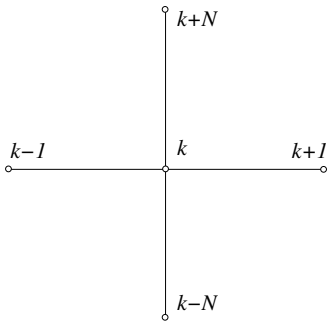
beim Modellproblem nicht in so eindeutiger Weise eine speziell Anordnung der Indexmengen $\{(i,j) : i,j = 1,...,N\}$ als besonders bevorzugt ansehen. Wir werden eine spezielle Nummerierung der Unbekannten untersuchen, die zu in gewisser Weise ausgezeichneter Struktur der Koeffizientenmatrix führt. Dabei denken wir uns die Gleichungen, die Randterme enthalten, in die übrigen eingesetzt.

Nummerierung in Richtung der Koordinatenachsen

Zunächst wird man an die Nummerierung der Unbekannten in Richtung der Koordinatenachsen denken, d.h. man ordnet die Indizes der Unbekannten u_{ij} aus Resultat 8.4 in der Form $(u_{11}, u_{21}, ..., u_{N1}, u_{12}, ..., u_{N2}, ..., u_{NN})$:



Bei dieser Nummerierung ergibt sich offensichtlich eine Bandstruktur für die Gleichungsmatrix, da jede Unbekannte höchstens mit ihren beiden direkten Nachbarn und den beiden mit einer Nummer N höher oder niedriger durch einen nichtverschwindenden Koeffizienten verknüpft ist:



Man erhält folgende Gleichungsmatrix:

$$\left(\begin{array}{cccc|cccc|cccc|cccc}
-4 & 1 & 0 & 0 & 1 & & & & & & & & & & & \\
1 & \ddots & \ddots & & & \ddots & & & & & & & & & & \\
 & \ddots & \ddots & 1 & & & \ddots & & & & & & & & & \\
 & & 1 & -4 & & & & 1 & & & & & & & & \\
\hline
1 & & & & -4 & 1 & 0 & 0 & & & & & & & & \\
 & \ddots & & & 1 & \ddots & \ddots & & & & & & & & & \\
 & & \ddots & & & \ddots & \ddots & 1 & & & & & & & & \\
 & & & 1 & & & 1 & -4 & & & & & & & & \\
\hline
 & & & & & & & & \ddots & & & & 1 & & & \\
 & & & & & & & & & \ddots & & & & \ddots & & \\
 & & & & & & & & & & \ddots & & & & \ddots & \\
 & & & & & & & & & & & \ddots & & & & 1 \\
\hline
 & & & & & & & & 1 & & & & -4 & 1 & 0 & 0 \\
 & & & & & & & & & \ddots & & & 1 & \ddots & \ddots & \\
 & & & & & & & & & & \ddots & & & \ddots & \ddots & 1 \\
 & & & & & & & & & & & 1 & & & 1 & -4
\end{array}\right)$$

Die Matrix zerfällt in natürlicher Weise in N^2 Blöcke, die ebenfalls eine besonders einfache Struktur besitzen: Bezeichnen wir die auftretenden Blöcke mit $A_{\nu\mu}$, $\nu, \mu = 1, ..., N$ so gilt

$$A_{\nu\nu} = \begin{pmatrix}
-4 & 1 & 0 & 0 & ... & 0 \\
1 & \ddots & \ddots & 0 & ... & 0 \\
0 & & \ddots & & & \vdots \\
\vdots & & & \ddots & \ddots & 1 \\
0 & & ... & & 1 & -4
\end{pmatrix}$$

$A_{\nu\nu}$ ist also eine Tridiagonalmatrix sehr einfacher Bauart. Weiter gilt

$$A_{\nu,\nu+1} = A_{\nu+1,\nu} = \begin{pmatrix}
1 & 0 & ... & 0 \\
0 & \ddots & & 0 \\
0 & ... & 0 & 1
\end{pmatrix}, \qquad \nu = 1, ..., N-1$$

sowie $A_{\nu\mu} = (0)_{N\times N'}$ für $|\nu - \mu| \geq 2$. Von den N^2 Elementen sind also "fast alle" Null; es treten nur

$$N(N + 2(N-1)) + 2(N-1)N = 5N^2 - 4N$$

von Null verschiedene Elemente auf.

Kapitel 9

Finite Elemente

9.1 Schwache Formulierung von Randwertaufgaben

Betrachten wir erneut die Randwertaufgabe

$$-\underbrace{(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2})}_{=:\Delta u} = f \quad \text{in } \Omega, \qquad u = 0 \quad \text{auf } \partial\Omega \tag{9.1}$$

mit gegebenem (z.B. stetigem) $f : \Omega \to \mathbb{R}$, wobei $\Omega \subset \mathbb{R}^2$ ein polygonales Gebiet sei.

Zuerst leiten wir die sogenannten *schwache Formulierung* der obigen Randwertaufgabe her. Dazu nehmen wir an, dass u eine auf $\overline{\Omega}$ stetige und in Ω zweimal stetig differenzierbare Lösung von (9.1) ist.

Sei $V \subset C(\overline{\Omega})$ der Raum der stetigen Funktionen, die in Ω *stückweise* stetig differenzierbar und auf $\partial\Omega$ gleich Null sind. Multiplizieren wir die Gleichung (9.1) mit einer Funktion $\varphi \in V$ so ergibt sich

$$\int_\Omega (-\Delta u)(x,y)\varphi(x,y)\,d(x,y) = \int_\Omega f(x,y)\varphi(x,y)\,d(x,y).$$

Mit dem Gaußschen Integralsatz und der Ausnutzung der Randbedingung $u = 0$ auf $\partial\Omega$ folgt

$$\begin{aligned}
&\int_\Omega (-\Delta u)(x,y)\varphi(x,y)\,d(x,y)\\
&= \int_\Omega \nabla u(x,y)\nabla\varphi(x,y)\,d(x,y) - \int_{\partial\Omega} \frac{\partial u}{\partial \nu}(x,y)\,\underbrace{\varphi(x,y)}_{=0 \text{ auf } \partial\Omega}\,d(x,y)\\
&= \int_\Omega \nabla u(x,y)\nabla\varphi(x,y)\,d(x,y)\\
&= \int_\Omega f(x,y)\varphi(x,y)\,d(x,y) \quad \text{für jedes } \varphi \in V.
\end{aligned}$$

Die resultierende Integralbeziehung

$$\int_\Omega \nabla u(x,y)\nabla\varphi(x,y)\,d(x,y) = \int_\Omega f(x,y)\varphi(x,y)\,d(x,y) \quad \text{für jedes } \varphi \in V \tag{9.2}$$

betrachten wir als "Ersatz" für die Differentialgleichung (9.1). Wir nennen daher (9.2) die *schwache* Formulierung von (9.1). Es lässt sich mit Mitteln der Funktionalanalysis zeigen, dass (9.2) eine eindeutige Lösung $u \in V$ hat. Dazu benutzt man die Theorie der *Sobolevräume*.

Nun führen wir zur Abkürzung folgende Bezeichnungen ein. Für $u, v \in V$ sei

$$a(u,v) := \int_\Omega \nabla u(x,y) \nabla v(x,y)\, d(x,y)$$
$$b(v) := \int_\Omega f(x,y) v(x,y)\, d(x,y).$$

Damit ist eine Bilinearform $a : V \times V \to \mathbb{R}$ und eine Linearform $b : V \to \mathbb{R}$ definiert. Die schwache Formulierung (9.2) der Randwertaufgabe (9.1) lässt sich somit wie folgt leichter schreiben: finde $u \in V$ so dass gilt

$$a(u,\varphi) = b(\varphi) \text{ für alle } \varphi \in V. \tag{9.3}$$

Das Auffinden einer Lösung von (9.3) ist ein unendlichdimensionales Problem, da der Raum V ein unendlichdimensionaler Funktionenraum ist. Als Nächstes werden wir eine endlichdimensionale Approximation an (9.3) suchen.

9.2 Ritz-Galerkin Approximation

Im Raum V der stetigen und in Ω *stückweise* stetig differenzierbaren Funktionen, die auf $\partial\Omega$ gleich Null sind, wählen wir einen k-dimensionalen Unterraum $V_k \subset V \subset C(\overline{\Omega})$. Zur näherungsweisen Lösung der schwachen Formulierung (9.3) der Randwertaufgabe (9.1) suchen wir nun nach einer Lösung $\tilde{u} \in V_k$, die

$$\int_\Omega \nabla \tilde{u}(x,y) \nabla \varphi(x,y)\, d(x,y) = \int_\Omega f(x,y)\varphi(x,y)\, d(x,y) \quad \text{für jedes } \varphi \in V_k \tag{9.4}$$

erfüllt. Eine äquivalente Formulierung ist: finde $\tilde{u} \in V_k$ mit

$$a(\tilde{u},\varphi) = b(\varphi) \text{ für alle } \varphi \in V_k. \tag{9.5}$$

Wir heben zwei wesentliche Unterschiede von (9.4) bzw. (9.5) im zu Gegensatz zu (9.2) bzw. (9.3) hervor:

1. Gesucht wirde eine Lösung $\tilde{u}$ im endlichdimensionalen Unterraum V_k.
2. Die Bedingung $a(\tilde{u},\varphi) = b(\varphi)$ wird nur für die Elemente φ des endlichdimensionalen Unterraumes V_k gefordert.

Welche Gestalt hat eine solche Lösung $\tilde{u}$? Dazu wählen wir eine Basis $\{\varphi_1, ..., \varphi_k\}$ von V_k und betrachten den Ansatz

$$\tilde{u} = \sum_{j=1}^{k} \alpha_j \varphi_j$$

mit $\alpha_1, ..., \alpha_k \in \mathbb{R}$. Eingesetzt in (9.4) erhalten wir

$$\sum_{j=1}^{k} \alpha_j \int_\Omega \nabla\varphi_j(x,y)\nabla\varphi_l(x,y)\,d(x,y) = \int_\Omega f(x,y)\varphi_l(x,y)\,d(x,y), \qquad l = 1, ..., k$$

beziehungsweise

$$\sum_{j=1}^{k} \alpha_j a(\varphi_j, \varphi_l) = b(\varphi_l) \text{ für alle } l = 1, \ldots, k.$$

Dabei handelt es sich offenbar um ein lineares Gleichungssystem in den k Unbekannten $z = (\alpha_1, \ldots, \alpha_k)^T$

$$(*) \qquad Az = b$$

wobei

$$A_{jl} = A_{lj} = a(\varphi_j, \varphi_l) = \int_\Omega \nabla\varphi_j(x,y)\nabla\varphi_l(x,y)\,d(x,y), \qquad j, l = 1, ..., k$$
$$b_l = b(\varphi_l) = \int_\Omega f(x,y)\varphi_l(x,y)\,d(x,y), \qquad l = 1, ..., k$$

Die Matrix $A = (A_{jl})_{j,l=1}^n$ heisst *Steifigkeitsmatrix* und der Vektor $b = (b_1, \ldots, b_k)^T$ heisst *Lastvektor*. Man rechnet leicht nach, dass A positiv definit ist. Dieses Verfahren, d.h. die Wahl des k-dimensionalen Unterraums, die Wahl der Basis und die anschließende Bestimmung der Unbekannten $(\alpha_1, \ldots, \alpha_k)^T$ durch Lösung des linearen Gleichungssystems $(*)$ nennt man *Ritz-Galerkin-Verfahren.*

Eine der wichtigsten Fragen beim Ritz-Galerkin-Verfahren ist die folgende: wie gut approximiert die Lösung $\tilde{u}$ von (9.4) die exakte Lösung u von (9.2)? Eine Antwort darauf wird im folgenden Satz gegeben.

Satz 9.1 (Céas Lemma)**.**
Ist $u \in V$ eine Lösung von (9.2) *und $\tilde{u} \in V_k$ eine Lösung von* (9.4)*, so gilt mit der sogenannten Energienorm $\|\cdot\|_a := \sqrt{a(\cdot,\cdot)}$ die Abschätzung:*

$$\|\tilde{u} - u\|_a = \min_{\varphi \in V_k} \|\varphi - u\|_a.$$

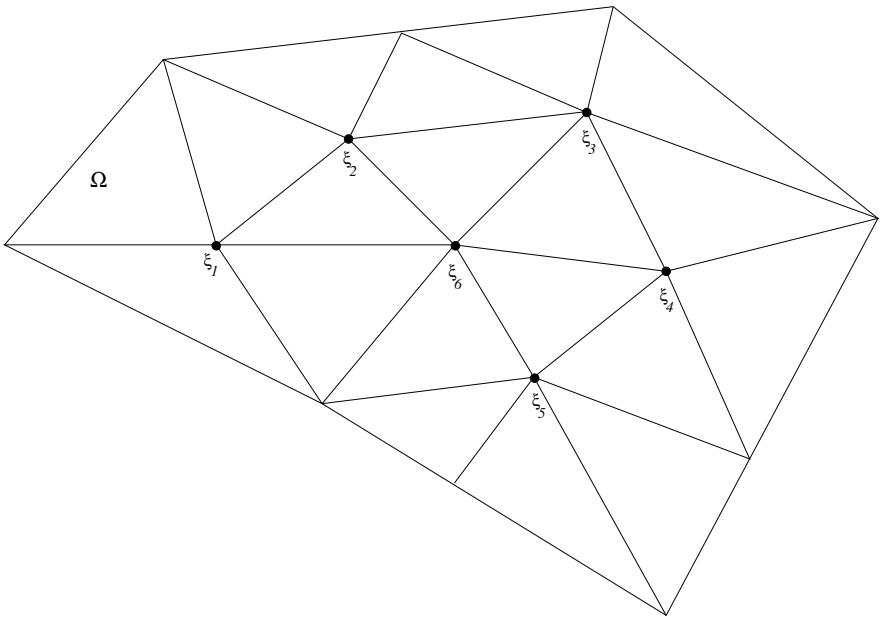
Céas Lemma besagt, dass die Ritz-Galerkin-Approximation $\tilde{u}$ unter allen anderen Elementen der Raumes V_k gemessen in der Energienorm $\|\cdot\|_a$ die optimale Näherung an die exakte Lösung u ist. Alle weiteren, konkreten Abschätzungen des Fehlers basieren auf dieser Erkenntnis.

9.3 Umsetzung des Ritz-Galerkin-Verfahrens

Offensichtlich sollte man die Dimension k der Unterraumes V_k möglichst gross wählen, um eine gute Approximation zu erreichen. Dafür hätte man gerne, dass A eine dünn besetzte Matrix ist. Dies erreicht man, indem man die Basisfunktionen so wählt, dass sie einen kleinen Träger haben. Dabei ist der Träger einer stetigen Funktion φ durch $\text{supp}(\varphi) = \overline{\{x \in \overline{\Omega} : \varphi(x) \neq 0\}}$ definiert.

Beachtet man, dass nur dann $A_{jl} \neq 0$ ist, wenn $\text{supp}(\varphi_j) \cap \text{supp}(\varphi_l) \neq \emptyset$ gilt, so sieht man, dass bei "kleinen" Trägern "viele" der Matrix-Einträge $A_{jl} = 0$ sind.

Im Folgenden betrachten wir eine spezielle Wahl des Unterraumes V_k und eine spezielle Konstruktion einer Basis. Dazu zerlegen wir das zweidimensionale Grundgebiet Ω geschickt in Dreiecke. Diesen Prozess nennt man *Triangulierung von* Ω. Die Triangulierung eines Grundgebietes kann z.B. wie im folgenden Bild aussehen:



Wir bezeichnen die *inneren* Knotenpunkte mit $\xi_1, ..., \xi_k$. Für $j = 1, ..., k$ sei die Funktion $\varphi_j : \overline{\Omega} \to \mathbb{R}$ definiert durch die folgenden drei Forderungen:

1. $\varphi_j(\xi_i) = \delta_{ij}$
2. auf jedem Dreick der Triangulierung sei $\varphi_j(x, y) = ax+by+c$ linear (a, b, c können dabei von Dreieck zu Dreieck verschieden sein)
3. $\varphi_j = 0$ auf $\partial\Omega$

Der Graph von φ_j ist also eine *Pyramide* mit Spitze in ξ_j, die in den Nachbarknoten von ξ_j auf 0 hinuntergeht und auf dem Rest der Menge Ω gleich Null ist.

Die Basisfunktionen φ_j heißen dann *Formfunktionen* (node shape functions). Mit V_k bezeichnen wir die lineare Hülle der Formfunktionen $\varphi_1, \ldots, \varphi_k$. Das Ritz-Galerkin-Verfahren angewandt auf diesen speziellen Unterraum V_k ist ein Beispiel für eine Näherungsmethode, die in der Literatur als *Finite-Elemente-Methode* bekannt ist.

Es gibt durch die Wahl anderer Unterräume zahlreiche weitere Beispiele für die Finite-Elemente-Methode. So kann man neben den stückweise linearen Basisfunktionen auch andere Basisfunktionen benutzen. Ebenso werden Zerlegungen des Grundgebiets Ω in Vierecke anstelle von Dreiecken betrachtet. Und im Fall von drei Raumdimensionen werden Zerlegungen in Tetraeder bzw. Prismen verwendet.